ParisTech
INSTITUT DES SCIENCES ET TECHNOLOGIES
PARIS INSTITUTE OF TECHNOLOGY

TELECOM ParisTech

2012-ENST-00xx

EDITE - ED 130

Doctorat ParisTech

T H È S E

pour obtenir le grade de docteur délivré par

TELECOM ParisTech

Spécialité « Communications et Électronique »

présentée et soutenue publiquement par

Josep TORRAS FLAQUER

le 12 décembre 2011

Méthodes probabilistes pour l'estimation de la fiabilité dans la logique combinatoire

Directeur de thèse : **Lirida NAVINER**
Co-encadrement de la thèse : **Jean-Marc DAVEAU**

Jury

M. Habib MEHREZ, Professeur, Université Paris VI	Président
M. Jean-Luc LERAY, Docteur, CEA	Rapporteur
M. Michael NICOLAIDIS, Professeur, TIMA	Rapporteur
M. Philippe ROCHE, Docteur, STMicroelectronics	Examinateur
Mme. Lirida NAVINER, Professeur, Telecom Paristech	Directeur de thèse
M. Jean-Marc DAVEAU, Docteur, STMicroelectronics	Co-directeur de thèse

TELECOM ParisTech
école de l'Institut Télécom - membre de ParisTech

THÈSE

Remerciements

Je voudrais commencer par remercier M. Jean-Luc Leray et M. Michael Nicolaidis, qui m'ont fait l'honneur d'avoir accepté le rôle de rapporteurs pour cette thèse. Leurs remarques ont sans aucun doute aidé à améliorer la qualité de ce manuscrit. Je voudrais aussi remercier M. Habib Mehrez pour avoir présidé le jury de cette thèse, ces questions lors de la soutenance m'ont aidé à enrichir la vision que j'avais de mon propre travail. J'exprime ici ma reconnaissance envers M. Philippe Roche, qui m'a fait l'honneur d'être examinateur de mon travail lors de la soutenance de thèse, mais surtout pour m'avoir accueilli aux sein de l'équipe RAD-HARD, et de m'avoir appris beaucoup choses pendant ces trois dernières années, en lien ou pas avec l'électronique ou l'extra-galactisme.

Cette thèse n'aurait jamais vu le jour sans le soutien de mes deux directeurs de thèse, Mme. Lirida Naviner et M. Jean-Marc Daveau, qui ont su m'encourager dans les moments de doute et avec qui j'ai pu partager les moments de joie. Je veux profiter de ces lignes pour leur témoigner de la confiance qu'ils m'ont faite, de tous les conseils qu'ils m'ont donnés et d'avoir su partager leur expérience et leur grandes qualités scientifiques avec moi. Si cette thèse a de la valeur, ce sera sans doute grâce à eux.

Pendant ces trois années, j'ai eu la chance de partager mon bureau avec Slawosz Uznanski. Ce serait très difficile pour moi de résumer toutes ses qualités en quelques mots. Qu'il sache qu'il a toute mon estime et mes amitiés. De même pour Fady Abouzeid, que Slawosz et moi avons rencontré un peu plus tard et qui est venu égayer nos semaines. Qui plus est, Fady aime bien mon humour, et il faut dire que ce n'est pas facile. Jamais je n'aurais pensé qu'à ma sortie de ST j'aurais gagné deux aussi bons amis. Je voudrais aussi remercier les autres membres de l'équipe RAD-HARD, Gilles Gasiot et Sylvain Clerc, leur bonne humeur et gentillesse ont beaucoup fait pour mon adaptation à l'équipe. Je les remercie aussi d'avoir toujours été disponibles pour partager leur sagesse et leur expérience avec moi. Je veux aussi remercier tous les gens qui ont fait partie de l'équipe ou avec qui j'ai travaillé, bu un café ou tout simplement discuté d'un problème d'estimation de la complexité STM : Fabian, Pascale, Dominique, Cyril, Can, Charly, Max et Carmelo.

Je veux aussi remercier mes collègues de l'école, qui m'ont très bien accueilli pendant chacune de mes visites : Guttenberg, Florient, Tian, Chantal, Arwa, Ting, Mouhammad et Waqqas. Si j'ai pu bien travailler lors des mes séjours à Paris est sans doute grâce à eux.

Je voudrais aussi remercier ma copine, Louise, pour toutes les relectures du manuscript qu'elle a faites, mais surtout, pour son soutien indéfectible et pour tous les moments passés ces dernières années. Louise est la personne qui a vécu de plus près tous mes moments d'angoisse, mais aussi tous mes moments du bonheur. J'ai été très chanceux de l'avoir près de moi.

Je voudrais finir par remercier ma famille, et je me permettrai de le faire en notre langue. Vull donar les gràcies als meus pares, que sempre m'han animat malgrat la distància, i que, sobretot, m'han inculcat valors sense els quals no hauria arribat aquí. També vull donar les

gràcies a la iaia, que és la que més pateix amb l'allunyament, espero que tot això valgui la pena. Vull donar les gràcies al meu germà, Xevi, i la seva dona, Leila, per tots els consells que m'han donat i pels ànims que m'han transmès. I per acabar, donar les gràcies al meu nebot, Damià, la seva arribada m'ha fet relativitzar les meves preocupacions i veure el final de la tesi des d'un angle different.

Abstract

Digital circuits used in such domains as automotive, medical, space or nuclear need to satisfy high reliability requirements. In addition, continuous downscaling of consumer electronics consisting in increased integration and lower voltage supply, affects system's sensitivity to several phenomena involved in transient and permanent faults generation : particle strike, thermal noise, crosstalk, etc. These faults may cause different effects in the system behavior ranging from silent fault (no effect) to a total loss of functionality.

Transient faults in memories and sequential elements have largely dominated the overall soft error rate (SER) of systems, thus, correction and prevention techniques for these devices are well known, and their application is widely spread. Though, it is expected that the contribution of combinational logic elements to the system's SER becomes dominant with CMOS technology downscaling. Hence, there is a need to fulfill the lack of available models and methodologies that take into account the combinational logic's contribution to the reliability loss. Two main approaches exist to assess this issue :

1. Fault Injection techniques.
2. Analytical models.

The work presented in this manuscript is focused on the analytical approach, also known as probabilistic approach. First, an in-depth analysis of the state of the art is done pointing out the main limitations of probabilistic models. Second, innovative heuristics and approaches are proposed, improving the performance of state of the art methods. Then, new metrics addressing the selective hardening problem and FMDEA analysis are investigated. Finally, we validate our approaches by comparing the performances of our methods with alternative techniques.

Table des matières

Introduction **1**

1 Fiabilité des circuits électroniques numériques **5**
1.1 Introduction . . . 5
1.2 Généralités sur la fiabilité . . . 6
1.2.1 Définitions . . . 6
1.2.2 Classification des défaillances . . . 6
1.2.3 Métrique de la fiabilité . . . 7
1.3 Perte de fiabilité dans les circuits numériques . . . 9
1.3.1 Rendement des procédés . . . 9
1.3.2 Fautes transitoires dans la logique . . . 10
1.4 Techniques de protection aux fautes . . . 15
1.5 Conclusion . . . 17

2 Techniques d'analyse de la fiabilité **19**
2.1 Introduction . . . 19
2.2 Analyse de la fiabilité des cellules élémentaires . . . 20
2.3 Analyse de la fiabilité de fonctions logiques complexes . . . 20
2.3.1 Analyse par injection de fautes . . . 21
2.3.2 Probabilité de propagation de fautes . . . 22
2.4 Conclusion . . . 42

3 Proposition de méthodes d'estimation de la fiabilité **43**
3.1 Introduction . . . 43
3.2 Estimation des probabilités des signaux . . . 43
3.2.1 Estimation de la probabilité jointe de signaux à partir des probabilités conditionnelles . . . 45
3.3 Modélisation par Matrices de Probabilités Conditionnées . . . 48
3.3.1 Calcul de probabilités conditionnées d'état d'un signal dans la logique combinatoire . . . 48
3.3.2 Calcul de la fiabilité à partir des matrices CPM . . . 53
3.3.3 Réduction de la taille des matrices CPM intermédiaires . . . 58
3.3.4 Estimation de la corrélation des entrées . . . 58
3.3.5 Méthode CPM approximée . . . 59
3.4 Méthode CPM hiérarchique . . . 63
3.4.1 Calcul de probabilités de sortie d'un bloc combinatoire en fonction des entrées . . . 63
3.4.2 Décomposition modulaire disjointe . . . 65

3.4.3 Décomposition modulaire non-disjointe 66
3.4.4 Estimation de la complexité 67

4 Application : durcissement sélectif et analyse FMDEA 69
4.1 Introduction 69
4.2 Durcissement sélectif 69
4.2.1 Introduction 69
4.2.2 Techniques de durcissement 70
4.2.3 Calcul de la sensibilité de la fiabilité aux fautes des portes 72
4.2.4 Calcul hiérarchique de la sensibilité 79
4.2.5 Etude de cas : Additionneur Brent-Kung de 32 bits 80
4.3 Analyse FMDEA 83
4.3.1 Introduction à la norme ISO 26262 83
4.3.2 Analyse de la borne inférieure de la fiabilité d'un circuit 85
4.3.3 Résultats 91
4.4 Conclusions 95

5 Validation expérimentale 97
5.1 Introduction 97
5.2 Analyse des performances 97
5.3 Commentaires 103
5.4 Développement d'un outil de prédiction de la fiabilité 104

6 Conclusion 107

Liste des publications 109

Annexes 109

A Mise en oeuvre des modèles de fiabilité avec MATLAB 111

B Intégration des éléments de logique séquentielle 119
B.1 Méthodologie proposée 119
B.2 Résultats 121

Bibliographie 132

Table des figures

1.1 Superposition des effets de défaillance des composants électroniques. 7
1.2 Exemple d'impact de particule ionisante dans une jonction n+p polarisée en inverse. 12
1.3 Evolution du taux d'erreur dans la logique combinatoire et dans la logique séquentielle avec les noeuds technologiques d'après [30]. 12
1.4 Schéma de la propagation d'une erreur au travers de la logique combinatoire dans une configuration de pipeline. 13
1.5 Schéma de la propagation et l'atténuation d'une impulsion au travers d'un chemin logique : masquage électrique. 14
1.6 Schéma du phénomène de masquage temporel d'une faute. 14
1.7 Schéma du masquage logique d'une faute. 15
1.8 Arboresence des techniques de protection aux fautes. 16
1.9 Schéma typique de Triplication TMR dans la logique combinatoire. 16
1.10 Schéma typique de détection d'erreur pour une fonction quelconque. 17

2.1 Espace représentant les possibles fautes dans un système. 22
2.2 Diagramme de blocs représentant le calcul de fiabilité dans la logique combinatoire. 23
2.3 Circuit pour calculer la fiabilité du signal de sortie du circuit G. 24
2.4 Exemple de BDD pour une fonction logique. 26
2.5 Les effets de 6 possibles stuck-at d'une porte NOR sont modélisés avec 4 indicateurs de faute. 26
2.6 Porte AND à deux entrées et probabilité d'erreur ε. 28
2.7 Exemple de la décomposition d'un circuit logique en niveaux logiques. . . . 30
2.8 Différentes connexions définies pour aider à la construction de la matrice PTM globale. 31
2.9 Exemple de circuit contenant des connexions spéciales pour l'approche PTM 31
2.10 Exemple de la représentation dans le modèle des réseaux bayesiens. 33
2.11 Circuit combinatoire générique. 36
2.12 Porte NAND à 2 entrées. 37
2.13 **Circuit A** : circuit en forme d'arbre, sans corrélation entre signaux. **Circuit B** : circuit contenant un signal reconvergent créant des corrélations. 39
2.14 Circuit logique utilisé d'exemple pour illustrer la méthode SPRMP. 39
2.15 Circuit contenant deux super-portes marquées par des lignes pointillées. . . 41

3.1 Exemple de circuit avec des signaux corrélés. 45
3.2 Topologie de signaux corrélés : application de l'indépendance conditionnelle. 47
3.3 Porte logique XOR à deux entrées. 49

3.4 Arbre de décision d'une porte logique XOR à deux entrées. 49
3.5 Représentation graphique de l'espace de probabilité de la porte XOR. 50
3.6 Calcul de probabilités conditionnées à partir d'un arbre de décision. 50
3.7 Exemple de chemin logique pour le calcul de probabilités conditionnées. . . 51
3.8 Exemple de chemin logique pour le calcul de probabilités conditionnées. . . 52
3.9 Diagramme représentant les différents éléments d'un chemin de reconvergence. 53
3.10 Signaux corrélés. 54
3.11 Exemple de topologie classique contenant une source de reconvergence primaire (S1) et une secondaire (S2). 55
3.12 Exemple d'application des différents conditionnements dans une structure de chemins de reconvergence imbriqués. 55
3.13 Exemple du phénomène de masquage de conditionnement de source. 57
3.14 Exemple d'accumulation de sources de conditionnement. 58
3.15 Circuit générique pour la détection d'entrées incorrélées. 59
3.16 Exemple de cône logique précédant un signal S. 61
3.17 Chemin de corrélation basique pour la démonstration de l'influence du cône logique sur les corrélations. 61
3.18 Corrélation mesurée entre les signaux $S1$, $S2$, $T1$, $T2$, $U1$, $U2$, $V1$, $V2$, $W1$ et $W2$ avec le signal A. 62
3.19 Circuit exemple d'estimation partielle des reconvergences. 62
3.20 Exemple de bloc combinatoire générique. 63
3.21 Exemple de bloc combinatoire pour illustrer les modifications par rapport à CPM primaire. 64
3.22 Exemple de modularisation d'un circuit. 65
3.23 Exemple de modularisation non-disjointe d'un circuit. 66
3.24 Exemple de modularisation non-disjointe d'un circuit. 67

4.1 Porte logique XOR à deux entrées utilisée pour illustrer le calcul de fiabilité en fonction de la probabilité d'erreur de la porte. 72
4.2 Arbre de décision simplifié. 73
4.3 Fiabilité de la porte G en fonction de sa probabilité d'erreur pour 3 configurations des entrées. 75
4.4 Fiabilité de la porte G en fonction de sa probabilité d'erreur pour 3 configurations des entrées. 75
4.5 Exemple de circuits avec des signaux corrélés. 77
4.6 Exemple de configuration d'analyse de porte créant des corrélations. 77
4.7 Implémentation de la fonction logique XOR avec des portes de type NAND. 77
4.8 Résultats de la sensibilité des portes. 79
4.9 Exemple de circuit pour le calcul de la sensibilité de la fiabilité des sorties lors d'une approche hiérarchique. 80
4.10 Structure d'un additioneur de type Brent-kung de 16 bits. 81
4.11 Sensibilité des sorties par rapport à la porte 1 du circuit. 82
4.12 Sensibilité de la totalité des sorties vers toutes les portes du circuit. 82
4.13 Fiabilité des sorties de l'additionneur avant et après le durcissement des 10 portes les plus pesantes. 83
4.14 Sensibilité de la sortie 33 par rapport à toutes les portes du circuit. 84
4.15 Gain en fiabilité des 5 portes les plus pesantes pour la sortie 33. 84
4.16 Circuit combinatoire générique. 85

4.17 Circuit utilisé en exemple pour illustrer le calcul de la borne inférieure de fiabilité. 87
4.18 Diagramme de blocs représentant les limites de capacité d'analyse de la borne inférieure de l'approche CPM. 88
4.19 Diagramme représentant les blocs d'un circuit et la propagation des espaces de probabilités correspondants. 89
4.20 Topologie de blocs non disjoints utilisée pour illustrer l'application au calcul de la borne inférieure à partir de l'approche CPM hiérarchique non disjoint. 91
4.21 Diagramme de la propagation de l'espace de probabilités pour l'approche CPM hiérarchique disjointe. 91
4.22 Borne inférieure de fiabilité pour toutes les sorties de l'additionneur. 92
4.23 Borne inférieure de fiabilité et fiabilité pour une distribution donnée pour toutes les sorties de l'additionneur. 93
4.24 Taux de masquage sur les 4 premières portes du circuit. 94
4.25 Taux de masquage de toutes les fautes sur les sorties 1 et 33. 95

5.1 Diagramme de blocs du circuit *mux*. 98
5.2 Diagramme de blocs du circuit *z4m1*. 99
5.3 Diagramme de blocs du circuit *74283*. 100
5.4 Diagramme de blocs du circuit *74181*. 101
5.5 Diagramme de blocs du circuit *Brent Kung 32B*. 102
5.6 Temps de calcul nécessaire pour chaque sortie de l'additionneur. 103
5.7 Diagramme de blocs de l'outil de calcul de fiabilité. 104

A.1 Circuit d'exemple pour illustrer l'utilisation du code MATLAB 111
A.2 Découpage réalisé pour le circuit d'exemple pour illustrer l'utilisation du code MATLAB 116

B.1 Configuration typique d'un circuit séquentiel. 119
B.2 Modélisation d'un circuit séquentiel comme une suite de circuits combinatoires. 120
B.3 Diagramme du flux de l'algorithme proposé pour l'estimation de la fiabilité des circuits séquentiels. 120
B.4 Signaux définis par un inverseur ou un buffer logiques. 123

Liste des tableaux

2.1 Table de vérité de la fonction logique f. 26
2.2 Table de vérité probabiliste de la porte de la figure 2.6. 28

3.1 Probabilités d'occurrence des différents signaux du circuit de la figure 3.1. . 45
3.2 Résultats et temps de calcul des approches SPR, CPM et CPM approximé pour le circuit de la figure 3.19. 62

4.1 Table de vérité d'une porte XNOR. 92
4.2 Table de vérité en prenant compte des fautes pour une porte XNOR. 92

5.1 Performances pour le circuit *mux*. 98
5.2 Performances pour le circuit *z4m1*. 99
5.3 Performances pour le circuit *74283*. 100
5.4 Performances pour le circuit *74181*. 100
5.5 Performances pour l'additionneur Brent Kung 32 bits. 101
5.6 Tableau comparatif des performances pour tous les exemples traités. 103

B.1 Comparatif pour le bloc A. 121
B.2 Comparatif pour le bloc B. 121
B.3 Comparatif pour le bloc C. 122
B.4 Comparatif pour le bloc D. 122
B.5 Comparatif pour le bloc E. 122

Introduction

Avant-propos

Les circuits digitaux utilisés dans des domaines aussi variés que l'automobile, le médical ou le domaine spatial et nucléaire ont comme principale caractéristique de devoir avoir des taux de fiabilité très élevés. Ceci soit parce qu'ils mettent en jeu la sécurité humaine (automobile, médical) soit parce qu'ils opèrent en milieu agressif (spatial, nucléaire). L'intégration croissante des circuits intégrés, consistant en une augmentation de la densité d'intégration et une diminution de la tension d'alimentation, a pour effet d'augmenter la sensibilité des systèmes aux effets induits par des phénomènes tels que l'impact de particules ionisantes, le bruit thermique ou le crosstalk, qui provoquent l'apparition de fautes transitoires ou permanentes.

L'effet négatif des fautes transitoires a été longtemps prépondérant dans les mémoires. De ce fait, la problématique de la mesure du taux d'occurrence et des éventuelles mesures de correction à appliquer sont bien connues pour ce type de composant. Cependant, des études ont montré que pour les prochains noeuds technologiques (28/22nm), la sensibilité de la logique combinatoire aux fautes transitoires va rattraper, voire dépasser, celle des mémoires. Il est donc nécessaire de développer des méthodes et outils permettant d'évaluer le comportement de ces circuits lorsqu'ils sont soumis à des erreurs telles que celles créées par les radiations. Cette thèse se propose de développer des modèles analytiques permettant de calculer les probabilités de dysfonctionnement dans la logique combinatoire lorsqu'elle est soumise à des erreurs de type inversion de bit ou collage.

Un premier problème vient de la difficulté à utiliser des modèles exacts sur des circuits de grande taille. Dans cette thèse, le travail a consisté à développer de nouveaux modèles et améliorer ceux déjà existants afin de pouvoir traiter des circuits de taille plus importante. Un second problème vient du fait que la fiabilité de la logique met en jeu plusieurs phénomènes dont la modélisation est complexe (probabilité de propagation, effets de masquage, etc.) et qui doivent être pris en compte. Le travail de cette thèse est sous-tendu par la norme de certification et d'analyse FMDEA de fiabilité ISO26262/IEC61508 des circuits numériques pour le domaine automobile.

Motivations

L'estimation de la fiabilité des circuits logiques repose principalement sur 2 approches :
- les approches empiriques
- les approches analytiques

Les approches empiriques, telles que l'injection de fautes, consistent en la simulation ou l'émulation du circuit étudié et en l'injection de fautes de manière contrôlée de façon à pouvoir observer son comportement et étudier les conséquences de celles-ci. Les techniques

d'injection de fautes sont bien connues et indispensables pour réaliser des études sur la fiabilité au niveau fonctionnel du système. Cependant, l'injection de fautes présente des limitations qui restreignent son utilisation dans le cas de la logique combinatoire. En particulier, le nombre d'états à couvrir (chaque configuration de test comporte 2^N états pour un système à N entrées) la rend inapplicable dès que la taille des circuits croît. Ces limitations obligent à mettre en place des méthodologies et procédures complémentaires à l'injection de fautes. Les meilleurs candidats pour ce faire sont les modèles analytiques. Les approches analytiques modélisent le comportement aléatoire des circuits à partir de modèles probabilistes. A partir d'une description structurelle du circuit, de l'information du comportement aléatoire des signaux d'entrée et des portes du circuit, les probabilités de propagation d'une faute vers les sorties sont calculées.

Les approches analytiques aussi connues sous le nom de méthodes probabilistes ont, quant à elles, plusieurs limitations qui empêchent leur application à des circuits de grande taille ou à la certification de circuits. Il existe deux grandes classes de méthodes analytiques, les méthodes exactes et les méthodes approximatives :

- Les méthodes exactes calculent un résultat exact mais avec une complexité exponentielle, ce qui limite leur applicabilité aux circuits de grande taille.
- Les méthodes approximatives fournissent un résultat qui n'est pas exact mais en un temps linéaire ou polynomial. Dans la mesure où actuellement il n'existe pas de méthodes pour borner l'erreur commise, leur applicabilité dans le cadre de normes telles que l'ISO26262 reste limitée.

Objectifs

L'objectif principal de cette thèse est de proposer de nouveaux modèles et méthodes analytiques d'estimation de la fiabilité permettant d'obtenir un résultat exact ou une approximation contrôlée (c'est-à-dire bornée) tout en réduisant la complexité algorithmique et par conséquent le temps de calcul des méthodes déjà existantes. De l'analyse des méthodes existantes, il ressort que l'obstacle principal à l'obtention de méthodes exactes et efficaces en même temps est le traitement des signaux reconvergents, qui créent des corrélations entre signaux dans le circuit et forcent une énumération des états de celui-ci. Les approches développées devront en outre être capables d'étendre le type d'analyse fournie à des applications telles que le durcissement sélectif ou les métriques requises par une analyse en vue de la certification ISO26262.

Organisation de la thèse

Le manuscrit de thèse est organisé en cinq chapitres :

- **Chapitre 1** : Ce chapitre présente les généralités sur la mesure de la fiabilité au sens général : les différents concepts liés à la fiabilité, les métriques existantes, etc. Dans ce chapitre nous présentons aussi de manière succincte les différents phénomènes donnant lieu aux fautes transitoires, avec une petite emphase sur les particules ionisantes, jugées sources principales de ces fautes. Les différents effets de masquage des fautes dans la logique sont abordés et finalement, les principales mesures de protection et de tolérance des fautes sont présentées.
- **Chapitre 2** : Dans ce chapitre nous abordons les principales caractéristiques des différentes approches existantes pour l'analyse de la fiabilité des fonctions logiques

complexes : l'injection de fautes et les modèles analytiques. Nous proposons aussi une étude de l'état de l'art des approches analytiques où nous présentons les caractéristiques des différents modèles existants ainsi qu'une analyse de leurs limitations.

- **Chapitre 3** : Ce troisième chapitre constitue le coeur du travail de recherche. Nous y présentons les deux modèles qui ont été développés au cours de la thèse, avec une analyse détaillée de leurs principes de fonctionnement. Nous y décrivons une approche nouvelle permettant de traiter les signaux reconvergents de manière plus efficace que les approches existantes ainsi qu'une méthode hiérarchique permettant de surmonter la complexité exponentielle du problème pour une large classe de topologies de circuits.
- **Chapitre 4** : Ce chapitre est consacré à l'analyse de circuits basée sur les méthodes proposées dans le chapitre précédent. Au-delà de la seule estimation de la fiabilité des sorties du circuit, ces différentes analyses permettent l'extraction de paramètres qui élargissent l'information fournie par ces modèles. Ainsi, il est possible d'obtenir des informations permettant de modifier la structure du circuit afin d'en améliorer la fiabilité de manière optimale. Aussi, il est possible de déterminer des bornes inférieures de fiabilité dans le cas où toutes les informations sur le comportement des entrées du circuit ne seraient pas disponibles. Nous terminons ce chapitre par une étude de cas, un additionneur 32 bits de type Brent-Kung, qui illustre l'application des différentes analyses et valide les approches proposées. Nous montrons qu'il est possible d'obtenir un résultat de fiabilité exact pour ce circuit comportant 65 entrées et 33 sorties, ce qui constitue un résultat de qualité supérieure à celles des approches exactes reportées dans la littérature.
- **Chapitre 5** : Dans ce chapitre nous effectuons une étude plus générique des performances des modèles proposés et les comparons à celles des autres méthodes alternatives. Nous présentons également les caractéristiques principales de l'outil d'analyse de fiabilité développé en C++ pendant cette thèse.

Chapitre 1

Fiabilité des circuits électroniques numériques

1.1 Introduction

La fiabilité d'un système, au sens large, est la capacité de celui-ci à être fonctionnel dans un ensemble de conditions déterminées, pendant une période de temps définie [1]. Dans le cadre de cette thèse, nous étudions la fiabilité des circuits électroniques numériques. Dans ce cas précis, la fiabilité représente la probabilité d'avoir un état correct des signaux de sortie étant donné un état des signaux d'entrée.

La miniaturisation progressive du transistor dans la technologie CMOS suivant la loi de Moore [2] a pour conséquence la réduction de la tension d'alimentation ainsi que du temps de commutation des transistors. Le gain en autonomie et capacité de calcul des systèmes électroniques a permis l'apparition d'une large variété d'applications, appareils, logiciels, etc. de plus en plus performants. D'un autre côté, la réduction de la taille du transistor jusqu'à l'échelle nanométrique a aussi eu pour effet d'augmenter la sensibilité du système aux fautes [3]. D'une façon générale, les fautes peuvent être classées en deux catégories :

- Fautes systématiques
- Fautes induites

D'une part, les défauts systématiques, liés à la variabilité des matériaux et l'imprécision des processus de fabrication, se traduisent par une augmentation des imperfections et des défaillances dans les circuits [4], [5], [6]. D'autre part, les fautes induites sont provoquées par des éléments en général externes au circuit [7], [28], [8], [9]. Ces deux effets se traduisent par une perte en fiabilité du système [3], [10]. Cette perte oblige les concepteurs à mettre en place des procédures de vérification qui ralentissent et augmentent le coût du processus de conception de circuits [11]. Aussi, des mesures de durcissement des circuits concernant les défauts induits doivent être mises en place. D'une manière très générale, nous pouvons considérer que ces mesures consistent en l'ajout de blocs de détection et correction d'erreurs ou en la modification de la structure du circuit par l'ajout de redondances pour réduire sa sensibilité aux erreurs [12]. L'ajout de blocs correcteurs ou la modification structurelle comporte une augmentation de la consommation et de la surface requise, et une diminution de la fréquence de calcul, allant à l'encontre des gains obtenus par les technologies fortement sub-microniques.

Afin de surmonter les limitations liées à la perte de fiabilité des composants électroniques, il est nécessaire de disposer d'outils et de techniques permettant de minimiser les

coûts de durcissement des circuits. L'objectif de notre travail s'inscrit dans ce besoin d'apporter des améliorations dans la modélisation du comportement des circuits en présence de fautes.

Dans ce chapitre nous présentons les concepts généraux relatifs à la fiabilité des systèmes et les métriques associées. Nous présentons différents mécanismes de défaillance dans l'électronique numérique ainsi que des techniques et des mécanismes de protection aux fautes.

1.2 Généralités sur la fiabilité

Dans cette section nous présentons les différents concepts concernant la fiabilité et leur définition. Ensuite, une classification des défaillances selon leur impact sur le comportement du système est faite. Finalement, différentes techniques pour la mesure de la fiabilité sont présentées.

1.2.1 Définitions

L'étude de la fiabilité s'inscrit dans le domaine plus large de la Sûreté de Fonctionnement (SdF). La SdF est définie comme l'aptitude d'une entité à satisfaire une ou plusieurs fonctions requises dans des conditions données [13]. La SdF englobe plusieurs concepts liés à cette définition. Une déclinaison en quatre thématiques est proposée [14] :

- Fiabilité : aptitude d'un composant ou d'un système à fonctionner pendant un intervalle de temps donné. D'une manière générale, la valeur de fiabilité est designée par R ou R(t), provenant du mot anglais *reliability*.
- Maintenabilité : aptitude d'un composant ou d'un système à être maintenu ou remis en état de fonctionnement, en général appelée M(t).
- Disponibilité : aptitude d'un composant ou d'un système à être en état de marche à un instant donné.
- Sécurité : aptitude d'une entité à ne pas conduire à des accidents inacceptables.

L'étude de la SdF nécessite donc l'analyse de ces paramètres. Les travaux réalisés au cours de cette thèse visent à apporter des améliorations concernant l'analyse de la fiabilité.

1.2.2 Classification des défaillances

Les opérations défaillantes d'un composant menacent le bon fonctionnement du système, ainsi une opération anormale est appelée de diverses manières selon l'impact qu'elle a sur le composant où elle a lieu et la façon dont elle affecte le système. La classification et les définitions les plus souvent utilisées pour les opérations défectueuses sont répertoriées dans [19] :

- Faute : opération incorrecte dans un composant du système
- Erreur : état incorrect du système
- Défaillance : état du système caractérisé par une réponse incorrecte aux stimuli qu'il reçoit.

Nous pouvons affirmer alors qu'une faute peut provoquer une erreur qui elle-même peut ultérieurement provoquer une défaillance. Il est important de souligner que cette classification ne correspond pas à une simple classification à partir du niveau hiérarchique où l'opération anormale agit. Ainsi, une erreur dans un transistor peut se traduire en une

défaillance d'une porte logique qui en même temps peut impliquer une faute dans le circuit où elle est inserée.

Nos travaux concernent notamment les fautes. Nous visons l'analyse de l'influence des fautes dans la logique combinatoire. Les fautes pouvant provenir des entrées ou d'un élément du circuit, nous proposerons des modèles de calcul de la probabilité de propagation de ces fautes, et donc, la probabilité qu'elles provoquent des erreurs dans le système, c'est-à-dire provoquent un état incorrect des sorties.

1.2.3 Métrique de la fiabilité

Typiquement, la spécification de la fiabilité d'un système ou d'un élément se présente sous forme de taux d'erreur ou défaillance, c'est-à-dire, la fréquence avec laquelle se produit une erreur ou une défaillance. En général, le taux d'erreur d'un élément est designé par λ, ou $\lambda(t)$, et s'exprime en [*erreurs/unité de temps*] [20]. Les phénomènes menaçant le bon fonctionnement d'un système sont de nature aléatoire, par conséquent les mesures de fiabilité sont normalement basées sur des méthodes probabilistes et statistiques (prédiction) ou sur des méthodes empiriques et de test (estimation) [18].

Pour les composants électroniques, l'évolution temporelle du taux d'erreur est représentée à partir de la courbe dite de la baignoire [21] (figure 1.1), qui agglomère trois phénomènes ayant un comportement différent en fonction du temps. Cette courbe représente l'addition des phénomènes de mortalité infantile, de fautes aléatoires et de vieillissement [22], [23]. La mortalité infantile est décroissante au cours du temps car les défauts induits lors de la fabrication du composant sont plus enclins à devenir visibles dans les premiers temps d'utilisation. La mortalité infantile est due à des problèmes liés à la fabrication des composants. Les fautes aléatoires sont dues à diverses causes, qui seront présentées plus en détail par la suite du document. Elles sont notamment dues à l'environnement, et ont donc un taux d'occurrence constant. La partie plate de la courbe représente le temps de vie utile. Le vieillissement, qui croît avec le temps, est dû à la fatigue des composants et à leur dégradation au cours du temps [23].

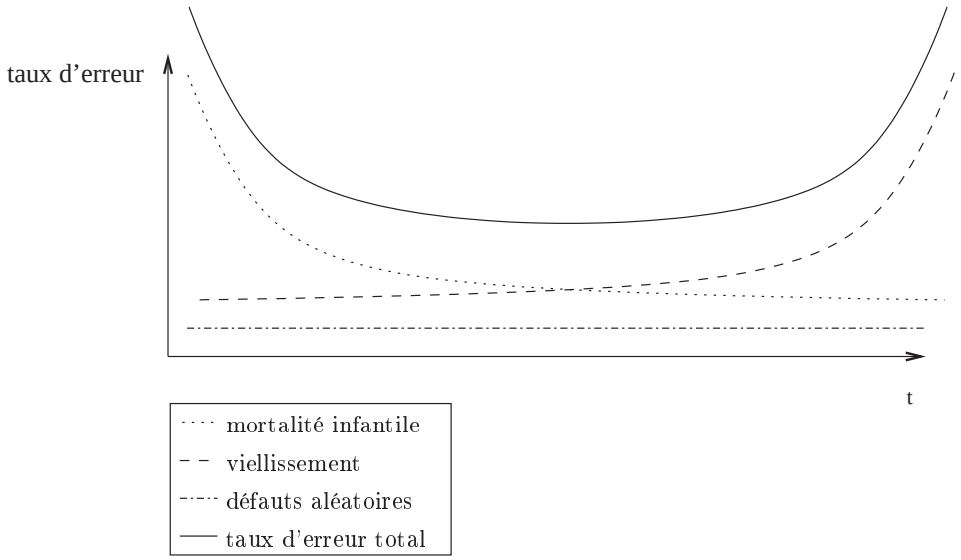


FIG. 1.1 – Superposition des effets de défaillance des composants électroniques.

Les paragraphes suivants présentent les métriques les plus utilisées pour exprimer la fiabilité et le taux d'erreur d'un composant ou système.

Fiabilité comme une probabilité de faute

La fiabilité d'un composant ou système est exprimée à partir de son taux d'erreur selon la formule suivante :

$$R(t) = e^{-\int_0^t \lambda(x)dx} \tag{1.1}$$

Cette mesure est la plus intuitive et celle qui correspond à la définition formelle de la fiabilité. Elle représente la probabilité qu'a un système ou composant de subir une opération incorrecte jusqu'à l'instant t.

Failures In Time - FIT

Une mesure très utilisée pour exprimer la fiabilité d'un circuit est le FIT, de l'anglais Failure-In-Time. Le FIT est défini comme le nombre de défaillances pour toutes les 10^9 heures d'utilisation (soit 114.000 années). Le FIT est une mesure très répandue pour spécifier la fiabilité d'un système, et plusieurs standards et normes l'utilisent comme mesure de leurs contraintes de fiabilité. Cependant, la fiabilité d'un système n'est pas indépendante du temps, c'est-à-dire constante, alors que le FIT est une mesure constante.

Mean Time Before Failure - MTBF

Il existe une autre mesure qui prend en compte l'évolution temporelle du taux d'erreur, appelée MTBF, de l'anglais mean-time-before-failure, ou mean-time-between-failures.

En général, le MTBF est exprimé en heures, et peut être calculé comme suit :

$$MTBF = \frac{1}{\lambda(t)} \tag{1.2}$$

Le MTBF mesure le temps moyen qui s'écoule entre l'apparition de deux fautes. Comme illustré sur la figure 1.1, pendant le temps de vie utile du composant (hors mortalité infantile et vieillissement), le taux d'erreur peut être considéré constant. La valeur de fiabilité peut alors être obtenue à partir de l'expression suivante :

$$R(t) = e^{-\frac{t}{MTBF}} \tag{1.3}$$

Mean Time to Repair - MTTR

Une autre mesure pour l'évaluation des systèmes qui sont conçus pour être résistants aux fautes est le *Mean Time To Repair* (MTTR), c'est-à-dire, le temps moyen mis par le système pour se rétablir d'une défaillance. Cette mesure concerne notamment l'étude de la maintenabilité. Si μ est le taux de réparation d'un système, le MTTR est :

$$MTTR = \frac{1}{\mu} \tag{1.4}$$

Commentaires

Les travaux réalisés au cours de cette thèse concernent principalement les blocs de logique combinatoire insérés entre deux étages de flip-flops (structure de pipe-line dans la logique synchrone) et pendant la vie utile des composants. La mesure la plus adéquate pour mesurer le taux de défaillance dans ces conditions est R (équation 1.1). Pour ce calcul, le taux d'erreur des composants, $\lambda(x)$, est considéré constant et l'intégrale de la formule 1.1 se réalise sur la durée d'un cycle d'horloge.

Dans la réalité, le calcul du taux de défaillance des systèmes complexes n'est pas évident, et le calcul de fiabilité doit intégrer beaucoup de facteurs [24]. La fiabilité des circuits électroniques numériques peut être abordée de deux points de vue différents :

- fiabilité fonctionnelle, c'est-à-dire que le système réalise la fonction correcte.
- fiabilité du signal, c'est-à-dire la probabilité qu'un signal soit dans un état correct étant données les entrées du système.

Dans le cadre de cette thèse, les travaux portent sur l'estimation de la fiabilité du signal.

1.3 Perte de fiabilité dans les circuits numériques

Dans cette section nous présentons de manière succincte les principaux problèmes et sources de défaut rencontrés lors de la fabrication des composants, ainsi que les différentes sources de perte de fiabilité pendant la vie utile des composants.

1.3.1 Rendement des procédés

Le rendement d'un procédé de fabrication représente la fraction de puces fabriquées n'ayant pas de défaut sur le nombre total de puces fabriquées (*yield* en anglais). Plus le rendement est élevé, plus la qualité du procédé est grande [25]. Dans les travaux présentés par Zorian et al. [26], une classification des défauts possibles a été réalisée :

- Systématiques : dus à des variations géométriques dans le processus de fabrication.
- Induites par conception : erreurs spécifiques de conception.
- Paramétriques : associées aux variations statistiques dans le dopage, la longueur du canal, l'épaisseur d'oxide de grille, etc.
- Défauts induits : courts-circuits aléatoires, impacts de particules, vias manquants, contamination, etc.

L'influence de chaque type de défaut varie selon le noeud technologique considéré. En effet, pour les technologies plus anciennes, les défauts induits étaient la source principale de perte de fiabilité et la méthode pour les surmonter consistait en l'amélioration des processus de fabrication. L'amélioration se basait sur l'utilisation de machines de fabrication plus performantes et des meilleurs matériaux. Ainsi, l'amélioration des procédés était indépendante des concepteurs de circuits. Au fur et à mesure que la technologie des semi-conducteurs approche des dimensions nanométriques et que les machines utilisées lors de la fabrication atteignent leurs limites de précision, les sources principales de perte de fiabilité deviennent les défauts systématiques et induits par le procédé de fabrication. En effet, plus le noeud technologique utilisé est de taille réduite, plus le risque d'avoir une différence entre la géométrie conçue et la géométrie obtenue après la fabrication est grand.

Malgré les efforts réalisés par l'industrie des semi-conducteurs, il est admis que ces défauts deviendront inévitables, et que les techniques de tolérance aux fautes seront nécessaires pour assurer un rendement acceptable [27].

Même si les effets de perte de fiabilité liés aux procédés de fabrication sont un des défis majeurs de l'industrie de semi-conducteurs, il y a d'autres facteurs qui altèrent le bon fonctionnement des circuits, notamment les fautes transitoires et intermittentes dues à l'environnement. Les travaux de cette thèse portent notamment sur ces effets. Dans la section suivante, nous présentons les types de fautes qui sont concernés, les mécanismes qui les provoquent ainsi que leur comportement en considérant la nature des circuits d'un point de vue fonctionnel et structurel.

1.3.2 Fautes transitoires dans la logique

Le comportement fiable d'un circuit est lié à sa capacité à tolérer la présence de fautes pendant son fonctionnement. Ainsi, les fautes, selon leur comportement temporel, peuvent être déclinées en trois groupes différents [28] :

1. Permanentes : elles sont provoquées par des changements irréversibles dans un élément du circuit. Elles sont connues sous le nom de *hard-error*.
2. Intermittentes : il s'agit de fautes d'une durée déterminée qui ont lieu d'une manière répétitive au cours du temps.
3. Transitoires : fautes aléatoires, d'une durée déterminée. Elles sont dues à des agents externes au circuit, comme les particules ionisantes, le bruit thermique, etc. Elles sont aussi connues sous le nom de *soft-error*. Elles ont la propriété de disparaître lorsque la source de l'erreur disparaît.

Les fautes permanentes sont notamment liées au rendement des procédés, et pendant la vie utile des composants elles peuvent être considérées comme négligeables [28]. Les fautes intermittentes sont surtout liées à des variations paramétriques et peuvent être contrôlées en utilisant des procédures de test adéquates.

Les fautes transitoires représentent un défi pour la conception de circuits fiables car leur nature aléatoire nécessite une modélisation complexe. Les travaux de cette thèse concernent notamment les fautes de type transitoire dans la logique combinatoire. Leur importance dans la problématique de la fiabilité est croissante. Dans les sections qui suivent nous présentons les raisons de cette augmentation.

1.3.2.1 Sources d'erreurs des fautes transitoires

Une faute transitoire, ou *soft error*, est une faute qui, lorsqu'elle se produit dans un circuit a comme conséquence qu'un signal ou une donnée deviennent erronés. Les fautes transitoires peuvent être créées par les mécanismes suivants [29] :

- Particules alpha : émises par la désintégration radioactive des impuretés telluriques des matériaux contenus dans le boitier des puces.
- Protons et neutrons de haute énergie : il s'agit de particules provenant des rayons cosmiques qui atteignent le niveau du sol.
- Neutrons thermiques : Il s'agit de neutrons qui ont perdu une grande part de leur énergie cinétique et ils sont en équilibre thérmique avec leur environnement. Le flux de référence est de $14\frac{n}{cm^2h}$ (flux de référence à la ville de New York).
- Bruit thermique.
- Chute de la tension d'alimentation.
- Interférences liées à l'intégrité du signal (crosstalk).
- Bruit inductif.
- Bruit du substrat.

La plupart de ces mécanismes sont bien connus, mais leur analyse et prise en compte pour l'estimation de ses effets est d'autant plus complexe que le niveau d'intégration des circuits augmente et que la taille du noeud utilisé diminue.

Parmi tous ces mécanismes, les plus critiques sont les impacts des neutrons et des particules alpha [9]. Plusieurs paramètres influent sur l'apparition d'une faute lors de l'impact d'une particule : l'énergie de la particule qui impacte, la géométrie de l'impact, la localisation de l'impact et la conception du circuit logique. Les circuits logiques avec une capacité élevée et des tensions d'alimentation plus importantes sont moins sensibles aux erreurs. Cette combinaison de capacité de charge est décrite par le paramètre appelé charge critique, Q_{crit}, c'est-à-dire, la charge électrique minimale déposée nécessaire pour provoquer une commutation de la valeur logique de l'élément de mémorisation.

Un charge critique plus élevée signifie une sensibilité moindre aux fautes transitoires. La réduction de la taille de transistors et des tensions d'alimentation, d'un côté désirable pour la performance du circuit, a pour effet d'augmenter la sensibilité du système aux fautes transitoires car la charge critique diminue avec la réduction du noeud technologique. Ainsi, l'importance des fautes transitoires augmente au fur et à mesure que la taille des transistors diminue.

1.3.2.2 Défaillances provoqueés par des particules ionisantes

De l'interaction silicium/ particule ionisante découlent trois effets différents : *Single Event Effect* (SEE), *Total Ionizing Dose* (TID), et *Displacement and Damage* (DD). Le SEE est provoqué par une seule particule énergétique et peut introduire des erreurs temporaires ainsi que des erreurs permanentes. Le TID a pour conséquence une dégradation graduelle des performances électriques. Le DD crée des défauts dans la structure de l'élément dus aux collisions des particules incidentes avec le réseau cristallin. L'effet le plus important pour l'étude que nous avons menée est le SEE car il est associé aux fautes temporaires.

Les particules transfèrent leur énergie à la matière sous la forme d'une ionisation, c'est-à-dire une création de paires électron-trou. L'ionisation peut être induite de manière directe, si c'est la particule elle-même qui ionise la matière, ou de manière indirecte, si ce sont les produits de réactions nucléaires (neutron - noyaux) atomiques qui sont responsables de l'ionisation. Cet effet de ionisation est à l'origine de la création d'un courant parasite dans les transistors qui mène à l'inversion des valeurs logiques contenues dans le circuit si la charge déposée est supérieure à la charge critique.

Il existe deux approches qui expliquent la création d'un courant dans un transistor. La première considère que le courant d'un transistor est formé par deux composantes, l'une caractérisée par la collection de charges sous l'action d'un champ électrique et l'autre par la diffusion de charges [37]. La deuxième, plus récente, attribue la collection de charges à un réarrangement capacitif des charges [38].

Selon l'approche classique, la création d'un courant lors d'un passage d'une particule ionisante peut être expliquée de deux manières selon la position de la trajectoire de la particule ionisante par rapport à une jonction PN, c'est-à-dire, si la particule traverse la jonction ou passe à proximité :

- Si la particule traverse une jonction, le champ électrique sépare les paires électron-trou et un pic de courant est observé ; ceci correspond à la composante créée par la collection des charges. La deuxième composante est formée de charges collectées par diffusion.

– Si la particule passe à proximité de la jonction, seul le courant de diffusion est créé. La figure 1.2 représente les effets de diffusion et collection dans une jonction N+P polarisée en inverse lors d'un impact de particule ionisante.

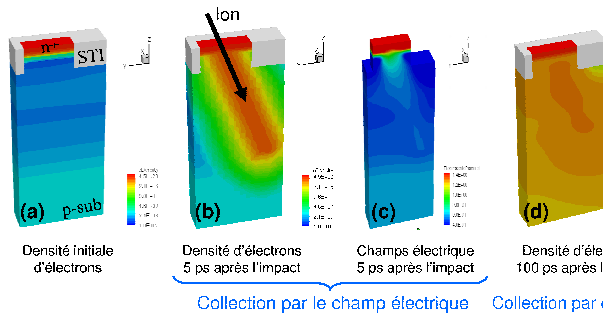


Fig. 1.2 – Exemple d'impact de particule ionisante dans une jonction n+p polarisée en inverse.

Selon l'approche capacitive, l'augmentation de la capacité de la structure associée à l'apparition de la trace provoque l'impulsion initiale de courant. Elle se caractérise par son opposition à l'approche classique, car elle considère que l'ensemble des modifications électriques induites par l'ion dans l'ensemble de la structure doivent être prises en compte. Le courant créé dépend des caratéristiques intrinsèques du matériau et non des caractéristiques électriques du composant.

Finalement, nous parlons de *Single Event Upset* (SEU) lorsque la faute apparait dans un élément de mémorisation (mémoire, bascule) et de *Single Event Transient* (SET) lorsque la faute a lieu dans un élément de logique combinatoire. Le travail de [30] souligne l'importance croissante des fautes transitoires dues à la logique combinatoire dans le taux global de SER des circuits.

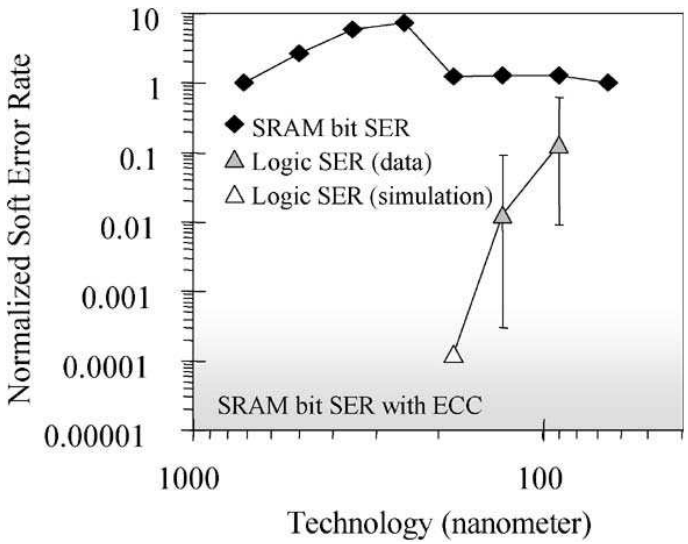


Fig. 1.3 – Evolution du taux d'erreur dans la logique combinatoire et dans la logique séquentielle avec les noeuds technologiques d'après [30].

Les travaux de cette thèse s'inscrivent dans l'analyse de la fiabilité dans la logique combinatoire. Nous voyons sur la figure 1.3 que le taux d'erreur des éléments de mémorisation et de logique séquentielle est constant pour tous les noeuds technologiques. Par contre, la contribution due à la logique combinatoire croît au fur et à mesure que le noeud technologique utilisé decroît.

1.3.2.3 Mécanismes de masquage des fautes

La protection des éléments de mémorisation contre les SEU est largement utilisée dans l'industrie et de nombreuses techniques de détection et de correction ont été développées.

En raison des effets de masquage existants, l'influence des SETs sur la fiabilité a longtemps été considérée comme négligeable. En effet, l'inversion temporaire de la valeur d'un bit dans la logique combinatoire ne conduit pas forcement à un dysfonctionnement du circuit. Des mécanismes de protection sont intrinsèquement présents dans les circuits électroniques. Lors de l'apparition d'une faute, pour que celle-ci devienne visible et ait un effet sur le comportement du circuit, il faut qu'elle soit propagée le long du cône logique et atteigne un élément de mémorisation. En fonction des caractéristiques physiques, architecturales et fonctionnelles du système, des effets de masquage peuvent se produire. Par conséquent, le taux d'erreur observé (avec masquage) est différent du taux d'ocurrence de fautes réel (sans masquage). Ceci oblige à développer des modèles permettant de mesurer ce masquage afin d'obtenir le taux d'erreur observé et par conséquent une bonne mesure de la fiabilité du circuit étudié.

La figure 1.4 représente une structure de logique combinatoire dans un circuit numérique dans une topologie de logique synchrone. Il s'agit du schéma basique de propagation et de mémorisation d'une faute dans une structure de logique synchrone.

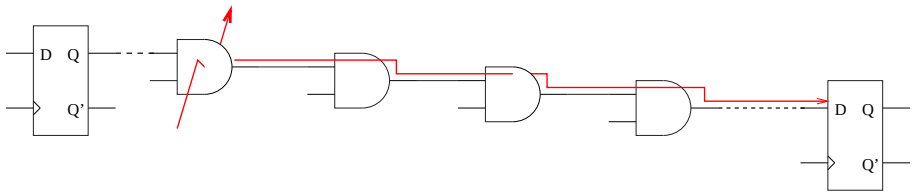


FIG. 1.4 – Schéma de la propagation d'une erreur au travers de la logique combinatoire dans une configuration de pipeline.

La manière la plus commune de modéliser les fautes est sous forme d'impulsion de courant [31]. Pour ce faire, deux approches sont possibles. Premièrement, une modélisation très précise (en SPICE par exemple) qui nécessite beaucoup de paramètres. Par exemple, l'impulsion de courant résultante d'un impact de particule peut être décrite comme une double exponentielle. Une autre possibilité consiste en l'utilisation de modèles plus simples, comme en forme triangulaire ou trapezoïdale, qui nécessitent seulement des informations sur la durée et/ou sur l'amplitude [32], [33] et [34]. Le seconde approche consiste en l'utilisation de modéles logiques (au sens booléen) et probabilistes pour déterminer la probabilité qu'une faute soit masquée. C'est cette approche que nous développons dans cette thése.

Les trois effets qui peuvent masquer une faute sont le masquage électrique, temporel et logique [35]. Dans les sections qui suivent, nous trouvons une briève description de chacun de ces effets.

Masquage électrique - Atténuation des impulsions

Un SEU ou SET est électriquement masqué si l'impulsion correspondant à la faute est atténuée par les propriétés électriques des portes sur son chemin de propagation de manière à ce que l'impulsion résultante n'ait ni la durée ni l'amplitude suffisantes pour être capturée par un élément de mémorisation. Ainsi, le masquage électrique dépend des caractéristiques de la faute (durée et amplitude) et des caractéristiques du chemin électrique par lequel elle se propage. Le masquage électrique est la combinaison de deux effets électriques qui diminuent la puissance d'une impulsion lorsque celle-ci traverse une porte logique :

- Les délais temporels provoqués par le temps de commutation des transistors qui ont comme conséquence l'augmentation des temps de montée et descente de l'impulsion.
- L'amplitude d'une impulsion de courte durée peut diminuer puisque la porte commence à commuter avant que l'impulsion ait atteint son amplitude maximale.

La figure 1.5 présente un schéma de l'effet d'atténuation décrit ci-dessus.

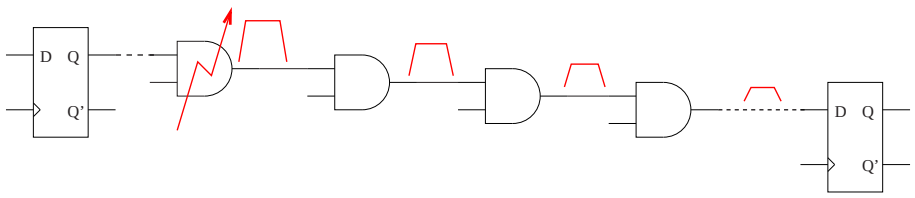


Fig. 1.5 – Schéma de la propagation et l'atténuation d'une impulsion au travers d'un chemin logique : masquage électrique.

Masquage temporel

Pour qu'une valeur soit capturée par l'élément de mémorisation il faut qu'elle soit stable pendant la fenêtre temporelle d'écriture de l'élément en question (temps de "setup" et de "hold"). Dans le cas d'une faute transitoire, il faut qu'elle arrive dans cette fenêtre de lecture et qu'elle ait une durée suffisante pour être capturée. Le cas échéant, la faute sera masquée. La figure 1.6 présente un schéma basique du masquage temporel. Le masquage

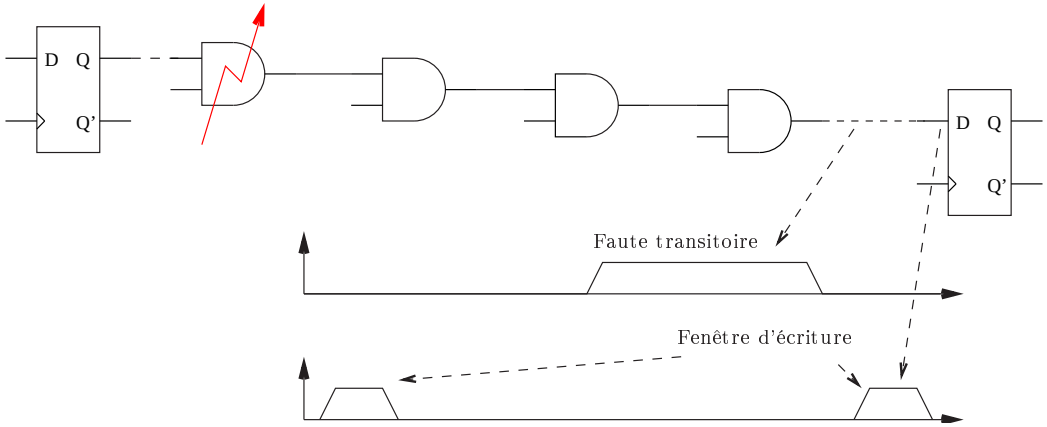


Fig. 1.6 – Schéma du phénomène de masquage temporel d'une faute.

temporel dépend fortement de la fréquence d'opération du circuit.

Masquage logique

Le masquage logique d'une faute transitoire se produit lorsqu'une valeur erronée ne fait pas partie de l'ensemble de signaux qui définissent l'état des signaux de sortie du bloc combinatoire où la faute a eu lieu. Pour une porte logique, la valeur d'une entrée peut figer la valeur de sortie de la porte, indépendamment de la valeur des autres entrées. Par exemple, si l'une des entrées d'une porte OR à 2 entrées a une valeur '1' logique, la sortie sera 1, quelle que soit la valeur de l'autre entrée. Ainsi, lorsqu'une valeur erronée est propagée jusqu'à une porte dont la valeur de sortie est figée par une autre entrée, la propagation de l'erreur s'arrête, et il y a un masquage logique de la faute. Sur la figure 1.7, nous trouvons une représentation basique de ce phénomène.

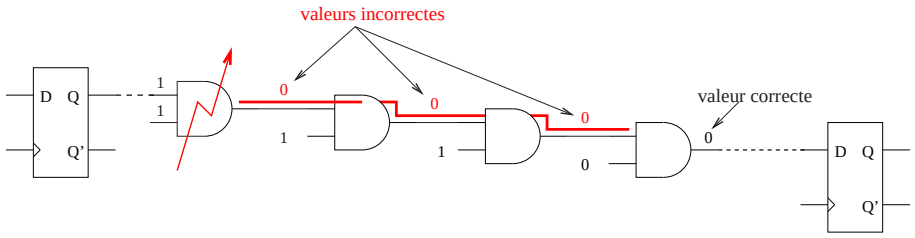


FIG. 1.7 – Schéma du masquage logique d'une faute.

Évolution avec la technologie

Il est intéressant de remarquer que les effets de masquage évoluent avec les progrès de la technologie CMOS :

- Avec la réduction de la taille du transistor et de la tension d'alimentation, les seuils de tension pour la commutation sont eux aussi réduits. Par conséquent, le taux de masquage électrique est réduit car des impulsions avec une amplitude faible sont en mesure de se propager et provoquer une erreur.
- La diminution des temps de propagation des portes logiques conduit à une augmentation de la fréquence du signal d'horloge, ce qui a pour effet de réduire aussi le masquage temporel, car la durée du cycle d'horloge étant réduite, les temps de setup et de hold (fenêtre temporelle de capture) sont réduits permettant ainsi l'échantillonage de transitoires plus courts.
- Le masquage logique est indépendant de la technologie utilisée, car il dépend seulement de la topologie du circuit.

Il est donc manifeste qu'avec le progrès technologique, non seulement la sensibilité aux perturbations des composants électroniques est augmentée mais aussi que leurs mécanismes de masquage naturel sont réduits.

1.4 Techniques de protection aux fautes

Le réel besoin de pallier aux effets des erreurs sur les circuits électroniques a donné lieu à de nombreuses recherches à ce sujet [43]. Les mécanismes de protection aux fautes se déclinent notamment en deux catégories : la tolérance aux fautes et la prévention des

fautes. La figure 1.8 présente l'arboresence des approches existantes pour la protection aux fautes, ainsi que quelques exemples pour chaque approche (définies plus tard dans cette section).

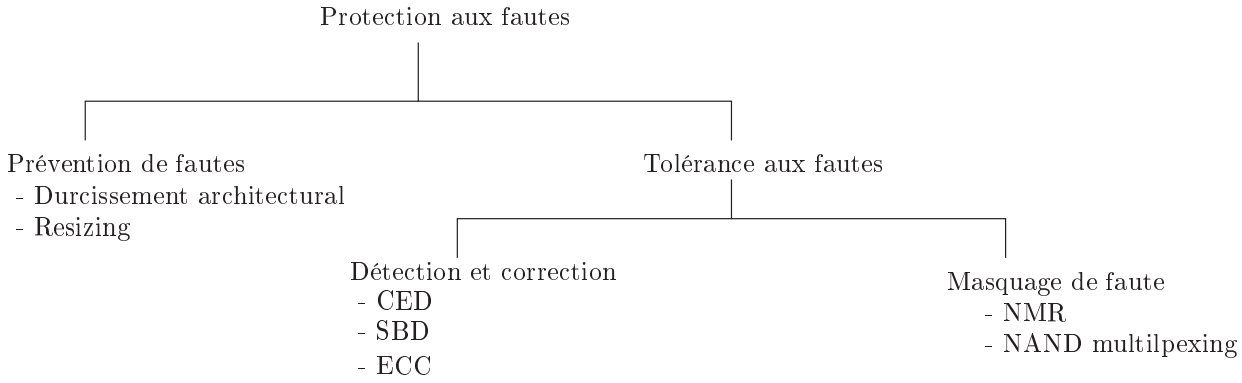


Fig. 1.8 – Arboresence des techniques de protection aux fautes.

Les circuits tolérants aux fautes sont conçus pour être pleinement opératifs en présence d'erreurs. Ainsi, la tolérance aux fautes peut être obtenue de deux façons : par masquage d'erreurs ou par détection et correction d'erreurs. Le masquage d'erreurs ne modifie pas le comportement normal du circuit. Typiquement, ces techniques consistent en l'ajout de redondances fonctionnelles suivi d'un arbitre. Un exemple de technique de tolérance aux fautes est la N-Modular Redundancy (NMR) [39], consistant en l'ajout de redondances fonctionnelles. Sur la figure 1.9 est représenté un schéma typique de triplication (TMR), où chaque bloc de logique est un duplicata du bloc original, le résultat est obtenu par vote majoritaire. Une autre technique basé sur la duplication est le *NAND multiplexing*,

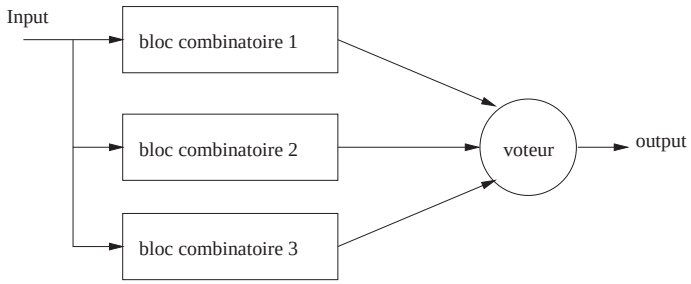


Fig. 1.9 – Schéma typique de Triplication TMR dans la logique combinatoire.

introduite par J. Von Neumann [56]. Elle consiste notamment en la duplication massive des composants du circuit suivi d'interconnections aléatoires.

Typiquement, les techniques de détection et correction d'erreurs consistent en la détection et signalisation d'une faute et l'éxécution postérieure d'une procédure de correction. Pour ce faire, plusieurs méthodes existent, telles que le *Concurrent Error Detection* (CED) [40], *Error Correcting Codes* (ECC) [41] ou le *Signed Binary Digit* (SBD) [42].

Les techniques de tolérance aux fautes sont généralement conçues suivant une hypothèse sur le nombre maximal de fautes tolérables (généralement une). Si le nombre de fautes est supérieur à cette limite, le système n'est plus capable de les détecter ou de les corriger.

La figure 1.10 représente le schéma typique d'une topologie de détection d'erreur, avec les blocs rédundants ajoutés à la fonction principale.

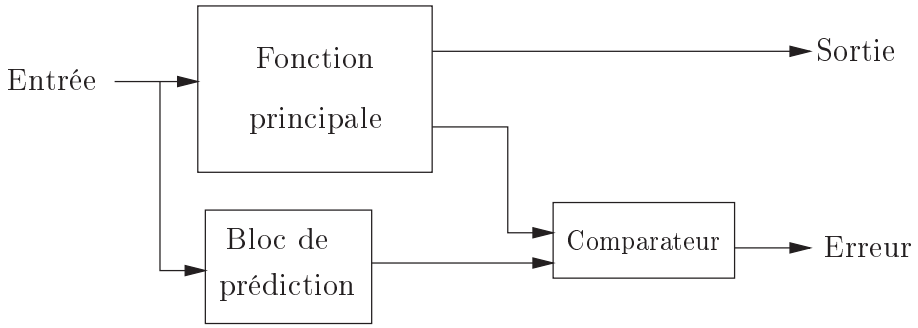


FIG. 1.10 – Schéma typique de détection d'erreur pour une fonction quelconque.

Les différences principales entre les techniques de tolérance et les techniques de détection et correction résident dans la nature de la réaction subie par le système lors de l'apparition d'une faute. Les approches de tolérance sont conçues de manière à ce que la détection et la correction soient implicitement réalisées à partir des redondances utilisées, tandis que les techniques de détection et correction consistent en l'ajout de blocs spécifiques qui réalisent séparement ces deux tâches. Les figures 1.9 et 1.10 représentent ces deux types de construction.

Les alternatives à ce type d'approche sont les techniques de prévention de fautes. Ces techniques consistent en le durcissement des composants afin d'éviter l'apparition de fautes. Les techniques de durcissement se déclinent en trois familles principales [12]

- Modification des procédés de fabrication.
- Modification architectural des composants. Cette technique est appelée *Robust By Design* en anglais (RBD)
- Modification de la taille des dispositifs sensibles.

Ces techniques ne seront pas abordées ici.

1.5 Conclusion

Dans ce premier chapitre nous avons présenté le cadre général de cette étude. Nous avons présenté de manière succincte le domaine de la sûreté de fonctionnement et de la fiabilité. Après que les différentes mesures disponibles pour l'étude de la fiabilité aient été répertoriées, nous avons mis l'accent sur celle qui est la plus adéquate pour notre cas d'étude : la mesure de la fiabilité comme probabilité de faute.

Ensuite, nous avons souligné l'importance de l'étude de la dégradation de la fiabilité dans la logique : au fur et à mesure que l'industrie des semi-conducteurs utilise des technologies de plus en plus sub-microniques, la sensibilité du système et des composants aux fautes augmente. Nous avons analysé les principaux agents pour cette perte de fiabilité et avons détaillé le comportement des fautes dans le domaine de l'étude de cette thèse, à savoir la logique combinatoire.

Finalement, nous avons présenté différents types de techniques et d'approches existantes pour combattre les limitations produites par la perte de fiabilité.

Chapitre 2

Techniques d'analyse de la fiabilité dans les circuits électroniques numériques

2.1 Introduction

Le besoin croissant d'assurer le bon fonctionnement des circuits électroniques a fortement entrainé le développement de nombreuses méthodes visant à l'estimation et au contrôle de la fiabilité. De même, il est nécessaire de disposer d'approches permettant l'estimation de la fiabilité durant la phase de conception afin de pouvoir prendre des mesures de durcissement adéquates. Notre travail s'incrit dans ce type d'analyse.

La grande variété de phénomènes qui détériore les composants électroniques et modifie le fonctionnement attendu des circuits ainsi que la complexité grandissante de ces systèmes nécessitent un large spectre d'approches pour être maitrisés, allant de l'analyse du transistor isolé jusqu'au système complet. Pour réaliser une estimation correcte de la fiabilité, la combinaison d'analyses à différents niveaux fonctionnels et hiérarchiques est nécessaire.

Dans cette section sont présentés différents types d'approches possibles pour l'analyse des éléments d'un circuit numérique, consistant notamment en :

- Analyse des cellules élémentaires : cela concerne l'analyse individuelle des éléments de mémorisation et des portes logiques basée sur les propriétés physiques de ces composants.
- Analyse de fonctions logiques complexes : cela prend en compte les données des cellules basiques afin d'évaluer la fiabilité d'une manière plus globale. Cette analyse prend en compte non seulement les caractéristiques physiques du système, mais aussi les caractéristiques architecturales et fonctionelles.

L'analyse de cellules élémentaires sera présentée de manière concise, visant à introduire seulement les principaux types d'analyses existantes. Dans la partie consacrée à l'analyse de fonctions complexes nous présentons les deux types d'approches existantes :

- L'injection de fautes.
- Les approches probabilistes dans la logique combinatoire.

Concernant l'injection de fautes, nous présentons ses caractéristiques principales, ainsi que ses limitations, qui justifient la pertinence de l'approche probabiliste. Finalement, nous présentons toutes les techniques relatives à l'approche probabiliste dans la logique combinatoire, ainsi qu'une analyse critique de leurs avantages et limitations.

2.2 Analyse de la fiabilité des cellules élémentaires

L'analyse des cellules basiques d'un système est essentielle pour l'évaluer postérieurement dans sa globalité. Dans cette section nous présentons les différents types d'approches pour l'analyse du taux de SEU, qui concerne les éléments de mémorisation, et du SET, qui concerne la logique. Typiquement, la mesure de la fiabilité de ce type de composant est donnée par le SER.

Durant la vie utile d'un composant (cellules SRAM, flip-flops, porte logique, etc.), le phénomène dominant affectant la fiabilité de ce type de composant est celui des fautes de type transitoire, tels que SEU et SET (c.f. figure 1.1). La prédition du SER n'est pas une tâche évidente, car les fautes sont distribuées de manière aléatoire dans le temps, et en plus leur effets ne sont pas permanents, leur observation est alors dificile [46]. Pour l'estimation du SER d'un composant de ce type il existe deux types d'approches principales :

1. Approches expérimentales : les tests radiatifs.
2. Approches analytiques : modélisation et la simulation.

Une description des conditions des tests radiatifs se trouve dans [44] et [45]. Les coûts des tests radiatifs étant très élevés, leur utilisation est surtout destinée à la validation de composants. Les tests radiatifs ne sont pas adaptés à l'étude des cellules de logique. En effet, l'observation d'un SET est très compliquée car elle nécessite un accès à la valeur contenue par le composant. Les approches analytiques par simulation se sont imposées donc naturellement. Ce type d'approche se décline en deux branches : les simulations basées sur des modèles physiques du composant et les simulations multi-dimensionnelles du composant (notamment la TCAD) [47].

Concernant les simulations basées sur des modèles physiques, l'approche la plus utilisée est le modèle de diffusion-collection. Cette approche n'est pas très coûteuse en termes de complexité, mais son efficacité devient limitée lors de l'analyse de composants à géométrie réduite [48]. Une approche plus précise mais aussi plus coûteuse est basée sur l'utilisation de codes utilisant des modèles d'hydrodynamique et d'équilibre énergétique, [47] et [48]. Une alternative très précise mais aussi très coûteuse est les simulations basées sur le méthodes de type Monte-Carlo [49]. Ce type de simulation nécessite un grand nombre d'itérations pour obtenir une bonne estimation.

Les simulations multi-dimensionnelles modélisent l'impact des particules sur la cellule élémentaire, et sont capables de simuler les différentes géométries d'impact avec plus de précision que les méthodologies précédentes.

Nous avons discuté sur la modélisation physique au niveau de composants des fautes transitoires. En montant d'un niveau hiérarchique, ce type de modélisation peut être utilisé pour compléter les modèles visant à l'évaluation de la fiabilité au niveau logique des circuits.

2.3 Analyse de la fiabilité de fonctions logiques complexes

Dans cette section nous présentons les différentes approches effectuant une analyse de la fiabilité au niveau des fonctions logiques complexes. Ce type d'analyse doit prendre en compte la fiabilité des éléments dont les fonctions sont compos'ees et la façon dont les fautes se propagent à travers les divers éléments. En général, les méthodes appliquées pour le faire consistent en la reproduction du comportement du circuit en présence de fautes ou en un calcul analytique de la probabilité de propagation de fautes dans le circuit.

2.3.1 Analyse par injection de fautes

L'injection de fautes est une technique qui vise à l'évaluation de la tolérance aux fautes d'un système. Ceci est réalisé à partir de la comparaison du comportement du système en présence de fautes avec le réponse du système en l'absence de fautes, [50]. Pour ce faire, des fautes sont injectées dans un modèle du circuit pendant qu'il exécute une tâche donnée afin d'étudier les effets qu'elles produisent. Le modèle de fautes le plus utilisé est le bit-flip, consistant en l'inversion de la valeur d'un bit d'information dans un élément de mémorisation du circuit. Lorsque le bit-flip atteint une mémoire ou une flip-flop, la valeur stockée change.

Pour l'évaluation de l'effet de la faute injectée, trois effets sont proposés [52] :

1. Une faute qui produit un comportement erroné du circuit est classée comme défaillance.
2. Une faute qui ne produit pas de comportement erroné mais qui laisse, à la fin de l'exécution, le système dans un état différent de celui attendu, est classée comme faute latente.
3. Une faute qui ne produit pas d'effet est classée comme silencieuse.

L'injection de fautes peut se réaliser de deux manières différentes [50] :

1. Par simulation
2. Par émulation

Dans les paragraphes suivants nous présentons les caractéristiques principales de ces deux approches.

2.3.1.1 Injection de fautes par simulation

Les approches basées sur la simulation utilisent une simulation discrète d'un modèle du circuit et réalisent l'injection des fautes. Typiquement, ce processus consiste en l'exécution de la simulation jusqu'à l'instant d'injection, en la modification du signal correspondant et, finalement, en la reprise de l'exécution et enfin le classement de la faute.

Les systèmes de simulation ont donc besoin des fonctionnalités suivantes :

- Contrôle de l'exécution : lancement, arrêt, reprise.
- Injection de la faute : modification du signal.
- Observation des signaux de sortie : détection des défaillances.
- Observation de l'état du circuit : détection des fautes latentes.

L'approche par simulation est préférable lorsqu'il est nécessaire de réaliser une analyse de fiabilité pendant le processus de conception d'un circuit, car elle est applicable même s'il n'y a pas de prototype du circuit et permet l'analyse de presque toutes les fautes possibles. Cependant, ce type d'approche nécessite beaucoup de temps de simulation [51].

2.3.1.2 Injection de fautes par émulation

L'utilisation de FPGA pour émuler le comportement du circuit à vitesse réelle de fonctionnement permet d'accélérer le processus d'injection de fautes par rapport à la vitesse des simulations. L'émulation consiste à exécuter les mêmes tâches que la simulation sur un FPGA. La complication de l'utilisation de FPGAs réside dans le fait qu'elles sont moins flexibles pour réaliser le contrôle de l'injection de fautes. En général il y a une répartition des tâches de contrôle entre un logiciel de contrôle et le FPGA.

L'approche par émulation est préférable quand un prototype du produit est déjà disponible, ou quand le système est trop grand pour être modelé et simulé à un coût acceptable [51].

2.3.1.3 Commentaires

Avec l'intégration croissante des circuits numériques, le nombre d'états du système entraine une explosion combinatoire du nombre d'états à tester, ainsi le processus d'injection de fautes devient de plus en plus couteux et lent. Pour un système avec n flip-flops, le nombre d'états à considérer est 2^n, ce qui est impossible pour des systèmes de grande taille. En plus de la dimension spatiale (les différentes flip-flops), il faut aussi prendre en compte le fait que les fautes peuvent arriver à des instants différents durant l'exécution du programme. Il faut donc rajouter la dimension temporelle. Finalement, différents types de fautes peuvent être considérés. La figure 2.1 représente un diagramme basique de l'espace de fautes.

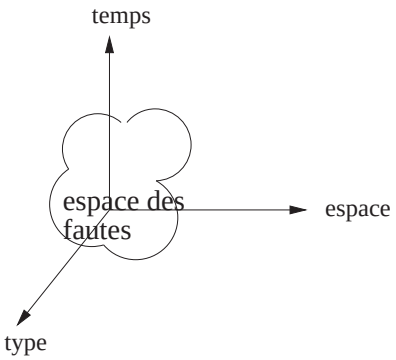


Fig. 2.1 – Espace représentant les possibles fautes dans un système.

Un alternative à l'injection exhaustive est l'injection de fautes statistique [53], basée sur la théorie de l'échantillonage, [54], et qui vise à établir des bornes pour l'erreur avec un degré de confiance établi à partir de l'injection d'un nombre réduit de fautes.

D'autres réductions de la procédure sont possibles [52] :

- L'exécution du circuit sans faute peut être évitée si l'état du circuit est disponible à l'instant de l'injection.
- L'exécution du circuit en présence de fautes peut être arrêtée aussitôt que le classement de la faute a été réalisé.

L'injection de fautes est réalisée sur les flip-flops du système sous étude, par conséquent elle ne prend pas en compte les fautes qui sont provoquées par les blocs de logique combinatoire du circuit. Ainsi, il est nécessaire de développer des alternatives à l'injection de fautes qui prennent en compte cet effet.

2.3.2 Modélisation de la probabilité de propagation de fautes dans la logique combinatoire

L'étude de la fiabilité dans la logique combinatoire consiste en le calcul de la probabilité des signaux de sortie du circuit d'être dans un état correct (comparé à un comportement idéal du circuit) quand le circuit est sujet à la présence de fautes. Ainsi, ce calcul nécessite

plusieurs paramètres pour être effectué. Typiquement, on considère le taux d'erreur des portes logiques et des entrées (signaux de sortie de l'étage de flip-flops précédent dans le cas typique d'une structure de pipe-line) du circuit comme donnée du problème, et la fiabilité est obtenue à partir de modèles de propagation des fautes dans la logique. La figure 2.2 représente un diagramme de ce type d'analyse.

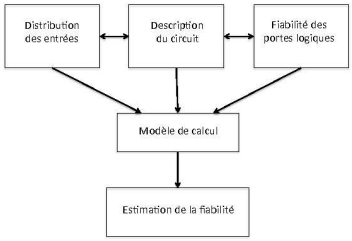


FIG. 2.2 – Diagramme de blocs représentant le calcul de fiabilité dans la logique combinatoire.

La mesure de fiabilité des blocs de logique combinatoire est liée à trois phénomènes : le masquage électrique, le masquage temporel et le masquage logique. Les natures différentes de ces trois types d'approches rendent difficile l'obtention de modèles considérant les trois effets. Par conséquent, l'analyse de fiabilité doit, en combinant des modèles, considérer les différents masquages possibles, ou simplifier les modèles afin de réduire la complexité de l'analyse pour que celle-ci devienne réalisable.

L'analyse de la fiabilité dans la logique combinatoire peut être approchée de deux manières différentes [55] :

- Fiabilité de signal : probabilité que le signal logique soit correct.
- Fiabilité fonctionnelle : Probabilité que le circuit réalise l'opération correctement.

Dans les paragraphes qui suivent, nous présentons les principaux travaux qui ont été faits pour l'analyse de la fiabilité de la logique combinatoire et qui présentent leurs mesures en utilisant ces deux paramètres.

Approche de Von Neumann

Le travail de Von Neumann [56] est l'un des premiers à modéliser le comportement de systèmes créés à partir de composants qui ne sont pas fiables et le premier à porter un intérêt à la logique probabiliste. Ses travaux portent sur l'analyse des bornes inférieures de la TMR et concernent notamment la fiabilité fonctionnelle. Selon Von Neumann, un système est considéré fiable si sa probabilité d'être dans un état correct est supérieure à un certain seuil. Lorsque la probabilité est inférieure, les résultats fournis par le système sont décorrélés de ses entrées.

Méthodes basées sur l'utilisation de classes fonctionnelles équivalentes

Le travail de McCluskey et al. [57] présente la méthode des classes fonctionnelles équivalentes, consistant en le regroupement des fautes qui ont le même effet sur le comportement

du circuit. Les types de fautes concernées par cette approche sont les collages de bit (ou *stuck-at*) ce qui la rend peu appropriée pour l'analyse de fautes transitoires.

Le travail de Ogus [58] est pionnier dans la problématique de la fiabilité du signal. Ce travail présente une approche basée sur la méthode de classes fonctionnelles équivalentes pour estimer la probabilité d'avoir un signal correct à la sortie. Cette approche fournit un résultat exact, mais son application reste limitée car la méthode des classes fonctionnelles équivalentes énumère tous les états possibles du système, ce qui devient irréalisable pour des circuits de taille importante. Des simplifications du modèle sont proposées, mais les résultats sont loin d'être satisfaisants.

Modèle probabiliste des réseaux logiques

Dans le même document Ogus propose une autre approche basée sur le modèle probabiliste des réseaux logiques [59]. Dans cette approche, la probabilité d'un signal S est définie comme $s = P(S = 1)$, et donc : $P(S = 0) = 1 - s$. À partir d'une série de règles qui sont présentées dans ce travail, les auteurs affirment qu'il est possible de calculer la probabilité du signal de sortie.

Étant donné qu'il est possible de calculer la probabilité que la sortie d'un circuit soit '1', les auteurs affirment qu'il est possible de calculer la fiabilité du circuit G à partir d'un circuit H dont la sortie serait égale à '1' si la sortie du circuit G est correcte. Ceci en construisant un circuit B dans lequel sont injectées toutes les fautes possibles du circuit G, qui, quant à lui, a un comportement idéal. La sortie du circuit H est établie comme une fonction XNOR des sorties des circuits G et B, et donc, elle sera égale à '1' lorsque les deux sorties de G et B sont égales. Donc, la fiabilité du circuit G correspond à la probabilité du signal de sortie du circuit H. Le schéma de blocs de cette approche est représenté sur la figure 2.3. Le signaux Z et Zb représentent respectivement les sorties des circuits G et B, I représente les entrées primaires du circuit G, et F correspond aux fautes que l'on injecte dans le circuit B. Les fautes modélisées pour ce modèle sont les 'stuck-at', ce qui le rend

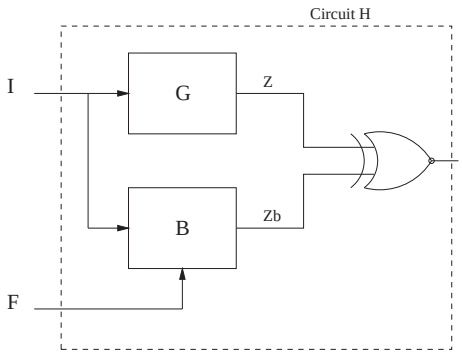


Fig. 2.3 – Circuit pour calculer la fiabilité du signal de sortie du circuit G.

peu approprié pour la problématique des fautes transitoires. De plus, les règles utilisées pour calculer la probabilité du signal de sortie assument implicitement que les signaux sont indépendants, ce qui n'est pas réaliste. Les auteurs ne donnent pas d'informations ni sur la précision ni sur la complexité de leur modèle.

Un travail très similaire a été présenté dans [60]. Les auteurs proposent une heuristique permettant de calculer la probabilité du signal de sortie (et non la fiabilité) d'un circuit

combinatoire en présence de fautes. Le modèle est limité par la complexité exponentielle des algorithmes proposés et est restreint aux fautes de type 'stuck-at'.

Approche de Koren

Une approche traitant aussi la fiabilité du signal est présentée par Koren [61]. Dans ce travail sont définis deux types de fiabilité pour un signal S :

- Fiabilité du signal : probabilité que le signal S soit correct à l'instant t.
- Fiabilité accumulative : probabilité que le signal S soit correct dans l'intervalle $[0, t]$.

Les auteurs présentent une série de règles basées sur la topologie du circuit pour calculer les différentes fiabilités proposées. Ces règles sont très similaires aux travaux présentés dans [58],[59] et [60]. L'intérêt de la déclinaison de la fiabilité en deux formes reste très limité dans une configuration de logique synchrone et visant l'analyse de fautes transitoires, car la fiabilité accumulative vise à intégrer la possibilité de correction d'un signal, ce qui est évident avec les fautes transitoires.

Circuit Equivalent Graph

Un autre travail concernant les fautes de type 'stuck-at' est présenté par Dokouzgiannis et al. [62]. Les auteurs présentent une méthode permettant de calculer de manière exacte la fiabilité d'un circuit de logique combinatoire. Ils utilisent pour cela une représentation du circuit en forme de CEG (Circuit Equivalent Graph). La valeur de la fiabilité est trouvée à partir d'une analyse systématique du CEG, ce qui est un avantage par rapport aux travaux de [58], [59] et [60] qui sont difficilement paramétrisables et nécessitent beaucoup de travail manuel. Dans ce travail, le CEG est construit à partir des graphes équivalents de portes élémentaires qui constituent le circuit, appelés GEGs. A partir du CEG, les auteurs affirment qu'il est possible de trouver les vecteurs de fautes dominants qui mènent à un état correct du système, et trouver la fiabilité à partir de leur probabilité d'occurrence. La complexité de la méthode croît exponentiellement avec le nombre d'entrées du circuit sous étude, ce qui limite fortement son application à des circuits de grande taille. Les auteurs affirment qu'avec cette approche il est possible d'analyser des circuits ayant jusqu'à 20 entrées primaires. Cette approche est orientée vers la modélisation des fautes fixes et l'utilisation des graphes CEG rend difficile l'application à des analyses liées aux fautes transitoires.

Approche basée sur l'utilisation de diagrammes de décision binaires (BDD)

Le travail présenté dans [63] propose un algorithme d'évaluation de la fiabilité des circuits de logique combinatoire conçus pour être tolérants aux fautes. Le modèle s'applique aux fautes de type 'stuck-at'. Pour modéliser l'effet de fautes multiples les auteurs proposent l'utilisation d'indicateurs de fautes comme variable de contrôle. Les indicateurs de fautes consistent en des variables booléenes qui prennent valeur 1 seulement si la faute qui leur est associée est présente. Ainsi, pour la faute i son indicateur est f_i.

Pour éviter l'énumération de toutes les fautes, ils recourent aux BDD (Binary Decision Diagram) [64]. Un exemple de BDD est donné pour la fonction du tableau 2.1 sur la figure 2.4.

Le modèle utilise un BDD du circuit sans fautes et un autre contenant les fautes possibles. Cette méthode est limitée par la taille des circuits, les auteurs affirment qu'au-delà d'une certaine taille les BDD souffrent d'une perte d'efficacité qui limitera l'application

x_1	x_2	x_3	f
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

Tab. 2.1 – Table de vérité de la fonction logique f.

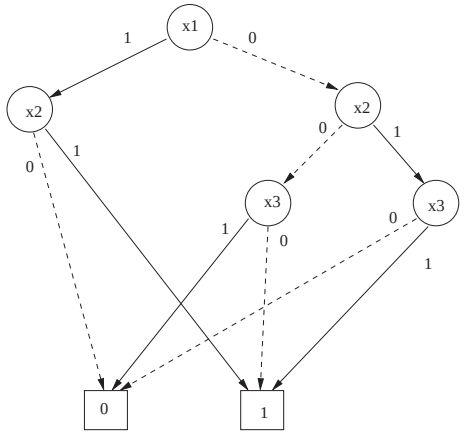


Fig. 2.4 – Exemple de BDD pour une fonction logique.

de leur méthode à des circuits de taille moyenne. Sur la figure 2.5 on trouve un exemple d'utilisation d'indicateurs de fautes sur une porte NOR. Pour cette porte, dont les entrées sont x_1 et x_2, on peut avoir 6 différents 'stuck-at', c'est-à-dire, 4 pour les entrées, x_1 ou x_2 'stuck-at-1' ou 'stuck-at-0', et 2 pour la sortie : 'stuck-at-1' ou 'stuck-at-0'. Sur la figure 2.5 les signaux f_i correspondent aux indicateurs de fautes utilisés pour construire le BDD. Pour ce cas précis, les indicateurs sont :

- f_1 : x_1 'stuck-at-0'
- f_2 : x_2 'stuck-at-0'
- f_3 : x_1 'stuck-at-1', x_2 'stuck-at-1' et la sortie 'stuck-at-0'
- f_4 : sortie 'stuck-at-1'

Ceci permet de réduire le nombre de noeuds à utiliser dans le BDD.

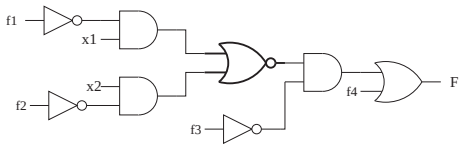


Fig. 2.5 – Les effets de 6 possibles stuck-at d'une porte NOR sont modélisés avec 4 indicateurs de faute.

Modélisation à partir de glitchs de courant

Le travail de Omana et al. [65] montre que la propagation d'un glitch à travers une porte logique peut être modélisée par le retard de la porte et le chemin sensible, c'est-à-dire si l'état d'un signal influe ou non sur la sortie, et le délai de propagation associé modélisé par une chaine d'inverseurs. Ces deux approximations prennent en compte le masquage électrique et le masquage logique. Ce modèle porte sur l'effet des impacts de particules alpha et vise à étudier la sensibilité du circuit à ce type d'impact. La sensibilité du circuit est calculée pour un noeud donné par rapport à une sortie donnée en considérant un état figé des entrées. Ces conditions restreignent l'application pratique de la méthode, malgré une précision correcte (environ 90%) comparée aux simulations de type HSPICE.

Une autre approche qui vise à la caractérisation de la propagation de l'impact des particules le long des chemins sensibles du circuit a été présentée par Zhang et al. [66]. Cette caractérisation est faite à partir de simulations SPICE, en prennant en compte le masquage électrique utilisant la durée, l'amplitude et la distribution des entrées. Concernant le masquage logique les auteurs proposent l'utilisation de BDDs. Un BDD statique pour modéliser le circuit sans fautes et un autre BDD dynamique pour modéliser les éventuels impacts de particules. Aussi, le masquage temporaire est pris en compte en appliquant un facteur à chaque impact qui prend en compte la durée du glitch et de la fenêtre de lecture. Pour appliquer leur méthode à des circuits de grande taille, les auteurs proposent une partition des BDDs, ce qui comporte une perte de précision car les corrélations entre les signaux sont négligées.

Le travail de Zivanov et Marculescu [67] propose une approche similaire aux travaux présentés ci-dessus. Ils utilisent le modèle de masquage électrique d'Omana [65]. Pour le masquage logique, ils associent des BDDs et ADDs (Algebraic Decision Diagram) à chaque porte logique. La probabilité de propagation d'une faute à partir d'un noeud donné jusqu'à une sortie donnée est trouvée en construisant tous les ADDs du chemin à traverser. Dans ce travail, deux types d'analyse de fiabilité sont faits :

- Probabilité d'une sortie donnée étant donné que les portes qui définissent l'état de cette sortie peuvent être défaillantes
- Probabilité que les sorties dépendantes d'un noeud donné soit erronées étant donnée une faute dans ce noeud.

Pour ces analyses, les vecteurs d'entrée sont générés de manière aléatoire. Les résultats montrent une bonne corrélation avec des simulations HSPICE, mais les auteurs ne donnnent pas d'information sur le nombre de vecteurs d'entrée qui doivent être générés pour évaluer le masquage logique.

Rao et al. [68] utilisent une approche similaire à [65] et [67], mais ils proposent l'utilisation de la fonction de densité de probabilité de Weisbull pour modéliser les glitchs, contrairement au modèle trapézoïdal proposé auparavant. Ils utilisent des simulations SPICE, contenant la description des caratéristiques des portes logiques ainsi que des impacts pour évaluer leur propagation. Le masquage logique est estimé en générant 500.000 vecteurs d'entrée, ce qui limite son application car le coût de calcul est important.

D'autres travaux utilisant les simulations de propagation de glitchs de courant ont été proposés dans [69], [71] et [70]. Les travaux de Wang, [69] et [71], proposent l'utilisation de modèles MOSFET pour évaluer avec précision les effets non-linéaires des transistors MOS fortement sub-microniques. Le travail d'Entrena [70] propose une approche novatrice basée sur l'émulation et non pas la simulation des glitchs, ce qui accélère le processus d'estimation. Ces travaux sont limités car ils sont restreints au masquage électrique et

ignorent le masquage logique, ce qui limite leur précision.

Matrices de Transfert de Probabilité

Les matrices de transfert de probabilité (ou PTM de l'anlgais *Probability Transfer Matrix*) [80, 81] sont à la base de plusieurs techniques qui ont été développées ultérieurement ainsi que les travaux réalisés au cours de cette thèse. La méthode PTM est l'une des premières à utiliser une formulation matricielle pour modéliser le comportement aléatoire tant des cellules élémentaires utilisées dans la logique que de la totalité du système.

Pour un circuit de logique combinatoire la matrice PTM caractérisant ce système contient toutes les probabilités des possibles états de sortie étant donné un état à l'entrée. Si on appelle M la matrice PTM, l' élément $M(i,j)$ contient la probabilité d'avoir l'état i en sortie en sachant que l'état des entrées est j, c'est-à-dire : $P(output = i|input = j)$. Pour un système avec N entrées et M sorties, la taille de la matrice PTM est $2^N \times 2^M$, car tous les états y sont représentés. À titre d'exemple, considérons une porte AND à 2 entrées et une probabilité d'erreur ε, représentée en figure 2.6.

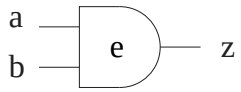


Fig. 2.6 – Porte AND à deux entrées et probabilité d'erreur ε.

Sa matrice PTM est construite à partir du tableau 2.2

ab	$z=0$	$z=1$
00	$1-\varepsilon$	ε
01	$1-\varepsilon$	ε
10	$1-\varepsilon$	ε
11	ε	$1-\varepsilon$

Tab. 2.2 – Table de vérité probabiliste de la porte de la figure 2.6.

Pour un état d'entrée donné, la sortie attendue en conditions normales, c'est-à-dire sans erreur, a une probabilité $1-\varepsilon$, la sortie erronée a une probabilité ε. Ce tableau est construit à l'aide de la table de vérité de la porte AND. La matrice PTM de cette porte est donc :

$$PTM_{AND} = \begin{pmatrix} 1-\varepsilon & \varepsilon \\ 1-\varepsilon & \varepsilon \\ 1-\varepsilon & \varepsilon \\ \varepsilon & 1-\varepsilon \end{pmatrix}$$

Avec les principes utilisés pour trouver la matrice PTM, nous pouvons aussi construire une matrice idéale de transfert (ITM). Cette matrice représente le comportement de l'élément considéré en l'absence de fautes, c'est-à-dire avec un comportement idéal. La matrice ITM modélise donc un comportement déterministe et non aléatoire, les probabilités qu'elle contient ont une valeur soit de 1 (évènement sûr) soit de 0 (évènement impossible). La

matrice ITM de la porte AND présentée précédement (en figure 2.6) est :

$$ITM_{AND} = \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}$$

La valeur de la fiabilité est donnée par la probabilité que l'élément réalise une opération correcte. Cette probabilité peut être exprimée de manière intuitive ainsi :

$$R = \sum_{\forall i} p(out = correct|input = i)p(input = i) \tag{2.1}$$

C'est-à-dire, la somme des probabilités de réaliser une opération correcte étant donnée une entrée, pondérées par la probabilité d'occurrence de chaque entrée. Si on dénote $\overrightarrow{v}$ le vecteur contenant les probabilités des entrées d'un élément, et en considérant ses matrices PTM et ITM, sa fiabilité est exprimée ainsi :

$$R = ||\overrightarrow{v} \cdot ITM * PTM|| \tag{2.2}$$

Le produit matriciel $ITM * PTM$ donne la matrice résultante du produit de toutes les paires d'éléments des matrices ITM et PTM. L'opérateur $||\cdot||$ est la norme vectorielle.

L'obtention de la matrice ITM est intéressante parce qu'elle permet d'identifier les états corrects afin de pouvoir calculer la fiabilité de l'élément sous étude. Ainsi, on peut aussi exprimer la fiabilité comme :

$$R = \sum_{ITM(i,j)=1} PTM(i,j)p(i) = \sum_{ITM(i,j)=1} p(j|i)p(i) \tag{2.3}$$

Jusqu'ici nous avons présenté comment l'estimation de la fiabilité est réalisée à partir des matrices PTM et ITM d'un circuit. Par la suite nous allons présenter comment ces deux matrices sont construites à partir des matrices élémentaires des éléments (portes logiques et connexions) qui constituent le circuit.

L'approche PTM utilise deux calculs matriciels différents pour calculer la PTM globale de deux éléments selon qu'ils sont connectés en série ou en parallèle. Les processus permettant d'obtenir la matrice PTM et ITM étant équivalents, nous parlerons dorénavant exclusivement du processus de construction de la matrice PTM globale pour des raisons de simplicité. Le même processus sera appliqué pour la construction de la matrice ITM.

La matrice PTM globale de deux éléments en série est le produit des matrices PTM élémentaires des deux éléments. Considérons un circuit G formé par les portes g_1 et g_2 connectées en série, caractérisées chacune par les matrices PTM_1 et PTM_2 respectivement. La matrice PTM globale, PTM_G contient les éléments (i, j) représentant la probabilité que la porte g_2 ait une sortie égale à j donné que g_1 est à l'état i en entrée. Cette probabilité est donnée par l'addition de toutes les valeurs des signaux intermédiaires qui sont à la fois sorties de g_1 et entrées de g_2. L'expression est donnée par : $PTM_G(i,j) = \sum_k PTM_1(i,k)PTM_2(k,j)$. Cette opération, appliquée pour toutes les valeurs possibles de i, j et k correspond au produit $PTM_G = PTM_1 \cdot PTM_2$.

La matrice globale de deux éléments en parallèle correspond au produit de kronecker des matrices PTM des deux éléments. Si on considère deux matrices, A et B :

$$A = \begin{pmatrix} a_{11} & \dots & a_{1N} \\ \vdots & \ddots & \vdots \\ a_{M1} & \dots & a_{MN} \end{pmatrix}$$

$$B = \begin{pmatrix} b_{11} & \dots & a_{1L} \\ \vdots & \ddots & \vdots \\ b_{K1} & \dots & a_{KL} \end{pmatrix}$$

Le produit de kronecker de ces deux matrices est défini ainsi :

$$A \otimes B = \begin{pmatrix} a_{11} \cdot B & \dots & a_{1L} \cdot B \\ \vdots & \ddots & \vdots \\ b_{K1} \cdot B & \dots & a_{KL} \cdot B \end{pmatrix}$$

La matrice résultante du produit de kronecker de deux matrices contient tous les produits élément par élément des deux matrices. De ce fait, si les tailles des matrices A et B sont $2^N \times 2^M$ et $2^L \times 2^K$ respectivement, la taille de la matrice résultante est $2^{NL} \times 2^{MK}$.

Considérons un circuit G formé par deux portes en parallèle, g_1 et g_2 et leurs matrices caractéristiques PTM_1 et PTM_2 respectivement. La matrice PTM globale du circuit G, PTM_G, contient dans chaque élément les deux probabilités de deux états sortie de g_1 et g_2 conditionnés par deux états d'entrée de g_1 et g_2. Ceci correspond au produit de kronecker des deux matrices PTM des deux portes g_1 et g_2.

Les circuit réels sont formés par des structures plus compliquées que la simple connexion de deux éléments soit en série soit en parallèle. Cependant, ils peuvent être décomposés en une connexion multiple d'éléments. L'approche PTM utilise une décomposition des circuits en plusieurs groupes connectés en série entre eux, où chacun de ces groupes est formé par plusieurs composants connectés en parallèle entre eux. Nous avons utilisé le circuit C de la figure 2.7 pour illustrer cette procédure. Pour ce circuit, la matrice PTM_C est le

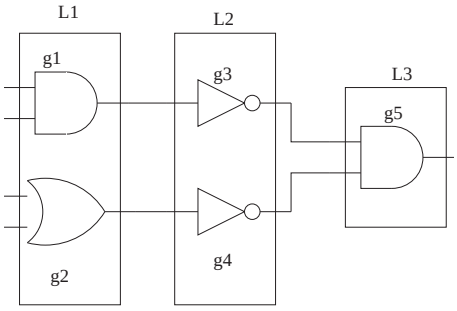


Fig. 2.7 – Exemple de la décomposition d'un circuit logique en niveaux logiques.

produit de kronecker des matrices PTM des trois groupes montrés sur la figure : $PTM_C = PTM_{L1} \cdot PTM_{L2} \cdot PTM_{L3}$, avec $PTM_{L1} = PTM_{g1} \otimes PTM_{g3}$, $PTM_{L2} = PTM_{g2} \otimes PTM_{g4}$ et $PTM_{L4} = PTM_{g5}$.

A part les matrices PTM de portes logiques standard (OR,XOR,AND, etc.) il est intéressant de définir la matrice PTM de trois topologies correspondant à différents types de connexions entre portes (figure 2.8). Ce type de connexions aide à la construction de la matrice PTM globale. Par exemple, le circuit de la figure 2.9 contient une connexion du type fanout. Par conséquent, la matrice PTM correspondant au premier niveau logique, est calculée à partir des matrices PTM du type *wire*, correspondant à une connexion simple, et du type *fanout*, correspondant à un signal connecté à plusieurs éléments. Ainsi, la



FIG. 2.8 – Différentes connexions définies pour aider à la construction de la matrice PTM globale.

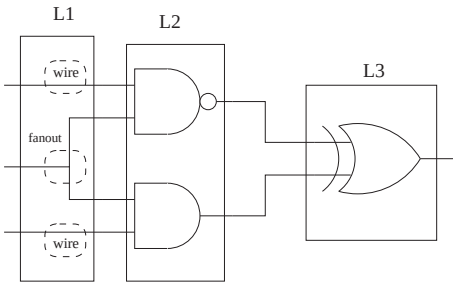


FIG. 2.9 – Exemple de circuit contenant des connexions spéciales pour l'approche PTM

.

matrice PTM du niveau L_1 est donnée par l'expression $PTM_{L1} = PTM_{wire} \otimes PTM_{fanout} \otimes PTM_{wire}$.

L'approche PTM est intéressante car elle introduit une formulation matricielle pour la modélisation de circuits, ce qui a donné suite à différentes approches comme nous le verrons par la suite du document. Aussi, l'approche PTM est très utile car elle permet de modéliser plusieurs types de fautes dans les portes logiques. Les fautes du type stuck-at sont modélisées avec une probabilité d'erreur égale à 1 dans la matrice PTM, les fautes transitoires sont modélisées avec une probabilité $0 < \varepsilon < 1$. Le modèle du circuit, une fois la matrice PTM globale construite, permet l'étude de la fiabilité pour différentes configurations des entrées, en faisant varier leur probabilité d'occurrence. Le résultat fourni est exact. En effet, la modélisation à partir de la matrice PTM, énumère tous les états possibles du circuit un par un, ce qui élimine les problèmes qui apparaissent avec la corrélation inter-signaux. Malgré ces avantages, l'approche PTM présente des limitations qui empêchent son utilisation pour tous types de circuits. Notamment, la taille de la matrice PTM d'un circuit croît de manière exponentielle avec le nombre d'entrées et sorties du circuit. Nous rappelons que pour un circuit à N entrées et M sorties, la taille de la matrice est $2^N \times 2^M$. Ceci limite l'application de la méthode à des circuits de taille réduite. La décomposition en niveaux logiques d'un circuit afin de pouvoir appliquer la méthode présente une forte complexité d'implémentation car elle est difficilement paramétrisable, ce qui limite l'application industrielle de la méthode.

Le travail présenté dans [83] propose des modifications de l'approche PTM de base pour réduire la taille des matrices intermédiaires utilisées pour le calcul de la matrice PTM globale. Lors de la modélisation des connexions comme représenté sur la figure 2.8, les matrices PTM sont formées par une majorité de valeurs mises à 0. Lors du produit de kronecker de ces matrices, une grande partie de l'information contenue correspond à des états du circuit qui ne sont pas atteignables, ce qui rend les éléments qui contiennent cette information inutiles. Les auteurs proposent une méthodologie pour utiliser seulement

l'information utile des matrices PTM, ce qui comporte une réduction des coûts de calcul sans compromettre l'exactitude de la méthode. Cependant, le gain apporté par cette méthodologie reste limité et l'application de la méthode à des circuits de taille importante n'est toujours pas réalisable.

Champs Markoviens Aléatoires

Le travail présenté dans [73] propose l'utilisation de champs Markoviens aléatoires, (ou MRF, de l'anglais *Markov Random Field*) pour la conception de circuits [72]. Un MRF est un modèle graphique dans lequel un ensemble de variables aléatoires ayant des propriétés markoviennes sont décrites par un graphe non orienté.

Les auteurs affirment que l'utilisation de MRF permet la conception implicite de circuits robustes aux fautes à partir d'éléments qui ne sont pas fiables. Par contre, leur approche est limitée à des circuits extrêmement petits et n'acceptent pas la présence de signaux reconvergents.

Probabilistic Gate Model

Un autre modèle a été présenté par Han et al. dans [75], appelé Probabilistic Gate Models (PGM). Cette approche associe chaque signal à une variable aléatoire qui détermine la probabilité de ses états. Pour ce faire, les auteurs utilisent le concept de probabilité de signal, défini comme la probabilité que le signal soit à l'état logique 1.

Pour obtenir le PGM d'une porte il faut réaliser l'opération suivante :

$$X_i = \begin{bmatrix} p_i & 1-p_i \end{bmatrix} \cdot \begin{bmatrix} 1-\varepsilon \\ \varepsilon \end{bmatrix} \tag{2.4}$$

Le terme p_i désigne la somme des minterms des entrées qui produisent la valeur logique 1. Le terme ε signifie la probabilité que la porte réalise une opération erronée. X_i est la probabilité que la sortie de la porte i soit à 1. Par exemple, pour une porte NAND à deux entrées, le terme p_i devient $p_i = 1 - X_{input1}X_{input2}$, où X_{input1} et X_{input2} sont les probabilités des entrées d'avoir une valeur logique égale à 1. Cette méthode assume implicitement que les signaux d'entrée d'une porte sont statistiquement indépendants, ce qui n'est pas réaliste dans le cas général. Ceci se traduit par une perte de précision.

Les auteurs proposent une modification de l'algorithme, consistant en une partition du circuit en niveaux topologiques afin de traiter correctement l'effet des signaux reconvergents, mais cette modification comporte une explosion du nombre d'états à considérer, ce qui limite l'application pour des circuits de grande taille.

Un travail plus récent des mêmes auteurs [76] basé sur le PGM propose une approche modulaire, consistant en un découpage du circuit en plusieurs modules, qui sont analysés de manière hiérarchique. Les corrélations entre les signaux sont prises en compte seulement à l'intérieur des modules, mais pas entre différents modules, ce qui comporte une perte de précision. Les auteurs comparent la précision de leurs résultats à des simulations Monte-Carlo du circuit sous étude, ayant une différence d'environ 9% dans le pire cas.

Réseaux Bayesiens

Rejimon et Bhanja ont présenté un modèle qui est basé sur l'utilisation de réseaux bayesiens [77]. Un réseau bayesien est un modèle graphique probabiliste qui représente la fonction de probabilité jointe d'un ensemble de variables aléatoires à partir d'un graphe

acyclique orienté (DAG). Dans ce graphe, les noeuds correspondent aux variables aléatoires et les arcs aux dépendances causales directes.

Dans le modèle qu'ils proposent, chaque porte est représentée par la probabilité de sortie conditionnée par ses entrées. Le circuit entier est modélisé à partir de ces probabilités en prenant compte des interconnections entre les portes et représenté sous forme de DAG.

Pour un circuit donné, on dénote ses entrées comme $Z_1, \ldots, Z_N$, les signaux internes comme $X_1, \ldots, X_M$ et les sorties $Y_1, \ldots, Y_K$; ceci représente le circuit sans faute. D'une manière équivalente, on peut dénoter les signaux internes et les sorties du circuit en considérant la présence de fautes comme suit : $X_1^e, \ldots, X_M^e$ et $Y_1^e, \ldots, Y_M^e$. Alors, la présence d'une erreur peut être détectée si les sorties des deux circuits sont différentes. Ainsi, pour la sortie i quelconque, ceci peut s'exprimer avec la formule suivante (où $\oplus$ représente l'opération ou exclusif) :

$$E_i = Y_i^e \oplus Y_i \tag{2.5}$$

La probabilité d'une erreur sur la sortie i est donc $P(E_i = 1)$.

Pour calculer la probabilité d'erreur d'une sortie, les auteurs proposent de construire un graphe composé de deux sous-graphes, l'un correspondant au circuit sans erreur et l'autre correspondant au circuit en présence d'erreurs. Les auteurs proposent l'introduction de noeuds supplémentaires correspondant aux portes XOR, dont les entrées sont les sorties des circuits sans faute et de celui avec fautes, servant à signaler la présence d'une erreur sur une sortie. La probabilité d'erreur d'une sortie est donnée par la probabilité que les noeuds supplémentaires soient dans l'état 1 logique. Par exemple, dans la figure 2.10, se trouve représenté un circuit simple, sa représentation en absence de faute et celle en présence de fautes, les portes XOR servant à signaler les erreurs et le graphe correspondant.

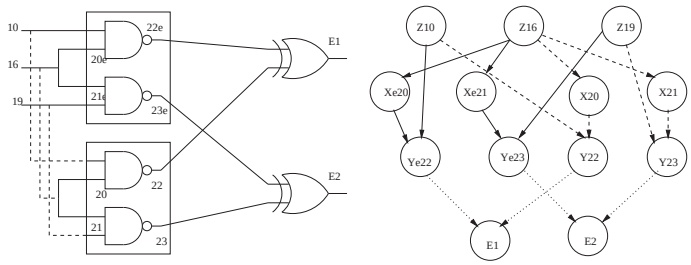


FIG. 2.10 – Exemple de la représentation dans le modèle des réseaux bayesiens.

Les auteurs affirment que leur approche est exacte, cependant, les résultats publiés de leur travail montrent une différence importante par rapport aux résultats d'autres méthodes exactes. De plus, les auteurs affirment que leur approche utilise la représentation minimale de l'espace de probabilités des états du circuit, ce qui n'est pas demontré par leur travaux.

Probabilistic Model Checking

Les travaux de Badhuri et al. [78] introduisent l'utilisation de chaines de markov en temps discret (DTMC) pour développer un modèle de calcul de fiabilité appelé Probabilistic Model Checking (PMC). Ce modèle utilise la description des cellules logiques en forme de DTMC et la matrice de transition qui y est associée. La fiabilité du circuit est obtenue à partir d'une partition du circuit en plusieurs niveaux logiques et la modélisation de chacun de ces niveaux à partir d'une DTMC. Ainsi, les probabilités de sortie d'un niveau sont

utilisées comme probabilités d'entrée du niveau suivant. L'inconvénient principal de cette approche est le nombre d'états nécessaires pour des DTMCs qui croît exponentiellemnet avec la taille du circuit. Cependant, une réduction du modèle est possible en utilisant des MTTBs (Multi-Terminal Binary Decision Diagrams) car les matrices de transition sont en général très creuses. Les auteurs affirment que cette approche est compatible avec PGM et PTM.

Probabilité de Propagation de l'Erreur (EPP)

Le travail de Asadi et al. [79] présente une approche systématique pour le calcul du taux d'erreur dans un cône de logique combinatoire. Les auteurs présentent un calcul novateur de la probabilité de propagation d'erreurs à partir de la probabilité des signaux et de la structure topologique du circuit. Dans cette approche, les chemins structurels sont extraits en partant de l'endroit où l'erreur a lieu vers toutes les sorties atteignables de ce point. Selon la localisation de l'erreur, les signaux et les portes sont déclinés en trois groupes :

1. **on-path signal** est un signal qui est dans le chemin entre l'endroit de l'erreur et une sortie atteignable
2. **on-path gate** est une porte avec au moins un signal *on-path*
3. **off-path signal** est un signal qui n'est pas *on-path* mais connecté à l'entrée d'une porte *on-path*.

Le calcul de la probabilité de propagation d'erreur est effectué en parcourant le circuit et en utilisant les probabilités des signaux *off-path* et des règles de propagation, pour lequelles chaque type de porte a ses propres expressions analytiques permettant de calculer la propagation.

Les auteurs affirment que leur approche permet de calculer le SER avec un gain de 4 et 5 ordres de grandeur par rapport aux approches par simulation avec des résultats assez similaires (12.3% dans le pire cas). Les auteurs affirment que leur approche est capable de traiter correctement les signaux reconvergents qui sont des *on-path*, mais il est nécessaire d'estimer aussi les probabilités des signaux **off-path**, ce que les auteurs négligent. Aussi, le temps de calcul qu'ils montrent dans leurs travaux ne prend pas en compte cette estimation.

Méthodes basées sur l'observabilité

Choudhury et al. [82] ont présenté une approche basée sur l'observabilité des signaux de sortie d'un circuit pour l'évaluation de sa fiabilité. L'observabilité est une métrique qui est souvent utilisée dans le domaine du test. Dans le domaine de la fiabilité, l'observabilité d'un signal quelconque du circuit est la probabilité qu'une faute dans ce signal affecte la sortie du circuit. Les auteurs affirment que les observabilités peuvent être obtenues en utilisant des approches symboliques, BDDs ou par simulation.

Le principe de cette technique est assez simple. En considérant un circuit avec une seule sortie y, si Ω est l'ensemble de portes dans le circuit et $G \subseteq \Omega$ l'ensemble de portes qui ont souffert une faute, la sortie y sera fautive si $|G|$ est impair ($|G|$ est la cardinalité de l'ensemble G), car si $|G|$ est pair les fautes seraient masquées entre elles. L'expression permettant de calculer la fiabilité à partir de cette idée se trouve dans l'équation 2.6, où $\delta_y(\vec{\epsilon})$ est la probabilité d'erreur de y, ϵ_i est la probabilité d'erreur de la porte i du circuit et o_i est l'observabilité de cette porte i.

$$\delta_y(\vec{\epsilon}) = \frac{1}{2}(1 - \prod_{i \in \Omega}(1 - 2\epsilon_i o_i)) \tag{2.6}$$

Les auteurs affirment que cette technique est d'autant plus précise que les probabilités ϵ_i sont petites et que le circuit est petit, ce qui réduit en grande partie son applicabilité. Les sources d'inexactitude de l'estimation sont dues au fait que :

- L'approche ne considère pas de corrélation entre les signaux.
- L'observabilité des portes est calculée en considérant l'absence des fautes, ce qui n'est réaliste.

Analyse de la fiabilité "simple passe"

Dans le même document, Choudhury et al. [82] proposent une autre approche appelée *single-pass reliability analysis*. L'idée centrale de cette technique est que l'observation d'une erreur sur une sortie du circuit est due à deux phénomènes :

- Apparition d'une faute dans une porte
- Propagation de cette faute à tavers des portes sur son chemin logique.

Quand ces deux phénomènes se combinent, la probabilité d'erreur à la sortie de la porte g est donnée par :

- $p_{(g_{0\to1})}$, probabilité d'erreur quand la valeur correcte est 0.
- $p_{(g_{1\to0})}$, probabilité d'erreur quand la valeur correcte est 1.

Initialement, les $p_{(x_i1\to0)}$ et $p_{(x_i0\to1)}$ pour les entrées x_i de la porte sont connues. Ainsi, les probabilités des x_i sont combinées pour obtenir un vecteur des poids des entrées qui est en même temps combiné avec les probabilitées d'occurrence de faute de la porte pour obtenir les probabilités $p_{(g_{0\to1})}$ et $p_{(g_{1\to0})}$. Ainsi, la probabilité d'erreur est décrite dans l'équation 2.7 :

$$\delta_y(\epsilon) = p(y=0)p(y_{0\to1}) + p(y=0)p(y_{1\to0}) \tag{2.7}$$

Cette opération est appliquée une fois sur chacune des portes en suivant un ordre topologique, ce qui correspond à une complexité linéaire de la méthode. Cependant, les calculs de probabilités proposés par les auteurs considèrent implicitement que les signaux sont indépendants, ce qui est faux dans le cas général et implique une perte de précision de la méthode.

Probabilistic Binomial Model

Dans [74], les auteurs proposent l'utilisation d'un modèle binomial pour l'analyse de la fiabilité des circuits combinatoires, appelé PBR (*Probabilistic Binomial Model*). Pour un circuit combinatoire générique, tel que représenté en figure 2.11, où x représente le vecteur des entrées, de taille m, et y le vecteur des sorties, de taille n, la fiabilité, R, de ce circuit peut être exprimée comme suit :

$$R = \sum_{j=0}^{2^m-1} p(y = correct|x_j)p(x_j) \tag{2.8}$$

Le terme $p(y = correct|x_j)$ nous donne la probabilité d'avoir un état correct en sortie étant donné l'état j des entrées, et $p(x_j)$ étant la probabilité d'avoir l'état j à l'entrée.

Considérons un circuit constitué de w portes logiques, y_i est la sortie de la porte i de ce circuit. Il est défini le vecteur $\hat{e}$ comme vecteur de configuration d'erreur. Ce vecteur contient les éléments $(e_1, e_2, \ldots, e_w)$, où $e_i = 1$ représente l'injection d'une faute à la sortie de la porte g_i et $e_i = 0$ représente un fonctionnement correct de la porte. Le nombre d'erreurs injectées est donc le nombre de 1 dans le vecteur $\hat{e}$. L'ensemble de vecteurs



Fig. 2.11 – Circuit combinatoire générique.

contenant k erreurs est désigné comme $\hat{e}_{w:k}$. Le nombre de vecteurs pour chaque valeur de k est donné par la combinatoire $C_k^w = \frac{w!}{(w-k)!k!}$. $\hat{e}_{w:0}$ dénote le vecteur ne contenant pas d'erreurs. D'autre part, si l'on considère que $y(x_j, \hat{e}_{w:k})$ représente la sortie lorsque l'entrée est x_j et les fautes sont désignées par $\hat{e}_{w:k}$, on peut affirmer que $y(x_j, \hat{e}_{w:0})$ détermine le comportement correct de la sortie, puisqu'elle est produite en l'absence de faute.

Par conséquent, l'équation suivante est vraie dès qu'il se produit un masquage d'erreur dans la sortie du circuit :

$$\overline{y(x_j, \hat{e}_{w:0}) \oplus y(x_j, \hat{e}_{w:k})} = 1 \tag{2.9}$$

Où l'opérateur $\oplus$ dénote la fonction logique OU exclusif.

Supposons maintenant que les portes ont une probabilité q de générer correctement la sortie. Alors, la probabilité d'occurrence de k erreurs dans un circuit avec w portes est donnée par la fonction $f(q)$:

$$f(q) = (1-q)^k q^{w-k} \tag{2.10}$$

Cette équation correspond au modèle de comportement binomial des variables aléatoires, d'où le nom du modèle présenté.

La fiabilité, R, peut maintenant être décrite comme suit :

$$R = \sum_{k=0}^{w} f(q) c_k \tag{2.11}$$

Cette équation décrit la fiabilité comme une combinaison de toutes les configurations d'erreur possibles, où le facteur c_k est un poids pour chaque valeur de k qui pondère toutes les configurations d'erreur qui mènent vers un résultat correct de la sortie du circuit, c'est-à-dire, toutes les configurations qui masquent la présence de k fautes. Les coefficients c_k sont calculés avec l'équation suivante :

$$c_k = \sum_{l=1}^{C_k^w} \sum_{j=1}^{2^m-1} p(x_j) \overline{y(x_j, \hat{e}_{w:0}) \oplus y(x_j, \hat{e}_{w:k}(l))} \tag{2.12}$$

Le modèle PBR fournit un résultat exact. Par contre, il est difficilement applicable à des circuits de taille importante car le calcul des coefficients nécessite l'énumération de toutes les configurations d'erreur possibles. Cette opération est d'une complexité exponentielle $O(2^w)$, où w est le nombre de portes dans le circuit. Cependant, les auteurs affirment qu'il est possible de trouver un bon compromis entre la précision et le temps de calcul en négligeant le calcul des coefficients qui ont un faible impact, c'est-à-dire un poids mineur sur la fiabilité, c_k dans l'équation 2.11. En effet, une configuration comportant un nombre très élevé d'erreurs dans les portes est très peu probable, ainsi la valeur donnée par l'équation 2.10 va être très proche de 0.

Signal Probability Reliability

Le modèle SPR (Signal Probability Reliability) est basé sur l'approche PTM présentée ci-dessus [85, 86]. Là où le modèle PTM réalise une description globale du circuit pour calculer les différentes probabilités des signaux, l'approche SPR réalise un calcul matriciel local propageant les probabilités des signaux jusqu'aux sorties de porte en porte.

Le modèle SPR introduit la décomposition d'un signal du circuit en quatre états possibles : deux valeurs possibles pour les états logiques 0 ou 1, et deux valeurs possibles représentant l'état correct ou incorrect du signal notés c et i respectivement. Les quatre états possibles pour un signal S sont donc $0_c, 0_i, 1_i, 1_c$. Les états corrects correspondent aux états des signaux quand il n'y a pas de faute dans le circuit. Considérons la porte NAND à deux entrées de la figure 2.12.

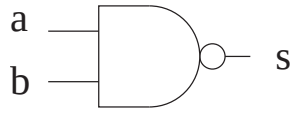


FIG. 2.12 – Porte NAND à 2 entrées.

Si les entrées a et b sont toutes les deux à l'état 0_c (0 correct), alors l'état attendu en sortie du signal s est 1_c. S'il se produit une erreur dans cette porte, l'état de la sortie sera inversé, il va être alors dans l'état 0_i. Considérons maintenant que le signal a est dans l'état 1_c et le signal b à l'état 1_i. S'il n'y a pas d'erreur dans la porte, le signal s sera dans l'état 0_i, car l'état correct de b $\overline{1_i} = 0_c$ aurait mené à une sortie égale à 1_c. Par contre, s'il se produit une erreur, l'état défectueux du signal b sera masqué, et l'état en sortie sera 1_c. Cette décomposition est utile pour modéliser le masquage logique des fautes.

SPR utilise une formulation matricielle pour propager les probabilités des signaux des entrées vers les sorties du circuit. A chaque signal est associée une matrice SPR contenant les probabilités d'occurrence des 4 états possibles. Pour un signal S, sa matrice SPR_S est :

$$SPR_S = \begin{pmatrix} p(s = 0_c) & p(s = 1_i) \\ p(s = 0_i) & p(s = 1_c) \end{pmatrix} = \begin{pmatrix} s_1 & s_2 \\ s_3 & s_4 \end{pmatrix}$$

Afin de simplifier la notation, la probabilité de chaque état pour le signal S est notée comme suit s_1, s_2, s_3, s_4, correspondant respectivement aux états 0_c, 0_i, 1_i et 1_c. La fiabilité du signal, R_S, est donnée par l'addition des probabilités représentant les états corrects du circuit, c'est-à-dire la trace de la matrice SPR :

$$R_S = p(s = 0_c) + p(s = 1_c) = s_1 + s_4 = Tr(SPR_S) \tag{2.13}$$

Considérons la porte NAND de la figure 2.12, caractérisée par les matrices PTM_{NAND} et ITM_{NAND}. Les signaux d'entrée sont caractérisés par leur matrice SPR :

$$SPR_A = \begin{pmatrix} a_1 & a_2 \\ a_3 & a_4 \end{pmatrix}, SPR_B = \begin{pmatrix} b_1 & b_2 \\ b_3 & b_4 \end{pmatrix}$$

La propagation des probabilités des signaux d'entrée à travers une porte logique, c'est-à-dire la matrice SPR_S du signal de sortie s de la porte, est obtenue à partir du produit des probabilités jointes des entrées I_{porte} et de la fonction de transfert de la porte, c'est-à-dire sa matrice PTM, dans ce cas PTM_{NAND}. Les probabilités obtenues avec ce produit sont

réparties entre les différents états du signal de sortie suivant la fonctionnalité de la porte, définie par la matrice ITM_{NAND}. La matrice I_{porte} contient les probabilités de toutes les combinaisons possibles des états des signaux d'entrée a et b, celle-ci est obtenue à partir du produit de kronecker des matrices SPR_A et SPR_B, chaque position de la matrice contient une paire $a_i \cdot b_j$.

$$I_{porte} = SPR_A \otimes SPR_B = \begin{pmatrix} a_1b_1 & a_1b_2 & a_2b_1 & a_2b_2 \\ a_1b_3 & a_1b_4 & a_2b_3 & a_2b_4 \\ a_3b_1 & a_3b_2 & a_4b_1 & a_4b_2 \\ a_3b_3 & a_3b_4 & a_4b_3 & a_4b_4 \end{pmatrix}$$

Les auteurs proposent le calcul de la matrice intermédiaire $P(S)$ avant l'obtention de la matrice de sortie SPR_S. La matrice $P(S)$ correspond au produit de la matrice I_{porte} avec la matrice PTM_{NAND}.

$$P(S) = I_{porte} \times PTM_{NAND} \tag{2.14}$$

Les éléments de la matrice SPR_S sont obtenus à partir de $P(S)$, selon les expressions suivantes, où $0, r$ et $1, r$ représentent une position dans la matrice $(colonne, ligne)$.

$$p(s = 0_c) = s_1 = \sum_{r:ITM_{NAND}(0,r)=1} P(S)_{[0,r]} \tag{2.15}$$

$$p(s = 1_i) = s_2 = \sum_{r:ITM_{NAND}(0,r)=0} P(S)_{[0,r]} \tag{2.16}$$

$$p(s = 0_i) = s_3 = \sum_{r:ITM_{NAND}(1,r)=1} P(S)_{[1,r]} \tag{2.17}$$

$$p(s = 1_c) = s_4 = \sum_{r:ITM_{NAND}(1,r)=0} P(S)_{[1,r]} \tag{2.18}$$

Cette opération permet une formulation beaucoup plus simple :

$$SPR_S = ITM_{NAND}^T \times (SPR_A \otimes SPR_B) \times PTM_{NAND} = ITM_{NAND}^T \times I_{porte} \times PTM_{NAND} \tag{2.19}$$

La méthode SPR propage les probabilités des signaux de porte en porte suivant un ordre topologique en utilisant sur chacune d'elles l'équation 2.19. Cette méthode nécessite donc une estimation de la distribution des signaux d'entrée. De ce fait, sa complexité est linéaire, ce qui est très avantageux pour l'analyse des circuits de grande taille, puisque l'estimation devient réalisable et ne souffre pas de limitations de mémoire ou de temps d'exécution trop élevés.

Malgré les avantages de cette approche, la plus grande limitation de SPR vient du fait qu'elle n'est pas exacte dans le cas général. L'estimation des probabilités jointes de deux signaux à partir du produit de kronecker de leurs matrices SPR considère implicitement que les deux signaux sont statistiquement indépendants. En effet, si l'on considère deux événements aléatoires α et β, $p(\alpha \cap \beta) = p(\alpha) \cdot p(\beta)$ seulement si α et β sont décorrélés. Le produit kronecker des deux matrices SPR, SPR_A et SPR_B, réalise les produits $p(a = etat_i) \cdot p(b = etat_j)$ pour tous les états possibles de a et b. Pour que des signaux soient décorrélés, il faut qu'ils n'aient aucun signal en commun en amont du circuit, c'est-à-dire, que le circuit ait une forme d'arbre, ou encore, qu'il n'y ait pas de signaux reconvergents. Sur la figure 2.13. on trouve la représentation de ces deux types de circuits.

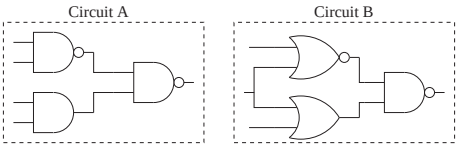


FIG. 2.13 – **Circuit A** : circuit en forme d'arbre, sans corrélation entre signaux. **Circuit B** : circuit contenant un signal reconvergent créant des corrélations.

Signal Probability Reliability Multi-Pass

La méthode SPRMP (Signal Probability Reliability Multi-Pass) a été proposée par les auteurs du modèle SPR afin de surmonter les limitations rencontrées par ce dernier en présence de signaux corrélés dans le circuit [85, 86]. Le principe de calcul de SPRMP reste le même que pour SPR, les probabilités sont propagées en utilisant l'équation 2.19.

Pour surmonter la limitation due à la corrélation entre signaux et faire une estimation exacte de la fiabilité, la méthode SPRMP réalise un algorithme itératif. Cet algorithme consiste à réaliser une analyse SPR du circuit, mais à la différence de la méthode originale, SPRMP propage seulement un état possible des signaux qui sont source de reconvergence. Les matrices SPR obtenues pour les signaux à chaque itération sont pondérées par la probabilité d'occurrence de l'état des sources de reconvergence pris en compte dans cette itération. Le nombre de passes à réaliser correspond au nombre d'états possibles des signaux qui sont sources de reconvergence.

Considérons le circuit de la figure 2.14. Les portes, G_i, sont caractérisées par leurs matrices ITM_{G_i} et PTM_{G_i}. Les signaux A,B et C sont caractérisés par leurs matrices SPR : SPR_A, SPR_B et SPR_C.

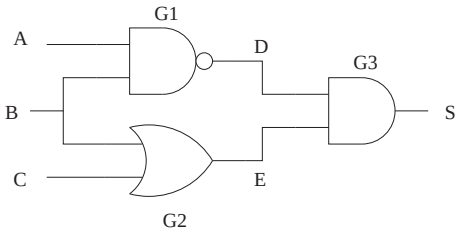


FIG. 2.14 – Circuit logique utilisé d'exemple pour illustrer la méthode SPRMP.

Le signal B est une source de reconvergence, car les signaux D et E en sont dépendants et sont tous les deux en entrée de la porte G_3. Pour propager un seul état de B il suffit de considérer ce seul état comme possible dans sa matrice SPR, donc, la position correspondant à l'état vaut 1 et tous les autres éléments de la matrices sont mis à 0. Si SPR_{B_i} est la matrice utilisée à l'itération i, avec $i = \{1, 2, 3, 4\}$, les calculs à réaliser pour ce circuit sont :

$$SPR_{D_i} = ITM_{G_1} \times (SPR_A \otimes SPR_{B_i}) \times PTM_{G_1} \tag{2.20}$$

$$SPR_{E_i} = ITM_{G_2} \times (SPR_{B_i} \otimes SPR_C) \times PTM_{G_2} \tag{2.21}$$

$$SPR_{S_i} = ITM_{G_3} \times (SPR_{D_i} \otimes SPR_{E_i}) \times PTM_{G_3} \tag{2.22}$$

La matrice SPR_S finale contenant l'information sur la fiabilité du circuit est calculée ainsi :

$$SPR_S = \sum_{i=1}^{4} SPR_{S_i} \times w_i \tag{2.23}$$

La matrice SPR_{B_i} utilisée à chaque itération et le poids w_i sont détaillés ci-dessous :

$$i = 1, SPR_{B_1} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, w_1 = p(b = 0_c) = b_1$$

$$i = 2, SPR_{B_2} = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}, w_2 = p(b = 1_i) = b_2$$

$$i = 3, SPR_{B_3} = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}, w_3 = p(b = 0_i) = b_3$$

$$i = 4, SPR_{B_4} = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}, w_4 = p(b = 1_c) = b_4$$

Où b_i sont les éléments de la matrice SPR_B.

La méthode SPRMP est exacte, mais elle montre des limitations liées à la complexité. Le nombre d'itérations à réaliser correspond au nombre d'états possibles des signaux qui sont sources de reconvergence. Étant donné que chaque signal a 4 états possibles, N signaux ont 4^N états possibles. Le nombre d'iterations croît donc exponentiellement avec le nombre de sources de reconvergence présentes dans le circuit, ce qui limite son application à des circuits avec peu de sources de reconvergences.

Les auteurs montrent dans leurs travaux qu'il est possible de négliger quelques sources de reconvergence afin de réduire le temps de calcul nécessaire, et que l'erreur d'estimation commise est d'autant plus grande que le nombre de sources de reconvergence négligées augmente.

Estimation de la fiabilité par theorem-proving

Hasan et al. [84] proposent une méthodologie basée sur la logique probabiliste de Von Neumann [56], en construisant des règles de propagation de fautes dans les portes logiques combinatoires. Les auteurs affirment que leur méthode est exacte, mais la comparaison des estimations obtenues est réalisée avec la méthode PGM, qui n'est pas exacte, et cela ne prouve donc pas leur affirmation. De plus, dans les règles qu'ils proposent pour la propagation dans les portes, la corrélation entre les entrées des portes est négligée. Finalement, aucune estimation de la complexité de cette approche n'est rapportée.

Méthodes basées sur la construction de super-portes

Le travail de Yu et al. [88] propose une approche basée sur la technique PTM visant à réduire la complexité de celle-ci. La construction des super-portes consiste en l'aggrégation de plusieurs portes logiques qui sont traitées séparamment du reste des super-portes construites pour l'étude du circuit. Les auteurs définissent un critère pour la construction des super-portes de manière à obtenir des ensembles disjoints. Ainsi, pour une porte logique w dans un circuit CC avec une seule sortie, la super-porte $SG(w)$ contenant w, est le plus petit sous-circuit de CC dont la sortie est w et les entrées de cette porte sont contrôlées

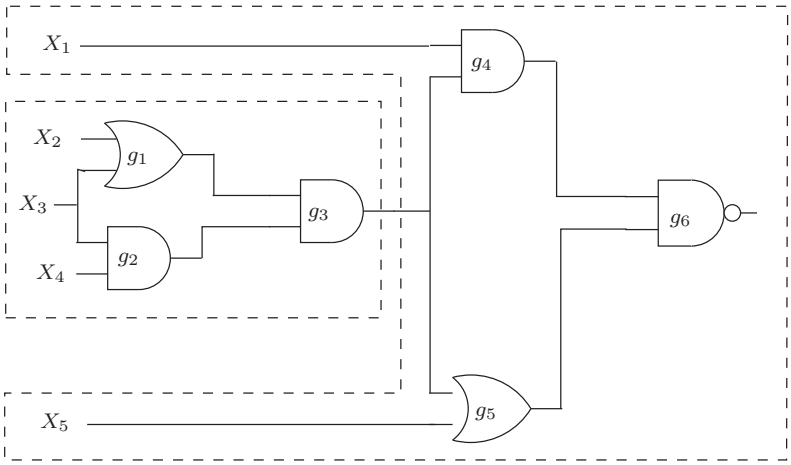


FIG. 2.15 – Circuit contenant deux super-portes marquées par des lignes pointillées.

indépendamment par les entrées de CC. Sur la figure 2.15 nous trouvons un exemple de construction de super-portes pour un petit circuit.

Les entrées des différentes super-portes sont indépendantes. Si cette condition ne peut être maintenue la super-porte devient le circuit entier, appliquant alors l'approche PTM primaire.

La partition en super-portes peut mener à des chemins de reconvergence partagés par plusieurs super-portes, ce qui comporte un traitement erroné des corrélations et par conséquent une perte de précision de la méthode. Dans ce travail, les auteurs appliquent cette approche pour l'estimation de la fiabilité dans la logique séquentielle et montrent des résultats comparant leurs travaux à une étude réalisée sur plusieurs cycles d'horloge avec des simulations Monte-Carlo. Ils n'estiment pas l'erreur commise dans l'estimation avec d'autres méthodes exactes.

Méthodes trigonométriques

Yu et al. [89] proposent une autre technique novatrice consistant en une décomposition de la probabilité de signal de manière trigonométrique. Leur approche vise à estimer la fiabilité d'un ciruit en présence de fautes ayant une très faible probabilité (10^{-8} à l'occurrence).

Par la combinaison des identités trigonométriques et la décomposition de taylor, une faute dans une porte est simulée comme une rotation. Les corrélations entre les signaux sont estimées à partir de la méthode BAM, qui a une complexité acceptable mais n'est pas exacte.

L'idée principale réside dans l'interprétation des probabilités comme des angles. Soit s un signal du circuit, et $p(s) = p(s = 1)$ et $p(\bar{s}) = p(s = 0)$, et $p(s) + p(\bar{s}) = 1$, sachant qu'il est possible trouver un angle θ tel que $\cos^2(\theta) + \sin^2(\theta) = 1$, tel que $\cos^2(\theta) = p(s)$ et $\sin^2(\theta) = p(\bar{s})$. Ainsi, à partir d'une décomposition en série de Taylor des angles modélisant les signaux et en prennant seulement quelques éléments de la décomposition, les auteurs proposent une approximation de la probabilité d'erreur.

Il n'y a pas d'information sur la complexité de la méthode. Les auteurs comparent son coût de calcul avec des simulations Monte-Carlo consistant en 320 millions d'itérations.

Leur méthode se montre beaucoup plus rapide mais avec une erreur d'environ 8% dans le pire des cas. L'intérêt de la méthode reste limité à des cas où la probabibilité d'erreur est très faible.

2.4 Conclusion

Les différentes approches et modèles existants pour l'analyse de la fiabilité des circuits de logique combinatoire ont été présentés. La performance de ces méthodes a été mesurée à partir de deux paramètres : la précision et la complexité.

La précision est un paramètre important car il est nécessaire de disposer de résultats pertinents pour plusieurs raisons. Dans le cas d'un éventuel durcissement du circuit le concepteur doit être en mesure d'assurer que sa stratégie est adéquate. Pour ce faire, l'utilisation de méthodes d'estimation exactes permet de valider de façon formelle le processus de durcissement. Ceci est particulièrement important lorsque l'on se place dans la perspective d'une conformité à la norme ISO26262 où le concepteur doit être en mesure de prouver formellement la fiabilité de son système.

Dans le cas de la conception de circuits robustes, les méthodes exactes permettent de justifier de manière sûre la fiabilité du système. Les méthodes exactes permettent de réaliser des stratégies de durcissement d'une manière optimale assurant un gain maximum en fiabilité pour un coût (surface, consommation, timing, etc) donné.

D'un autre coté, les méthodes d'estimation de la fiabilité doivent pouvoir être appliquées au plus grand nombre de circuits possible. Les approches souffrant d'une complexité trop importante ne peuvent être appliquées qu'à des circuits de taille reduite, ce qui limite énormément leur utilisation.

Il y a donc un réel besoin de disposer de méthodes qui soient exactes et de complexité acceptable pour être appliquées à toute taille de circuit. Cependant, les méthodes présentées dans ce chapitre sont soit exactes avec une complexité exponentielle, soit approximées et de complexité linéaire ou polynomiale. Il y a donc un manque d'outils pour l'estimation de la fiabilité de la logique combinatoire. L'objectif de cette thèse a été de développer des méthodes assurant un résultat exact tout en réduisant la complexité des approches exactes de l'état de l'art.

Un des effets qui limite le plus le développement de méthodes exactes est la corrélation spatiale des signaux. Il est connu que le calcul de la corrélation est un problème NP-complet [60], ce qui rend énormément difficile la conception de méthodes ayant les proprietés désirées. Une bonne alternative pourrait être l'invention de méthodes permettant de borner l'erreur commise par des approches qui ne sont pas exactes.

Chapitre 3

Proposition de méthodes d'estimation de la fiabilité dans la logique combinatoire

3.1 Introduction

L'objectif principal des travaux de recherche développés au cours de cette thèse a été d'obtenir un modèle d'estimation de la fiabilité dans la logique combinatoire présentant un bon compromis entre la complexité des calculs et la précision des résultats obtenus. Dans ce chapitre, nous proposons un modèle basé sur l'utilisation des probabilités conditionnelles des signaux sous forme de matrice. Notre modèle utilise les matrices de probabilités d'erreur dans les portes [80, 81] et le concept de fiabilité du signal [85, 86] décrits dans le chapitre 2. La nouvauté du modèle consiste en l'utilisation des probabilités conditionnées pour la représentation des états des signaux afin de bien prendre en compte les effets des corrélations et ainsi les traiter avec une complexité moindre.

Dans un premier temps, nous montrons comment l'utilisation de probabilités conditionnées permet de calculer de manière exacte la probabilité de deux signaux corrélés. Ensuite, nous présentons une formulation matricielle pour l'utilisation de ces probabilités. Cette formulation donne lieu à deux méthodes. La première offre un traitement plus efficace des corrélations tandis que la deuxième permet de réduire la complexité du calcul des probabilités des signaux corrélés. Finalement, nous discutons des limitations de ces modèles ainsi que des améliorations et réductions possibles.

3.2 Estimation des probabilités des signaux

Dans le chapitre précédent nous avons vu que l'un des obstacles principaux à l'obtention de modèles probabilistes d'une complexité abordable est le problème d'estimation de la probabilité jointe d'un ensemble de signaux corrélés. L'étude de la distribution des probabilités des signaux a de nombreuses applications, non seulement liées à l'estimation de la fiabilité, mais aussi pour l'estimation de la consommation, et de nombreuses techniques ont été proposées. Nous présentons par la suite des travaux concernant l'estimation de la probabilité des signaux, sans être appliqués directement au sujet de la fiabilité, le reste des principales approches concernant ce sujet ayant été présenté dans le chapitre précédent.

Le problème d'estimation de la probabilité des signaux est d'une complexité notable

[60], si bien qu'il est nécessaire de trouver un équilibre entre la précision de l'estimation et son coût de calcul. A cet effet, nous pouvons classer les approches en trois catégories, selon la nature des résultats qu'elles produisent :

- Calcul approximé : [90, 91, 92, 93]
- Calcul des bornes inférieures et supérieures de la probabilité du signal : [94],[95], [97]
- Calcul exact : [59]

Une des premières techniques proposées pour le calcul approximé est la méthode COP [90], dont la complexité est linéaire avec le nombre de portes du circuit. Les résultats ne sont pas exacts car elle suppose que les signaux sont statistiquement indépendants. Une amélioration de cette méthode est l'algorithme WAA (de l'anglais Weighted Averaging Algorithm) [91]. Cet algorithme estime les effets de premier ordre des sources de corrélation et est linéaire en nombre d'entrées du circuit et en nombre de sources de reconvergence, mais n'est pas exact sur l'estimation des effets des sources de reconvergence internes au circuit. La méthode STAFAN [92], qui est basée sur des simulations sans faute nécessite un grand nombre d'itérations pour obtenir une précision acceptable. Le travail présenté dans [93] propose deux méthodes pour corriger les effets de la corrélation des signaux. La première méthode, l'algorithme DWAA (Dynamic WAA), pondère les probabilités des signaux reconvergents et non pas des entrées du circuit et réalise un processus itératif faisant une mise à jour des probabilités à chaque passe. La deuxième méthode calcule la dépendance conditionnelle des signaux ainsi que leur probabilité, et elle est appelée méthode des coefficents de corrélation. La complexité de ces méthodes dépend de la structure du circuit, mais leurs résultats sont plus précis que ceux de la méthode WAA.

Un travail pionnier dans l'estimation des bornes inférieures et supérieures de la probabilité d'état d'un signal est l'algorithme de découpage (cutting algorithm) [94]. Cet algorithme consiste à diviser le circuit combinatoire en différents blocs selon la localisation des signaux reconvergents. Les probabilités sont propagées des entrées vers le sorties, et les bornes trouvées à la sortie d'un bloc sont utilisées à l'entrée du bloc suivant. La complexité et la précision de l'algorithme dépendent du découpage réalisé. Les auteurs n'ont pas démontré que l'algorithme de découpage calcule les bornes des probabilités de manière exacte. Une approche basée sur l'algorithme de Savir permettant de calculer les bornes de manière exacte a été présentée dans [95], mais ce travail ne spécifie pas la complexité de l'approche proposée et ne présente pas de résultats expérimentaux, ce qui permettrait de juger la validité de l'approche. Une autre méthode visant à l'estimation des bornes de probabilité basée sur l'utilisation des OPDDs [96] a été présentée dans [97]. Il s'agit d'une méthode itérative, et la précision dépend du nombre d'itérations à réaliser. Les auteurs affirment que leur approche fournit des bornes plus strictes que l'algorithme de découpage et les OPDDs, cependant le nombre d'itérations à réaliser pour y parvenir est très élevé.

Les approches de type exact sont fortement limitées par la complexité de leur calcul. Par exemple, l'approche présentée dans [59] calcule des équations qui ont une forme de polynomes donnant accès aux probabilités de signaux. Le nombre de termes à sauvegarder pour leur calcul croît de manière exponentielle.

L'objectif de nos travaux consiste à trouver une nouvelle approche de calcul des probabilités des signaux avec un résultat le plus exact possible en essayant de réduire au maximum sa complexité et permettant son application à l'analyse de la fiabilité.

3.2.1 Estimation de la probabilité jointe de signaux à partir des probabilités conditionnelles

Dans cette section, nous démontrons qu'il est possible de calculer la probabilité jointe d'occurrence de plusieurs états de signaux corrélés à partir du conditionnement des probabilités de l'état source de reconvergence. Tout au long de cette section nous utiliserons la décomposition en quatre états proposée dans [85, 86].

3.2.1.1 Définition du problème lié à la corrélation des signaux

L'approche d'estimation de la fiabilité à partir de la probabilité de signal nécessite un calcul des probabilités d'occurrence des signaux présents dans le circuit. En général, ce calcul est réalisé moyennant une propagation des probabilités, en partant des entrées primaires du circuit vers les sorties primaires en traversant les portes logiques dont le circuit est constitué. D'une manière générale, les méthodes basées sur la propagation probabiliste expriment les probabilités des signaux de sortie de la porte comme une fonction des probabilités des signaux d'entrée de la porte. Ainsi, l'estimation des probabilités jointes à l'entrée est nécessaire pour calculer correctement la probabilité en sortie. Ce calcul est trivial lorsque les signaux d'entrée sont statistiquement indépendants. Si l'on considère un ensemble de signaux $S = (s_1, s_2, \ldots, s_N)$, la probabilité qu'ils soient dans un état déterminé est :

$$p(s_1 = etat_i \cap s_2 = etat_j \cap \ldots \cap s_N = etat_k) = p(s_1 = etat_i) \cdot p(s_2 = etat_j) \ldots p(s_N = etat_k) \tag{3.1}$$

La probabilité jointe correspond au simple produit de la probabilité de chaque signal étant dans l'état donné. Ainsi, les probabilités pour chaque état de chaque signal peuvent être obtenues en réalisant une opération de propagation.

Comme il a été démontré dans le chapitre précédent, l'expression (3.1) n'est plus valable lorsque les signaux ne sont pas statistiquement indépendants. Pour illustrer notre propos considérons le circuit de la figure 3.1, où les portes et les entrées sont considérées idéales, c'est-à-dire, exemptées de toute erreur.

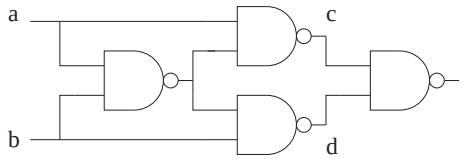


FIG. 3.1 – Exemple de circuit avec des signaux corrélés.

La table de vérité pour les signaux c et d, étant données les entrées a et b se trouve dans le tableau 3.1.

a	b	c	d	$p(c)$	$p(d)$	$p(c \cap d)$
0_c	0_c	1_c	1_c	$\frac{3}{4}$	$\frac{3}{4}$	$\frac{2}{4}$
0_c	1_c	1_c	0_c	$\frac{3}{4}$	$\frac{1}{4}$	$\frac{1}{4}$
1_c	0_c	0_c	1_c	$\frac{1}{4}$	$\frac{3}{4}$	$\frac{1}{4}$
1_c	1_c	1_c	1_c	$\frac{3}{4}$	$\frac{3}{4}$	$\frac{2}{4}$

TAB. 3.1 – Probabilités d'occurrence des différents signaux du circuit de la figure 3.1.

A partir de ce tableau, nous pouvons conclure que les probabilités des états pour les signaux c et d sont respectivement $p(c = 1_c) = 0.75$, $p(c = 0_c) = 0.25$ et $p(d = 1_c) = 0.75$, $p(d = 0_c) = 0.25$ si l'on considère les entrées équiprobables. Nous pouvons extraire les probabilités jointes des signaux c et d, $p(c \cap d)$, à partir de ce tableau. Ainsi, ces probabilités sont :

- $p(c = 0_c \cap d = 0_c) = 0$
- $p(c = 0_c \cap d = 1_c) = 0.25$
- $p(c = 1_c \cap d = 0_c) = 0.25$
- $p(c = 1_c \cap d = 1_c) = 0.5$

Il est directement observable que l'équation 3.1 ne peut pas être appliquée pour ce circuit car $p(c = etat_i \cap d = etat_j) \neq p(c = etat_i) \cdot p(d = etat_i)$.

La probabilité jointe de plusieurs événements corrélés admet plusieurs formules. Si l'on considère $Z = (z_1, z_2, \ldots, z_N)$ un ensemble de signaux statistiquement dépendants, la probabilité d'occurrence d'un état est exprimée comme suit :

$$p(z_1 = i \cap \ldots \cap z_N = k) = p(z_1 = i) + p(z_2 = j) + \ldots + p(z_N = k) - p(z_1 = i \cup \ldots \cup z_N = k) \tag{3.2}$$

$$\begin{aligned} p(z_1 = i \cap \ldots \cap z_N = k) &= p(z_1 = i | z_2 = j \ldots z_N = k) \cdot p(z_2 = j | z_3 = l \cap \ldots \cap z_N = k) \ldots \\ &\cdot \quad p(z_{N-1} = l | z_N = k) \cdot p(z_N = k) \end{aligned} \tag{3.3}$$

Ces deux formules contiennent seulement les probabilités des éléments dont on veut calculer les probabilités jointes. Il est difficile d'implémenter des procédures permettant d'utiliser ces deux expressions. Ainsi, il est donc nécessaire de mettre au point une approche permettant de calculer les probabilités jointes sans utiliser directement ces deux équations.

Dans la section suivante nous démontrons qu'il est possible de calculer la probabilité jointe de plusieurs signaux à partir de la propriété d'indépendance conditionnelle de certains événements aléatoires [87].

3.2.1.2 Signaux conditionnellement indépendants

Selon la théorie des probabilités, deux événements corrélés A et B, sont conditionnellement indépendants étant donné un troisième événement C si l 'occurrence ou la non-occurrence de A, et l'occurrence ou la non-occurrence de B sont indépendantes étant donné C. D'une manière plus intuitive, les événements A et B sont conditionnellement indépendants si, sachant que l'événement C se produit, l'occurrence ou non de l'événement A ne donne pas d'information sur l'événement B. De même, l'occurrence ou non de l'événement B ne donne pas d'information sur l'événement A. Le conditionnement par C élimine la corrélation entre A et B. Cette propriété est exprimée formellement par :

$$p(A \cap B | C) = p(A|C) \cdot p(B|C) \tag{3.4}$$

De manière équivalente :

$$p(A | B \cap C) = p(A|C) \tag{3.5}$$

Cette propriété peut être appliquée à un ensemble de signaux reconvergents dans un circuit logique afin de les décorréler. En effet, considérons la topologie représentée sur la figure 3.2. Cette topologie correspond typiquement à une structure de signaux corrélés.

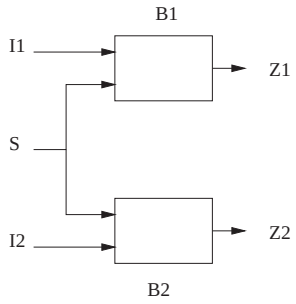


FIG. 3.2 – Topologie de signaux corrélés : application de l'indépendance conditionnelle.

S est l'ensemble des signaux qui sont à l'entrée des blocs combinatoires $B1$ et $B2$. $I1$ et $I2$ sont les ensembles de signaux qui sont seulement aux entrées du bloc $B1$ et $B2$ respectivement. Du fait du partage de l'ensemble d'entrées S, les sorties des blocs, $Z1$ et $Z2$ respectivement, sont corrélées. Cependant, les entrées $I1$ et $I2$ sont statistiquement indépendantes, et le fonctionnement (occurrence d'une faute ou non) des portes des blocs $B1$ et $B2$ est aussi indépendant. Pour un état de S figé, l'état de $Z1$ dépend seulement de $I1$ et du fonctionnement des portes de $B1$. De même, l'état de $Z2$ dépend de $I2$ et des portes de $B2$. Puisque, $I1$, $I2$ et le fonctionnement de $B1$ et $B2$ ne sont pas statistiquement corrélés, les sorties $Z1$ et $Z2$ sont statistiquement indépendantes pour un état de S figé, c'est-à-dire, elles sont conditionnellement indépendantes.

Les probabilités que $Z1$ et $Z2$ soient dans un état donné sont des fonctions de S, $I1$, $I2$, $B1$ et $B2$:

- $p(Z1 = i) = f1(S, I1, B1)$
- $p(Z2 = j) = f2(S, I2, B2)$

Pour le calcul de ces deux probabilités, figer l'état de $S = k$, équivaut à conditionner les événements correspondant aux états de $Z1$ et $Z2$, c'est-à-dire $(Z1 = i|S = k)$ et $(Z2 = j|S = k)$. Ainsi, les probabilités de $Z1$ et $Z2$ deviennent :

- $p(Z1 = i|S = k) = f1(I1, B1)$
- $p(Z2 = j|S = k) = f2(I2, B2)$

Puisque ces deux probabilités ne dépendent pas des mêmes éléments elles sont donc conditionnellement indépendantes, l'expression 3.6 est vérifiée :

$$p((Z1 = i \cap Z2 = j)|S = k) = p(Z1 = i|S = k) \cdot p(Z2 = j|S = k) : \tag{3.6}$$

Pour calculer $p(Z1 = i \cap Z2 = j)$ il suffit de considérer tous les cas possibles de conditionnement de S et d'appliquer une pondération à chacun d'eux. Le poids à prendre en compte correspond à la probabilité d'occurrence de l'état considéré, c'est-à-dire, $p(S = k)$. Ceci s'exprime par la formule suivante :

$$\begin{aligned} p((Z1 = i \cap Z2 = j) &= \sum_{\forall k} p((Z1 = i \cap Z2 = j)|S = k) \cdot p(S = K) \\ &= \sum_{\forall k} p(Z1 = i|S = k) \cdot p(Z2 = j|S = k) \cdot p(S = K) \end{aligned} \tag{3.7}$$

Nous vérifions ainsi qu'il est possible de calculer les probabilités jointes des états d'un ensemble de signaux corrélés à partir d'un conditionnement par les signaux qui sont source de reconvergence. Dans la section suivante nous montrons une manière de calculer les

probabilités conditionnées dans la logique combinatoire et comment cela peut s'appliquer au calcul de la fiabilité.

3.3 Modélisation par Matrices de Probabilités Conditionnées

Dans cette section nous présentons la méthode des matrices de probabilités conditionnées (CPM). A partir de la modélisation des signaux proposée dans [85, 86] (modèle SPR), et de la modélisation des portes par matrices PTM [80, 81], nous présentons une approche novatrice utilisant des probabilités conditionées permettant la décorrélation des signaux et l'obtention d'un résultat exact, de manière à ce que ces matrices puissent être stockées, et ainsi réduire le coût du calcul. Par la suite, nous présentons les différentes étapes nécessaires pour ce faire.

3.3.1 Calcul de probabilités conditionnées d'état d'un signal dans la logique combinatoire

La problématique du calcul de probabilités conditionnées des états de signaux se résume à figer l'état des signaux qui conditionnent, signaux appelés sources de reconvergence, et à propager les probabilités vers les signaux à conditionner. Typiquement, les signaux qui conditionnent sont les sources de reconvergence et les signaux à conditionner sont les signaux qui appartiennent au chemin de reconvergence, c'est-à-dire situés sur le chemin entre le source de reconvergence et la porte où la reconvergence a lieu. Mais le conditionnement peut être appliqué à toutes les topologies de circuit.

3.3.1.1 Arbre de décision probabiliste

La représentation d'un circuit en forme d'arbre de décision est une approche très utile pour la compréhension du conditionnement des états. Les arbres de décision sont des représentations graphiques du comportement probabiliste du circuit. Nous proposons une définition plus formelle :

Définition 1 *Un arbre de décision A_C est une représentation du circuit combinatoire C formée par des noeuds et par des arcs. Les noeuds représentent les entrées et sorties primaires du circuit et les portes logiques de C. Les arcs représentent les états possibles de chaque noeud de A_C. Chaque branche de l'arbre représente un état du système, de manière que tous les états sont représentés..*

Ainsi, les noeuds représentant des portes ont deux arcs sortants (fautive ou non), et les noeuds représentant des signaux ont 4 arcs sortants : les quatre états possibles du signal. Chaque arc a une probabilité associée, correspondant à la probabilité d'occurrence de l'état. Les probabilités conditionnées correspondent aux probabilités d'atteindre un certain état d'un noeud en partant de l'état d'un autre noeud.

Par exemple, considérons la porte XOR de la figure 3.3, caracterisée par une probabilité d'erreur $p_{err} = p$, et avec les signaux A et B en entrée, considérés indépendants.

Connaissant les matrices SPR des signaux A et B, qui contiennent les probabilités de leurs états, il est possible de bâtir l'arbre de décision qui définit le comportement de cette porte, (c.f. figure 3.4).

Les noeuds du type A et B correspondent aux signaux d'entrée de la porte, ils ont donc 4 arcs sortants correspondant aux 4 états $0_c, 0_i, 1_i, 1_c$. Les noeuds du type G ont chacun

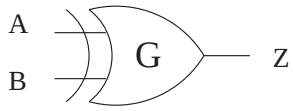


FIG. 3.3 – Porte logique XOR à deux entrées.

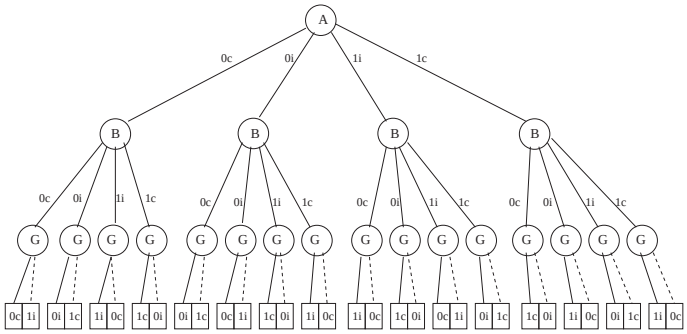


FIG. 3.4 – Arbre de décision d'une porte logique XOR à deux entrées.

deux arcs sortants, qui correspondent à un fonctionnement correct de la porte ou défaillant (en pointillé).

A partir de l'arbre de décision d'un circuit nous pouvons calculer les probabilités d'occurrence de ses états. En faisant l'hypothèse que les entrées du circuit sont statistiquement indépendantes, nous faisons la proposition suivante pour le calcul de ces probabilités :

Proposition 1 *Etant donné un circuit C et son arbre de décision correspondant A_C, la probabilité d'occurrence d'un état d'un circuit C est donnée par le produit des probabilités associées aux arcs de la branche de A_C représentant cet état.*

En effet, lorsque les états des éléments du circuit sont indépendants, la probabilité de chaque état du circuit est donnée par les produits des probabilités d'état de chacun des éléments.

Ainsi, pour l'exemple de la figure 3.3 l'état correspondant à la première branche de l'arbre (tout à gauche sur la figure), $(A = 0_c, B = 0_c, \overline{\varepsilon}, Z = 0_c)$, où $\overline{\varepsilon}$ représente un fonctionnement correct de la porte G et ε le contraire, a une probabilité $p_{branche_1} = p(A = 0_c) \cdot p(B = 0_c) \cdot (1 - p)$.

Il est important de souligner que la probabilité d'une branche ne donne pas la probabilité d'occurrence de la sortie correspondant à cet état. Par exemple, il est possible de représenter de manière graphique l'espace des probabilités du comportement de la porte G. Sur la figure 3.5 les carrés plus grands (trait plus épais) représentent les 4 états de A, chacun d'eux est divisé en 4 autres carrés (trait plus fin) représentant les 4 états de B, et chacun des petits carrés est divisé en deux (trait pointillé) représentant une opération défectueuse ou correcte de la porte.

Chacune des branches de l'arbre de décision est représentée par les rectangles b_i, l'intersection entre branches est donc nulle. Ainsi, si l'on considère l'occurrence d'un état de $Z = z_k$ comme l'union de toutes les branches qui mènent à cet état, sa probabilité d'occurrence est la somme des probabilités de toutes les branches ayant pour résultat $Z = z_k$. Pour rappel, la probabilité de l'union de deux événements aléatoires E et D

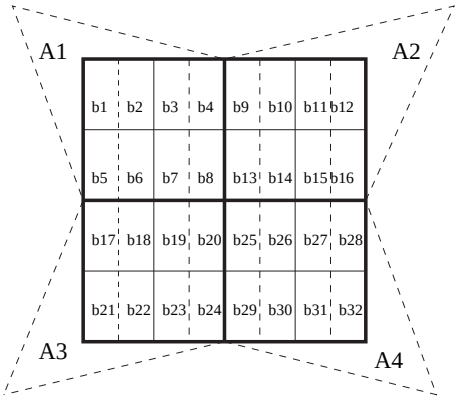


Fig. 3.5 – Représentation graphique de l'espace de probabilité de la porte XOR.

est $p(E \cup D) = p(E) + p(D) - p(E \cap D)$, si $p(E \cap D) = 0$ (comme dans notre cas) $p(E \cup D) = p(E) + p(D)$. Donc, la probabilité $p(Z = z_k)$ est :

$$p(Z = z_k) = \sum_{\forall b_i | Z = z_k} p(b_i) \tag{3.8}$$

Les probabilités conditionnées peuvent être calculées en utilisant la représentation en forme d'arbre de décision. Considérons que l'on souhaite obtenir les probabilités de Z conditionnées par A : $p(Z = z_k | A = a_l) \forall k, l$, ces probabilités sont calculées en considérant seulement les cas qui partent des arcs représentant l'état $A = a_l$. Le calcul est réalisé de manière identique à celle présentée ci-dessus, sauf que la probabilité correspondant à l'arc $A = a_l$ n'est pas prise en compte. La figure 3.6 représente l'interprétation graphique de ce calcul.

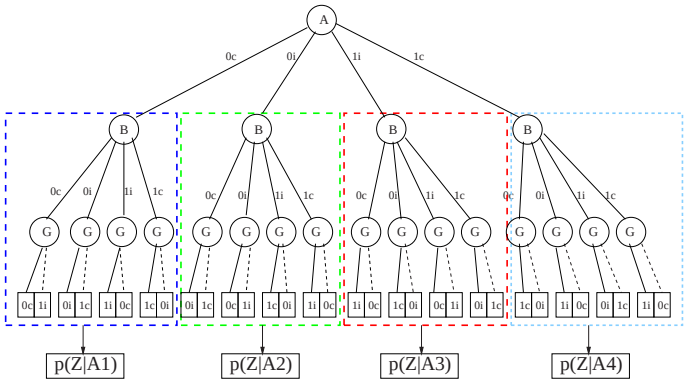


Fig. 3.6 – Calcul de probabilités conditionnées à partir d'un arbre de décision.

Cette manière de représenter le calcul de conditionnement est aussi valable si l'on souhaite conditionner le calcul par d'autres variables. Dans ce cas, les branches à prendre en compte pour le calcul sont celles qui partent pour chacun des cas de la combinaison des variables conditionnantes.

Le calcul analytique des probabilités conditionnées à partir des arbres de décision correspond aux mêmes résultats qu'en utilisant la méthode SPRMP. En effet, la méthodologie

proposée par SPRMP consiste à figer l'état du signal source de reconvergence et à propager cet état jusqu'aux signaux de sortie. Pour propager un seul état d'un signal, la matrice SPR de ce signal vaut 1 dans l'élément correspondant à cet état, et 0 pour les autres, ce qui équivaut à supprimer l'aspect aléatoire du signal. En probabilité, conditionner par un événement signifie que l'occurrence de cet événement est sûre, il n'est plus aléatoire ou encore, sa probabilité vaut 1, exactement comme avec la méthode SPRMP.

La technique que nous proposons se base sur l'utilisation de matrices contenant les probabilités conditionnées des signaux. Nous définissons les matrices de probabilité comme suit :

Définition 2 *La matrice de probabilité conditionnée (CPM) d'un signal Z par un signal A est la matrice contenant toutes les probabilités des états de Z conditionnés par tous les états de A. De manière à ce que l'élément (i, j) contient la probabilité $p(z_i/a_j)$*

Ainsi, chaque colonne de la matrice contient les 4 états du signal Z conditionnés par un des états de A, et chaque ligne un état de Z conditionné par tous les états de A. Cette matrice est de la forme suivante :

$$CPM_Z = \begin{pmatrix} p(z_1/a_1) & p(z_1/a_2) & p(z_1/a_3) & p(z_1/a_4) \\ p(z_2/a_1) & p(z_2/a_2) & p(z_2/a_3) & p(z_2/a_4) \\ p(z_3/a_1) & p(z_3/a_2) & p(z_3/a_3) & p(z_3/a_4) \\ p(z_4/a_1) & p(z_4/a_2) & p(z_4/a_3) & p(z_4/a_4) \end{pmatrix}$$

D'une manière plus générale, nous définissons une matrice CPM d'un signal Y quelconque conditionné par un ensemble de N signaux $S = (S^1, S^2, \ldots, S^N)$. La matrice CPM_Y, de taille 4×4^N, est :

$$CPM_Y = \begin{pmatrix} p(y_1/s_1^1 \cap \ldots \cap s_1^N) & \ldots & p(y_1/s_4^1 \cap \ldots \cap s_4^N) \\ p(y_2/s_1^1 \cap \ldots \cap s_1^N) & \ldots & p(y_1/s_4^1 \cap \ldots \cap s_4^N) \\ p(y_3/s_1^1 \cap \ldots \cap s_1^N) & \ldots & p(y_1/s_4^1 \cap \ldots \cap s_4^N) \\ p(y_4/s_1^1 \cap \ldots \cap s_1^N) & \ldots & p(y_1/s_4^1 \cap \ldots \cap s_4^N) \end{pmatrix}$$

La formulation par matrice est intéressante car elle permet une mise en oeuvre efficace. En effet, les conditionnements peuvent être stockés et éviter ainsi les propagations multiples réalisées par la méthode SPRMP.

3.3.1.2 Probabilités conditionnées dans un chemin logique

Nous présentons une méthode pour calculer les probabilités conditionnées sur un chemin de portes logiques. Pour ce faire, considérons le chemin logique représenté sur la figure 3.7.

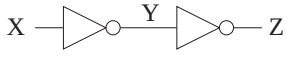


FIG. 3.7 – Exemple de chemin logique pour le calcul de probabilités conditionnées.

Considérons que nous voulions calculer les probabilités de la sortie de la dernière porte, signal Z, en fonction de l'entrée X. L'objectif est donc de calculer toutes les probabilités

$p(z_i/x_k)$, c'est-à-dire, la matrice $CPM_{Z/X}$ définie par :

$$CPM_{Z|X} = \begin{pmatrix} p(z_1/x_1) & p(z_1/x_2) & p(z_1/x_3) & p(z_1/x_4) \\ p(z_2/x_1) & p(z_2/x_2) & p(z_2/x_3) & p(z_2/x_4) \\ p(z_3/x_1) & p(z_3/x_2) & p(z_3/x_3) & p(z_3/x_4) \\ p(z_4/x_1) & p(z_4/x_2) & p(z_4/x_3) & p(z_4/x_4) \end{pmatrix}$$

Nous avons montré comment trouver les probabilités conditionnées pour une porte logique. Ainsi, à partir des probabilités conditionnées des sorties de portes du chemin logique , $p(y_k|x_j)$ et $p(z_i|y_k)$ $\forall i, j, k$, les probabilités conditionnées du signal de sortie du chemin par le signal d'entrée sont données par l'expression 3.9 :

$$p(z_i/x_j) = \sum_{k=1}^{4} p(z_i/y_k) \cdot p(y_k/x_j) \tag{3.9}$$

Les probabilités $p(z_i/y_k)$ et $p(y_k/x_j)$ se trouvent respectivement dans les matrices CPM contenant les probabilités des entrées conditionnées par les sorties de chacune des portes de la figure 3.7. Ces deux matrices sont calculées en appliquant la méthode SPRMP localement sur chacune des portes. L'équation 3.9 correspond aux produits d'une ligne de la matrice CPM_1, contenant $p(z_i|y_k)$, par une colonne de la matrice CPM_2, contenant $p(y_k|x_j)$. En effet :

$$CPM_{Z|X} = \begin{pmatrix} p(z_1/x_1) & p(z_1/x_2) & p(z_1/x_3) & p(z_1/x_4) \\ p(z_2/x_1) & p(z_2/x_2) & p(z_2/x_3) & p(z_2/x_4) \\ p(z_3/x_1) & p(z_3/x_2) & p(z_3/x_3) & p(z_3/x_4) \\ p(z_4/x_1) & p(z_4/x_2) & p(z_4/x_3) & p(z_4/x_4) \end{pmatrix} =$$

$$\begin{pmatrix} p(z_1/y_1) & p(y_1/y_2) & p(z_1/y_3) & p(z_1/y_4) \\ p(z_2/y_1) & p(z_2/y_2) & p(z_2/y_3) & p(z_2/y_4) \\ p(z_3/y_1) & p(z_3/y_2) & p(z_3/y_3) & p(z_3/y_4) \\ p(z_4/y_1) & p(z_4/y_2) & p(z_4/y_3) & p(z_4/y_4) \end{pmatrix} \cdot \begin{pmatrix} p(y_1/x_1) & p(y_1/x_2) & p(y_1/x_3) & p(y_1/x_4) \\ p(y_2/x_1) & p(y_2/x_2) & p(y_2/x_3) & p(y_2/x_4) \\ p(y_3/x_1) & p(y_3/x_2) & p(y_3/x_3) & p(y_3/x_4) \\ p(y_4/x_1) & p(y_4/x_2) & p(y_4/x_3) & p(y_4/x_4) \end{pmatrix}$$

Ce calcul peut être généralisé pour un chemin logique quelconque (c.f. figure 3.8). Les signaux qui interviennent dans les conditionnements sont marqués en rouge. La matrice CPM de chaque porte contient les probabilités de sa sortie en fonction des entrées faisant partie du chemin de conditionnement.

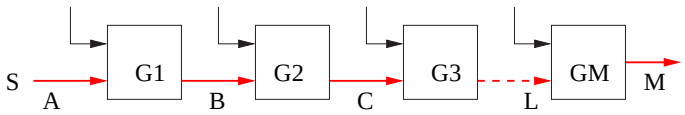


Fig. 3.8 – Exemple de chemin logique pour le calcul de probabilités conditionnées.

Ainsi, pour un telle configuration, nous pouvons trouver la sortie M de ce chemin conditionnée par son entrée S en utilisant l'expression 3.10 :

$$CPM_{M|S} = CPM_M \cdot CPM_{M-1} \ldots CPM_1 \tag{3.10}$$

Cette formulation permet de calculer des probabilités conditionnées dans un chemin logique. Dans la section suivante nous montrons quels sont les signaux par lesquels il faut conditionner, et les procédures pour trouver la fiabilité des signaux de sortie d'un bloc combinatoire en utilisant les matrices CPM.

3.3.2 Calcul de la fiabilité à partir des matrices CPM

Nous avons déjà montré que le calcul de la probabilité jointe d'un ensemble de signaux corrélés peut être obtenue à partir d'une procédure utilisant un conditionnement par les signaux sources de reconvergence. Nous avons aussi montré comment calculer ces probabilités conditionnées dans la logique combinatoire. Nous présentons dans cette section comment cela s'applique au calcul de la fiabilité d'un circuit.

Afin de simplifier la présentation de la méthode, nous considérerons les définitions suivantes :

- *Source de reconvergence :* signal créant des corrélations dans un chemin logique. La sortance des signaux source de reconvergence est supérieure à 1.
- *Puits de reconvergence :* Porte logique ayant deux ou plus signaux corrélés en entrée.
- *Chemin de reconvergence :* Chemin logique (portes logiques) entre une source de reconvergence et son puits de reconvergence.
- *Source de reconvergence primaire :* source de reconvergence qui n'est pas sur le chemin de reconvergence d'une autre source de reconvergence.
- *Source de reconvergence secondaire :* source de reconvergence qui est sur le chemin de reconvergence d'une autre source de reconvergence.
- *Puits de reconvergence final :* puits de reconvergence qui n'est pas sur un chemin de reconvergence plus grand (qui l'englobe).
- *Puits de reconvergence partiel :* puits de reconvergence qui est sur un chemin de reconvergence plus grand.

Pour le graphe de la figure 3.9, où chaque noeud peut représenter une entrée ou une porte logique, et chaque arc représente une connexion, nous avons les éléments suivants :

- Sources de reconvergence primaires : A.
- Sources de reconvergence secondaires : B, E.
- Puits de reconvergence partiel : H, I.
- Puits de reconvergence final : J.

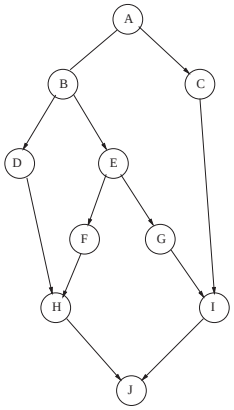


FIG. 3.9 – Diagramme représentant les différents éléments d'un chemin de reconvergence.

Considérons maintenant le circuit de la figure 3.10. L'objectif est de calculer les probabilités $p(x_i \cap y_j)$ afin de déterminer de façon exacte les probabilités des états de Z.

Nous devons calculer la matrice M_{XY} contenant les probabilités jointes exactes des signaux X et Y. Ainsi la fiabilité du signal Z de sortie peut être obtenue à partir de

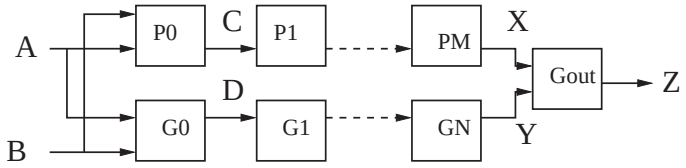


Fig. 3.10 – Signaux corrélés.

l'expression 3.11 :

$$SPR_Z = ITM^T_{G_{out}} \cdot M_{XY} \cdot PTM^T_{G_{out}} \tag{3.11}$$

Chacune des probabilités $p(x_i \cap y_j)$ peut être calculée avec cette formule

$$p(x_i \cap y_j) = \sum_{k=1}^{4}\sum_{l=1}^{4} p(x_i/a_k \cap b_l) \cdot p(y_j/a_k \cap b_l) \cdot p(a_k) \cdot p(b_j) \tag{3.12}$$

Les termes $p(x_i/a_k \cap b_l)$ et $p(y_j/a_k \cap b_l)$ se trouvent dans les matrices $CPM_{X/A\cap B}$ et $CPM_{Y/A\cap B}$, qui sont obtenues à partir du produit des matrices CPM des portes $G0, \ldots, GM$ et $P0, \ldots, PM$ respectivement. Ces deux produits sont :

$$CPM_{X/A\cap B} = \prod_{i=0}^{M} CPM_{Pi} \tag{3.13}$$

$$CPM_{Y/A\cap B} = \prod_{i=0}^{N} CPM_{Gj} \tag{3.14}$$

Les termes $p(a_k)$ et $p(b_j)$ de l'équation 3.12 se trouvent dans les matrices SPR caractérisant les signaux A et B.

Les structures des chemins de reconvergences des circuits réels sont plus complexes. Dans l'exemple précédent, le circuit consiste seulement en deux chemins de reconvergences créés par deux sources indépendantes. Dans le cas général, il y a des chemins de reconvergences imbriqués, des chemins en parallèle, des reconvergences qui ont lieu à l'intérieur d'autres chemins de reconvergences plus grands, etc. Par la suite, nous montrons comment traiter ces cas avec l'approche CPM.

Une propriété importante de l'approche CPM pour traiter toutes ces configurations est qu'elle applique un conditionnement seulement par les sources les plus proches des chemins de reconvergence. C'est-à-dire, avec les matrices CPM, le conditionnement réalisé est stocké dès qu'une source de reconvergence secondaire est trouvée lors du parcours du circuit, et un nouveau conditionnement seulement par la source secondaire est introduit à partir de ce point. Par exemple, si l'on considère le diagramme de la figure 3.11, où le signal $S1$ est une source primaire, et le signal $S2$ est une source secondaire. Ainsi, pour ce circuit les conditionnements à réaliser sont les suivants :

- Signal de sortie de A conditionné par $S1$
- Signal de sortie de B conditionné par $S2$
- Pour $S2$:
  - Stockage d'une matrice contenant $S2$ conditionné par $S1$
  - Signaux de sortie de $S2$ conditionnés par lui-même (matrice CPM identité)

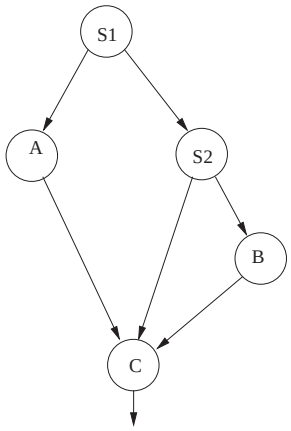


FIG. 3.11 – Exemple de topologie classique contenant une source de reconvergence primaire (S1) et une secondaire (S2).

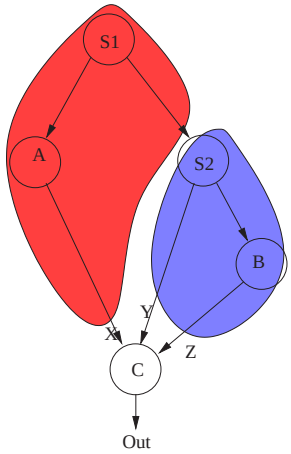


FIG. 3.12 – Exemple d'application des différents conditionnements dans une structure de chemins de reconvergence imbriqués.

Sur la figure 3.12 se trouve un schéma des conditionnements réalisés pour ce circuit.

Pour le calcul des probabilités jointes nous introduisons le concept de poids du conditionnement :

Définition 3 *Le poids d'un conditionnement est la probabilité d'occurrence de l'état des sources de reconvergence par lequel le conditionnement est fait.*

Ainsi, pour calculer la matrice M_{XYZ} contenant les probabilités $p(x_i \cap y_j \cap z_k)$ il suffit de réaliser le calcul :

$$p(x_i \cap y_j \cap z_k) = \sum_{l=1}^{4} \sum_{n=1}^{4} p(x_i/S1_l) \cdot p(y_j/S2_n) \cdot p(z_k/S2_n) \cdot p(S1_l \cap S2_n) \tag{3.15}$$

Le terme $p(S1_l \cap S2_n)$ correspond au calcul du poids de conditionnement et s'exprime à partir de la formule suivante :

$$p(S1_l \cap S2_n) = p(S2_n/S1_l) \cdot p(S1_l) \tag{3.16}$$

La fiabilité du signal Out peut être obtenue en appliquant :

$$SPR_{out} = ITM_C^T \cdot M_{XYZ} \cdot PTM_C \tag{3.17}$$

Dans la méthode CPM, les conditionnements s'appliquent d'une manière hiérarchique. Ainsi, même si un signal dépend d'une source de reconvergence, il se peut que ce signal ne soit pas conditionné d'une manière directe par cette source mais implicitement. Ainsi, nous proposons la définition suivante :

Définition 4 *Masquage de source : pour un signal S sur les chemins de reconvergence de plusieurs sources, la topologie de ces chemins provoquent un conditionnement direct du signal S par seulement une partie des sources de reconvergences, le conditionnement par le reste de sources étant implicite.*

Dans l'exemple précédent (figure 3.12) les signaux Y et Z dépendent de la source $S1$, mais nous calculons seulement un conditionnement direct par la source $S2$. Ce phénomène mérite d'être remarqué car il modifie la manière de calculer les poids montrée dans les équations (3.12) et (3.16). En effet, il peut arriver que les signaux d'entrée d'une porte qui est un puits de reconvergence ne soient pas conditionnés directement par toutes les sources de corrélation. Considérons le circuit de la figure 3.13.

Dans ce circuit, les signaux E et F sont les entrées du puits de reconvergence, porte G. Ces deux signaux sont conditionnés seulement par les sources C et D, qui sont eux-mêmes conditionnés par A et B. Avec la méthode CPM, les matrices à calculer sont les suivantes :

- CPM_C, contenant les éléments $p(c_i/a_j \cap b_k)$
- CPM_D, contenant les éléments $p(d_i/a_j \cap b_k)$
- CPM_E, contenant les éléments $p(e_i/c_j \cap d_k)$
- CPM_F, contenant les éléments $p(f_i/c_j \cap d_k)$

A partir des matrices SPR des signaux A et B il faut calculer la matrice M_{EF} contenant les éléments $p(e_l \cap f_n)$ définis par l'équation 3.18, où la probabilité $p(c_j \cap d_k)$ est trouvée à partir de l'équation 3.19.

$$p(e_l \cap f_n) = \sum_{j=1}^{4} \sum_{k=1}^{4} p(e_l/c_j \cap d_k) \cdot p(f_n/c_j \cap d_k) \cdot p(c_j \cap d_k) \tag{3.18}$$

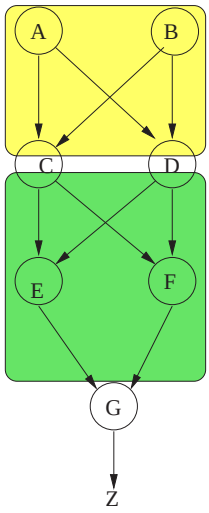


FIG. 3.13 – Exemple du phénomène de masquage de conditionnement de source.

$$p(c_j \cap d_k) = \sum_{r=1}^{4} \sum_{s=1}^{4} p(c_j/a_r \cap b_s) \cdot p(d_k/a_r \cap b_s) \cdot p(a_r) \cdot p(b_s) \tag{3.19}$$

La fiabilité de la sortie est :

$$SPR_Z = ITM_G^T \cdot M_{EF} \cdot PTM_G \tag{3.20}$$

Nous voulons généraliser le calcul de poids et trouver une formule qui définisse les calculs à réaliser pour tous les cas possibles. Considérons S l'ensemble des N signaux qui sont à l'entrée d'un puits de reconvergence, formé par les signaux $S^a, S^b, \ldots, S^n$. Considérons R l'ensemble de M sources de reconvergence qui conditionnent directement l'ensemble S, formé par les signaux $R^a, R^b, \ldots, R^m$. Finalement, T est l'ensemble de O sources de reconvergence cachées, formé par $T^a, T^b, \ldots, T^o$. Le terme $p(s_i^a \cap \ldots s_j^n)$, où s_α^l est le signal S^l dans l'état α ($\alpha = 1 \ldots 4$) quelconque, est obtenu à partir de la formule suivante :

$$\begin{aligned} p(s_{\alpha_1}^a \cap \ldots \cap s_{\alpha_n}^n) \quad = \quad & \left(\sum p(s_{\alpha_1}^a / r_{\beta_k}^a \cap \ldots \cap r_{\beta_m}^m) \cdot \ldots \cdot p(s_{\alpha_n}^n / r_{\beta_a}^a \cap \ldots \cap r_{\beta_m}^m) \right) \\ & \cdot \left(\sum p(r_{\beta_a} / t_{\phi_a}^a \cap \ldots \cap t_{\phi_o}^o) \prod_{k=a}^{o} p(t_{\phi_k}^k) \right) \end{aligned} \tag{3.21}$$

Cette équation est divisée en deux parties :

- La première, constituée des signaux atteignant directement le puits (conditionnés par les sources correspondantes).
- La deuxième partie contient l'information correspondant aux sources masquées.

Cette équation peut être interprétée de manière intuitive comme la somme des probabilités de tous les états des sources directes qui mènent à l'état $s_{\alpha_1}^a \cap \ldots \cap s_{\alpha_n}^n$ pondérées par l'addition de tous les états des sources indirectes qui mènent à chacun des états des sources directes considérées dans l'addition.

3.3.3 Réduction de la taille des matrices CPM intermédiaires

La méthode CPM conditionne tous les signaux par les sources de reconvergences les plus proches de ces signaux et stocke ce conditionnement dans une matrice CPM. La taille de ces matrices CPM est donnée par 4×4^N, où N est le nombre de sources de reconvergence par lesquelles la matrice CPM correspondante est conditionnée. Par conséquent, le nombre de colonnes croît exponentiellement avec N. Ceci peut représenter une limitation lors de l'application à des circuits complexes, du fait de la capacité de stockage limitée des machines.

Considérons l'exemple de la figure 3.14, où le signal A est conditionné par l'ensemble de N sources S_A, et le signal B est conditionné par l'ensemble de M sources S_B.

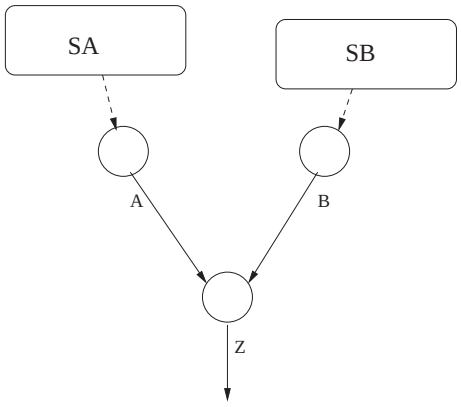


Fig. 3.14 – Exemple d'accumulation de sources de conditionnement.

Le signal Z, selon l'approche CPM, sera donc conditionné par l'ensemble S_A et S_B, ce qui a pour résultat une matrice de taille $4 \times 4^{N+M}$. Il est possible que la machine sur laquelle s'exécute l'estimation de la fiabilité n'ait pas la capacité de stocker une matrice de cette taille. La solution que nous proposons à ce problème consiste à introduire des signaux de conditionnement intermédiaires même s'ils ne sont pas source de conditionnement. Pour le traitement postérieur du calcul de fiabilité dans la sortie du circuit, ces signaux seront considérés comme étant des sources de reconvergence secondaires. Pour le circuit de la figure 3.14, nous pouvons stocker le conditionnement introduit par S_A (ou S_B éventuellement), et introduire un conditionnement par A. Ainsi, le signal Z, sera conditionné seulement par S_B et A, et la taille de sa matrice sera $4 \times 4^{M+1}$, évitant l'explosion combinatoire et les problèmes qui en découlent.

3.3.4 Estimation de la corrélation des entrées

Une des limitations des modèles probabilistes d'estimation de fiabilité consiste à prendre en compte l'effet des corrélations introduites directement par les entrées primaires du circuit. La solution que nous proposons consiste à considérer les entrées comme des sources de reconvergence. Pour ce faire, il est nécessaire de disposer des valeurs de probabilités jointes des signaux d'entrée, et non pas seulement de la matrice SPR qui les caractérise individuellement.

L'introduction des entrées primaires comme sources de corrélation entraine une augmentation du coût de calcul de la méthode CPM. Il est donc intéressant de pouvoir identifier

les entrées qui ne sont pas corrélées entre elles afin de simplifier le calcul.

Considérons que nous disposons des probabilités d'entrée, $p(\vec{I})$ d'un circuit G quelconque, comme représente figure 3.15, et que nous diposons aussi des probabilités individuelles des signaux d'entrée $p(I_j)$, pour $j = 1 \dots N$.



FIG. 3.15 – Circuit générique pour la détection d'entrées incorrélées.

Nous voulons détecter la présence d'un signal décorrélé du reste des entrées. Considérons que nous voulions évaluer si un signal I_j est décorrélé de l'ensemble $\vec{I}$, et que nous disposons des probabilités suivantes :

- $p(\vec{I}) = p(I_{1_\alpha} \cap \dots \cap p(I_{j_\beta}) \dots \cap I_{N_\gamma})$
- $p(I_{j_\beta})$ pour $\beta = 1, 2, 3, 4$.

Si le signal I_j est indépendant du reste des signaux, et en considérant I' comme l'ensemble des signaux moins I_j alors nous pouvons affirmer :

$$p(\vec{I}) = p(\vec{I'}) \cdot p(I_j) \tag{3.22}$$

C'est-à-dire, la probabilité d'un état de I est la probabilité de l'état de I' multipliée par la probabilité de I_j. Considérons maintenant que les signaux de I' sont dans un état figé. Considérons finalement les définitions suivantes : $I_1 = I' \cup I_{j_1}$, $I_2 = I' \cup I_{j_2}$, $I_3 = I' \cup I_{j_3}$ et $I_2 = I' \cup I_{j_4}$. Chacun de ces termes est l'union des états de I' figés avec chacun des états possibles de I_j. A partir de l'équation 3.22, nous pouvons définir les équations suivantes :

$$p(\vec{I'_1}) = \frac{p(I_1)}{p(I_{j_1})} = k_1 \tag{3.23}$$

$$p(\vec{I'_2}) = \frac{p(I_2)}{p(I_{j_2})} = k_2 \tag{3.24}$$

$$p(\vec{I'_3}) = \frac{p(I_3)}{p(I_{j_3})} = k_3 \tag{3.25}$$

$$p(\vec{I'_4}) = \frac{p(I_4)}{p(I_{j_4})} = k_4 \tag{3.26}$$

Si effectivement I_j est indépendant du reste des signaux, les valeurs de k_l seront les mêmes. Ainsi, nous pouvons appliquer ce calcul à tous les signaux d'entrée pour identifier ceux qui sont indépendants.

3.3.5 Méthode CPM approximée

Il peut être possible d'accepter une perte de précision du modèle si nous voulons réduire la complexité de l'estimation de la fiabilité. En effet, le modèle CPM permet de proposer une heuristique qui estime partiellement la corrélation des signaux. En général, les approches de l'état de l'art qui ne sont pas exactes soit ignorent l'effet des signaux reconvergents soit les

prennent en compte partiellement mais en utilisant des critères très rigides, sans apporter de solutions ni de méthodologies sur la pertinence des sources de reconvergence négligées. Notamment, les auteurs de l'approche SPRMP, [85] et [86], montrent que la précision de l'estimation dépend du nombre de signaux reconvergents pris en compte pour l'analyse mais ils ne proposent pas de critères permettant de choisir quelles sources prendre en compte. La technique PGM modulaire, [76], prend en compte les corrélations à l'intérieur des modules, ignorant les corrélations entre les modules. Nous présentons maintenant une approche basée sur l'information contenue dans les matrices CPM pour mesurer la corrélation entre deux signaux.

3.3.5.1 Estimation de la corrélation entre deux signaux

Lorsque deux signaux S et T sont indépendants, nous pouvons affirmer que pour tous les états des deux signaux le conditionnement de l'un par l'autre n'a pas d'effet sur la valeur de la probabilité de l'état. Cela peut s'exprimer par :

$$p(s_i/t_j) = p(s_i), \forall i, j \tag{3.27}$$

$$p(t_j/s_i) = p(t_j), \forall i, j \tag{3.28}$$

De ces équations nous pouvons déduire :

$$\frac{p(s_i/t_j)}{p(s_i)} = 1 \tag{3.29}$$

Ainsi, pour deux signaux X et Y quelconques, nous pouvons interpréter le rapport $\frac{p(x_k/y_l)}{p(x_k)}$, ou $\frac{p(y_l/x_k)}{p(y_l)}$ comme une mesure de la corrélation entre eux, et pouvons affirmer qu'elle est d'autant plus faible que ce rapport est proche de 1.

Une autre manière de mesurer la corrélation est en relation avec la probabilité jointe de deux signaux. Pour deux signaux quelconques, X et Y, la probabilité jointe de chacun de leurs états s'exprime à partir de l'équation :

$$p(x_k \cap y_l) = p(x_k/y_l) \cdot p(y_l) = p(y_l/x_k) \cdot p(x_k) \tag{3.30}$$

Ce produit devient directement le produit des états lorsque les deux signaux sont indépendants. Par conséquent, nous pouvons interpréter le rapport entre la probabilité conjointe de deux signaux et le produit entre leurs probabilités comme une mesure de sa corrélation, donnée par l'équation 3.31 :

$$c_{XY_{kl}} = \frac{p(x_k \cap y_l)}{p(x_k) \cdot p(y_l)} \tag{3.31}$$

La corrélation entre X et Y sera d'autant plus faible que les différents facteurs $c_{XY_{kl}}$ seront proches de 1.

Pour un chemin logique quelconque, l'état d'un signal S dépend des états des signaux du cône logique qui le précède ainsi que du comportement des portes de ce cône. Sur la figure 3.16 nous avons représenté un exemple de cône logique. De manière intuitive, nous pouvons dire que plus ce cône est grand, plus l'influence d'un des signaux de ce cône sur S est petite, car il y a beaucoup de paramètres qui définissent l'état de S.

La corrélation crée par un signal source de reconvergence sera d'autant plus faible que les signaux reconvergents seront décorrélés du signal source de reconvergence. Par exemple, considérons le circuit de la figure 3.17.

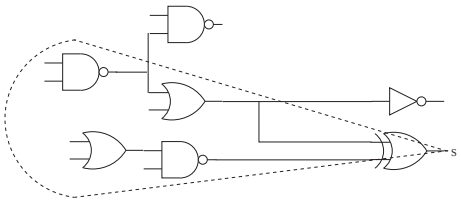


FIG. 3.16 – Exemple de cône logique précédant un signal S.

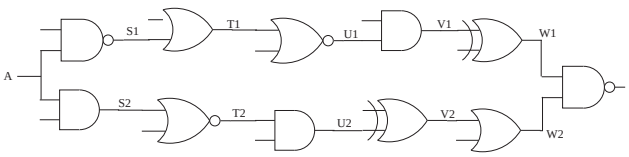


FIG. 3.17 – Chemin de corrélation basique pour la démonstration de l'influence du cône logique sur les corrélations.

Pour ce circuit nous proposons de mesurer la corrélation entre le signal A, et les signaux de sortie des portes logiques du circuit $S1$, $S2$, $T1$, $T2$, $U1$, $U2$, $V1$, $V2$, $W1$ et $W2$, afin de démontrer que cette corrélation diminue au fur et à mesure que la distance entre les deux signaux augmente. Pour ce faire, nous pouvons calculer les matrices CPM des signaux du circuit conditionnés par A, ainsi que les matrices SPR contenant leurs probabilités statiques. Ensuite, nous proposerons une mesure basée sur ces matrices.

Définition 5 *L'estimation de la décorrélation C_{AB} entre deux signaux A et B est une mesure de la dépendance de B par rapport à A. Elle est donnée par :*

$$C_{ab} = \sum_{\forall i \forall j} |1 - \frac{p(a_j/b_i)}{p(b_i)}| \tag{3.32}$$

Cette mesure estime la déviation moyenne des valeurs $\frac{p(a_j/b_i)}{p(b_i)}$ par rapport à 1 (valeur indicant une décorrélation totale). Plus cette mesure est élevée plus les signaux sont corrélés.

Nous avons réalisé ce calcul 100 fois pour chaque signal de sortie des portes du circuit de la figure 3.17, en donnant à chaque signal d'entrée une distribution aléatoire et en moyennant les résultats.

En bleu est représentée la corrélation des signaux des portes du chemin 1 (partie supérieure de la figure 3.17), en vert la corrélation pour le chemin 2 (partie inférieure). Sur l'axe des abscisses, se trouvent les valeurs en fonction de la distance des signaux au signal A ($S1$, $S2$ se trouvent à distance 1, $W1$ et $W2$ à distance 5). Nous voyons que la corrélation avec le signal A diminue avec la distance, et devient très faible après 5 étages de portes.

3.3.5.2 Estimation partielle des reconvergences

Nous proposons maintenant une heuristique qui prend en compte la corrélation seulement là où elle est la plus forte, c'est-à-dire dans les puits de reconvergence les plus proches des sources. Ceci revient à une estimation partielle des corrélations, comportant une perte de précision mais un gain en vitesse d'analyse.

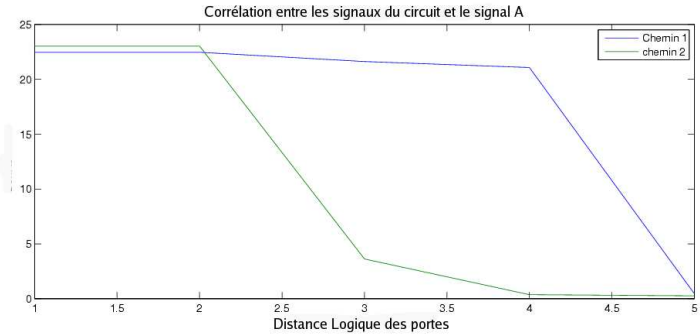


Fig. 3.18 – Corrélation mesurée entre les signaux $S1$, $S2$, $T1$, $T2$, $U1$, $U2$, $V1$, $V2$, $W1$ et $W2$ avec le signal A.

Méthode	R	Erreur (%)	Temps (s)
SPR	0.72	8.55	0.0015
CPM	0.7873	0	0.03
Estimation partielle	0.7865	0.11	0.0034

Tab. 3.2 – Résultats et temps de calcul des approches SPR, CPM et CPM approximé pour le circuit de la figure 3.19.

Nous avons appliqué cette estimation au circuit de la figure 3.19. Pour ce circuit, nous avons trois signaux sources de reconvergences : A, B et C.

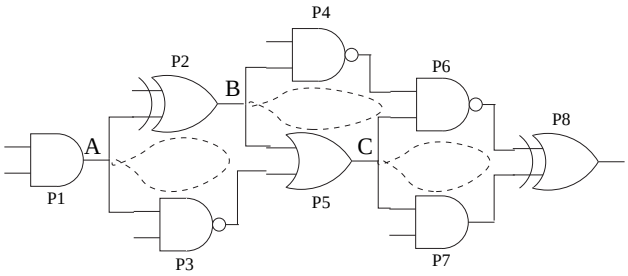


Fig. 3.19 – Circuit exemple d'estimation partielle des reconvergences.

Les corrélations introduites par chacun d'eux sont prises en compte seulement jusqu'au puits le plus proche de chacun : porte $P5$ pour A, $P6$ pour B et $P8$ pour C. Nous avons comparé les résultats obtenus avec ceux issus d'une méthode exacte, CPM, et avec SPR, représentés sur la table 3.2.

La valeur de la fiabilité obtenue avec l'approche d'estimation partielle des corrélations est beaucoup plus proche de la valeur réelle que celle réalisée avec la méthode SPR, et avec un temps d'exécution deux fois plus grand. Le temps d'exécution pour l'approche CPM est 10 fois supérieur au temps requis par l'approche d'estimation partielle.

3.4 Méthode CPM hiérarchique

L'approche CPM est susceptible d'avoir des limitations lorsqu'un signal doit être conditionné par un grand nombre de sources de reconvergence. Cependant, ce problème peut être surmonté grâce à l'utilisation de probabilités conditionnées de manière hierarchique. Nous proposons une approche basée sur un découpage du circuit en blocs permettant de limiter le nombre de conditionnements subis par les signaux du circuit. Ensuite, l'utilisation de probabilités conditionnées permet de combiner les informations obtenues pour chaque bloc du découpage afin de calculer la fiabilité de sortie du circuit.

3.4.1 Calcul de probabilités de sortie d'un bloc combinatoire en fonction des entrées

Nous présentons dans ce paragrahe une approche hiérarchique visant à contenir le problème d'explosion de la taille des matrices CPM. Cette méthode repose sur une partition du circuit en sous-blocs sur lesquels nous allons appliquer la méthode CPM. Nous présentons les modifications nécessaires à la méthode CPM permettant de l'appliquer de façon hiérarchique.

Pour un bloc combinatoire quelconque composé de N entrées et M sorties, comme représenté en figure 3.20, nous pouvons calculer la fiabilité des signaux de sortie en fonction des probabilités des signaux en entrée. Pour l'approche CPM, qui utilise une décomposition des signaux en 4 états, ce conditionnement se traduit par une matrice de taille $4^N \times 4^M$. Cette matrice a une croissance exponentielle, mais la partition du circuit en blocs permet de borner cette croissance.



FIG. 3.20 – Exemple de bloc combinatoire générique.

Afin de calculer les sorties d'un bloc en fonction de ses entrées nous pouvons appliquer l'approche CPM en applicant quelques modifications par rapport à la technique de base. Ainsi, pour adapter CPM au type de calcul que nous proposons il faut réaliser les changements suivants :

- Les signaux conditionnants sont non seulement les sources de reconvergence réelles du bloc, mais aussi les entrées primaires du bloc et les sources de reconvergence internes au bloc.
- Afin de garder le conditionnement par les seules entrées primaires :
  - Il ne faudra pas appliquer de poids pour les entrées, qu'elles soient ou pas source de reconvergence.
  - Il faut appliquer un poids aux sources de reconvergence qui ne sont pas des entrées primaires, afin d'éliminer le conditionnement qu'elles introduisent et garder seulement celui des entrées primaires sur les sorties.

D'une manière intuitive, nous pouvons considérer les entrées du bloc comme des sources de reconvergence primaires, et les sources de reconvergence internes au bloc comme des sources de reconvergence secondaires. Le poids à appliquer à ces sources secondaires sera

fonction des états des entrées. Il est plus simple de comprendre cette procédure à partir d'un exemple. Considérons le bloc combinatoire de la figure 3.21. L'objectif est de calculer la matrice contenant les probabilités $p(x_i \cap y_j / a_k \cap b_l)$ prenant en compte la corrélation introduite par le signal interne C. En appliquant la méthode CPM normalement mais sans

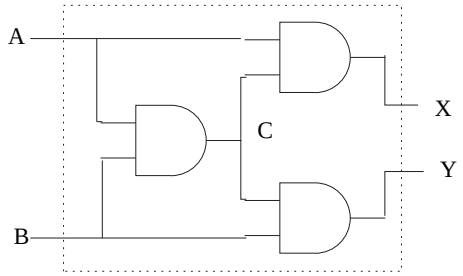


Fig. 3.21 – Exemple de bloc combinatoire pour illustrer les modifications par rapport à CPM primaire.

réaliser le calcul de poids, nous trouverions les probabilités suivantes :

- Signal C : $p(c_m / a_k \cap b_l)$
- Signal X : $p(x_i / a_k \cap c_m)$
- Signal Y : $p(y_j / b_l \cap c_m)$

A partir de ces valeurs, nous pouvons trouver les probabilités $p(x_i \cap y_j / a_k \cap b_l)$ en appliquant le poids correspondant à C :

$$p(x_i \cap y_j / a_k \cap b_l) = \sum_{m=1}^{4} p(x_i / a_k \cap c_m) \cdot p(y_j / c_m \cap b_l) \cdot p(c_m / a_k \cap b_l) \tag{3.33}$$

La matrice contenant ces probabilités est donc :

$$CPM_{XY/AB} = \begin{pmatrix} p(x_1 \cap y_1 / a_1 \cap b_1) & p(x_1 \cap y_1 / a_1 \cap b_2) & \dots & p(x_1 \cap y_1 / a_4 \cap b_4) \\ p(x_1 \cap y_2 / a_1 \cap b_1) & p(x_1 \cap y_2 / a_1 \cap b_2) & \dots & p(x_1 \cap y_2 / a_4 \cap b_4) \\ \vdots & \dots & \dots & \vdots \\ p(x_4 \cap y_4 / a_1 \cap b_1) & p(x_4 \cap y_4 / a_1 \cap b_2) & \dots & p(x_4 \cap y_4 / a_4 \cap b_4) \end{pmatrix} \tag{3.34}$$

A partir des éléments de l'équation 3.34 nous pouvons calculer les probabilités jointes de A et B en utilisant l'équation suivante :

$$p(x_i \cap y_j) = \sum_{k=1}^{4} \sum_{l=1}^{4} p(x_i \cap y_j / a_k \cap b_l) \cdot p(a_k) \cdot p(b_l) \tag{3.35}$$

La fiabilité de ce bloc combinatoire correspond à la somme des probabilités des états représentant des états corrects des signaux X et Y, c'est-à-dire :

$$R_{XY} = p(x_1 \cap y_1) + p(x_1 \cap y_4) + p(x_4 \cap y_1) + p(x_4 \cap y_4) \tag{3.36}$$

Nous avons montré comment calculer les probabilités des états de sortie d'un bloc combinatoire en fonction des états de ses entrées. Dans les sections suivantes, nous montrons comment ce type de calcul est appliqué à l'estimation de la fiabilité de circuits combinatoires.

3.4.2 Calcul de la fiabilité à partir de la décomposition modulaire disjointe d'un circuit

Considérons un circuit et la description de chacun de ces blocs, c'est-à-dire, les sorties en fonction des entrées pour chacun d'entre eux. Il est possible de calculer les probabilités de sortie du circuit en effectuant une propagation des probabilités des entrées vers les sorties en suivant un ordre topologique entre les blocs.

Nous définissons deux blocs comme disjoints s'ils ne partagent aucun signal d'entrée. En effet, une décomposition du circuit en blocs disjoints permet une propagation séquentielle des probabilités entre les blocs à cette seule condition. Les probabilités de sortie d'un bloc font partie des probabilités d'entrée d'un bloc postérieur. Si les blocs sont effectivement disjoints, et seulement dans ce cas, la propagation est possible, même dans le cas de signaux reconvergents, car leur corrélation est prise en compte soit en interne dans le traitement du bloc, soit dans les signaux de sortie de la matrice CPM qui le caractérise, (c.f. section 3.4.1).

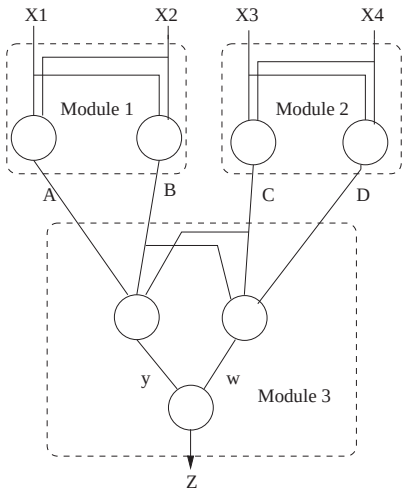


FIG. 3.22 – Exemple de modularisation d'un circuit.

Considérons maintenant le circuit de la figure 3.22, où :
- $X1$, $X2$, $X3$ et $X4$ sont les entrées primaires.
- Z est le signal de sortie.
- Les noeuds de la figure représentent des portes logiques quelconques.
- les cadres en pointillé représentent la modularisation proposée pour ce circuit.

Nous pouvons trouver les probabilités du signal Z en fonction des entrées du module 3, signaux A, B, C et D, obtenant donc les probabilités $p(z_i/a_j \cap b_k \cap c_l \cap d_m)$. De la même manière, nous pouvons trouver les probabilités $p(a_j \cap b_k)$ et $p(c_l \cap d_m)$ des sortie des deux modules 1 et 2. Sachant que les signaux A et B sont indépendants de C et D nous pouvons affirmer que $p(a_j \cap b_k \cap c_l \cap d_m) = p(a_j \cap b_k) \cdot p(c_l \cap d_m)$. Ainsi, nous pouvons trouver les probabilités des états de Z avec l'équation suivante :

$$p(z_i) = \sum_{\forall j \forall k} \sum_{\forall l \forall m} p(z_i/a_j \cap b_k \cap c_l \cap d_m) \cdot p(a_j \cap b_k) \cdot p(c_l \cap d_m) \tag{3.37}$$

L'obtention de la matrice contenant les probabilités $p(a_j \cap b_k)$ requiert 16 itérations de calcul, du fait que $X1$ et $X2$ sont des sources de corrélations, elles imposent l'application

d'un conditionnement par les $4^2 = 16$ états (donc 16 itérations de calcul). Il en est de même pour les probabilités $p(c_l \cap d_m)$. Pour calculer les probabilités $p(z_i/a_j \cap b_k \cap c_l \cap d_m)$ le coût est de $4^4 = 256$ itérations. Pour la méthode CPM primaire, le coût d'analyse de ce même circuit est supérieur, car les signaux d'entrée de la porte finale du circuit, y et w, sont conditionnés respectivement par $X1$, $X2$, B et C, et $X3$, $X4$, B et C, ce qui a un coût de $4^4 = 256$ itérations chacun. Ainsi, pour trouver les probabilités jointes d'entrée de la porte finale, le calcul à réaliser implique un total de $4^6 = 4096$ itérations car les signaux d'entrée sont conditionnés par un ensemble de 6 sources de reconvergence ($X1$, $X2$,$X3$, $X4$, B et C).

Afin de généraliser la méthode, le processus à réaliser peut se décrire en deux étapes :

- Traitement des blocs séparément, dont :
  - Blocs connectés aux entrées primaires : calcul des probabilités statiques des sorties.
  - Blocs non connectés aux entrées primaires : calcul des probabilitées de sortie conditionnées par les entrées.
- Propagation des probabilités suivant un ordre topologique.

3.4.3 Calcul de la fiabilité à partir de la décomposition modulaire non-disjointe d'un circuit

Dans l'approche hiérarchique, la condition que les blocs soient disjoints est restrictive et ne peut être maintenue dans tous les cas. En effet, considérons l'exemple de la figure 3.23 formé par deux blocs combinatoires, A et B, où SA est un ensemble d'entrées pour le bloc A, SB est un ensemble d'entrées pour le bloc B, et SC est un ensemble d'entrées partagées par le bloc A et B.

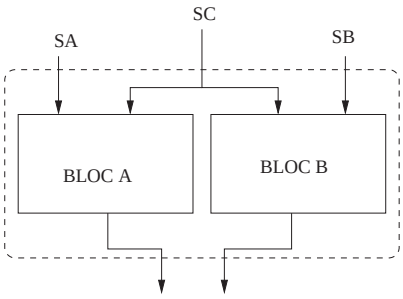


Fig. 3.23 – Exemple de modularisation non-disjointe d'un circuit.

Afin de respecter la condition que les blocs soient disjoints pour ce cas nous devrions considérer un module contenant les deux blocs. Or, la matrice représentant un bloc a une taille maximale bornée (due aux limitations de stockage des machines), car sa taille est $4^N \times 4^M$ pour un bloc avec N entrées et M sorties, donc une croissance exponentielle. Dans le cas où il n'est pas possible de réaliser un découpage en blocs disjoints, il faut réaliser ce découpage d'une manière à minimiser le nombre de signaux partagés entre les blocs. Dans le processus de propagation de probabilités il faut garder le conditionnement seulement par les signaux partagés entre blocs, de manière à ce que le reste des sources de reconvergence soit traité en interne dans les modules. Le traitement individuel des blocs ne change pas, seul le processus de propagation est affecté.

Considérons l'exemple de la figure 3.24, consistant en une modification de l'exemple de la figure 3.22, avec l'introduction d'une nouvelle entrée, le signal $X2$, qui crée une

entrée partagée par deux modules. Pour des raisons de simplicité, nous considérons que les modules 1 et 2 ne peuvent pas être unifiés.

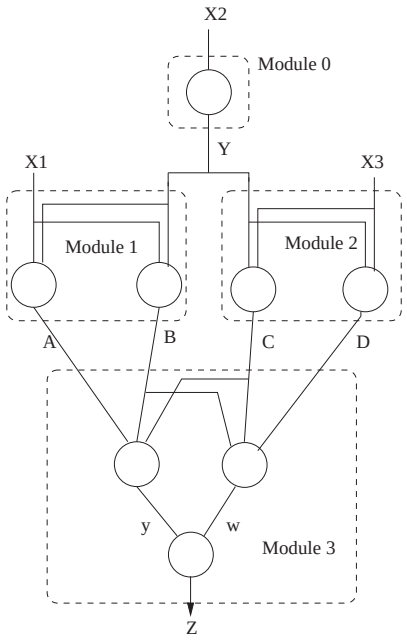


FIG. 3.24 – Exemple de modularisation non-disjointe d'un circuit.

Pour cet exemple, les probabilités à calculer pour chaque bloc sont les suivantes :
- Module 0 : $p(y_i)$.
- Module 1 : $p(a_j \cap b_k / x1_o \cap y_i)$
- Module 2 : $p(c_l \cap d_m / x2_r \cap y_i)$
- Module 3 : $p(z_n / a_j \cap b_k \cap c_l \cap d_m)$

Ainsi, la propagation de probabilités des entrées vers les sorties se réalise en deux étapes.
- D'abord les conditionnements des blocs par les signaux non partagés sont supprimés, pour cet exemple, les signaux $X1$ et $X3$ des modules 1 et 2 respectivement.
- Ensuite, la propagation du conditionnement par les signaux partagés jusqu'aux entrées du bloc contenant la sortie ($X2$ dans notre cas).

Cela conduit aux équations suivantes :

$$p(a_j \cap b_k / y_i) = \sum_{o=1}^{4} p(a_j \cap b_k / x1_o \cap y_i) \cdot p(x1_o) \tag{3.38}$$

$$p(c_l \cap d_m / y_i) = \sum_{r=1}^{4} p(c_l \cap b_m / x2_r \cap y_i) \cdot p(x2_r) \tag{3.39}$$

$$p(a_j \cap b_k \cap c_l \cap b_m) = \sum_{i=1}^{4} p(a_j \cap b_k / y_i) \cdot p(c_l \cap d_m / y_i) \cdot p(y_i) \tag{3.40}$$

3.4.4 Estimation de la complexité

Afin d'estimer la complexité de l'approche CPM modulaire, considérons les définitions suivantes :

- N_I : est le nombre maximum de signaux conditionnants subis par un bloc du circuit.
- N_C : est le nombre maximum de signaux partagés dans l'approche CPM modulaire non-disjoint qui reconvergent dans un bloc.
- N_S : est le nombre total de sources de reconvergence du circuit.
- N_B : est le nombre total de blocs du circuit.

Pour l'approche disjointe nous pouvons proposer une borne supérieure pour la complexité qui est donnée par $O(N_B \cdot 4_I^N)$. c'est-à-dire, le nombre de blocs à traiter multiplié par la complexité du bloc le plus restreignant. Donc, cette complexité est linéaire avec le nombre de blocs et exponentielle avec la taille du plus grand bloc. A partir de ceci, nous pouvons faire deux hypothèses :

- Pour un circuit avec un nombre très élevé de blocs $N_B >> 4_I^N$), l'élément dominant du produit $N_B \cdot 4_I^N$, est N_B. Ainsi, la complexité de la méthode est approximativement linéaire. En utilisant la notation $4_I^N = k$, la complexité devient $k \cdot N_B$, donc une croissance linéaire avec N_B avec une pente de $k = 4_I^N$
- Pour le cas où $N_B >> 4_I^N$) n'est pas valable, l'élément le plus restreignant est 4_I^N, donc une croissance exponentielle avec N_I, sachant qu'idéalement $N_I << N_S$

Pour l'approche non-disjointe, nous faisons l'hypothèse $N_C > N_I$, ce qui est réaliste pour des circuits qui ne sont pas petits. Dans cette hypothèse nous pouvons estimer la complexité de CPM modulaire $O(4^{N_C})$. Ainsi, la partition optimale en blocs du circuit doit être faite de manière à respecter la condition $N_I < N_C$ et en minimisant la valeur de N_C. Pour le cas extrême N_C vaut 0, c'est-à-dire une séparation disjointe. Nous pouvons donc affirmer qu'un respectant les hypothèses montrées ci-dessus lors du découpage du circuit, la méthode CPM hiérarchique disjointe montre une complexité moindre.

Chapitre 4

Application des techniques probabilistes au durcissement sélectif et à l'analyse FMDEA : étude de l'additioneur Brent-Kung 32 bits

4.1 Introduction

L'analyse de la fiabilité telle que nous l'avons exposée jusqu'ici est une mesure qui fournit notamment des informations qui peuvent être utilisées de deux manières :

1. Le concepteur évalue la fiabilité de son système, et décide d'une éventuelle modification dans le cas où cette fiabilité serait jugée insuffisante.
2. Le concepteur garantit à son client que le circuit fonctionnera avec la fiabilité calculée pour l'ensemble des conditions considérées lors de l'analyse réalisée.

L'analyse de fiabilité telle que présentée dans les chapitres précédents fournit des informations permettant de caractériser un circuit comme fiable/non-fiable compte tenu du niveau de fiabilité attendu. Nous montrerons dans ce chapitre l'utilisation de ces informations afin de :

- Proposer une méthodologie de conception visant le durcissement du circuit de manière optimale en termes de coût et de gain de fiabilité.
- Garantir une borne inférieure de la fiabilité pour toutes les conditions d'utilisation possibles.

Le premier point est lié au durcissement sélectif. Dans le deuxième point, il s'agit de l'analyse FMDEA (Failure Mode Diagnosis and Effect Analysis) [98]. Par la suite du document nous présentons plus en détail ces deux types d'analyses ainsi que les méthodologies que nous proposons pour les réaliser.

4.2 Durcissement sélectif

4.2.1 Introduction

Les techniques de protection des circuits aux fautes consistent notamment en l'ajout de redondances fonctionnelles et blocs de détection et/ou correction d'erreurs (c.f. chapitre

1). Elles ont un impact significatif sur les métriques de qualité du circuit (surface, consommation et vitesse). Une des approches les plus répandues pour le durcissement des circuits est la redondance modulaire triple (TMR : Triple Modular Redundancy), qui conduit à une augmentation en surface de 200% [99]. Le durcissement sélectif vise à réduire le coût lié à l'ajout de redondance. Dans le cas de la TMR, il s'agit de remplacer la triplication exhaustive de tous les blocs constituant le circuit par une triplication ciblée à seulement quelques uns de ces blocs. Ces blocs en question sont choisis en fonction de leur rôle dans la transformation d'une faute en erreur.

4.2.2 Techniques de durcissement

Plusieurs approches ont été proposées dans la littérature pour mener à terme le durcissement sélectif. Par la suite nous étudions les travaux principaux concernant ce sujet.

Le travail de Mohanram et Touba [100] propose un classement des noeuds du circuit selon leur criticité, qui est calculée à partir de trois paramètres :

1. La probabilité que la valeur d'une sortie dépende de l'état d'un noeud
2. Le taux avec lequel est généré un SEU de puissance suffisante pour être propagé jusqu'aux sorties
3. La probabilité que le SEU soit capturé par une des latches en sortie.

Ces travaux concernent seulement la triplication de la logique séquentielle et des mémoires, sans prendre en compte l'effet de la logique combinatoire.

Dans [101] est décrite une méthodologie pour rendre robustes aux radiations les cellules de mémorisation insérées dans des FPGAs. Cela se fait à partir d'un classement réalisé selon la sensibilité du système à une éventuelle faute dans ces cellules. Pour ce classement, les auteurs prennent en compte la classification des fautes selon leur effet sur le système : persistant ou non-persistant. Les cellules les plus enclines à provoquer des fautes persistantes sont davantage robustifiées.

La méthode proposée dans [102] tient compte des fautes dans la logique. Le classement des portes et des entrées est effectué selon la probabilité de propagation d'une éventuelle faute ayant lieu sur chacune de ces portes et entrées. Deux heuristiques pour calculer les probabilités des signaux sont proposées. Pour la première, exacte, aucune estimation de la complexité est faite, mais les auteurs affirment qu'elle est difficilement applicable à des circuits de grande taille. Quant à la deuxième heuristique, non exacte, les auteurs affirment qu'elle conduit à une erreur de 34% pour le pire cas lors de son application sur un circuit de 2 entrées et 7 portes. Cela laisse supposer que l'erreur peut être importante pour des circuits de taille plus importante.

L'approche décrite dans [103] fait suite aux travaux de Asadi [79]. La méthodologie est basée sur un classement des portes à partir de leur EPP. Comme il a été énoncé dans le chapitre 2, l'approche EPP n'est pas exacte, ce qui peut provoquer un durcissement non-optimal. Ceci parce que le classement des portes réalisé à partir de résultats inexacts pourrait amener à durcir davantage des portes qui ont été jugées critiques de manière erronée.

Une approche basée sur le classement des portes logiques selon leur contribution au SER est proposée dans [104]. Pour chaque porte, les auteurs calculent un facteur qui tient compte de la probabilité qu'un SET soit généré dans cette porte. Ce facteur est une fonction des caractéristiques physiques de la porte et de la distribution de probabilité de ses entrées. Cette probabilité est pondérée par le facteur de masquage logique et électrique

correspondant à une faute dans cette porte. Pour calculer la distribution de probabilité d'entrée des portes et les probabilités des signaux permettant d'obtenir le taux de masquage logique, les auteurs proposent l'utilisation d'une méthode dite de *forward* et *backward traversing*, dont ils ne présentent pas d'estimation de sa complexité ni sa précision.

Polian et al. [105] présentent une méthode de calcul de la contribution de chaque noeud au SER total tenant compte des effets de masquage logique. La méthode utilisée pour calculer les probabilités des signaux a une complexité linéaire avec la taille du circuit mais ne prend pas en compte les corrélations.

Un classement des noeuds selon leur contribution au SER total du circuit est également proposé dans [106]. Pour ce faire, les auteurs prennent en compte les effets de masquage électrique, temporel et logique. Pour mesurer le masquage logique ils utilisent le concept d'observabilité, qui est inversement proportionnel au facteur de masquage logique de chaque noeud. L'algorithme permettant de calculer les probabilités des signaux afin d'obtenir l'observabilité calcule la borne supérieure de l'observabilité dans le cas de signaux reconvergents, ce qui peut conduire à un mauvais classement des noeuds.

Le travail de Marques et al. [107] propose une approche permettant d'améliorer la fiabilité d'un système en respectant un budget (consommation, surface, fiabilité, etc.) donné. Cette approche prend en compte le masquage logique des fautes. La procédure utilisée par cette méthodologie utilise deux étapes :

1. Construction d'une libraire d'architectures candidates.
2. Evaluation des candidates et choix d'une architecture.

Dans le premier pas, les architectures candidates sont construites de manière à respecter le budget en termes de consommation et surface. Aussi, pour qu'une architecture soit candidate, sa fiabilité doit être supérieure à celle du circuit original. Dans le deuxième pas, les architectures sont ordonnées suivant une métrique de qualité définie par le concepteur, et l'architecture choisie pour l'implémentation est la première à atteindre le niveau de fiabilité spécifié dans le budget. Cette approche fournit un résultat optimal pour le rapport coût/fiabilité de l'architecture choisie. Cependant, la construction de toutes les architectures candidates possibles ainsi que l'évaluation de leur fiabilité (réalisée avec la méthode SPRMP) a une grande complexité algorithmique. Cette méthodologie se base sur une approche réalisée par force brute.

Le travail de Naviner et al. [108] propose une méthode pour choisir les éléments à durcir davantage pour un circuit de logique combinatoire. Les auteurs proposent deux critères pour établir le choix :

1. **Sensibilité** : il s'agit d'une mesure permettant d'évaluer l'impact d'une éventuelle perte de fiabilité d'un élément sur la fiabilité de la totalité du circuit. Plus l'impact est important, plus l'élément est sensible.
2. **Eligibilité** : cette mesure donne l'ordre dans lequel les différents éléments doivent être durcis de manière à obtenir un gain maximal de fiabilité pour la totalité du circuit.

Les résultats fournis par cette approche sont optimaux. La méthode utilisée pour l'évaluation de la fiabilité est le PBR [74] (c.f. chapitre 2), en faisant l'hypothèse que seules des fautes simples peuvent avoir lieu. Lors de l'analyse d'un circuit de taille importante, le nombre d'états du système à simuler ainsi que le nombre de fautes à injecter par l'approche PBR grandissent exponentiellement, ce qui peut restreindre le nombre de circuits traitables par cette approche.

Une alternative au durcissement sélectif est la tolérance sélective aux fautes, consistant en la protection du circuit à d'éventuelles fautes ayant lieu dans des signaux spécifiques, préalablement considerés comme critiques [109]. Ce type d'analyse est plutôt lié à la fiabilité fonctionnelle et non à la fiabilité du signal. Aucune information sur la complexité algorithmique de l'approche proposée n'est donnée dans ce travail.

Commentaires

Pareillement aux approches analytiques d'estimation de la fiabilité, les techniques d'analyse de la sensibilité et criticité des noeuds du circuit souffrent des mêmes limitations :

1. Complexité algorithmique : [107, 108]
2. Imprécisions : [102, 103, 104, 105]

Logiquement, cette conclusion est triviale car les méthodes d'analyse de la sensbilité sont basées sur les méthodes d'analyse de la fiabilité. Par conséquent, la faisablité de ce type d'analyse repose principalement sur la faisabilité de la méthode analytique prise comme base de l'analyse.

4.2.3 Calcul de la sensibilité de la fiabilité aux fautes des portes

Nous proposons une approche basée sur l'utilisation de la méthode CPM pour analyser l'influence de la fiabilité de chaque porte du circuit sur la fiabilité finale du circuit. Cette analyse permet d'identifier les portes qui offriront une augmentation optimale de la fiabilité dans le cas d'un éventuel durcissement sélectif du circuit.

Dans un premier temps, nous montrons comment nous pouvons trouver la fiabilité d'une porte en fonction de sa probabilité d'erreur. Ensuite, nous appliquons ce principe pour calculer la fiabilité d'un circuit en fonction des probabilités d'erreur dans ses portes.

4.2.3.1 Probabilité d'un signal en fonction des probabilités d'erreur de la porte

Il est possible de calculer la fiabilité d'une porte logique en fonction de sa probabilité d'erreur de multiples façons. Nous proposons d'utiliser les propriétés des probabilités conditionnelles pour ce faire. Dans la figure 3.4 du chapitre 3 nous avons utilisé une décomposition en forme d'arbre pour introduire le conditionnement par des signaux. Cette décomposition permet aussi d'introduire le conditionnement par l'état de la porte : défaillant ou correct. Considérons la porte logique de la figure 4.1, avec une probabilité d'erreur p_e.

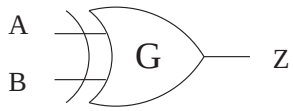


Fig. 4.1 – Porte logique XOR à deux entrées utilisée pour illustrer le calcul de fiabilité en fonction de la probabilité d'erreur de la porte.

Nous pouvons modéliser le comportement fautif de cette porte à partir d'un arbre de décision simplifié (voir figure 4.2). Sur cette figure, les noeuds représentent respectivement :

– $S1$: tous les états des entrées de la porte qui mènent à l'état 0_c de la sortie lorsque la porte réalise l'opération correctement ou à l'état 1_i lors d'une opération défaillante.

- $S2$: les états qui mènent à l'état 1_i en cas d'opération normale et à l'état 0_c autrement.
- $S3$: les états qui mènent à l'état 0_i en cas d'opération normale et à l'état 1_c autrement.
- $S4$: les états qui mènent à l'état 1_c en cas d'opération normale et à l'état 0_i autrement.

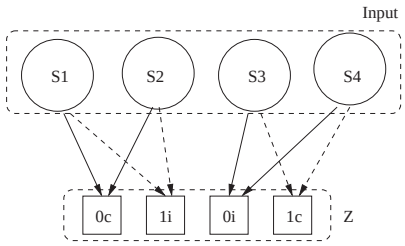


FIG. 4.2 – Arbre de décision simplifié.

En supposant que les propabilités des états des entrées , $p(S1)$, $p(S2)$, $p(S3)$ et $p(S4)$, soient connues, et en définissant la probabilité d'opération correcte dans la porte comme $q_e = 1 - p_e$, la probabilité des différents états du signal Z de sortie est donnée par :

$$\begin{aligned} p(Z = 0_c) = p(z_1) &= p(S1) \cdot q_e + p(S2) \cdot p_e \\ p(Z = 1_i) = p(z_2) &= p(S2) \cdot q_e + p(S1) \cdot p_e \\ p(Z = 0_i) = p(z_3) &= p(S3) \cdot q_e + p(S4) \cdot p_e \\ p(Z = 1_c) = p(z_4) &= p(S4) \cdot q_e + p(S3) \cdot p_e \end{aligned} \tag{4.1}$$

Lorsque nous conditionnons l'état du signal Z par l'état d'un signal S_i, nous figeons la probabilité de cet état S_i à '1' et celles de tous les autres états $S_j \neq S_i$ à '0'. Pour faire un conditionnement par rapport à l'état de la porte, nous pouvons appliquer le même principe en agissant sur la matrice PTM qui modélise l'état de la porte. Ainsi, nous pouvons trouver les expressions des équations en 4.1 en appliquant la méthode SPR en deux pas :

1. Premièrement, en considérant un comportement idéal : $PTM = ITM$.
2. Puis, en considérant un comportement complètement erroné. Pour cela nous définissons la matrice $\overline{ITM}$, de manière que $\overline{ITM(i,j)} = 0$ si $ITM(i,j) = 1$, et $\overline{ITM(i,j)} = 1$ si $ITM(i,j) = 0$.

Soit $\overline{\epsilon}$ l'état qui dénote un comportement correct de la porte, et ϵ l'état incorrect de cette porte. Nous pouvons calculer les probabilités $p(z_i/\overline{\epsilon})$ et $p(z_i/\epsilon)$ par les deux équations suivantes :

$$SPR_{Z_{\overline{\epsilon}}} = \begin{pmatrix} p(z_1/\overline{\epsilon}) & p(z_2/\overline{\epsilon}) \\ p(z_3/\overline{\epsilon}) & p(z_4/\overline{\epsilon}) \end{pmatrix} = ITM_G^T \times (SPR_A \otimes SPR_B) \times ITM_G \tag{4.2}$$

$$SPR_{Z_{\epsilon}} = \begin{pmatrix} p(z_1/\epsilon) & p(z_2/\epsilon) \\ p(z_3/\epsilon) & p(z_4/\epsilon) \end{pmatrix} = ITM_G^T \times (SPR_A \otimes SPR_B) \times \overline{ITM_G} \tag{4.3}$$

Dans les sections suivantes, nous verrons qu'il est intéressant de pouvoir mémoriser ces valeurs pour évaluer de manière efficace l'impact d'une porte sur tout un bloc. Ainsi,

nous proposons une formulation matricielle (ce qui facilite le stockage) faisant apparaître l'information de fiabilité de la porte en fonction de l'état de la porte :

$$CPM_{G/\epsilon} = \begin{pmatrix} p(z_1/\overline{\epsilon}) & p(z_1/\epsilon) \\ p(z_2/\overline{\epsilon}) & p(z_2/\epsilon) \\ p(z_3/\overline{\epsilon}) & p(z_3/\epsilon) \\ p(z_4/\overline{\epsilon}) & p(z_4/\epsilon) \end{pmatrix} \tag{4.4}$$

La première colonne de cette matrice contient les valeurs de sortie de la porte considérant un comportement idéal, tandis que la deuxième colonne contient celles pour un comportement fautif. L'expression (4.4) a été obtenue avec SPR, en considérant que les signaux d'entrée A et B sont indépendants. Dans le cas où ces signaux ne sont pas indépendants, il faut appliquer d'autres méthodes plus adaptées.

A partir des expressions des équations 4.1, nous pouvons exprimer la fiabilité de la porte comme une fonction de sa probabilité d'erreur :

$$\begin{aligned} R_G = p(z_1) + p(z_4) &= (p(S1) \cdot q_e + p(S2) \cdot p_e) + (p(S4) \cdot q_e + p(S3) \cdot p_e) \\ &= (p(S1) + p(S4)) \cdot q_e + (p(S2) + p(S3)) \cdot p_e \\ &= (p(S1) + p(S4)) \cdot (1 - p_e) + (p(S2) + p(S3)) \cdot p_e \\ &= (p(z_1/\overline{\epsilon}) + p(z_4/\overline{\epsilon})) \cdot (1 - p_e) + (p(z_1/\epsilon) + p(z_4/\epsilon)) \cdot p_e \end{aligned} \tag{4.5}$$

Cette expression est généralisable à toutes les portes logiques car il est possible de trouver les probabilités $p(z_1/\overline{\epsilon})$, $p(z_4/\overline{\epsilon})$, $p(z_1/\epsilon)$ et $p(z_4/\epsilon)$ pour n'importe quelle porte avec les équations 4.2 et 4.3. De manière intuitive, nous pouvons dire que l'équation 4.5 exprime la probabilité d'avoir un état correct en sortie de la porte en fonction du comportement de la porte (correct ou fautif).

A partir de l'équation 4.5 nous pouvons tracer une courbe qui exprime l'évolution de la fiabilité du signal Z en fonction de sa probabilité d'erreur pour les valeurs $p_e \in [0, 1]$. Nous avons réalisé ce calcul pour trois configurations différentes des signaux d'entrée, en faisant varier leur fiabilité R_{in}, figure 4.3.

- Premier cas (bleu sur la figure), les signaux d'entrée sont considérés sans faute, $R_{in} = 1$.
- Deuxième cas (vert sur la figure), $R_{in} = 0.9604$.
- Troisième cas (rouge sur la figure), $R_{in} = 0.8836$.

Cette représentation graphique permet d'interpréter l'impact du comportement alétoire de la porte sur le fonctionnement du circuit. Nous pouvons remarquer que la fiabilité atteinte en sortie de la porte dépend de la probabilité d'erreur de cette porte mais aussi de la fiabilité des signaux en entrée.

Pour illustrer l'effet conjoint de la fiabilité des signaux d'entrée et de la probabilité d'erreur d'une porte, considérons un cas très défavorable où les entrées ont une fiabilité très basse $R_{in} = 0.64$ (voir fig. 4.4). Nous pouvons constater que les valeurs de la fiabilité de sortie de la porte peuvent être supérieures à la valeur de fiabilité des entrées, ce qui constitue un gain en fiabilité dû au masquage logique introduit par cette porte (figure 4.4).

4.2.3.2 Sensibilité d'un bloc de logique combinatoire

Le calcul de fiabilité d'une porte en fonction de sa probabilité d'erreur peut être étendu au calcul de fiabilité d'un bloc logique à partir des probabilités d'erreur des portes logiques

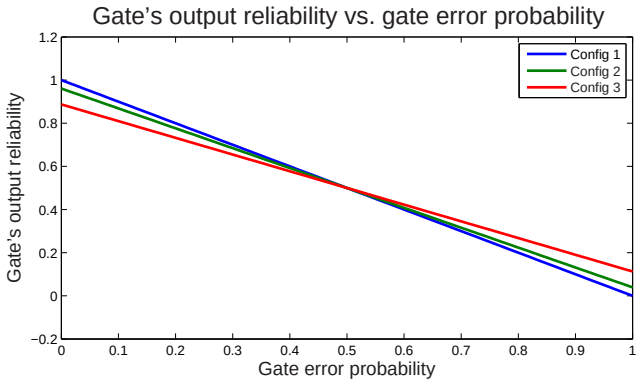


FIG. 4.3 – Fiabilité de la porte G en fonction de sa probabilité d'erreur pour 3 configurations des entrées.

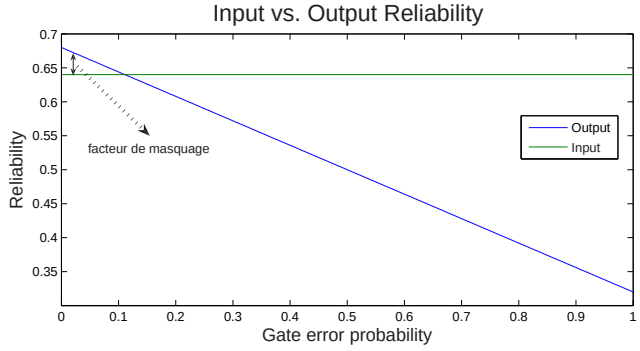


FIG. 4.4 – Fiabilité de la porte G en fonction de sa probabilité d'erreur pour 3 configurations des entrées.

qui le constituent. Ce calcul permet d'identifier les rôles relatifs des portes logiques vis-à-vis de la fiabilité du bloc, ce qui est très utile pour le durcissement sélectif.

Pour un bloc de logique combinatoire composé de N portes nous proposons les définitions suivantes :

- Y est la sortie du bloc
- G_i correspond à la i-ème porte du circuit.
- S_i est le signal de sortie de la porte G_i.
- p_{ei} est la probabilité d'erreur de la porte G_i.
- $\overline{\epsilon_i}$ est l'état correspondant à un fonctionnement correct de la porte G_i.
- ϵ_i est l'état correspondant à un fonctionnement incorrect de la porte G_i.

L'objectif est de calculer la fiabilité de la sortie Y comme une fonction des p_{ei}. Pour ce faire, il faut calculer toutes les probabilités $p(y_j/s_{i_k})$, c'est-à-dire, la probabilité de l'état y_j (signal Y étant dans l'état j) quand le signal S_i est dans l'état k. Nous devons aussi calculer la probabilité des états du signal S_i en fonction de l'état de la porte G_i comme nous l'avons exposé dans la section précédente de manière à obtenir : $p(s_{i_k}/\overline{\epsilon_i})$ et $p(s_{i_k}/\epsilon_i)$.

Ces probabilités peuvent être obtenues par les deux équations suivantes :

$$p(y_j/\overline{\epsilon_i}) = \sum_{k=1}^{4} p(y_j/s_{i_k}) \cdot p(s_{i_k}/\overline{\epsilon_i}) \tag{4.6}$$

$$p(y_j/\epsilon_i) = \sum_{k=1}^{4} p(y_j/s_{i_k}) \cdot p(s_{i_k}/\epsilon_i) \tag{4.7}$$

De ces deux équations nous pouvons extraire la fiabilité comme une fonction de la probabilité d'erreur de la porte G_i :

$$R(p_{ei}) = (p(y_1/\overline{\epsilon_i}) + p(y_4/\overline{\epsilon_i})) \cdot (1 - p_{ei}) + (p(y_1/\epsilon_i) + p(y_4/\epsilon_i)) \cdot p_{ei} \tag{4.8}$$

Pour la mise en oeuvre de cette approche, l'introduction du conditionnement par le signal S_i représente une augmentation de la complexité du calcul d'un facteur 4. Lorsque le circuit n'a pas, ou peu de sources de reconvergence, cette augmentation peut ne pas être très significative. Cependant, dans le cas général et pour réduire au maximum le coût de l'analyse, il est préférable d'introduire le conditionnement directement par l'état de la porte.

Le conditionnement par rapport à l'état de la porte est réalisé en faisant l'analyse de fiabilité "classique" avec l'approche CPM jusqu'à la porte G_i considérée, et en rajoutant les deux états possibles de la porte G_i pour la suite de l'analyse. Cela induit un coût dans la complexité d'un facteur 2. L'introduction du conditionnement se fait par la modification de la matrice PTM de la porte G_i en deux étapes, comme montré dans la section précedente :

- $PTM_{G_i} = \overline{ITM_{G_i}}$, état sans faute.
- $PTM_{G_i} = \overline{ITM_{G_i}}$, comportement fautif.

Considérons une topologie comme celle représentée sur la figure 4.5. Les signaux d'entrée de la porte G_i sont conditionnés par l'ensemble des signaux sources de reconvergence T (indiqué sur les équations avec la notation I/T_m). Les deux opérations à réaliser pour introduire le conditionnement par ϵ_i sont :

$$Y_i/(T_m \cap \epsilon_i) = ITM_{NAND}^{T} \cdot I/T_m \cdot \overline{ITM_{NAND}} \tag{4.9}$$

$$Y_i/(T_m \cap \overline{\epsilon_i}) = ITM_{NAND}^{T} \cdot I/T_m \cdot ITM_{NAND} \tag{4.10}$$

Ces opérations doivent être réalisées pour tous les états T_m des signaux de conditionnement des entrées. Lorsque le signal de sortie de G_i est une source de reconvergence (cf. figure 4.6), la procédure varie légèrement. L'idée consiste à stocker le conditionnement de l'état de la porte et à propager seulement le conditionnement par le signal de sortie Z_i. S'il s'agit d'une source primaire (c'est-à-dire, qu'elle n'est pas conditionnée par d'autres sources), il faut stocker une matrice comme décrit dans la section précédente avec l'équation (4.4).

Pour illustrer cette approche nous montrons de manière détaillée les calculs à réaliser pour un bloc correspondant à la fonction logique XOR réalisée avec des portes logiques de type NAND (voir figure 4.7).

Pour ce circuit, nous considérons que les entrées A et B sont idéales et elles ont une distribution uniforme :

$$SPR_A = SPR_B = \begin{pmatrix} 0.5 & 0 \\ 0 & 0.5 \end{pmatrix}$$

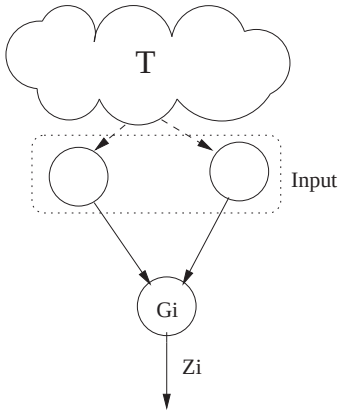


FIG. 4.5 – Exemple de circuits avec des signaux corrélés.

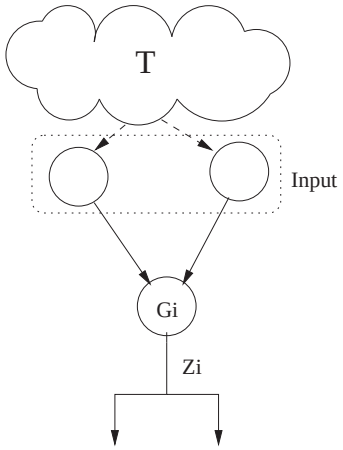


FIG. 4.6 – Exemple de configuration d'analyse de porte créant des corrélations.

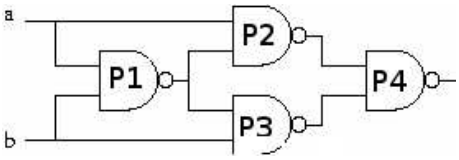


FIG. 4.7 – Implémentation de la fonction logique XOR avec des portes de type NAND.

Les portes du circuit sont considérées toutes les deux avec la même probabilité d'erreur $p_e = 0.05$:

$$ITM_{NAND} = \begin{pmatrix} 0 & 1 \\ 0 & 1 \\ 0 & 1 \\ 1 & 0 \end{pmatrix}, PTM_{NAND} = \begin{pmatrix} 0.05 & 0.95 \\ 0.05 & 0.95 \\ 0.05 & 0.95 \\ 0.95 & 0.05 \end{pmatrix}$$

Nous détaillons ensuite les calculs à réaliser pour analyser les effets des portes P_1, P_2 et P_4. La symétrie du circuit nous permet d'éviter de détailler les calculs pour P_3, car la

procédure est identique à celle appliquée pour P_2.

Calculs pour analyser la sensibilité à P_1 Le signal z_1 est une source secondaire de reconvergence conditionnée par les sources primaires A et B. Nous calculons donc les probabilités :

- Porte P_1 : $p(z_{1_i}/a_j \cap b_k \cap \epsilon_1)$ et $p(z_{1_i}/a_j \cap b_k \cap \overline{\epsilon_1})$.
- Porte P_2 : $p(z_{2_i}/a_j \cap z_{1_l} \cap \epsilon_1)$ et $p(z_{2_i}/a_j \cap z_{1_l} \cap \overline{\epsilon_1})$.
- Porte P_3 : $p(z_{3_i}/b_k \cap z_{1_l} \cap \epsilon_1)$ et $p(z_{3_i}/b_k \cap z_{1_l} \cap \overline{\epsilon_1})$.
- Porte P_4 : $p(z_{4_i}/a_j \cap b_k \cap z_{1_l} \cap \epsilon_1)$ et $p(z_{4_i}/a_j \cap b_k \cap z_{1_l} \cap \overline{\epsilon_1})$.

Après obtention des probabilités de la sortie conditionnées par les sources de reconvergences et ϵ_1, nous pouvons trouver la sortie conditionnée seulement par ϵ_1 :

$$p(z_{4_i} \cap \epsilon_1) = \sum_{j=1}^{4}\sum_{k=1}^{4}\sum_{l=1}^{4} p(z_{4_i}/a_j \cap b_k \cap z_{1_l} \cap \epsilon_1) \cdot p(a_j) \cdot p(b_k) \cdot p(z_{1_l}/a_j \cap b_k \cap \epsilon_1) \quad (4.11)$$

$$p(z_{4_i} \cap \overline{\epsilon_1}) = \sum_{j=1}^{4}\sum_{k=1}^{4}\sum_{l=1}^{4} p(z_{4_i}/a_j \cap b_k \cap z_{1_l} \cap \epsilon_1) \cdot p(a_j) \cdot p(b_k) \cdot p(z_{1_l}/a_j \cap b_k \cap \overline{\epsilon_1}) \quad (4.12)$$

Calculs pour analyser la sensibilité à P_2 Pour la porte P_2, nous pouvons profiter des calculs réalisés pour la porte P_1 sachant que $p(\epsilon_1) = p_e$ et $p(\overline{\epsilon_1}) = 1 - p_e$. De même, nous pouvons directement reporter les calculs réalisés pour la porte P_3 :

- Porte P_1 : $p(z_{1_i}/a_j \cap b_k) = p(z_{1_i}/a_j \cap b_k \cap \epsilon_1) \cdot p_e + p(z_{1_i}/a_j \cap b_k \cap \overline{\epsilon_1}) \cdot (1 - p_e)$
- Porte P_2 : $p(z_{2_i}/a_j \cap z_{1_l} \cap \epsilon_2)$ et $p(z_{2_i}/a_j \cap z_{1_l} \cap \overline{\epsilon_2})$.
- Porte P_4 : $p(z_{4_i}/a_j \cap b_k \cap z_{1_l} \cap \epsilon_2)$ et $p(z_{4_i}/a_j \cap b_k \cap z_{1_l} \cap \overline{\epsilon_2})$.

Les probabilités $p(z_{4_i}/\epsilon_2)$ et $p(z_{4_i}/\overline{\epsilon_2})$ sont obtenues avec :

$$p(z_{4_i} \cap \epsilon_2) = \sum_{j=1}^{4}\sum_{k=1}^{4}\sum_{l=1}^{4} p(z_{4_i}/a_j \cap b_k \cap z_{1_l} \cap \epsilon_2) \cdot p(a_j) \cdot p(b_k) \cdot p(z_{1_l}/a_j \cap b_k \cap \epsilon_2) \quad (4.13)$$

$$p(z_{4_i} \cap \overline{\epsilon_2}) = \sum_{j=1}^{4}\sum_{k=1}^{4}\sum_{l=1}^{4} p(z_{4_i}/a_j \cap b_k \cap z_{1_l} \cap \epsilon_2) \cdot p(a_j) \cdot p(b_k) \cdot p(z_{1_l}/a_j \cap b_k \cap \overline{\epsilon_2}) \quad (4.14)$$

Sensibilité à la porte P_4 Pour le calcul de la sensibilité à la porte P_4, nous avons déjà réalisé les calculs nécessaires pour les portes P_1, P_2 et P_3 dans les étapes précédentes. Le calcul est le même que celui réalisé dans la section 4.2.3.1 pour une porte simple.

La figure 4.8 présente les résultats de sensibilité de la fiabilité du circuit par rapport aux probabilités d'erreurs des portes P_1 à P_4. Ces résultats ont été obtenus en supposant les probabilités d'erreur dans l'intervalle $[0, 0.2]$.

Nous constatons que la porte la plus sensible est P_4. Ce résultat s'explique par le fait qu'une erreur dans la porte P_4 ne pourra pas être masquée par d'autres portes. Nous pouvons remarquer que la porte P_1, même si elle est plus éloignée de la sortie que P_2 et P_3, a le même effet qu'elles. Nous faisons l'hypothèse que cela est dû à la multiplicité de son signal de sortie. Ainsi, une éventuelle faute dans cette porte se propage par plusieurs chemins, compensant donc l'effet de la distance à la sortie par rapport à P_2 et P_3.

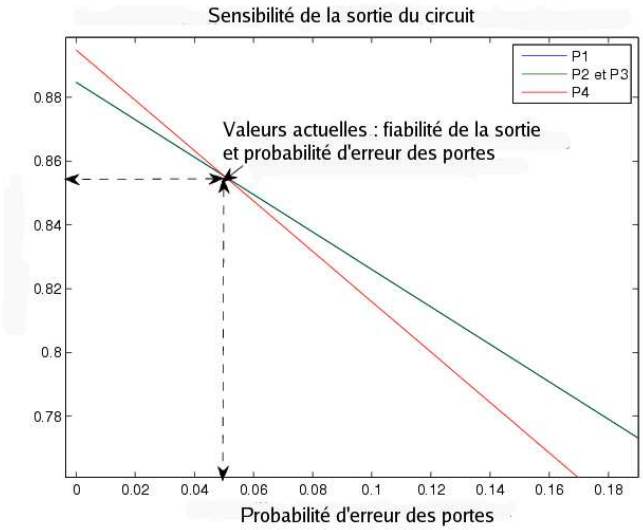


FIG. 4.8 – Résultats de la sensibilité des portes.

4.2.4 Calcul hiérarchique de la sensibilité

La méthode présentée dans la section précédente s'avère coûteuse lors de son application à des circuits complexes. Pour diminuer ce coût, nous proposons d'utiliser la méthode CPM hiérarchique. Le circuit est vu comme un assemblage de modules (blocs logiques), chaque module étant lui-même constitué d'un assemblage de portes logiques. Chaque module du circuit est analysé pour chacune de portes qu'il contient de manière à obtenir ses sorties en fonction des états de ses portes.

Considérons un circuit quelconque, découpé en blocs, comme représenté sur la figure 4.9. Pour ce circuit nous définissons :

- Z : ensemble de M signaux de sortie du circuit $(Z_1, \ldots, Z_M)$.
- Y : ensemble de K signaux de sortie du module Y $(Y_1, \ldots, Y_K)$.
- G : ensemble de N portes du module Y $(G_1, \ldots, G_N)$.
- ϵ_i représente l'état fautif de la porte G_i.
- $\overline{\epsilon_i}$ représente l'état sans faute de la porte G_i.

Supposons que nous disposons de toutes les probabilités relatives aux états de sortie du module Y conditionnés par les états de ses portes : $p(y_i/\epsilon_l)$ et $p(y_i/\overline{\epsilon_l})$, où y_i est l'état i pour le signal Y.

En suivant les régles de propagation présentées dans les sections 3.3 et 3.4, les probabilités du signal de sortie Z du circuit sont obtenues en fonction des probabilités de sortie du bloc Y $p(z_j/y_i)$:

$$p(z_j/\overline{\epsilon_l}) = p(z_j/y_i) \cdot p(y_i/\overline{\epsilon_l}) \tag{4.15}$$

$$p(z_j/\epsilon_l) = p(z_j/y_i) \cdot p(y_i/\epsilon_l) \tag{4.16}$$

Le découpage en modules permet d'éviter la propagation du conditionnement à travers les différentes portes de Y tout au long du circuit. De même, les informations déjà calculées pour un module peuvent être réutilisées pour l'analyse d'autres modules. Dans ce cas, le

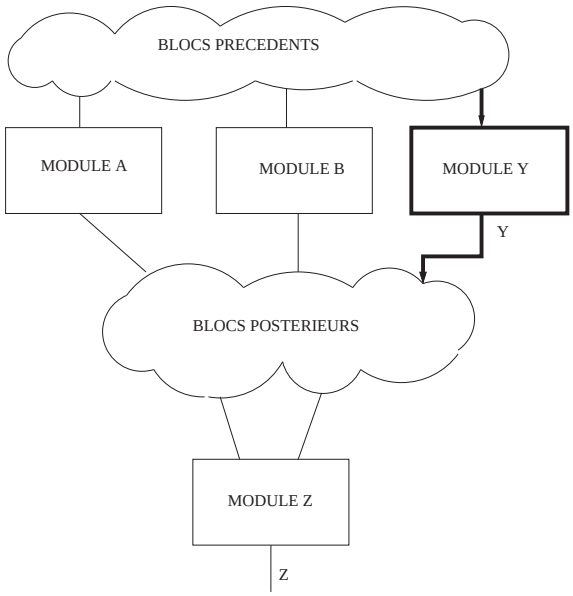


Fig. 4.9 – Exemple de circuit pour le calcul de la sensibilité de la fiabilité des sorties lors d'une approche hiérarchique.

calcul de la sensibilité se fait en changeant le conditionnement seulement par les blocs qui sont affectés par la porte qui est analysée.

L'analyse de la fiabilité du circuit en fonction de l'état des portes vise à aider le concepteur du circuit lors d'un éventuel durcissement, en donnant des directives sur les portes les plus critiques. Notamment, nous pouvons envisager deux stratégies pour quantifier l'effet de chacune des portes :

1. Les portes offrant un gain maximal lors du durcissement du circuit
2. Les portes ayant une plus faible probabilité de masquage lorsqu'elles réalisent une opération défaillante (une faute a lieu sur elles).

Le premier type d'analyse traite de la recherche de la meilleure solution en tenant compte d'un budget donné. Le budget correspond au surcoût (surface, consommation) maximal pour l'ajout de redondance. La "meilleure" solution correspond à la topologie conduisant à la fiabilité maximale ou à la topologie de budget minimal permettant d'atteindre un niveau de fiabilité donné.

Le deuxième type d'analyse permet d'évaluer les probabilités de masquage lors de l'apparition d'une faute simple dans le circuit, et donc d'identifier les portes critiques ayant un impact plus élevé sur la perte de fiabilité. Cette analyse peut être menée en considérant que toutes les portes ont un comportement idéal ($PTM = ITM$), sauf la porte sous étude, qui a un comportement fautif $PTM = \overline{ITM}$.

4.2.5 Etude de cas : Additionneur Brent-Kung de 32 bits

Pour illustrer l'application de notre méthode d'analyse de la fiabilité du circuit en fonction des éventuelles fautes dans les portes ou les entrées, nous avons choisi l'additionneur de type Brent-Kung de 32 bits. Ce circuit contient 65 signaux d'entrée, 33 signaux de sortie et 240 portes logiques.

L'additioneur de type Brent-Kung calcule et propage une retenue intermédiaire pour toutes les positions qui sont puissance de deux [110]. La structure pour un additioneur Brent-Kung de 16 bits est représenté sur la figure 4.10.

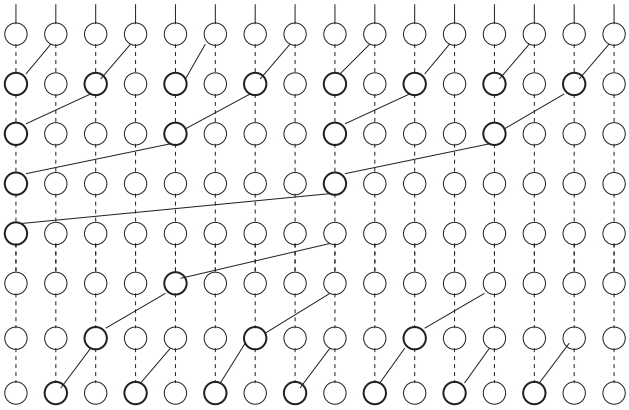


FIG. 4.10 – Structure d'un additioneur de type Brent-kung de 16 bits.

Cette structure permet de réaliser un découpage du circuit permettant d'appliquer l'approche CPM hiérarchique disjointe, c'est-à-dire que différents blocs ne partagent pas les mêmes signaux d'entrée . La possibilité d'utiliser l'approche disjointe nous permet de réaliser les analyses nécessaires, ce qui n'est pas possible en utilisant ni la méthode CPM globale ni l'approche SPRMP à cause du nombre trop élevé de sources de reconvergence.

La métrique que nous proposons pour évaluer la sensibilité d'une sortie du circuit à la fiabilité du signal en sortie d'une de ses portes est basée sur l'équation (4.8). Nous pouvons réécrire cette équation comme suit :

$$R(p_{ei}) = R_{\overline{\epsilon_i}} \cdot (1 - p_{ei}) + R_{\epsilon_i} \cdot p_{ei} \tag{4.17}$$

La représentation graphique de (4.17) est une droite de pente $R_{\epsilon_i} p_{ei} - R_{\overline{\epsilon_i}}$. La pente est d'autant plus marquée qu'une sortie est sensible à la probabilité d'erreur de la porte. Cette pente est négative car plus la probabilité d'erreur est faible (proche de 0), plus la valeur de la fiabilité sera élevée. Ainsi, nous utilisons la valeur absolue de la pente comme métrique de la sensibilité.

La sensibilité de la sortie i à la porte j sera notée $\gamma_{i,j}$. Nous représentons l'ensemble des $\gamma_{i,j}$ sous forme d'une matrice Γ, où la position (i, j) contient l'element $\gamma_{i,j}$:

$$\Gamma = \begin{pmatrix} \gamma_{1,1} & \gamma_{1,2} & \cdots & \gamma_{1,240} \\ \vdots & \cdots & \cdots & \vdots \\ \gamma_{33,1} & \gamma_{33,2} & \cdots & \gamma_{33,240} \end{pmatrix}$$

Chaque colonne dans $\gamma_{i,j}$ présente les sensibilités de toutes les sorties par rapport à une porte donnée. Par exemple, la figure 4.11 représente le contenu de la première colonne et donne la sensibilité de chacune des sorties par rapport à la porte 1 du circuit.

Nous définissons l'impact global de la porte j sur la fiabilité du circuit comme $\beta_j = \sum_{i=1}^{33} \gamma_{i,j}$. Dans ce cas, plus le paramètre β_j est élevé, plus l'impact de la porte est important. Nous avons représenté tous les termes β_j sur la figure 4.12. Cette représentation permet d'identifier facilement les portes ayant un plus fort impact sur tout le circuit.

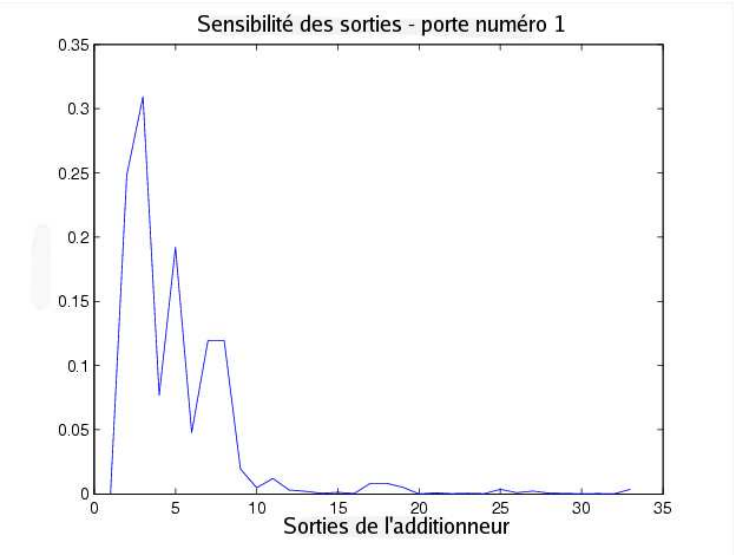


Fig. 4.11 – Sensibilité des sorties par rapport à la porte 1 du circuit.

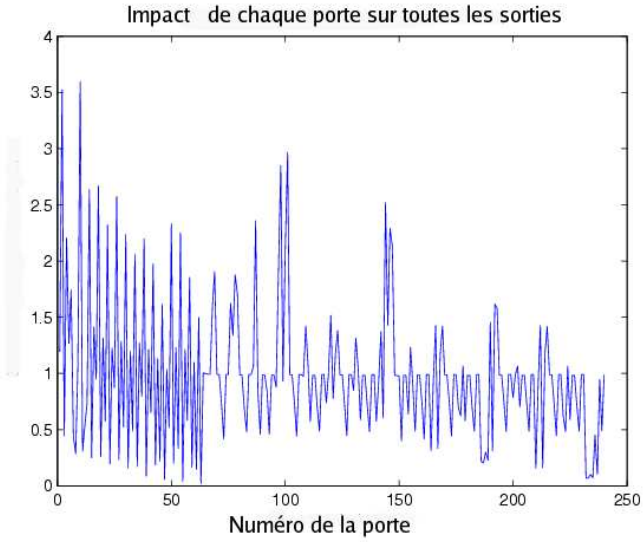


Fig. 4.12 – Sensibilité de la totalité des sorties vers toutes les portes du circuit.

Les portes ayant un numero plus bas sont celles connectées à plus de sorties. Nous pouvons remarquer que ces portes ont globalement un plus grand impact.

Supposons que le durcissement rende la porte parfaite, c'est-à-dire, avec une probabilité d'erreur nulle. Afin de quantifier le gain de la fiabilité des sorties du circuit, nous avons considéré la fiabilité des sorties de l'additionneur avant et après le durcissement des 10 portes de plus grand impact (voir figure 4.13).

Nous pouvons remarquer que le gain est plus grand pour les premières sorties et presque inexistant pour les dernières. Ceci peut être problématique si les sorties dont les gains étaient plus faibles sont les sorties les plus significatives du circuit. Par exemple, pour le

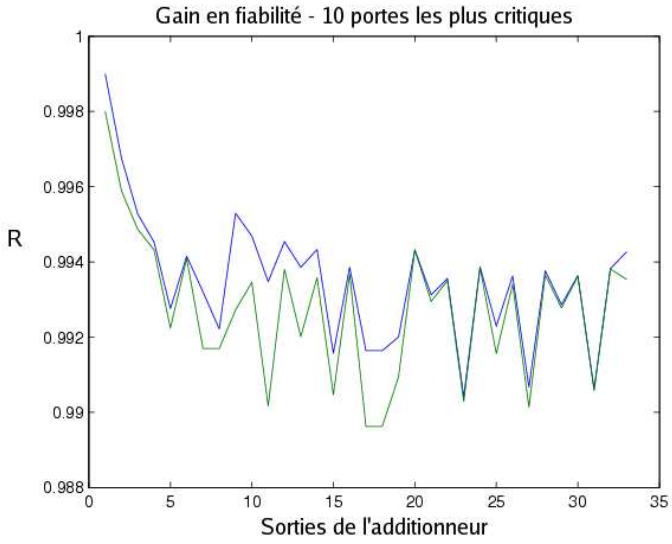


FIG. 4.13 – Fiabilité des sorties de l'additionneur avant et après le durcissement des 10 portes les plus pesantes.

cas de l'additionneur nous pouvons considérer qu'il serait préférable de durcir davantage les sorties de poids fort.

La matrice Γ permet de trouver les portes les plus significatives pour la fiabilité d'un signal donné. Si nous prenons les éléments d'une ligne de cette matrice, nous obtenons la sensibilité de la sortie représentée par cette ligne par rapport à toutes les portes du circuit. D'une part, cela nous fournit une information plus détaillée sur chaque signal, d'autre part, cela nous permet de durcir ceux qui sont définis comme critiques (suivant par exemple un critère de fonctionnalité). La figure 4.14 présente l'influence de chaque porte sur la sortie 33, celle qui donne le bit d'information le plus significatif. La figure 4.15 présente le gain de fiabilité de chacune des sorties après un durcissement des 5 portes les plus importantes pour la sortie 33.

4.3 Analyse FMDEA

4.3.1 Introduction à la norme ISO 26262

La norme ISO26262 est un standard apparu en 2010 qui porte sur la sécurité fonctionnelle des systèmes électroniques embarqués pour l'automobile. Il s'agit d'une évolution de la norme IEC61508. Cette norme introduit des mesures et des méthodologies à mettre en place pour la qualification des systèmes électroniques embarqués dans l'automobile. Les circuits sont classés en 4 niveaux différents, appelés SIL (*Security Integrity Level*). Plusieurs mesures des effets des fautes sont proposées, tant au niveau des blocs, qu'au niveau système et fonctionnel.

Il y a deux types de mesures dans cette norme qui peuvent concerner le type d'analyse que nous proposons. Ainsi, la norme stipule que s'il n'est pas possible de démontrer le contraire, toute faute ayant lieu dans le circuit deviendra visible (avec un impact au niveau fonctionnel), c'est-à-dire, qu'il ne faut pas considérer un éventuel masquage de celle-ci. Ceci

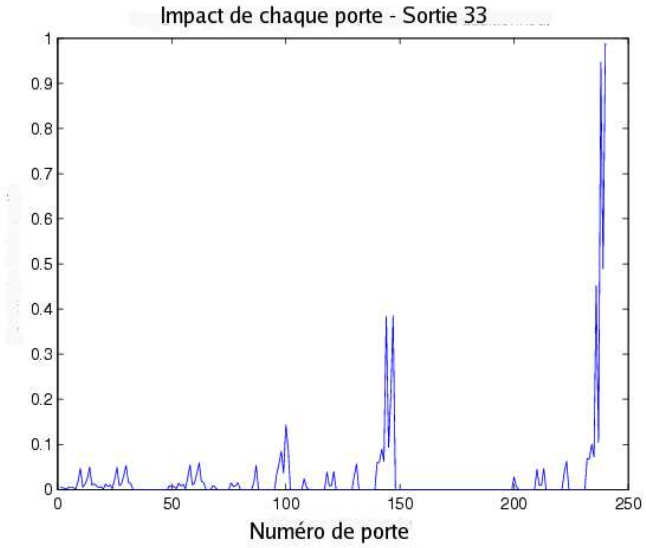


Fig. 4.14 – Sensibilité de la sortie 33 par rapport à toutes les portes du circuit.

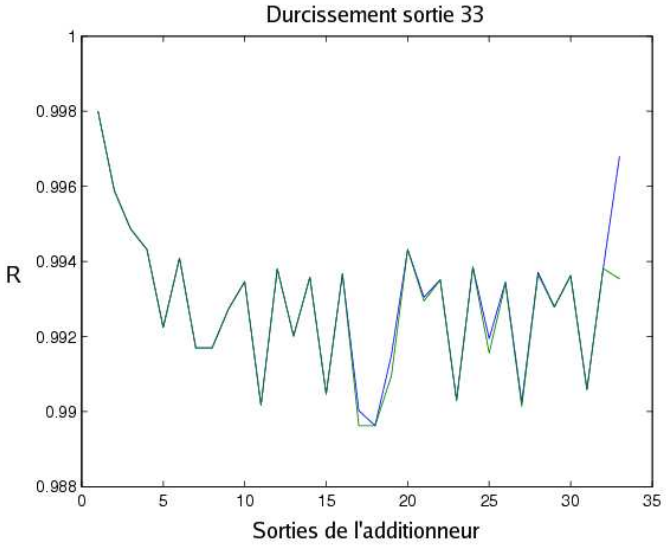


Fig. 4.15 – Gain en fiabilité des 5 portes les plus pesantes pour la sortie 33.

est une modification sensible par rapport à la norme IEC61508, qui établit qu'une faute deviendra visible avec une probabilité de 50%. Ainsi, dans le cadre de l'ISO26262, afin d'obtenir un certain niveau SIL, il faut prévoir des mesures permettant de tolérer une faute dans tous les endroits possibles du circuit. Cependant, il n'est pas réaliste de considérer que toutes les fautes deviennent visibles. Aussi, avec un calcul du taux de masquage d'une faute, tel que montré dans la section précédente, nous pouvons identifier les endroits où les fautes seront plus facilement masquées et éviter ainsi l'ajout non-nécessaire de redondances.

Un autre type de mesure intéressante pour la validation ISO concerne l'évaluation de la fiabilité du circuit. En général, les mesures de fiabilité fournies dépendent des conditions

considérées pendant l'analyse. Pour les modèles probabilistes, ces conditions consistent notamment en la probabilité d'erreur des portes et la distribution des entrées. Une amélioration de la qualité de l'analyse fournie consiste à donner une borne inférieure de la fiabilité, assurant que le système ne sera jamais en dessous de cette valeur. Dans la section qui suit nous proposons une méthodologie permettant de calculer les bornes inférieures de la fiabilité, et ensuite nous montrons les résultats obtenus utilisant cette approche sur l'additionneur brent kung 32 bits, ainsi que les résultats obtenus pour le taux de masquage des fautes.

4.3.2 Analyse de la borne inférieure de la fiabilité d'un circuit

L'utilisation des matrices de probabilités conditionnées (CPM) permet de calculer la borne inférieure de la fiabilité pour toutes les configurations possibles des entrées. Nous proposons deux approches basées sur les matrices CPM, d'abord avec l'utilisation de l'approche CPM primaire, qui permet de calculer la borne inférieure exacte pour certains cas et une borne pessimiste, dépendant de la taille du circuit analysé, et ensuite l'utilisation de l'approche CPM modulaire qui permet de trouver la borne inférieure exacte pour tous les cas.

4.3.2.1 Borne inférieure exacte de la fiabilité en utilisant l'approche CPM primaire

Soit un circuit logique combinatoire quelconque pour lequel nous pouvons calculer les probabilités des états de sa sortie conditionnés par les états des entrées, c'est-à-dire un circuit caractérisé par une matrice CPM. La borne inférieure de la fiabilité de ce circuit est donnée par l'état des entrées pour lequel la fiabilité a une valeur minimale. La matrice CPM caractérisant le circuit en fonction des états des entrées s'obtient selon les sections précédentes.

Considérons le circuit représenté sur la figure 4.16, où I représente les n entrées du circuit et Z est la sortie. Considérons aussi que I_i représente un des 4^n états possibles de l'entrée, $p(I_i)$ est la probabilité d'occurrence de cet état et $R_z(I_i)$ est la fiabilité du circuit étant donné que l'entrée est dans l'état I_i. La fiabilité du circuit peut s'exprimer par :

$$R_z = \sum_{i=1}^{4^n} R_z(I_i) \cdot p(I_i) \tag{4.18}$$

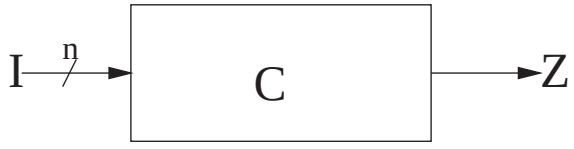


FIG. 4.16 – Circuit combinatoire générique.

Proposition 2 *La borne inférieure de la valeur de la fiabilité R_z est donnée par la valeur minimale de $R_z(I_i)$ pour toutes les distributions possibles des probabilités des entrées et toutes les valeurs de i.*

Démonstration 1 *Afin de démontrer cette proposition, considérons les définitions suivantes :*

- $R_z(I_M)$ *la valeur minimale de l'ensemble de toutes les* $R_z(I_i)$. *Ainsi* $R_z(I_M) < R_z(I_i) \forall i \neq M$.
- $p(I_{\overline{M}})$ *la probabilité d'occurrence de toute entrée autre que* I_M. *Cela signifie* $\sum_{\forall i \neq M} p(I_i) = 1 - p(I_M)$.
- $R_{\overline{M}} = \sum_{\forall i \neq M} R_z(I_i) \cdot p(I_i)$

Nous proposons une démonstration par apagogie. Ainsi nous cherchons à démontrer que l'inégalité (4.19) est fausse.

$$R_z = \sum_{i=1}^{4^n} R_z(I_i) \cdot p(I_i) < R_z(I_M) \tag{4.19}$$

D'où :

$$(1 - p(I_{\overline{M}})) \cdot R_z(I_M) + R_{\overline{M}} < R_z(I_M) \tag{4.20}$$

D'où :

$$R_z(I_M) - p(I_{\overline{M}}) \cdot R_z(I_M) + R_{\overline{M}} < R_z(I_M) \tag{4.21}$$

D'où :

$$-p(I_{\overline{M}}) \cdot R_z(I_i) + R_{\overline{M}} < 0 \tag{4.22}$$

Nous pouvons réécrire les termes $p(I_{\overline{M}}) \cdot R_z(I_i)$ *et* $R_{\overline{M}}$ *comme suit :*

$$\begin{aligned} p(I_{\overline{M}}) \cdot R_z(I_i) \quad = \quad & p(I_1) \cdot R_z(I_i) + \ldots + p(I_{M-1}) \cdot R_z(I_i) + \\ & p(I_{M+1}) \cdot R_z(I_i) + \ldots + p(I_{4^n}) \cdot R_z(I_i) \end{aligned} \tag{4.23}$$

$$\begin{aligned} R_{\overline{M}} \quad = \quad & p(I_1) \cdot R_z(I_1) + \ldots + p(I_{M-1}) \cdot R_z(I_{M-1}) + \\ & p(I_{M+1}) \cdot R_z(I_{M+1}) + \ldots + p(I_{4^n}) \cdot R_z(I_{4^n}) \end{aligned} \tag{4.24}$$

En regroupant les termes de ces deux expressions, l'inéquation 4.22 devient :

$$\sum_{\forall i \neq M} p(I_i)(R_z(I_i) - R_z(I_M)) < 0 \tag{4.25}$$

Etant donné que tous les termes $p(I_i)$ *sont égaux ou supérieurs à 0 (car il s'agit de probabilités), et que les termes* $R_z(I_i) - R_z(I_M)$ *sont aussi positifs (car* $R_z(I_i) > R_z(I_M) \forall i$*), l'inégalité donnée par l'inéquation 4.25 devient impossible. Nous démontrons ainsi que la borne inférieure de la fiabilité est donnée par* $R_z(I_M)$.

Nous pouvons trouver la borne inférieure à partir des matrices CPM. En effet, nous avons déjà montré comment calculer une matrice CPM contenant les sorties d'un bloc (ou d'un circuit) combinatoire conditionnées par ses entrées, c'est-à-dire, la probabilité que la sortie soit dans un état j donné lorsque l'entrée est dans l'état i : $p(Z_j/I_i)$ pour l'exemple de la figure 4.16. Ainsi, nous pouvons déterminer la borne inférieure du circuit sous étude en cherchant l'état Z_j qui minimise la fiabilité de Z, en parcourant tous les états I_i possibles. Dit autrement, nous pouvons déterminer la borne inférieure de fiabilité en cherchant dans la matrice CPM caractérisant le circuit, l'état des entrées qui minimise la fiabilité des sorties.

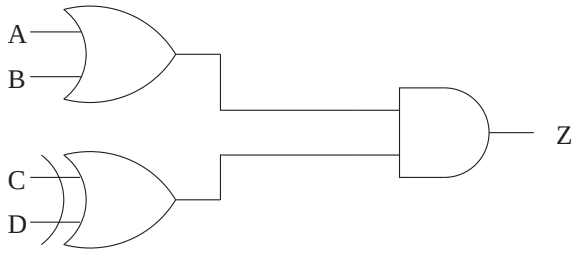


FIG. 4.17 – Circuit utilisé en exemple pour illustrer le calcul de la borne inférieure de fiabilité.

Pour illustrer cette approche, considérons par exemple le circuit représenté sur la figure 4.17. Les trois portes du circuit ont la même probabilité d'erreur égale à 0,05.

Nous avons réalisé l'analyse de la borne inférieure de fiabilité suivant deux hypothèses :

1. Les entrées sont considérées idéales.
2. Les entrées peuvent être fautives.

Pour le premier cas, la matrice CPM obtenue (contenant tous les termes $p(z_i/a_j \cap b_k \cap c_l \cap d_m)$ est représentée dans l'équation 4.26. Pour des raisons d'affichage, nous montrons la matrice transposée de la matrice CPM_z, ainsi chaque colonne représente un seul état du signal Z conditionné par les états de A, B, C et D, et chaque ligne représente les quatre états de Z conditionnés par un seul état de A, B, C et D. La matrice n'a que 16 lignes car pour les quatre entrées seuls deux des états corrects sont considérés ($4^2 = 16$).

$$CPM_z = \begin{pmatrix} 0.9990 & 0.0010 & 0 & 0 \\ 0.9980 & 0.0020 & 0 & 0 \\ 0.9980 & 0.0020 & 0 & 0 \\ 0.9980 & 0.0020 & 0 & 0 \\ 0.9980 & 0.0020 & 0 & 0 \\ 0 & 0 & 0.0030 & 0.9970 \\ 0 & 0 & 0.0030 & 0.9970 \\ 0 & 0 & 0.0030 & 0.9970 \\ 0.9980 & 0.0020 & 0 & 0 \\ 0 & 0 & 0.0030 & 0.9970 \\ 0 & 0 & 0.0030 & 0.9970 \\ 0 & 0 & 0.0030 & 0.9970 \\ 0.9990 & 0.0010 & 0 & 0 \\ 0.9980 & 0.0020 & 0 & 0 \\ 0.9980 & 0.0020 & 0 & 0 \\ 0.9980 & 0.0020 & 0 & 0 \end{pmatrix} \quad (4.26)$$

La valeur minimale de fiabilité donnée par cette matrice se trouve dans les lignes 7,8,9,11,12 et 13, et elle est de 0.997.

Pour le cas où les entrées peuvent contenir des fautes, nous ne pouvons pas montrer la matrice obtenue (256 colonnes, ou 256 lignes pour sa matrice transposée). Dans ce cas, la valeur minimale trouvée pour la fiabilitée est de 0.001, ce qui est extrêmement pessimiste.

Commentaires

Nous avons présenté une méthode permettant de calculer la borne inférieure de la fiabilité à partir de l'approche CPM. Cette limite calculée de manière stricte est nécessaire pour la certification du circuit. Elle est également nécessaire dans le cas d'applications critiques (par exemple, militaires, médicales ou dans le domaine de la sécurité).

Dans le cas où les contraintes de fiabilité sont moins sévères, nous pouvons nous contenter d'une borne moins pessimiste pour la fiabilité. En effet, nous pouvons prévoir que la borne inférieure sera donnée pour des cas où toutes ou un grand nombre d'entrées se trouvent dans un état fautif, or cette configuration est très peu probable.

Pour obtenir une borne moins pessimiste nous pouvons, par exemple, supposer que les entrées sont correctes ou bien limiter le nombre de fautes possibles à considérer. Le calcul de la borne inférieure est donc fait en prenant en compte seulement les colonnes de la matrice CPM correspondant à des états qui vérifient les conditions que nous avons imposées.

L'impact de particules sur un circuit peut affecter plusieurs portes dans un même cycle d'horloge. Ce phénomène est connu sous le nom de Multiple Cell Upset (MCU). Les études de [111] ont démontré que l'occurrence de MCU élevés dus à des impacts simultanés de plusieurs particules est peu probable. Ainsi, nous pouvons limiter de manière réaliste le nombre d'entrées étant dans un état fautif au nombre maximum de portes qui peuvent être affectées lors d'un seul impact de particule. Il faut toutefois noter que l'occurrence de MCU dépend du noeud technologique utilisé. Plus la technologie est avancée, plus le nombre de cellules pouvant être affectées par l'impact d'une particule est important.

D'un autre côté, la méthode basée sur l'approche CPM primaire est limitée par la taille du circuit. En effet, la taille de la matrice CPM caractérisant le circuit croît exponentiellement avec la taille du circuit. Cette limitation peut être surmontée en faisant une partition du circuit. Ceci est illustré dans la figure 4.18, où I représente les n entrées du circuit et Z en est la sortie.

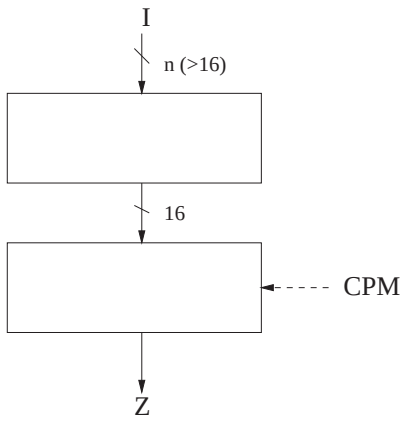


Fig. 4.18 – Diagramme de blocs représentant les limites de capacité d'analyse de la borne inférieure de l'approche CPM.

Nous pouvons trouver une partition du circuit de manière à regrouper la sortie du circuit dans un bloc ayant 16 entrées au maximum (10 ou 12 dans la réalité menant à des matrices de quelques millions de colonnes). Nous calculons alors la matrice CPM de ce bloc contenant les états des sorties en fonction des entrées afin d'obtenir la borne inférieure

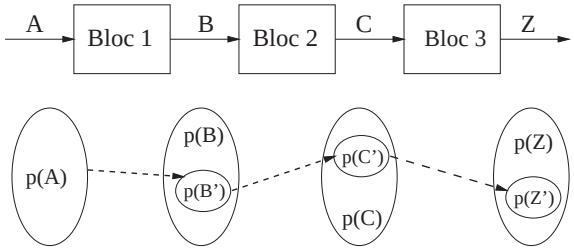


FIG. 4.19 – Diagramme représentant les blocs d'un circuit et la propagation des espaces de probabilités correspondants.

de ce bloc. L'estimation trouvée en faisant cette partition n'est pas la borne inférieure de valeur maximale. C'est-à-dire, sans la limitation technique due à la capacité de stockage des matrices CPM on pourrait trouver une borne inférieure de la fiabilité qui serait moins pessimiste que celle trouvée avec la partition du circuit. Dans la section suivante nous proposons l'utilisation de la méthode CPM modulaire afin de trouver une borne moins pessimiste pour des circuits de taille importante.

4.3.2.2 Borne inférieure exacte de la fiabilité en utilisant l'approche CPM hiérarchique disjointe

L'utilisation de l'approche CPM hiérarchique change sensiblement la manière de calculer la borne inférieure de la fiabilité par rapport à l'approche CPM. Avec l'approche CPM, dès qu'il est possible de trouver une expression qui modélise le comportement du circuit en sa totalité (la matrice CPM), c'est-à-dire tous les états possibles, il suffit de trouver le pire cas dans cette expression. Cette démarche devient impossible lorsque nous ne disposons pas d'une forme unitaire pour décrire le comportement du circuit. Pour le cas où l'approche basée sur CPM n'est pas applicable, nous proposons une méthodologie consistant en la partition du circuit comme dans l'approche CPM hiérarchique disjointe et en la propagation de toutes les distributions possibles entre les blocs. Ce principe est illustré à partir de l'exemple de la figure 4.19.

Les différents blocs représentent la partition du circuit, les signaux A, B, C et Z sont les entrées et sorties des blocs (dont A et Z sont les entrées et sorties primaires respectivement). Le terme $p(A)$ représente tous les états possibles des entrées. Les termes $p(B)$, $p(C)$ et $p(Z)$ sont les distributions de ces signaux. La figure 4.19 représente aussi l'ensemble de toutes les distributions possibles de $p(A)$, $p(B)$, $p(C)$ et $p(Z)$, et les sous-ensembles $p(B')$, $p(C')$ et $p(Z')$. $p(B')$ est le sous-ensemble de toutes les distributions possibles de $p(B)$ à la sortie du bloc 1 étant donnés tous les états possibles des signaux d'entrée A et leur distribution associée $p(A)$. De même pour $p(C)$ et $p(C')$ étant donné $p(B')$ en entrée, et pour $p(Z)$ et $p(Z')$ étant donné $p(C')$ en entrée. Cette procédure permet de réduire le nombre d'états à propager et donc, réduire la complexité algorithmique de l'approche.

Par exemple, considérons la matrice CPM d'une porte XOR étant conditionnée par une de ses entrées (l'autre entrée est considérée avec une distribution idéale et uniforme), et une probabilité d'erreur de 0.05. Dans cette matrice, seulement deux valeurs pour les colonnes sont obtenues, de manière que nous pouvons nous contenter de propager seulement ces deux

valeurs, et non les quatre valeurs correspondant aux quatre états du conditionnement.

$$CPM_{XOR} = \begin{pmatrix} 0.4995 & 0.0005 & 0.0005 & 0.4995 \\ 0.0005 & 0.4995 & 0.4995 & 0.0005 \\ 0.0005 & 0.4995 & 0.4995 & 0.0005 \\ 0.4995 & 0.0005 & 0.0005 & 0.4995 \end{pmatrix}$$

La borne inférieure exacte de la fiabilité du circuit de la figure 4.19 sera donnée par l'état de $p(Z')$ ayant une fiabilité plus faible.

Quant à l'approche basée sur le CPM primaire appliquée sur des circuits d'une taille supérieure à ce qui est modélisable par une simple matrice CPM, considérons par exemple le circuit de la figure 4.19, et que nous diposions seulement de la matice CPM_Z. Ainsi, le sous-ensemble d'états de Z contenus dans cette matrice est plus important que celui trouvé avec l'approche CPM hiérarchique, car le nombre d'états possibles en entrée pris en compte par l'approche CPM primaire est plus grand que celui avec l'approche hiérarchique par propagation. Donc, vraisemblablement, la borne inférieure donnée par l'approche non hiérarchique sera plus pessimiste, car le sous-ensemble d'états de Z contemplés est plus important.

Commentaires

De manière rigoureuse, la fiabilité des blocs combinatoires n'est pas une fonction strictement croissante avec la fiabilité des entrées, même si ça reste vrai pour un spectre assez large de conditions. De ce fait, l'analyse de la borne inférieure exacte ne peut pas être faite à partir de la simple propagation du pire cas. Pour l'exemple de la figure 4.19 cela consisterait en la propagation de l'état de A qui a une plus faible fiabilité. D'ailleurs, cela constituerait une solution triviale, car pour une entrée où tous les états sont incorrects la fiabilité vaut 0, ce qui se traduirait par une sortie avec une fiabilité nulle aussi. Par conséquent, il convient de réaliser une analyse exhaustive garantisant que tous les cas possibles sont parcourus et donc, que la borne inférieure trouvée est exacte. Ce calcul devient très coûteux lorsque le nombre d'entrées du circuits augmente, car le nombre d'états augmente de manière exponentielle.

Les simplifications et hypothèses que nous avons proposées pour l'analyse FMDEA basée sur l'approche CPM primaire peuvent être appliquées à l'analyse basée sur le CPM modulaire, afin de réduire la complexité du calcul et aussi trouver une borne inférieure non stricte moins pessimiste.

L'application de cette approche est aussi limitée à des topologies du circuit n'acceptant pas un découpage en modules disjoints. De ce fait, l'application de l'analyse FMDEA basée sur la méthode CPM hierarchique non disjointe s'impose.

4.3.2.3 Borne inférieure exacte de la fiabilité en utilisant l'approche CPM hiérarchique non disjointe

Le principe de l'analyse FMDEA basée sur l'approche CPM hiérarchique non disjointe est similaire à celui de l'analyse hiérarchique disjointe. Considérons l'exemple de la figure 4.20, consistant en trois blocs combinatoires, dont l'ensemble de signaux B est partagé par les blocs 1 et 2.

Nous devons propager l'espace de distributions A et B jusqu'aux signaux Z en gardant le conditionnement par B, comme montré sur la figure 4.21. De cette manière, nous avons

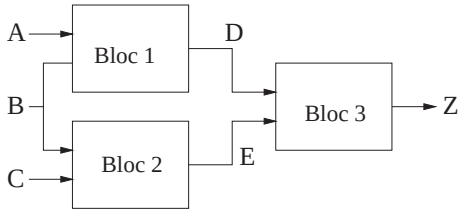


FIG. 4.20 – Topologie de blocs non disjoints utilisée pour illustrer l'application au calcul de la borne inférieure à partir de l'approche CPM hiérarchique non disjoint.

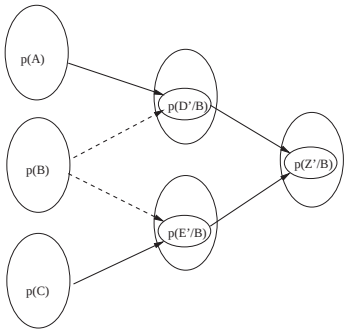


FIG. 4.21 – Diagramme de la propagation de l'espace de probabilités pour l'approche CPM hiérarchique disjointe.

l'espace des probabilités de Z conditionnées par B. Pour trouver la borne inférieure de fiabilité de Z nous devons chercher la distribution de B minimisant la fiabilité de Z.

Commentaires

Les réductions et éventuelles hypothèses proposées dans les sections précédentes peuvent aussi être appliquées pour l'analyse basée sur l'approche hiérarchique non disjointe afin de réduire la complexité du calcul et d'obtenir une borne moins pessimiste.

4.3.3 Résultats

Nous présentons les résultats obtenus pour deux types d'analyses :
- Borne inférieure de la fiabilité en supposant des entrées correctes
- Taux de masquage d'une faute ayant lieu sur une entrée ou sur une porte pour une distribution donnée des entrées.

Pour l'analyse de la borne inférieure de fiabilité, nous avons considéré que toutes les portes du circuit ont la même probabilité d'erreur, $p = 0.001$. Nous avons considéré que les entrées du circuit sont parfaites, c'est-à-dire sans erreur. Ceci comporte l'étude de 2^{65} états différents pour les entrées, ce qui est irréalisable sans utiliser seulement la propagation des colonnes de matrices CPM qui se répètent. L'éventuelle considération des états fautifs des entrées comporte l'étude de 4^{65} états différents, ce qui est impossible à réaliser en raison du trop grand nombre d'états à considérer. Le calcul de la borne inférieure en considérant seulement des entrées correctes a pris $2031s$, en utilisant la plateforme MATLAB sur une machine de 4Ghz..

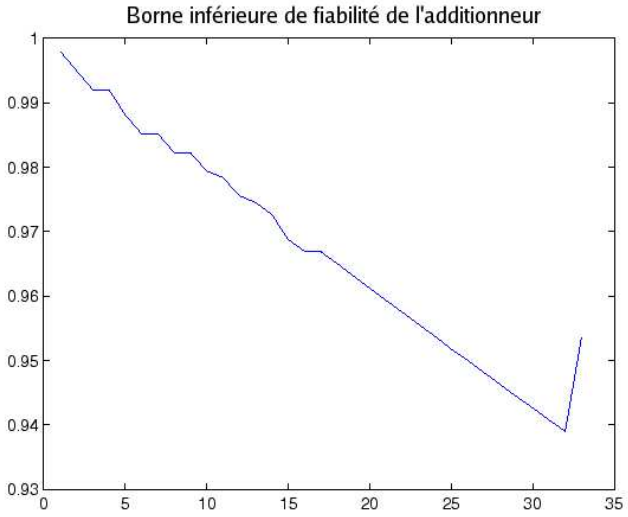


Fig. 4.22 – Borne inférieure de fiabilité pour toutes les sorties de l'additionneur.

Sur la figure 4.22 nous avons représenté la borne inférieure de fiabilité obtenue.

Nous voyons que la tendance générale de la borne inférieure est de décroître lorsque les sorties du circuit ont une profondeur logique plus importante. Ceci s'explique par le fait que plus il y a de signaux qui interviennent sur l'état d'un signal de sortie, plus le nombre de fautes pouvant provoquer une erreur sur cette sortie est important. Or, nous observons un pic sur la dernière sortie. La dernière porte pour cette sortie, à la différence du reste des sorties, n'est pas une porte de type XNOR à deux entrées. En effet, les portes de type XNOR ont un facteur de masquage très faible.

Considérons la table de verité de la porte logique XNOR, montrée en table 4.1.

A	B	Z
0	0	1
0	1	0
1	0	0
1	1	1

Tab. 4.1 – Table de vérité d'une porte XNOR.

Le tableau 4.2 représente les fautes simples sur les entrées du circuit et la conséquence : changement de l'Őtat de sortie de cette porte (colonne Z').

AB	Etat des entrées modifié	Z	Z'
00	0**1**, 1**0**	1	0
01	0**0**, 1**1**	0	1
10	1**1**, 0**0**	0	1
11	1**0**, 0**1**	1	0

Tab. 4.2 – Table de vérité en prenant compte des fautes pour une porte XNOR.

Nous remarquons que l'occurrence d'une faute simple en entrée de cette porte ne peut pas être masquée. Pour qu'un masquage ait lieu, il faudrait avoir une faute sur les deux entrées ou une faute sur la porte elle-même.

L'occurrence d'une double faute est moins probable que l'occurrence d'une faute simple, et aussi, l'occurrence d'une faute sur la porte est moins probable qu'un fonctionnement correct de celle-ci. De ce fait, l'occurrence des mécanismes de masquage est peu probable pour la porte XNOR. Ainsi, la sortie 33 n'étant pas définie par une porte XNOR mais par un autre type (AND-OR à 3 entrées), elle offre un taux de masquage qui explique le pic de la courbe de la borne inférieure de fiabilité. Ce type d'information peut aider à changer la conception du circuit de manière à favoriser une construction intrinsèquement robuste de celui-ci.

Sur la figure 4.23 nous avons représenté une comparaison de la borne inférieure de fiabilité (en vert sur la figure) avec la valeur de fiabilité en considérant une distribution équiprobable et sans faute pour les entrées (en bleu sur la figure).

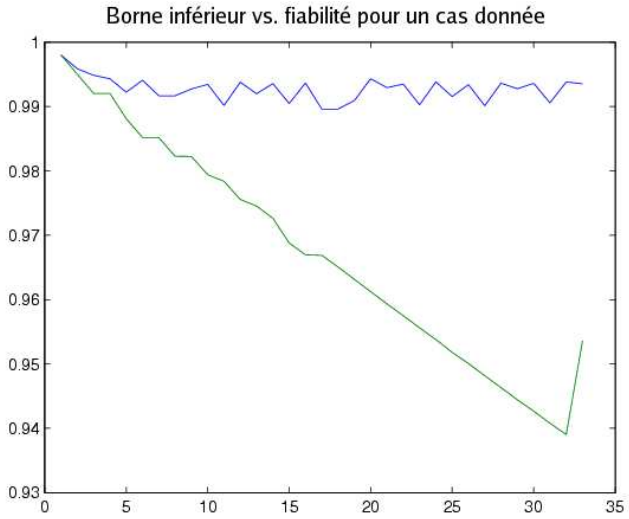


FIG. 4.23 – Borne inférieure de fiabilité et fiabilité pour une distribution donnée pour toutes les sorties de l'additionneur.

La fiabilité obtenue ne suit pas la même tendance décroissante que la borne inférieure, car pour une distribution donnée nous obtenons un résultat moyen. Or la borne inférieure résulte d'un cas extrême, caractérisé par beaucoup de fautes et peu de masquages.

L'autre facteur analysé est le taux de masquage d'une éventuelle faute sur une des portes ou une des entrées du circuit. Cette analyse est réalisée de deux manières selon que la faute a lieu sur une porte ou sur un entrée :

- Pour modéliser une faute simple sur une entrée nous avons mis dans la matrice SPR qui la caractérise les valeurs de cette entrée : $SPR_{I_{faulty}} = \begin{pmatrix} 0 & 0.5 \\ 0.5 & 0 \end{pmatrix}$.Toutes les autres entrées sont considérées équiprobables et sans faute, c'est-à-dire $SPR_{I_{golden}} = \begin{pmatrix} 0.5 & 0 \\ 0 & 0.5 \end{pmatrix}$. Les portes du circuit sont considérées idéales, donc $PTM = ITM$.
- Pour modéliser une faute sur une porte nous avons considéré que la matrice PTM qui

la caractérise contient une probabilité d'erreur égale à l'unité, c'est-à-dire $PTM = \overline{ITM}$. Toutes les autres portes sont considérées idéales $PTM = ITM$.

Les résultats obtenus dans la section précédante concernant la sensibilité des portes sur la fiabilité du circuit ont été présentés sous forme de matrice. Nous pouvons utiliser cette même formulation. Cependant, l'interprétation des informations données par cette matrice est sensiblement différente pour l'analyse du taux de masquage. Il est compliqué de quantifier le taux de masquage d'une faute, car en général, les états de plusieurs sorties du circuit dépendent de la propagation ou du masquage de la faute. Ainsi, nous ne pouvons pas donner une mesure du taux de masquage qui considère toutes les sorties en même temps. Par exemple, considérons une faute qui est masquée pour toutes les sorties sauf une pour laquelle elle est toujours propagée. En moyenne, son taux de masquage sera très élevé car cette faute n'a pas d'effet sur la plupart des sorties, or il s'agit d'une faute qui devient toujours visible, car elle est toujours propagée jusqu'à une sortie.

Nous avons représenté sur la figure 4.24, la fiabilité obtenue pour toutes les sorties lorsqu'une faute apparaît dans chacune des 4 premières portes du circuit. Lorsque la fiabilité est 0, la faute est transmise, lorsque la fiabilité est 1, la faute est masquée. Les valeurs intermédiaires représentent les probabilités de masquage.

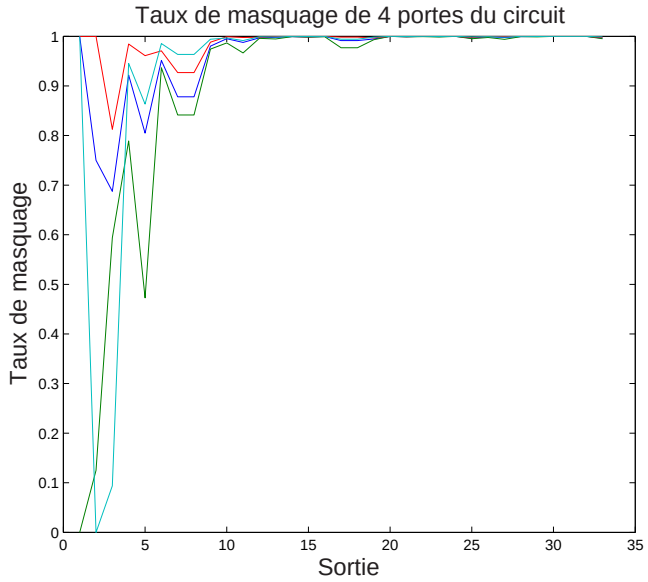


Fig. 4.24 – Taux de masquage sur les 4 premières portes du circuit.

Ces fautes ont un fort impact pour les 10 premières sorties, mais sont presque invisibles pour le reste. Pour évaluer l'impact réel d'une faute, il faut complémenter l'analyse du taux de masquage avec une analyse fonctionnelle. Pour un additionneur, les erreurs avec un impact plus fort sont celles qui modifient l'état de la sortie du bit d'information le plus significatif (sortie 33). Sur la figure 4.25, nous avons représenté les taux de masquages de toutes les fautes simples (de multiplicté 1) possibles (240 portes + 65 entrées) sur les premières et dernières sorties.

Nous voyons que pour la sortie 33, seule 1 faute est critique (taux de masquage 0). Ceci est un gain important pour l'application de la norme ISO26262, qui spécifie que lorsque l'on ne peut pas démontrer le contraire, toutes les fautes doivent être considérées critiques.

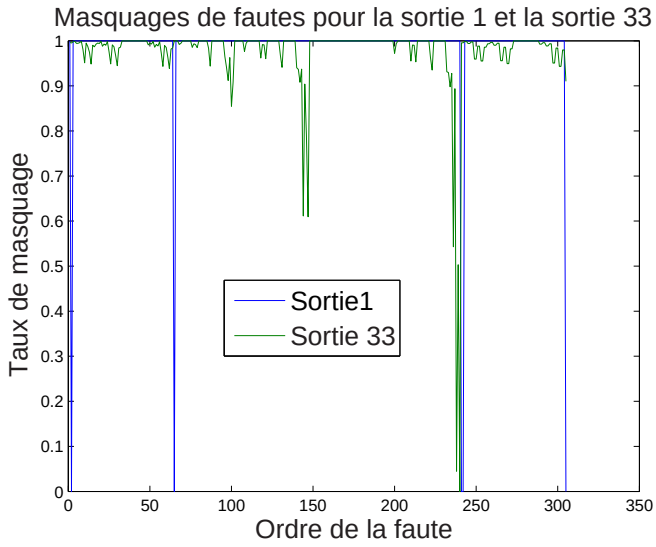


FIG. 4.25 – Taux de masquage de toutes les fautes sur les sorties 1 et 33.

L'information fournie par notre analyse permet d'épargner des mesures de durcissement qui ne seraient pas efficaces, c'est-à-dire, qu'elles apporteraient un gain en fiabilité très réduit par rapport à leur coût.

A partir de cette figure, nous pouvons aussi réaliser une comparaison du comportement des masquages pour les deux sorties. Pour la première, S1, il y a très peu de fautes qui impactent son comportement, mais elles ont un impact très fort dès leur apparition. Ceci est dû au fait que cette sortie dépend de peu de portes. La sortie 33 est impactée par un grand nombre de fautes, mais leur impact est en général faible, car cette sortie dépend d'un grand nombre de portes, et donc les possibilités de masquage sont élevées.

4.4 Conclusions

Nous avons présenté différents types d'analyse basés sur la méthode CPM. Ces analyses enrichissent l'information fournie au concepteur de manière à ce qu'il ait une meilleure vue d'ensemble des aspects concernant la fiabilité, l'aidant ainsi à réaliser des choix plus judicieux pour un éventuel durcissement. Aussi, nous avons montré la versatilité de l'approche CPM, qui est déclinable en différentes configurations et également adaptable au cadre de la norme ISO26262.

Chapitre 5

Validation expérimentale

5.1 Introduction

Dans ce chapitre nous nous consacrons à l'application pratique des modèles théoriques présentés dans le chapitre 3. Dans un premier temps nous présentons les performances obtenues pour l'application des modèles CPM, CPM hiérarchique et SPRMP sur plusieurs circuits. Cette partie a pour objectif la validation expérimentale des prédictions de réduction du temps de calcul réalisées dans la partie théorique de ce manuscrit. Dans la deuxième partie de ce chapitre nous présentons l'outil de calcul de fiabilité développé au cours de cette thèse. Dans l'état actuel de l'outil, le modèle CPM hiérarchique n'est pas intégré. Par conséquent nous n'avons pas pu l'utiliser pour étudier et comparer les différentes approches. Cependant, nous considérons intéressant d'en faire la présentation car, même si le modèle CPM hiérarchique n'est pas disponible, l'outil a un niveau de maturité et de complexité qui démontrent le souhait d'aller au-delà du développement théorique de modèles pour rendre ce type d'approche industrialisable.

5.2 Analyse des performances

Nous avons utilisé la plateforme MATLAB pour évaluer les performances de différents circuits. MATLAB est une plateforme pour laquelle il est difficile d'obtenir un outil paramétrable, c'est-à-dire qui réalise les analyses de manière automatique pour une description du circuit donnée. Il s'agit donc d'une plateforme peu adaptée pour une application industrielle de ce type d'analyse. D'un autre côté, cette plateforme est très adaptable et permet l'application des méthodes proposées tout en évitant des processus de développement logiciel qui permettraient l'obtention de résultats. Cependant, son utilisation nécessite une charge de travail additionnelle car les descriptions structurelles du circuit sous analyse ainsi que les différents calculs à réaliser doivent être introduits manuellement. Dans l'annexe A nous avons détaillé l'analyse d'un exemple avec la plateforme MATLAB pour illustrer la méthodologie mise en place pour l'obtention des résultats de cette section.

Pour comparer les performances des approches proposées nous avons choisi 5 circuits :

- 2 circuits correspondant au benchmark *LGsynth91*, synthetisés sur des bibliothèques de standard celles développées par ST :
  - *mux*.
  - *z4m1*.
- 2 circuits correspondant au benchmark *74x* :

- *74183* : Additionneur rapide de 4 bits.
- *71281* : ALU de 4 bits.

– Additionneur 32 bits de type Brent Kung

Les analyses ont été faites en considérant une distribution équiprobable (uniforme) et sans faute pour les entrées, et une probabilité d'erreur des portes $p = 0.001$.

Analyse du circuit *mux* La fonction logique du circuit *mux* a été prise dans le benchmark *LGSynth91*. Le circuit a 23 entrées, 1 sortie et 35 sources de reconvergence. La structure de ce circuit est représentée sur la figure 5.1. Ce diagramme de blocs a été utilisé pour appliquer l'approche CPM hiérarchique (H-CPM).

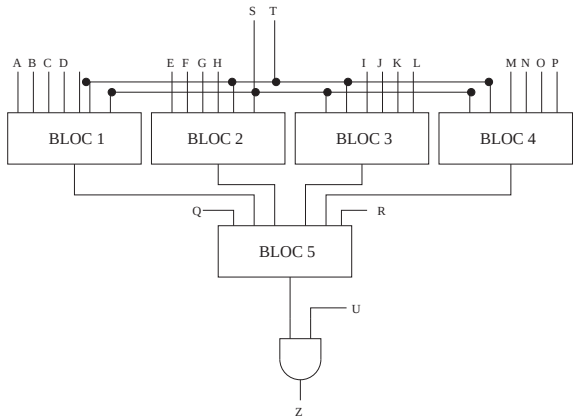


FIG. 5.1 – Diagramme de blocs du circuit *mux*.

Les performances obtenues pour les approches SPRMP, CPM, et CPM hiérarchique sont résumées dans le tableau 5.1.

Approche	Fiabilité	Temps de calcul (s)
$SPRMP$	0.9836	235
CPM	0.9836	155
$H-CPM$	0.9836	19,8

TAB. 5.1 – Performances pour le circuit *mux*.

Malgré le nombre élevé de sources de reconvergences, SPRMP est capable d'éxécuter l'estimation en un temps raisonable car la plupart des sources ne reconverge pas dans le même puits. En effet, seuls les signaux S et T sont communs à tous les blocs. Cette configuration particulière évite le problème majeur de SPRMP, c'est-à-dire, l'accumulation de chemins de reconvergences. Quant à l'approche CPM, elle offre un traitement plus efficace des corrélations, mais le temps requis est du même ordre que pour SPRMP. L'approche CPM hiérarchique traite les blocs séparément, ce qui permet une réduction importante de la complexité, comme c'est mis en évidence par le résultat obtenu.

Analyse du circuit *z4ml* Pour le circuit *z4ml*, la décomposition en blocs proposée pour l'approche CPM hiérarchique est représentée sur la figure 5.2. Ce circuit, de taille plus petite (7 entrées, 4 sorties) que le reste des circuits analysés dans ce chapitre, montre une

performance pour l'approche CPM hiérarchique comparable aux autres circuits (tableau 5.2).

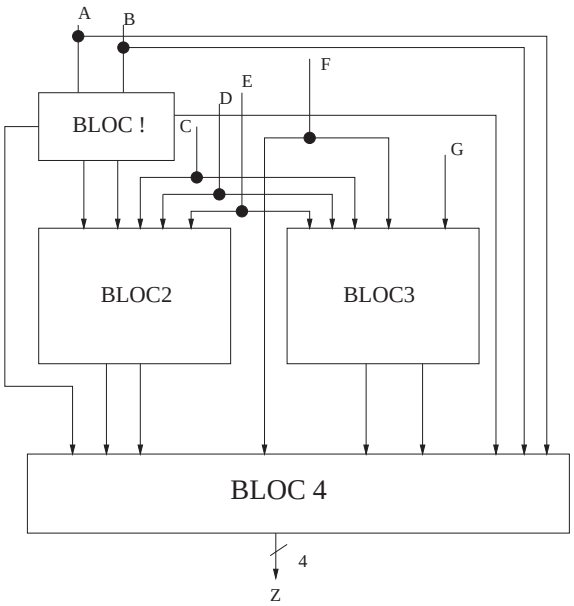


FIG. 5.2 – Diagramme de blocs du circuit *z4m1*.

La configuration de ce circuit comportant un nombre relativement élevé de sources de reconvergence et peu de chemins de reconvergence en forme de cône (chemins qui évoluent séparément les uns des autres) nous a empêchés de trouver un découpage plus disjoint (un nombre plus réduit de conditionnements à garder), ce qui a une pénalisation en termes de coût, car la réduction de complexité est moindre. Cependant, ce cas est intéressant car, malgré le découpage non disjoint et le nombre relativement élevé de sources de reconvergence communes aux différents blocs, nous pouvons le traiter avec un coût raisonnable en faisant un regroupement de sources de manière séquentielle, évitant ainsi l'accumulation d'un grand nombre de conditionnements à traiter en une seule fois. Par exemple, pour les blocs 4 et 3, le conditionnement par les signaux C,D,E et F est résolu en premier, séparément des autres. Postérieurement, le reste des conditionnements pour tous les signaux d'entrée au bloc 4 est résolu, évitant l'explosion combinatoire d'un grand nombre d'états à traiter.

Approche	Fiabilité	Temps de calcul (s)
$SPRMP$	0.9765	753
CPM	0.9765	535
$H-CPM$	0.9765	45

TAB. 5.2 – Performances pour le circuit *z4m1*.

Analyse du circuit *74283* Le troisième circuit analysé est le 74283, composé de 9 entrées et 5 sorties. Le découpage en blocs proposé pour ce circuit est montré sur la figure 5.3. Les performances obtenues sont montrées dans le tableau 5.3.

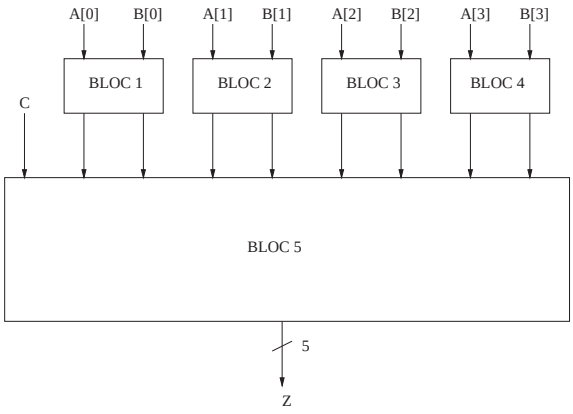


Fig. 5.3 – Diagramme de blocs du circuit *74283*.

Approche	Fiabilité	Temps de calcul (s)
$SPRMP$	0.9912	980
CPM	0.9912	143
$H-CPM$	0.9912	84

Tab. 5.3 – Performances pour le circuit *74283*.

La structure de ce circuit présente une forte présence de masquage de source (c.f. chapitre 3), ce qui permet à CPM d'offrir un gain important par rapport à SPRMP. L'approche CPM hiérarchique est toujours la plus performante. Même s'il s'agit d'un découpage disjoint, la taille du bloc 5 est très importante (presque la moitié du circuit). La complexité du traitement de ce bloc empêche une réduction plus importante de la complexité comparée à d'autres cas cas où le CPM hiérarchique disjoint est applicable (par exemple, le circuit Brent Kung 32 bits analysé plus loin dans cette section).

Analyse du circuit *74181* Le circuit suivant que nous avons analysé est le *74181*, une ALU de 4 bits. Il s'agit d'un circuit à 14 entrées et 8 sorties. Le découpage proposé se trouve sur la figure 5.4. Sur le tableau 5.4 nous avons reporté les performances des différentes approches.

Pour ce cas précis, l'approche CPM hiérarchique offre un gain important par rapport aux approches CPM et SPRMP car il s'agit d'un circuit avec de nombreuses sources de reconvergence, et le fait de pouvoir les traiter séparément est la clé pour obtenir un coût raisonnable. De même que pour le circuit *z4m1*, l'évaluation des sorties de certains blocs a été réalisée séquentiellement. Nous avons appliqué ici ce principe, réduisant davantage le coût de calcul. La structure du *74181* est assez similaire de celle du *74283*, les deux ayant un grand bloc contenant les sorties, ce qui se reflète également par le temps de calcul assez

Approche	Fiabilité	Temps de calcul (s)
$SPRMP$	0.9879	14057
CPM	0.9879	10034
$H-CPM$	0.9879	77

Tab. 5.4 – Performances pour le circuit *74181*.

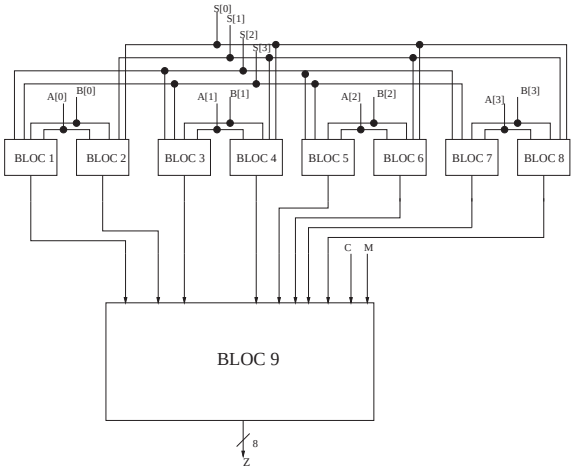


FIG. 5.4 – Diagramme de blocs du circuit *74181*.

similaire pour les deux cas.

Analyse du circuit *Brent kung 32 bits* Finalement, nous montrons les performances obtenues pour l'additionneur brent kung de 32 bits. Sur la figure 5.5, nous avons représenté le découpage en blocs seulement jusqu'à la cinquième sortie, car le circuit est trop grand pour pouvoir représenter le diagramme de blocs sur une seule figure.

Avec les approches SPRMP et CPM nous n'avons pu obtenir la fiabilité que des 9 premières sorties, le temps de calcul requis pour l'obtention des sorties postérieures étant trop élevé pour être traitées par ces approches. Sur le tableau 5.5 nous présentons les performances seulement jusqu'à la sortie 9.

Approche	Fiabilité	Temps de calcul (s)
$SPRMP$	0.9935	18207
CPM	0.9935	6045
$H-CPM$	0.9935	0.7

TAB. 5.5 – Performances pour l'additionneur Brent Kung 32 bits.

Le gain obtenu par l'approche CPM hiérarchique est spécialement remarquable pour ce cas. En effet, la structure de ce circuit permet d'appliquer l'approche disjointe, ce qui diminue le nombre de conditionnements à calculer pour les blocs, combiné à la réalisation d'un découpage consistant en un grand nombre de blocs de petite taille, permettant de traiter chaque bloc avec un coût très réduit, forment une combinaison de facteurs très favorables pour le CPM hiérarchique malgré la taille et le grand nombre de chemins de reconvergence du circuit.

Avec l'approche CPM hiérarchique nous avons pu traiter tout l'additionneur, ce qui surpasse toutes les approches exactes existantes. La structure de ce circuit a permis de réaliser une mesure du temps de calcul sortie par sortie, représentée sur la figure 5.6. Nous pouvons donc observer que la croissance du temps de calcul est linéaire. En effet, ce circuit est découpé en un grand nombre de blocs, dont la complexite de traitement est beaucoup plus faible que la complexité de traitement du circuit entier. Chaque bloc du circuit a une complexité exponentielle. En faisant une simplification, considérons que le circuit consiste

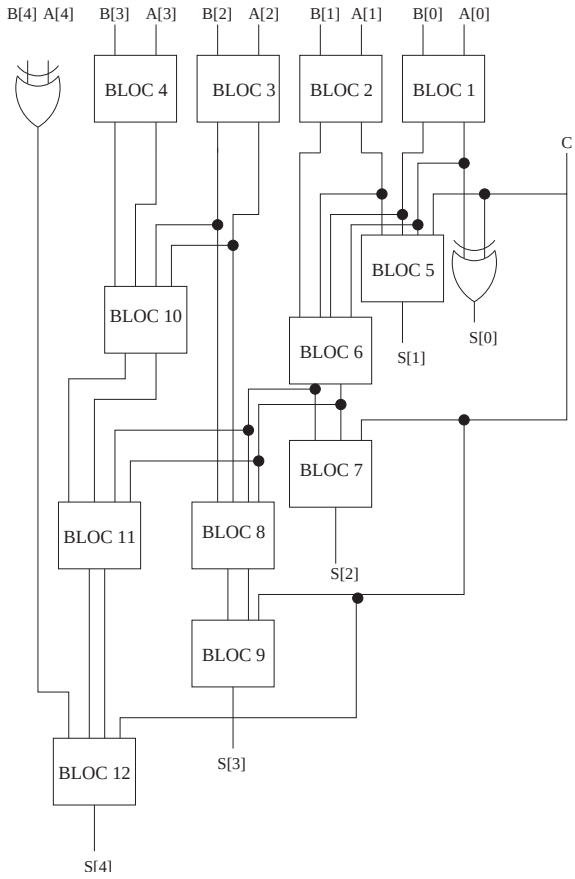


Fig. 5.5 – Diagramme de blocs du circuit *Brent Kung 32B*.

Circuit	# I/O	# portes	# Reconvergences	t (H-CPM)	t (CPM)	t (SPRMP)
mux	23/1	30	35	20 s.	155 s.	235 s.
74283	9/5	35	15	89	143	980
74181	14/8	62	26	75	10034	14057
z4m1	7/4	31	19	45	535	753
BK32	65/33	240	265	3.2	N.A.	N.A

TAB. 5.6 – Tableau comparatif des performances pour tous les exemples traités.

en M blocs de complexité 4^N chacun, et que $M >> 4^N$ (N petit). Ainsi, si l'on considère que $4^N = k$ est une constante, la complexité pour traiter le circuit avec M blocs devient kM.

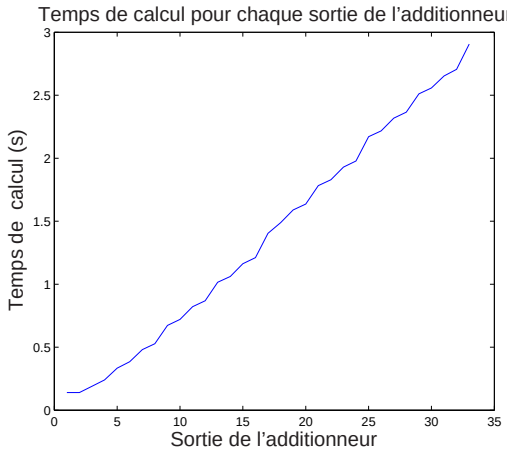


FIG. 5.6 – Temps de calcul nécessaire pour chaque sortie de l'additionneur.

Finalement, le tableau 5.6 résume les performances obtenues pour chacun des circuits.

5.3 Commentaires

A partir de l'étude des performances des circuits analysés nous pouvons confirmer les prédictions faites dans la section théorique (c.f. chapitre 3). En effet, la conclusion génerale que nous pouvons faire est :

- L'approche CPM propose une manière plus efficace de traiter les signaux reconvergents, sans pour autant offrir une réduction de la complexité du problème.
- L'approche CPM hiérarchique parvient à une réduction de la complexité du problème d'estimation de la probabilité jointe de signaux reconvergents.

Les performances de l'approche CPM hiérarchique surclassent celles offertes par le CPM. Cependant, l'approche CPM a permis d'introduire un nouveau traitement pour les signaux reconvergents par les probabilités conditionnées des états des signaux. Même si son utilisation avec des circuits complexes reste limitée car le gain offert dépend fortement de la topologie du circuit analysé et (car il n'y a pas de réduction de la complexité intrinsèque du problème) , elle est nécessaire pour le traitement des blocs individuellement lors de l'application du CPM hiérarchique. Donc, malgré le fait que CPM ait été conçue dans un

premier temps comme une technique d'analyse de fiabilité à part entière, nous considérons que son utilité principale devrait être le traitement des blocs pour le CPM hiérarchique.

Quant au CPM hiérarchique, il s'agit d'une approche qui introduit des nouvelles heuristiques pour l'estimation de la probabilité des signaux et dont le gain offert dépend de la possibilité d'effectuer un découpage du circuit plus ou moins favorable. Malgré ce gain variable, ses performances ont montré au moins une décade de différence par rapport au SPRMP, permettant d'élargir le taille des circuits qui peuvent être traités avec précision par les modèles analytiques.

5.4 Développement d'un outil de prédiction de la fiabilité

L'application des modèles analytiques d'estimation de fiabilité nécessite une plateforme d'exécution. Une part des travaux d'application développés au cours de cette thèse a consisté en l'implémentation d'un outil de calcul. Dans cette section, nous présentons une brève description de cet outil.

Sur la figure 5.7 nous avons représenté le diagramme de l'outil. Dans l'état actuel, l'outil compte 30 K lignes de code C++. Par la suite nous détaillons les caractéristiques de chaque partie du diagramme.

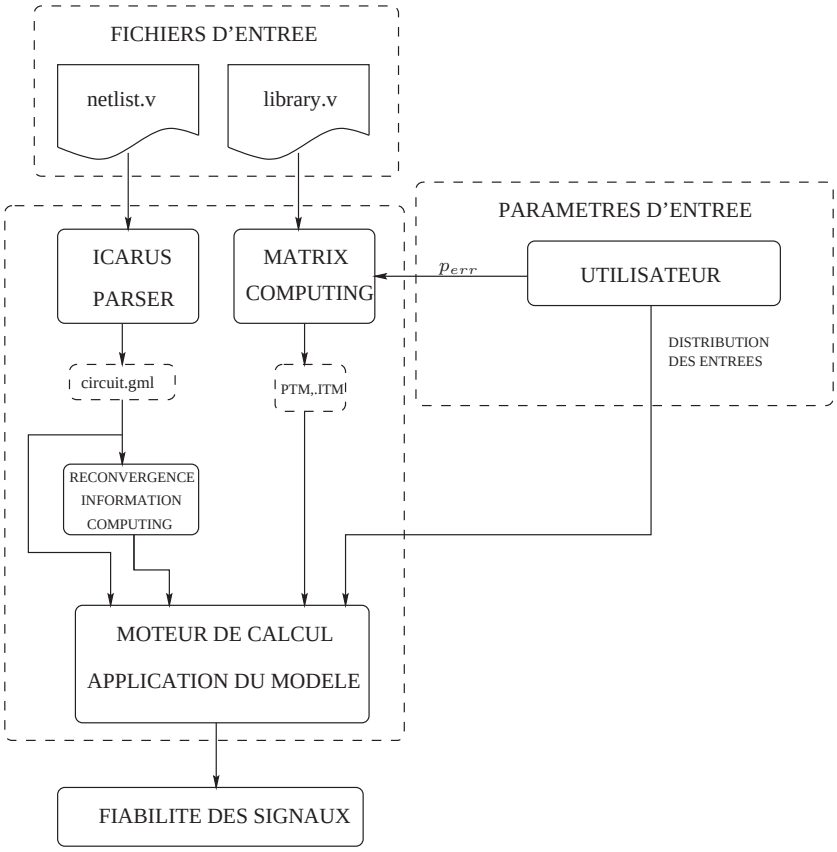


FIG. 5.7 – Diagramme de blocs de l'outil de calcul de fiabilité.

Les paramètres d'entrée qui doivent être fournis sont :

– **netlist.v** Il s'agit d'une description du circuit en format *netlist* portes en *verilog*.

- **library.v** Les librairies sur lesquelles la netlist du circuit est synthétisée. Elle contient la description fonctionnelle des portes utilisées dans le circuit. A partir de cette description nous pouvons déterminer les matrices PTM et ITM, combinées avec leur probabilité d'erreur qui est, quant à elle, un autre paramètre d'entrée.
- **Probabilité d'erreur des portes** Ce paramètre est introduit manuellement dans l'état actuel de l'outil, mais pour l'obtention de résultats plus réalistes, l'obtention des vraies probabilités d'erreur des éléments de la libraire doit être envisagée.
- **Distribution des entrées** Ce paramètre correspond à la matrice SPR caractérisant chacun des signaux d'entrée du circuit. De même que pour les probabilités d'erreur des portes, un module réalisant l'estimation doit être intégré afin d'obtenir des résultats plus réalistes.

Les modules qui forment le coeur de l'outil sont :

- **Parser Icarus** Il s'agit d'un parser verilog qui fournit une description du circuit sous forme de graphe à partir du fichier verilog du circuit. Ce graphe est utilisé par le moteur de calcul lors de l'application de la méthode d'estimation choisie.
- **Matrix Computing** Ce bloc calcule les matrices PTM et ITM des portes logiques du circuit à partir de la description fonctionnelle des portes contenues dans le fichier *library.v* et de la probabilité d'erreur fournie comme paramètre d'entrée.
- **Reconvergence Information Computing** Ce bloc utilise la description sous forme de graphe du circuit pour calculer toutes les informations nécessaires permettant d'appliquer les méthodes d'estimation de fiabilité : les signaux sources de reconvergence, les puits de reconvergence, les conditionnements subis par chaque signal sont identifiés. Il utilise des algorithmes issus de la théorie de graphes pour ce faire.
- **Moteur de calcul - Application du modèle** Ce module est le coeur de l'outil. Il s'agit d'un moteur de calcul qui réalise les opérations correspondantes (propagation des matrices SPR, conditionnement de signaux, application du poids, etc.) de chaque modèle afin d'obtenir l'estimation de la fiabilité des sorties. En l'état actuel, les modèles de calcul disponibles sont SPR, SPRMP et CPM. Le modèle CPM hiérarchique n'a pas pu être implémenté. Pour cette raison, nous n'avons pas pu utiliser l'outil pour la comparaison des performances des méthodes.

Finalement, les paramètres de sortie de l'outil sont les matrices SPR caratérisant les signaux, à partir desquelles nous pouvons immédiatement obtenir la fiabilité de chacun des signaux, avec un intérêt spécial, évidemment, pour les signaux de sortie.

L'objectif du développement de cet outil est son application industrielle. Pour ce faire, il est nécessaire aussi d'intégrer les éléments de la logique séquentielle dans les modèles analytiques. Dans l'annexe B nous présentons une première proposition de méthodologie pour ce faire.

Chapitre 6

Conclusion

Cette thèse a porté sur l'étude de la fiabilité des circuits numériques. La problématique de l'estimation de la fiabilité dans les blocs de logique combinatoire a été l'axe central des travaux développés. Plusieurs aspects liés à cette problématique ont été traités : le développement de modèles d'estimation, l'identification des noeuds critiques d'un circuit, l'analyse FMDEA et l'estimation des bornes inférieures de la fiabilité. Nous présentons ici les principales contributions des recherches réalisées ainsi que les perspectives pour les travaux qui continueront cette thèse.

Contributions

Cette thèse contribue au développement des modèles analytiques d'estimation de la fiabilité dans la logique combinatoire en proposant une approche novatrice pour la réduction de la complexité du calcul tout en ayant un maximum de précision possible. Le gain dans le rapport qualité/coût de ce type d'analyse a été l'axe principal de recherche.

Dans un premier temps les limitations de ce type d'approche ont eté mises en évidence à partir d'une étude approfondie de la littérature sur le sujet. L'estimation des corrélations entre les signaux, provoquées par les signaux reconvergents, a eté identifiée comme l'obstacle principal pour l'obtention de modèles avec une bonne précision et un coût réduit.

Les travaux de recherche ont tout d'abord porté sur l'obtention d'une méthodologie permettant le traitement de signaux reconvergents de la manière la plus efficace possible. Une première solution a été proposée : le modèle CPM, qui utilise une représentation du comportement aléatoire et fautif des portes et des signaux déjà existante dans la littérature. L'approche CPM propose une nouvelle technique basée sur une formulation matricielle qui réduit le nombre d'états à propager pour l'obtention de la valeur de la fiabilité des sorties. La contribution principale de CPM est le traitement plus efficace des signaux reconvergents. Cependant, la réduction apportée par CPM dépend de la topologie du circuit, et aussi, cette approche est limitée par la croissance exponentielle des matrices intermédiaires utilisées. De ce fait, une autre approche, basée sur le CPM, mais introduisant un découpage du circuit a été proposée. Cette approche, plus qu'un traitement plus efficace des signaux reconvergents, offre une réduction de la complexité, ce qui permet de traiter des circuits de plus grande taille. En effet, avec le CPM hiérarchique il est possible de traiter un circuit de 65 entrées et 33 sorties avec une complexité linéaire.

Deuxièmement, les travaux de recherche ont visé la proposition de méthdologies pour l'obtention des différentes métriques qui vont au-delà d'une classification de circuit fiable

ou non fiable. A cet effet, nous avons proposé une méthodologie d'obtention du taux de masquage des fautes simples ainsi que de la borne inférieure de fiabilité, visant une application pour la norme ISO26262. De plus, une approche permettant d'identifier les portes du circuit les plus critiques a aussi été proposée. L'identification des noeuds critiques permet de guider le durcissement et minimser les surcoûts liés à l'ajout de redondance.

Perspectives

Les travaux présentés dans ce manuscrit laissent de la place pour des améliorations possibles. Du point de vue théorique plusieurs recherches doivent être menées. Notamment, l'approche CPM hiérarchique s'est avérée utile pour l'analyse de circuits de taille importante, cependant, le découpage réalisé est fait manuellement et *ad-hoc*, ce qui limite son applicabilité à des circuits de taille industrielle. La proposition d'algorithmes d'analyse du graphe du circuit réalisant un découpage optimal est nécessaire pour parvenir à une application industrielle de cette approche.

Une autre amélioration possible de l'approche hiérarchique consisterait en la réduction des matrices CPM utilisées pour l'analyse des blocs. En effet, comme montré dans le chapitre 4, ces matrices sont fréquemment formées des colonnes qui se répètent. L'utilisation de diagrammes de décision algébriques (ADD) pour la compression de matrices devrait contribuer à cette réduction [81].

Du point de vue applicatif, l'industrialisation complète des approches proposées passe par l'intégration du modèle CPM hiérarchique dans l'outil de calcul, de manière à évaluer les circuits automatiquement et intégrer l'outil dans le flot de conception. Une autre amélioration de la précision de résultats consisterait en le raffinement des modèles d'erreur dans les portes. Une information plus détaillée sur leur taux d'erreur pourrait améliorer l'accord entre le modèle et la réalité.

Liste des publications

Publications scientifiques

Journaux

J. Torras Flaquer, J-M. Daveau, L.A.B. Naviner, P. Roche, "Fast reliability analysis of combinatorial logic circuits using conditional probabilities", Microelectronics Reliability Journal, Elsevier, vol. 50, No. 9-11, pp. 1215-1218, 2010.

Conférences

J. Torras Flaquer, J-M. Daveau, L.A.B. Naviner, P. Roche, "An approach to reduce computational cost in combinatorial logic netlist reliability analysis using circuit clustering and conditional probabilities", 17th International On Line Test Symposium (IOLTS) , pp. 98-103, Juillet 2011.

J. Torras Flaquer, J-M. Daveau, L.A.B. Naviner, P. Roche, "Handlind reconvergent paths using conditional probabilities in combinatorial logic netlist reliability estimation", 17th IEEE International Conference on electronics, Circuits and Systems (ICECS), pp. 263-267, 2010.

J. Torras Flaquer, J-M. Daveau, L.A.B. Naviner, P. Roche, "Fast reliability analysis of combinatorial logic circuits using conditional probabilities", 21st European Symposium on Reliability of Electron Devices, Failure Physics and Analysis (ESREF), 2010.

Brevets

J. Torras Flaquer, J-M. Daveau, L.A.B. Naviner, P. Roche, "Method for estimating the reliability of an electronic circuit, corresponding computerized system and computer program product", Publication number 20110246811, Application number 13074204, 2011.

Annexe A

Mise en oeuvre des modèles de fiabilité avec MATLAB

Pour illustrer la procédure utilisée pour l'obtention des résultats nous montrons le code utilisé pour la mise en oeuvre des approches SPRMP, CPM et CPM hiérarchique d'un petit exemple, représenté sur la figure A.1.

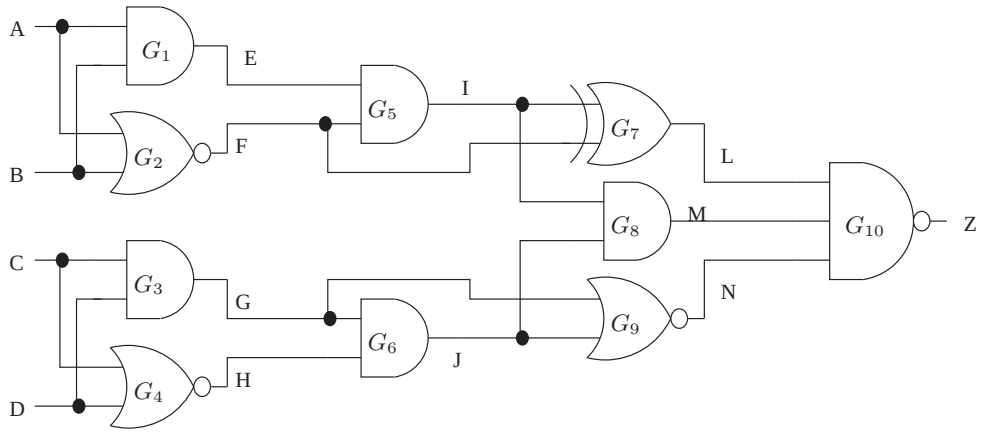


FIG. A.1 – Circuit d'exemple pour illustrer l'utilisation du code MATLAB

Les fonctions utilisées sont les suivantes :

- `function [SPR_out]=SPR(M_input, PTM_gate, ITM_gate)`
  `% calcul de la matrice SPR de sortie d'une porte logique`
  `% M_input est la matrice de distribution des entr\'ees`
  `% PTM_gate, ITM_gate, matrices caracterisant la porte`
  `SPR_out=(ITM.')*M_input*PTM`
- `function [SPR_decorrelated, w]=matrix_decorrelated(SPR_signal,index)`
  `% Retourne la matrice SPR decorrele selon la valeur d'index, et le poids`
  `% correspondant a cette iteration`
  `if(index==1)`
  `SPR_decorrelated=[1 0; 0 0]:`
  `w=SPR_signal(1,1);`
  `elseif(index==2)`

```
      SPR_decorrelated=[0 1; 0 0]:
      w=SPR_signal(1,2);
    elseif(index==3)
      SPR_decorrelated=[0 0; 1 0]:
      w=SPR_signal(2,1);
    elseif(index==4)
      SPR_decorrelated=[0 0; 0 1]:
      w=SPR_signal(2,2);
    end;
```

Le code pour l'approche SPRMP est le suivant :

```
function [SPR_Z_SPRMP]=example_SPRMP(SPR_A, SPR_B, SPR_C, SPR_D,
                                     PTMg1, PTMg2, PTMg3, PTMg4, PTMg5, PTMg6,PTMg7,
                                     ITMg1, ITMg2, ITMg3, ITMg4, ITMg5, ITMg6,ITMg7)

SPR_Z_SPRMP=(1:2,1:2)=0;
for a=1:4
 [SPR_a_aux,wa]=matrix_decorrelated(SPR_A,a);
 for b=1:4
  [SPR_b_aux,wb]=matrix_decorrelated(SPR_B,b);

  SPR_E=SPR(kron(SPR_a_aux, SPR_b_aux), PTMg1,ITMg1);
  SPR_F=SPR(kron(SPR_a_aux, SPR_b_aux), PTMg2,ITMg2);

  for f=1:4
   [SPR_f_aux,wf]=matrix_decorrelated(SPR_F,f);

   SPR_I=SPR(kron(SPR_E, SPR_f_aux), PTMg5,ITMg5);

   for c=1:4
    [SPR_c_aux,wc]=matrix_decorrelated(SPR_C,c);

    for d=1:4
     [SPR_d_aux,wd]=matrix_decorrelated(SPR_D,d);

     SPR_G=SPR(kron(SPR_c_aux, SPR_d_aux), PTMg3,ITMg3);
     SPR_H=SPR(kron(SPR_c_aux, SPR_d_aux), PTMg4,ITMg4);

     for g=1:4
      [SPR_g_aux,wg]=matrix_decorrelated(SPR_G,g);

      SPR_J=SPR(kron(SPR_G, SPR_g_aux), PTMg6,ITMg6);

      for i=1:4
       [SPR_i_aux,wi]=matrix_decorrelated(SPR_i,i);

        SPR_L=SPR(kron(SPR_f_aux, SPR_i_aux), PTMg7,ITMg7);
```

```
       for j=1:4
        [SPR_j_aux,wj]=matrix_decorrelated(SPR_J,j);

        SPR_M=SPR(kron(SPR_i_aux, SPR_j_aux), PTMg8,ITMg8);
        SPR_N=SPR(kron(SPR_g_aux, SPR_j_aux), PTMg9,ITMg9);

        SPR_Z_aux=SPR(kron(SPR_L, kron(SPR_M, SPR_N)), PTMg10,ITMg10);

        SPR_Z_sprmp=SPR_ZSPRMP+SPR_Z_aux*wa*wb*wc*wd*wf*wg*wi*wj;

       end;
      end;
     end;
    end;
   end;
  end;
 end;
end;
```

Le code correspondant à l'approche CPM est le suivant :

```
function [SPR_Z_CPM]=example_CPM(SPR_A, SPR_B, SPR_C, SPR_D,
                                 PTMg1, PTMg2, PTMg3, PTMg4, PTMg5, PTMg6,PTMg7,
                                 ITMg1, ITMg2, ITMg3, ITMg4, ITMg5, ITMg6,ITMg7)

SPR_Z_CPM=(1:2,1:2)=0;
ind=1;
for a=1:4
 [SPR_a_aux]=matrix_decorrelated(SPR_A,a);
 for b=1:4
  [SPR_b_aux]=matrix_decorrelated(SPR_B,b);

  SPR_E=SPR(kron(SPR_a_aux, SPR_b_aux), PTMg1,ITMg1);
  SPR_F=SPR(kron(SPR_a_aux, SPR_b_aux), PTMg2,ITMg2);

  CPM_E(1,ind)=SPR_E(1,1);
  CPM_E(2,ind)=SPR_E(1,2);
  CPM_E(3,ind)=SPR_E(2,1);
  CPM_E(4,ind)=SPR_E(2,2);

  CPM_F(1,ind)=SPR_F(1,1);
  CPM_F(2,ind)=SPR_F(1,2);
  CPM_F(3,ind)=SPR_F(2,1);
  CPM_F(4,ind)=SPR_F(2,2);

  ind=ind+1;
 end;
end;
```

```
for c=1:4
 [SPR_c_aux,wc]=matrix_decorrelated(SPR_C,c);
 for d=1:4
  [SPR_d_aux,wd]=matrix_decorrelated(SPR_D,d);

  SPR_G=SPR(kron(SPR_c_aux, SPR_d_aux), PTMg3,ITMg3);
  SPR_H=SPR(kron(SPR_c_aux, SPR_d_aux), PTMg4,ITMg4);

  CPM_G(1,ind)=SPR_G(1,1);
  CPM_G(2,ind)=SPR_G(1,2);
  CPM_G(3,ind)=SPR_G(2,1);
  CPM_G(4,ind)=SPR_G(2,2);

  CPM_H(1,ind)=SPR_H(1,1);
  CPM_H(2,ind)=SPR_H(1,2);
  CPM_H(3,ind)=SPR_H(2,1);
  CPM_H(4,ind)=SPR_H(2,2);

  ind=ind+1;
 end;
end;

ind=1;
for e=1:16
 SPR_e_aux(1,1)=CPM_E(1,e);
 SPR_e_aux(1,2)=CPM_E(2,e);
 SPR_e_aux(2,1)=CPM_E(3,e);
 SPR_e_aux(2,2)=CPM_E(4,e);

 for f=1:4
  [SPR_f_aux]=matrix_decorrelated(SPR_F,f);

  SPR_I=SPR(kron(SPR_e_aux, SPR_f_aux), PTMg5,ITMg5);
  CPM_I(1,ind)=SPR_I(1,1);
  CPM_I(2,ind)=SPR_I(1,2);
  CPM_I(3,ind)=SPR_I(2,1);
  CPM_I(4,ind)=SPR_I(2,2);

  ind=ind+1;
 end;
end;

ind=1;
for h=1:16
 SPR_h_aux(1,1)=CPM_H(1,e);
 SPR_h_aux(1,2)=CPM_H(2,e);
 SPR_h_aux(2,1)=CPM_H(3,e);
 SPR_h_aux(2,2)=CPM_H(4,e);
```

```
 for g=1:4
  [SPR_g_aux]=matrix_decorrelated(SPR_G,g);

  SPR_J=SPR(kron(SPR_h_aux, SPR_g_aux), PTMg6,ITMg6);
  CPM_J(1,ind)=SPR_J(1,1);
  CPM_J(2,ind)=SPR_J(1,2);
  CPM_J(3,ind)=SPR_J(2,1);
  CPM_J(4,ind)=SPR_J(2,2);

  ind=ind+1;
 end;
end;

for f=1:4
 [SPR_f_aux]=matrix_decorrelated(SPR_F,f);
 for i=1:4
  [SPR_i_aux]=matrix_decorrelated(SPR_I,i);

  SPR_L=SPR(kron(SPR_f_aux,SPR_i_aux), PTMg7,ITMg7);

  for j=1:4
   [SPR_j_aux]=matrix_decorrelated(SPR_J,j);

   SPR_M=SPR(kron(SPR_i_aux,SPR_j_aux), PTMg8,ITMg9);

   for g=1:4
    [SPR_g_aux]=matrix_decorrelated(SPR_g,j);

    SPR_N=SPR(kron(SPR_g_aux,SPR_j_aux), PTMg9,ITMg9);

    % calcul du signal Z en fonction de I,F,G,J
    SPR_Z_aux=SPR(kron(SPR_L,kron(SPR_M,SPR_N)), PTMg10,ITMg10);


    %Calcul du poids de cette iteration
    w=0;
    for a=1:4
      [XX, wa]=matrix_decorrelated(SPR_A,a);
      for b=1:4
        [XX, wb]=matrix_decorrelated(SPR_B,b);
for c=1:4
         [XX, wc]=matrix_decorrelated(SPR_C,c);
 for d=1:4
         [XX, wd]=matrix_decorrelated(SPR_D,d);

 % calcul de la colonne et du poids correspondant pour la matrice CPM_F
 col_f=(a-1)*4+b;
```

```
 wf=CPM_F(f,col_f);

 % calcul de la colonne et du poids correspondant pour la matrice CPM_G
 col_g=(c-1)*4+d;
 wg=CPM_G(f,col_g);

 % calcul de la colonne et du poids correspondant pour la matrice CPM_I
 col_i=(a-1)*16+(h-1)*4+f;
 wi=CPM_I(i,col_i);

 % calcul de la colonne et du poids correspondant pour la matrice CPM_J
 col_j=(c-1)*16+(d-1)*4+g;
 wj=CPM_I(j,col_j);

 w=w+wa*wb*wc*wd*wf*wi*wg*wj;
end;
       end;
      end;
     end;

     SPR_Z_CPM=SPR_Z_CPM+SPR_Z_aux*w;

    end;
   end;
  end;
 end;
```

Pour montrer le code utilisé pour le CPM hiérarchique, nous indiquons d'abord le découpage réalisé, figure A.2.

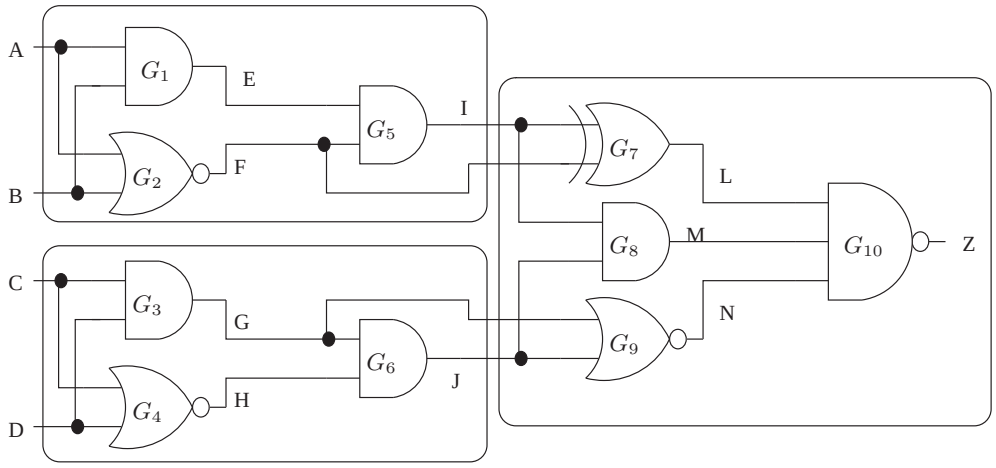


Fig. A.2 – Découpage réalisé pour le circuit d'exemple pour illustrer l'utilisation du code MATLAB

Nous détaillons seulement les calculs pour obtenir les matrices SPR_IF, contenant $p(i_\alpha, f_\beta)$, et SPR_JG, contenant $p(j_\alpha, g_\beta)$, et comment ces matrices s'utilisent pour trouver

SPR_Z. Les matrices CPM_I, CPM_J et CPM_Z se calculent de la même manière que pour l'approche CPM.

Ainsi, le code pour l'approche CPM hiérarchique est :

```
function [SPR_Z_HCPM]=example_HCPM(SPR_A, SPR_B, SPR_C, SPR_D,
                                   PTMg1, PTMg2, PTMg3, PTMg4, PTMg5, PTMg6,PTMg7,
                                   ITMg1, ITMg2, ITMg3, ITMg4, ITMg5, ITMg6,ITMg7)
SPR_Z_HCPM(1:2,1:2)=0;
SPR_IF(1:4,1:4)=0;
SPR_JG(1:4,1:4)=0;

for a=1:4
 [XX, wa]=matrix_decorrelated(SPR_A,a);
 for b=1:4
  [XX, wb]=matrix_decorrelated(SPR_B,b);

  col_j=(a-1)*4+b;
  for f=1:4
   [SPR_f_aux]=matrix_decorrelated(SPR_F,f);
   wf=CPM_F(f,col_f);
   col_i=(a-1)*16+(b-1)*4+f;
   SPR_I(1,1)=CPM_I(1,col_i);
   SPR_I(1,2)=CPM_I(2,col_i);
   SPR_I(2,1)=CPM_I(3,col_i);
   SPR_I(2,2)=CPM_I(4,col_i);


   SPR_IJ=SPR_IJ+kron(SPR_f_aux,SPR_I)*wa*wb*wj;
  end;
 end;
end;

for c=1:4
 [XX, wc]=matrix_decorrelated(SPR_C,c);
 for d=1:4
  [XX, wd]=matrix_decorrelated(SPR_D,d);

  col_j=(c-1)*4+d;
  for g=1:4
   [SPR_g_aux]=matrix_decorrelated(SPR_G,g);
   wg=CPM_g(g,col_g);
   col_j=(c-1)*16+(d-1)*4+g;

   SPR_J(1,1)=CPM_J(1,col_j);
   SPR_J(1,2)=CPM_J(2,col_j);
   SPR_J(2,1)=CPM_J(3,col_j);
   SPR_J(2,2)=CPM_J(4,col_j);
```

```
   SPR_JG=SPR_JG+kron(SPR_j_aux,SPR_G)*wc*wd*wg;
  end;
 end;
end;

for i=1:4
 for f=1:4
 % calcul de la position dans la matrice SPR_IF pour obtenir le poids correspondant
 wif=return_weight(SPR_IF,i,f);

 for j=1:4
  for g=1:4
   % calcul de la position dans la matrice SPR_JG pour obtenir le poids correspondant
   wjg=return_weight(SPR_JG,j,g);

   col_z=(i-1)*64+(f-1)*16+(j-1)*4+g;

   SPR_Z_HCPM(1,1)=SPR_Z_HCPM(1,1)+CPM_Z(1,col_z)*wif*wjg;
   SPR_Z_HCPM(1,2)=SPR_Z_HCPM(1,2)+CPM_Z(2,col_z)*wif*wjg;
   SPR_Z_HCPM(2,1)=SPR_Z_HCPM(2,1)+CPM_Z(3,col_z)*wif*wjg;
   SPR_Z_HCPM(2,2)=SPR_Z_HCPM(2,2)+CPM_Z(4,col_z)*wif*wjg;
  end;
 end;
end;
```

Annexe B

Intégration des éléments de logique séquentielle dans l'outil de calcul

B.1 Méthodologie proposée

Tout au long de ce manuscrit nous avons traité l'estimation de la fiabilité dans la logique combinatoire. La suite logique de ce type d'analyse consiste à intégrer les éléments et type de connexions propres de la logique séquentielle.

Typiquement, un circuit de logique séquentielle est constitué d'une part des entrées et sorties primaires, d'un bloc de logique combinatoire, et d'autre part des entrées et sorties correspondants aux signaux de rebouclage connectés à travers de bascules. Sur la figure B.1 nous avons représenté le diagramme typique de ce type de circuit.

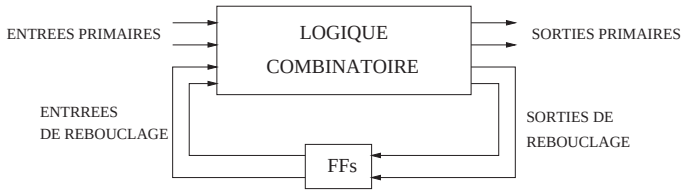


FIG. B.1 – Configuration typique d'un circuit séquentiel.

La méthodologie que nous proposons consiste à réaliser un calcul de point fixe. Ce type de calcul consiste en un processus itératif de calcul jusqu'à l'obtention de résultats convergents. De manière intuitive, nous pouvons considérer les circuits séquentiels comme des suites infinies du même circuit combinatoire, où les entrées de rebouclage du bloc combinatoire sont les sorties de rebouclage du circuit précédent. Pour les entrées de rebouclage du premier bloc combinatoire, nous attribuons une valeur initiale pour leur distribution de probabilité (matrice SPR). Cette modélisation est représentée sur la figure B.2. Pour le restant des blocs, la distribution des sorties de rebouclage est atribuée aux entrées de rebouclage du bloc suivant. Il est évident qu'il n'est pas possible de réaliser un nombre infini d'itérations pour parvenir à la convergence des résultats. Ainsi, nous avons défini un critère de convergence à partir d'un seuil : nous comparons pour chaque bloc chacune des

matrices SPR de toutes ses sorties (primaires et de rebouclage), B.1 :

$$\alpha = \sum_{i=0}^{1} \sum_{j=0}^{1} |SPR_n(i,j) - SPR_{n-1}(i,j)| \tag{B.1}$$



Fig. B.2 – Modélisation d'un circuit séquentiel comme une suite de circuits combinatoires.

Lorsque la somme de tous les paramètres α de toutes les sorties du circuit est inférieure au seuil défini, nous arrêtons le processus itératif et prenons la valeur des matrices SPR des signaux à cette itération comme définitive. Nous avons représenté ce procédé sur la figure B.3.

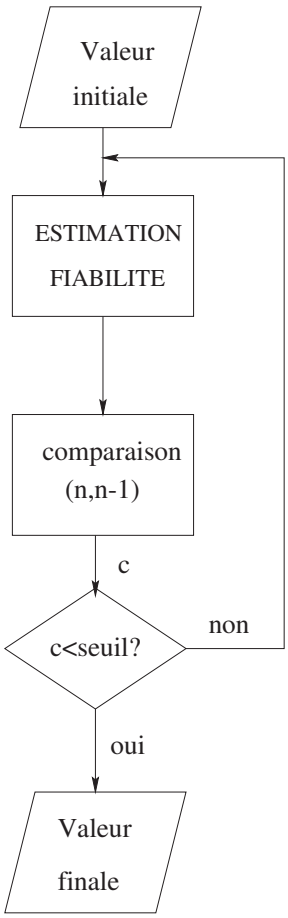


Fig. B.3 – Diagramme du flux de l'algorithme proposé pour l'estimation de la fiabilité des circuits séquentiels.

B.2 Résultats

Afin de valider cette approche, nous avons réalisé cette procédure sur plusieurs exemples. Aussi, afin de vérifier les résultats obtenus nous avons réalisé une simulation de ces mêmes exemples en réalisant une injection de fautes sur les portes de manière probabiliste, avec le même taux d'occurrence que pour le modéle analytique. Avec l'outil, nous avons utilisé le modèle CPM pour l'estimation de la fiabilité.

Nous avons utilisé un filtre médian de 8B, qui a été découpé en plusieurs blocs qui ont été analysés séparément. Dans les tableaux qui suivent nous présentons les fiabilités estimées avec les deux approches, ainsi que l'erreur commise. Pour l'analyse nous avons considéré une probabilité d'erreur dans toutes les portes de 0.05 et une distribution équiprobable des entrées.

Sortie	Simulation VHDL	Approche analytique	Erreur
n_1	0.9499	0.95	0.01 %
n_4	0.7725	0.7630	1.23 %
n_7	0.646	0.6102	5,54 %
n_{11}	0.9273	0.9275	0.02 %
n_{12}	0.8845	0.8847	0.02 %
n_{13}	0.7923	0.7773	2.4 %
n_{14}	0.7626	0.7496	1.2 %
n_{26}	0.863	0.8505	1.4 %
n_{36}	0.9502	0.95	0.02 %
n_{37}	0.9501	0.95	0.01 %
n_{50}	0.791	0.7746	2.1 %
n_{73}	0.917	0.9168	0.22 %
$Etat_0$	0.7931	0.7923	0.09 %

TAB. B.1 – Comparatif pour le bloc A.

Sortie	Simulation VHDL	Approche analytique	Erreur
n_2	0.7763	0.7669	1.2 %
n_{43}	0.951	0.95	0.1 %
n_{45}	0.906	0.905	0.1 %
n_{59}	0.906	0.905	0.1 %
$Etape_{a1_0}$	0.815	0.7956	2,4 %
$Etat_2$	0.8054	0.7893	2.0 %

TAB. B.2 – Comparatif pour le bloc B.

Les résultats montrent une erreur qui n'est pas négligeable pour certains cas. En général, les résultats indiquent que plus le taux d'erreur est important, plus l'erreur commise grandit. A partir de l'observation de l'estimation fournie par les simulations VHDL nous pouvons déduire quelques explications pour l'apparition de l'erreur. Plusieurs des signaux présents dans les tableaux ci-dessus ont une fiabilité très proche de 0.95. En général dans le circuit analysé, ces signaux viennent d'une configuration comme celle montrée dans la figure B.4., c'est-à-dire qu'ils sont la sortie d'un inverseur ou d'un buffer logique. Malgré la simplicité du cas, les simulations en VHDL fournissent un résultat différent de 0.95 (pour une porte avec une seule entrée, avec des entrées idéales, la fiabilité de la sortie est la

Sortie	Simulation VHDL	Approche analytique	Erreur
n_3	0.95	0.95	0.0 %
n_9	0.607	0.5427	11.9 %
n_{17}	0.773	0.7442	3.87 %
n_{21}	0.778	0.7464	4,57 %
n_{24}	0.8552	0.8349	2.43 %
n_{27}	0.846	0.8239	2.7 %
n_{78}	0.951	0.95	0.1 %
n_{81}	0.806	0.7909	1.9 %
n_{94}	0.8941	0.8918	0.3 %
n_{99}	0.9275	0.9275	0.0 %
n_{100}	0.885	0.8847	0.03 %
n_{124}	0.9221	0.9223	0.02 %
n_{139}	0.7874	0.7732	1.8 %
$Etape_{a0_3}$	0.8172	0.8091	1,0 %

Tab. B.3 – Comparatif pour le bloc C.

Sortie	Simulation VHDL	Approche analytique	Erreur
n_{20}	0.8233	0.7981	3.1 %
n_{31}	0.8924	0.8886	0.4 %
n_{32}	0.8522	0.8497	0.2 %
n_{33}	0.883	0.8823	0.07 %
n_{89}	0.904	0.9036	0.04 %

Tab. B.4 – Comparatif pour le bloc D.

Sortie	Simulation VHDL	Approche analytique	Erreur
n_{29}	0.7594	0.7688	1.2 %
n_{30}	0.7882	0.7987	1.3 %
n_{34}	0.842	0.846	0.5 %
n_{35}	0.8072	0.814	0.8 %
$Etape_{a0_1}$	0.6758	0.6450	4.7 %
$Etape_{a0_2}$	0.6292	0.5942	5.8 %
DSO	0.773	0.7544	2.4 %
BYP	0.7955	0.7934	0.3 %

Tab. B.5 – Comparatif pour le bloc E.

fiabilité de la porte elle-même). Ceci indique qu'il y a un problème de précision dans les simulations, effet qui s'accumule lors de la concaténation de plusieurs portes. Nous n'avons pas pu indentifier la source de ce problème, mais plusieurs causes sont possibles : les séquences pseudo-aléatoires utilisées pour simuler la probabilité d'erreur ne sont pas tout à fait uniformes, ou un problème de précision dans la définition du seuil pour faire basculer ou pas la sortie d'une porte.

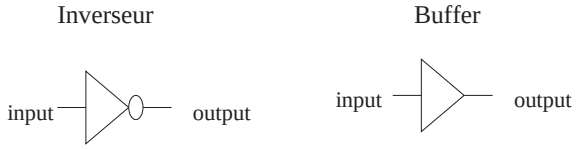


FIG. B.4 – Signaux définis par un inverseur ou un buffer logiques.

D'autres sources d'imprécisions sont possibles. Nous avons essayé de simuler les entrées du circuit de manière à ce que leur corrélation temporelle soit nulle (la valeur d'une entrée au cyle n ne dépend pas de la valeur au cycle $n-1$). Probablement, les séquences pseudo-aléatoires utilisées lors de l'attribution de valeurs aux entrées introduisent une corrélation temporelle qui induit une corrélation spatiale (c'est le type de corrélations traitées tout au long de ce manuscrit) dans les signaux qui n'est pas prise en compte dans le modèle analytique, introduisant donc une divergence des résultats.

Une autre source d'erreur peut venir de la définition du seuil de convergence que nous utilisons pour l'estimation dans l'approche analytique. Mais nous n'avons pas pu utiliser un seuil plus strict car la librairie mathématique utilisée dans l'outil d'analyse n'a pas une précision absolue, et les résultats divergent (la fiabilité trouvée tend vers 0 ou vers infini) lorsque le nombre d'itérations réalisé est très élevé.

Malgré les erreurs, les résultats obtenus ne sont pas aberrants, ce qui est un bon signe pour l'approche que nous proposons, or, le modèle proposé doit être raffiné, de même que les simulations utilisées pour la comparaison, afin d'obtenir une validation complète de l'approche.

Bibliographie

[1] A. Birolini, *Quality and Reliability of Technical Systems : Theory-Practice-Management* Springler-Verlag, 1994

[2] G.E. Moore, *Cramming more components onto integrated circuits Electron.* Vol.38, pp. 114-117, Avril 1965.

[3] D.T. Franco, J.F. Naviner et L. Naviner, *Yield and reliability issues in nanoelectronic technologies*, Annales des télécommunications, vol. 61, pp. 1422-1457, 2006

[4] M.R. Stan, P.D. Franzon, S.C. Goldstein, J.C. Lach, M.M. Ziegler, *Molecular Electronics : from devices and interconnect to circuits and architecture*, Proceedings of the IEEE, Vol. 91, pp. 1940-1957, 2003

[5] M.A. Breuer, S.K. Gupta, T.M. Mak *Defect and error tolerance in then presence of massive number of defects*, Design and Test for Computers IEEE, Vol. 23, pp. 216-227, 2004

[6] S. Bokar, T. Karnik, S. Narendra, J. Tschanz, A. Keshavarzi *Parameter Variations and impact on circuits and microarchitecture*, Proceedings Design Automation Conference, pp. 338-342, 2003

[7] P. Hazucha, C. Svensson, *Impact of cmos technology scalling on the atmospheric neutron soft error rate*, IEEE Transactions on Nuclear Science, Vol. 47, pp. 2586-2594, 2000

[8] Y. Zorian, *Nanoscale design and test challenges*, IEEE Computer Journal, Vol. 38, pp. 36-39, 2005

[9] R. Baumann *Soft errors in advanced computer systems*, IEEE Design and Test of Computers, Vol. 22, pp. 259-266, 2005

[10] R. Baumann, *The impact of technology scaling on soft error rate performance and limits to the efficacy of error correction*, Electron Devices Meeting (IEDM),pp. 329-332, 2002

[11] J.A. Carballo, S.R. Nassif *Impact of design-manufacturing interface on SoC design Methodologies*, IEEE Design and Test of Computers, Vol. 21, pp. 183-191, 2004

[12] L. Anghel, *Les limites technologiques du Silicium et Tolérance aux Fautes*, Thèse de doctorat, Laboratoire Techniques de l'Information et de la Microélectronque pour l'Architecture de l'Ordinateur (TIMA), 2000

[13] A. Villemeur, *Sûreté de fonctionnement des systèmes industriels*, Eyrolles, 1988

[14] LIS sous la direction de J.-C. Laprie, *Guide de la sûreté de fonctionnement*, Cépaduès, 1995

[15] W. Schemmert, G. Zimmer, *Threshold voltage sensivity of ion-implanted m.o.s transistors due to process variations*, Electronics Letters, Volume 10, n. 10, 1974, pp. 151-152

[16] Laszlo B. Kish, *End of moore's law : Thermal (noise) death of integration in micro and nano electronics*, Physics Letters A, Vol. 305, pp. 144-149, decembre 2002

[17] N. Seifert et al., *Radiation-induced soft error rates of advanced cmos bulk devices*, Reliability Physics Symposium Proceedings, pp.217-225, 2006.

[18] C.E. Ebeling, *An introduction to Reliability and Maintanability Engineering*, McGraw Hill, Boston, 1997

[19] D. Prasad, *Dependability terminology : A comparative study*, Rapport Technique, Computer Science Department, University of New York, 1994

[20] M. Rausand, A. Hoyland, *System Reliability Theory : Models, Satistical methods and applications*, John Wuiley and Sons, 2004

[21] H.E. Ascher, *Evaluation of Repairable System Reliability Using the "Bad-As-Old" Concept*, IEEE Transactions on Reliability, Vol.R-17, n.2, pp. 103-110, 1968

[22] G.F. Foxhall, *Bipolar linear integrated circuit reliability*, International Electron Devices Meeting, Vol.22, pp. 41, 1976

[23] M.H. Woods, *Reliability in MOS integrated circuits*, International Electron Devices Meeting, Vol. 30, pp. 50-55, 1984

[24] K.E. Portz, H.R. Smith, *Method for Determination of Reliability*, IRE Transactions on Reliability and Quality Control, Vol. PGRQC-11, pp. 65-73, 1957

[25] N. Sirisantana, B.C. Paul, K. Roy *Enhancing yield at the end of the technology roadmap*, IEEE Design and test of Computers, vol. 21, pp. 563-571, 2004

[26] Y. Zorian, D. Gizopoulos, C. Vandenberg, and P. Magarshack, *Guest editors' introduction : Design for yield and reliability, Design and Test of Computers, IEEE*, Mai-Juin 2004

[27] International Technology Roadmap for Semiconductors, *Itrs report 2005, executive summary*, http ://www.itrs.net/Links/2005ITRS/ExecSum2005.pdf, 2005

[28] C. Constantinescu, *Trends and challenges in VLSI circuit reliability*, Micro. IEEE, Vol.23, pp. 14-19, 2003.

[29] S. Mukherjee, *Architecture Design for Soft Errors*, Morgan Kaufmann, 2008

[30] R.C. Baumann, *Radiation-induced soft errors in advanced semiconductor technologies*, IEEE Transactions on Device and Materials eliability, Vol. 5, n.5, pp. 305-316, 2005

[31] N. Miskov-Zivanov, D. Marculescu *Formal modelling and reasoning for reliability analysis*, IEEE Design Automation Conference (DAC), pp. 531-536, 2010

[32] H.Asadi, M.B.Tahoori *Soft Error Derating Computation in Sequantial Circuits* Proc. of International Conference on Computer Aided Design (ICCAD), pp. 497-501, 2006

[33] S. Borkar *Tackling variability and Reliability Challenges*, IEEE Design and Test of Computers, Vol. 22, 2005

[34] . Miskov-Zivanov, D. Marculescu *Circuit Reliability Analysis uisng Symbolic Techniques*, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD), vol. 25, n. 1é, pp. 2638-2639, 2006

[35] P. Shivakumar,M. Kistler,W. Keckler, D. Burger, and L. Alvisi. *Modeling the effect of technology trends on the soft error rate of combinational logic* ,International Conference on Dependable Systems and Networks, pp. 389-398, 2002.

[36] J.F. Ziegler et W.A. Landford, *Effect of Cosmic Rays on Computer Memories*,Nature, vol. 206, pp. 776-778, novembre 1979

[37] J. Bourrieau *Effet sur les composants et les systemes*, Cours RADECS 89, Septembre 1989

[38] G. Hubert, *Elaboration d'une methode de prediction du taux d'aleas logiques induits dans une memoire SRAM par des neutrons atmospheriques*, Thèse de Doctorat, Université de Montpellier, Janvier 2001

[39] D.P. Sewiorek, *Reliability Modelling of Compensating Module Failures in Majority Voted Redundancy*, IEEE Transactions on Computers, Vol. C-24, n. 5, pp. 525-533, 1975

[40] S. Ghosh, N.A. touba, S. Basu, *Synthesis of low power CED circuits based on parity codes*, 23rd IEEE Proceedings VLSI Test Sympoium, pp. 315-320, 2005

[41] P.J. Meaney, S.B. Swaney, P.N. Sanda, L. Spainhower, *IBM z990 soft error detection and reconvery*, IEEE Transactions on Device and Materials Reliability, Vol. 5, n. 3, pp. 419-427, 2005

[42] P.K. Lala, A. Walker *On-line error detectable carry-free adder design*, Proceedings Defect and Fault tolerant in VLSI systems, 2001, pp.66-71, 2001

[43] M. Mishra, S.C. Goldstein, *Defect Tolerance at the end of the road map*, Proceeedings Test Conference (ITC 2003), pp. 1201-1210, 2003

[44] JEDEC, *Test procedure for the management of Single-event effects in semiconductors devices from heavy ion irradiation*, JEDS57, 1996

[45] JEDEC, *Measurement and reporting of alpha particle and terrestrial cosmic ray induced Soft Erros in semiconductor devices*, JESD89A, Octobre 2006

[46] N. Bidokhti *SEU concept to reality (allocation, prediction, mitigation)*, Reliability and Maintenability Symposium (RAMS), 2010

[47] P.E. Dodd, L.W. Massengill, *Basic mechanisms and modelling of single-event upset in digital microelectronics*, IEEE Transactions on Nuclear Science, Vol.50, pp. 583-602, 2003

[48] M.S. Lundstrom *Fundamentals of Carrier Transportation*, Addison-Wesley, 1990, Vol. X

[49] H. L. Anderson *Metropolis, Monte Carlo and the Maniac*, Los alamos Science, Vol. 14, 1986

[50] R.K. Iyer, D. Tang, *Experimental analysis of Computer System Dependability*, Fault-Tolerant Computer System Design, Prentice Hall, 1996

[51] P. Civera, L. Macchiarulo, M. Rebaudengo, M. Sonza Reorda, M. Violante, *Exploiting FPGA-based Techniques for Fault Injection Campaigns on VLSI Circuits*, IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems, pp. 250-258, 2001

[52] M. Garcia Valderas, M. Portela Garcia, R. Fernandez Cardenal, C. Lopez Ongil, Luis Entrena, *Advanced Simulation and Emulation Techniques for Fault Injection*, IEEE international Symposium on Industrial Electronics, 2007

[53] J.M. Daveau, A. Blampey, G. Gasiot, J. Bulone, P. Roche *An industrial fault injection platform for soft-error dependability analysis and hardening of complex system-on-a-chip*, IEEE International Reliability Physics Symposium, 2009

[54] C. Grinstead, J. Laurie-Snell, *Introduction to probability*, American Methematical Society, 1997

[55] S. Amarel, J.A. Brzozowski, *Theoretical considerations on reliability properties of recursive triangular switch networks*, Redundancy Techniques for computing systems, Wilcow and Mann Ed., 1962

[56] J. von Neumann, *Probabilistic logics and the synthesis of reliable organisms from unreliable components Automata Studies (Annals of Mathematics Studies), Princeton University Press*, pp. 43-98, Mai 1956.

[57] E.J. McCluskey, E.F. Clegg, *Fault equivalence in combinational logic networks*, IEEE. Transactions on Computers, Vol. C-20, pp. 1286-1293, 1971

[58] Roy C. Ogus, *The probability of a Correct Output from a Combinational Circuit*, IEEE Transactions on Computers, Vol.c-24, n 5 pp. 534-544, Mai 1975.

[59] K.P. Parker et E.J McCluskey, *Probabilistic Treatement of general combinational networks*, IEEE Transactions on Computers, Vol.c-24, n 5 pp. 668-670, Juin 1975.

[60] K.P. Parker et E.J McCluskey, *Analysis of logic circuits with faults using input signal probabilities*, IEEE Transactions on Computers, Vol.c-24, n 5 pp. 573-578, Juin 1975.

[61] Israel Koren, *Analysis of the Signal reliability Measure and an Evaluation Procedure*, IEEE Transactions on Computers, Vol.c-28, n 3, pp. 244-249, Mars 1979

[62] S.P. Doukouzgiannis et J.M Kontoleon, *Exact Reliability Analysis of Logic Circuits*, IEEE Transactions on Reliability, Vol.37, n. 5, pp. 493-500, Decembre 1988

[63] A. Bogliolo, M. Damiani, P. Olivo, B. Ricó, *Reliability Evaluation of Combinational Logic Circuits by Symbolic Simulation*, VLSI Test Symposium,, pp. 235-242, 1995

[64] K.S. Brace, R.L Rudell, R.E. Bryant, *Efficient implementation of a BDD package*, Proceedings of the Design Automation Conference, pp. 40-45, 1990

[65] M. Omana, D. Rossi, C. Metra, *Model for transient fault susceptibility pf combinational Circuits*, Journal of Electronic Testing : theory and applications, , Vol. 20, 2004

[66] B. Zhang, W.S. Wang, M. Orshansky, *FASER : Fast Analysis of Soft Error Rate susceptibility for cell-based designs*, International Symposium on quality electronic Design, pp.- 760, 2006

[67] N. Miskov-Zivanov, D. Marculescu, *Circuit Reliability analysis using symbolic techniques*, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol. 25, pp. 2638-2649, 2006

[68] R. Rao, K. Chopra, D.T. Blaauw, D.M. Sylvester, *Computing the Soft Error Rate of a combinational logic circuit using parametrized descriptors*, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol. 26, pp. 468-479, 2007

[69] F. Wang, Y. Xie, *Soft Error Rate Analysis for Combinational Logic Using an Accurate Electrical Masking Model*, IEEE Transactions on Dependable ans Secure Computing, Vol. 8,pp. 137-146, 2001

[70] L. Entrena, M. Valderas, R.F. Cardenal ; M.P. Garcia, C.L. Ongil, *SET Emulation Considering Electrical Masking Effects*, IEEE Transactions on Nuclear Science, Vol. 56, pp. 2021-2025, 2009

[71] F. Wang, Y. Xie, R. Rajaraman, B. Vaidyanathan, *Soft Error Rate Analysis for Combinational Logic Using An Accurate Electrical Masking Model*, VLSI Design Conference, pp. 165-170, 2007

[72] R. Kindermann, J.L. Snell, *Markov Random Fields and Their Applications*, American Mathematical society, 1980

[73] J. Chen, J. Mundy, Y. Bai, S.M. Chan, P. Petrica et R.I. Bahar, *A Probabilistic Approach to Nano-computing*, Proceedings of the Second Workshop on Non-silicon Computing, 2003

[74] M.C.R. de Vasconcelos, D.T. Franco, L.A de B. Naviner, J.F. Naviner, *Relianility Analsysis of Cpmbinational Circuits Based on a Probabilistic Binomial Model*, Proceedings of the 6th Northeast workshop on Circuits and Systems and TAISA conference, pp.310-313, 2008

[75] J. Han, E. Taylor, Jianbo Gao et Jose Fortes, *Faults, Error Bounds and Reliability of Nanoelectronic Circuits*, Proceedings of the 16th IEEE Interantional Conference on Application Specific Systems, ARchitecture Processors 2005, pp. 247-253, 2005

[76] J. Han et al., *reliability Evaluation of logic circuits using probabilistic gate models*, Journal of Microelectronics Reliability, Elsevier, 2010

[77] T. Rejimon et S. Bhanja, *Scalabable Probabilistic Computing Models using Bayesian Networks*, Proceedings of the 48th Midwest Symposium on Circuits and Systems, 2005, pp.712-715, 2005

[78] D. Bhaduri, S. Shukla, P. Graham, M. Gokhale, *Scalable techniques and tools for reliability analysis of logic circuits*, 20th International Conference on VLSI Design, pp. 705-710, 2007

[79] G. Asadi, M.B. Tahoori, *An Accurate SER Estimation Method Based on Propagation Probability*, Proceedings of the Design, Automation and Test in Europe Conference and Exhibition, pp.306-307, 2005

[80] S. Krishnaswamy, G.F. Viamontes, I.L Markov et J.P. Hayes, *Accurate Reliability Evaluation and Enhancement via Probabilistic Transfer Matrices*, Proceedings of the Design, Automation and Test in Europe Conference and Exhibition (DATE 05), pp. 282-287, 2005

[81] S. Krishnaswamy, G.F. Viamontes, I.L Markov et J.P. Hayes, *Probabilistic Transfer Matrices in symbolic Reliability Analysis of Logic Circuits*, ACM Transactions on Design Automation of Electronic Systems, vol.13, n. 1, article 8, 2008

[82] M.R. Choudhury, K. Mohanram, *Accurate and scalable reliability analysis of logic circuits*, Design and Test in Europe Conference (DATE), pp.1-6, 2007

[83] L.A.B.Naviner, M.C.R de Vasconcelos, D.T. Franco et J.F. Naviner, *Efficient Computation of Logic Circuits Reliability Based on Probabilistic Transfer Matrix*, Proceedings of the 2008 International Conference on Design and Technology of Integrated Systems in Nanoscale Era, pp.1-4, 2008

[84] O. Hasan, J. Patel, S. Tahar, *On the Accurate Reliability Analysis of Combinational Circuits using Theorem Proving*, IEEE Interantional NEWCAS conference, pp. 273-276, 2010

[85] D.T. Franco, M.C. Vasconcelos, L. Naviner et J.F Naviner, *Reliability Analysis of Logic Circuits based on Signal Probability*, IEEE International Conference on electronics, Circuits and Systems (ICECS), pp.670-673, 2008

[86] D.T. Franco, M.C. Vasconcelos, L. Naviner et J.F Naviner, *Reliability of Logic Circuits Under Multiple Simultaneous Faults*, 51st Midwest symposium on Circuits and Systems, pp. 265-268, 2008

[87] A.P.Dawid, *Conditional Independence in Statistical theory*, *Journal of the Royal Statistical Society*, Series B41(1), pp. 1-31, 1979

[88] C.C. Yu, J.P. Hayes, *Scalable and Accurate Estimation of Probabilistic Behavior in Sequential Circuits* , VLSI Test Symposium, pp. 165-179, 2010

[89] C.C. Yu, J.P. Hayes, *Trigonometric Method To handle Realistic Error Probabilities in Logic Circuits*, Design and Test in Europe Conference (DATE),pp.1-6, 2011

[90] F. Brglez, P. Pownall, R. Hum, *Application of testability analysis : from ATPG to critical delay path tracing*, Proceeding International Test Conference, p. 75, 1984

[91] B. Krishnamurthy, I.G. Tollis, *Improved techniques for estimating signal probabilities*, Proceedings International Test Conference, p. 244, 1986

[92] S.K. Jain et V.D. Agrawal, *Satistical fault analysis*, IEEE Design and Test, Février, p.38, 1985

[93] S. Ercolani, M. Favalli, M. Damiani, P. Olivo, B. Ricco, *Estimate of signal probability in combinational logic networks*, Proceedings of the 1st European Test Conference, pp. 132-138, 1989

[94] J. Savir, G.S. Ditlow, P.H. Bardell, *On random pattern test length*, IEEE Transactions on Computers, pp. 79-90, 1984

[95] G. Markowsky, *Bounding Signal Probabilities in Combinational Circuits*, IEEE Transactions on Computers, Vol. C-36, pp. 1247-1251, 1987

[96] R. Kodavarti, D. Ross, *Signal Probability Calculations using Partial Functional Manipulations*, Proceedings of VLSI Test Symposium, pp. 194-200, 1993

[97] A. Dutta, N.A. Touba, *Iterative OPDD based signal probability calculation*, Proceedings VLSI Test Symposium, pp. 77, 2006

[98] J.C. Laprie, *Dependable Computing and Fault Tolerant Concepts and Terminology*, 25th Symposium on Fault Tolerance Computing, 1995

[99] M. Fazeli et al., *Low Energy Single Event Upset/Single Event Transient-Tolerant Latch for Deep Submicron Technologies*, IET Computers and Digital Techniques, vol. 3, n. 3, 2009, pp. 289-303

[100] K. Mohanram, N.A. Touba, *Partial Error Masking to Reduce Soft Error Failure Rate in Logic Circuits*, Proceedings of the 18th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT'03),pp. 433-440, 2003

[101] B. Pratt, M. Caffrey, P. Graham, K. Morgan, M. Wirthlin, *Improving FPGA Design Robustness with Partial TMR*,44th Annual International Reliability Physics Symposim 2006, pp. 226-232, 2006

[102] S. Baloch, T. Arslan, A. Stoica, *Probability Based Partial Triple Modular Redundancy Technique for Reconfigurable Architectures*, IEEE Aerospace Conference, pp. 1-7, 2008

[103] H. Asadi, M.B. Tahoori, *Soft Error Hardening for Logic-level Designs*, IEEE international Sympoisum on Circuits and Systems, 2006

[104] A.K. Nieuwland, S. Jasarevic, G. Jerin, *Combinational Logic Soft Error analysis and Protection*, Proceedings of the 12th IEEE International On-Line Testing Symposium (IOLTS), 2006

[105] I, Polian, S.M. Reddy, B. Becker, *Scalable Calculation of Logical Masking Effects for Selective Hardening Against Soft Errors*, IEEE Computer Society Annual Symposium on VLSI, pp. 257-262, 2008

[106] K. Bhattacharya, N. Rangahathan, *A New Placement Algorithm for Reduction of Soft Errors in Macrocell Based Design of Nanometer Circuits*, 2009 IEEE Computer Annual Symposium on VLSI, pp.91-96, 2009

[107] E.C. Marques, L.A.B. Naviner, J.F. Naviner, *A Method for Efficient Implementation of Reliable Processors*, Proceedings of IEEE International Mid West Symposium on Circuits and Systems, 1250-1253, 2010

[108] L.A.B. Naviner, J.F. Naviner, T. Bant, G.G.S. Junior *Reliability analysis based on significance*, Conference on Micro nano-electronics, Technology and Applications (CMTA'11), pp.1-7, 2011

[109] M. Augustin, M. Gossel, R. Kraemer, *Reducing the Area Overhead of TMR-Systems by Protecting specific Signals*, Proceedings of the 16th IEEE International On-Line Testing Symposium, pp. 268-273, 2010

[110] R.P. Brent, H.T. Kung, *A regular layout for parallel adders*, IEEE Transactions on Computers, Vol. C-31, n. 3, pp. 260-264, 1982

[111] K. Iniewski, *Radiation effects in semiconductors*, CRC Press Taylor and Francis Group, Boca Raton, USA, 2010