



# Les limites technologiques du silicium et tolérance aux fautes

Lorena Anghel

► **To cite this version:**

Lorena Anghel. Les limites technologiques du silicium et tolérance aux fautes. Autre [cs.OH]. Institut National Polytechnique de Grenoble - INPG, 2001. Français. tel-00002907

**HAL Id: tel-00002907**

**<https://tel.archives-ouvertes.fr/tel-00002907>**

Submitted on 28 May 2003

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE

## THESE

pour obtenir le grade de  
**DOCTEUR DE L'INPG**

*Spécialité: « MICROELECTRONIQUE »*

préparée au laboratoire TIMA dans le cadre de l'Ecole Doctorale « *Electronique, Electrotechnique, Automatique, Télécommunications, Signal* »

**Lorena ANGHEL**

le 15 décembre 2000

### Titre

**LES LIMITES TECHNOLOGIQUES DU SILICIUM ET TOLERANCE AUX  
FAUTES**

*Directeur de thèse :*

**Mihail Nicolaidis**

### **JURY**

|                        |                             |
|------------------------|-----------------------------|
| M. Guy Mazare          | , <b>Président</b>          |
| M. Yervant Zorian      | , <b>Rapporteur</b>         |
| M. Matteo Sonza-Reorda | , <b>Rapporteur</b>         |
| M. Mihail Nicolaidis   | , <b>Directeur de thèse</b> |
| M. Eric Dupont,        | , <b>Examineur</b>          |

*Dedic aceasta teza, familiei mele  
si nu in ultimul rand, prietenului meu,  
fara de care nimic din toate acestea nu ar fi fost posibil*

« If you perceive that there are four possible ways in which something can go wrong, and circumvent these, then a fifth way, unprepared for, will promptly develop »

Murphy law

## **Remerciements**

Cette thèse a été réalisée au sein du groupe Reliable Intergrated Systems (RIS) du Laboratoire Techniques de l'Informatique et de la Microélectronique pour l'Architecture de l'Ordinateur (TIMA). Je tiens à remercier Monsieur Bernard Courtois, Directeur de recherche au CNRS et Directeur du laboratoire TIMA de m'avoir accueillie et donné les moyens pour accomplir mon travail de recherche.

J'adresse mes remerciements à Monsieur Guy Mazaré, Professeur et Directeur de l'ENSIMAG, pour avoir bien voulu me faire l'honneur de présider le jury de cette thèse.

Messieurs Matteo Sonza Reorda, Professeur à l'Université Polytechnique de Torino, et Yervant Zorian Directeur Scientifique à LogicVision Inc., sont les rapporteurs de cette thèse. Je tiens à les remercier pour l'intérêt qu'ils ont porté à mes travaux et leurs remarques judicieuses.

J'exprime également ma reconnaissance à Monsieur Eric Dupont, Président de la compagnie IROC Technologies, pour sa participation au jury de cette thèse ainsi que pour ses remarques constructives surtout pendant les derniers mois de thèse et l'opportunité qui m'a donné pour participer au projet IROC S81, qui m'a permit entre autres de voir l'aboutissement de mes travaux dans le développement d'un circuit industriel.

Je tiens à exprimer ma plus grande gratitude à Monsieur Mihail Nicolaidis, Directeur de Recherche au CNRS, pour m'avoir proposé ce sujet, pour l'encadrement de mon travail, ses conseils, ses critiques et ses encouragements, ainsi que pour sa disponibilité, sa bonne humeur et son optimisme.

J'aimerais également remercier mes collègues, enseignants et chercheurs auprès desquels j'ai beaucoup appris : Kholdun Torki, Paul Amblard, Regis Leveugle, Emmanuel Simeu, Alain Guyot, Nasserline Zergainoh.

J'adresse toute mon amitié à tous les membres du TIMA et CMP. Je salue également tous les thésards et stagiaires, et en particulier : Dan, pour sa participation active dans le déroulement de ce travail, Gabriela et Bogdan. Je leurs souhaite bonne chance.

Un merci tout particulier à Chantal Benis, Patricia Chassat et ces derniers temps à Elisabeth Crenais qui ont su m'aider à surmonter dans la bonne humeur les tâches administratives.

Je voudrais remercier l'équipe d'ingénieurs de IROC : Antoine, Rajmond, Damien, Paul, Moez, Florin, et les amis : Cristina, Dan, Casiana, Monica, Bobby, Nora, Raoul, Catherine, pour leurs encouragements permanents surtout ces derniers mois.

Une pensée tout particulière va à Céline et son tout petit bonhomme, ainsi qu'à Jacques, les deux lecteurs de mon manuscrit, impitoyables face aux fautes d'orthographe et aux mauvaises tournures de phrase.

# TABLES DES MATIERES

## INTRODUCTION ..... 13

|                                 |    |
|---------------------------------|----|
| ORGANISATION DU MANUSCRIPT..... | 19 |
|---------------------------------|----|

## CHAPITRE I. LES TECHNOLOGIES NANO-MÉTRIQUES ET LEURS DÉFIS ... 21

|  |    |
|--|----|
| 1.1. LES NOUVEAUX PROBLÈMES RENCONTRÉS DANS LES TECHNOLOGIES SOUS-MICRONIQUES.....                       | 22 |
| 1.2. LES DÉFIS DES TECHNOLOGIES NANOMÉTRIQUES .....  | 24 |
| 1.3. LES EFFETS DES TECHNOLOGIES SOUS-MICRONIQUES SUR LES INTERCONNEXIONS.....                           | 29 |
| 1.3.1. <i>Les phénomènes RC</i> .....  | 29 |
| 1.3.2. <i>L'intégrité du signal (« Signal Integrity »)</i> .....   | 32 |
| 1.4. L'IMPACT DES TECHNOLOGIES SOUS-MICRONIQUES SUR LE FLOT DE CONCEPTION DES CIRCUITS INTÉGRÉS .....    | 34 |
| 1.5. LES EFFETS DES TECHNOLOGIES NANOMÉTRIQUES SUR LES MODÈLES DE FAUTES ET LE TEST DE FABRICATION ..... | 37 |
| 1.6. L'ENVIRONNEMENT RADIATIF ET SES CONSÉQUENCES SUR LES CIRCUITS INTÉGRÉS NANOMÉTRIQUES .....          | 42 |
| 1.6.1. <i>Les différentes interactions</i> .....   | 42 |
| 1.7. LES EFFETS DU RAYONNEMENT SUR LES CIRCUITS INTÉGRÉS .....   | 44 |
| 1.7.1 <i>Terminologie</i> .....  | 44 |
| 1.8. LES DIFFÉRENTS TYPES DE DÉFAILLANCES DANS LES CIRCUITS INTÉGRÉS.....                                | 45 |
| 1.8.1. <i>La dose cumulée</i> .....  | 45 |
| 1.8.2. <i>SEU et Multiple-Bit Upset</i> .....  | 46 |
| 1.8.3. <i>Latch-up</i> .....   | 48 |
| 1.9. MOYENS DE RÉDUCTION DES IMPACTS DES SEES.....   | 50 |
| 1.9.1. <i>Le blindage</i> .....  | 50 |
| 1.9.2. <i>Le durcissement des composants</i> .....   | 50 |
| 1.9.3. <i>Durcissement par la conception au niveau de la cellule</i> .....                               | 53 |
| 1.9.4. <i>La prévention au niveau du système</i> .....   | 55 |
| 1.10. LES EFFETS DE LA MINIATURISATION .....   | 55 |
| CONCLUSION.....  | 59 |

## CHAPITRE II. MODÉLISATION ET SIMULATION DES IMPULSIONS TRANSITOIRES ..... 63

|  |    |
|--|----|
| 2.1. MODÈLE LOGIQUE DE L'IMPULSION TRANSITOIRE RÉSULTANT DE L'IMPACT DES PARTICULES ALPHA DANS LES CIRCUITS CMOS ..... | 63 |
| 2.2. MODÈLE TEMPOREL D'IMPULSION TRANSITOIRE .....   | 66 |
| 2.3. ANALYSE DE LA PROPAGATION DE L'IMPULSION TRANSITOIRE DANS UN CIRCUIT LOGIQUE .....                                | 69 |

|   |    |
|---|----|
| 2.4. L'ENVIRONNEMENT DE LA SIMULATION DES FAUTES TRANSITOIRES ..... | 70 |
| 2.5. LES AVANTAGES DE LA SOLUTION PROPOSÉE .....                    | 75 |

### **CHAPITRE III. CIRCUITS AUTO-CONTRÔLABLES FACE AUX DÉFAUTS CRITIQUES DANS LES TECHNOLOGIES NANOMÉTRIQUES ..... 77**

|  |     |
|--|-----|
| 3.1. DÉFINITIONS.....  | 79  |
| 3.2. LES CODES DE DÉTECTION D'ERREURS .....  | 81  |
| 3.2.1. <i>Le code de parité</i> .....  | 81  |
| 3.2.2. <i>Code double rail</i> .....   | 82  |
| 3.2.3. <i>Codes non ordonnés</i> .....   | 82  |
| 3.2.4. <i>Codes arithmétiques</i> .....  | 83  |
| 3.3. LES CONTRÔLEURS* .....  | 83  |
| 3.4. LES CIRCUITS 'SÛRS EN PRÉSENCE DE FAUTES' FACE AUX FAUTES À COMPORTEMENT COMPLEXE [ANG'00B] ..... | 88  |
| 3.4.1. <i>Extension du modèle de collage logique</i> .....   | 88  |
| 3.4.2. <i>Fautes à comportement complexe</i> .....   | 92  |
| 3.4.3. <i>Classes de circuits « sûrs en présence de fautes »</i> .....                                 | 96  |
| 3.5. CIRCUITS NE PRÉSENTANT PAS LA PROPRIÉTÉ 'À CHEMINS SENSIBILISÉS SÛRS' .....                       | 106 |
| 3.6. RÉSULTATS EXPÉRIMENTAUX.....  | 110 |
| CONCLUSION.....  | 114 |

### **CHAPITRE IV. CIRCUITS TOLÉRANTS LES PERTURBATIONS TRANSITOIRES À L'AIDE DES REDONDANCES TEMPORELLES ..... 117**

|  |     |
|--|-----|
| 4.1. TOLÉRANCE AUX FAUTES TRANSITOIRES BASÉE SUR LA REDONDANCE TEMPORELLE .....                                      | 118 |
| 4.1.1. <i>Implémentation du principe de fonctionnement à l'aide des portes logiques</i> .....                        | 120 |
| 4.1.2. <i>Surcoût matériel et dégradation de vitesse. Couverture des défauts</i> .....                               | 122 |
| 4.2. CIRCUITS TOLÉRANTS LES PERTURBATIONS COMBINANT LA REDONDANCE MATÉRIELLE ET TEMPORELLE .....                     | 123 |
| 4.2.1. <i>Principe de fonctionnement</i> .....   | 124 |
| 4.2.2. <i>Redondance temporelle basée sur les éléments de préservation d'état [ANG 00c]</i> .....                    | 133 |
| 4.2.3. <i>Evaluations du coût matériel et de la dégradation de vitesse</i> .....                                     | 135 |
| 4.2.4. <i>Evaluation de l'efficacité de la tolérance aux défauts</i> .....   | 136 |
| 4.3. TECHNIQUES DE DÉTECTION DE FAUTES TRANSITOIRES ET DE FAUTES DE TIMING À L'AIDE DE REDONDANCES TEMPORELLES ..... | 138 |
| 4.3.1. <i>Principe de fonctionnement</i> .....   | 138 |
| 4.3.2. <i>Implémentation à l'aide des portes standard et protection des latches [NIC '99]</i> .....                  | 140 |
| 4.3.2.1. <i>Dégradation des performances</i> .....   | 142 |
| 4.3.2.2. <i>Coût matériel</i> .....  | 143 |
| 4.3.2.3. <i>Couverture des fautes</i> .....  | 143 |
| 4.3.2.4. <i>L'implémentation du circuit de distribution de l'horloge</i> .....                                       | 144 |
| 4.3.3. <i>Evaluation du coût matériel [ANG 00a]</i> .....  | 145 |
| 4.3.4. <i>Méthodes de réduction du coût matériel</i> .....   | 149 |
| 4.3.4.1. <i>Implémentation</i> .....   | 149 |
| 4.3.4.2. <i>Evaluation du surcoût d'implémentation</i> .....   | 152 |
| 4.3.5. <i>Evaluation de l'efficacité de détection des fautes transitoires et des erreurs de timing</i> .....         | 154 |
| CONCLUSION.....  | 156 |

### **CONCLUSIONS ET PERSPECTIVES ..... 159**

### **REFERENCES ..... 163**

## Introduction

L'introduction des technologies des circuits intégrés (VLSI) a abouti à une amélioration substantielle de la fiabilité des systèmes électroniques, par rapport aux technologies utilisant des composants discrets (tubes, transistors discrets, etc). Néanmoins, ces dernières années, la complexité des circuits intégrés s'est accrue de façon spectaculaire. Cette tendance s'accompagne d'une réduction des marges de bruit. Par conséquent, les circuits et les systèmes électroniques modernes deviennent de plus en plus "exposés" aux erreurs ou bien aux défaillances.

Dans le passé, on pouvait s'assurer d'un niveau de fiabilité des circuits VLSI suffisant pour une grande majorité d'applications, en éliminant les circuits défaillants par le biais du test de fabrication.

La génération des vecteurs de test des circuits VLSI a vécu plusieurs phases. Dans un premier temps, correspondant à des circuits très peu complexes, un test exhaustif était utilisé, ceci étant permis par la faible complexité des composants. Dans un deuxième temps, on a généré des vecteurs de test manuellement, en utilisant une approche fonctionnelle. Quand cette approche est devenue caduque à cause d'une complexité des composants de plus en plus élevée, l'approche structurelle a pris la relève. Cette approche se base sur la connaissance de la structure du circuit et vise à détecter les défauts affectant chacun des composants du circuit.

Le modèle des fautes largement adopté dans cette approche consiste dans le modèle de collage logique. Ce modèle admet le collage de chaque entrée et de chaque sortie d'une



porte logique à une valeur logique constante '0' ou '1'. L'adoption généralisée de ce modèle était due à deux raisons : la facilité de la génération des vecteurs de test automatique par des algorithmes nécessitant un temps de calcul raisonnable, et une couverture satisfaisante de défauts réels rencontrés dans plusieurs générations des circuits intégrés. Les « ATPG » utilisant des approches de type algorithme D [ROTH '66][ROTH '67], PODEM [GOEL'81], FAN [FUJ' 83] et autres ont permis aux industriels de générer des vecteurs de test et d'accomplir le test de fabrication de façon économique. Les simulateurs de fautes jouent aussi un rôle important car ils sont plus rapides que les « ATPG » et permettent d'éliminer rapidement un grand nombre de fautes par simulation des vecteurs de test fonctionnels et/ou aléatoires. Néanmoins, l'augmentation de la complexité des circuits intégrés a aussi rendu cette approche caduque à cause d'une diminution de la contrôlabilité et de l'observabilité des circuits, dont les causes sont principalement la diminution du rapport entrées – sorties/nœuds internes, l'augmentation de la profondeur combinatoire et de façon plus grave, l'augmentation de la profondeur séquentielle. Ces changements rendent l'approche « ATPG » inopérante car le temps de génération des vecteurs de test pour des circuits de plus en plus complexes devient vite irréaliste. En même temps, la longueur du test augmente beaucoup et rend le coût du test sous testeur inacceptable. L'introduction de l'approche « scan-path » [KOB'68] [FUN'75] permet par la suite de contourner ce problème, car on peut contrôler directement les entrées et observer les sorties de blocs entiers, y compris leurs entrées et leurs sorties d'état (dans le cas de blocs séquentiels). Malgré un coût d'implémentation élevé, le « scan-path » a été adopté de façon très généralisée par l'industrie après une première période de réticences et de résistance. Bien sûr, la raison de cette adoption était économique, le coût du test sans « scan-path » était devenu plus élevé par rapport au coût en surface de silicium supplémentaire introduit par l'insertion du « scan-path ». La facilité de diagnostic dans le cas des circuits avec « scan-path » a aussi joué un rôle très important. Mais les technologies VLSI ont continué leur évolution vers des circuits de plus en plus complexes et de plus en plus rapides tels que des « systems-on-chip » (SOC), incluant des parties hétérogènes (processeurs, DSP, mémoires, parties analogiques, parties logiques reconfigurables). La technique du « scan-path » devient inefficace pour de tels systèmes car l'augmentation de la complexité des différents blocs embarqués résulte en un nombre de vecteurs de test de plus en plus élevé. De plus, l'augmentation du nombre de « latches » résulte en des chaînes de

« scan-path » comportant plusieurs milliers de cellules. Ainsi, chaque vecteur de test nécessite d'être décalé plusieurs milliers de fois avant d'être acheminé jusqu'à sa destination. Cette situation allonge de façon significative le temps de test et par conséquent son coût.

D'autre part, la technique « scan path » ne permet pas de tester les circuits à leur vitesse de fonctionnement normale. Ce handicap est d'autant plus contraignant que le fonctionnement des circuits est de plus en plus influencé par les défauts affectant le comportement temporel (« timing faults » dans la terminologie anglaise), vu les fréquences de fonctionnement de plus en plus élevées. De plus, l'approche « scan path » est inefficace pour certaines parties telles que les mémoires ou parties analogiques. Ces nouvelles contraintes poussent depuis plusieurs années un nombre croissant de compagnies à adopter l'approche « BIST – Built In Self-Test » [GEL'87] [BEN'75][BON'79][FAS'82][KAR'89]. Cette approche réduit la durée de test relative au « scan-path », permet de tester les circuits à leur vitesse de fonctionnement normale et réduit le coût du testeur car elle évite le besoin de testeurs à la fois très rapides et capables de tester des parties hétérogènes (tels que les parties logiques, les mémoires et les parties analogiques).

Et si l'adoption du « BIST » pour les mémoires embarquées est aujourd'hui généralisée, la seule raison pour laquelle des réticences persistent pour le « BIST » de parties logiques est liée à l'inefficacité relative des outils et des techniques de « BIST » logique. En effet, il est encore difficile de créer des structures de « BIST » compactes, permettant une couverture de test élevé. Cependant, ces réticences se dissipent au fur et à mesure que les progrès de la technologie rendent les autres techniques inopérantes.

Ces considérations prennent en compte les défauts survenant au cours de la fabrication des circuits intégrés. En théorie, ces défauts peuvent être détectés, et par conséquent, éliminés pendant les tests de fabrication. Malheureusement, le test de fin de fabrication ne peut pas protéger les circuits contre toutes les défaillances qui se manifestent durant le fonctionnement du circuit. Par exemple, on ne pourrait pas se protéger contre les fautes transitoires par le test de fabrication car ces fautes ne sont pas permanentes, leurs causes étant le plus souvent externes aux circuits (exemple : l'impact des particules ionisantes). Des techniques de tolérance aux fautes [SIEW'92],[PRAD'86] ou de durcissement (par procès ou par architecture) [DIEH'83],[MUSS'96],[SEXT'89],[CALI'96] sont les seules alternatives permettant de se protéger contre les défaillances qui

ont pour cause ces types de fautes. Par conséquent, la tolérance aux fautes était d'une grande utilisation dans les premiers systèmes informatiques, à cause d'une faible fiabilité des composants de l'époque, tels que les tubes semi-conducteurs et autres composants discrets. L'amélioration drastique de la fiabilité obtenue par l'introduction des circuits VLSI a confiné les techniques de tolérance aux fautes dans les applications spécifiques exigeant des niveaux de sécurité élevés (par exemple, le domaine ferroviaire), ou des applications évoluant dans des environnements sévères (le domaine spatial). On pourrait s'interroger si cette tendance d'amélioration de la fiabilité des circuits VLSI va continuer dans l'avenir, ou au contraire, risquerait de s'inverser dans un avenir proche, créant ainsi les conditions pour un grand retour à des techniques de tolérance aux fautes. Nous avons pris le pari de la dernière évolution depuis le début de ce projet de thèse. Les différentes annonces et études publiées cette année par les principaux leaders de l'industrie des semi-conducteurs et systèmes informatiques semblent confirmer cette estimation [CAT'99], [COH'99], [BAUM'99]. Les causes d'une telle dégradation de la fiabilité sont difficiles à neutraliser, car elles sont la conséquence du progrès technologique, qu'il faudra donc, stopper pour l'éliminer.

En fait, les technologies de silicium s'approchent de leurs limites physiques en terme de réduction de tailles des transistors, de réduction de la tension d'alimentation ( $V_{DD}$ ) et de la tension de seuil ( $V_T$ ), d'augmentation de la vitesse de fonctionnement et du nombre de dispositifs intégrés dans une puce (dizaines, voir même centaines de millions de transistors). En s'approchant de ces limites, les circuits deviennent de plus en plus sensibles à toute source de bruit : phénomènes de couplage capacitif (en anglais «cross-talks»), influence électro-magnétique, bruit sur les lignes d'alimentation «ground-bounce», ainsi qu'aux phénomènes radiatifs (particules alpha et neutrons atmosphériques) qui, pendant longtemps, ne représentaient un problème significatif que pour les circuits intégrés situés dans les environnements sévères (essentiellement le domaine spatial ou nucléaire). Aujourd'hui, le phénomène des erreurs softs produites par l'interaction du rayonnement avec les circuits intégrés prend de l'ampleur dans les altitudes de vols commerciaux et bientôt sur terre. C'est ainsi que le problème de la fiabilité des circuits intégrés devient de nouveau un problème majeur.

Les grandes compagnies de systèmes informatiques se préoccupent maintenant de tous ces aspects. Selon Intel, les erreurs transitoires dues aux particules cosmiques (on les appelle « soft errors ») deviennent le deuxième grand problème après l'augmentation des courants de fuites dans le domaine sous-micronique [CAT'99]. La plupart des compagnies observent une augmentation importante des erreurs 'softs' à partir de la technologie 0,25  $\mu\text{m}$ .

La simple raison de l'augmentation de la sensibilité face aux bruits est en fait la réduction de la tension d'alimentation ( $V_{DD}$ ) et de la taille du transistor, diminution qui entraîne une réduction de la capacité des nœuds du circuit. Ainsi la charge électrique stockée dans un nœud est considérablement réduite. La charge déposée par une particule cosmique lors de son impact sur le silicium peut donc plus facilement renverser la valeur logique associée au nœud. Un tel renversement, s'il affecte un nœud d'une cellule mémoire, peut basculer son état logique, résultant en une erreur SEU (Single Event Upset). Les mémoires étaient les parties les plus sensibles aux SEUs dans un système informatique. C'est ainsi que pour les tolérer, les concepteurs ont fait appel aux codes détecteurs / correcteurs d'erreurs, dans une première phase ou aux mémoires durcies aux radiations.

Néanmoins, la miniaturisation des circuits entraîne également une croissance de la sensibilité des parties logiques-combinatoires. En effet, le temps de transition logique des portes dans les technologies allant en dessous de 0,35 $\mu\text{m}$  devient plus petit que les durées moyennes des fautes transitoires générées à la suite d'un impact de particules énergétiques (d'habitude de l'ordre de quelques centaines de picoseconds). Ces impulsions transitoires ne seront alors plus filtrées par les portes logiques et se propageront sans encombre jusqu'aux sorties du circuit ou elles sont susceptibles d'être capturées par les latches de sortie. Cet événement aurait lieu si l'arrivée de l'impulsion aux sorties du circuit coïncidait avec le front actif de l'horloge. Et, comme la fréquence d'horloge des systèmes actuels devient de plus en plus grande, qu'elle va dépasser bientôt 1 GHz, la probabilité de capturer une telle erreur augmentera proportionnellement.

Un autre type de problème est lié aux erreurs de timing. La variation des paramètres de procédé technologique, des défauts de type 'open' ou court-circuit affecte souvent la

vitesse de fonctionnement du circuit. Plus les circuits deviennent rapides, plus ces types de fautes vont se manifester en erreurs de timing. La génération automatique de vecteurs de test pour ce type de fautes nécessite des temps de calcul très élevés et des temps de test très longs, donc très coûteux. C'est ainsi qu'une partie des circuits peut passer les tests de fabrication avec des fautes de timing.

L'objectif de cette thèse consiste à étudier la possibilité d'utilisation des techniques de tolérance aux fautes dans le but de maintenir des niveaux de fiabilité acceptables pour les prochaines générations de circuits sous-microniques avancés. Néanmoins, les techniques de tolérance aux fautes traditionnelles utilisant une redondance massive (duplication, triplification) sont très coûteuses. Elles sont acceptables dans les systèmes à haute valeur ajoutée (applications critiques en sécurité, satellites, transactions bancaires, ...), mais inacceptables pour la grande majorité d'applications. Par conséquent, notre étude des techniques de tolérance aux fautes implique un faible coût. Une telle technique utilise les circuits auto-contrôlables (« self-checking » dans la littérature anglaise), mais ces circuits ont été développés pour couvrir des fautes permanentes de type collage logique. Les fautes auxquelles nous nous intéressons dans cette étude étant différentes (fautes transitoires et fautes de timing), nous étudions dans un premier temps le comportement des circuits « self-checking » consacrés à ce type de fautes. Nous proposerons des implémentations permettant une meilleure prise en compte de ces dernières. Par la suite, nous étudierons les techniques utilisant une redondance temporelle. En fait, comme les fautes de timing et les fautes transitoires sont des fautes temporelles, nous pouvons exploiter une redondance temporelle afin de nous protéger contre elles, en minimisant le coût matériel. Nous étudions une technique de détection de fautes basée sur la redondance temporelle ainsi qu'une technique de tolérance aux fautes utilisant une méthode spatio-temporelle. Il est intéressant de pointer un fait remarquable : la technique de détection de fautes utilise la redondance temporelle sans pour autant affecter la vitesse de fonctionnement du circuit. Finalement, l'efficacité de protection obtenue par ces techniques doit être évaluée de façon précise. Dans ce but, nous avons développé, une méthodologie de simulation des fautes transitoires pour valider nos analyses.

## ***Organisation du manuscrit***

Le manuscrit est organisé en quatre grandes parties. L'étude commence par une présentation des nouvelles technologies sous-microniques et de leur impact sur la conception et le test des circuits intégrés, suivi par une description des effets des radiations spatiales sur les circuits intégrés.

La deuxième partie présente une méthode de simulation avec injection de fautes transitoires. Le chapitre suivant analyse le comportement des circuits auto-contrôlables implémentés dans une technologie sous-micronique. Dans ces technologies, certaines classes de circuits 'sûrs en présence des fautes' préservent leurs propriétés tandis que certains circuits peuvent les perdre. Des circuits 'non-sûrs en présence d'une faute' spécifique aux technologies sous-microniques peuvent avoir une couverture de fautes meilleure que les circuits 'sûrs par rapport aux mêmes fautes'.

Le quatrième chapitre propose des techniques de tolérance aux fautes transitoires et de timing en exploitant la nature temporelle de ces deux types de fautes. Une évaluation des techniques de tolérance aux fautes transitoires et de timing est donnée en termes de surcoût en surface et de dégradation des performances. Une évaluation de l'efficacité de détection des fautes complète cette analyse.



## Chapitre I

### Les technologies nano-métriques et leurs défis

La première partie de ce chapitre sera consacrée aux problèmes actuels rencontrés par le concepteur de circuits intégrés dans les technologies sous-microniques. Ces problèmes représentent aujourd'hui un nouveau défi pour les concepteurs de VLSI, mais aussi pour les grands producteurs de systèmes électroniques, qui risquent de voir une baisse considérable de fiabilité et de qualité de leurs produits, s'ils ne trouvent pas de solutions adéquates. Une partie des problèmes sera bientôt, ou est déjà prise en compte, par certains outils de conception de circuits et systèmes intégrés (CAO) qui existent sur le marché. Avec l'arrivée des technologies nanométriques, les phénomènes transitoires, déjà difficiles à modéliser et à prédire, sont aujourd'hui de plus en plus évidents et nombreux, et constituent la cause de plus de 80% des défaillances dans les systèmes informatiques [SIEW'92]. Ce chapitre traite une partie des phénomènes transitoires apparus avec l'arrivée des technologies sous-microniques et discute l'impact de ces problèmes sur l'avancement du processus technologique vers le nanométrique. Parmi les effets transitoires les plus difficiles à maîtriser à l'aide d'outils CAO, on note l'impact des particules provenant du rayonnement cosmique (par exemple, les neutrons atmosphériques) ou de la désintégration des isotopes instables présents dans les matériaux des circuits intégrés et de leur encapsulation (particules alpha), ainsi que les défauts de fabrication et les phénomènes de couplage impossibles à détecter avec les tests fonctionnels classiques. La fin du chapitre est dédiée à la présentation de l'environnement radiatif spatial et terrestre et les conséquences de l'interaction des particules atmosphériques avec les circuits sous-microniques avancés, ainsi que les moyens technologiques mis en œuvre pour s'en prémunir.



## **1.1. Les nouveaux problèmes rencontrés dans les technologies sous-microniques**

Le comportement des circuits intégrés devient de plus en plus en plus complexe au fur et à mesure qu'on miniaturise les composants, qu'on réduit leur niveau d'alimentation, et qu'on augmente leur vitesse et leur complexité. Les retards dus aux interconnexions sont déjà beaucoup plus importants que les temps de propagation des portes logiques, et les transistors ne fonctionnent pas de la même façon qu'auparavant.

Tout d'abord, on analyse le comportement des interconnexions qui deviennent de plus en plus complexes. Alors que la taille des transistors diminue de plus en plus, les transistors opèrent à des vitesses de plus en plus élevées, mais la longueur des interconnexions reste stable, voire même augmente, étant donné que les dimensions des puces tendent à augmenter.

Au même temps, la largeur des lignes d'interconnexions doit être réduite pour suivre la tendance de miniaturisation. Ainsi, on est obligé d'augmenter la hauteur des lignes pour ne pas aggraver le problème d'électromigration (l'introduction du cuivre va aussi dans le même sens). Ces tendances modifient de façon drastique la répartition des retards dans les circuits, les retards dus aux interconnexions devenant bien plus importants que les retards dus aux parties actives. Le couplage capacitif entre les interconnexions ou « crosstalks » devient un phénomène majeur (augmentation des capacités latérales entre les interconnexions et le substrat), tandis que le couplage inductif ne peut plus être négligé à cause de l'augmentation significative de la vitesse de fonctionnement.

Du côté des parties actives - et tout d'abord du point de vue de la modélisation - les effets du deuxième ordre commencent à devenir importants et on ne peut plus les ignorer comme dans le passé. On voit donc une augmentation de la sensibilité au bruit à cause de la réduction du niveau de l'alimentation, de la réduction des capacités des nœuds à la sortie d'une porte logique, mais également à cause de l'augmentation de la vitesse de fonctionnement. Le bruit sur les lignes d'alimentation (« ground bounce ») apparaît ainsi

comme un problème majeur. La sensibilité aux particules (rayonnement cosmique et particules alpha) devient inacceptable. La technologie actuelle (0,18 $\mu\text{m}$ ) est plutôt vulnérable pour les applications fonctionnant à bord de satellites et dans le domaine de l'avionique. Mais avec l'arrivée des technologies nanométriques (100nm), le nombre d'erreurs provenant des particules atmosphériques sera inacceptable même au niveau terrestre.

Le traitement de ces problèmes nécessite des outils de modélisation et de simulation les plus adéquats, qui ne sont pas disponibles dans les outils de CAO traditionnels. Une grande partie des solutions offertes par les outils de conception de CAO (outils de synthèse, de simulation et de placement-routage) n'est plus valable pour les circuits conçus dans une technologie sous-micronique. Les modèles SPICE du transistor sont souvent incomplets, et ne prennent pas en compte les effets du deuxième ordre (négligeables dans les technologies microniques) ou la variation des performances du transistor, qui affectent sérieusement le temps et le seuil de commutation du transistor. L'électro-migration, la consommation de puissance, l'intégrité du signal, la précision de l'horloge, les couplages parasites entre les différentes parties du système, ne sont pas traités complètement par les outils de CAO dont on dispose actuellement. Par exemple, une mauvaise prise en compte des phénomènes de couplage capacitif réduit le rendement de fonctionnement du circuit de 30% voir de 40% pour certaines compagnies [LAMM'98].

De plus, même si on dispose d'outils adéquats, la prise en compte de ces phénomènes ne sera complète qu'après le placement-routage. Ainsi, le concepteur ne pourra vérifier le respect de ses contraintes (vitesse, consommation, surface) qu'à la fin de cette phase, nécessitant ainsi plusieurs allers et retours entre la phase de synthèse et de placement-routage. Cette situation devient d'autant plus critique, que la pression du marché réduit de plus en plus le temps dont disposent les concepteurs pour finaliser un projet.

Mais, en supposant que le concepteur dispose d'outils adéquats et du temps nécessaire pour traiter ces problèmes, il reste toujours à résoudre le problème du test. L'augmentation de la complexité et de la vitesse des circuits résulte en une testabilité de plus en plus réduite et une inaptitude des testeurs en termes de vitesse, de taille de mémoire (largeur et profondeur), en termes de nombre d'entrées-sorties et de capacité à traiter des circuits intégrés hétérogènes (incluant des parties logiques, des mémoires, des parties

analogiques, voire même des MEMs). Tout ceci affecte la qualité et le prix du test de façon alarmante. Outre cela, un autre problème critique provient de la complexité des modèles de fautes qu'on doit prendre en compte pour les technologies sous-microniques avancées. La raison principale de cet accroissement de complexité est l'augmentation de la vitesse de fonctionnement. A des cadences dépassant 1 GHz, les défauts de « timing » deviennent prédominants. Néanmoins, ces défauts nécessitent des temps de calcul pour l'ATPG, et des temps de test qui sont le plus souvent irréalistes. De plus, la génération de vecteurs de test devrait prendre en compte non seulement l'activation des défauts, mais aussi d'autres phénomènes affectant les retards dans le circuit (crosstalks et ground-bounce). On se rend compte qu'il devient de plus en plus difficile d'éliminer tous les défauts de fabrication. Ainsi, pour les défauts indétectables, la seule solution restante est de les détecter pendant le fonctionnement du circuit en utilisant des techniques de tolérance aux fautes. Ce problème, ainsi que l'augmentation de la sensibilité face aux impacts des particules ionisantes, jouent en faveur d'une introduction systématique de la tolérance aux fautes dans la phase de conception des circuits intégrés. Tel est le pari pris dans l'élaboration de cette thèse.

Par la suite, nous discuterons plus en détail les causes principales de l'apparition de la majorité des phénomènes que nous venons d'énumérer.

## **1.2. Les défis des technologies nanométriques**

Plusieurs sources indépendantes [SYL'99] prévoient que pour les circuits fabriqués en technologies sous-microniques, 80% ou plus de leurs temps de propagations critiques sont directement liés aux temps de propagation dans les interconnexions. Les interconnexions et leur impact sur la vitesse de fonctionnement du circuit font partie aujourd'hui des phénomènes capables d'arrêter la continuation de la loi de Moore ( «la densité d'intégration double tous les 18 mois »). Parmi les effets capables d'arrêter le progrès technologique, on peut mettre en évidence plusieurs facteurs comme :

- l'augmentation des courants de fuite,
- les retards de propagation dans les liaisons métalliques, de type résistance-capacité (RC),

- les bruits provenant du couplage capacitif (crosstalks),
- la réduction de la fiabilité des circuits sous-microniques due à l'augmentation de la densité de courant et du champ électrique dans l'oxyde (électromigration)
- l'augmentation de la consommation de puissance. [SEMATECH]

Dans le tableau 1.1 on présente l'évolution prévisionnelle des semi-conducteurs (Carte Technologique SIA [SEMA '98] [SYL 99]) avec les données les plus importantes concernant l'avenir du progrès technologique dans le domaine nanométrique et la prédiction du progrès dans la fabrication des mémoires et des circuits combinatoires.

| Année  | 1997    | 1999    | 2001    | 2003    | 2006    | 2009    |
|--|---------|---------|---------|---------|---------|---------|
| Technologie  | 250nm   | 180nm   | 150nm   | 130nm   | 100nm   | 70nm    |
| <b>Progrès</b>   |         |         |         |         |         |         |
| <b>V<sub>DD</sub> (V)</b>                                | 1.8-2.5 | 1.5-1.8 | 1.2-1.5 | 1.2-1.5 | 0.9-1.2 | 0.6-0.9 |
| <b>V<sub>T</sub> (V)</b>                                 | 0.625   | 0.450   | 0.375   | 0.375   | 0.3     | 0.225   |
| <b>T oxyde (nm)</b>                                      | 4-5     | 3-4     | 2-3     | 2-3     | 1.5-2   | <1.5    |
| <b>Champ max E (MV/cm)</b>                               | 4-5     | 5       | 5       | 5       | >5      | >5      |
| <b>Epaisseur ligne métal (µm)</b>                        | 0.5     | 0.45    | 0.34    | 0.34    | 0.26    | 0.2     |
| <b>Espacement des pistes métalliques (µm)</b>            | 0.3     | 0.23    | 0.17    | 0.17    | 0.13    | 0.1     |
| <b>DRAMs</b>   |         |         |         |         |         |         |
| <b>Dimensions DRAM (bits)</b>                            | 64M     | 256M    | 1G      | 1G      | 4G      | 64G     |
| <b>Dimensions Cellule (µm<sup>2</sup>)</b>               | 0.56    | 0.22    | 0.14    | 0.09    | 0.036   | 0.014   |
| <b>Dimensions Chip (mm<sup>2</sup>)</b>                  | 280     | 400     | 445     | 560     | 790     | 1120    |
| <b>Bits/cm<sup>2</sup></b>                               | 96M     | 270M    | 380M    | 770M    | 2200M   | 6100M   |
| <b>Circuits Combinatoires</b>                            |         |         |         |         |         |         |
| <b>Microprocesseur Transistors/Chip</b>                  | 11M     | 21M     | 40M     | 76M     | 200M    | 520M    |
| <b>Circuits Combinatoires Transistors/cm<sup>2</sup></b> | 3.7M    | 6.2M    | 10M     | 18M     | 39M     | 84M     |
| <b>Vitesse d'horloge (MHz)</b>                           | 750     | 1250    | 1500    | 2100    | 3500    | 6000    |
| <b>Temps de propagation des portes (ps)</b>              | 16-17   | 12-13   | 10-12   | 9-10    | 7       | 4-5     |

Tableau 1.1 Tendances technologiques : progrès technologique, mémoires DRAM et logique combinatoire. Source : SIA National Technology Roadmaps for Semiconductors [SEMA '98].

Au vu de ce tableau, quelques commentaires s'imposent. Tout d'abord, il s'agit de la nécessité de contenir la consommation de puissance dans certaines limites. La consommation de puissance dynamique d'un circuit est définie par  $P = \alpha C V_{DD}^2 f$ , où  $\alpha$  représente l'activité électrique du circuit,  $C$  est la capacité en commutation, et  $f$  est la fréquence de fonctionnement. Du fait de la dépendance de la puissance dynamique avec  $V_{DD}^2$ , la façon la mieux appropriée de réduire la consommation de puissance passe par la réduction de la tension d'alimentation  $V_{DD}$ , tendance qu'on voit clairement dans le tableau 1.1, quand on passe d'une génération technologique à une autre. Mais cette réduction entraîne, comme on va le voir par la suite, une chute importante du niveau de fiabilité du circuit.

Ensuite, afin d'assurer une fiabilité acceptable pour l'oxyde de grille, il est nécessaire de maintenir un champ électrique plus petit que 5 à 6 MV/cm. De plus, si l'épaisseur de l'oxyde est inférieure à 2 nm, on voit une augmentation importante des courants de fuites due au phénomène de courant tunnel direct à travers la couche d'oxyde. A cause de l'importance croissante du courant tunnel direct, l'épaisseur de l'oxyde de grille se limitera probablement aux alentours de 2 nm, avec une marge limite de 1,5 nm pour les technologies les plus petites.

Les tensions de seuils ( $V_T$ ) sont estimées autour de la valeur  $V_{DD}/4$ . Mais pour une technologie plus courte que 0,1  $\mu\text{m}$ , la tension de seuil décroît fortement, ce qui entraîne de nouveau des courants de fuites très élevés. Cet effet est un des problèmes majeurs dans la course à une miniaturisation très poussée.

La diminution de l'épaisseur des fils métalliques ne suit pas le même facteur de miniaturisation que les transistors : il reste plus modérée. Les causes et les effets de cette diminution seront discutés en détail par la suite. Actuellement, les interconnexions métalliques constituent un des principaux soucis de la plus grande majorité des ingénieurs des procédés, des concepteurs de circuits intégrés et des créateurs d'outils CAO.

Concernant les mémoires, l'association de l'industrie des semiconducteurs [SEMA'98] prévoit qu'en l'an 2010 on produira des mémoires DRAM à 64 Gbits conçues en technologie de 70 nm. Selon [SEMA'98], la dimension d'une cellule DRAM sera de 10 % seulement la surface actuelle (voir figure 1.1.a). Les microprocesseurs connaîtront également une évolution très forte. Actuellement, à 40 M transistors par chips ou 10 M

transistors par  $\text{cm}^2$ , les microprocesseurs se retrouveront autour de l'année 2010 à 520 M transistors par chip ou l'équivalent de 84 M transistors par  $\text{cm}^2$ , soit environ un facteur de multiplication de 8. La vitesse d'horloge sera de l'ordre de 6 GHz pour une durée de temps de propagation moyenne par porte de 4 ps (figure 1.1 b). En conséquence, la charge d'un nœud de circuit en commutation sera ramenée aux alentours de la valeur de 0 fF !

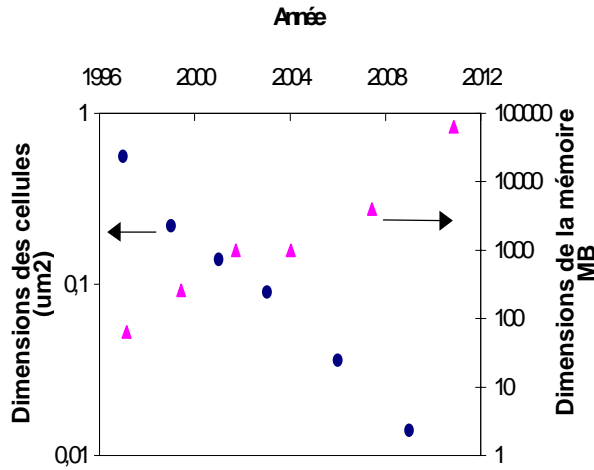


Figure 1.1. a) Tendances technologiques des mémoires DRAM

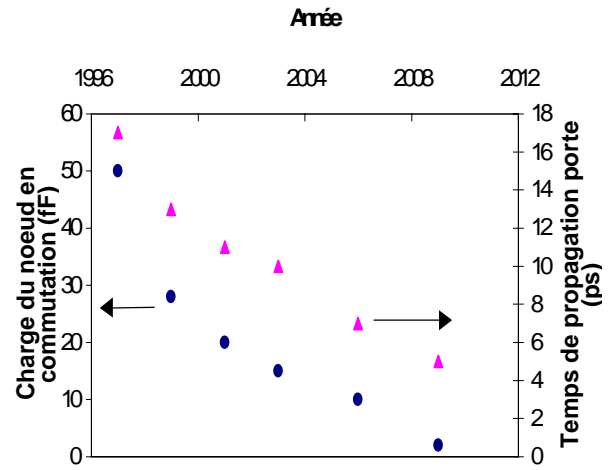
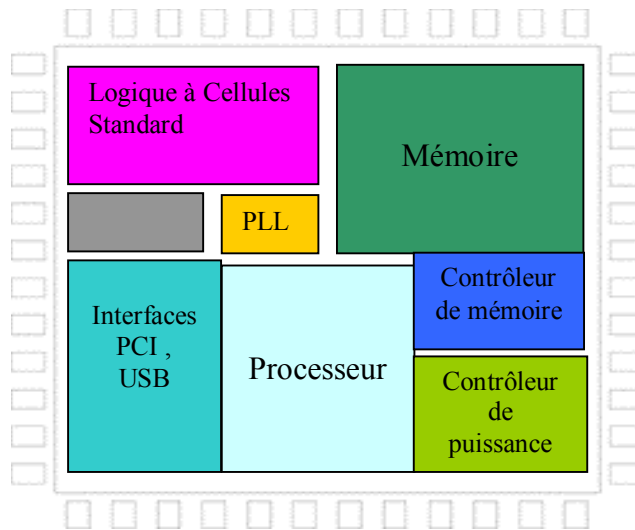


Figure 1.1. b) Tendances technologiques des circuits combinatoires

Un autre aspect important est celui de la discontinuité significative dans les performances d'un ASIC au-delà de 0,5  $\mu\text{m}$ . Cette discontinuité est due au changement dans la physique des structures sous-microniques. Pour illustrer cette affirmation, prenons l'exemple d'un système intégré typique comportant plusieurs blocs tel celui présenté dans la figure 1.2.



**Figure 1.2. Architecture typique de SOC (System-On-A-Chip)**

Plusieurs problèmes peuvent apparaître concernant les performances de ces blocs. Par exemple, la structure du processeur nécessite des bus de données de très grande vitesse, et on utilise souvent la logique dynamique afin d'assurer une vitesse de fonctionnement optimale. Les mémoires RAM contiennent un nombre impressionnant de transistors, et elles sont conçues en utilisant le transistor le plus petit que la technologie puisse fabriquer. Les blocs contenant de la logique aléatoire nécessitent des cellules de bibliothèque complexes et sont implémentés avec des connexions métalliques dont les retards de propagation sont de nature très complexe. Les blocs analogiques et les PLLs sont reliés au comportement analogique du transistor.

Pour chacun de ces exemples, les vrais problèmes apparaissent quand les outils de CAO ne modélisent pas les performances réelles du transistor et ne prennent pas en compte les retards de propagation dans les interconnexions. En effet, au dessous du  $0,5\mu\text{m}$ , la modélisation précise de la commutation du transistor, ainsi que la consommation de puissance et la relation entre les temps de propagation d'une porte et ses interconnexions deviennent très complexes. Les concepteurs de circuits intégrés trouvent aujourd'hui des différences de 70% entre les temps de propagation obtenus par simulations et par les mesures [KATZ'95].

Ce qui suit présente tout d'abord les implications de l'évolution technologique sur la réduction de la fiabilité du système. La plupart des problèmes qui seront discutés plus loin constituent des facteurs pouvant contribuer à bloquer l'avancée des technologies.

### 1.3. Les effets des technologies sous-microniques sur les interconnexions

#### 1.3.1. Les phénomènes RC

Ce paragraphe discute de l'importance de la modélisation précise des interconnexions dans une technologie sous-micronique, et de l'influence du temps de propagation dans les interconnexions sur les performances du circuit.

Pour les technologies au-delà de  $0,25\mu\text{m}$ , les retards de propagation ont un impact sur la vitesse de fonctionnement non seulement significatif mais souvent dominant. La figure 1.3 montre un chemin de propagation typique dans un circuit intégré dont les interconnexions sont modélisées par des couplages résistance-capacité (RC).

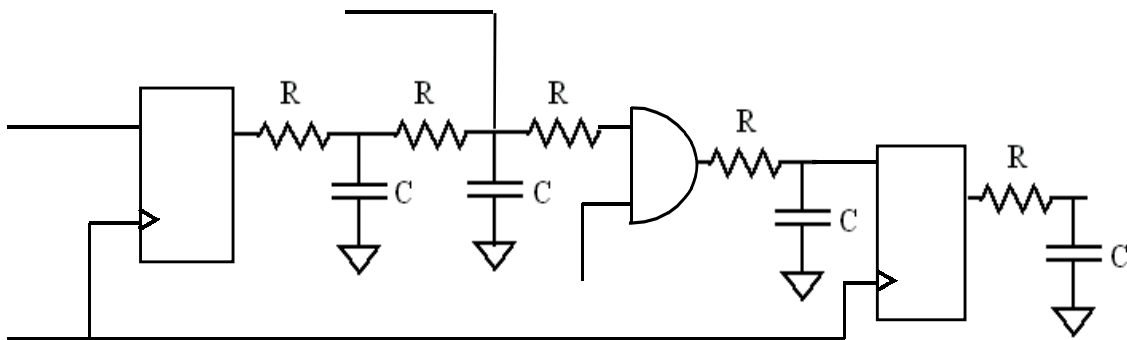


Figure 1.3 Exemple d'un chemin de propagation typique dans un circuit intégré.

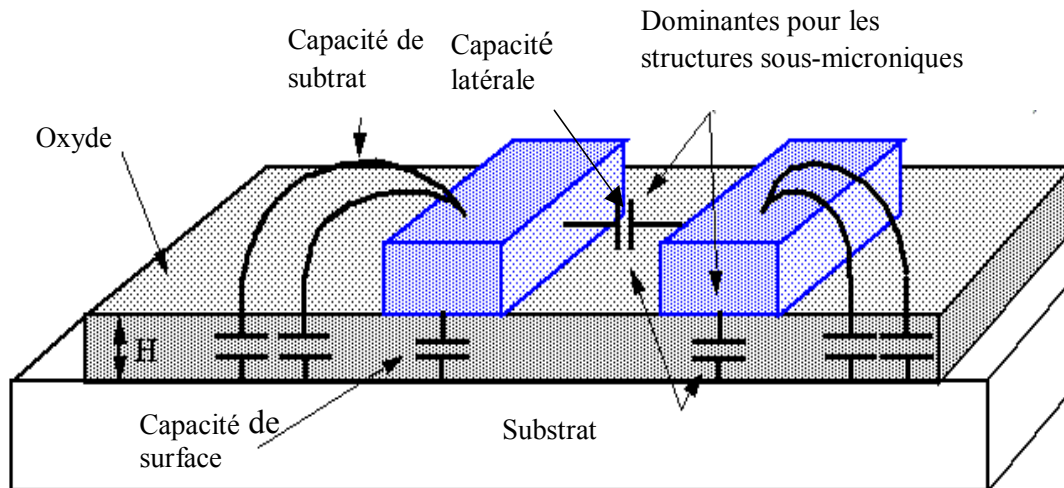
Le problème qui se pose aujourd'hui, et que les concepteurs de circuits intégrés rencontrent de plus en plus souvent, est l'augmentation du temps de propagation dans les interconnexions, phénomène quelque peu contradictoire à première vue avec la réduction des dimensions. Par exemple, le retard de type RC (résistance-capacité) de 1 mm de métal en technologie de  $0,5\mu\text{m}$  était de 15ps. Pour une technologie de  $0,1\mu\text{m}$ , ce retard augmente d'une façon significative, passant à 340 ps. La principale raison de l'augmentation du retard est, de fait, due à l'augmentation de la résistance du fil de métal.



Afin de garder la résistance du fil métallique à des valeurs acceptables, on doit éviter de réduire les dimensions du fil de la même façon que les dimensions du transistor. Mais, pour pouvoir augmenter la densité d'intégration (ce qui est le but de la miniaturisation), nous sommes obligés de réduire la largeur du fil. On ne peut donc, jouer que sur l'épaisseur du fil. Ceci amène à une augmentation du rapport hauteur/largeur (facteur d'aspect) qui mène par la suite à une augmentation du rapport entre la capacité latérale et la capacité totale. De plus, comme l'espace entre les fils diminue progressivement pour une plus grande densité d'intégration, la capacité latérale se trouve encore plus augmentée.

Pour conclure, *la capacité de couplage entre les fils métalliques augmente à cause de deux facteurs : l'augmentation du rapport d'aspect du fil métallique permettant de conserver la résistance des fils à des valeurs convenables, et la tendance naturelle de diminution de l'espace entre les fils.* Par ailleurs, l'augmentation du rapport d'aspect est aussi nécessaire afin de contenir la densité de courant dans certaines limites et d'éviter une dégradation importante de la fiabilité, causée par le phénomène d'électromigration. Cette augmentation de la valeur des capacités de couplage produit du bruit, tel que le crosstalk, sur lequel on reviendra plus loin.

Notons aussi un autre problème. Actuellement, les fonderies maîtrisent des technologies incorporant 5 ou 6 couches de métal, le calcul de la capacité équivalente d'un nœud du circuit devient complexe. La figure 1.4 montre une structure de circuit implémentée sur silicium avec la distribution des capacités de couplage entre ses différentes parties.



**Figure 1.4. Distribution des capacités dans les technologies sous-micronique.**

Une autre manière de réduire la résistance du fil métallique est l'utilisation de nouveaux matériaux ayant une conductivité plus élevée que celle de l'aluminium. C'est avec ce dernier que les fils de métal étaient fabriqués. Mais, au surplus de leur résistance croissante à petite dimension, les fils d'aluminium présentent une sensibilité relativement élevée à un autre phénomène gênant : l'électromigration. L'électromigration est associée à de fortes densités de courant et se manifeste quand un transport d'atomes de métal se produit au travers les fils électriques sous l'effet d'une différence de potentiel électrique local. Les fils de métal se détériorent, des trous apparaissent et des zones d'accumulation de métal se forment (pouvant même donner lieu à des courts-circuits avec les fils adjacents) avec comme conséquence la création de défauts physiques et/ou l'augmentation locale de la puissance dissipée. Ces deux facteurs ont un impact très important sur la fiabilité des circuits. Récemment, les fonderies IBM Corp., Motorola Inc. et Schaumburg, Ill ont proposé indépendamment des moyens de remplacer l'aluminium par le cuivre. Elles ont développé également des solutions technologiques afin de surpasser les éventuels problèmes soulevés par le dépôt de cuivre sur silicium. Entre autres avantages, le cuivre a une résistivité 40 à 45% plus faible que celle de l'aluminium, les capacités de couplage étant réduites car les dimensions des fils peuvent être diminuées sans risque et leur fiabilité est meilleure grâce à des effets d'électromigration moins marqués.

Néanmoins, cette solution est temporaire. Les dimensions des lignes métalliques pourront être diminuées dans un premier temps avec le même rapport que celui des

transistors. Par la suite, on se retrouvera aux limites de densités de courant dans le cuivre et on sera obligé d'appliquer les mêmes contraintes que dans le cas de l'aluminium, pour pouvoir continuer la miniaturisation. Les mêmes problèmes que ceux exposés précédemment réapparaîtront.

### 1.3.2. L'intégrité du signal (« Signal Integrity »)

Le terme **intégrité du signal** - ou « signal integrity » dans la littérature anglaise - désigne la qualité des signaux engendrés dans un circuit et qui pourraient éventuellement être altérés par des interférences dues aux autres éléments du circuit ou du système. On pourra aussi étendre cette notion pour prendre en compte toute interférence, y compris celles provenant du rayonnement cosmique. Des efforts de plus en plus importants sont concentrés aujourd'hui pour prévenir les différentes interférences dans les circuits intégrés. [GREE'98].

Dans le passé, « l'intégrité du signal » dans les circuits intégrés était un souci pour les concepteurs de circuits fonctionnant en radiofréquences (RF). Aujourd'hui, à cause d'une migration de plus en plus poussée vers les nanotechnologies, maintenir « l'intégrité du signal » devient un défi de plus en plus difficile à relever pour les concepteurs de circuits. Par exemple, en deçà de 0,5  $\mu\text{m}$ , « l'intégrité du signal » devient plus qu'un défi, à cause du temps de montée du signal très court et des dimensions des circuits, elles aussi réduites. A partir de 0,25  $\mu\text{m}$ , le phénomène ground bounce augmente considérablement à cause des temps de montée très courts et des éléments parasites du boîtier. Les concepteurs se trouvent face à un nouveau problème : trouver, localiser et éventuellement réduire les interférences possibles le plus tôt possible dans le flux de conception.

Les principales sources d'aggravation du problème de « l'intégrité du signal » sont identifiées depuis quelques années déjà. Il s'agit des interférences entre signaux voisins durant leur phase d'activité et des perturbations induites sur des éléments actifs du circuit par le bruit des lignes d'alimentation ou du substrat. Nous ajouterons ici, l'aggravation de la sensibilité des circuits aux radiations. Un problème majeur que nous discuterons dans un chapitre à part, et dont les différents acteurs de la microélectronique commencent seulement

à prendre conscience. Le problème des interférences entre les lignes voisines est lié tout d'abord à l'augmentation des vitesses de fonctionnement. Après tout, un fil métallique peut être vu comme une résistance pour des fréquences basses, comme une capacité pour les fréquences moyennes, comme une inductance pour les hautes fréquences, et comme une antenne aux très hautes fréquences. Le changement de paramètres géométriques a aussi un rôle prépondérant. La modification du rapport d'aspect (les lignes d'interconnexions deviennent plus hautes que larges) et le rapprochement des lignes augmentent substantiellement l'effet de capacité latérale et renforcent le couplage capacitif des lignes voisines. Le rapprochement des lignes et l'augmentation des vitesses de fonctionnement (i.e. quand  $dV/dt$  est élevé) renforce le couplage inductif. Ce phénomène s'ajoute au couplage capacitif et devient très significatif en deçà de 0,18 $\mu$ m. Le crosstalk devient actuellement d'une grande complexité pour le concepteur.

Malheureusement, la majorité des outils d'extraction des paramètres 3D ne réalisent pas l'extraction de l'inductance, donc il est fort probable que les circuits de l'avenir vont présenter de plus en plus d'erreurs provenant du manque d'une bonne estimation des éléments parasites. De plus, pour optimiser les performances, l'extraction des paramètres parasites doit être faite au niveau du layout. Ce qui amène, en supposant qu'on dispose des outils adéquats, à des simulations complexes, coûteuses en temps CPU et à de nombreux aller-retour entre « back-end » et « front-end », incompatibles avec des cycles de conception de plus en plus courts.

- **Le couplage du substrat.** Puisque le substrat et le « well » ont une résistance finie, tout courant les traversant induit une chute de tension. Cette chute de tension modifie le seuil du transistor se trouvant au-dessus de la région du substrat ou du well traversé par le courant. Ce problème s'aggrave avec la miniaturisation car la réduction de la dimension verticale augmente la résistance du substrat et du well.
- **Le ground bounce.** Le bruit sur les lignes d'alimentation induit par la commutation simultanée d'un grand nombre d'éléments peut causer des problèmes « d'intégrité du signal ». Le nombre croissant d'éléments actifs intégrés dans les circuits contribue à l'augmentation des courants. A cause des vitesses de fonctionnement élevées, le temps dans lequel les capacités du circuit sont chargées et/ou déchargées diminue, augmentant ainsi l'intensité des courants transitoires. La réduction des niveaux d'alimentation

diminue les charges stockés dans les capacités, ayant ainsi un effet bénéfique sur l'intensité du courant, mais en même temps elle mène à réduire le seuil des transistors et les marges de bruit, et ceci a pour résultat global une dégradation de « l'intégrité du signal ».

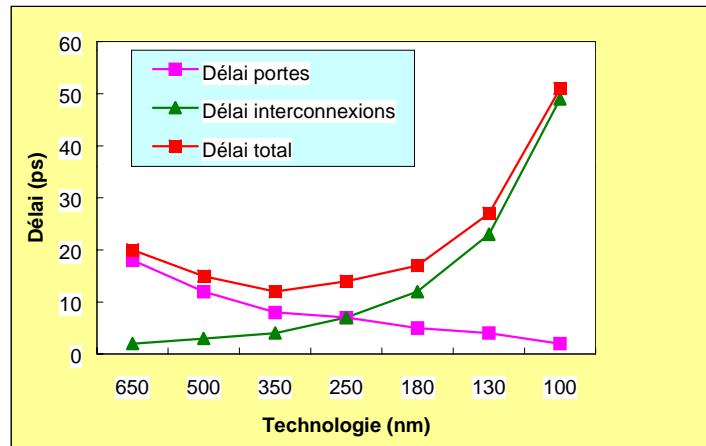
Des solutions pour traiter le problème de « l'intégrité du signal » commencent à apparaître. Par exemple, on peut réduire les interférences en mettant des contraintes sur l'activité du système, pendant la phase de synthèse logique, ou en limitant les valeurs  $dI/dt$ , ou  $dV/dt$ , ou le nombre de sorties qui commutent simultanément. On a aussi le choix au niveau du placement-routage, et on peut diminuer le crosstalk ou maîtriser l'imprécision du signal d'horloge. Cependant, les outils de conception d'aujourd'hui ne sont pas suffisants pour résoudre la complexité des problèmes qui se posent. De toute façon, la conception d'un circuit intégré ne pourra plus se faire comme elle se faisait dans le passé.

#### **1.4. L'impact des technologies sous-microniques sur le flot de conception des circuits intégrés**

Les technologies sous-microniques vont poser dans le futur, des problèmes importants aux concepteurs des circuits intégrés, aussi à cause de leur impact sur le flot de conception. Dans les technologies du passé, où le temps de propagation des portes est dominant, la méthodologie de conception est focalisée sur l'optimisation des performances en diminuant le retard de propagation des portes logiques. Aujourd'hui, deux aspects sont à prendre en compte : la modélisation de plus en plus complexe du temps de propagation dans les interconnexions et le problème de l'intégrité du signal, voire la fluctuation du retard d'une porte logique à cause du bruit.

La figure 1.6.a présente le flot de conception traditionnel qui sépare les deux principes de base dans la conception des circuits intégrés : la conception logique, qui produit la description « netlist », et la conception physique ou la production du layout, les retards d'interconnexion étaient estimés à travers des modèles. Par conséquent, les effets des interconnexions n'étaient déterminés avec précision qu'après le placement-routage,

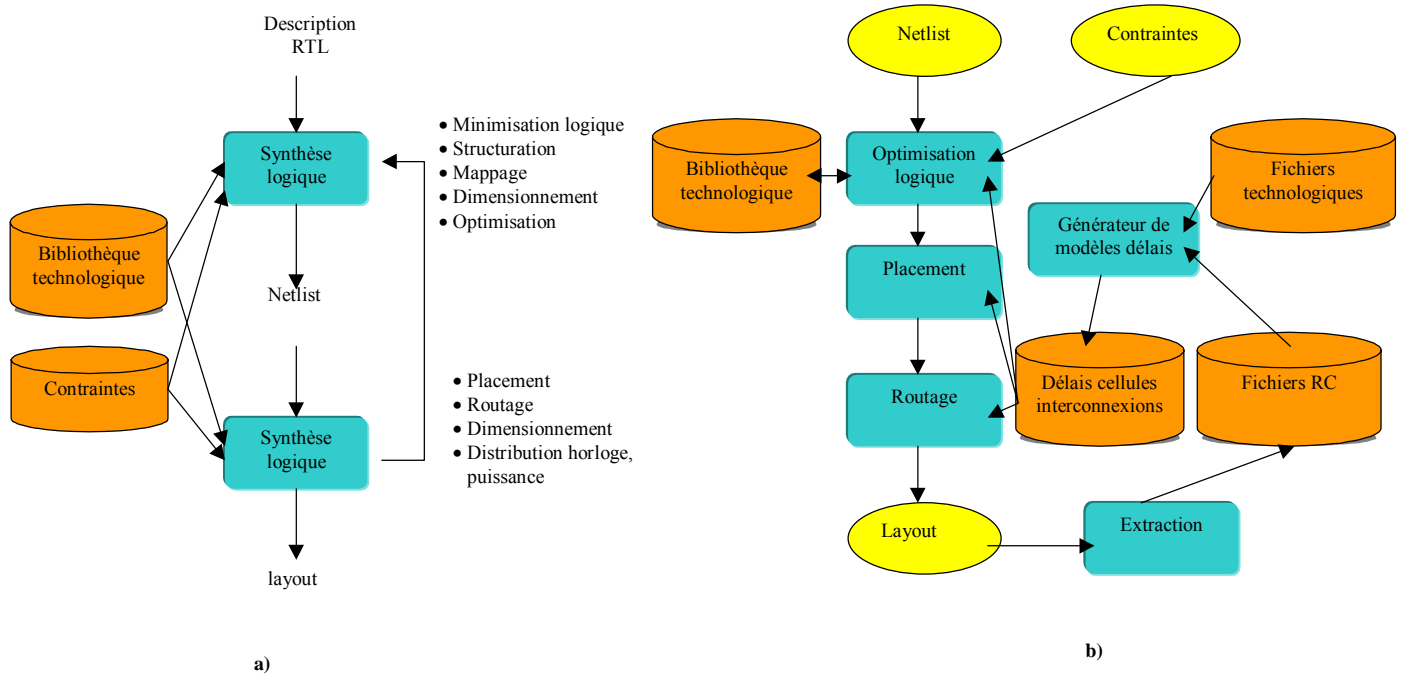
c'est à dire assez tard dans le flow de conception. Si les concepteurs détectaient des différences, une modification du dessin était entamée, et le placement-routage était refait. Tant que les retards dominants étaient ceux des réseaux de portes logiques, cette méthodologie fonctionnait convenablement. Mais, comme actuellement les retards des interconnexions deviennent dominants (voir figure 1.5), la méthodologie de dessin doit changer.



**Figure 1.5. Les effets des technologies nanométriques sur les retards du circuit.**

Dans la figure 1.6.b on présente le flow de conception modifié pour les technologies sous-microniques, où les deux principes sont plutôt combinés.

La synthèse logique transforme la description HDL en un graphe de cellules de bibliothèque et des connexions entre celles-ci. Ce graphe est aussi nommé « netlist ». À travers la synthèse logique, le circuit est optimisé en fonction des contraintes de l'utilisateur (surface minimale, contraintes de temps de propagation, de dissipation de puissance, d'activité, etc.). Le dessin physique est la transformation du netlist en masques qui seront ensuite utilisés pour la fabrication du circuit. Les circuits à contraintes fortes, nécessitent plusieurs itérations entre la synthèse logique et le dessin physique avant d'arriver à une implémentation effective. Pendant la synthèse logique, le modèle d'interconnexions est prédit - et non pas calculé -, donc les capacités des interconnexions sont approximatives, et les modèles RC résultant sont loin de la réalité.



**Figure 1.6 Flow de conception : a) technique traditionnelle, b) technique modifiée pour les technologies sous-microniques**

L'opération de placement-routage prend en compte ce modèle pour le calcul des retards de propagation du circuit. Si les retards de propagation des interconnexions deviennent plus importants que les retards des portes, la possibilité que l'excès de capacitance viole les contraintes temporelles augmente pendant la phase de routage. Alors, les concepteurs découvriront probablement ces problèmes seulement après la nouvelle extraction des paramètres RC. A ce moment, les outils réactualisent le netlist synthétisé avec les nouveaux éléments extraits, et une nouvelle opération d'optimisation est lancée. Malheureusement, si la capacité estimée pendant la synthèse logique est très loin de la réalité, le nombre d'itérations entre la synthèse logique et l'extraction des paramètres peut devenir très grand, faute principalement d'une bonne optimisation itérative de l'outil de synthèse. La phase de conception risque de devenir un goulot d'étranglement dans le « time-to-market » d'où le besoin de nouveaux outils capables de faire une synthèse globale, c'est à dire synthèse logique et placement-routage simultanément deviennent nécessaires, en admettant, bien sûr, que ces outils soient capables de prendre en compte les phénomènes complexes régissant les retards de propagation et l'intégrité du signal dont nous avons discutés précédemment. Tant que ces outils ne seront pas disponibles un risque

demeurera pour le concepteur de voir, après fabrication, que son circuit ne respecte pas les spécifications.

### **1.5. Les effets des technologies nanométriques sur les modèles de fautes et le test de fabrication**

Les procédés de fabrication des circuits intégrés étant imparfaits, ils introduisent fréquemment des défauts. Ainsi, même en supposant que le concepteur a parfaitement maîtrisé tous les phénomènes se produisant dans un circuit intégré et a produit un dessin correct, il se trouve que certains circuits issus de la chaîne de fabrication ne fonctionnent pas selon les spécifications, à cause de certains défauts introduits par le processus de fabrication.

L'introduction des technologies nanométriques dans le procédé de fabrication déjà imparfait auparavant, présente un défi très important pour l'identification de nouveaux défauts. Les défauts commencent à avoir une très grande variété de formes, des défauts ponctuels localisés (matériel manquant, ou accumulation de matériel), ou des défauts affectant des surfaces plus grandes (changement de capacité due à une implantation ionique défectueuse). Les défauts sont imprévisibles en position et en effet, et le procédé qui les crée a un très grand nombre de variables.

La densité élevée d'intégration rend le procédé de fabrication imprécis. La simulation des fautes et l'ATPG deviennent très longues et peu pratiques. De plus, les ouvertures dans l'oxyde, les vias (connexions entre les différents niveaux de métallisation) manquantes ou mal formées induisent un comportement aléatoire, impossible à prédire. D'autres comportements imprévisibles sont provoqués par les crosstalks, ou l'apparition d'inductance sur les fils longs. Comportement imprévisible veut dire que le modèle de fautes sera incomplet, et des approches probabilistes deviennent nécessaires ; ce sont des approches qui nécessitent une activation répétitive des défauts. Ainsi, la modélisation de fautes passe du domaine discret au domaine continu [AITK'99].

On distingue deux types de défauts : **les défauts globaux** produits par un décalage de certaines valeurs paramétriques affectant toute une partie, voire même l'ensemble de la



tranche de fabrication, et **les défauts localisés** dus le plus souvent à des particules de poussière qui se déposent sur un niveau de masque et produisent un excès ou un manque de matériel sur une couche du circuit intégré.

A. **Les défauts globaux** sont facilement détectables, surtout quand le décalage de la valeur paramétrique est important. Mais, dans certains cas, quand le décalage paramétrique est faible, le défaut peut se manifester comme une faute de retard du chemin de propagation, ce qui nécessite des conditions de test très particulières (sensibilisation robuste de certains chemins) pour être détectable.

B. **Les défauts localisés** sont très variés, incluant les courts-circuits entre deux lignes, le circuit ouvert (ou coupure d'une ligne), le contact coupé, le contact résistif, le contact partiel (exemple court-circuit grille-oxyde, grille-source, grille-drain ou source-drain). L'impact du défaut sur le fonctionnement du circuit dépend du type de défaut, de ses paramètres (par exemple, la résistance d'un court-circuit), des éléments affectés par ce défaut, des chemins par lesquels les effets du défaut se propagent, des conditions de fonctionnement du circuit, tels que vecteur ou transition appliquée aux entrées, vitesse de fonctionnement, niveau d'alimentation, voire même température. Par exemple, les courts-circuits placés aléatoirement, seront un souci très sérieux à l'avenir, principalement à cause de l'augmentation des niveaux de métallisation et la réduction des dimensions. Ainsi, les effets de courts-circuits seront de plus en plus complexes, augmentant la variété des comportements d'un court-circuit [AITK 99]. La résistance critique d'un défaut (en deçà de laquelle un défaut ne peut plus être détecté) commence à être variable avec la fréquence de test. Alors, l'effet de ce défaut sur le fonctionnement du circuit dépend de la fréquence de fonctionnement du circuit. Les ouvertures donnent le plus de souci aux ingénieurs de test, car dans certaines situations, le circuit continue à fonctionner correctement mais seulement à basse fréquence. Dans d'autres situations, des fautes de type collage logique ou fautes de retards peuvent se manifester.

Déterminer avec précision l'impact du défaut sur le fonctionnement du circuit nécessite des simulations complexes, le plus souvent du type SPICE. Néanmoins, ce type de simulation ne peut pas être envisageable comme moyen de génération de vecteurs de test à cause d'un temps de calcul excessif. On utilise par conséquent, des modèles de fautes permettant la simplification de la méthode de génération des vecteurs de test et la simulation. Dans le passé, le modèle de collage logique a été imposé très rapidement

comme un standard pour la génération automatique de vecteurs de test et pour la simulation des fautes. Les causes de son succès sont, en fait, la simplicité du modèle permettant un temps réaliste pour la génération des vecteurs de test, même pour des circuits complexes, ainsi qu'une bonne couverture des défauts physiques. L'obtention d'une bonne couverture des défauts physiques, qui induisent très souvent un comportement différent de celui des collages logiques, pourrait s'expliquer par le fait que le test des fautes de collage logique contrôle chaque nœud du circuit (c'est à dire applique les deux valeurs logiques '0' et '1' sur chaque nœud), et permet d'observer chaque nœud à chacune des deux valeurs logiques (le nœud étant à '1', on sensibilise au moins un chemin connectant le nœud avec les sorties, et même chose le nœud étant à '0'). Ces conditions de travail sont nécessaires pour l'excitation et la propagation de tout autre défaut, et il n'est donc pas étonnant que le test du collage logique ait permis dans le passé une bonne couverture des défauts réels. Néanmoins, cette couverture reste incomplète, et d'autres types de test, tels que les tests  $I_{DDQ}$  et des tests de retard ont été utilisés en complément, surtout dans les applications exigeant un niveau de qualité élevée.

Cependant, on constate actuellement une lente dégradation de cette situation. Tout d'abord, le test  $I_{DDQ}$  devient peu à peu inutilisable. Ceci est dû à l'augmentation des courants de fuites, qui accompagnent le processus de miniaturisation [WILL'99], et qui masquent les courants induits par les défauts. Par ailleurs, les vitesses de fonctionnement de plus en plus élevées augmentent le pourcentage de défauts qui se manifestent en fautes de retard [AITK'99]. Pour des fréquences d'horloge dépassant 1 GHz, même les défauts induisant des petites variations de retard se manifesteront comme fautes de retard proprement dite. Or une faute de retard localisée (par exemple, faute de retard associé à une porte) peut facilement rester indétectable par un test de collage logique, car son excitation nécessite deux vecteurs de test et non un seul. De plus, sa propagation nécessite la sensibilisation d'un chemin maximisant ce retard, et non un chemin quelconque liant le site du défaut aux sorties.

Dans la suite nous parlerons d'un autre type de défaut qui pourrait être difficile à détecter. Il s'agit de défauts produisant un niveau indéterminé (c'est à dire niveau proche des seuils des portes logiques) sur le nœud affecté. Ainsi, selon le seuil de chaque porte logique, ce niveau sera interprété par certaines portes comme niveau logique '0' et par d'autres comme niveau logique '1'. Cependant, ces erreurs ont le plus souvent un

comportement de type erreur temporelle. Considérons par exemple un inverseur et supposons que son entrée reçoit un niveau logique '0' franc et après un niveau logique '1' dégradé, légèrement supérieur au seuil de l'inverseur.

Si on applique à l'entrée de l'inverseur le niveau '1' dégradé, la sortie de l'inverseur prendra la valeur '0' car son niveau d'entrée est supérieur à son seuil.

Cependant, si auparavant, la sortie de l'inverseur était à '1', la transition logique de sortie de '1' à '0' sera lente à cause du niveau '1' dégradé en entrée de l'inverseur. On observe donc, un comportement de type faute de retard, bien plus difficile à détecter qu'un collage logique.

Les fautes de retard ont été modélisées par deux modèles : les fautes de retard de portes (quand une porte logique fonctionne plus lentement que prévu), les fautes de retard d'un chemin de propagation (quand la transition logique est ralentie tout au long d'un chemin de propagation).

Les principales sources de fautes de retard :

- Un nombre important des fautes de retard peut rester non détecté lors d'un test de type collage logique. Etant donné que les fautes de retard de la porte logique ne considèrent pas de modifications du retard sur l'ensemble d'un chemin, on peut manquer des phénomènes de retard subtils qui affecteront les technologies sous-microniques. Ainsi, le test de fautes de retard du chemin de propagation devient indispensable [AITK'99].
- La génération des vecteurs de test pour les fautes de retard est un processus très consommateur de temps CPU, spécialement si on considère les fautes de retard du chemin de propagation. Ce test nécessite la prise en compte d'un nombre de chemins de propagation très élevé. De plus, avec les outils modernes de synthèse qui, en optimisant la longueur du chemin de propagation critique, produisent des circuits balancés ayant une majorité des chemins proches en temps de propagation du chemin critique, on est obligé de tester le retard d'une majorité de chemins. Pour des circuits complexes le nombre de chemins à tester devient vite énorme. Le temps de CPU pour l'ATPG et le temps de test deviennent vite irréalistes.
- Les choses se compliquent encore plus, si l'on considère les différentes interférences dont nous l'avons discuté au paragraphe 1.3.2. Le crosstalk produit un retard maximal

quand les deux lignes couplées effectuent deux transitions dans des directions opposées, l'une de '1' à '0' et l'autre de '0' à '1'. D'autre part, le couplage par les lignes d'alimentation ou par le substrat se manifeste quand le circuit se trouve au maximum de son activité, alors qu'au repos ces phénomènes disparaissent. Ainsi, ces phénomènes se manifestent aussi comme des défauts de « timing », puisque leurs effets disparaîtront si on réduit la vitesse de fonctionnement de manière à s'éloigner de la phase d'activité maximale. Ces interférences, même si elles sont maîtrisées par le concepteur qui aura réduit leur intensité et/ou les aura pris en compte pour déterminer la fréquence d'horloge, restent potentiellement présentes dans le circuit. Si une faute de retard se produit dans le circuit, la situation de pire cas peut se présenter dans les conditions de fonctionnement qui combinent l'activation du défaut et l'activation des interférences (crosstalk). Ceci veut dire que, pour détecter les fautes de retard, l'ATPG doit générer des paires de vecteurs d'entrée qui activent à la fois les fautes de retard et les crosstalks ou les bruits de type ground bounce. Evidemment, ce test est irréaliste, même pour des circuits de complexité moyenne.

On pourra conclure que le passage vers des vitesses de fonctionnement très élevées augmentent le pourcentage des fautes de « timing » et une partie de plus en plus grande de ces défauts restera indétectable par le test de fabrication. Dans ce contexte, la seule solution permettant de ne pas produire un nombre croissant de circuits ayant un fonctionnement erroné de façon intermittente consiste à concevoir les circuits en utilisant des techniques de tolérance aux fautes de type « timing ». L'option 'tolérance aux défauts' semble devenir une nécessité à cause d'une autre source de dégradation de la fiabilité des circuits intégrés nanométriques : les fautes transitoires. Ce problème est discuté en détails dans la section suivante.

## **1.6. L'environnement radiatif et ses conséquences sur les circuits intégrés nanométriques**

Ce paragraphe décrit les types de radiations présentes dans l'espace et à la surface terrestre, et leurs interactions avec les matériaux composant les circuits intégrés submicroniques. Les événements résultant de ces interactions sont présentés avec leurs conséquences. Sont évoqués aussi des moyens de protection contre les phénomènes radiatifs agissant sur les différentes structures analysées.

### **1.6.1. Les différentes interactions**

Les particules interagissant avec les circuits intégrés sont essentiellement les particules légères, comme les photons, les neutrons, les électrons, les protons ainsi que les ions lourds. Les principales sources d'émission de ces radiations sont le soleil et les zones de piégeage de la magnétosphère et les rayons cosmiques. L'influence du soleil sur la terre s'exerce par une activité continue, le vent solaire, et de façon plus intense par les éruptions solaires. Le flux de ces particules légères augmente d'environ 50% pendant les périodes d'activité solaire. Normalement, le champ magnétique terrestre dévie les trajectoires des particules chargées en mouvement, en protégeant l'atmosphère terrestre contre les éruptions solaires et le flux cosmique.

Les plus importantes sources de particules énergétiques qui concernent les concepteurs des circuits pour le domaine spatial sont les protons et les électrons piégés dans la ceinture de Van Allen, les ions lourds piégés dans la magnétosphère, les protons et les ions lourds qui proviennent des radiations cosmiques et de l'activité du soleil.

1. **Les interactions avec les photons** se manifestent sous trois formes : l'effet photoélectrique, l'effet Compton et la matérialisation. Les différences entre ces interactions résultent de la valeur de l'énergie incidente des photons interagissant avec les matériaux.

2. **Les interactions avec les particules chargées.** Il s'agit des protons, des particules alpha et des ions. Plus les particules ont une masse et une énergie importante, plus elles provoquent de dommages. Les protons s'avèrent avoir un pouvoir ionisant insuffisant pour provoquer des effets directs mais ils peuvent produire des interactions nucléaires en interagissant avec les noyaux rencontrés. Les particules chargées (ions et particules alpha) sont les plus dangereuses ; elles sont responsables de SEEs (Single Event Effects dans la littérature anglaise) tels que les latch-ups et les SEU (Single Event Upsets) et elles contribuent également à l'effet de dose cumulée. Néanmoins, dans les nouvelles technologies submicroniques utilisées dans les applications spatiales, on trouve de plus en plus de SEUs dues à des protons.
  
3. **Les interactions avec les neutrons.** Les neutrons ne produisent pas de façon directe des SEEs ou des effets de dose cumulée car ils sont électriquement neutres. Cependant, ils ont des effets indirects importants. Le neutron est une particule ayant une masse, mais pas de charge électrique, il ne peut être décéléré ou arrêté que lors d'une collision avec un noyau. Lors de la collision, des phénomènes différents peuvent survenir : phénomènes qui ont pour conséquences soit la déformation du réseau matériel avec des atomes qui quittent la maille cristalline ; émissions secondaires de rayons  $\gamma$  ou de particules alpha selon l'énergie incidente du neutron. En 1984, certains auteurs ont prédit que les neutrons atmosphériques vont créer des SEEs dans les circuits intégrés [SILB '84]. Mais ce fut seulement en 1992 que les premiers SEE dus aux neutrons ont été observés dans les mémoires SRAMS à bord d'avions civils et militaires [TAB '92]. Depuis, de nombreux SEUs ont été enregistrés dans différentes applications, et les événements induits par les neutrons commencent à devenir un problème majeur pour les circuits microélectroniques.

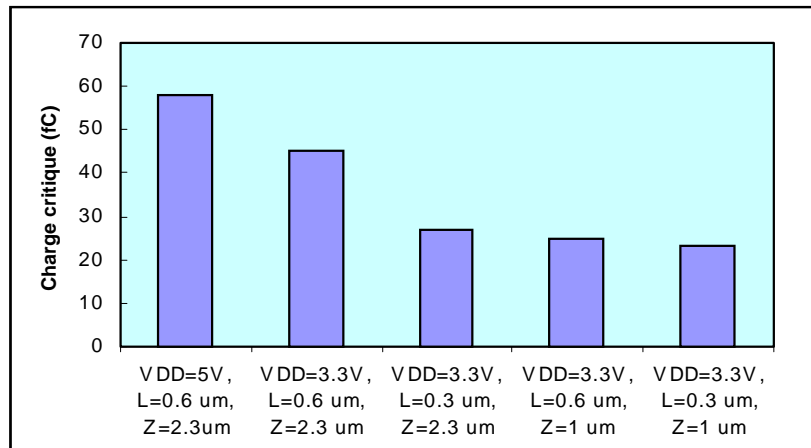
## 1.7. Les effets du rayonnement sur les circuits intégrés

### 1.7.1 Terminologie

Le nombre de particules arrivant par unité de surface et de temps est le **flux**, donné en particules/cm<sup>2</sup>/s. L'intégration de ce flux dans le temps donne la densité de particules ou fluence ( $F$ ) exprimée en particules/cm<sup>2</sup>. Une particule interagissant avec la matière lui transmet tout ou partie de son énergie. La quantité d'énergie déposée par la particule par unité de longueur de trajectoire est appelée pouvoir d'arrêt ou **LET (Linear Energy Transfert)**. Le LET dépend de la nature de l'ion incident, de la matière diffusante et de l'énergie de la particule. Il est habituellement donné en MeV cm<sup>2</sup>/g.

Deux paramètres sont utilisés pour quantifier de façon empirique la sensibilité des circuits intégrés vis-à-vis des particules. Il s'agit de la **section efficace** et du **seuil de sensibilité** ou LET seuil. Le seuil de sensibilité est lié à la charge minimale déposée par une particule nécessaire pour perturber le fonctionnement d'un circuit. Le LET seuil est défini comme étant le LET minimum que doit induire une particule pour perturber le fonctionnement d'un circuit. La section efficace  $\sigma$  est donnée par le rapport entre le nombre de perturbations et la fluence reçue. Plus la section efficace est élevée et/ou plus le LET seuil est faible, plus un composant est sensible. La connaissance de l'environnement (spectre du LET), d'une part, et du composant ( $\sigma = \sigma(\text{LET})$ ), d'autre part, permet de prédire le taux d'événements.

La **charge critique (Qcrit)** est un paramètre de circuit qui dépend de la capacité et de la résistance du nœud atteint par la particule énergétique. En effet, la charge critique est la quantité minimale de charge collectée par un nœud sensible nécessaire pour la production d'un « upset ». Cette charge diminue progressivement avec la miniaturisation des technologies et l'abaissement des tensions d'alimentation (figure. 1.7.). De plus, elle présente une dépendance avec la fréquence de fonctionnement du système.



**Figure 1.7. Evolution de la charge critique pour des paramètres technologiques différents (L = longueur du canal CMOS, Z = largeur du canal CMOS) [DET' 96].**

L'évolution technologique en matière d'intégration des circuits et de diminution de taille des transistors a entraîné une diminution de la charge critique. De quelques picocoulombs, la charge critique diminue aujourd'hui jusqu'à 10 femtocoulombs.

## 1.8. Les différents types de défaillances dans les circuits intégrés

Les phénomènes qui ont pour origine l'impact des particules sur les circuits sont de trois types : dérive des paramètres électriques et électroniques due à la dose cumulée (tension de seuil pour les transistors CMOS, ou modification du gain des transistors  $\beta$  pour les technologies bipolaires), basculement aléatoire d'un bistable (upset) et mise en conduction d'un thyristor parasite (latch-up). Tous ces effets ne sont possibles dans tous les circuits, soit que la charge critique correspondante à l'effet est trop importante, soit que la technologie de fabrication du circuit est plus élaborée (par exemple : substrat imbriqué ou SOI), soit du fait de la dimension des composants du circuit (les technologies nanométriques sont plus particulièrement sensibles).

### 1.8.1. La dose cumulée

On appelle 'effet de la dose cumulée' l'apparition de charges électriques parasites dans l'oxyde de grille. Le résultat en est un abaissement de tensions de seuil des transistors.



L'évolution vers les technologies nanométriques apporte quelques améliorations sur l'effet de la dose cumulée. En effet, à cause de la réduction de la tension d'alimentation  $V_{DD}$  (qui entraîne une diminution de la tension de seuil du transistor CMOS), ainsi qu'à la diminution de l'épaisseur de la couche d'oxyde de grille, les phénomènes liés à la dose cumulée sont réduits.

### 1.8.2. SEU et Multiple-Bit Upset

On appelle SEU (Single Event Upset) un effet d'ionisation dans le matériau résultant du passage d'une particule chargée dans un circuit, phénomène qui peut causer un « upset » (aussi appelé « bit-flip » dans la littérature anglaise). L'upset est en fait le changement de l'information contenue dans un point mémoire en son inverse [PET '83]. La particule incidente produit des paires électrons-trous le long de sa trajectoire. Cette quantité de charge sera collectée au niveau d'une jonction PN bloquée suivant trois modes (figure 1.8.) : charges directement collectées au niveau de la zone de déplétion, charges collectées par effet de « funneling » et de charges collectées par diffusion. Les deux premiers modes de collection de charges entraînent une impulsion de courant immédiate, par contre, les charges collectées par diffusion ont un effet retardé.

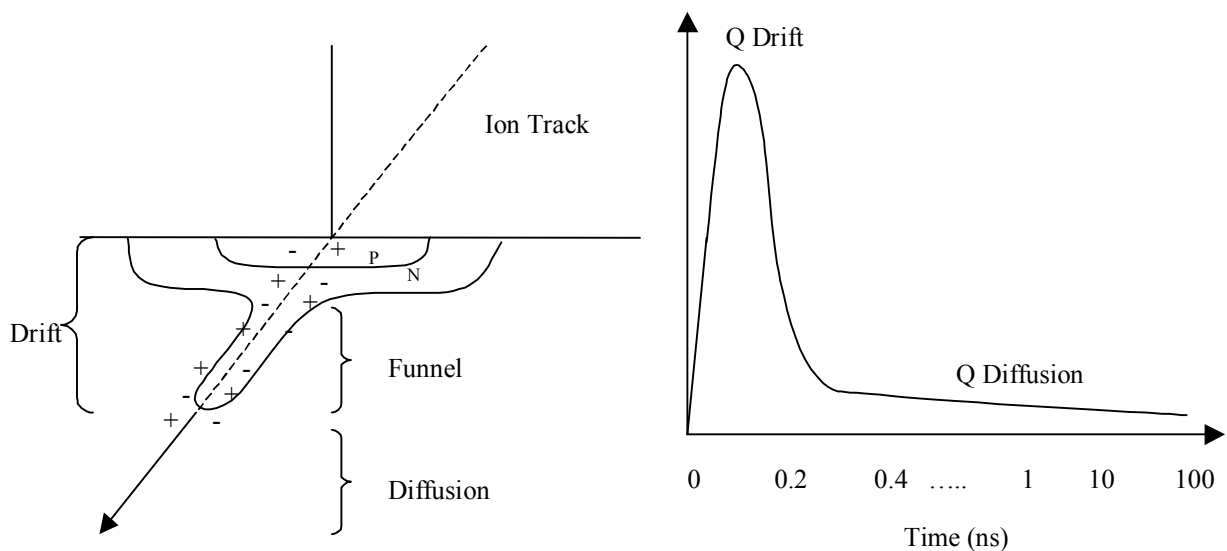
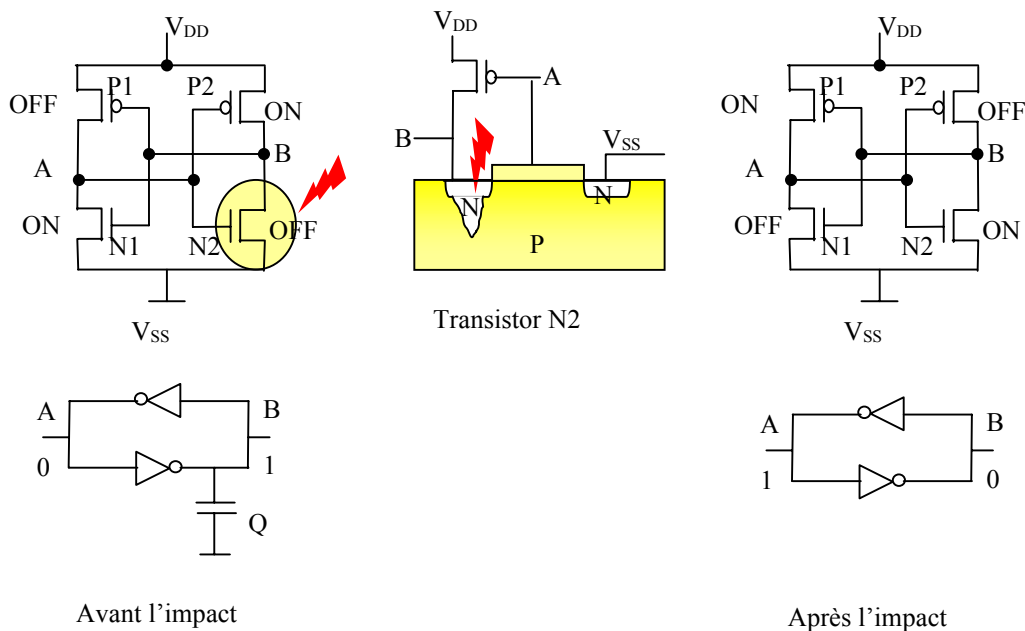


Figure 1.8. Création d'une impulsion de courant au niveau d'une jonction PN bloquée.

Des simulations «mixed-mode» montrent qu'un transistor bloqué de type NMOS est plus sensible à ce phénomène qu'un transistor PMOS [DET '96].

Les upsets se produisent dans les points mémoire, quand la charge électrique collectée après l'impact d'une particule, dépasse la charge critique nécessaire pour changer l'état logique de la cellule. Alors, la cellule de mémoire change d'état logique et l'information est perdue. Considérons le cas d'un bistable CMOS, schématisé dans la figure 1.9. Le circuit est conçu de sorte qu'il a deux états logiques stables, l'un qui représente un "0" et l'autre qui représente un "1". Dans chacun des états logiques stables, deux transistors sont dans un état passant ("ON") et les deux autres dans un état bloquant ("OFF"). Un upset se produit, par exemple, quand une particule énergétique touche le drain du transistor N2 bloqué. L'impulsion de courant produite dans le nœud B change son potentiel, cette variation est transmise à l'autre moitié de la cellule bistable.



**Figure 1.9 Mécanisme d'upset pour un point mémoire.**

Si cette impulsion produit une variation de tension suffisante pour provoquer la modification de l'état du transistor P1, le nœud A changera d'état. Si cet état est maintenu suffisamment longtemps pour que l'effet de la mémorisation intervienne, il y aura un basculement logique (figure 1.9) [PICK '83]. L'upset est un phénomène non destructif et

réversible (par exemple, le point mémoire pourra être corrigé par le processus normal d'écriture).

Tout composant électronique possédant des points de mémorisation est sensible aux phénomènes d'upset (mémoire dynamique ou statique, microprocesseurs et périphériques de microprocesseurs, bascules, latch...).

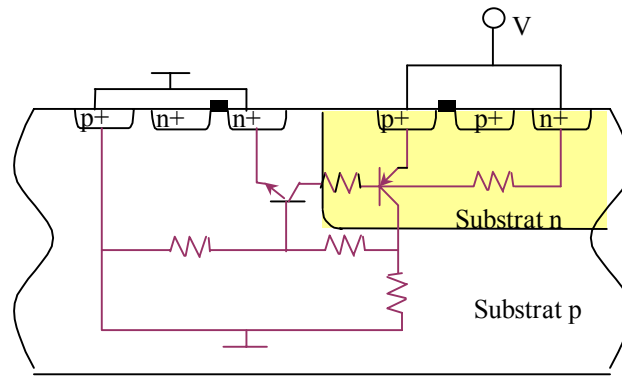
Traditionnellement, seulement les ions lourds étaient considérés comme la cause des upsets. Les technologies à forte densité d'intégration et la diminution de la taille des transistors a entraîné une diminution de la charge critique (la charge critique peut atteindre aujourd'hui 10 fC).

La sensibilité aux upsets s'accroît de telle sorte que les protons soient susceptibles d'en produire dans les circuits fonctionnant en environnement spatial, augmentant ainsi leur taux de plusieurs ordres de grandeur du fait d'un plus grand nombre de protons provenant du vent solaire.

Les Multiple-Bit Upsets (MBU) apparaissent du fait des charges qui diffusent dans le substrat, pouvant ainsi être collectées par plusieurs éléments du circuit. C'est l'exemple des mémoires DRAMs et SRAMs à quatre transistors où plusieurs cellules peuvent changer d'état logique simultanément. Les circuits peuvent être rendus moins sensibles aux particules énergétiques qui génèrent des MBU par l'ajout de zones fortement implantées dont le type est opposé au type des nœuds sensibles.

### 1.8.3. Latch-up

Le phénomène de Latchup ou « Single Event Latch-up » ou SEL, est la mise en conduction d'un thyristor (structure NPNP) parasite présent dans tous les circuits intégrés CMOS (figure. 1.10).



**Figure 1.10. Mise en évidence de la structure parasite NPNP dans un inverseur CMOS sur silicium.**

Le latch-up crée un chemin de conduction parasite direct entre la masse et l'alimentation, ce qui induit un échauffement supplémentaire important dans le circuit et entraîne une forte augmentation de la consommation. Ces phénomènes peuvent parfois causer la destruction du circuit, d'où l'utilisation de dispositifs de coupure d'alimentation en cas d'une forte augmentation de courant [MA' 89].

Le phénomène de latch-up joue un rôle très important surtout pour les systèmes spatiaux. L'évolution technologique augmente la sensibilité au latch-up des futurs circuits intégrés, et le phénomène s'avère de plus en plus gênant dans les circuits à faible tension d'alimentation et à faible consommation. Parfois, le phénomène de latch-up peut entraîner de faibles changements de consommation de courant dans les circuits. Alors, les tests logiques ou les tests  $I_{DDQ}$  ne peuvent pas les détecter parce que le circuit ne présente pas d'erreurs ni de fautes détectables dans un test de courant.

Les mémoires sont les parties considérées comme les plus sensibles aux phénomènes de type SEU. Dans un circuit logique, la charge collectée sur un nœud critique (drain d'un transistor MOS bloqué) est rapidement évacuée vers les  $V_{DD}$  ou la masse, et l'état logique du circuit n'est pas influencé. Une cellule de mémoire, en revanche, peut être facilement perturbée par une impulsion transitoire induite par l'impact d'une particule ionisante, car cette impulsion peut se propager au long de la boucle asynchrone composant la cellule de mémorisation, et changer l'état de la cellule (figure 1.9). Pour cette raison, les techniques de durcissement par conception, utilisées dans le domaine spatial visent à protéger les points de mémorisation.

Les mémoires RAMs sont protégées le plus souvent par un code détecteur/correcteur d'erreurs, tandis que les points de mémoires distribués (latches et bascules) sont protégés en utilisant des cellules durcies.

## **1.9. Moyens de réduction des impacts des SEEs**

Un moyen de faire face aux risques des radiations consiste à minimiser la probabilité d'apparition des effets qui leurs sont dus. Trois types d'actions permettent cette prévention : le blindage, dont le but est de stopper le passage des particules, le durcissement au niveau des procédés de fabrication ou de conception de la cellule, et la tolérance aux fautes au niveau circuit ou au niveau système.

### **1.9.1. Le blindage**

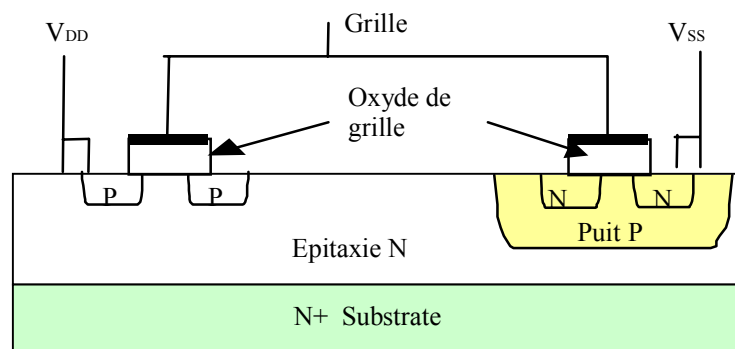
Il a pour but d'éliminer, ou du moins de réduire, le flux et l'énergie des particules parvenant jusqu'aux composants à l'intérieur du satellite. On utilise une feuille d'aluminium de quelques millimètres d'épaisseur. Les électrons sont considérablement ralentis par la présence du blindage, mais ce dernier devient complètement inefficace contre les rayons cosmiques qui peuvent avoir des énergies de plusieurs MeV, voire de l'ordre du GeV [BOUR'91]. Les blindages sont particulièrement inefficaces contre les neutrons pour lesquels plusieurs mètres de béton seraient nécessaires pour que leur énergie soit diminuée sensiblement.

### **1.9.2. Le durcissement des composants**

Deux approches permettent de diminuer la sensibilité aux rayonnements. L'une, au niveau technologique, utilise des méthodes de fabrication de composants plus robustes ; l'autre agit soit au niveau de la conception du système entier où le durcissement se réalise plutôt grâce à des architectures tolérantes aux fautes transitoires, soit par la modification de la taille des dispositifs sensibles aux SEE (tels les registres, les points mémoire, ...)

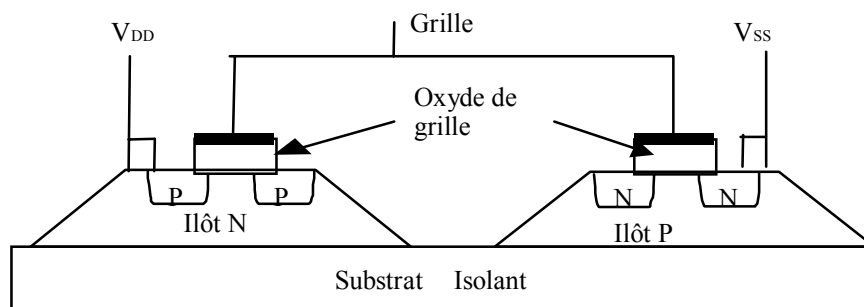
La technologie CMOS est la technologie dominante pour la plupart des applications d'aujourd'hui, qu'elles soient commerciales pour le grand public ou spatiales. Cette technologie permet la conception de circuits à faible coût, rapides et consommant peu de courant. Plusieurs méthodes et processus technologiques ont été développés pour réduire la sensibilité des technologies CMOS aux radiations. Les plus représentatives sont :

- CMOS sur substrat épitaxié : procédé technologique qui conduit à une forte réduction de la sensibilité au latch-up et à une légère amélioration de la résistance aux upsets [DIEH '83] (figure. 1.11).



**Figure 1.11. La technologie CMOS sur substrat épitaxié.**

- CMOS sur substrat isolant : structure appelée SOI (Silicium On Insulator), qui a pour effet de supprimer la structure parasite PNPN, la rendant totalement insensible au latch-up et réduisant considérablement la sensibilité aux upsets (figure. 1.12).



**Figure 1.12 Structure d'inverseur réalisée dans la technologie SOI**

Le coût élevé du SOI rend, pour l'instant, ces circuits peu répandus par rapport à la technologie CMOS classique [MUSS '96].

- D'autres matériaux peuvent être utilisés - par exemple, le GaAs - présentant une très grande tolérance à l'effet de la dose cumulée et une bonne insensibilité aux phénomènes de latch-up. Mais ces composants sur GaAs ne s'utilisent pas à cause de leur coût et leur consommation très élevées.

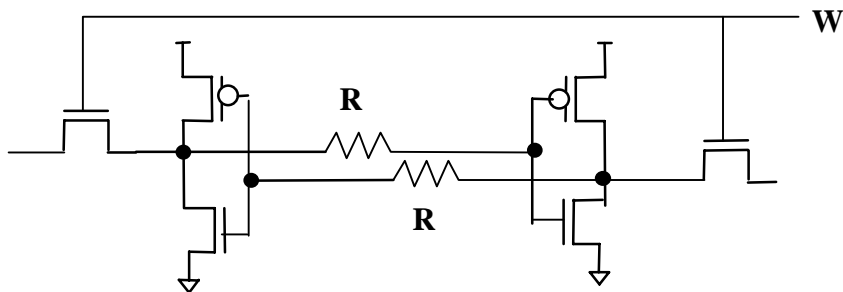
Pour revenir au cas de la technologie CMOS, on peut utiliser deux moyens pour réduire la sensibilité aux SEUs :

- **Augmentation des dimensions des transistors** : elle consiste à augmenter la longueur du canal de transistor. Mais, cette solution est en opposition avec la tendance actuelle qui va vers une diminution de la taille et de la consommation des composants.
- **Atténuation de l'impulsion du courant transitoire induite par une particule** : Généralement, il y a deux moyens pour atténuer l'impulsion de courant : soit on diminue la charge collectée par un nœud sensitif ( $Q_{coll}$ ), soit on maximise la charge critique nécessaire pour produire un upset ( $Q_{crit}$ ).

Par la suite, on va détailler quelques méthodes qui exploitent ces deux façons d'atténuer l'impulsion de courant.

- L'utilisation de la technologie ayant des substrats structurés sert à minimiser la charge totale collectée par le nœud sensitif. Cependant, ces approches technologiques ne sont plus suffisantes pour la construction d'un circuit électronique durci face aux upsets, et notamment pour des technologies où les tailles de transistors sont plus petites que  $4\mu\text{m}$ . Le choix des éléments est également très important. En effet, dans un circuit vulnérable aux SEUs, il faudrait éviter les nœuds flottants ou les nœuds en haute impédance comme les DRAM et les composants de type « charge-coupled ». Les techniques de durcissement par la conception ne doivent pas avoir d'impact sur les performances du circuit (en termes de vitesse et de consommation) et ne doivent pas rendre la conception du circuit très complexe non plus.

- Afin de maximiser la charge critique ( $Q_{crit}$ ) d'un nœud sensible, une solution possible consiste à augmenter la capacité du nœud sensible. Cependant, en rajoutant de telles capacités, les performances du circuit diminuent et la surface globale augmente. Ceci est en complète contradiction avec les tendances souhaitées actuellement pour le VLSI [DIEH '83].
- Une autre moyen permettant de augmenter  $Q_{crit}$  se base sur l'ajout de résistances de contre-réaction [SEXT '89]. Cette solution est très efficace pour les SRAM, et consiste à intégrer des résistances en polysilicium dans la boucle de régénération de réponse du bistable (figure.1.13). Ainsi, l'impulsion de courant résultant du passage d'une particule sera atténuée.



**Fig.1.13 Bistable avec résistance de contre-réaction.**

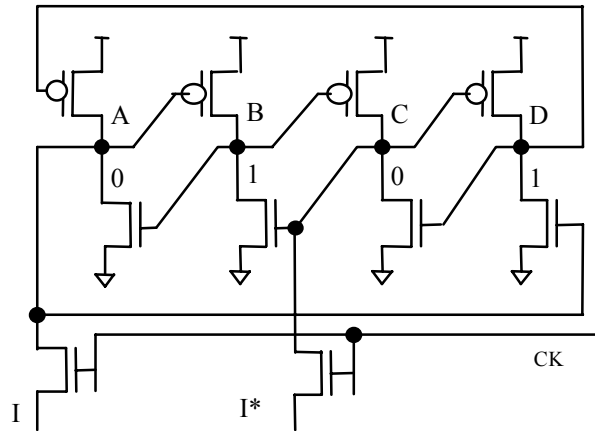
Le processus technologique de durcissement par l'ajout de résistances est particulièrement délicat : il demande un contrôle très fin des résistances de contre-réaction par les paramètres technologiques (tels que le dopage ou la corrosion du polysilicium de haute résistivité).

### 1.9.3. Durcissement par la conception au niveau de la cellule

Les points mémoire, qui se trouvent dans la partie critique d'un circuit, tels que les latches distribués dans le circuit logique, ne seront pas durcis par la procédure de rajout de résistances de contre-réaction, car ces résistances ont une influence importante sur la vitesse du circuit. Afin de durcir ces latches, il est préférable d'utiliser d'autres moyens liés à la conception comme, par exemple, l'ajout de transistors. La figure 1.14 présente une telle réalisation. Il s'agit de la cellule DICE (Dual Interlocked CE11), conçue pour être



insensible aux SEUs [CALI '96].



**Figure 1.14** Architecture de la cellule DICE.

Pour cela, la donnée est stockée dans quatre nœuds au lieu de deux comme se serait le cas d'une cellule conventionnelle. Ainsi, on utilise quatre inverseurs au lieu de deux (cas d'une cellule conventionnelle) (figure 1.9). Chacun des quatre inverseurs a les grilles des deux transistors NMOS et PMOS découplées, l'une étant contrôlée par le nœud se trouvant à gauche de l'inverseur, et l'autre par le nœud se trouvant à droite. Cette configuration permet que, si une impulsion transitoire change la valeur d'un des quatre nœuds, il reste toujours un couple de nœuds consécutifs non affectés par l'impulsion et ayant les valeurs '0' et '1'. Par exemple, si dans la figure 1.14 le nœud A est affecté par l'upset, le couple de nœuds assurant le non basculement de la cellule de mémoire est le couple B - C. La valeur '1' du nœud B déconnecte C de  $V_{DD}$ , tandis que la valeur '0' du nœud B déconnecte B de la masse Gnd. Ainsi, la valeur du nœud C ne peut pas être changée de '0' à '1', et la valeur de B ne peut pas être changée de '1' à '0'. Les deux nœuds gardent leurs valeurs correctes, empêchant à tout moment la cellule de basculer. Les courants circulant par les transistors restaurent rapidement les valeurs des nœuds A et D.

Des tests ont montré que la cellule était immune aux upsets jusqu'à un LET de 60 Mev.cm<sup>2</sup>/mg. Au delà de ce LET, des upsets sont possibles pour cause d'impulsions créées par la même particule sur deux nœuds différents. Des précautions topologiques sont nécessaires pour une protection parfaite. L'augmentation de la surface pour la cellule DICE est de 91% par rapport à une cellule conventionnelle.

En fonction de l'application à laquelle le circuit/système intégré est destiné, le choix sera fait pour une protection totale ou partielle. Par exemple, dans les applications critiques et de sécurité, les circuits seront durcis, d'une part, par un processus technologique, et d'autre part par la conception au niveau circuit et système. Pour les applications commerciales, on choisira les méthodes de durcissement les moins chères en terme de coût en surface, de processus technologique et d'impact sur les performances du système.

#### **1.9.4. La prévention au niveau du système**

Les deux moyens de prévention précédemment cités (technologique et architectural) ne permettent toujours pas une protection totale face aux erreurs produites par les upsets. D'ailleurs, les applications commerciales ne peuvent pas se permettre un coût très élevé de conception et de processus technologique. D'autres précautions doivent donc être prises en compte au niveau du système, comme la détection et la correction d'erreurs (EDAC), la redondance matérielle et/ou logicielle, le chien de garde,...

Tous ces moyens sont coûteux. Il est donc indispensable de pouvoir évaluer la sensibilité d'un circuit au système et de décider du niveau de protection nécessaire pour ne pas augmenter le coût du système de façon injustifiée.

Pour ce faire, des simulations de l'impact d'une particule énergétique avec des calculs de prédiction sont nécessaires. Une fois qu'un circuit ou un composant est caractérisé, il devient possible soit de l'éliminer, soit de prévoir les actions adéquates, pour diminuer les risques lorsque ces systèmes fonctionnent dans l'espace ou dans des environnements à risques.

#### **1.10. Les effets de la miniaturisation**

Dans le passé, le phénomène de SEUs représentait un problème pour les circuits intégrés utilisés dans les environnements sévères, tels que le spatial ou les réacteurs nucléaires. De plus, seuls les circuits de mémorisation étaient supposés sensibles.

Malheureusement, comme nous l'avons déjà dit, la miniaturisation des circuits intégrés a des effets néfastes sur leur sensibilité aux diverses sources de perturbation. Les phénomènes de SEU ne font pas exception à la règle. Bien au contraire, il devient une source majeure de perte de fiabilité. Les causes principales de cette dégradation sont la réduction du niveau de l'alimentation  $V_{DD}$  et la réduction des dimensions des dispositifs, qui réduisent ainsi les capacités des nœuds d'un circuit. Ainsi, la charge stockée sur un nœud  $Q=C*V_{DD}$  est doublement affectée, nécessitant une charge collectée sensiblement plus faible, susceptible de produire une erreur logique.

Une augmentation de la sensibilité aux upsets est observée tout premièrement, dans les circuits de mémorisation.

Cependant, les parties logiques considérées pratiquement insensibles par le passé sont aussi maintenant affectées. Outre la réduction de la charge stockée dans les nœuds d'un circuit logique, il existe deux autres facteurs qui affectent sa sensibilité :

- Tout d'abord, la miniaturisation est accompagnée d'une augmentation importante de la vitesse de fonctionnement des composants. Ainsi, le temps de transition logique d'une porte  $T_1$ , devient inférieur à la durée de l'impulsion transitoire  $D_{TR}$  induite par l'impact des particules (de 50 à quelques centaines de picosecondes). Par conséquent, ces impulsions ne sont plus filtrées par les portes logiques. Auparavant, à  $T_1 > 2D_{TR}$  l'impulsion était immédiatement stoppée, à  $D_{TR} < T_1 < 2D_{TR}$  l'impulsion se propageait avec atténuation, et à  $D_{TR} > T_1$  l'impulsion se propageait sans aucune atténuation [BAZ'97]. Ainsi, dans les technologies actuelles, où l'on a affaire au troisième cas, les impulsions se propagent dans les réseaux de portes logiques jusqu'aux sorties du circuit où elles peuvent être capturées par les latches.
- Une impulsion est capturée par les latches, si son instant d'arrivée à la sortie du circuit coïncide avec le front actif de l'horloge. La probabilité de capturer une impulsion d'une durée donnée augmente alors, proportionnellement à la fréquence d'horloge. Par conséquent, la sensibilité des parties logiques augmente au fur et à mesure que les technologies s'approchent des limites possibles en termes de dimensions des composants, de tension d'alimentation et de vitesse.

Deux types de particules sont responsables des phénomènes de SEUs dans l'environnement terrestre, les **particules alpha** et les **neutrons** (car la plupart des protons provenant du vent solaire sont retenus par la ceinture de Van Allen).

- Les **particules alpha** sont produites par la désintégration des isotopes radioactifs se trouvant dans les matériaux de fabrication des circuits [MAY'78]. Comme la décontamination isotopique préalable des matériaux coûte excessivement cher, il semble peu probable de résoudre ce problème en éliminant cette cause.
- Les **neutrons atmosphériques** constituent une deuxième cause. Leur origine est le rayonnement cosmique et l'activité solaire. Le flux de neutrons atmosphériques a une bande d'énergie très large, mais ce sont les neutrons thermiques qui sont les plus réactifs (1-10 MeV). On estime qu'en moyenne, il y a 20 neutrons par heure qui bombardent 1 cm<sup>2</sup> de la surface de la terre. Le flux varie avec l'altitude (le maximum se situant aux alentours de 18km d'altitude [TAB'92], [NORM'96]) et avec la latitude (le flux de neutrons est plus faible à l'Equateur et plus important autour des pôles). Notons également que les neutrons, n'ayant pas de charge électrique, ne peuvent pas induire des paires d'électrons – trous dans le silicium, et donc, ils ne peuvent pas produire des SEUs de manière directe. Le phénomène d'upset se produit de façon indirecte par les particules secondaires (telles les particules alpha et les nucleus de lithium) avec les atomes de Bore (B<sup>10</sup>) [NORM'96]. Nous avons donc, au bout du compte, un SEU produit par une particule alpha.

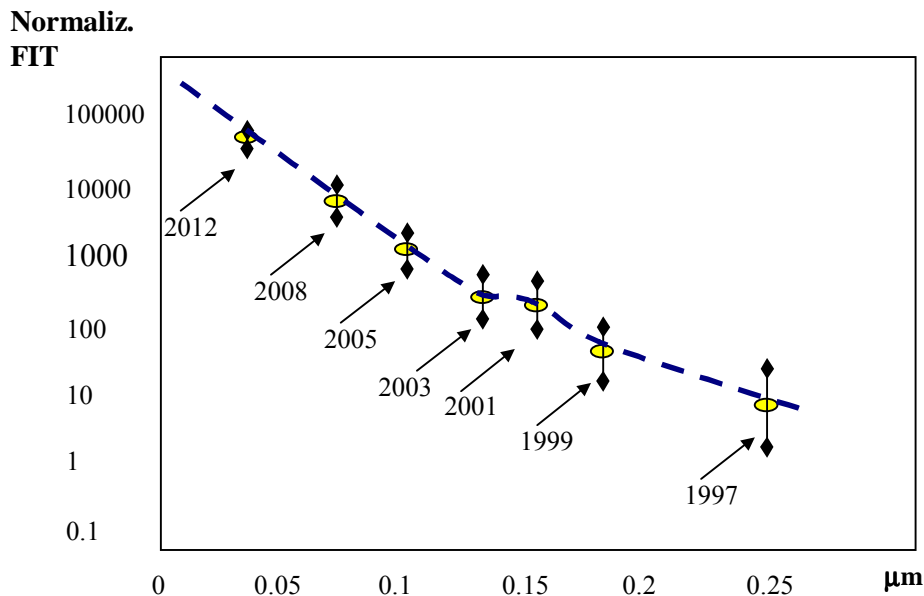
Cette analyse nous permet d'anticiper les tendances, mais ce n'est que très récemment que des résultats expérimentaux viennent nous fournir des données quantitatives, très inquiétantes, par ailleurs.

Ainsi, selon les expérimentations menées par la compagnie AMD, les circuits logiques sont dorés et déjà aussi sensibles que les points mémoire [COH'99].

Selon des données venant d'IBM, dans une mémoire SRAM de 1GB en 1996 (technologie de 0,25um), le nombre d'upsets au niveau de la mer est de 0,2 upsets/cm<sup>2</sup>/jour [DEL'99]. On voit déjà qu'il existe un problème en environnement terrestre et qui s'accroît avec

l'altitude. Sur les vols inter- et transcontinentaux, les choses deviennent encore plus sérieuses avec l'augmentation du flux de neutrons en altitude.

Une autre étude venant d'une collaboration entre AMD, Intel et Compaq [COH'99] traite les tendances des prochaines générations technologiques. L'étude démontre une augmentation du nombre d'upsets de façon exponentielle avec la réduction de  $V_{DD}$ . Par ailleurs, sa réduction a une influence relativement bien plus importante que la miniaturisation géométrique, car elle contribue à 75% sur l'augmentation de la sensibilité aux SEUs. La figure 1.15. montre les résultats obtenus dans cette étude.



**Figure 1.15** Prédiction des erreurs softs (étude réalisée par AMD, Intel, Compaq)

L'axe vertical étant logarithmique, on observe une augmentation exponentielle du nombre de défaillances dans le temps (FIT). Les FITs sont donnés en valeurs normalisées, et non absolues (les constructeurs ne souhaitent pas publier de données sensibles). Les lignes verticales montrent l'influence de  $V_{DD}$  sur le FIT, à technologie égale.

Considérant comme point de départ les données d'IBM pour la mémoire à un GByte, on peut extrapoler une sensibilité de une SEU toutes les deux heures, pour la même mémoire dans le courant de l'année prochaine. On se rend compte que les prochaines

générations technologiques seraient inutilisables si les constructeurs ne prenaient pas de mesures préventives pour résoudre ce problème.

## **Conclusion**

Dans ce chapitre, nous avons mis en évidence et discuté d'un grand nombre de phénomènes liés à l'évolution technologique vers les composants nanométriques qui auront des conséquences néfastes sur la fiabilité des circuits intégrés. Nombreux facteurs contribuent à la dégradation de la fiabilité. Tout d'abord, la miniaturisation met en jeu plusieurs phénomènes ignorés dans le passé, mais qui devront être pris en compte à l'avenir, à cause de leur influence grandissante, voire dominante, sur le fonctionnement des circuits. Pour résoudre cette complexité induite par la miniaturisation, il faudra des outils de conception plus performants ; maints d'entre eux ne sont pas disponibles actuellement. Par ailleurs, maîtriser ces nouveaux phénomènes demande un allongement du temps de conception incompatible avec la tendance du marché qui par lui-même exige la réduction du « time-to-market », le temps de lancement d'un nouveau produit sur le marché. De plus, on exige aussi l'allongement de la durée de vie des nouveaux produits. En conséquence, le risque de voir sur le marché des produits défectueux pour cause d'erreurs de conception augmente. L'exemple du Pentium III, conçu pour fonctionner à 1,13 GHz et qui a dû être retiré du marché récemment [SCI'00], montre que ce risque est réel. Ce ne sera probablement pas une exception dans l'avenir. Il semblerait que, pour certaines configurations d'utilisation, le Pentium III ne peut plus fonctionner au-delà de 800 MHz. Il s'agirait donc d'un problème de timing mal géré lors de sa phase de conception.

Mais, en supposant que les concepteurs aient pu parfaitement régler tous les problèmes de conception d'un produit totalement conforme aux spécifications, le problème de fiabilité n'est pas pour autant résolu.

Comme nous l'avons vu, l'impact des défauts de fabrication sur le fonctionnement des circuits devient de plus en plus complexe, et on se trouve devant l'incapacité de produire des programmes de test assurant une couverture satisfaisante pour ces types de défauts.

C'est le cas, par exemple des fautes de timing. Du fait de l'augmentation des vitesses de fonctionnement, même de faibles augmentations de retard se manifesteront comme des

fautes de timing. Ainsi, la proportion de fautes de timing augmente, augmentant à son tour la probabilité de déclarer bon un circuit défaillant, étant donné que les fautes de timing sont beaucoup plus difficiles à détecter que les fautes de collage logique.

C'est aussi le cas des défauts qui ne produisent pas une erreur franche à la sortie de l'élément défaillant, mais produisent plutôt un niveau logique dégradé. Ces défauts sont souvent difficiles à détecter, et ils se manifestent le plus souvent comme des fautes de timing.

Finalement, supposons qu'un fabricant ait payé le prix nécessaire à la génération d'un programme de test capable d'assurer une couverture de défauts très élevée, et pour ce faire aurait payé aussi le prix d'un temps de test suffisamment long pour mettre au point ce programme. Il livrerait au client des circuits sans aucune défaillance de fabrication, ni erreur de conception, sans avoir pour autant assuré un niveau de fiabilité suffisant. Comme nous l'avons vu, la miniaturisation réduit les marges de bruit de façon dramatique, les circuits deviennent très sensibles aux perturbations, notamment aux impacts des particules alpha et des particules cosmiques, tels que les neutrons atmosphériques. Devant des taux d'erreurs que nous pouvons déjà qualifier de désastreux, il ne reste que deux solutions : soit arrêter le progrès technologique en arrêtant la miniaturisation, ce qui évidemment est utopique, soit trouver des techniques de conception permettant de tolérer ces erreurs. C'est à ce dernier point que nous avons voué la plus grande part de ce travail.

L'utilisation des codes correcteurs d'erreurs tels que le code d'Hamming offre une solution efficace pour les mémoires. Ainsi, le problème principal qu'on posera par la suite concernera les techniques de conception de circuits logiques tolérants les défauts. Les techniques de tolérance aux fautes traditionnelles pour les circuits logiques (duplication, triplication) sont très coûteuses, et ne pourront pas être acceptées pour les applications à faible valeur ajoutée, tels que les produits grand public. On doit donc chercher des solutions à moindre coût. Mais le coût d'une solution peut dépendre fortement des fautes dont on cherche à se protéger. Ainsi, dans un souci de faible coût, on s'intéresse uniquement aux défauts qui ont pour effet une dégradation de la fiabilité des circuits intégrés, le but n'étant pas d'assurer un niveau de fiabilité très élevée, comme c'est le cas d'une application critique en sécurité, mais plutôt de maintenir la fiabilité à son niveau actuel.

Dans ce but, les fautes à prendre en compte dans un premier temps sont les fautes transitoires, telles que les SEUs, car elles auront un impact très néfaste sur les prochaines générations de circuits intégrés. Ensuite, il faut nous occuper des défauts de fabrication difficiles à détecter, c'est-à-dire les fautes de timing tout d'abord, et éventuellement les fautes produisant des niveaux dégradés. Mais ces dernières se manifesteront la plupart du temps comme des fautes de timing.

Celles-ci seront donc le plus souvent couvertes par des techniques couvrant les fautes de timing proprement dites. En constatant que les fautes transitoires (tels que les SEUs et les fautes de timing) n'affectent que les aspects temporels du signal, on va regrouper ces fautes sous la dénomination de *fautes ou défauts temporels*, et on s'intéressera principalement à elles.



## Chapitre II

### Modélisation et simulation des impulsions transitoires

#### **2.1. Modèle logique de l'impulsion transitoire résultant de l'impact des particules alpha dans les circuits CMOS**

Suite à l'analyse du chapitre I, nous nous sommes fixés pour objectif d'étudier des techniques de conception de circuits numériques tolérants les fautes temporelles, telles que les fautes transitoires et les fautes de timing (ou fautes de retard), et ceci à faible coût d'implémentation. Pour pouvoir évaluer l'efficacité de ces techniques et être capables de choisir la technique à moindre coût satisfaisant les objectifs fixés en termes de couverture de défauts, nous avons besoin d'outils de simulation de fautes transitoires et de fautes de retard. Etant donné que nous disposons d'outils commerciaux pour la simulation de fautes de retard (voir, par exemple, FastScan et FlexTest de Mentor Graphics), nous nous sommes intéressés ici à la simulation de fautes transitoires.

L'impact d'une particule chargée dans les circuits a été extensivement étudié dans le passé récent [CHA 93]. Si une particule énergétique percute une zone sensible d'un transistor (par exemple la région de drain d'un transistor NMOS bloqué, conformément à la figure 1.9.), elle crée une colonne très dense de paires d'électrons-trous dans la jonction p-n. Une partie des charges générées par l'impact se recombine, une autre est collectée au nœud par des phénomènes de dérive-diffusion dus à la différence de potentiel dans la région désertée et au champ électrique qui en résulte.

Le mouvement des paires électrons-trous produit une impulsion de courant transitoire très brève au nœud affecté. L'amplitude de cette impulsion dépend des propriétés de la particule (énergie, type) et des propriétés physiques du circuit. L'impulsion de courant est souvent modélisée par une double exponentielle [CHA 93], dont l'équation (1) et la forme d'onde sont données dans la figure 2.1. :

$$I(t) = I_0(e^{-t/\tau_\alpha} - e^{-t/\tau_\beta}), \quad (1)$$

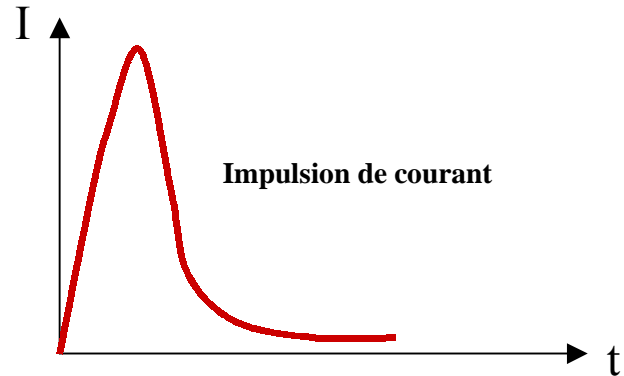
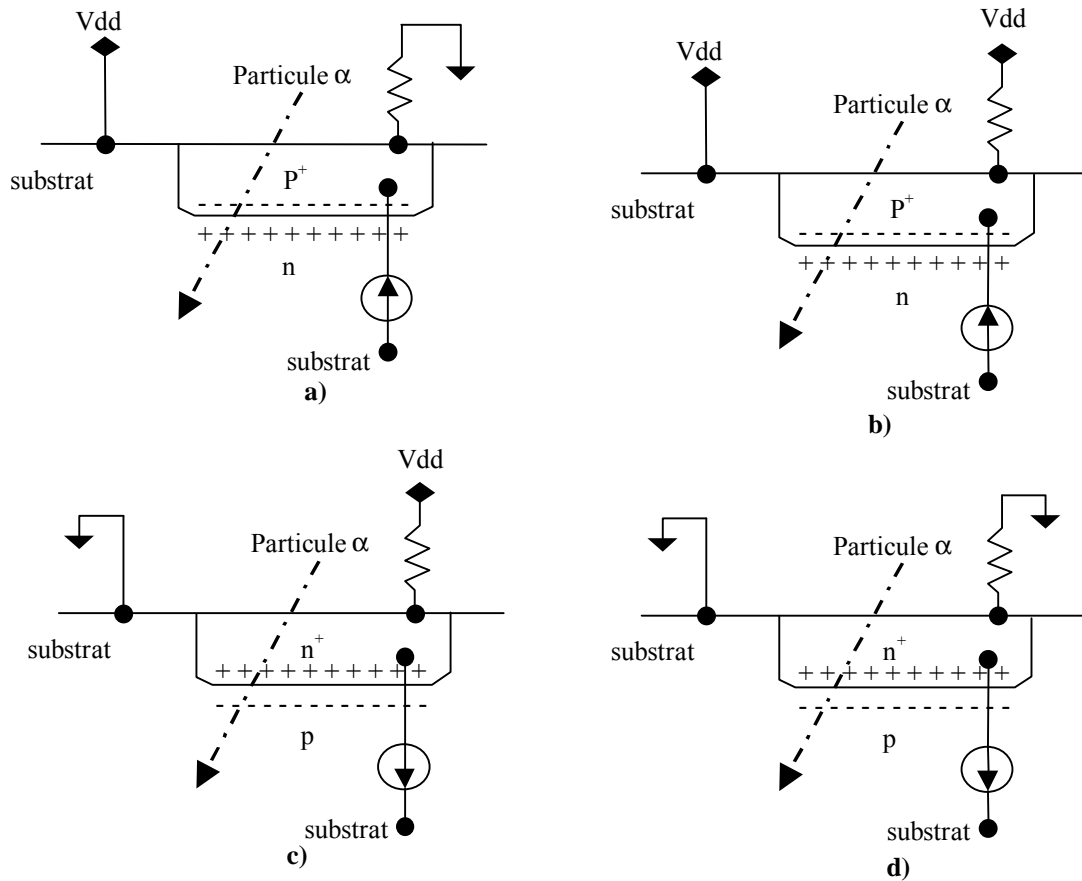


Figure 2.1. Forme d'onde d'une impulsion de courant transitoire.

où  $\tau_\alpha$  est le temps de collection de charge de la jonction  $p^+n$  et  $\tau_\beta$  est le temps d'établissement de la trace initiale créée par la particule. Ces constantes de temps dépendent de la technologie utilisée. Les valeurs typiques sont  $\tau_\alpha = 1.64 \cdot 10^{-10}$  secondes et  $\tau_\beta = 5 \cdot 10^{-10}$  secondes. Le coefficient  $I_0$  peut être soit positif, soit négatif, selon que la particule touche la région de drain d'un transistor NMOS ou d'un transistor PMOS.

La figure 2.2. montre quatre possibilités d'évolution pour le cas d'un inverseur CMOS. Les résistances représentent des transistors en conduction, les rectangles les régions de drain des transistors et les flèches correspondent à la direction du courant. Pour les cas **a)** et **b)** le potentiel au nœud  $p^+$  augmente. Pour les cas **c)** et **d)** le potentiel au nœud  $n^+$  diminue. Les cas **b)** et **d)**, n'influencent pas l'état logique du circuit parce que le nœud est déjà au potentiel envisagé. Les cas **a)** et **c)** affectent temporairement la valeur électrique de la tension du nœud à cause de l'impulsion transitoire.



**Figure 2.2. Quatre possibilités d'évolution pour l'impact d'une particule**

Le transistor affecté par l'impact d'une particule chargée peut être soit dans un état stable, soit en commutation. Les particules chargées affectant le circuit pendant la phase de commutation, avant que le circuit se stabilise, ne produisent pas d'erreurs. Le deuxième cas, correspondant au circuit à l'état stable est le seul intéressant et on analyse le cas de l'impact d'une particule dans un état stable du circuit ('0' ou '1' logique).

## 2.2. Modèle temporel d'impulsion transitoire

L'impact d'une particule chargée sur un nœud sensible provoque une impulsion de tension d'une largeur qui dépend de plusieurs paramètres : la technologie de fabrication du transistor, la charge électrique totale collectée et la valeur de la capacitance du nœud sensible. Si on considère le transistor PMOS comme une résistance linéaire de valeur  $R$ , on obtient une solution analytique pour l'expression de la tension de sortie en fonction de temps :

$$V_0(t) = R \cdot I_0 \left[ \frac{e^{-t/\tau_\alpha} - e^{-t/RC}}{1 - RC/\tau_\alpha} - \frac{e^{-t/\tau_\beta} - e^{-t/RC}}{1 - RC/\tau_\beta} \right] + V_{dd} \quad 2)$$

L'équation 2) exprime la dépendance de la tension  $V_0$  à l'aide de deux paramètres  $RI_0$  et  $RC$ , ce qui simplifie l'expression de cette tension de sortie en éliminant la dépendance de la charge collectée. Une représentation encore plus simplifiée est le modèle linéaire représenté dans la figure 2.3, qui sera comparé avec des simulations SPICE. Les simulations SPICE étaient faites sur un inverseur ayant plusieurs charges d'injection et plusieurs capacités de sorties. Le seuil logique a été fixé à 2.5V pour extraire les formes d'onde logiques.

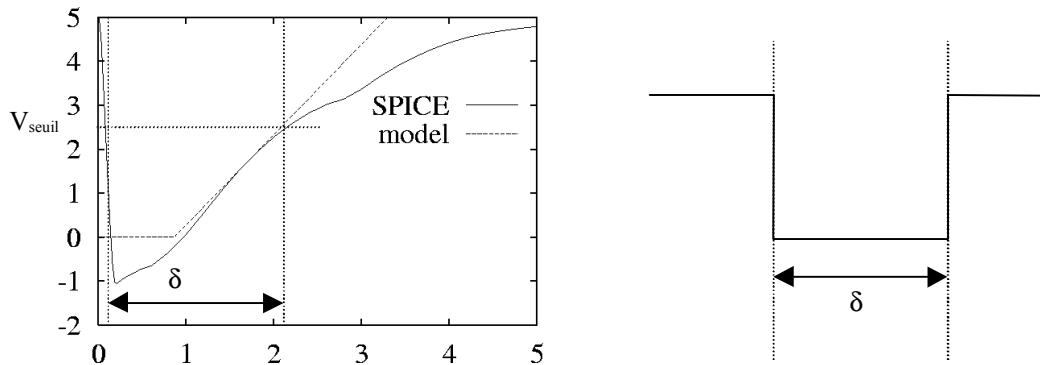
La largeur de l'impulsion peut être exprimée par une équation linéaire :

$$PW_1 = A \frac{L}{W} I_0 + B \frac{L}{W} C_{out} + Const \quad 3)$$

où  $PW_1$  représente la largeur de l'impulsion au point d'injection et les constantes  $A$ ,  $B$  et  $Const$  peuvent être calculées en utilisant des simulations SPICE.

La durée de l'impulsion transitoire est donc une fonction de la charge injectée et de la capacité au nœud d'injection. On remarque deux régions distinctes : la première, pour les petites capacités du nœud ; dans ce cas, la durée de l'impulsion transitoire augmente avec la

charge injectée. La deuxième correspond aux grandes capacités du nœud d'injection, et la charge injectée n'est pas suffisante pour inverser la valeur logique du nœud sensible. Alors, la durée de l'impulsion est plus courte (figure 2.3.).



**Figure 2.3** Modèle de la forme d'onde de l'impulsion transitoire au nœud affecté et le modèle logique (impulsion rectangulaire de largeur  $\delta$ )

Pour pouvoir évaluer la sensibilité face aux SEUs, dans le cas d'un circuit complexe, on sera amené à faire un très grand nombre de simulations. L'utilisation du modèle exponentiel pour l'impulsion nécessitera des simulations SPICE très coûteuses en temps CPU. Elle limitera ainsi, de façon importante le nombre de simulations et la qualité de l'évaluation. Nous avons, en conséquence besoin d'un modèle simplifié qui permet une simulation rapide, mais qui ne compromet pas la qualité du résultat de la simulation.

Pour une porte logique qui reçoit en entrée l'impulsion électrique, la forme exacte de cette impulsion n'est pas d'une grande importance. On pourrait en fait, obtenir le même résultat à la sortie de la porte logique, en utilisant une impulsion électrique de forme rectangulaire et d'amplitude égale à  $V_{DD}$  dont on aurait choisi la durée de façon judicieuse. Une bonne approximation est de prendre une durée égale au temps  $\delta$  pendant lequel l'amplitude de l'impulsion originelle dépasse le niveau de seuil de la porte logique (voir figure 2.4). Les caractéristiques des impulsions créés dans un circuit par les impacts de particules ne sont pas d'un seul genre. La forme, l'amplitude et la durée des impulsions dépendent du type et de l'énergie de la particule incidente, ainsi que de l'emplacement et de l'orientation de l'impact (angle d'incidence de la particule). Il est donc inutile de chercher à simuler des impulsions ayant une forme, une durée et une amplitude très précises.

L'utilisation du modèle de l'impulsion rectangulaire dont la durée est choisie judicieusement est suffisante.

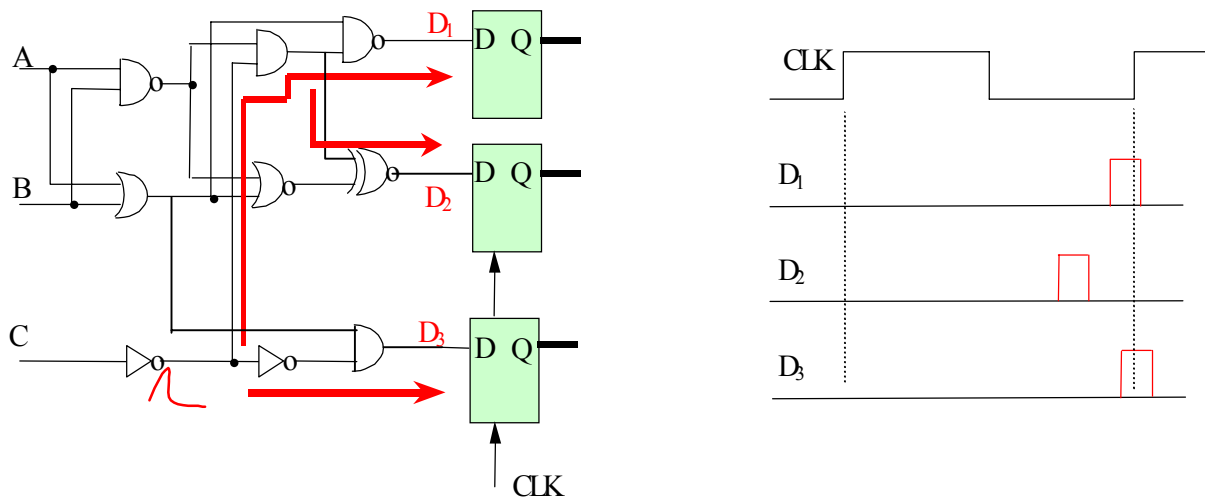
Par la suite cette dernière impulsion sera nommée '**impulsion équivalente**'. Par conséquent, on pourra effectuer des simulations en utilisant uniquement des impulsions équivalentes. Puisque la durée de l'impulsion équivalente varie, sa durée pourra être choisie prenant en compte soit une durée moyenne de l'impulsion transitoire, soit une durée pessimiste, ou bien dans un intervalle de durées considérées comme représentatives. Ce choix appartient au concepteur de circuits intégrés tolérant les fautes transitoires, qui décide ainsi de la durée d'impulsion transitoire qu'il veut tolérer. Lors de la simulation logique on utilise un signal d'injection de fautes, qui est en fait un signal logique rectangulaire d'une largeur  $\delta$ .

Pour ce qui nous concerne, nous savons de la littérature que les impulsions transitoires créées par les particules ionisantes ont des durées de l'ordre de 50 à quelques centaines de picosecondes. Pour valider nos analyses, nous utiliserons donc dans nos évaluations, des impulsions de cette ordre de durée car pour les technologies que nous avons utilisées nous n'avons pas pu avoir accès aux caractéristiques technologiques.

En conclusion, on peut modéliser les impulsions transitoires créées par les particules ionisantes par des impulsions carrées, ayant la durée comme unique paramètre. Néanmoins, dans les circuits logiques synchrones que nous étudions dans le reste du manuscrit, le résultat final de l'impulsion dépend d'un deuxième paramètre temporel. Il s'agit de l'instant de création de l'impulsion dans un cycle d'horloge. Cet instant nous permet de déterminer à quel moment, l'impulsion propagée dans le réseau des portes logiques parviendra à chaque sortie du circuit, permettant de savoir si l'impulsion sera capturée par le latch de sortie. Nous obtenons donc un modèle temporel d'impulsion déterminé par deux paramètres, la durée d'impulsion et l'instant de création de cette impulsion, en fait l'instant de l'impact de la particule sur le circuit.

### 2.3. Analyse de la propagation de l'impulsion transitoire dans un circuit logique

On envisage par la suite à déterminer les conséquences d'une impulsion transitoire sur les résultats fournis par un circuit synchrone. Un cas typique de circuit logique affecté par une impulsion transitoire est présenté dans la figure 2.4.



**Figure 2.4. Scénario de propagation d'une impulsion transitoire dans un circuit combinatoire et les formes d'ondes des sorties affectées.**

Les trois chemins de propagation sont activés par un vecteur d'entrée adéquat. Ainsi, l'impulsion transitoire arrive aux sorties  $D_1$ ,  $D_2$ ,  $D_3$  en fonction de la longueur du chemin de propagation correspondant. Si elle arrive à une sortie au moment du front actif de l'horloge du système, elle sera capturée dans le latch comme une valeur erronée (le cas des sorties  $D_1$  et  $D_3$ ).

Dans d'autres cas, l'impulsion transitoire peut être atténuée avant son arrivée aux sorties. Si sa durée est plus grande que les temps de transition des portes logiques, il peut se propager aux sorties sans atténuation [BAZ'97]. Comme les portes logiques deviennent très rapides dans les nouvelles technologies, avec des temps de propagation très courts, les impulsions transitoires ne seront plus atténuées, même pour des particules d'énergie réduite [NIC'99]. Néanmoins, la durée de l'impulsion finale pourrait être réduite ou augmentée selon les portes traversées.

Il existe deux modes de modification de la durée d'une impulsion :

- Premièrement, il s'agit de l'asymétrie du retard des portes logiques pour les transitions montantes et descendantes. Si le retard de propagation de la transition montante est supérieur au retard de propagation de la transition descendante, la porte en cause va élargir une impulsion positive et va raccourcir une impulsion négative. L'inverse se produit si la relation des retards des impulsions est inversée. Cette façon de modifier les durées des impulsions aura un impact faible, car l'impulsion s'inversant après le passage de chaque porte logique, subira généralement des réductions et des augmentations alternées de sa durée, qui se compenseront mutuellement plus ou moins.
- Le deuxième mode de modification de la durée des impulsions est dû aux chemins reconvergeants. L'existence des chemins reconvergeants ayant des retards différents peut modifier de façon importante ce paramètre en superposant plusieurs impulsions sur le nœud reconvergent. Cette augmentation de la durée d'impulsion rend le circuit plus sensible, car elle augmente la probabilité que l'impulsion est présente sur les sorties au moment de l'arrivée du front actif de l'horloge.

On voit, donc, que nous disposons d'un modèle temporel de l'impulsion transitoire, qui doit se propager dans le circuit, en prenant en compte les retards des transitions montantes et descendantes des portes du réseau logique. Pour réaliser cette tâche, on a besoin d'un simulateur temporel, utilisant des fichiers SDF (Standard Delay File) et le circuit lui-même.

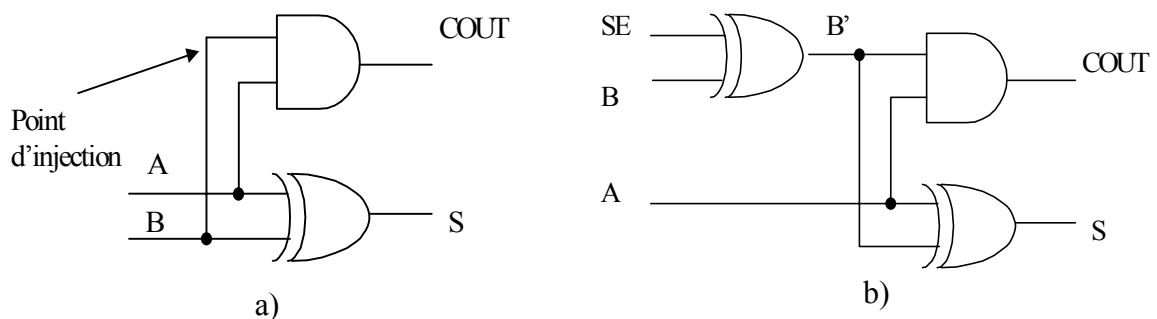
#### **2.4. L'environnement de la simulation des fautes transitoires**

Se basant sur ces principes, un environnement de simulation a été mis au point [ALEX'00] dans le cadre d'un projet de DEA que nous avons encadré. Cet environnement est utilisé de la manière suivante.



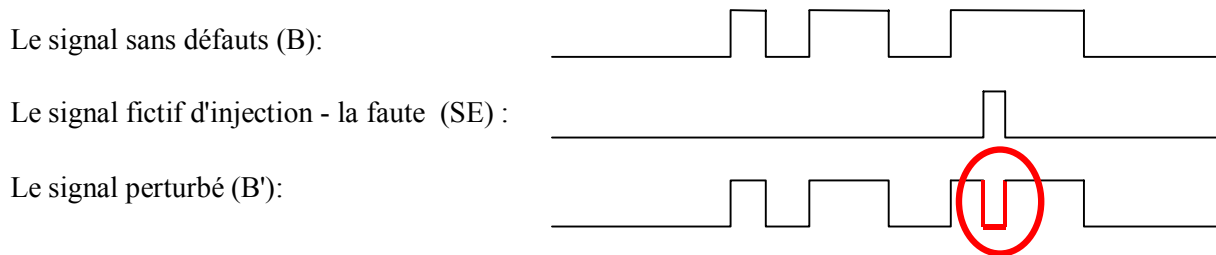
Tout d'abord, on réalise une représentation structurelle du circuit qui sera utilisé lors de l'injection des fautes transitoires. Partant de cette description dans un langage HDL comme Verilog ou VHDL, le flot de conception se poursuit par une opération de synthèse logique utilisant des outils comme 'Design Analyzer' (Synopsys) ou 'Ambit' (Cadence). La synthèse donne une description structurelle du circuit au niveau des portes logiques qui sera sauvegardée dans un fichier netlist Verilog, car l'algorithme de simulation a été greffé sur le simulateur VerilogXL de Cadence. En ce qui concerne les points d'injection, il est nécessaire d'analyser le circuit, de parcourir le fichiers netlist afin de choisir les nœuds d'injection. Chaque instance contenue dans la description d'un module Verilog peut-être soi-même un module ou un élément de bibliothèque (une porte logique élémentaire). Une procédure aléatoire automatique de recherche des nœuds d'injection a été mise en place pour assurer une répartition uniforme des points d'injection sur toute la surface du circuit sensible aux fautes transitoires. Une deuxième possibilité existe également, elle consiste en l'analyse des modules et de la représentation graphique du circuit. Ainsi, le concepteur peut choisir manuellement les points d'injection.

Pour injecter une faute transitoire sur un noeud, on peut utiliser deux approches. Soit qu'on modifie la structure physique du circuit pour simuler l'effet de la faute, soit qu'on modifie le comportement du circuit pendant la simulation selon le modèle de faute. Si on choisit de réaliser l'injection de fautes transitoires en modifiant le circuit (première méthode), on insère une porte logique XOR au point d'injection, commandé par un signal d'erreur (SE), comme le montre la figure 2.5. Une telle modification nécessite de reconstruire plusieurs niveaux hiérarchiques de la description du circuit, impliquant un temps additionnel de CPU important.



**Figure 2. 5. Modification manuelle du circuit : Avant a) et après l'injection de faute b).**

La deuxième solution est réalisée directement par l'outil de simulation à l'aide de la description Verilog. En effet on corrompt la valeur d'un signal arbitraire pour une période de temps égale à la durée du défaut transitoire à un instant de temps précis (voire figure 2.6). C'est la solution qu'on a choisi d'utiliser pendant l'algorithme de simulation car elle est moins coûteuse en temps CPU.



**Figure 2.6. Formes d'onde correspondantes au deuxième méthode d'injection de faute.**

Les deux méthodes sont équivalentes d'un point de vue fonctionnel.

Pour faire la simulation avec injection de fautes, on doit disposer d'un ensemble de vecteurs d'entrée. Ces vecteurs d'entrée seront déterminés par le concepteur (par exemple, des vecteurs d'entrée provenant d'un fichier « stimuli »). En absence de tels vecteurs, on peut utiliser l'ensemble complet des vecteurs du circuit. Mais dans ce cas, la simulation devient trop longue. Alors, on peut faire appel à une séquence de vecteurs de test aléatoire, assez longue mais suffisante la plupart du temps. Les vecteurs de test sont générés pseudo-aléatoirement en utilisant des circuits de type *LFSR* (*Linear Feedback Shift Register*). La dimension des LFSRs doit être en accord avec la largeur des bus d'entrée du circuit. Par exemple, pour un multiplicateur de type Wallace à deux entrées sur 16 bits nous avons utilisé un circuit LFSR de dimension 32 qui aura une période de  $2^{32}-1$  cycles d'horloge.

L'ensemble de vecteurs de test avec les points d'injection, les durées des impulsions et les instants d'injections déterminent l'espace de la simulation. Si on utilise 1000 vecteurs d'entrée à 200 points d'injection par vecteur et 5 instants d'injection par point d'injection, on aura un million de fautes différents à simuler. En pratique, on simulera plusieurs millions de fautes pour une meilleure confiance dans les résultats.

L'interprétation des résultats de l'injection des fautes est le pas suivant. Il consiste à décider si les valeurs présentes dans les registres de sortie sont correctes ou pas. Cette décision se fait par comparaison logique entre les valeurs de sortie du circuit sous test avec les valeurs correctes de référence.

En fait, pour chaque vecteur d'entrée, on réalise plusieurs cycles de simulation avec injection de fautes. Le premier cycle de simulation est sans injection de fautes et les valeurs des sorties sont considérées comme valeurs correctes de référence pour les cycles de simulation suivants.

L'évaluation des résultats d'une faute se fait en comparant les valeurs réelles des sorties avec les valeurs de référence. Si des circuits supplémentaires de détection de fautes sont prévus, comme des circuits de calcul de parité ou des contrôleurs DRC, on vérifie également si la détection a eu lieu afin d'évaluer l'efficacité du mécanisme de détection. On récupère les résultats d'évaluation grâce à un calcul probabiliste d'occurrence d'erreurs et de détection d'erreurs. Une fois que tous les fautes ont été simulés pour le vecteur de test courant, on passe à un autre vecteur de test.

L'implémentation effective de l'algorithme a été faite à l'aide d'un simulateur *event-driven*, (l'outil de simulation commercial est VerilogXL) sur lequel on a greffé l'algorithme proposé.

#### Algorithme de simulation avec injection de défauts dans le cas d'un circuit combinatoire

```

pour chaque vecteur de test
  évaluer les événements du circuit sans faute
  retenir les valeurs des sorties comme référence
  pour chaque faute
    injection de la faute (en utilisant un modèle logique- temporel)
    évaluer les événements provoqués par la faute
    récupérer les nouvelles sorties
    comparer les valeurs des sorties avec les valeurs de référence
    traitement de résultats (calcul des probabilités)
    rétablir le circuit avant injection (si nécessaire)
  end 'pour chaque'faute
end 'pour chaque'vecteur de test

```

L'environnement de test est implémenté par une description Verilog. Pour aider le concepteur, l'environnement de test peut être généré automatiquement, selon les caractéristiques du circuit sous test (nombre de bits aux entrées/sorties et méthode de

détection/tolérance aux fautes : un ou plusieurs bits de parité, contrôleur double-rail, etc.). Après chaque cycle de simulation, on dispose des valeurs des sorties du circuit et on peut commencer à analyser et à interpréter les résultats.

Le but du calcul est l'évaluation de la sensibilité du circuit aux fautes transitoires (par sensibilité d'un circuit on doit comprendre nombre de fautes créant des erreurs aux sorties rapporté au nombre de fautes injectés), ou l'estimation du rapport entre le nombre d'erreurs détectées et le nombre total d'erreurs affectant les sorties du circuit. Ceci permet le calcul de l'efficacité de la méthode de détection/tolérance aux fautes transitoires. Par la suite, nous prenons comme exemple l'évaluation de l'efficacité d'un mécanisme de détection d'erreurs.

En analysant le fonctionnement d'un circuit contenant un mécanisme de détection d'erreurs, on remarque quatre possibilités.

**1. Fonctionnement normal :** les sorties du circuit sont correctes et la sortie d'erreur n'indique pas d'erreur. La probabilité que le circuit fonctionne correctement est notée par  $p_{correct}$ .

**2. Fonctionnement incorrect avec détection d'erreurs :** les sorties sont incorrectes à cause d'une erreur qui est détectée (signalée par la sortie d'erreur). Ainsi, la dernière opération sera répétée. La probabilité qui correspond à ce fonctionnement sera  $p_{détection}$ .

**3. Fonctionnement incorrect sans détection d'erreurs :** les sorties sont incorrectes et le mécanisme de détection ne détecte pas la faute. Cette situation est mauvaise, car les résultats sont erronés et seront transmis aux étages suivants, compromettant le bon fonctionnement du système. La probabilité associée est  $p_{nondétection}$ .

**4. Fonctionnement correct avec indication d'erreur :** les circuits qui assurent la détection d'erreur ont probablement été affectés par la faute. Ceci n'est pas considéré comme une situation grave, mais le circuit répète quand même la dernière opération sans qu'il en soit besoin. (Souvent, ces méthodes de détection sont accompagnées d'un mécanisme de reprise consistant à répéter la dernière opération ou le dernier cycle d'opérations). La probabilité associée est  $p_{faux}$ .

L'efficacité de détection du circuit est le rapport entre les erreurs détectées aux sorties du circuit et le nombre total d'erreurs apparues aux sorties du circuit. Ainsi, l'efficacité est donnée par le rapport:

$$eff = \frac{P_{detection}}{P_{non-detection} + P_{detection}} \cdot 100(\%)$$

## **2.5. Les avantages de la solution proposée**

Dans ce chapitre nous avons défini un modèle temporel pour les fautes transitoires, qui ne compromettent pas la qualité des simulations. L'intérêt de ce modèle est une réduction importante du temps de la simulation par comparaison aux simulations de type SPICE nécessaires quand on utilise des impulsions décrites par une double exponentielle. Le modèle temporel a été ensuite utilisé pour développer un outil de simulation. L'avantage de cet outil est une diminution significative du temps de simulation.

Par exemple, pendant la validation de la méthode on a caractérisé l'efficacité d'une méthode de détection de fautes transitoires pour un circuit complexe (un multiplicateur de Wallace sur 16 bits) par l'utilisation de 100 points d'injections à 35 décalages chacun et 250 vecteurs de test, ce qui fait 875000 étapes de simulations pour 6 heures de calcul. Auparavant, une caractérisation équivalente nécessitait plus d'une semaine de simulation et quelques heures de travail pour réaliser les calculs statistiques. La qualité des simulations n'est pas compromise malgré la vitesse. Les résultats obtenus concordent avec les résultats classiques. Grâce à un environnement intégré de conception, la plupart des opérations nécessaires sont automatisées, réduisant ainsi la durée de l'intervention du concepteur.

Quant aux ressources nécessaires, notre méthode de simulation est comparable à la simulation différentielle des fautes et beaucoup moins exigeante que la simulation concurrentielle. Les besoins en mémoires pour la simulation de centaines de fautes sont semblables aux besoins en mémoire pour la simulation d'une seule faute, grâce au traitement séquentiel. Le coût d'implémentation est réduit parce qu'on a greffé l'algorithme sur le simulateur logique VerilogXL de Cadence. Cependant, il est possible d'utiliser d'autres supports qui peuvent simuler des circuits décrits en Verilog.



**Chapitre III****Circuits auto-contrôlables face aux défauts critiques dans les technologies nanométriques**

La méthode traditionnelle de tolérance aux fautes dans le cas des circuits logiques consiste dans la triplification du circuit et l'utilisation d'un circuit qui choisit les sorties des trois copies afin de sélectionner la valeur majoritaire (technique TMR – Triple Modular Redundancy). Mais, le surcoût matériel de cette méthode est très élevé, dépassant 200%. Il est donc inacceptable pour des applications à faible valeur rajoutée comme les produits grand public.

Dans le cas de fautes transitoires, il existe une méthode de tolérance aux fautes moins coûteuse. Il s'agit de la combinaison de la détection d'erreurs concurrentielle avec une procédure de reprise, afin de réaliser la correction d'erreurs.

Pour les fautes de timing, la correction d'erreurs est assurée par le même principe, mais pendant la phase de reprise, la fréquence d'horloge sera réduite afin d'éliminer les conséquences de la faute.

La technique de détection concurrentielle consiste dans la vérification des résultats fournis par un circuit pendant son fonctionnement normal. Elle peut être réalisée par une procédure classique, qui consiste en la duplication du système digital suivie de la comparaison des sorties provenant des deux copies. Cette technique a pour principal désavantage un surcoût matériel très important, plus de 100%, qui représente aussi un désavantage majeur pour les applications orientées vers le grand public.

Les circuits « auto-contrôlables » (self-checking dans la littérature anglaise) [AND 71], [AND 73], [ASH 77], [SMI 77], [SMI 78], [JHA 93] sont une alternative intéressante pour la détection concurrentielle du fait de leur coût beaucoup plus faible. Ils détectent les fautes permanents, mais aussi les fautes transitoires. Un nombre significatif d'implémentations de circuits auto-contrôlables a été proposé dans le passé. Une grande partie des ces circuits offrent une détection d'erreurs à faible coût (par exemple des multiplicateurs auto-contrôlables utilisant des codes arithmétiques [PET 58], [PET 72], [AVI 73], [ALZ 99]. Notre but est de pouvoir utiliser ces circuits pour le contrôle concurrentiel de fautes transitoires et de fautes de timing, mais aussi de fautes qui impliquent des valeurs logiques indéterminées sur le site d'occurrence de la faute.

Les fautes transitoires seront notre première préoccupation car ils représentent la cause la plus importante de dégradation de la fiabilité dans les technologies submicroniques avancées. Les fautes de timing étant la deuxième cause, ils devront aussi être couverts. Une troisième cause seront les défauts produisant des valeurs indéterminées. Ces défauts se manifestent très souvent comme de fautes de timing. Ainsi ils seront couverts par une technique détectant les fautes de timing. Pour ces raisons, on va prendre en compte seulement ces types de fautes, si la technique de tolérance aux fautes employée pour les autres fautes peut les détecter sans surcoût matériel significatif.

Par ailleurs, *les fautes de collage logique sont détectées facilement pendant les tests de fabrication et elles n'ont pas besoin du contrôle en ligne dans la majorité des applications.* Malheureusement, les circuits auto-contrôlables ont été développés pour détecter ce type de fautes, et peuvent ne pas être efficaces pour les fautes qui nous intéressent dans cette étude. *On analysera donc, le comportement des circuits auto-contrôlables vis-à-vis de fautes transitoires, fautes de timing et défauts produisant des valeurs indéterminées, afin d'évaluer leur efficacité, et d'y remédier dans les situations où ils présentent une faible couverture de fautes par rapport à ce qui nous intéresse.*



### 3.1. Définitions

Dans les circuits auto-contrôlables, le circuit complexe est partitionné en plusieurs blocs fonctionnels, chacun de ces blocs étant implémenté d'après le principe présenté dans la figure 3.1. Cette structure consiste dans un bloc fonctionnel, fournissant des sorties qui appartiennent à un code détecteur d'erreurs. Ensuite, un contrôleur de code vérifie si les sorties du bloc fonctionnel appartiennent au code et réalise ainsi la détection concurrentielle d'erreurs.

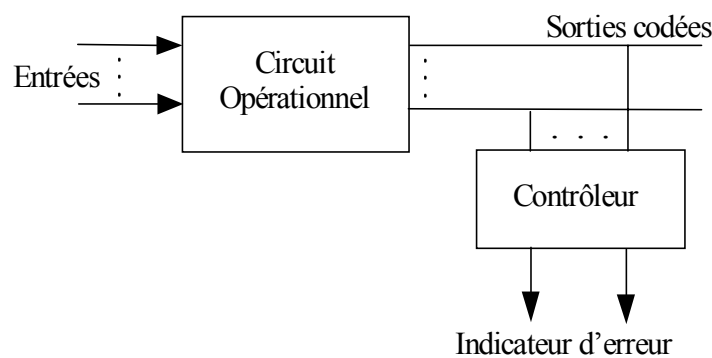


Figure 3.1. – La structure générale d'un circuit auto-contrôlable.

Dans la suite, nous présenterons les classes de circuits auto-contrôlables définissant ainsi leurs propriétés diverses. Ces propriétés ont été introduites par Carter [CAR 68] et formalisées par Anderson [AND71].

**Définition 1 : Circuit 'sûr en présence de fautes' (« Fault Secure »)** – Pour un ensemble de fautes  $F$ , un circuit  $G$  est 'sûr en présence de fautes', si pour chaque faute  $f \in F$ , les sorties erronées n'appartiennent pas au code de sortie.

Cette propriété garantit que, pour l'ensemble de fautes  $F$ , le circuit  $G$  ne génère pas de sorties erronées qui n'appartiennent pas au code, et qui sont donc détectées par le contrôleur. Une autre propriété utile est la propriété d'auto-testabilité.

**Définition 2 : Auto-testabilité (« Self Testing »)** - Pour chacun de fautes modélisés affectant le circuit  $G$ , il y a au moins un vecteur d'entrée pendant le fonctionnement normal du circuit qui produit des sorties qui n'appartiennent pas au code de sortie.

Cette deuxième propriété évite donc l'existence de fautes redondants. De telles fautes restent indétectables et peuvent être combinées à de nouvelles fautes survenant plus tard, pour donner de fautes multiples. Cependant, en présence de fautes multiples, il n'est pas garanti que le circuit reste 'sûr en présence de fautes'. Il peut donc, produire des sorties erronées non détectables. La propriété 'd'auto-testabilité' nous évite cette situation. La combinaison de ces deux propriétés dans la propriété 'totalement auto-contrôlable' offre le plus haut niveau de protection.

**Définition 3. Circuit totalement Auto-Contrôlable (« Totally Self-Checking »)** – Le circuit  $G$  est totalement auto-contrôlable s'il est 'sûr en présence de fautes' *et* 'auto-testable' pour chacune de fautes de  $F$ .

La propriété de 'sûreté en présence de fautes' est la plus importante des deux, parce qu'elle garantit la détection d'erreur pour n'importe quelle faute unique. Par contre, elle est la plus difficile à réaliser. La propriété 'd'auto-testabilité' est facile à réaliser, spécialement pour les fautes de collage logique. Une faute de collage logique qui n'est pas détectable correspond à une redondance logique. Ainsi, ces fautes peuvent être éliminés en utilisant un système de minimisation qui élimine les redondances logiques. [McC 71].

Pour les fautes transitoires, la propriété 'd'auto-testabilité' n'a pas de sens, parce qu'il n'y a pas de fautes dans le circuit, mais seulement des erreurs créées par une source externe. Dans ce cas, la seule propriété qui s'applique est la 'sûreté en présence de fautes'. Pour certains autres types de fautes, comme les courts-circuits, par exemple, il n'est pas toujours facile d'éliminer toute faute qui n'est pas détectable. Dans ce cas, et pour toute application qui nécessite un très haut niveau de sécurité, on pourra imposer des contraintes structurelles au circuit de façon à garantir qu'une faute indétectable ne détruise pas la propriété de 'sûreté en présence de fautes' [NIC 87]. Ces techniques utilisent la propriété suivante.

**Définition 4 : 'Fortement sûr en présence de fautes' (« Strongly Fault Secure »):** Un circuit  $G$  est 'fortement sûr en présence de fautes'  $F$ , si pour chaque faute  $f \in F$ , soit qu'il est 'totalement auto-contrôlable', soit qu'en présence de  $f$  il préserve la propriété de 'sûreté en présence de fautes'.

Les circuits ‘fortement sûrs en présence de fautes’ constituent la classe la plus large des circuits satisfaisant la propriété de ‘totalement auto-contrôlable’. Par contre, dans le cas de fautes spécifiques, telles que les fautes de court-circuit, les contraintes additionnelles garantissant cette propriété impliquent un surcoût matériel qui n’est pas justifié pour la plupart des applications.

Etant donné que dans cette étude nous ne recherchons pas à protéger les circuits dans n’importe quelle situation pour atteindre un niveau de sécurité très élevée, mais cherchons seulement à éviter une dégradation de la fiabilité dans les circuits VLSI, nous nous intéressons dans la suite uniquement à la propriété de ‘sûreté en présence de fautes’ qui garantit la détection d’erreurs produites après l’occurrence d’une première faute. Les circuits ‘sûrs en présence de fautes’ ayant été développés dans le passé pour prendre en compte les collages logiques, nous étudierons dans ce chapitre comment se comportent les circuits vis-à-vis des fautes transitoires, des fautes de timing et des défauts produisant des niveaux indéterminés.

### **3.2. Les codes de détection d’erreurs**

Mise à part la technique classique employée pour la détection des fautes qui utilise la duplication du circuit opérationnel et un comparateur vérifiant l’inégalité des sorties, d’autres techniques ont été développées pour réduire le surcoût d’implémentation, notamment des techniques qui emploient des codes de détection d’erreurs. Ces techniques permettent une réduction du coût d’implémentation par rapport à la duplication. Par la suite on présentera, sans entrer dans les détails, quelques-uns des codes les plus utilisés dans les techniques d’auto-contrôlabilité des circuits logiques.

#### **3.2.1. Le code de parité**

C’est le code de détection d’erreurs le plus simple et le moins coûteux puisque son implémentation consiste à ajouter un bit supplémentaire aux bits de données. Le bit ajouté,

représente la parité du nombre de « 1 » dans le mot de données considéré (bit de parité paire). Il existe également le cas complémentaire, le bit de parité impaire, dont la valeur est le complément du bit de parité impaire. Ce code détecte toutes les erreurs simples ou plus généralement les erreurs sur un nombre impair de bits. En exploitant cette limitation de la couverture de fautes, des techniques utilisant plusieurs bits de parité ont été développées pour la détection des erreurs multiples ne présentant pas un surcoût d'implémentation très élevé. La technique du code de parité est beaucoup utilisée pour la détection de fautes sur les bus de données, dans les registres, ou dans les mémoires.

### 3.2.2. Code double rail

Ce code consiste à dupliquer et à inverser les bits de données. Il permet de détecter des erreurs multiples, mais la redondance matérielle introduite est similaire à la duplication (code de redondance maximale). L'utilisation de cette technique est réduite à des applications critiques de petite surface.

### 3.2.3. Codes non ordonnés

Les codes non ordonnés détectent toutes les erreurs unidirectionnelles. Ce sont les erreurs multiples, telles que tous les bits erronés du mot comportent le même type d'erreur (soit uniquement « 0 » vers « 1 », soit uniquement « 1 » vers « 0 »). Si une telle erreur affecte un mot du code, le mot résultant n'appartient plus au code non ordonné et ainsi l'erreur est détectable. Les codes non ordonnés les plus intéressants sont les codes m-parmi-n, et le code de Berger.

Le code m-parmi-n est un code non séparable (les bits de données et de contrôle sont mélangés), et il est composé de mots de n bits contenant exactement m bits dont la valeur est égale à « 1 ».

Le code de Berger est un code séparable non ordonné où la partie de contrôle représente le nombre de « 0 » de la partie donnée. Le nombre de bits de contrôle est égal à  $\lceil \log_2(n+1) \rceil$ , pour n bits de la partie donnée.

Ces codes sont utilisés pour détecter les erreurs unidirectionnelles affectant surtout les PLA, les mémoires ROM ou les circuits logiques sans inverseur avec de la logique partagée.

### 3.2.4. Codes arithmétiques

Les codes arithmétiques sont intéressants pour les circuits arithmétiques, parce qu'ils sont préservés par les opérations arithmétiques comme l'addition, la soustraction et la multiplication. Ces codes sont divisés en deux catégories : les codes séparables et les codes non séparables. Dans les codes séparables, sur une base  $A$ , les mots codes sont obtenus en associant à la partie donnée  $X$ , une partie de contrôle  $X' = |X|_A$  (le code résidu), ou  $X' = A - |X|_A$  (le code résidu inverse). Dans les codes arithmétiques non séparables, sur une base  $A$ , on parle de mots codes égaux au produit des mots d'origine (non codés) par la base  $A$  ( $X.A$ ).

### 3.3. Les contrôleurs\*

La fonction d'un contrôleur est de signaler l'occurrence d'une erreur sur ses entrées en générant un signal d'erreur sur ses sorties et l'occurrence d'entrées correctes par un signal de bon fonctionnement. Pour cela, on considère l'ensemble des mots de sortie du contrôleur indiquant un bon fonctionnement comme l'espace code de sortie, et l'ensemble des mots indiquant une erreur comme l'espace non code de sortie. Le contrôleur doit vérifier la propriété suivante :

---

\* Cette partie est présentée pour compléter le document, mais sa lecture n'est pas nécessaire pour la compréhension des travaux que nous présentons dans le reste du document.

**Définition 5 : Codes disjoints (« Codes Disjoint ») :** Un contrôleur est à « codes disjoints » s'il associe aux mots code d'entrée des mots code de sortie, et aux mots non code d'entrée des mots non code de sortie.

Si une faute reste indétectable dans le contrôleur, il peut masquer une erreur produite dans le circuit qu'il contrôle. Afin d'éviter cette situation, le contrôleur doit être auto-testable (conformément à la définition 2).

Si entre l'occurrence de deux fautes, il s'écoule un temps suffisamment long, une faute dans un contrôleur à « codes disjoints » et « auto-testable » est toujours détectée avant que la deuxième faute apparaisse.

**Propriété :** Un contrôleur doit avoir au moins deux sorties.

Si le contrôleur a une seule sortie qui prend pour valeur  $z$  en absence d'erreur, et  $z'$  en présence d'erreur, où  $z = 0$  ou  $1$ , une erreur forçant la sortie à la valeur  $z$  n'est jamais détectée par le contrôleur. Pour cette raison, on utilise des contrôleurs à deux sorties, dont les valeurs 01 et 10 indiquent le fonctionnement correct, tandis que les valeurs 00 et 11 indiquent le fonctionnement erroné. Par conséquent, l'augmentation du nombre de sorties du contrôleur augmente sa fiabilité.

La conception d'un contrôleur « auto-testable » est une tâche difficile parce que tous les fautes internes doivent être détectées en n'utilisant que des mots code aux entrées, qui sont les mots avec lesquels le contrôleur sera exercé pendant le fonctionnement normal du circuit. Heureusement, pour les codes les plus utilisés dans la pratique, comme les codes de parité, les codes double rail, unidirectionnels, m-parmi-n, arithmétiques, etc., des contrôleurs « auto-testables » ont été proposés dans la littérature.

- **Le contrôleur de parité** est, en fait, un arbre de parité qui calcule la parité de ses entrées. Le circuit est facilement testable, puisque quatre vecteurs de tests sont suffisants pour tester un arbre composé des portes OU Exclusif à deux entrées. Néanmoins, ce contrôleur n'a qu'une sortie, et l'application des mots code aux entrées (des mots dont la parité est paire, ou des mots dont la parité est impaire) ne détecte qu'une seule de deux fautes de collage de la sortie (voir figure 3.2.a). Un contrôleur de

parité à deux sorties peut être réalisé en divisant les entrées en deux sous-groupes et en utilisant un arbre de parité pour chaque groupe (voir figure 3.2. b). Comme les entrées sont codées en parité paire dans cet exemple, et la sortie d'un arbre est inversée, les valeurs 01 et 10 indiquent le fonctionnement correct, tandis que les valeurs 00 et 11 indiquent une détection d'erreurs. Quatre mots de code sont suffisants pour tester ce contrôleur [AND 71].

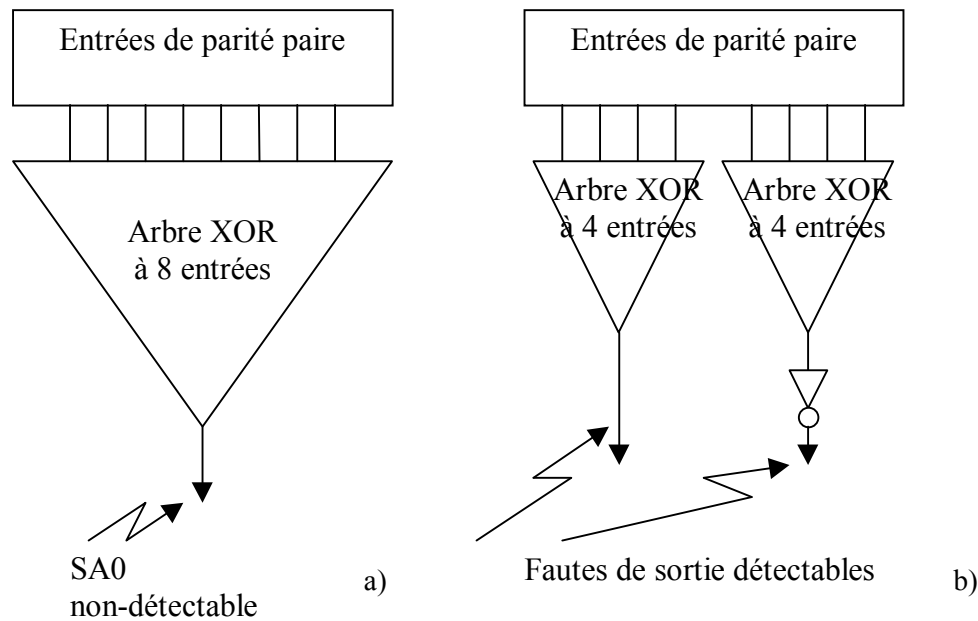
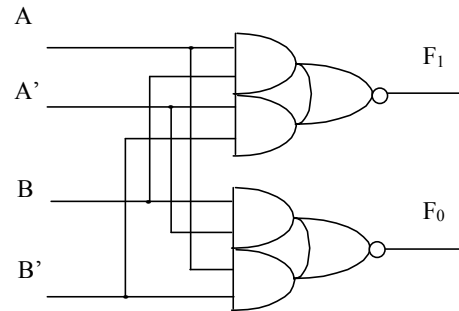


Figure 3.2. a) Contrôleur de parité non auto-testable, b) Contrôleur de parité auto-testable.

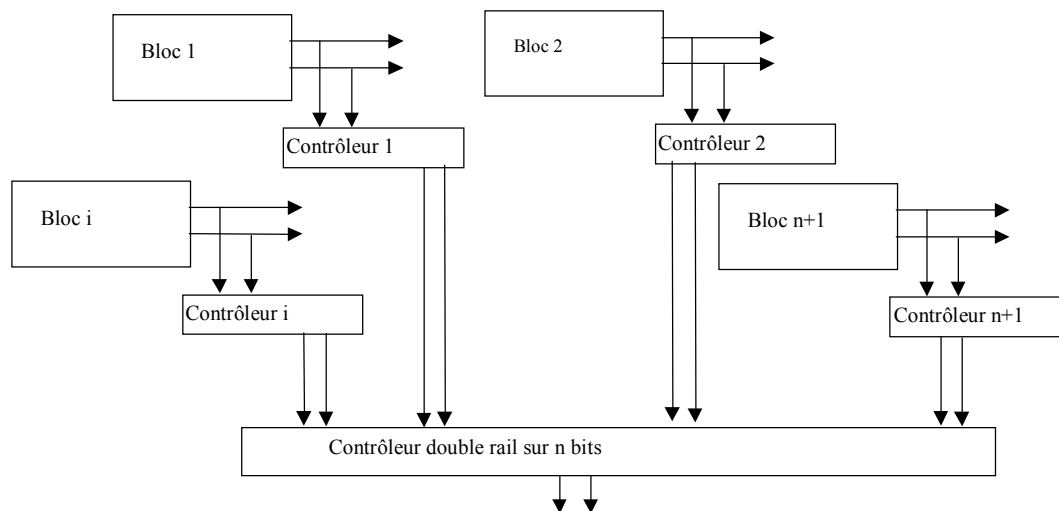
- Maintenant, nous allons présenter le **contrôleur double rail « auto-testable »**. La cellule double rail est représentée dans la figure 3.3, et l'on peut remarquer que  $F_0 = A \oplus B$ , et  $F_1 = (A \oplus B)'$ . Ce contrôleur est facilement testable puisque quatre mots de code sont suffisants pour tester un contrôleur double rail de n'importe quelle taille [AND 71].

Le contrôleur double rail est très utile dans la conception des circuits auto-contrôlables, car il peut être utilisé pour le contrôle des blocs dupliqués en inversant une sortie, ou des blocs duaux.



**Figure 3.3. Cellule de base du contrôleur double rail.**

Une autre utilisation très intéressante consiste dans la compression de plusieurs signaux d'erreurs fournis par plusieurs contrôleurs dans un circuit auto-contrôlable complexe. Chaque contrôleur du système fournit une paire de signaux codés en double rail, voir figure 3.4, puis un contrôleur double rail peut compresser les paires fournies par ces contrôleurs en une seule paire codée en double rail.



**Figure 3.4. Compactage des signaux d'erreurs dans un contrôleur double rail.**

- Les codes arithmétiques les plus utilisés dans les circuits auto-contrôlables sont les codes résiduels à coût minimal. Le circuit de contrôle pour les codes basés sur le code résiduel (partie entourée par une ligne en pointillé dans la figure 3.5) est généralement composé d'un générateur modulo A, qui génère le résidu du résultat de l'opération, un



bloc d'inverseurs, un traducteur et un contrôleur double rail fournissant le signal d'erreur. Les codes résiduels à coût minimal sont les codes utilisant une base de type  $A=2^k-1$  [AVI '73].

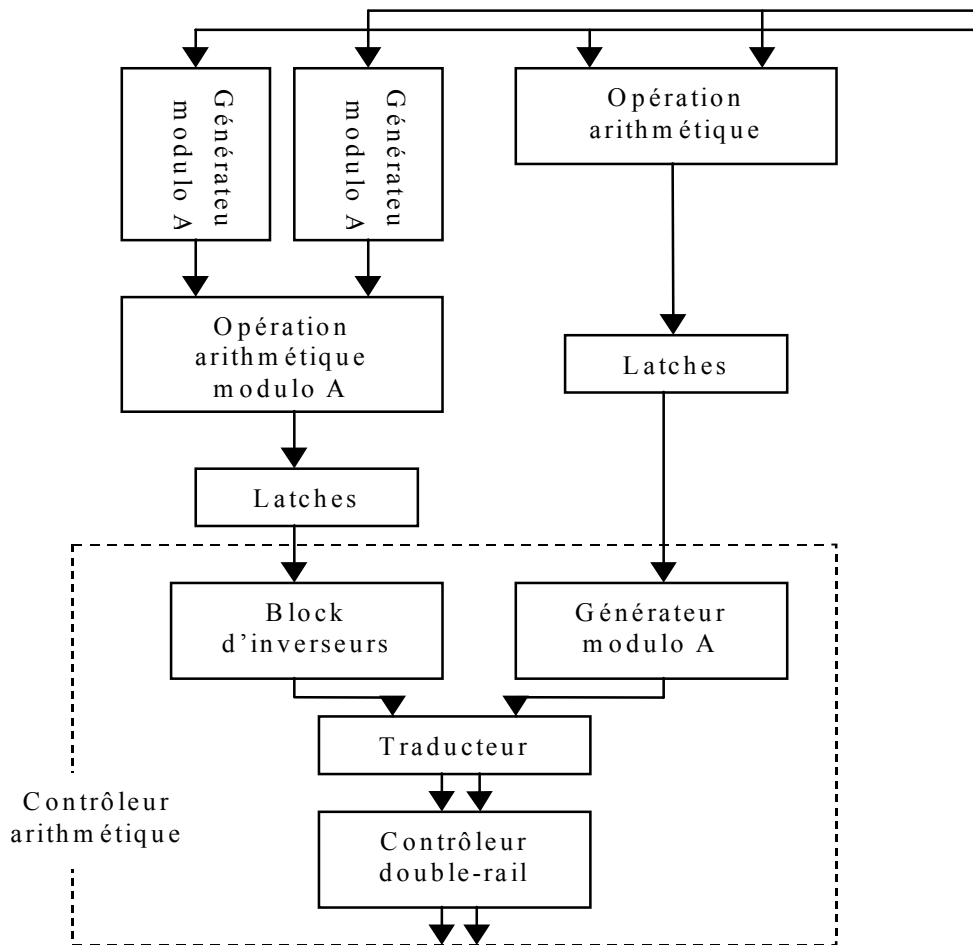


Figure 3.5. Structure d'un circuit auto-contrôlable basé sur le code résidu.

- Le **contrôleur pour le code de Berger** utilise un circuit (un générateur) qui reçoit comme entrées la partie de données du mot code et qui génère comme sorties la partie contrôle. Ensuite un contrôleur double rail compare cette partie de contrôle avec la partie de contrôle provenant du mot code. Ce contrôleur est auto-testable pour les fautes affectant le bloc générateur, et plus généralement ce bloc est à codes disjoints pour tout faute affectant le bloc générateur. Une implémentation typique pour le circuit générateur consiste en un compteur de «1», dont la réalisation même consiste en un réseau contenant des cellules d'additionneurs complets et des demi-additionneurs.

### **3.4. Les circuits ‘sûrs en présence de fautes’ face aux fautes à comportement complexe [ANG’00b]**

Les circuits auto-contrôlables sont conçus de manière que le bloc fonctionnel soit ‘sûr en présence de fautes’ pour les fautes de type collage logique. Cependant, les fautes de collage logique sont facilement détectables par les techniques ATPG, ainsi que par les techniques de DFT et de BIST. Donc, elles seront éliminées pendant les tests de production et ne nécessitent pas la détection en ligne pour les applications commerciales. En ce qui concerne les autres types de fautes, les considérations ne sont plus les mêmes. Dans le cas de fautes transitoires, de fautes de timing ou de défauts induisant des niveaux logiques indéterminés, la détection des erreurs en provenance de ces fautes peut être toujours nécessaire à cause de leur impact sur la fiabilité des systèmes conçus dans une technologie nanométrique.

Pour la suite, on considère des circuits logiques synchrones pour lesquels les sorties des circuits sont capturées par des latches ou des bascules. En conclusion, les valeurs traitées par les contrôleurs sont des valeurs logiques, même si certaines valeurs non-déterminées peuvent apparaître aux sorties du circuit combinatoire à cause d’une faute. Si les sorties du circuit ‘auto-contrôlable’ avaient des valeurs non déterminées, d’autres problèmes pourraient apparaître pendant l’interprétation de ces niveaux logiques non déterminés pour lesquels des solutions sont proposées dans l’article [NIC 86].

#### **3.4.1. Extension du modèle de collage logique**

Pour la simplicité, dans ce qui suit, le terme ‘sûr en présence de fautes’ sera utilisé pour appeler un circuit ‘sûr en présence de fautes’ pour le modèle de collage logique. On cherche à analyser le comportement de ces circuits face aux fautes à comportement complexe. Par une analyse attentive des circuits ‘sûrs en présence de fautes’, on montrera dans un premier temps que la propriété de ‘sûreté en présence de fautes’ reste valable pour un modèle de fautes plus large.

Soit  $N$  l'ensemble des nœuds du circuit. En exploitant la propriété de 'sûreté en présence de fautes', on affirme qu'une faute de type collage logique, qui affecte n'importe quel nœud du circuit appartenant à l'ensemble  $N$ , crée des erreurs détectables.

Considérons maintenant, l'ensemble de fautes affectant n'importe quel élément du circuit, de type : nœuds appartenant au set  $N$ , transistor, ou porte logique du circuit.

Un faute qui apparaît sur un tel type d'élément affecte un seul nœud appartenant à l'ensemble  $N$  avant que, éventuellement, elle soit propagée en plusieurs nœuds.

- Cette affirmation est évidente si l'élément affecté est un nœud de l'ensemble  $N$ .
- Si l'élément affecté est une porte logique, premièrement, la faute se propage vers un seul nœud de l'ensemble  $N$  (la sortie de cette porte logique, par exemple), avant qu'éventuellement elle soit propagée aux plusieurs nœuds de l'ensemble  $N$ .
- Si l'élément affecté est un transistor, ce transistor peut appartenir à une porte, et dans ce cas on peut appliquer le principe antérieur. Le transistor en cause peut être aussi un transistor utilisé comme interrupteur pour connecter ou déconnecter deux lignes. Dans ce cas, les terminaux du transistor sont des nœuds du circuit qui appartiennent à  $N$ . Le faute affectant le transistor, affectera par la suite les valeurs d'un de ces terminaux, en fonction de la direction de la propagation du signal à travers le transistor.

En conclusion, à un certain instant de temps, n'importe quel faute considérée perturbe d'abord la valeur d'un seul nœud du set  $N$ .

Soit  $nf$  un nœud de l'ensemble  $N$ , affecté par une faute  $f$ . On démontrera que la propriété de 'sûreté en présence de fautes' reste valable pour les fautes considérées ci-dessous s'ils ont les attributs suivants :

**Attribut 1. Faute sur un seul élément du circuit :** Chaque faute  $f$  affecte un seul élément du circuit (nœud, transistor, porte logique).

**Attribut 2. Erreur logique :** Chaque faute  $f$  crée des valeurs logiques erronées ('0' ou '1') au nœud  $nf$ .

**Attribut 3. Valeur erronée au repos :** Chaque faute  $f$  crée des valeurs erronées sur le nœud  $nf$ , qui restent stables pendant la période du cycle d'horloge pendant laquelle la valeur du nœud  $nf$  est supposée avoir fini ses transitions (phase de repos du nœud).

Les fautes dont les attributs sont présentés auparavant ont un comportement similaire à celui de fautes de collage logique, mais ils représentent une classe de fautes plus générale. Par exemple, si un faute de type collage logique (SA1 ou SA0) affecte le nœud  $nf$ , ce nœud reste bloqué à une valeur logique permanente ('1' ou '0'). Dans le nouveau modèle de fautes, le nœud  $nf$  peut changer de valeur logique d'un cycle d'horloge à l'autre. Donc, si un faute appartenant au modèle étendu, produit des erreurs au nœud  $nf$ , elles peuvent être du type 0->1 pendant certains cycles d'horloge et du type 1->0 pour d'autres cycles d'horloge, ce qui n'est pas le cas pour le collage logique.

On va utiliser ce modèle de fautes étendu comme un outil d'analyse du comportement des circuits 'sûrs en présence de fautes' pour les fautes suivants : fautes transitoires, fautes de timing et défauts qui impliquent des valeurs non déterminées au nœud  $nf$ .

Pour ce nouveau modèle de fautes, on montre que la propriété de 'sûreté en présence de fautes' est préservée.

**Théorème 1 :** Un circuit qui est 'sûr en présence de fautes' pour les fautes de type collage logique simple, reste 'sûr en présence de fautes' pour le modèle de fautes étendu, caractérisé par les attributs 1, 2 et 3.

**Démonstration:** On considère une faute  $f$  caractérisée par les attributs 1, 2 et 3, affectant un nœud  $nf$ , pendant le cycle d'horloge  $Ci$ . D'après ses attributs, la faute  $f$  induit une valeur logique erronée  $e$  au nœud  $nf$ , et cette valeur est stable pendant toute la période de stabilité du nœud  $nf$ , dans le cycle d'horloge  $Ci$ . Le comportement du faute  $f$  pendant le cycle d'horloge  $Ci$ , est le même que si un faute de collage logique à une valeur  $e$  affectait le nœud  $nf$ . Puisque le circuit est 'sûr en présence de fautes' pour toutes les fautes de collages logiques affectant les nœuds de l'ensemble  $N$ , l'erreur résultante à la sortie du circuit sera ainsi détectée. Alors, les erreurs produites par des fautes du nouveau modèle étendu sont détectables. Q.E.D

Dans la suite, nous analyserons le comportement de fautes qui ne vérifient pas l'attribut 1. En effet, on considère le cas de fautes uniques qui peuvent affecter plusieurs nœuds du circuit. Ces fautes sont les «courts-circuits» car ils court-circuitent deux nœuds qui peuvent ainsi prendre des valeurs erronées.

- Si les deux lignes reliées par une faute de court-circuit appartiennent à la même porte logique, on a une faute qui affecte une seule porte, et l'attribut 1 est vérifié.
- Pour les autres fautes de court-circuit, l'attribut 1 reste valide si la résistance du court-circuit et les éléments (transistors) connectant les deux fils aux  $V_{DD}$  et Gnd sont tels que la tension sur les lignes court-circuitées correspond à une valeur erreur logique [NIC 87]. Dans cette situation, les deux fils affectés par la faute de court-circuit auront les mêmes valeurs logiques 00 ou 11. Comme une faute de court-circuit est activée seulement si des valeurs logiques opposées apparaissent sur les deux lignes (01 or 10), alors, dans ce cas, une seule ligne peut être affectée par l'erreur. Donc, à chaque fois qu'une erreur survient, la situation est équivalente à la situation dans laquelle une seule ligne court-circuitée est affectée par la faute. Dans ce cas, l'attribut 1 reste valide, ensemble avec l'attribut 2.

En ce qui concerne l'attribut 3, il peut rester valide, et dans ce cas, la propriété de 'sûreté en présence de fautes' reste valable. Si l'attribut 3 n'est pas valide, une erreur de timing est observée.

Supposons maintenant que la faute de court-circuit crée des valeurs non déterminées sur les deux lignes court-circuitées. Pour chaque ligne affectée par la faute, les portes logiques qui reçoivent en entrée ces valeurs non déterminées peuvent les interpréter comme des valeurs logiques erronées. Cette situation sera équivalente à l'occurrence d'une erreur double sur les deux lignes. Elle peut détruire la propriété de 'sûreté en présence de fautes' [NIC 87]. Une autre situation d'erreur double est aussi le cas d'un couplage capacitif entre deux fils métalliques, situation qui pourrait également détruire la propriété de 'sûreté de fonctionnement'.

Néanmoins, l'attribut 1 est respecté par la majorité des fautes physiques, car la plupart des fautes affectent un seul fil métallique, ou un contact, ou un via, ou un transistor (par exemple, ouvertures, ou courts circuits entre la grille et l'oxyde de grille, ou entre la grille et le drain, la grille et la source ou la source et le drain).

### 3.4.2. Fautes à comportement complexe

Nous allons considérer par la suite, les fautes qui sont conformes à l'attribut 1 (la faute  $f$  affecte un seul élément du circuit : nœud, transistor ou porte logique). On range le comportement de ces circuits dans certaines classes de fautes en fonction de deux autres attributs, soit :

**La classe A :** Le comportement erroné du circuit est en conformité avec les attributs 2 et 3.

**La classe B :** Le comportement du circuit erroné est en conformité avec l'attribut 3, mais pas avec l'attribut 2.

**La classe C :** Le comportement du circuit erroné est en conformité avec l'attribut 2, mais pas avec l'attribut 3.

**La classe D :** Le comportement du circuit erroné n'est en conformité ni avec l'attribut 2, ni avec l'attribut 3.

En effet, dans cette classification, on considère des classes de comportement erroné, plutôt que les classes de fautes. On est obligé de considérer des classes de comportement erroné, au lieu de classes de fautes, car certaines fautes peuvent changer leur comportement d'un cycle d'horloge à l'autre (ce qui n'était pas le cas pour le collage logique).

Ainsi, cette classification représente le comportement possible du nœud  $nf$ , affecté par une faute conforme à l'attribut 1, pendant un cycle d'horloge donné. Pour simplifier la terminologie, on va utiliser dans la suite le terme «classes de fautes» à la place de «classe de comportement erroné». On va prendre en compte les cas suivants :

- Si, pendant la phase de repos du nœud  $nf$ , une faute induit sur  $nf$  un niveau indéterminé, on parle d'une faute de **classe B**.
- Si, pendant la phase de repos le nœud  $nf$  prend une valeur logique erronée et ce niveau logique est atteint sans excès de retard, alors on parle d'une faute de **classe A**, pour lequel la propriété de 'sûreté en présence de fautes' reste valide.
- Si, pendant la phase de repos du nœud  $nf$ , il prend une valeur logique correcte mais atteinte après un délai significatif, nous avons alors une faute de timing.

Nous allons faire par la suite, une analyse plus détaillée des fautes de timing. Ces cas considèrent une faute unique, induisant une faute de retard au nœud  $nf$  (les fautes de

retard distribuées correspondent à des fautes multiples, et dans ce cas la propriété de ‘sûreté en présence de fautes’ n’est pas garantie dans aucun des cas à cause de la multiplicité de la faute).

- Si la faute induit un retard important au nœud *nf*, et si la valeur mémorisée à n’importe quelle sortie du circuit correspond à la valeur logique erronée du nœud *nf*, alors on a une faute de **classe A** pour lequel le circuit reste ‘sûr en présence de fautes’.
- Si le retard additionnel induit au nœud *nf* est plus court de façon à ce que la valeur correcte du nœud *nf* a le temps de se propager sur un sous-ensemble des sorties du circuit, alors on a une faute de **classe C**.

En conclusion, les fautes de timing peuvent affecter le comportement du circuit de plusieurs manières, en fonction du retard développé par la faute au nœud *nf*. Dans certains cas (fautes de **classe A**), le circuit préserve la propriété de ‘sûreté en présence de fautes’, mais pour les fautes appartenant à la **classe C**, une analyse plus détaillée sera nécessaire.

Le dernier cas considéré ici concerne les fautes transitoires.

- Si l’amplitude de l’impulsion transitoire atteint un niveau logique au nœud *nf*, alors on a une faute de **classe C**.
- Si l’amplitude de l’impulsion transitoire a un niveau non déterminé, alors on a une faute de **classe D**. Dans ce cas, à cause de son amplitude et de sa durée réduites, l’impulsion transitoire sera probablement atténuée avant qu’elle n’arrive à la sortie du circuit combinatoire. Par conséquent, elle peut ne pas créer d’erreurs à la sortie du circuit.

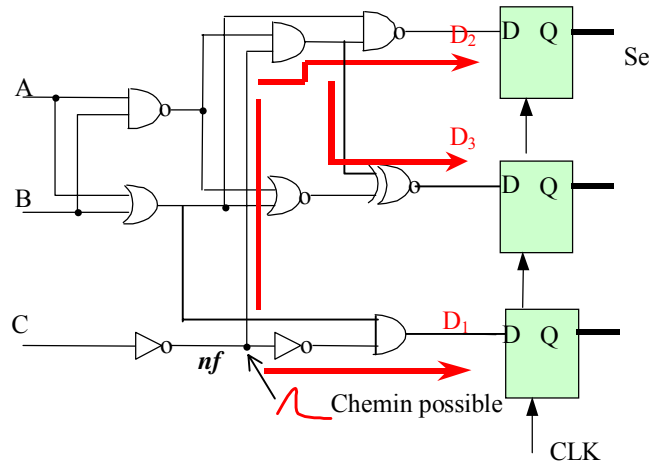
Par la suite, nous allons analyser l’effet de fautes de **classes B, C et D** sur la propriété de ‘sûreté en présence de fautes’. Cette discussion couvre tous les cas de fautes énoncés auparavant.

Afin de faciliter l’analyse d’un circuit ‘sûr en présence de fautes’, affecté par une faute de **classe B, C ou D**, on va comparer son comportement avec le comportement du même circuit en présence d’une faute *fe* de **classe A**.

Pour cette faute *fe*, on note avec *Se* l’ensemble des sorties erronées.

**Théorème 2.** Soit  $nf$  un nœud du circuit, affecté par une faute  $f$  de **classe B**, **C** ou **D**. Si les chemins de propagation, connectant le nœud  $nf$  avec les sorties du circuit, ne contiennent pas des chemins reconvergeants ayant des parités d'inversion différentes, alors l'erreur affectera seulement une partie de l'ensemble de sorties  $Se$ .

**Démonstration:** Supposons qu'une faute  $f$  de **classe B** ou **C**, affecte le nœud  $nf$  pendant le cycle d'horloge  $Ci$ . Supposons maintenant une faute  $fe$  de **classe A**, produisant une erreur logique  $e$  au nœud  $nf$ , pendant l'intervalle du cycle  $Ci$ , pendant lequel le nœud  $nf$  était supposé être à l'état stable (intervalle de repos). Cette erreur se propagera à travers tous les chemins sensibilisés connectant le nœud  $nf$  avec les sorties du circuit pendant le cycle  $Ci$  ( $D1$ ,  $D2$ ,  $D3$  dans la figure 3.6). Soit  $Se$  l'ensemble des sorties affectées par l'erreur  $e$ .



**Figure 3.6** Illustration de propagation d'une faute  $f$ , de classe **B** ou **C**.

On a les cas suivants pour la faute  $f$  :

1. Si  $f$  appartient à la **classe B**, le niveau logique non-déterminé du nœud  $nf$ , peut être interprété par certaines portes logiques comme niveau logique '0' et par d'autres portes logiques comme niveau logique '1' en fonction de leur seuil logique. Donc, il sera interprété comme valeur logique erronée seulement par certaines portes. Il sera ainsi propagé à un sous-ensemble du  $Se$ , en fonction des valeurs logiques des entrées A, B et C.



2. Si  $f$  appartient à la **classe C**, on a les cas suivants :

- Supposons qu'une faute transitoire affecte le nœud  $nf$ . La valeur logique du nœud  $nf$  est affectée pendant une durée de temps limitée (la durée de l'impulsion transitoire). L'impulsion transitoire peut se propager aux sorties appartenant à l'ensemble  $Se$ , par exemple, seulement à la sortie  $D1$  si  $(A, B, C) = (1, 1, 1)$  ou aux sorties  $D2, D3$ , si  $(A, B, C) = (0, 0, 1)$ , ou aux sorties  $D1, D2, D3$ , si  $(A, B, C) = (0, 1, 1)$ . Elle va arriver aux sorties à des moments différents (dans l'ordre  $D1, D2, D3$  si le chemin de propagation vers  $D1$  est le plus court, et celui qui mène à  $D3$  est le plus long). Donc, les erreurs seront présentes seulement dans un sous-ensemble du  $Se$ , au moment même de la capture des sorties par l'horloge du système.
- Supposons qu'un défaut crée un délai supplémentaire au nœud  $nf$ . Le délai se propagera aux sorties du circuit à travers les chemins ayant des temps de propagations différents. Le temps de propagation pour certains d'entre eux étant court, l'addition du retard supplémentaire peut ne pas induire un retard excessif. Donc, certaines sorties de l'ensemble  $Se$  ne seront pas affectées par l'erreur. De nouveau, seulement un sous-ensemble du  $Se$  sera erroné.

En conclusion, les fautes de **classes B** et **C** (par exemple, un défaut créant des niveaux non-déterminés au nœud  $nf$ , ou des fautes transitoires ou de timing de **classe C**), induisent des erreurs sur un sous-ensemble  $Se$ . On a démontré que, si les attributs 2 **ou** 3 ne sont pas valables pour une faute (donc fautes de **classe B** ou **C**), alors l'erreur à la sortie du circuit affectera seulement un sous-ensemble des sorties  $Se$ . Maintenant, si les attributs 2 **et** 3 ne sont pas valables (en fait, une faute de **classe D**), on peut facilement démontrer, que les erreurs vont affecter un sous-ensemble de sorties du  $Se$ , en combinant les arguments employés antérieurement pour chacun des attributs 2 et 3. Q.E.D

**Théorème 3:** Si le nœud  $nf$  est connecté avec plusieurs sorties du circuit à travers des chemins reconvergentes, ayant des parités d'inversion différentes, alors pour une faute appartenant aux **classes B, C** ou **D**, les erreurs peuvent affecter quelques sorties qui n'appartiennent pas à l'ensemble  $Se$ .

**Démonstration:** Soit  $Oe$  une des sorties de l'ensemble  $Se$ , mentionnées dans le théorème 3. Une faute  $fe$  (SA0) créant une erreur logique  $e$  ('0') au nœud  $nf$ , peut se propager à travers

les deux chemins reconvergent et peut générer des valeurs erronées opposées à la sortie **Oe**.

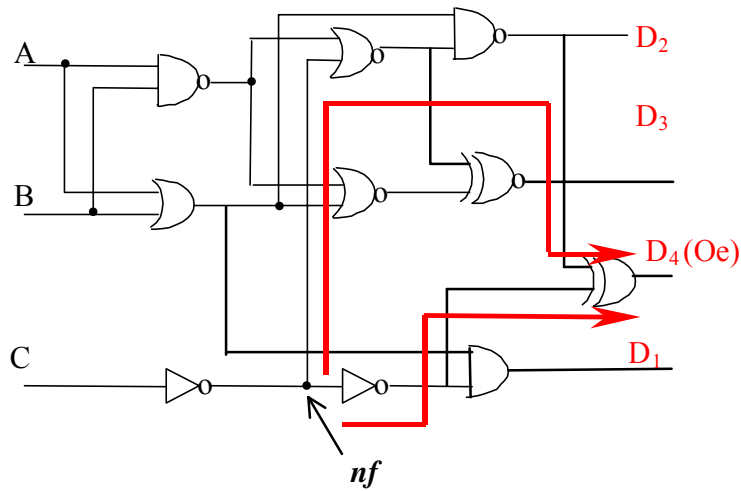


Figure 3.7. Illustration graphique du théorème 3.

Ces valeurs erronées peuvent s'annuler entre elles, résultant en une valeur globale correcte sur **Oe**. Prenons par exemple le cas du circuit présenté dans la figure 3.7, affecté par un faute créant sur le nœud **nf** une erreur logique. Pour le vecteur d'entrée (0, 1, 0) la sortie **D4** (Oe) prend pour valeur '0' pour le circuit avec ou sans faute. D'autre part, pour les fautes appartenant aux **classes B, C or D**, les erreurs propagées à travers seulement un sous-ensemble des chemins sensibles peuvent être capturées à la sortie **Oe**. Alors, l'annulation des erreurs ayant des valeurs opposées peut ne pas se produire, et une erreur peut apparaître à la sortie **Oe**. Q.E.D

Les théorèmes 2 et 3 sont très utiles pour l'analyse du comportement des différents classes des circuits 'sûrs en présence de fautes', face aux **classes de fautes B, C et D**.

### 3.4.3. Classes de circuits « sûrs en présence de fautes »

On peut partager les circuits auto-contrôlables en deux catégories : ceux qui atteignent la propriété de 'sûreté en présence de fautes' en exploitant deux attributs importants de l'erreur (la multiplicité maximale de l'erreur et la polarité de l'erreur), et ceux qui n'exploitent pas ces attributs. Les circuits de la première catégorie sont conçus de

façon à ce que, soit la multiplicité de l'erreur ne dépasse pas une certaine valeur  $t$ , soit la polarité de l'erreur vérifie certaines contraintes. Une contrainte usuelle sur la polarité de l'erreur exige que le même type d'erreur (0- $\rightarrow$  1 ou 1- $\rightarrow$  0) affecte toutes les sorties (erreur de type unidirectionnel).

Par la suite, on va présenter un théorème qui garantit la 'sûreté en présence de fautes' pour les fautes des **classes B, C et D**, dans le cas des circuits conçus à l'aide de contraintes sur la multiplicité de l'erreur et la polarité de l'erreur. Mais, définissons d'abord les deux attributs de l'erreur.

**Définition : Multiplicité de l'erreur** : le nombre de sorties du circuit affectées par l'erreur.

**Définition : Polarité de l'erreur** : la polarité de l'erreur détermine le type d'erreur affectant une sortie. Ainsi, on a la polarité 0 - $\rightarrow$  1 et la polarité 1 - $\rightarrow$  0.

**Théorème 4** : Si le nœud  $nf$  est connecté à chaque sortie du circuit à travers des chemins qui ne contiennent pas de branches reconvergentes ayant des parités d'inversion différentes, alors, la propriété de 'sûreté en présence de fautes' réalisée à l'aide des contraintes de multiplicité de l'erreur et/ou de polarité de l'erreur est préservée pour les fautes de **classe B, C ou D** affectant le nœud  $nf$ .

**Démonstration** : Nous avons vu qu'une faute  $f$ , appartenant aux **classes B, C ou D** affectant un nœud  $nf$ , est propagée aux sorties à travers un sous-ensemble de chemins à travers lesquels est propagée une faute  $fe$  appartenant à la **classe A**, qui affecte le nœud  $nf$ . Si entre le nœud  $nf$  et les sorties il n'y a pas de chemins reconvergentes avec des parités d'inversion différentes, alors l'erreur produite par la faute  $f$  affecte seulement un sous-ensemble de  $Se$  (sorties affectées par la faute  $fe$ ). Donc, la multiplicité des erreurs à la sortie ne peut pas être augmentée.

En ce qui concerne la polarité de l'erreur, elle sera la même pour chaque sortie dans le cas de deux fautes  $f$  et  $fe$  (car la valeur correcte à la sortie est la même dans les deux cas et ne peut pas donner lieu à deux polarités d'erreur différentes).

En conclusion, les fautes appartenant aux **classes B, C et D** n'augmentent pas la multiplicité de l'erreur et elles n'affectent pas les polarités de sorties. Donc, si la propriété de 'sûreté en présence de fautes' est construite à partir des ces propriétés de l'erreur, les fautes appartenant aux **classes B, C et D** n'affectent pas cette propriété. Q.E.D.

Par contre, si le nœud *nf* est connecté aux sorties du circuit à travers des chemins ayant des parités d'inversion différentes, alors, la multiplicité de l'erreur peut être augmentée par les fautes appartenant aux **classes B, C et D**. Les contraintes sur les polarités des erreurs en sortie peuvent aussi être mises en faute. En fait, les erreurs produites sur les sorties qui sont correctes dans le cas du faute de **classe A**, peuvent ne pas être en conformité avec les contraintes de polarité de l'erreur. On va montrer par la suite, que ces contraintes sur les attributs de l'erreur ne seront pas affectés pour certains circuits 'sûrs en présence de fautes', référés comme circuits 'à chemins sûrs en présence de fautes' (« path secure circuits », dans la littérature anglaise). Ainsi la propriété de 'sûreté en présence de fautes' restera vérifiée même en présence de fautes appartenant aux **classes B, C et D**.

Les circuits 'à chemins sûrs en présence de fautes' impliquent des contraintes sur la multiplicité de l'erreur et la polarité de l'erreur en utilisant deux concepts :

**Définition : La parité d'inversion des chemins connectant un nœud aux sorties du circuit [SMI 77] [SMI 78].** Il représente le nombre modulo 2 de portes logiques de type inverseur qui sont incluses dans un chemin.

La parité d'inversion d'un chemin détermine la polarité de l'erreur produite à la sortie du chemin de propagation quand une erreur de polarité donnée apparaît à l'entrée du chemin. Si l'erreur à l'entrée est 1->0, et la parité d'inversion est '0', la seule possibilité pour l'erreur à la sortie est 1-> 0, tandis que pour une parité d'inversion égale à '1', l'erreur à la sortie sera 0->1.

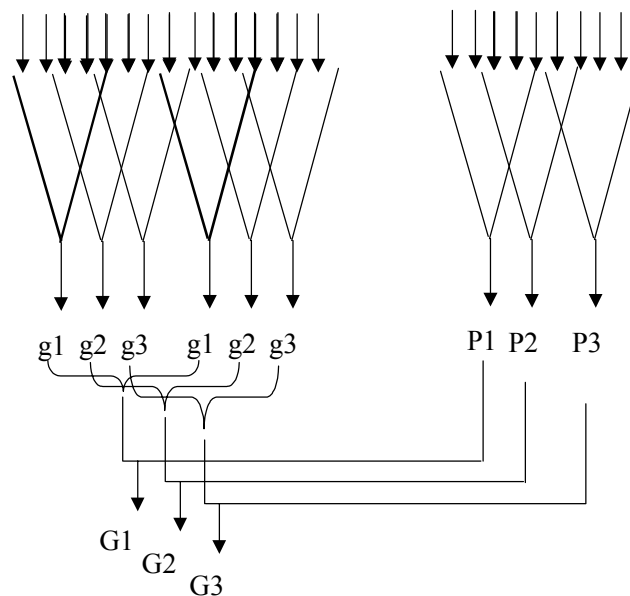
**Définition : Le degré de divergence d'un nœud du circuit [SMI 77] [SMI 78].** Il détermine le nombre de sorties connectées avec le nœud *nf*, à travers des chemins de propagation du circuit.

Si le degré maximal de divergence pour n'importe quel nœud du circuit est *md*, alors la multiplicité de l'erreur aux sorties produites par une faute affectant un seul nœud du circuit ne peut pas dépasser *md*.

Le degré de divergence d'un nœud du circuit et la parité d'inversion d'un chemin de propagation, considérés ensemble avec les codes détecteurs d'erreurs correspondants,

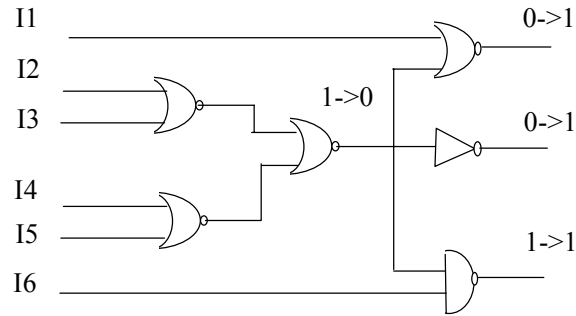
peuvent être utilisés pour la définition de plusieurs classes de circuits ‘structurellement sûrs en présence de fautes’ :

1. Le degré maximal de divergence est  $md$  et les sorties du circuit sont codées avec un code  $md$ -détecteur d’erreurs. Un cas particulier de cette classe de circuits sont les circuits utilisant un code ayant  $md=1$ . C’est le cas des circuits dont le degré maximal de divergence est égal à ‘1’, et dont les sorties sont codées avec le code de parité [SMI 77]. Un autre exemple sont les circuits à groupe de parité. Les sorties du circuit sont partagées en quelques groupes ( $g_1, g_2, g_3$ ). Chaque groupe de sorties ( $g_1$ ) est codé avec le code de parité, de parité P1, et chaque sous-circuit correspondant à un groupe de sorties a le degré maximal de divergence égale à 1. (figure 3.8) [DE 97][TOU 94][SAP 96]



**Figure 3.8** Circuit « à chemins sûrs en présence de fautes » avec les sorties codées avec un code de parité.

2. Pour chaque nœud du circuit, tous les chemins connectant le nœud avec les sorties du circuit ont la même parité d’inversion (voir l’exemple présenté dans la figure 3.9), et les sorties du circuit sont codées dans un code détecteur d’erreurs unidirectionnelles (codes non-ordonnés, par exemple le code m-parmi-n ou le code de Berger [SMI 77]).



**Figure 3.9. Circuit dont chaque nœud se connecte avec les sorties à travers des chemins ayant la même parité d'inversion.**

3. Pour chaque nœud du circuit, tous les chemins connectant ce nœud avec les sorties du circuit ont la même parité d'inversion ; le degré maximal de divergence de ce circuit est  $md$  et les sorties du circuit sont codées dans un code détectant les erreurs  $md$ -unidirectionnelles [NIC 91]. Ce groupe de circuits est une généralisation de la classe précédente.

**Théorème 5:** Dans un circuit 'structurellement sûr en présence de fautes', la propriété de 'sûreté en présence de fautes' est préservée pour une faute appartenant à la **classe B, C, ou D**.

**Démonstration:** Si le nœud  $nf$ , affecté par une faute, est connecté à chaque sortie du circuit à travers des chemins de propagation ayant la même parité d'inversion, alors pour les circuits qui sont 'sûrs en présence de fautes', par le biais de contraintes de multiplicité de l'erreur et de la polarité de l'erreur, la propriété de 'sûreté en présence de fautes' est préservée pour les fautes appartenant aux **classes B, C et D** (Théorème 4). Comme les circuits 'à chemins sûrs en présence de fautes' utilisent les contraintes de multiplicité et de polarité des erreurs, le résultat de ce théorème s'applique à ces circuits.

Il reste le cas où le nœud  $nf$  est connecté à une ou plusieurs sorties à travers des chemins de propagation ayant des parités d'inversion différentes. Néanmoins, ce cas est exclu pour les classes 2 et 3 des circuits 'structurellement sûrs', présentés auparavant. Cependant, pour le cas 1 de circuits 'sûrs en présence de fautes', le nœud  $nf$  peut être connecté à quelques sorties du circuit à travers des chemins de propagation ayant des parités d'inversion différentes. Alors la faute  $f$  peut créer des erreurs aux sorties avec une

multiplicité plus grande que celle obtenue dans le cas d'une faute *fe*, de **classe A**. Toutefois, la multiplicité ne peut pas dépasser la valeur du degré de divergence maximale *md*. Donc, la propriété de 'sûreté en présence de fautes' est de nouveau préservée. Alors, le théorème 5 est valable pour tous les circuits structurellement sûrs (cas 1, 2 et 3). Q.E.D

Dans la littérature, les circuits 'à chemins sûrs en présence de fautes' ne sont pas définis formellement. Le terme est utilisé pour caractériser les circuits des classes 1 et 2 définies plus haut [SMI 77][DE 94]. Il est intéressant de donner une définition plus générale des circuits 'à chemins sûrs en présence de fautes' qui reflète l'idée sous-jacente de ce terme et que nous pouvons utiliser par la suite pour présenter une version plus générale du théorème 5. Malgré le fait que les classes de circuits 1, 2 et 3 aient des caractéristiques totalement différentes, leurs structures ont une propriété commune, décrite dans la définition suivante, qui reflète le concept général de circuits 'à chemins sûrs en présence de fautes' :

**Définition** : Un circuit est 'à chemins sûrs en présence de fautes' si, pour chaque nœud du circuit, une erreur apparue au nœud et propagée à travers n'importe quel sous-ensemble des chemins connectant ce nœud avec les sorties du circuit, donne des erreurs sur les sorties, et ces erreurs sont détectables par le code de sortie.

Pour les fautes des **classes B, C et D**, on a pu voir (théorème 2) que les erreurs propagées à travers un sous-ensemble de chemins sensibilisés, contribuent aux erreurs apparues aux sorties. Alors, il devient évident qu'un circuit 'à chemins sûrs en présence de fautes' reste 'sûr en présence de fautes' pour les fautes appartenant aux **classes B, C et D**. On peut donc, énoncer le théorème suivant.

**Théorème 6.** Un circuit 'à chemins sûrs en présence de fautes' reste 'sûr en présence de fautes' pour les fautes des **classes B, C et D**.

Selon le théorème 6, toute structure du circuit, quelle soit connue ou à découvrir, vérifiant la propriété de 'chemins sûrs en présence de fautes', couvre les fautes de **classe B, C et D**. Elle peut donc être utilisée pour maintenir la fiabilité des circuits dans les technologies nanométriques. Mais, les circuits à 'chemins sûrs en présence de fautes' ne couvrent pas tous les circuits capables de maintenir la propriété de 'sûreté en présence de fautes' pour les

fautes de **classes B, C et D**. En fait, comme les erreurs se propagent à chaque instant donné uniquement à travers les chemins sensibilisés, il suffit de contenir les chemins sensibilisés de façon à propager uniquement des erreurs détectables. On peut donc, tout simplement, transposer la définition des circuits ‘à chemins sûrs en présence de fautes’ pour tenir compte des chemins sensibilisés.

**Définition :** Un circuit est ‘à chemins sensibilisés sûrs en présence de fautes’ si pour chaque nœud du circuit et pour chaque vecteur d’entrée, une erreur apparue au nœud et propagée à travers tout sous-ensemble des chemins sensibles connectant le nœud avec les sorties du circuit, donne toujours une erreur détectable par le code de sortie.

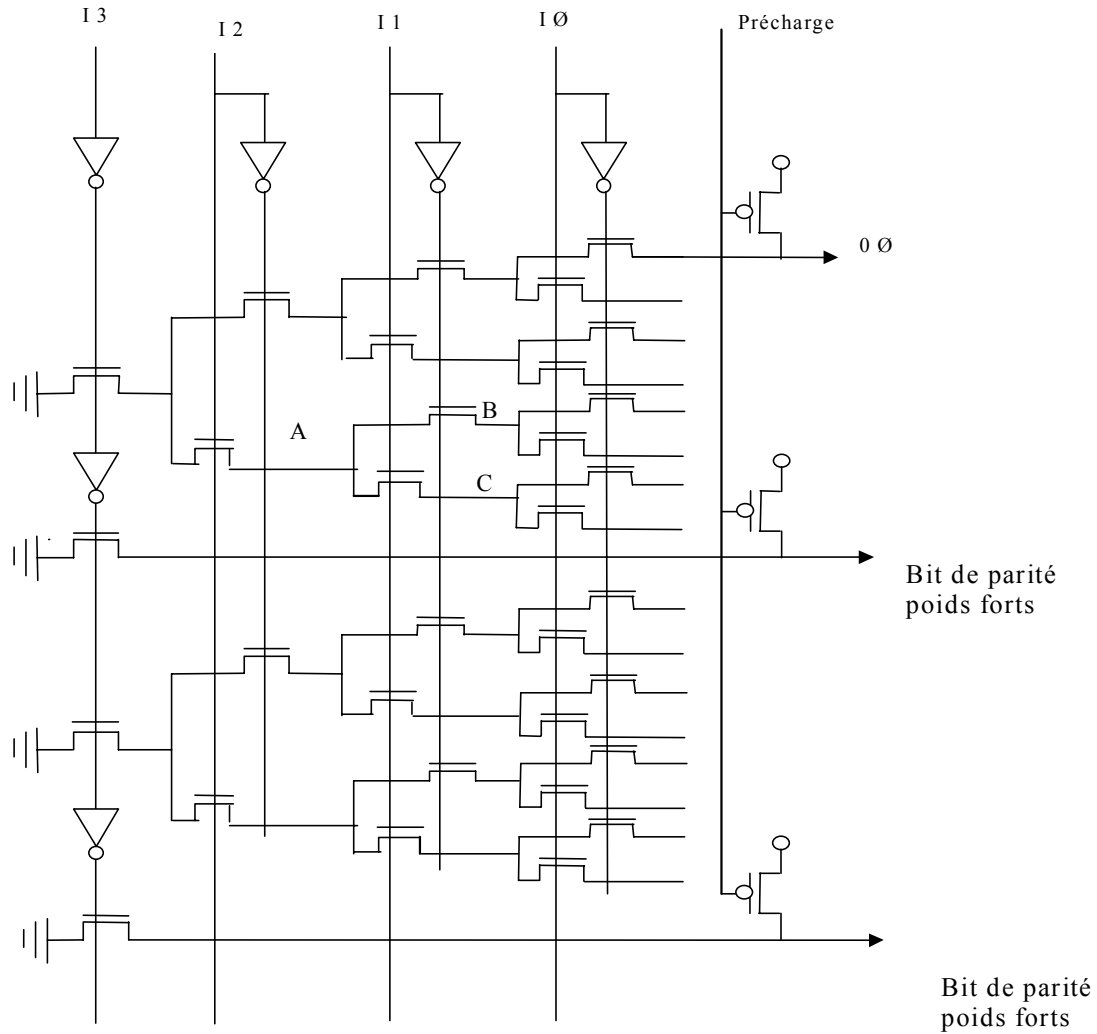
Avec cette définition, on limite les contraintes des circuits ‘à chemins sûrs en présence de fautes’ aux chemins sensibilisés à chaque instant donné. On peut vérifier qu’un circuit ‘à chemins sûrs en présence de fautes’ est en même temps un circuit ‘à chemins sensibilisés sûrs en présence de fautes’, donc les circuits ‘à chemins sûrs en présence de fautes’ sont une sous-classe de la nouvelle classe de circuits.

Par la suite, on introduit quelques cas concrets de circuits ‘à chemins sensibilisés sûrs en présence de fautes’.

1. Les sorties du circuit sont contrôlées par un code détecteur d’erreurs à multiplicité ***md***, et pour chaque vecteur d’entrée et chaque nœud du circuit, le nombre de sorties connectées avec le nœud à travers des chemins sensibilisés est plus petit ou égal à ***md*** (degré de divergence sensibilisé maximal égale à ***md***). Un cas particulier concerne les circuits dont le degré de divergence sensibilisé est égal à ‘1’ et dont les sorties sont codées avec le code de parité.

Dans la figure 3.10., on présente un circuit ‘à chemins sensibles en présence de fautes’ qui n’est pas ‘structurellement sûr en présence de fautes’. Il s’agit d’un circuit décodeur 1-parmi-n.





**Figure 3.10. Schéma du décodeur 1-parmi-n.**

Le décodeur est composé d'un réseau de transistors NMOS connectés dans un arbre binaire. Chaque niveau de l'arbre binaire est contrôlé par les valeurs directes et complémentaires de l'une des entrées du décodeur. On observe qu'un nœud appartenant à un niveau de l'arbre binaire est connecté avec deux nœuds du niveau suivant à travers deux transistors NMOS contrôlés par les valeurs directes et complémentaires de l'une des entrées du décodeur. Alors, il y a un seul chemin sensibilisé connectant un nœud de l'arbre binaire avec les nœuds du niveau suivant du même arbre. Ce principe est répété jusqu'à la sortie du décodeur, connectant à chaque fois le nœud avec le niveau suivant à travers une seule branche dans l'arbre. Donc, n'importe quel nœud du circuit est connecté aux sorties de l'arbre binaire à travers un seul chemin sensibilisé. En utilisant le code de parité, on obtient

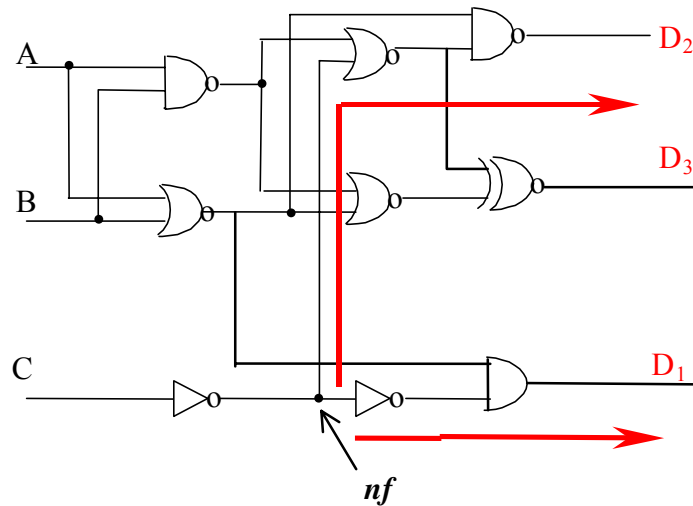
un circuit ‘à chemins sensibles en présence de fautes’. Dans la figure 3.10. l’implémentation du code de parité a été faite avec deux bits de parité, un bit contrôle les 8 bits les plus significatifs, l’autre les 8 bits les moins significatifs. La situation reste la même si les décodeurs utilisent des portes logiques.

On remarque qu’un nœud de l’arbre de transistors NMOS est connecté avec plusieurs sorties du circuit. Donc, le circuit n’est pas ‘à chemins sûrs en présence de fautes’ si on utilise le code de parité, mais il sera ‘sûr en présence de fautes’ pour la **classe de fautes B, C et D**, grâce à la propriété de ‘chemins sensibilisés sûrs en présence de fautes’.

De la même façon on peut étendre les autres classes de circuits ‘à chemins sûrs en présence de fautes’ connus pour obtenir des circuits ‘à chemins sensibles sûrs en présence de fautes’ :

2. Les sorties du circuit sont codées avec un code qui détecte toutes les erreurs unidirectionnelles, et pour chaque vecteur d’entrée et chaque nœud du circuit, le nœud est connecté aux sorties du circuit à travers des chemins sensibilisés ayant la même parité d’inversion.
3. Les sorties du circuit sont codées avec un code détectant les erreurs *md*-unidirectionnel, et pour chaque vecteur d’entrée et chaque nœud du circuit, le nœud est connecté avec les sorties à travers des chemins ayant la même parité d’inversion, et le nombre de ces chemins sensibilisés est plus petit ou égal à *md*.

Dans la figure 3.11. on présente un circuit dans lequel quelques nœuds sont connectés avec les sorties à travers des chemins qui n’ont pas les mêmes parités d’inversion. Par exemple, le nœud *nf* est connecté à la sortie **D<sub>1</sub>** à travers un chemin de propagation ayant la parité d’inversion égale à ‘1’, et à la sortie **D<sub>2</sub>** à travers un chemin de propagation ayant la parité d’inversion égale à ‘0’.



**Figure 3.11. Circuit à chemins sensibles n'ayant pas la même parité d'inversion.**

Ce circuit ne peut pas être 'à chemins sûrs en présence de fautes', en suivant les classes de ces circuits présentées antérieurement (classes 2 et 3). Par contre, cette structure peut faire partie d'un circuit 'à chemins sensibles sûrs en présence de fautes', car les chemins  $nf-D_1$  et  $nf-D_2$  ne peuvent pas être sensibilisés simultanément.

Etant donné que les erreurs se propagent seulement à travers les chemins sensibilisés, d'une façon similaire au théorème 6 pour les circuits 'à chemins sûrs en présence de fautes', on peut prouver le théorème suivant pour les circuits 'à chemins sensibilisés sûrs en présence de fautes'.

**Théorème 7 :** Un circuit 'à chemins sensibilisés sûrs' reste 'sûr en présence de fautes' pour les fautes appartenant aux **classes B, C et D**.

Par contre, si un circuit 'sûr en présence de fautes' n'est pas un circuit 'à chemins sensibilisés sûrs', une erreur apparue sur un nœud du circuit, est propagée à travers un sous-ensemble des chemins sensibilisés, pouvant donner des erreurs qui ne seront pas détectées par le code de sortie. Alors, si un circuit 'sûr en présence de fautes' n'est pas 'à chemins sensibilisés sûrs en présence de fautes', la propriété de 'sûreté en présence de fautes' peut ne plus être garantie pour les **classes de fautes B, C et D**.

Cette affirmation conduit au théorème suivant :

**Théorème 8.** Les circuits ‘à chemins sensibilisés sûrs’ sont la plus large classe de circuits ‘sûrs en présence de fautes’ qui garantit la propriété de ‘sûreté en présence de fautes’ pour les fautes appartenant aux **classes B, C et D**, donc pour les fautes d’intérêt dans les technologies sous-microniques avancées.

Les circuits ‘à chemins sûrs’ étant une sous-classe de circuits ‘à chemins sensibilisés sûrs’, on peut utiliser dans la suite le terme ‘à chemins sensibilisés sûrs’ pour englober les deux catégories.

### **3.5. Circuits ne présentant pas la propriété ‘à chemins sensibilisés sûrs’**

Considérons dans un premier temps un circuit auto-contrôlable, qui ne présente pas la propriété de ‘sûreté en présence de fautes’ pour les fautes de collage logique. Face à une faute de collage logique ou une faute appartenant à la **classe A**, un tel circuit peut se comporter de la façon suivante.:

**Cas 1.** Toutes les erreurs apparues à cause de la faute sont détectables par le code de sortie.

**Cas 2.** Certaines erreurs apparues aux sorties, produites par la faute, sont détectables et certaines ne sont pas détectables.

**Cas 3.** Aucune erreur produite par la faute n’est détectable.

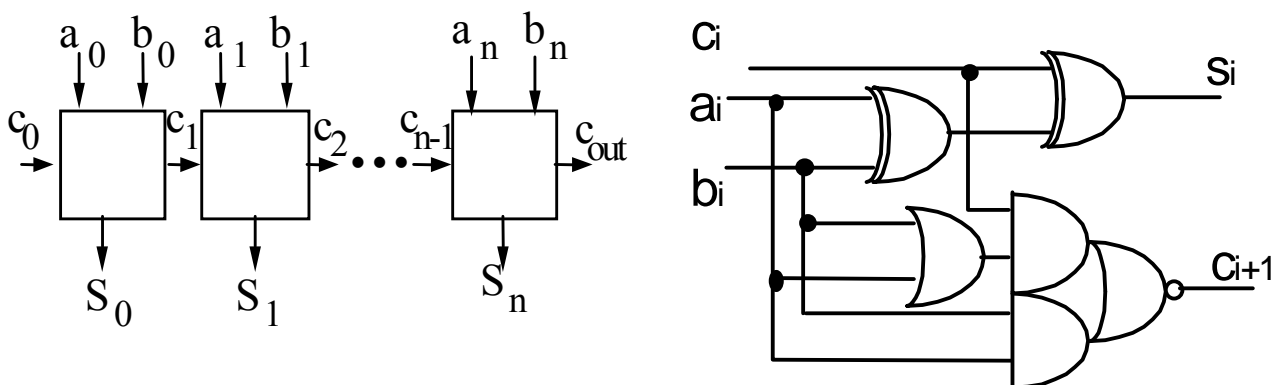
Comparons maintenant l’efficacité de détection des erreurs pour deux fautes affectant le même nœud *nf*, dont le premier appartient à l’une des **classes B, C or D** et le deuxième appartient à la **classe A**. Nous pouvons constater les situations suivantes :

- Si la faute de la **classe A** correspond au cas 1 décrit ci-dessus, alors, la faute de **classe B, C ou D**, soit maintient, soit réduit l’efficacité de détection de l’erreur.
- Si la faute de la **classe A** correspond au cas 2, alors, la faute appartenant aux **classes B, C ou D** peut maintenir, réduire ou augmenter l’efficacité de détection de l’erreur.

- Finalement, pour le cas 3, une faute de **classe B, C ou D** soit maintient, soit augmente l'efficacité de détection de l'erreur.

Cette analyse est triviale, mais elle permet de mettre en évidence une situation très intéressante : dans certaines structures de circuits, l'efficacité de détection de l'erreur peut être plus élevée pour les fautes de **classes B, C et D**, que pour les fautes appartenant à la classe A (qui incluent les fautes de collage logique). Cette observation est très importante, car elle pourrait, dans ce travail, nous permettre d'enlever les contraintes nécessaires à assurer la propriété de 'sûreté en présence de fautes' pour les collages logiques (et donc réduire le surcoût matériel), en assurant en même temps une efficacité de détection élevée pour les fautes de **classes B, C et D**.

Nous allons illustrer cette situation en considérant un additionneur à retenue propagée présenté dans la figure 3.12.a), qui utilise comme cellule de base une cellule d'additionneur complet dont le schéma est donné dans la figure 3.12.b).



**Figure 3.12.a) Additionneur à retenue propagée**      **Figure 3.12.b) Cellule d'additionneur complète**

La prédiction de la parité pour l'additionneur est calculée à partir de la relation suivante:  $P_S = P_A \oplus P_B \oplus P_C$ , (figure 3.13.), où  $P_S$  est la parité des sorties  $s_0, s_1, \dots, s_n$ ,  $P_A$  la parité de entrées  $a_0, a_1, \dots, a_n$ ,  $P_B$  la parité de entrées  $b_0, b_1, \dots, b_m$ , et  $P_C$  la parité des retenues  $c_0, c_1, \dots, c_{n-1}$ .

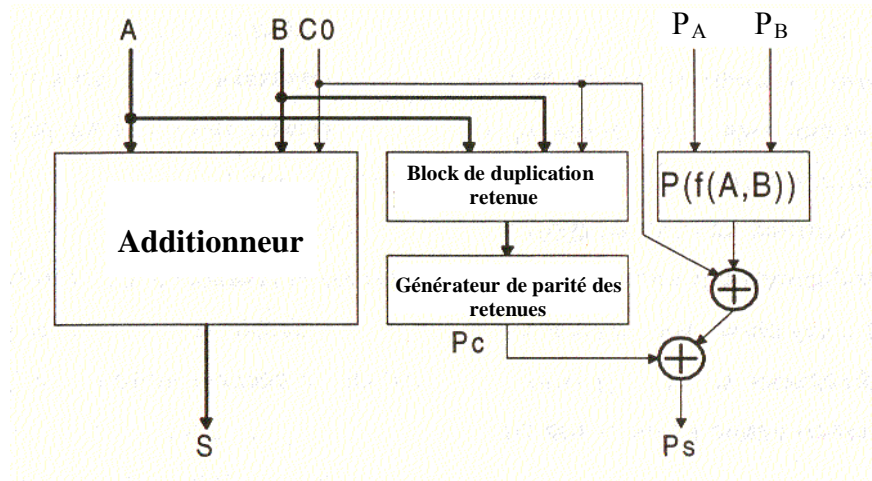


Figure 3.13. Schéma de prédiction de parité pour un circuit arithmétique (additionneur)

Chaque nœud appartenant aux portes logiques XOR dans la figure 3.12.b a un degré de divergence égal à '1', car il est connecté seulement à la sortie  $S_i$ . Donc, toutes les erreurs produites par une faute sur les deux portes XOR sont détectables, peu importe la classe de fautes (**classe A, B, C ou D**). Cela correspond à 60 % environ du nombre total de fautes (les deux portes XOR dans la cellule de l'additionneur complet correspondent à 60 % du coût matériel de l'additionneur à retenue propagée). Pour le reste de 40 % des fautes (i.e. ceux qui affectent les portes générant les signaux  $c_{i+1}$  dans la figure 3.12.b), la détection de l'erreur dépend de la façon dont on implémente le circuit de prédiction de la parité. Considérons une faute de collage logique (ou une faute appartenant à la **classe A**), affectant le circuit qui génère  $c_{i+1}$ . Une première erreur sera apparue sur le signal  $c_{i+1}$ . Ensuite, l'erreur peut être propagée sur plusieurs signaux de type retenue ( $c_{i+2}, \dots, c_{i+k}$ ). Finalement, chaque erreur sur une retenue  $c_r$ , est propagée à travers les portes XOR vers les sorties  $S_r$ . Ainsi, si  $P_C$  est généré à partir des retenues normales de l'additionneur, nous allons avoir le même nombre de bits erronés aux sorties de l'additionneur et aux entrées du circuit de prédiction de la parité. On a donc un nombre total d'erreurs pair. On constate que l'on ne peut pas détecter les erreurs produites par une faute de **classe A** qui affecte les portes logiques générant les signaux  $c_{i+1}$ . Cette situation correspond à une efficacité de détection très basse pour les fautes de la **classe A**, y compris les collages logiques (les erreurs générées par 40 % des fautes ne sont pas détectées). Pour résoudre ce problème, on réalise la prédiction de la parité en dupliquant le circuit de génération des retenues (figure 3.14.).

Pour un additionneur à retenues propagées, le surcoût est significatif et peut devenir encore plus élevé pour les additionneurs très rapides comme l'additionneur carry-lookahead.

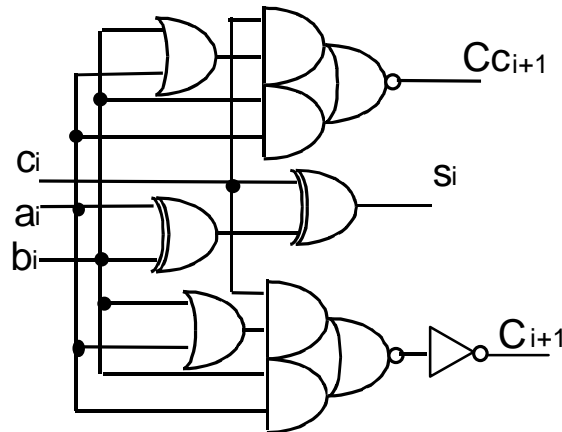


Figure 3.14. Cellule d'additionneur complet à retenue dupliquée.

Dans le cas de fautes appartenant aux **classes B, C and D**, ce surcoût matériel peut ne pas être nécessaire pour deux raisons :

- Si la parité est prédite à partir des retenues normales de l'additionneur, un faute qui affecte le circuit de génération de retenues (figure 3.12.b) crée dans une première étape des erreurs sur le signal  $c_{i+1}$ . Comme, pour les fautes appartenant aux **classes B, C ou D**, l'erreur n'est pas propagée par le même nombre de chemins que dans le cas des fautes de la **classe A**, alors, le nombre total d'erreurs affectant les sorties du circuit  $S_i$  et la parité  $P_C$  peut devenir impaire, résultant en une détection d'erreur. D'un point de vue probabiliste, la moitié de ces erreurs peut être détectée. L'efficacité de la détection sera cette fois de 80 %, au lieu de 60 % dans le cas de fautes de collage logique.
- Si la parité est prédite à partir des retenues dupliquées, alors, le circuit détecte 100 % des erreurs, si les fautes sont de type collage logique. En fait, pour les collages logiques, la structure à retenues dupliquées, réalise la propriété de 'sûreté en présence de fautes' en forçant un nombre impair d'erreurs qui affectent les sorties de l'additionneur et, par conséquent, les entrées du circuit de prédiction de la parité [NIC 97] [NIC 00]. Mais, pour un faute de **classe B, C ou D**, la multiplicité de l'erreur change, en réduisant l'efficacité de détection. On

peut démontrer que l'efficacité de détection reste la même seulement dans le cas des retenues non-dupliquées, c'est à dire 80 % des cas.

En conclusion, le surcoût matériel assez important résultant de l'implémentation des retenues dupliquées n'offre aucune amélioration au niveau de la détection d'erreurs pour les **classes de fautes B, C ou D**. On évite donc la duplication des retenues, et on peut par ailleurs augmenter l'efficacité de détection au delà de 80 %, en ajoutant des bits de parité supplémentaires. Cette solution a un coût inférieur au coût correspondant à la duplication des retenues, tout en augmentant l'efficacité de détection pour les fautes de **classes B, C et D**.

Pour conclure cette section, en ce qui concerne les circuits 'à chemins sûrs', on n'a pas besoin d'analyse d'efficacité, parce que ces circuits assurent une efficacité de détection de 100 % pour les fautes de **classes B, C et D**, comme pour les fautes de collage logique. Pour les circuits qui assurent une efficacité de détection à 100 %, cas de fautes de collage logique (circuits 'sûrs en présence de fautes'), mais qui ne sont pas 'à chemins sûrs' ni 'à chemins sensibles sûrs', on risque d'observer une perte significative au niveau de l'efficacité de détection. Ces circuits nécessitent une analyse de l'efficacité plus détaillée, à l'aide de la simulation de fautes. Les implémentations qui ne sont pas 'sûres en présence de fautes' pour les collages logiques peuvent, pour les **classes B, C et D** offrir une efficacité de détection similaire ou plus élevée que dans le cas des circuits qui sont 'sûrs en présence de fautes' pour les fautes de collage logique. D'autres solutions de redondance, à part celles utilisées pour les collages logiques, peuvent être plus efficaces pour les fautes appartenant aux **classes B, C et D**.

### **3.6. Résultats expérimentaux**

Dans ce sous-chapitre, nous analysons divers circuits auto-contrôlables, qui ne sont pas 'à chemins sensibilisés sûrs', pour déterminer l'efficacité de détection des fautes des **classes B, C et D**. Des modifications du mécanisme de détection sont aussi proposées pour améliorer l'efficacité de détection, et/ou réduire le surcoût matériel.



Des implémentations ‘à chemins sûrs’ sont souvent utilisées pour les fonctions logiques arbitraires, telles que les circuits logiques utilisés dans les parties contrôle des processeurs. Par contre, pour les circuits arithmétiques, les implémentations de circuits ‘à chemins sûrs’ ne sont pas efficaces et le surcoût matériel est excessif. D’autres solutions sont ainsi développées pour ces circuits, telles que la prédiction de parité combinée avec des techniques qui forcent les fautes de collage logique à se propager dans un nombre impair de sorties pour des additionneurs, multiplieurs et diviseurs [NIC 93][NIC 97][NIC 99], ou des codes arithmétiques pour les additionneurs et multiplieurs [AVI 73][ALZ 99]. Ces implémentations assurant la propriété de ‘sûreté en présence de fautes’ pour les collages logiques, ne sont pas ‘sûrs en présence de fautes’ pour les **classes de fautes B, C et D** (théorème 8). Nous avons donc évalué l’efficacité de ces solutions pour les fautes de **classes B, C et D**, en utilisant la simulation de fautes. En particulier, nous avons évalué cette efficacité pour les fautes transitoires, qui seront les fautes dominants dans les technologies nanométriques.

Les expérimentations sont faites à l’aide d’une simulation extensive de fautes transitoires. Nous avons utilisé pour cela l’outil de simulation décrit dans le chapitre 2. Lors de la simulation, nous avons utilisé des impulsions transitoires d’une durée de 0.4 ns, durée plus grande que le cas typique d’impulsions transitoires créées par l’impact d’une particule énergétique. Nous avons analysé un nombre important de circuits ‘sûrs en présence de fautes’ (additionneurs et multiplieurs), par exemple, des additionneurs à prédiction de la retenue [NIC 93], et des multiplieurs Wallace et Booth-Wallace avec prédiction de parité [NIC 97][NIC 99]. Les circuits ‘sûrs en présence de fautes’ ont été synthétisés à l’aide d’un outil existant dans le laboratoire, et présenté dans les articles : [NIC 97][NIC 99][ALZ 99]. Cet outil a été étendu pour implémenter des additionneurs et multiplieurs ‘non-sûrs en présence de fautes’, qui pourraient permettre d’augmenter l’efficacité de détection d’erreurs pour les fautes de **classes B, C et D**, tout en nécessitant un coût d’implémentation réduit par rapport aux solutions des circuits ‘sûrs en présence de fautes’ de collage logique.

Comme les fautes transitoires ont une occurrence aléatoire, ils peuvent affecter n’importe quel nœud à n’importe quel moment pendant le cycle d’horloge. Dans les simulations de fautes effectuées, l’instance d’injection de l’impulsion transitoire est distribuée uniformément pendant toute la durée du cycle d’horloge. Le point d’injection de l’impulsion transitoire a été choisi aléatoirement, et des simulations ont été faites pour un

nombre très élevé de vecteurs d'entrées aléatoires. Un total de  $2 \cdot 10^6$  simulations ont été faites pour chaque circuit sélectionné.

Les tableaux 3.1 et 3.2 montrent les résultats d'expérimentation pour les additionneurs et multiplieurs sûrs en présence de fautes logiques utilisant la prédiction de parité. Les additionneurs considérés utilisent des différents principes de génération de la retenue, par exemple des unités de type Carry Look-Ahead [HWA 79][WES 94], Brent et Kung [BRE 82], Kogge et Stone [KOG 73], Han et Carlson [HAN 87] et Sklanski [SKL 60]. Toutes les simulations ont été faites pour les circuits à 16 bits d'entrée. Pour assurer la propriété de sûreté en présence de fautes de collage logique, les additionneurs utilisent les schémas de contrôle/prédiction de la retenue [NIC 93][NIC 97].

Comme on peut voir dans le tableau 3.2 (colonne 'sûr en présence de fautes logiques'), les additionneurs/multiplieurs 'sûrs en présence de fautes' logiques ne donnent pas une efficacité de détection de 100 % pour les fautes transitoires. En effet, cette efficacité est proche à la prédiction approximative de 80 % que nous avons faite dans la section 3.4. Nous observons aussi un surcoût matériel significatif (tableau 3.1 colonne 'sûr en présence de fautes'). D'après la discussion présentée dans la section 3.4., nous avons implémenté aussi les additionneurs 'non-sûrs en présence de fautes' logiques dans lesquels la parité est prédite à partir des retenues normales du circuit. On observe un surcoût matériel réduit (tableau 3.1) et une efficacité de détection supérieure qui valide l'analyse de la section 3.4. Pour plus de détails, la troisième colonne du tableau 3.1. montre que la solution 'non-sûre en présence de fautes' utilisant 1 bit de parité offre une efficacité de détection similaire à la solution 'sûre en présence de fautes logiques' mais pour un surcoût matériel beaucoup plus faible (moins du tiers du surcoût de la solution des circuits 'sûrs en présence de fautes' logiques).

Pour améliorer l'efficacité de détection, nous avons considéré des schémas d'additionneurs à deux bits de parité. Le premier bit de parité calcule la parité des positions impaires de la sortie du circuit, tandis que le deuxième bit de parité calcule la parité des positions paires. Nous avons obtenu une efficacité de détection plus élevée par rapport à la version du circuit 'sûr en présence de fautes' logiques, et le surcoût matériel reste toujours plus faible. Finalement, la dernière colonne des tableaux montre les résultats pour un circuit 'non-sûr en présence de fautes' utilisant 4 bits de parité. L'efficacité de détection est

améliorée de façon significative, par rapport aux circuits ‘sûrs en présence de fautes’ logiques, et le surcoût d’implémentation est plus faible.

|                     | <b>Sûr en présence de fautes logiques</b> | <b>NS en présence de fautes et 1 bit de parité</b> | <b>NS en présence de fautes et 2 bits de parité</b> | <b>NS en présence de fautes et 4 bits de parité</b> |
|---------------------|---|--|---|---|
| Brent et Kung       | 28.25%                                    | 8.20%  | 13.09%  | 21.3%   |
| Carry Lookahead     | 29.85%                                    | 9.22%  | 14.89%  | 23.2%   |
| Han et Carlson      | 27.91%                                    | 7.89%  | 12.72%  | 20.3%   |
| Kogge et Stone      | 25.80%                                    | 6.79%  | 10.71%  | 17.5%   |
| Sklanski            | 26.67%                                    | 6.68%  | 11.58%  | 18.7%   |
| Multiplieur Wallace | 32.22%                                    | 16.1%  | 17.3%   | 19.4%   |
| Booth-Wallace       | 32,52%                                    | 16.8%  | 16.9%   | 18.6%   |

**Tableau 3.1. Surcoût matériel pour les additionneurs avec contrôle de parité/prédiction de parité et multiplieurs Wallace et Booth-Wallace.**

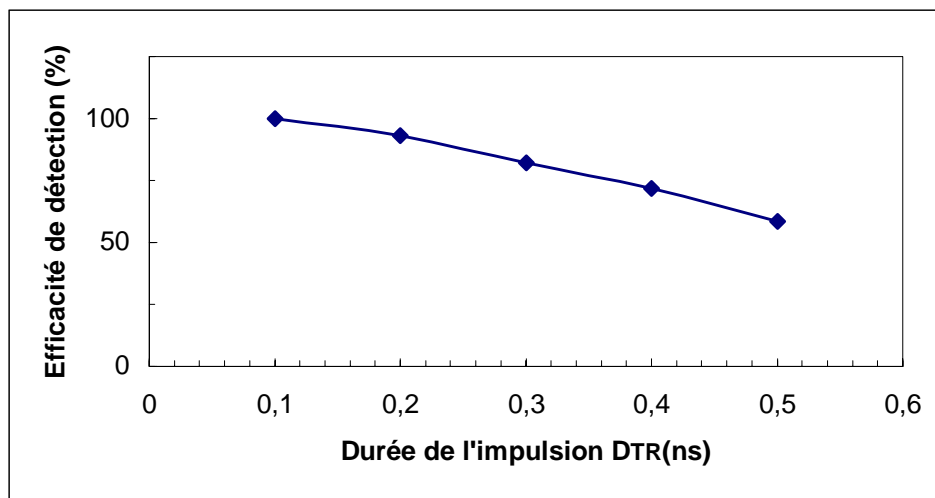
|                     | <b>Sûr en présence de fautes logiques</b> | <b>NS en présence de fautes et 1 bit de parité</b> | <b>NS en présence de fautes et 2 bits de parité</b> | <b>NS en présence de fautes et 4 bits de parité</b> |
|---------------------|---|--|---|---|
| Brent et Kung       | 79.7%                                     | 82.1%  | 85.2%   | 93.8%   |
| Carry Lookahead     | 78.5%                                     | 74.8%  | 84.3%   | 92.8%   |
| Han et Carlson      | 80.3%                                     | 77.9%  | 83.17%  | 93.5%   |
| Kogge et Stone      | 74,7%                                     | 72.2%  | 82.9%   | 93.2%   |
| Sklanski            | 76,5%                                     | 81.3%  | 83.5%   | 90.1%   |
| Multiplieur Wallace | 71.9%                                     | 76.5%  | 77.4%   | 80.1%   |
| Booth-Wallace       | 72.8%                                     | 77.1%  | 79.2%   | 82.5%   |

**Tableau 3.2. Efficacité de la détection de l’erreur pour les additionneurs avec contrôle de parité/prédiction de parité et multiplieurs Wallace.**

Les deux dernières lignes des tableaux 3.1 et 3.2 montrent les résultats pour des multiplieurs de type Wallace et Booth-Wallace à 16 bits utilisant la prédiction de parité. On observe les mêmes tendances que pour les additionneurs.

Ces résultats sont obtenus en injectant des impulsions transitoires d’une durée égale à 400 ps. En réalité, les impulsions transitoires provenant des impacts de particules ionisantes sur les circuits sont plus courtes (plus proches de 100 ps).

Dans la figure 3.15. on montre l'efficacité de détection d'un additionneur 'sûr en présence de fautes' (1 bit de parité), pour une durée variable de l'impulsion transitoire.



**Figure 3.15. Efficacité de détection d'un additionneur 'sûr en présence de fautes' en fonction de la durée de l'impulsion transitoire.**

On observe une amélioration de l'efficacité de détection avec la réduction de la durée de l'impulsion. Nous pensons que cette amélioration est due au fait que plus l'impulsion est courte, plus la probabilité que l'impulsion soit capturée par un latch est réduite. Ainsi, la réduction de l'impulsion transitoire diminue la multiplicité d'erreurs, favorisant les erreurs simples, qui sont détectées par le code de parité. Mais une analyse plus détaillée est nécessaire pour confirmer cette hypothèse. La figure 3.15 montre que les résultats du tableau 3.2 sont pessimistes (ils s'améliorent si on utilise des durées d'impulsion plus réalistes). On devrait donc, obtenir une efficacité de détection élevée en utilisant un nombre faible de bits de parité, donc moins coûteux en matériel.

## **Conclusion**

La conception des circuits 'sûrs en présence de fautes' est utilisée pour la réduction du surcoût d'implémentation matérielle obtenu lors d'une implémentation à l'aide de la redondance massive (duplication). Les circuits 'sûrs en présence de fautes' ont été proposés pour la détection de fautes de collage logique. Ces solutions s'avèrent moins efficaces pour les fautes caractérisant les technologies submicroniques de l'avenir. Afin de pouvoir

évaluer l'efficacité de ces circuits face aux fautes transitoires, fautes de timing et fautes provoquant des niveaux indéterminés, nous avons développé un modèle de fautes généralisant le modèle de collage logique. En utilisant ce modèle comme outil d'analyse, nous avons déterminé la plus large classe de circuits 'sûrs en présence de fautes' qui préserve cette propriété face aux autres modèles de fautes. Nous avons appelé cette classe : circuits 'à chemins sensibilisés sûrs'. Pour les circuits 'sûrs en présence de fautes' qui ne font pas partie de cette classe, l'efficacité de détection peut être compromise. Nous avons utilisé notre outil de simulation (chapitre 2) pour évaluer cette efficacité dans le cas de fautes transitoires, et nous avons proposé des solutions qui offrent une meilleure efficacité de détection que les circuits sûrs en présence de fautes logiques, et ceci à moindre coût matériel.



**Chapitre IV****Circuits tolérants les perturbations transitoires à l'aide des redondances temporelles**

L'avancée technologique de ces dernières années vers le domaine du nanométrique a permis l'intégration d'un nombre de plus en plus important de dispositifs sur la même puce. La fréquence d'horloge du système augmente avec chaque génération technologique, et la tension d'alimentation diminue pour réduire la puissance dissipée. Cependant ces évolutions entraînent une diminution très importante des marges de bruit et affectent sérieusement la fiabilité des circuits. De plus, les circuits deviennent de plus en plus sensibles aux facteurs externes, et plus particulièrement aux radiations cosmiques. Les radiations atmosphériques et les particules alpha provenant du matériel d'encapsulation affecteront de façon inacceptable le fonctionnement des nouveaux systèmes. Ainsi, la tolérance aux fautes s'impose afin de rendre les circuits plus robustes aux facteurs d'environnement, et de permettre la continuation de la loi de Moore.

Dans le chapitre précédent, nous avons étudié les techniques de circuits auto-contrôlables, avant de pouvoir implémenter la détection concurrentielle des erreurs avec un coût matériel plus faible que la duplication. Notre premier objectif étant de se protéger contre les fautes transitoires et les fautes de timing, on pourrait exploiter la nature temporelle de ces fautes afin de proposer des techniques de détection et de tolérance utilisant une redondance temporelle, donc à moindre coût matériel.

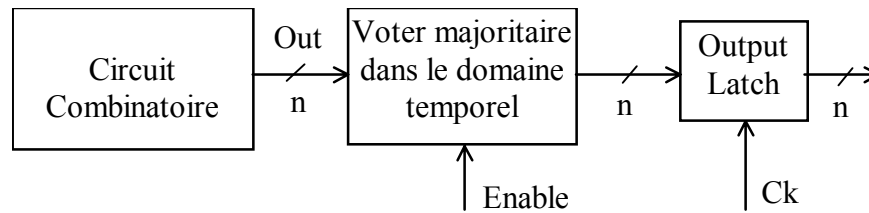
Dans ce chapitre nous étudierons plusieurs techniques de redondance temporelle permettant la tolérance aux fautes transitoires. Ces techniques nécessitent un coût matériel très inférieur à la triplication, mais imposent une réduction de la vitesse fonctionnement non négligeable. La combinaison des redondances matérielle et temporelle est ensuite étudiée afin de réduire la perte en vitesse par un coût matériel plus élevé. Finalement, nous étudions une technique de redondance temporelle permettant la détection des erreurs produites par les fautes transitoires et les fautes de timing. Cette technique nécessite un faible coût matériel et n'entraîne aucune perte de vitesse.

#### ***4.1. Tolérance aux fautes transitoires basée sur la redondance temporelle***

La théorie de l'information spécifie que dans le but de pouvoir distinguer une valeur correcte d'une valeur incorrecte du même signal, ce signal ou plus généralement l'information, doit être au minimum triplé. Dans le domaine matériel, le principe standard de la tolérance aux défauts prévoit trois copies du même système, leurs sorties étant ensuite votées.

Pour les systèmes et produits à faible valeur, tels que les produits destinés au grand public, le coût de cette méthode n'est pas acceptable. Une réduction très importante de ce coût matériel peut être obtenue en exploitant la nature temporelle de la faute transitoire. Comme ces fautes se manifestent dans une durée de temps limitée, nous allons concevoir des circuits ayant leurs valeurs de sorties stables pour une durée de temps convenablement plus grande que la durée d'une faute transitoire (typiquement, quelques centaines de picosecondes). Ensuite, la sortie de ce circuit sera contrôlée par un circuit de vote majoritaire fonctionnant dans le domaine temporel pour déterminer la valeur présente à la sortie du circuit pour la majorité de temps. La figure 4.1 montre cette solution pour un circuit à  $n$  entrées, où on utilise  $n$  "voters" connectés sur le bus de sortie du circuit. Une implémentation triviale du "voter" utilisera un intégrateur.





**Figure 4.1. Circuit tolérant aux perturbations utilisant le principe de redondance temporelle.**

Supposons, par exemple, qu'à l'instant  $t_0$  les sorties du circuit combinatoire sont déjà à l'état stable (les signaux se sont propagés à travers le chemin critique). Soit  $D_{TR}$  la durée de l'impulsion transitoire à la sortie du circuit. Cette durée d'impulsion dépend de plusieurs causes. Dans le cas des SEUs, elle dépend du type de la particule incidente, de son énergie et de l'inclinaison de la trajectoire, de la technologie de fabrication et de la configuration du circuit.  $D_{TR}$  peut être déterminée par des simulations mixtes (simulation de procès technologique 3D combinés à des simulations électriques de type SPICE).

Soit  $\epsilon$  la durée minimum par laquelle la valeur correcte doit dépasser la durée de la valeur incorrecte pour qu'elle soit choisie par le voter sans aucune ambiguïté.  $\epsilon$  dépend de la façon dont on réalise le circuit de vote. On conclut que le circuit doit être implémenté de façon à ce que, pendant le fonctionnement sans faute, les résultats d'un cycle de calcul du circuit combinatoire soient maintenus à leurs valeurs stables pour une durée au moins égale à  $2D_{TR} + \epsilon$ . Ainsi, s'il survient une faute transitoire dont la durée ne dépasse pas  $D_{TR}$ , la valeur correcte sera présente pour une durée au moins égale à  $D_{TR} + \epsilon$ . La différence des durées des valeurs correctes et incorrectes sera plus grande que  $\epsilon$ . Si on appelle  $D_{max}$  le délai maximum du circuit combinatoire,  $D_{voter}$  le délai maximum du circuit de vote,  $D_{hold}$  et  $D_{setup}$  les temps de hold et de setup du latch, on trouve une période d'horloge égale à  $D_{max} + D_{setup} + D_{hold} + 2D_{TR} + \epsilon + D_{voter}$ . La période de l'horloge d'une implémentation non tolérante aux fautes étant  $D_{max} + D_{setup} + D_{hold}$ , on trouve que la tolérance aux fautes transitoires est obtenue au prix d'un allongement de la période du cycle d'horloge égal à  $2D_{TR} + \epsilon + D_{voter}$ .

### 4.1.1. Implémentation du principe de fonctionnement à l'aide des portes logiques

Dans la figure 4.2.a on présente une réalisation du principe temporel de tolérance aux fautes utilisant uniquement des bibliothèques de cellules digitales standard. On utilise trois latches par sortie, dont les fronts actifs de l'horloge sont décalés d'une durée  $\delta$  l'un par rapport à l'autre. On capture ainsi les sorties du circuit à des instants différents pour éviter qu'une faute transitoire soit capturée par plus d'un latch à la fois. Si on prend en compte les temps de setup des latches (les durées de temps avant et après le front de l'horloge pendant lesquelles la valeur d'entrée du latch doit être stable), et si les deux latches reçoivent la même valeur d'entrée, alors que leurs signaux d'horloge sont retardés de  $\delta$ , une perturbation d'une durée plus petite que  $\delta - D_{setup} - D_{hold}$  ne pourra pas être capturée par plus d'un latch.

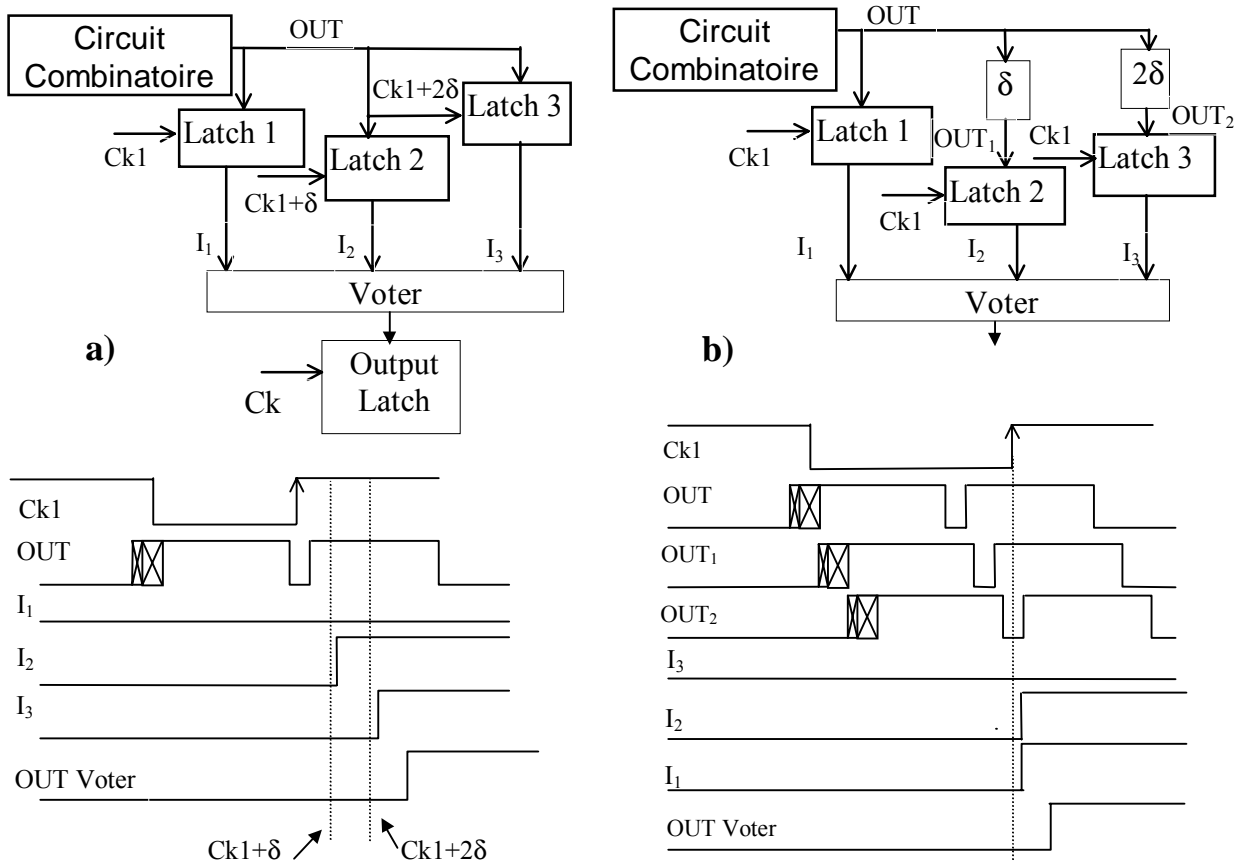


Figure 4.2. Implémentation au niveau des portes du principe présenté dans la figure 4.1.

Le concepteur détermine la durée maximale de l'impulsion transitoire  $D_{TR}$ , qui doit être tolérée afin d'assurer un niveau acceptable de tolérance aux erreurs soft, et calcule  $\delta = D_{TR} + D_{setup} + D_{hold}$ . Ensuite, les horloges des trois latches capturent la sortie OUT du circuit combinatoire de la manière suivante : d'abord, la sortie OUT est capturée dans le Latch1 à l'instant  $t_0$ , avec l'horloge Ck1 ; ensuite, la sortie OUT est capturée dans le Latch2, à l'instant  $t_0 + \delta (Ck1 + \delta)$  ; et enfin dans le Latch3 à l'instant  $t_0 + 2\delta (Ck + 2\delta)$ . Finalement, les données sont mémorisées dans le latch de sortie Output Latch à l'instant  $t_0 + 2\delta + D_{voter} = t_0 + 2D_{TR} + D_{voter} - 2D_{setup}$ .

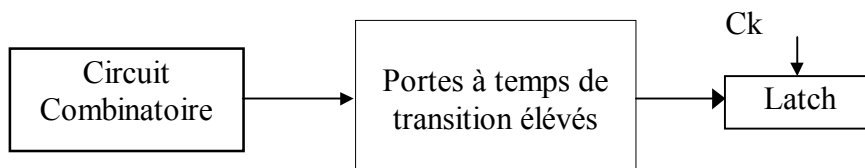
Il est évident que les impulsions transitoires de durée plus petite que  $D_{TR}$  ne pourront pas affecter plus d'un latch à la fois (Latch1, Latch 2 ou Latch 3) et dans ce cas une valeur correcte sera retrouvée dans le Output Latch (dans la figure 4.2.a on a considéré que la valeur du Latch1 était affectée par l'impulsion transitoire). Cette solution protège toutes les parties du circuit, sauf le latch de sortie Output Latch. Si la protection de ce latch est nécessaire, on doit prévoir un latch durci aux radiations [ROC '88], [WHI '91],[BES '94],[CAL '96], [NIC '97].

Dans la figure 4.2.b on montre une autre possibilité d'implémentation. Les trois latches (Latch1, Latch2 and Latch3) utilisent le même signal d'horloge (Ck1). L'entrée du Latch2 ( $OUT_1$ ) est retardée de  $\delta$  par rapport au signal OUT et celle du Latch3 ( $OUT_2$ ) de  $2\delta$ . De même, la valeur  $\delta$  est spécifiée en fonction de la durée de l'impulsion transitoire  $D_{TR}$  et de la durée de temps de setup et de hold du latch utilisé ( $\delta = D_{TR} + D_{setup} + D_{hold}$ ). De nouveau, une impulsion transitoire d'une durée plus petite que  $D_{TR}$  ne peut affecter plus d'un latch et donc, la sortie du circuit de vote est correcte. Dans cette implémentation, la période de l'horloge Ck1 sera allongée d'un retard égal à  $2D_{TR} + 2D_{setup} + 2D_{hold} + D_{voter}$ , en comparaison de l'horloge du circuit sans tolérance aux fautes.

Une autre implémentation du principe de la figure 4.2. est basée sur la propriété des portes logiques d'atténuer les perturbations transitoires ayant une durée plus petite que la moitié de leur temps de transition logique. Les impulsions transitoires de durée plus grande que le temps de propagation d'une porte se propagent à travers cette porte sans atténuation [BAZ '97]. En ce qui concerne les impulsions transitoires de durée plus petite qu'une certaine valeur, (approximativement la moitié du temps de la transition logique de la porte)

elles ne seront pas propagées à la sortie de cette porte, tandis que ces impulsions transitoires de durées intermédiaires se propageront vers la sortie du circuit avec une certaine atténuation.

Pour exploiter ces propriétés, on peut concevoir un circuit combinatoire en utilisant, au lieu de portes générant les sorties du circuit combinatoire, des portes de temps de transition logique plus élevés (voir figure 4.3.). Afin d'atténuer les impulsions transitoires affectant les entrées de ces portes, et aussi les nœuds du circuit combinatoire (voir figure 4.3.), les portes logiques doivent vérifier la condition :  $D_{portes} > 2D_{TR} + D_{marg}$ , où  $D_{portes}$  est le plus petit des temps de transition logique parmi les deux possibilités de transition logique (transitions de la porte de type 1->0 et 0->1), et où  $D_{marg}$  est une marge de sécurité. Comme la relation  $D_{portes} > 2D_{TR}$  exprime en fait l'approximation de la condition de l'élimination des impulsions transitoires (la valeur exacte dépend des caractéristiques du circuit),  $D_{marg}$  est utilisé ici pour corriger cette approximation.



**Figure 4.3. – Implémentation utilisant des portes logiques à temps de transition élevés.**

#### 4.1.2. Surcoût matériel et dégradation de vitesse. Couverture des fautes

Le surcoût matériel de la figure 4.2. correspond à deux latches supplémentaires et un circuit de vote majoritaire par sortie du circuit. Pour un circuit de petite taille, ce surcoût est très élevé (par exemple 78,13% pour un additionneur Carry Lookahead à 8 bits d'entrée). Il diminue lorsque le nombre de bits d'entrée augmente (64,31% pour le même additionneur sur 32 bits). S'il s'agit d'un circuit de taille plus importante (multiplicateur Wallace), le surcoût d'implémentation est plus faible, se situant à 14,4% pour un

multiplieur Wallace 16 x 16 et 6,35% pour le multiplieur 32 x 32. Donc, de façon évidente, cette méthode sera évitée pour les circuits de petite taille.

Dans le cas de la figure 4.3. le coût matériel reste assez faible car il s'agit simplement de remplacer les portes de sortie par des portes similaires à temps de propagation plus élevés.

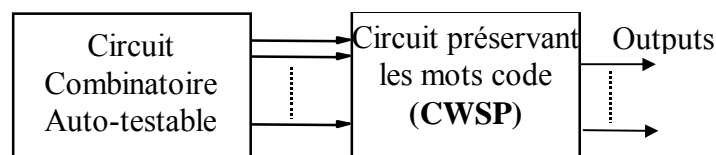
En ce qui concerne la dégradation des performances, on note, pour la figure 4.2., une réduction de la vitesse de fonctionnement du système correspondant au temps de propagation du voter et aux retards supplémentaires introduits par les éléments de retardement du signal d'horloge. Dans le cas de la figure 4.3., la réduction de vitesse est due aux portes de sortie à retard de propagation plus grand que le double d'une impulsion transitoire de durée  $D_{TR}$ . Par exemple, pour un circuit qui pourra fonctionner à 500 MHz la tolérance aux défauts transitoires d'une durée de 100 ps, introduira une dégradation de la vitesse de fonctionnement de l'ordre de 10 %. Pour éviter une telle dégradation, l'utilisation de ce principe doit être restreinte aux parties qui se trouvent en dehors des chemins critiques du système. Ce principe ne sera pas applicable aux circuits très rapides, dépassant 1 GHz, ou dans le cas de la tolérance des impulsions transitoires plus larges (par ex., 400 ps).

## ***4.2. Circuits tolérants les perturbations combinant la redondance matérielle et temporelle***

Afin d'étendre le domaine d'applications du principe de tolérance aux défauts basé sur la redondance temporelle, mais sans pour autant payer le coût de la triplication, nous pourrions essayer d'échanger une partie du coût en vitesse contre un coût supplémentaire en matériel. Ainsi, dans cette section, on se propose de réaliser la tolérance aux fautes transitoires partiellement à l'aide des moyens matériels, et partiellement à l'aide d'une redondance temporelle.

### 4.2.1. Principe de fonctionnement

Le principe consiste en la combinaison d'un schéma de détection d'erreurs (circuits auto-contrôlables), avec la redondance temporelle. Les circuits auto-contrôlables nécessitent un coût matériel moindre que la triplication, mais ils offrent seulement la détection d'erreur. Sachant que les circuits auto-contrôlables distinguent les valeurs correctes de sorties (mots code) de celles qui sont incorrectes (mots hors code), on va utiliser cette capacité de différenciation pour pouvoir sélectionner les valeurs correctes. Pour réaliser cela, on peut exploiter le fait que les résultats erronés (mots hors code) apparaissent sur les sorties du circuit pendant une courte durée de temps, alors que le reste du temps les sorties sont correctes (mots code). Dans cette situation, pour extraire les valeurs correctes à partir des signaux perturbés, on se propose d'observer les sorties du circuit auto-testable à l'aide de circuits asynchrones séquentiels capables de sélectionner le mot dans le code (voir figure 4.4) [NIC 99].



**Figure 4.4: Circuit tolérant les perturbations à l'aide d'un circuit préservant les mots code.**

Un principe de fonctionnement pour le circuit de sélection des mots du code, que nous appellerons **CWSP** (pour code word state preserving element en anglais), est le suivant.

- Si un mot du code apparaît aux entrées du circuit **CWSP**, le circuit produit à sa sortie un état précis déterminé par le fonctionnement correct.
- Si un mot hors code apparaît aux entrées du circuit **CWSP**, ses sorties préservent leurs valeurs antérieures.

De plus, on doit rallonger la période de l'horloge afin d'assurer la tolérance aux fautes transitoires. En effet, en augmentant suffisamment la durée de la période d'horloge, on peut

s'assurer qu'il y a toujours, soit avant l'occurrence de l'impulsion transitoire, soit après elle, une période de temps pendant laquelle les entrées du circuit reçoivent le mot correct (mot appartenant au code). Ainsi, si le mot correct est présent après l'occurrence de l'impulsion transitoire, les valeurs correctes seront générées aux sorties du circuit **CWSP** et ne seront plus perturbées, jusqu'au moment où elles seront capturées par les latches. Si de l'autre côté, le mot correct (mot du code) est présent après l'occurrence de l'impulsion transitoire, le circuit va générer des valeurs correctes à ses sorties. Dans le cas où une impulsion transitoire transformera le mot présent aux entrées du circuit **CWSP** en un mot hors code, le circuit **CWSP** va préserver à ses sorties les valeurs correctes selon le principe de fonctionnement du circuit **CWSP** annoncé ci-dessus.

Nous allons présenter ensuite, de façon plus précise, les propriétés nécessaires et suffisantes de transition d'état du circuit **CWSP** permettant de garantir la tolérance aux fautes transitoires, sous la condition que la période de l'horloge soit allongée comme précisé ci-dessus.

1 – Pour chaque mot d'entrée appartenant au code (**CWi**), et pour n'importe quel état présent, l'état suivant du circuit sera un état prédéterminé (**di**), correspondant à l'opération sans fautes ;

2 – Pour chaque mot d'entrée de type non-code (**NCWi**) obtenu à partir d'un mot code **CWi** affecté par une impulsion transitoire et pour l'état présent **di**, correspondant au mot code d'entrée **CWi**, l'état suivant est **di** ;

3 – L'état suivant correspondant à n'importe quel autre transition d'états, est 'X' (« don't care » en anglais).

Ces propriétés nous permettent de construire facilement les circuits **CWSP**.

Voici un exemple de construction : on considère un circuit combinatoire ayant une porte NAND à deux entrées dans l'étage de sortie. Pour transformer ce circuit en un circuit tolérant aux perturbations, on peut concevoir la partie combinatoire qui génère les entrées des portes NAND comme un circuit auto-contrôlable avec le code de parité. Le circuit auto-contrôlable a donc trois sorties, les deux entrées originales de la porte NAND et le bit de parité. Le circuit ainsi conçu, en présence d'une seule faute, génère des erreurs simples. C'est par exemple en circuit 'à chemins sensibilisés sûrs' (voir chapitre 3) : les sorties

erronées ayant pour cause une faute transitoire sont des mots n'appartenant pas au code de parité.

On construit la porte NAND comme un élément **CWSP** à trois entrées. Les mots du code d'entrée pour cet élément sont les quatre mots de trois bits ayant une parité paire et les mots hors code d'entrée de l'élément sont les quatre mots restants (mots à parité impaire). La matrice de transition d'état est obtenue à partir des propriétés décrites antérieurement (points 1, 2 et 3). Elle est présentée dans le tableau 4.1. Les quatre premières lignes du tableau correspondent aux transitions d'états à vecteurs d'entrées des mots code. Selon le point 1, pour ces entrées, l'état suivant ne dépend pas de l'état présent. On a donc marqué l'état présent comme 'X' (« don't care »). Les sept lignes suivantes correspondent aux transitions d'états à mots d'entrées hors code à partir des mots code affectés par une faute unique. L'état présent est l'état correspondant à l'entrée mot code. La dernière ligne du tableau ne peut pas se produire si une impulsion transitoire affecte un seul signal (erreur simple), alors l'état suivant est 'X' (« don't care »).

| Entrées | Parité | Etat présent | Etat ultérieur |
|---------|--------|--------------|----------------|
| 00      | 0      | X            | 1              |
| 01      | 1      | X            | 1              |
| 10      | 1      | X            | 1              |
| 11      | 0      | X            | 0              |
| 00      | 1      | 1            | 1              |
| 01      | 0      | 1            | 1              |
| 10      | 0      | 1            | 1              |
| 11      | 1      | 1            | 1              |
| 11      | 1      | 0            | 0              |
| 10      | 0      | 0            | 0              |
| 01      | 0      | 0            | 0              |
| 00      | 1      | 0            | X              |

**Table 4.1: Tableau de transition d'états pour un élément CWSP avec le code de parité et réalisant la fonction NAND à deux entrées.**

#### *Cas du code de duplication*

L'exemple de construction présenté précédemment nécessite de générer un bit de parité pour les entrées de toutes les portes logiques présentes sur le dernière étage du circuit combinatoire. On génère donc autant de bits de parité que de sorties du circuit. Ainsi, le



coût de cette solution pourra dépasser le coût de la duplication. Pour cette raison, on adopte dans cette section le code de duplication, dont le surcoût est fixe et égal à 100 %, et dont l'implémentation est très simple. On cherche alors à concevoir un circuit **CWSP** pour le code de duplication. Ce circuit réalise une fonction booléenne **G** pour les entrées appartenant au code de duplication et préserve l'état de sortie antérieur si les entrées sont hors du code de duplication. La matrice de transition d'état du circuit **CWSP** présente les caractéristiques suivantes :

- 1- Si les deux copies des entrées sont identiques (mot du code), l'état suivant est la sortie spécifiée par la fonction booléenne **G** ;
- 2- Si les deux copies des entrées ne sont pas identiques (mots qui n'appartiennent pas au code), l'état suivant sera l'état présent.

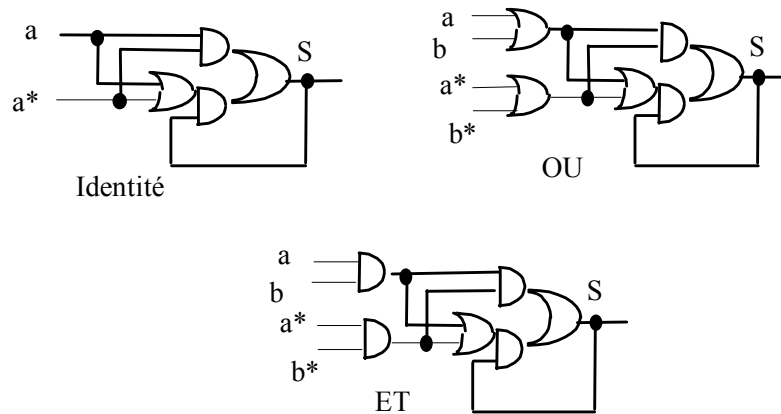
Le tableau 4.2. présente les matrices de transition d'état pour les portes logiques les plus utilisées dans la conception des circuits VLSI (l'inverseur, la fonction logique NOT OU, et NOT ET). Il présente aussi la matrice de transition d'états pour la fonction « identité » (fonction qui transfère les entrées aux sorties). Dans ce tableau, I représente une copie de l'entrée de la porte et I\* la deuxième copie.

| INVERSEUR |         |           |           | IDENTITE |         |           |           | NOT OU |         |           |           | NOT ET |         |            |           |
|-----------|---------|-----------|-----------|----------|---------|-----------|-----------|--------|---------|-----------|-----------|--------|---------|------------|-----------|
| Ent. I    | Ent. I* | Etat Prés | Etat Ult. | Ent. I   | Ent. I* | Etat Prés | Etat Ult. | Ent. I | Ent. I* | Etat Prés | Etat Ult. | Ent. I | Ent. I* | Etat. Prés | Etat Ult. |
| 0         | 0       | X         | 1         | 0        | 0       | X         | 0         | 00     | 00      | X         | 1         | 00     | 00      | X          | 1         |
| 1         | 1       | X         | 0         | 1        | 1       | X         | 1         | 01     | 01      | X         | 0         | 01     | 01      | X          | 1         |
| 1         | 0       | 1         | 1         | 1        | 0       | 1         | 1         | 10     | 10      | X         | 0         | 10     | 10      | X          | 1         |
| 1         | 0       | 0         | 0         | 1        | 0       | 0         | 0         | 11     | 11      | X         | 0         | 11     | 11      | X          | 0         |
| 0         | 1       | 1         | 1         | 0        | 1       | 1         | 1         | 00     | ≠00     | 1         | 1         | 00     | ≠00     | 1          | 1         |
| 0         | 1       | 0         | 0         | 0        | 1       | 0         | 0         | 00     | ≠00     | 0         | 0         | 00     | ≠00     | 0          | 0         |
|           |         |           |           |          |         |           |           | 01     | ≠01     | 1         | 1         | 01     | ≠01     | 1          | 1         |
|           |         |           |           |          |         |           |           | 01     | ≠01     | 0         | 0         | 01     | ≠01     | 0          | 0         |
|           |         |           |           |          |         |           |           | 10     | ≠10     | 1         | 1         | 10     | ≠10     | 1          | 1         |
|           |         |           |           |          |         |           |           | 10     | ≠10     | 0         | 0         | 10     | ≠10     | 0          | 0         |
|           |         |           |           |          |         |           |           | 11     | ≠11     | 1         | 1         | 11     | ≠11     | 1          | 1         |
|           |         |           |           |          |         |           |           | 11     | ≠11     | 0         | 0         | 11     | ≠11     | 0          | 0         |

**Table 4.2: Les matrices de transition d'états pour les blocs CWSP utilisant le code de duplication et réalisant les fonctions d'inverseur, identité, NOT-OU, et NOT-ET.**

Les éléments **CWSP** basés sur le code de duplication peuvent être conçus en utilisant des portes logiques appartenant à une bibliothèque de cellules standard. Par exemple, la figure 4.5. présente les éléments **CWSP** correspondants à la fonction

d'identité, et aux deux fonctions à deux entrées ET et OU, réalisées d'après la matrice de transition d'états (tableau 4.2).

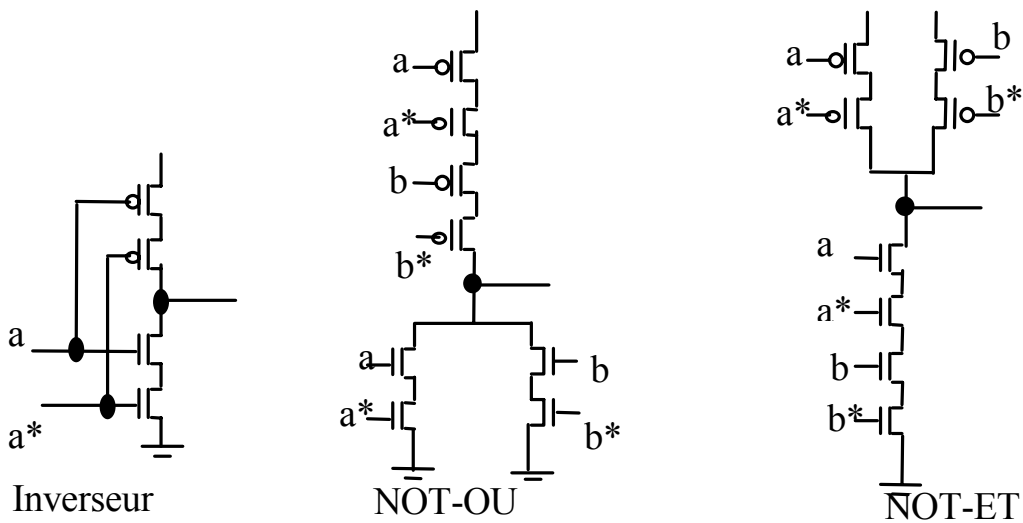


**Figure 4.5: L'élément CWSP utilisant le code de duplication et réalisant la fonction d'identité, OU et ET.**

En recherchant une solution plus compacte (niveau transistor), on peut observer que si on modifie une porte logique standard en remplaçant chaque transistor par une paire de transistors connectés en série et contrôlés par des signaux d'entrée dupliqués, la porte résultante a un comportement équivalent à un élément **CWSP** utilisant le code de duplication. En effet, dans une telle porte, si les entrées d'une paire de transistors sont égales, les deux transistors se comportent comme un seul transistor contrôlé par l'un des deux signaux dupliqués. On a donc un fonctionnement équivalent à la porte standard d'origine. D'autre part, si la paire de transistors reçoit des entrées dupliquées, la nouvelle porte se comporte comme la porte originale. Cette situation correspond au comportement d'un élément **CWSP** présenté auparavant au point 1. D'autre part, si les entrées d'une ou plusieurs paires de transistors n'ont plus des valeurs égales (à cause d'une faute transitoire), la paire de transistors en série se comporte comme un seul transistor se trouvant à l'état coupé. Dans ce cas, la nouvelle porte se comporte comme la porte originale dans laquelle l'état de quelques transistors a été changé vers l'état coupé. Un tel changement soit n'affecte pas la valeur de la sortie (quand les chemins connectant la sortie de la porte aux sources d'alimentation  $V_{DD}$  ou GND, ne passent pas par les transistors changés vers l'état « off »), soit met la sortie de la porte en haute impédance. Dans les deux cas, le fonctionnement préserve l'état de la sortie de la porte avant que le défaut transitoire

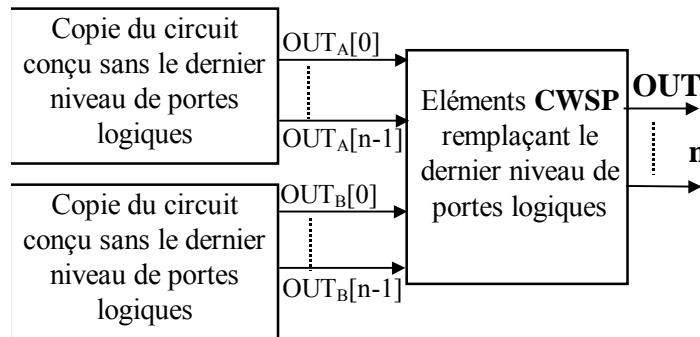
n'affecte une des entrées. Ce comportement correspond au comportement de l'élément **CWSP** spécifié au point 2.

Ainsi, n'importe quelle porte peut être transformée en un élément **CWSP** pour le code de duplication. Ceci a été obtenu en remplaçant chaque transistor d'une porte conventionnelle par deux transistors connectés en série, et contrôlés par des entrées dupliquées. La figure 4.6 présente les schémas au niveau transistors, pour l'inverseur, et pour les portes NOT-OU et NOT-ET. Ils réalisent les matrices de transitions d'états présentées dans le tableau 4.2.



**Figure 4.6- Portes CWSP utilisant les codes d'entrées de duplication et réalisant les fonctions d'inverseur, NOT-OU et NOT-ET.**

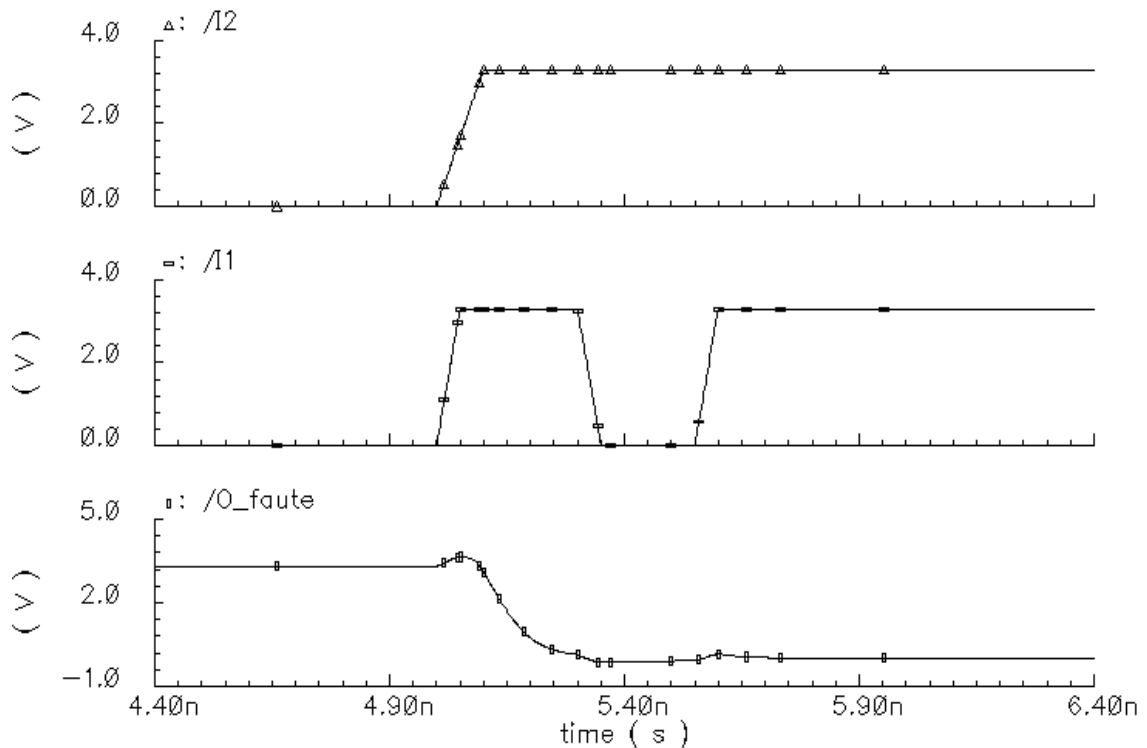
La figure 4.7 montre l'implémentation d'un circuit combinatoire tolérant les perturbations transitoires, basé sur la duplication.



**Figure 4.7- Circuit tolérant les perturbations basé sur la duplication**

Dans cette figure, le circuit combinatoire est dupliqué, à l'exception du dernier étage de portes logiques (les portes qui génèrent les sorties primaires du circuit combinatoire). Ces portes seront remplacées par les portes **CWSP** détaillées précédemment.

Déterminons maintenant les conditions temporelles pour lesquels ce schéma réalise la tolérance aux fautes transitoires. Soit  $D_{SS}$  la durée de temps dans un cycle d'horloge pour laquelle les sorties des deux copies du circuit sont stables (voir figure 4.8).



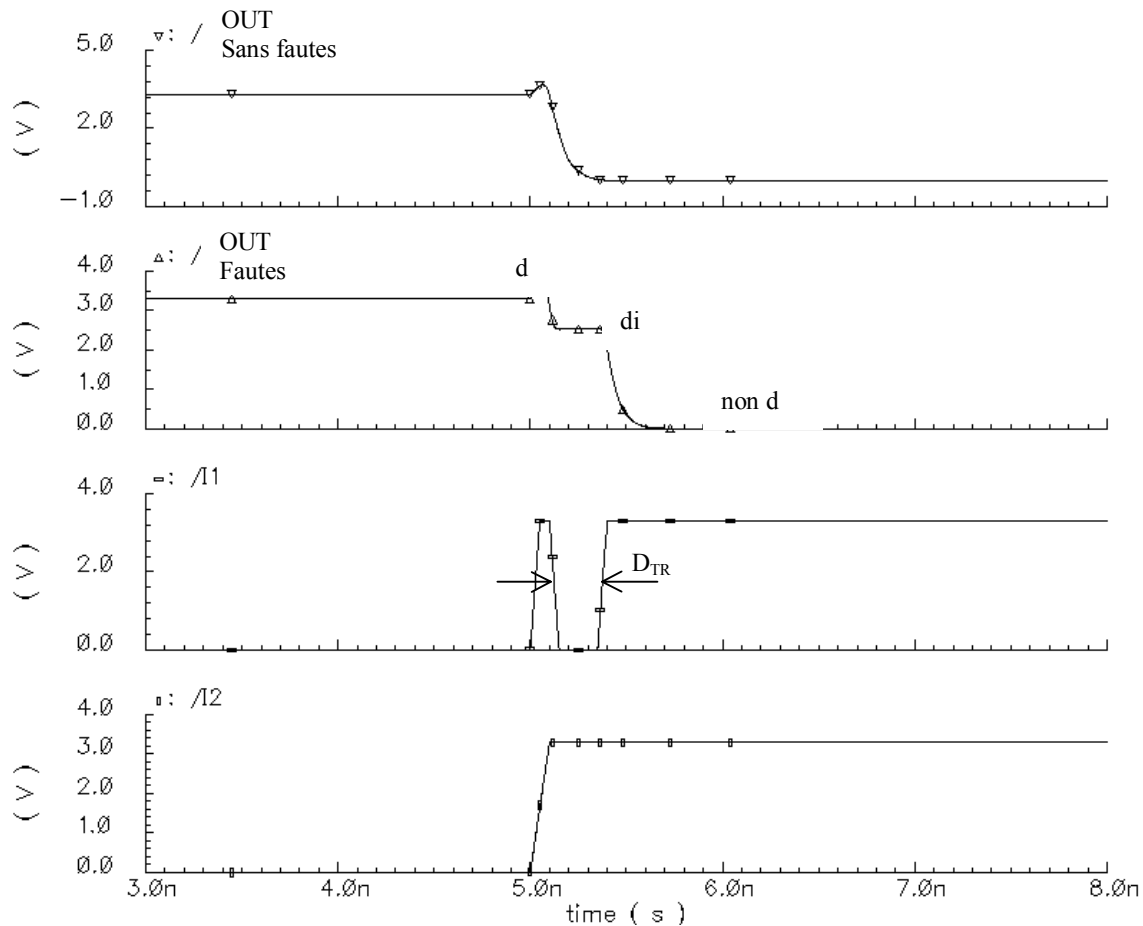
**Figure 4.8. Illustration du principe de fonctionnement d'un circuit CWSP inverseur.**

Soit  $D_{CWSP}$  le temps de transition logique d'une porte **CWSP** et  $D_{TR}$  la durée d'une impulsion transitoire affectant une des entrées de la porte **CWSP** (dans la figure 4.8, il est affecté par un défaut transitoire). Soit maintenant  $D_{CWSPb}$  la durée de stabilité des sorties avant que l'impulsion transitoire de durée  $D_{TR}$  apparaisse à la sortie si  $D_{CWSPb} \geq D_{CWSP}$ . Alors la sortie de l'élément de préservation de l'état (**CWSP**) sera la sortie correcte avant que l'impulsion transitoire survienne. Si l'impulsion transitoire transforme les entrées de la porte **CWSP** en mots hors code, l'élément préserve la valeur de sortie correcte déjà établie. Cette valeur sera alors capturée par les latches.

Soit  $D_{CWSPa}$  la durée de temps de stabilité des sorties après l'occurrence d'une faute transitoire de durée  $D_{TR}$ . Si  $D_{CWSPa} \geq D_{CWSP}$ , la sortie de la porte **CWSP** prend la valeur logique correcte après le passage de l'impulsion transitoire. Alors, au moment de l'arrivée du front montant de l'horloge, on retrouvera de nouveau la valeur logique correcte. La condition de stabilité  $D_{SS} \geq D_{TR} + 2D_{CWSP}$  garantit que, soit avant, soit après l'arrivée de l'impulsion transitoire, les sorties des deux circuits dupliqués seront stables pour une durée de temps au moins égale à  $D_{CWSP}$ . Ainsi n'importe quelle faute de durée plus petite que  $D_{TR}$  sera tolérée.

Par contre, cette condition est dérivée des conditions « pire cas ». Supposons maintenant que  $D_{TR} + D_{CWSP} < D_{SS} < D_{TR} + 2D_{CWSP}$ . Si l'impulsion transitoire de durée  $D_{TR}$  ne survient ni au début, ni à la fin de la période de stabilité dont la durée est  $D_{SS}$ , mais quelque part durant de cette période (voire figure 4.9, signal I1), on aura  $D_{CWSPa} < D_{CWSP}$  et  $D_{CWSPb} < D_{CWSP}$ . Ainsi, ni la période  $D_{CWSPa}$ , ni la période  $D_{CWSPb}$  ne sont suffisantes pour amener la sortie de la porte **CWSP** à sa valeur correcte. On risque donc de capturer une valeur erronée dans le latch. Néanmoins, selon la nature de la porte **CWSP**, il pourra être possible de réaliser la transition logique de la porte par le biais de deux périodes de stabilité, chacune ayant une durée inférieure à  $D_{CWSP}$ , mais les deux ensemble une somme de durées  $D_{CWSPa} + D_{CWSPb} > D_{CWSP}$ . Plus précisément, supposons que la sortie de la porte **CWSP** doit effectuer la transition **d** -> **non d**, mais que l'impulsion transitoire sur l'une des entrées vient perturber cette transition, on observera les phénomènes suivants :

Pendant le temps  $D_{CWSPb}$ , la sortie de la porte **CWSP** commence déjà sa transition et, au début de l'impulsion transitoire arrive à une valeur intermédiaire **di** (voir figure 4.9), car  $D_{CWSPb} < D_{CWSP}$ . A cause de la construction de la porte **CWSP**, cette valeur intermédiaire **di** sera présente seulement pendant la durée  $D_{TR}$  de la faute transitoire. A la fin du passage de l'impulsion transitoire, la sortie de la porte continue la transition de **di** à **non d**. Comme cette transition se fait entre **di** et **non d** et non pas entre **d** et **non d**, le temps nécessaire pour atteindre **non d** est plus court que  $D_{CWSP}$ , donc un temps  $D_{CWSPa}$  plus petit que  $D_{CWSP}$  pourrait être suffisant.



**Figure 4.9. Formes d'onde d'une porte CWSP inverseur. L'entrée I1 est affectée par un défaut transitoire de durée 200 ps.**

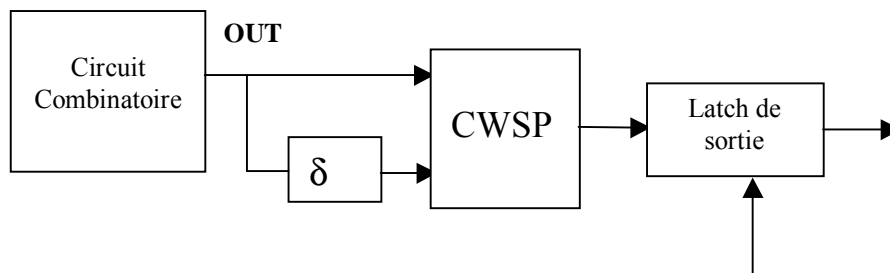
Nous trouvons donc la condition  $D_{CWSPa} + D_{CWSPb} < 2D_{CWSP}$ . La valeur exacte du  $D_{CWSPa} + D_{CWSPb}$  dépend de la construction de la porte **CWSP**. Dans le cas de l'implémentation au niveau transistors (figure 4.6), il est plus petit que pour une implémentation au niveau portes logiques (figure 4.5). Par des simulations électriques, on trouve que si la transition **d** -> **non d** se fait en deux étapes, par exemple **d** -> **di** -> **non d**, le temps nécessaire pour la transition est à peu près le même que si la transition se fait dans une seule étape. Alors, on obtient la condition  $D_{SS} > D_{TR} + D_{CWSP}$ . Si on prend en compte les temps de setup et de hold du bistable prévu à la sortie du circuit **CWSP**, le résultat final est  $D_{SS} > D_{TR} + D_{CWSP} + D_{SETUP} + D_{HOLD}$ . Pour l'implémentation d'un circuit sans tolérance aux défauts, on aura  $D'_{SS} > D_G + D_{SETUP} + D_{HOLD}$ , ou  $D_G$  est le retard de la porte conventionnelle que nous avons remplacée par la porte **CWSP**. On aura  $D_G < D_{CWSP}$ , à cause de la duplication des transistors dans la porte **CWSP**. Alors le délai supplémentaire

introduit par cette technique est  $D_{SS} - D_{SS}' = D_{TR} + D_{CWSP} - D_G$ , c'est à dire légèrement supérieur à  $D_{TR}$ . On réduit la perte de vitesse par rapport aux techniques de tolérance aux défauts transitoires (section 4.1) qui nécessitera un allongement de la période d'horloge supérieur à  $2D_{TR}$  au prix d'un surcoût significatif (duplication du circuit combinatoire).

Une manière d'implémenter cette technique consiste à utiliser des portes **CWSP** de type inverseur (figure 4.6). Dans ce cas, on implémente deux copies du circuit original qui génèrent des sorties inversées. Ensuite, ces sorties sont connectées aux entrées du circuit **CWSP** de type inverseur pour générer la sortie finale. Ce cas particulier sera utilisé dans les circuits où les chemins les plus longs sont activés très rarement. Dans ce cas, la fréquence d'horloge ne sera pas augmentée, car le circuit fonctionne correctement la plus grande partie du temps.

#### 4.2.2. Redondance temporelle basée sur les éléments de préservation d'état [ANG 00c]

Dans la suite, nous présentons une technique de tolérance aux fautes utilisant seulement la redondance temporelle et utilisant les éléments de préservation d'état. Le schéma d'implémentation est présenté dans la figure 4.10. On observe que, au lieu de dupliquer le circuit original, on duplique seulement le signal de sortie, et on connecte les deux signaux aux entrées d'un circuit **CWSP**. Une entrée de l'élément **CWSP** provient directement de la sortie du circuit combinatoire, tandis que la deuxième entrée est obtenue de cette même sortie retardée du laps de temps  $\delta$ .



**Figure 4.10** Technique de tolérance aux défauts basée sur la redondance temporelle et des circuits **CWSP**.

On observe un coût beaucoup moins élevé que dans le cas de la duplication avec éléments **CWSP**. En revanche, l'impact sur le cycle d'horloge sera plus élevé que dans la technique dupliquant le circuit combinatoire. La durée de  $\delta$  doit être au moins égale à  $D_{TR} - D_{CWSP}$ , pour éviter que les deux entrées de l'élément **CWSP** ne prennent des valeurs erronées pendant une période suffisante pour renverser la sortie de la porte **CWSP**. Pour le reste, l'analyse est similaire au cas présenté en figure 4.7. On obtient un allongement de la période d'horloge de  $2D_{TR} + D_{CWSP} - D_G$ . Cet allongement du temps est similaire à celui présenté dans la figure 4.13 et 4.14 (un peu inférieur, en réalité). Le coût matériel est inférieur à celui de la configuration de la figure 4.13, mais équivalent au coût matériel de celle de la figure 4.14. On pourrait se poser la question de l'intérêt de la solution présentée en figure 4.10 par rapport à celle de la figure 4.13. L'intérêt se présente dans le cas où on ne pourrait pas payer un allongement de la période de l'horloge aussi élevé que  $2D_{TR}$ . Sur le circuit de la figure 4.14., on peut diminuer cet allongement en réduisant le temps de transition logique  $D_{porte}$  de la porte de sortie en une valeur inférieure à  $2D_{TR}$ . Mais les impulsions d'une durée supérieure à  $D_{porte}/2$  (dont celles ayant une durée  $D_{TR}$ ), ne seront plus arrêtées. La situation est meilleure pour le cas de la figure 4.10. Supposons qu'on utilise la valeur de  $\delta$  spécifiée précédemment ( $\delta = D_{TR} - D_{CWSP}$ ), mais qu'on augmente la période de l'horloge d'une durée égale à  $D_{TR} - D_G$  seulement, au lieu de  $2D_{TR} + D_{CWSP} - D_G$ . Ce faisant, on ajoute à la période de l'horloge le retard supplémentaire de la figure 4.10 par rapport à une implémentation tolérant les défauts. Mais on ne rajoute pas l'allongement supplémentaire  $D_{TR} + D_{CWSP}$  qui sera nécessaire en cas de faute transitoire pour que les entrées de l'élément **CWSP** prennent leurs valeurs correctes pendant un temps au moins égal à  $D_{CWSP}$ . Dans cette situation, pour une sortie donnée **Si**, on ne peut pas garantir le blocage de l'impulsion transitoire pendant les cycles d'horloge qui calculent la valeur de **Si** en sensibilisant les chemins critiques du circuit. Mais pendant les autres cycles, la sortie **Si** sera prête plus tôt, augmentant sa durée de stabilité  $D_{SS}$  et augmentant également l'efficacité du mécanisme de tolérance aux défauts. Le même principe s'applique aussi dans le cas du circuit de la figure 4.4 pour réduire la perte en vitesse du circuit.



## Evaluations du coût matériel et de la dégradation de vitesse

Dans cette section nous évaluons l'impact des techniques de tolérance aux fautes en coût matériel et perte de vitesse. Pour cette évaluation, on a utilisé divers circuits arithmétiques (additionneurs et multiplicateurs). A cet effet, nous présentons dans le tableau 4.3. les résultats comparatifs de trois techniques : redondance massive TMR, duplication avec des éléments CWSP et redondance temporelle avec des éléments CWSP.

|                      | TMR         |           | Duplication + CWSP |           | Redondance temporelle (surface %) |                     | Redondance temporelle (perf. %) |                     |
|----------------------|-------------|-----------|--------------------|-----------|-----------------------------------|---------------------|---------------------------------|---------------------|
|                      | surface (%) | perf. (%) | surface (%)        | perf. (%) | $D_{TR}=0,45$<br>ns               | $D_{TR}=0,15$<br>ns | $D_{TR}=0,45$<br>ns             | $D_{TR}=0,15$<br>ns |
| <b>adders</b>        |             |           |                    |           |                                   |                     |                                 |                     |
| BK 8x8               | 194,35      | 18,33     | 90,88              | 15        | 23,71                             | 15,54               | 37                              | 17                  |
| CLA 8x8              | 194,31      | 17,74     | 90,81              | 14,51     | 23,89                             | 15,65               | 35,80                           | 16,45               |
| HC 8x8               | 194,48      | 19,64     | 91,09              | 16,07     | 23,17                             | 15,18               | 39,64                           | 18,2                |
| KS 8x8               | 195,09      | 25        | 92,07              | 20,45     | 20,61                             | 13,5                | 50,45                           | 23,18               |
| SKL 8x8              | 194,67      | 23,91     | 91,39              | 19,56     | 22,38                             | 14,67               | 48,26                           | 22,17               |
| BK 16x16             | 194,64      | 13,09     | 91,44              | 10,46     | 22,17                             | 14,53               | 25,81                           | 11,86               |
| CLA 16x16            | 194,43      | 15,27     | 91,11              | 12,5      | 23,04                             | 15,1                | 30,83                           | 14,16               |
| HC 16x16             | 195,03      | 15,27     | 92,07              | 12,93     | 20,55                             | 13,46               | 31,89                           | 14,65               |
| KS 16x16             | 195,78      | 22        | 93,27              | 17,30     | 17,43                             | 11,42               | 42,69                           | 19,61               |
| SKL 16x16            | 195,29      | 22        | 92,49              | 12,85     | 19,46                             | 12,75               | 31,71                           | 14,57               |
| BK 32x32             | 194,9       | 9,48      | 91,81              | 7,75      | 21,26                             | 13,93               | 19,14                           | 8,793               |
| CLA 32x32            | 194,67      | 10,57     | 91,43              | 8,65      | 22,24                             | 14,57               | 21,35                           | 9,808               |
| HC 32x32             | 195,55      | 14,10     | 92,84              | 11,53     | 18,57                             | 12,17               | 28,46                           | 13,08               |
| KS 32x32             | 196,37      | 17,74     | 94,16              | 14,5      | 15,16                             | 9,933               | 35,81                           | 16,45               |
| SKL 32x32            | 195,69      | 11,22     | 93,07              | 9,18      | 17,98                             | 11,78               | 22,65                           | 10,41               |
| <b>Multipliciers</b> |             |           |                    |           |                                   |                     |                                 |                     |
| Wallace 8            | 197,86      | 8,209     | 96,22              | 6,71      | 8,817                             | 5,777               | 16,57                           | 7,612               |
| Wall 16              | 198,71      | 4,7       | 97,93              | 3,84      | 5,377                             | 3,523               | 9,487                           | 4,359               |
| bo-wall 16           | 198,49      | 5,11      | 97,63              | 4,17      | 6,305                             | 4,131               | 10,31                           | 4,735               |
| Wall 32              | 199,43      | 3,22      | 99,06              | 2,63      | 2,449                             | 1,604               | 6,491                           | 2,982               |

Tableau 4.3. Calcul du surcoût en surface et dégradation des performances.

On remarque un surcoût matériel très élevé pour la technique TMR, près de 200%. Ce surcoût est dû à la triplification du circuit combinatoire et au coût du voter. Par contre, les latches de sortie ne sont pas triplés. On utilise en fait, un latch sur chaque sortie du voter. Ceci modère le surcoût matériel. Une observation s'impose : le latch n'est pas protégé ; pour être protégé, ce latch doit être durci. Dans ce cas, le coût matériel sera plus élevé. Habituellement, la surface d'un latch durci est deux fois plus grande que la surface d'un

latch normal. Dans le cas de la tolérance aux fautes obtenue par la combinaison de la duplication et l'élément **CWSP**, le surcoût d'implémentation est réduit presque de moitié, mais il reste toujours élevé : près de 100%. La technique de tolérance aux fautes implémentée à l'aide de la redondance temporelle (retard  $\delta$ ) et du circuit **CWSP** est la moins coûteuse en termes de surface supplémentaire. On a calculé cette surface pour deux durées d'impulsion transitoire,  $D_{TR} = 0,45$  ns et  $D_{TR} = 0,15$  ns. La deuxième durée d'impulsion transitoire est plus réaliste pour une impulsion transitoire créée par l'impact d'une particule ionisante. On observe un surcoût matériel très faible, sachant que le coût de la tolérance aux fautes habituelle dépasse 200 %. De plus, le surcoût matériel diminue si la taille du circuit augmente. Par exemple, pour un multiplicateur Wallace à 32 bits d'entrée, la surface supplémentaire est de 4,6 % seulement par rapport au circuit original ; pour une durée  $D_{TR}$  de 0,45 ns, elle sera réduite à 3,3% si l'impulsion transitoire est de 0,15 ns.

Les additionneurs présentent une surface supplémentaire plus élevée, car les circuits rajoutés (éléments de retard et élément **CWSP**) ont une surface qui devient significative en comparaison de la petite surface de l'additionneur. Dans tous les cas, le surcoût en surface reste toujours très bas par rapport aux 200 % du TMR. Il ne dépasse pas le 24 % pour l'implémentation considérant des impulsions transitoires de 450 ps, tandis que pour des impulsions de 150 ps, le surcoût matériel ne dépasse pas 16 %. En ce qui concerne les pertes en vitesse (colonne perf. du tableau 4.3.), la technique TMR donne les meilleurs résultats, alors que la technique de tolérance à base de redondance temporelle présente la plus grande dégradation de vitesse de fonctionnement. L'impact sur la vitesse du circuit est moins important quand la durée de l'impulsion transitoire est plus petite.

#### **4.2.4. Evaluation de l'efficacité de la tolérance aux fautes**

L'évaluation de l'efficacité des schémas de tolérance aux fautes a été vérifiée pour les circuits arithmétiques de différentes structures, additionneurs et multiplicateurs à 16 bits d'entrée. Les circuits ont été configurés d'après les schémas proposés dans les figures 4.7 et 4.10. Deux valeurs pour la durée de l'impulsion transitoire ont été considérées (0,45 ns et 0,15 ns). Plusieurs nœuds d'injection ont été choisis aléatoirement. L'évaluation de l'efficacité a été faite après un nombre important de simulations ( $2 \cdot 10^6$  simulations pour

chaque circuit), en utilisant notre propre simulateur présenté dans le chapitre 2. Etant donné que les portes **CWSP** ne font pas partie de la bibliothèque de cellules standard, pour les simulations nous avons construit le fichier *SDF* de la cellule **CWSP** inverseur que nous avons employé à l'aide des simulations SPICE. Cette construction a été facilitée par le fait que la cellule a toujours été utilisée avec le même fan-out. L'efficacité des schémas de tolérance aux fautes est définie comme le rapport entre le nombre d'erreurs corrigées avec le circuit **CWSP** et le nombre d'erreurs présentes aux sorties du circuit à cause de l'impulsion transitoire injectée. Les résultats d'évaluations sont présentés dans le tableau 4.4.

|               | <i>Duplication + CWSP</i>   | <i>Redondances temporelles</i> |                             |
|---------------|-----------------------------|--------------------------------|-----------------------------|
|               | Effic % ( $D_{TR}=0.45ns$ ) | Effic % ( $D_{TR}=0.45ns$ )    | Effic % ( $D_{TR}=0.15ns$ ) |
| BK16x16       | 100                         | 97.8                           | 98.9                        |
| CLA16x16      | 99.9                        | 98.4                           | 99.8                        |
| HC16x16       | 100                         | 99.1                           | 100                         |
| KS16x16       | 99.9                        | 96.9                           | 98.7                        |
| SKL16x16      | 99.9                        | 97.6                           | 99.2                        |
| Wallace       | 99.4                        | 95.8                           | 97                          |
| Booth-Wallace | 99.2                        | 94.6                           | 96.4                        |

**Tableau 4.4. Evaluation de l'efficacité des schémas présentés dans les figures 4.7 et 4.10.**

Dans le cas de l'implémentation de circuits suivant la redondance massive TMR, l'efficacité de la tolérance aux fautes est de 100 % si la faute transitoire affecte une seule copie du système. La duplication avec une élément **CWSP** donne une efficacité de correction proche de 100 %. Le cas de tolérance aux fautes obtenue par redondances temporelles combinées avec l'élément **CWSP** offre une efficacité de correction plus faible, mais qui reste cependant toujours élevée, pour un coût d'implémentation très faible (tableau 4.3).

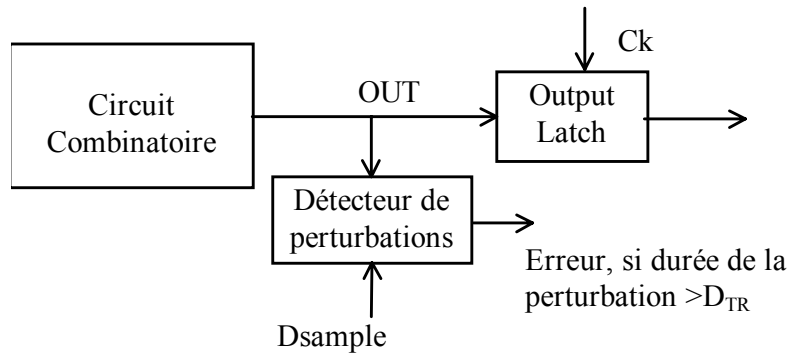
La possibilité d'erreurs échappant à la correction est due aux phénomènes d'amplification de l'impulsion transitoire à travers des chemins de propagation reconvergeants impliquant des retards différents. L'impulsion résultante à la sortie du circuit peut ainsi devenir plus large que l'impulsion originale et peut ne pas être corrigée par le circuit **CWSP**.

### ***4.3. Techniques de détection de fautes transitoires et de fautes temporelles à l'aide de redondances temporelles***

Les techniques de tolérance aux fautes temporelles présentées dans les sections précédentes permettent une réduction importante du coût matériel, par comparaison avec la technique traditionnelle de triplication. Néanmoins, il reste à payer une perte en vitesse non négligeable. Cet inconvénient deviendra de plus en plus gênant avec l'augmentation de la vitesse dans les circuits nanométriques. Une réduction de cette perte de vitesse pourrait être obtenue par la technique de redondance matérielle-temporelle, mais le prix à payer est la duplication du circuit. De plus, ces techniques prennent en compte les fautes transitoires, mais pas les fautes de timing. Pour contourner ces problèmes, on peut adopter l'approche détection d'erreur avec reprise. Alors, après chaque détection d'erreurs qui ont pour source les fautes transitoires, les opérations les plus récentes du circuit seront répétées afin de corriger le résultat. Si l'erreur détectée provient d'une faute de timing, les opérations seront répétées avec la fréquence d'horloge réduite. Pour effectuer la détection d'erreur à faible coût matériel, on va appliquer le principe de la redondance temporelle. On verra que cette technique permet la détection d'erreurs, produites par les fautes transitoires et les fautes de timing, avec un faible coût et ne réduit pas la vitesse de fonctionnement.

#### **4.3.1. Principe de fonctionnement**

Pour réaliser la détection des fautes transitoires par des moyens de type redondance temporelle, on propose le principe présenté en figure 4.11. On voit que le circuit combinatoire est contrôlé par un bloc capable de détecter l'apparition d'une perturbation transitoire sur les sorties du circuit.

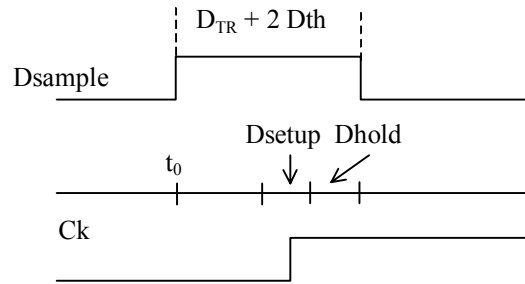


**Figure 4.11. Décteur de perturbations basé sur la redondance temporelle.**

Dans ce cas on s'intéresse seulement à la détection des fautes transitoires. Ainsi, il n'est pas nécessaire d'assurer que la valeur correcte est présente sur la sortie du circuit combinatoire (OUT) pendant la plus grande partie du temps. Dans le cas présent, une fois les résultats prêts, à l'instant  $t_0$ , la valeur de sortie doit être maintenue stable pendant au moins  $D_{TR} + 2D_{th}$  ( $D_{TR}$  est la durée de l'impulsion transitoire, et  $D_{th}$  est la durée minimale de la perturbation détectable par le détecteur, durée de seuil de détection).

La phase de stabilité est la phase durant laquelle le détecteur de perturbation pourra être activé, faute de quoi on aura une fausse détection d'erreur. Une durée de stabilité égale à  $D_{TR} + 2D_{th}$  est nécessaire pour assurer que l'impulsion transitoire ne couvrira pas toute la durée de stabilité, créant ainsi un signal erroné mais stable, qui ne sera pas détecté par le circuit de détection. En ajoutant  $2D_{th}$  à  $D_{TR}$  on assure que pour toute position d'une impulsion transitoire de durée  $D_{TR}$  il restera, soit à gauche, soit à droite de cette impulsion, pour une période au moins égale à  $D_{th}$ , pendant laquelle le signal prendra une valeur différente de la valeur imposée par la perturbation. Cette différence de valeurs est alors détectée car sa durée dépasse le seuil de détection  $D_{th}$ .

Le signal  $D_{sample}$  déclenche la phase d'activation du détecteur, qui commence au moment où le signal OUT est garanti d'être stable dans le pire cas (propagation par les chemins critiques du circuit combinatoire), et qui finit à l'instant  $t_0 + D_{TR} + 2D_{th}$ . Le signal  $Ck$  du latch sera activé à  $t_0 + D_{TR} - 2D_{th} - D_{hold}$ , c'est-à-dire à un moment égal à  $D_{hold}$  plutôt qu'à celui de la fin de phase d'activation du détecteur de perturbations. La figure qui suit (figure 4.12.) montre les relations entre les signaux  $D_{sample}$  et  $Ck$ .



**Figure 4.12. Relation entre les signaux  $D_{\text{sample}}$  et le signal d'horloge.**

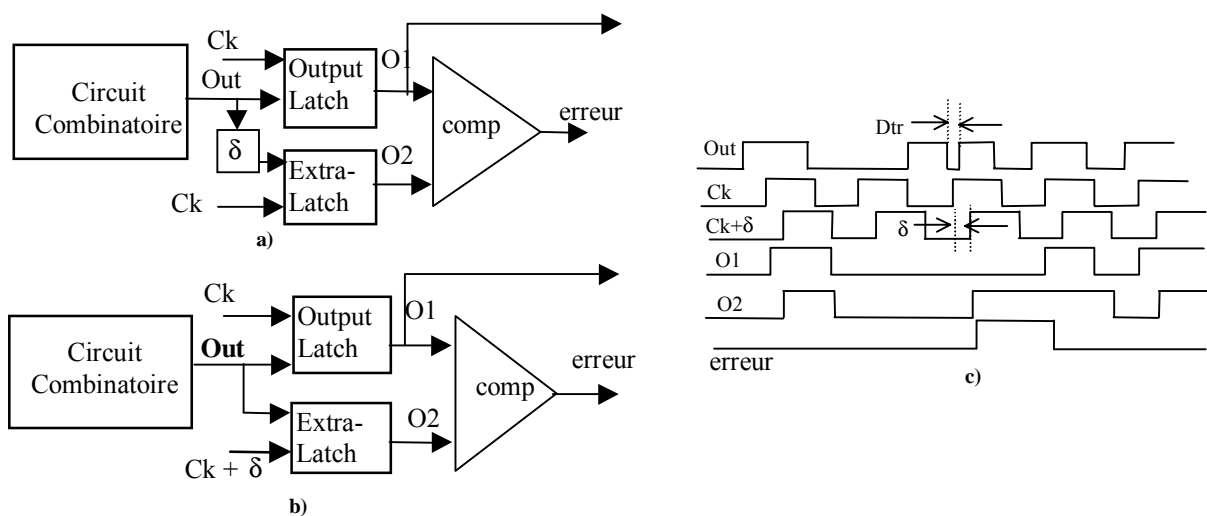
On observe que la phase de setup et de hold du circuit sont dans la phase de stabilité, mais la durée de cette phase n'est pas strictement nécessaire, car elle vaut  $D_{\text{TR}} + 2D_{\text{th}}$ . Ainsi, la période d'horloge est allongée d'une durée égale à  $D_{\text{TR}} + 2D_{\text{th}} - D_{\text{setup}} - D_{\text{hold}}$ . Cet allongement du cycle d'horloge est bien inférieur à l'allongement imposé par la technique de tolérance aux défauts (égal à  $2 D_{\text{TR}} + \varepsilon + D_{\text{voter}}$ ). Notons aussi que cet allongement pourrait être évité en décalant la phase de «sample» ( $D_{\text{sample}} = 1$ ) vers la gauche par rapport au front montant de l'horloge. Cette technique nécessitera des contraintes supplémentaires sur les retards du circuit combinatoire, que nous allons détailler plus loin.

#### **4.3.2. Implémentation à l'aide des portes standard et protection des latches [NIC '99]**

Une implémentation possible de la technique de détection des fautes transitoires décrite auparavant est présentée dans la figure 4.13.a. Le circuit combinatoire est connecté à un circuit supplémentaire capable de détecter les erreurs produites par une faute transitoire affectant un nœud du circuit ou par une faute de timing. Le schéma est composé de deux latches pour chaque sortie du circuit et un comparateur. L'entrée d'un de ces latches provient de la sortie du circuit combinatoire, tandis que l'entrée du deuxième latch provient de la même sortie, mais retardée de  $\delta$ . Ce  $\delta$  est déterminé à partir de la durée maximale de la faute transitoire qu'on se propose de détecter ( $D_{\text{TR}}$ ), et par le temps de setup du latch considéré lors de la conception ( $D_{\text{setup}}$ ). Le comparateur détecte toute inégalité

entre les sorties des deux latches. L'horloge Ck du système ainsi construite sera retardée de ce  $\delta$ , par rapport au système original, exclus les éléments de détection.

Une deuxième version du même principe est présentée dans la figure 4.13.b. L'horloge du système Ck capture la sortie du circuit combinatoire à l'instant  $t_0$  dans le latch noté par Output Latch dans la figure. Une deuxième horloge, dérivée de la première, capture la même sortie du circuit combinatoire à l'instant  $t_0 + \delta$  dans le deuxième latch (noté Extra-Latch). Le concepteur détermine la durée maximale de l'impulsion transitoire qui sera tolérée et il calcule  $\delta = D_{tr} + D_{setup} + D_{hold}$ .



**Figure 4.13. a), b) – Schéma de détection de défauts transitoires utilisant un latch supplémentaire et un comparateur ; 1c) Schéma de détection des fautes transitoires correspondant à l'implémentation b).**

En analysant les schémas présentés dans la figure 4.13, on observe que les impulsions transitoires ayant une durée ne dépassant pas  $D_{TR}$  ne peuvent pas être capturées par les deux latches (Output-Latch et Extra-Latch) simultanément. Le comparateur détecte l'occurrence de toute erreur due à une impulsion transitoire dont la durée ne dépasse pas  $D_{TR}$ . Notons aussi une différence importante avec le schéma générique de la figure 4.11. Dans la figure 4.11, le détecteur est placé en amont du latch de sortie ; ainsi, il ne détecte pas les erreurs produites par les particules touchant les nœuds du latch. En revanche, dans la figure 4.13, le détecteur (comparateur) est placé en aval du latch de sortie ; ainsi, il détecte les erreurs produites par les particules frappant les latches.

## Dégradation des performances

Il est clair que la solution de la figure 4.13.a allonge la période d'horloge d'une durée égale à  $\delta$ . Concernant la solution de la figure 4.13.b, on observe que l'introduction de la deuxième horloge retardée peut introduire du retard supplémentaire dans le fonctionnement du circuit. Néanmoins, une analyse temporelle attentive peut nous permettre d'éviter que cela se produise.

La sortie du circuit combinatoire est mémorisée dans le Output Latch à l'instant  $t_0$ , c'est à dire dès que les sorties du circuit sont supposées être prêtes. Ce résultat peut être utilisé par la suite en entrée d'un autre circuit combinatoire sans aucune dégradation de vitesse de fonctionnement. Malheureusement, on ne peut pas encore envoyer de nouvelles entrées au circuit, parce que ces nouvelles valeurs d'entrée peuvent affecter le résultat qui sera capturé par le latch supplémentaire (Extra-Latch). Néanmoins, cet inconvénient peut être éliminé en introduisant des contraintes sur les chemins les plus courts lors de la conception du circuit combinatoire. Pour expliquer plus en détail comment fonctionne cette technique, nous considérons le cas d'un circuit combinatoire à plusieurs entrées et sorties, et nous nous intéressons au retard de propagation minimal dans le circuit.

On peut avoir les situations suivantes:

- Le temps de propagation  $D_{min}$  du plus petit chemin entre n'importe quelle entrée et n'importe quelle sortie du circuit combinatoire respecte la relation  $D_{min} \geq \delta$ . Dans ce cas, les sorties du circuit ne sont pas affectées par les nouvelles entrées avant l'instant  $t_0 + D_{min}$ . Ainsi, le circuit présenté dans la figure 4.13.b peut fonctionner à la même fréquence d'horloge que le circuit original, sans que les valeurs capturées par le latch supplémentaire soient affectées.
- Le temps de propagation minimal du circuit ne respecte pas la condition  $D_{min} \geq \delta$ . Dans ce cas, afin de pouvoir fonctionner à la vitesse d'horloge maximale, on impose la condition  $D_{min} > \delta$  en ajoutant des éléments de retard dans les chemins de propagation dont le retard est plus court que  $D_{min}$ . Les éléments de retard peuvent être des inverseurs ou des buffers ou bien encore certaines portes existantes peuvent être



remplacées par des portes similaires ayant des temps de propagation supérieurs, pour que la condition  $D_{\min} \geq \delta$  soit remplie.

Ainsi, on peut toujours assurer que la vitesse de fonctionnement du circuit présenté dans la figure 4.13.b ne sera pas affectée par l'introduction de la technique de détection de fautes transitoires.

### **Coût matériel**

De point de vue coût matériel on observe une augmentation modérée, correspondant à quelques éléments de retards introduits sur les chemins de propagation ne remplissant pas la condition  $D_{\min} \geq \delta$ , un latch pour chaque sortie (le latch supplémentaire Extra-Latch), et un comparateur recevant en entrée les sorties de tous les paires de Output Latch et Extra Latch et générant un signal de sortie pour signaler l'occurrence d'une erreur dans un ou plusieurs paires de latches.

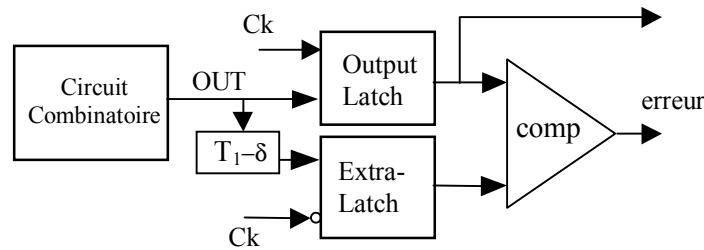
### **Couverture des fautes**

Nous avons vu qu'en fixant  $\delta = D_{TR} + D_{\text{setup}} + D_{\text{hold}}$ , le schéma de la figure 4.13. peut détecter toute impulsion transitoire d'une durée plus petite ou égale à  $D_{TR}$ , et affectant une ou plusieurs sorties du circuit combinatoire. On peut observer maintenant que ce schéma peut aussi détecter les erreurs de timing. En effet, une faute introduisant un retard inférieur à  $\delta$  ne pourra pas affecter l'Extra Latch. Ainsi, la comparaison des sorties de l'Output Latch détectera les erreurs. La valeur de  $\delta$  doit être fixée pour prendre en compte les fautes de timing et les fautes transitoires. Si on note  $D_{\text{tim}}$  le retard maximum de la faute de timing que nous cherchons à détecter,  $\delta$  sera choisi comme le maximum des deux valeurs  $D_{\text{tim}}$  d'une part, et  $D_{TR} + D_{\text{setup}} + D_{\text{hold}}$  d'autre part.  $D_{\text{tim}}$  étant habituellement bien plus grand que  $D_{TR}$ ,  $\delta$  prendra la valeur de  $D_{\text{tim}}$ . Notons finalement la détection de toute faute (permanent, transitoire ou de timing) affectant l'Output Latch ou l'Extra Latch.

## L'implémentation du circuit de distribution de l'horloge

Certaines difficultés peuvent être rencontrées lors de la conception du réseau de distribution d'horloge. En effet, lors de la synthèse logique, et plus tard, lors de la phase de placement-routage, l'imprécision du signal d'horloge (« clock skew ») doit être réduite au minimum. Cependant, garder ce paramètre à des valeurs acceptables s'avère parfois très difficile à cause de la limitation des outils de conception des circuits intégrés, et de la complexité du circuit. De plus, la valeur calculée  $\delta$  est proche de la valeur de l'imprécision d'horloge garantie par les outils CAO. Une mauvaise prise en compte de ce problème peut compromettre l'efficacité de détection du schéma. Des solutions d'implémentation du réseau d'horloge existent. Par exemple :

- L'horloge retardée,  $Ck+\delta$  (utilisée dans la figure 4.13.b) peut être obtenue à partir du signal d'horloge  $Ck$  en utilisant des éléments retardateurs. Les éléments permettant ce retard sont conçus de façon très précise, et le routage du signal d'horloge permettra de garder l'imprécision à son minimum.
- Une autre possibilité consiste en l'utilisation des mêmes  $\delta$  pour tous les blocs d'un circuit intégré et l'implémentation d'un deuxième réseau de distribution du signal d'horloge pour  $Ck+\delta$ . Cette solution est beaucoup plus souple, car la valeur de  $\delta$  est contrôlée de l'extérieur, mais elle nécessite d'augmenter la surface pour le deuxième arbre d'horloge. Il sera aussi difficile de minimiser les 'skews' entre les deux arbres.
- Une solution très simple consiste en l'utilisation du front montant de l'horloge  $Ck$  pour capturer les valeurs de sorties dans le Output Latch, et le front descendant de l'horloge pour le latch supplémentaire, Extra-Latch. Cette solution nécessite la modification du cycle d'horloge ; ainsi le niveau logique '1' de l'horloge doit avoir une durée égale à  $T_1=\delta$ . Si d'autres contraintes imposent  $T_1>\delta$ , il est possible de laisser inchangé le cycle d'horloge, mais un bloc de retardement supplémentaire, de valeur  $T_1-\delta$ , sera introduit à l'entrée du latch supplémentaire Extra-Latch (voire figure 4.14).



**Figure 4.14. Réseau d'horloge simplifié.**

Alors, si le front montant de l'horloge correspondant à l'Extra-Latch survient à  $t_0 + T_1$ , la valeur gardée dans ce latch est en effet la valeur présente à la sortie du circuit à l'instant  $t_0 + \delta$ . Ainsi, on peut utiliser un seul réseau de distribution d'horloge.

### 4.3.3. Evaluation du coût matériel [ANG 00a]

Pour l'évaluation du surcoût d'implémentation, on a considéré seulement la technique présentée dans la figure 4.13.b, car la vitesse de fonctionnement du circuit n'est pas affectée. Pour le circuit combinatoire, on va imposer des contraintes sur le temps de propagation minimal. On doit en effet imposer la condition  $D_{min} > \delta$ . Cette contrainte sera prise en compte lors de la synthèse logique, pendant la phase d'optimisation de surface et du temps de propagation. Lors de la phase d'optimisation le circuit a été considéré fonctionnant dans le meilleur des cas ou « best case », parce que ces conditions de fonctionnement correspondent aux temps de propagation les plus courts. Ainsi la contrainte  $D_{min} > \delta$  sera valable même pour les lots présentant une vitesse plus élevée que la vitesse nominale.

La technologie que nous avons à notre disposition pour l'implémentation des ces techniques est la technologie AMS 0,35 $\mu$ m (3,3V).

Le tableau 4.5 montre les pourcentages d'augmentation de surface pour plusieurs cas de multiplieurs de différentes tailles et structures. Les calculs de l'augmentation de la surface ont été faits pour des  $\delta$  égaux à 0,46ns, 0,6ns, 0,8ns et 1ns, respectivement. On a considéré également le cas de détection d'erreurs de timing pour lesquelles  $\delta$  a une valeur égale au 20% de la durée maximale de propagation du circuit considéré.

|                      | CP<br>(ns) | PAO(%) / TAO(%)       |                      |                      |                    |             |                         |
|----------------------|------------|-----------------------|----------------------|----------------------|--------------------|-------------|-------------------------|
|                      |            | $\delta=0.46$<br>(ns) | $\delta=0.6$<br>(ns) | $\delta=0.8$<br>(ns) | $\delta=1$<br>(ns) | 20%<br>Dmax | 20%<br>Dmax<br>amélioré |
| <b>Brown 16</b>      | 32         | 0.2/4.3               | 1.2/5.3              | 3.2/7.3              | 4.5/8.5            | 44.2/46.8   | 16.3/19.5               |
| <b>Brown 32</b>      | 66         | 0.4/2.4               | 0.5/2.5              | 1.5/3.5              | 2.6/4.6            | 18.2/19.8   | 6.9/8.68                |
| <b>Wallace 8</b>     | 7          | 0.9/7.5               | 3.1/9.5              | 3.4/9.9              | 3.5/10             | 10.8/17.3   | 5.7/11.4                |
| <b>Wallace 16</b>    | 11         | 0.1/3.5               | 0.2/3.6              | 0.4/3.8              | 0.5/4              | 15.3/18.3   | 6.2/9.2                 |
| <b>Wallace 32</b>    | 18         | 0.05/1.3              | 0.09/1.4             | 0.15/1.5             | 0.2/1.5            | 3.5/4.8     | 0.06/1.3                |
| <b>Booth-wall 16</b> | 11         | 0.03/4                | 0.7/4.7              | 1.1/5.2              | 1.2/5.3            | 17.9/21.3   | 4.36/8                  |
| <b>Booth-wall 32</b> | 16         | 0.09/2.3              | 0.12/2.3             | 0.16/2.3             | 0.2/2.4            | 3.2/5.4     | 0.1/2.26                |
| <b>Booth 8</b>       | 8          | 0.3/6.5               | 1/7.2                | 1.2/7.5              | 2.4/8.8            | 19.7/25.3   | 4.2/9.6                 |
| <b>Booth 16</b>      | 14         | 0.2/3.9               | 0.3/4                | 0.4/4.2              | 0.9/4.8            | 39.5/42     | 12.6/15.1               |
| <b>Booth 32</b>      | 20         | 0.1/2.6               | 0.16/2.7             | 0.2/2.78             | 0.3/2.9            | 3/5.62      | 0.09/2.6                |

**Tableau 4.5. Surcoût matériel pour des multiplicateurs de tailles et structures différentes.**

Dans la première colonne du tableau 4.5, est présenté le délai du chemin critique pour chaque circuit considéré. Toutes les autres colonnes montrent le surcoût matériel obtenu lors de l'introduction de la méthode de détection. La première composante de ce surcoût matériel (PAO) correspond au coût nécessaire à l'implémentation de la technique de la contrainte  $D_{min} > \delta$ . On remarque que ce coût est réduit, surtout quand on considère des multiplicateurs d'intérêt pratique, comme par exemple des multiplicateurs Wallace et Booth-Wallace. La deuxième composante du surcoût matériel (TAO) représente la surface totale rajoutée par rapport à la configuration initiale. TAO correspond au surcoût matériel pour l'implémentation de la contrainte  $D_{min} > \delta$  et au surcoût matériel correspondant au rajout des latches supplémentaires Extra-Latches et du comparateur. Pour des valeurs de  $\delta$  plus petites que 1 ns (valeur qui est largement supérieure à une durée typique d'impulsion transitoire générée par l'impact d'une particule chargée), le surcoût en surface reste toujours faible. Ces valeurs deviennent insignifiantes si la taille du multiplicateur augmente. Les résultats correspondent à l'implémentation des circuits dans la technologie AMS 0,35  $\mu\text{m}$ , et prennent en compte la surface et les temps de propagation des interconnexions estimés par l'outil de synthèse. Pour le cas de détection d'erreurs de timing ( $\delta = 20\%$  de Dmax), le surcoût de nouveau est faible, à l'exception des multiplicateurs Brown, qui ne sont pas d'intérêt pratique, vu leurs retards de propagation excessifs. Ces

circuits n'étant pas optimisés pour la vitesse de fonctionnement, ils présentent un très grande dispersion des valeurs de retard des différents chemins de propagation. Ce déséquilibre a un effet néfaste sur le coût de la contrainte  $D_{min} > \delta$ , qui diminue quand les retards d'un circuit sont équilibrés. Nous verrons dans la suite des méthodes permettant la réduction de ce coût dans les circuits non équilibrés pour obtenir les résultats présentés dans la dernière colonne du tableau 4.5.

Une analyse plus détaillée de la technique nous permet d'affiner notre compréhension sur les diverses causes de l'augmentation de surface.

- Une première partie de la surface ajoutée est due au latch supplémentaire introduit sur chaque sortie du circuit combinatoire ainsi qu'au comparateur. Le coût de ce dernier est une porte XOR et une porte OR à chaque sortie du circuit. Donc, on observe un coût fixe pour chaque sortie du circuit. Donc, le pourcentage d'augmentation de la surface est déterminée par le rapport entre la surface du circuit original et le nombre de ces sorties que nous appellerons *APO* (Area Per Output). Ainsi, nous avons  $APO = (\text{surface du circuit})/(\text{nombre de sorties})$ . Une augmentation de la surface relativement petite correspond à une valeur élevée du rapport *APO*.
- Une deuxième partie de la surface ajoutée apparaît pendant la phase de synthèse logique, due à la contrainte temporelle relative au temps de propagation minimal  $D_{min} > \delta$ . Cette augmentation de surface est notée PAO dans le tableau 4.5. Pour ces circuits dans lesquels les temps de propagation entre les entrées et les sorties ne présentent pas une grande dispersion des retards, ce deuxième surcoût en surface restera faible. Pour les circuits ayant un grand nombre de chemins de propagation à retards plus courts, le surcoût en surface augmente considérablement.

En minimisant les retards, les outils de synthèse modernes réduisent la durée des chemins critiques et ont tendance à générer des circuits équilibrés ayant des dispersions de retards réduits et résultant en un faible coût d'implémentation de la contrainte  $D_{min} > \delta$ . On les appelle circuits à temps de propagation équilibrés. Cependant il reste toujours des circuits qui n'entrent pas dans cette catégorie.

Les multiplicateurs présentent un surcoût réduit en surface car ils ont un rapport *APO* élevé, et ils sont généralement équilibrés du point de vue du temps de propagation (surtout les multiplicateurs rapides Wallace et Booth-Wallace).

Pour les circuits qui ne se prêtent pas facilement à cette technique, nous avons choisi diverses structures d'additionneurs. Les structures utilisées pour le calcul de surcoût matériel incluent des additionneurs carry-lookahead [HWA 79][WES 94], Brent & Kung [BRE 82], Han & Carlson [HAN 87], Kogge & Stone [KOG 73] et Sklanski [SKL 60]. Ces additionneurs sont les plus rapides et sont optimisés en surface et en fan-out. En fait, les additionneurs ont un rapport *APO* faible. De plus, leur temps de propagation sont très dispersés, car ils ont un nombre important de chemins à retards faibles. Par exemple, le temps de propagation entre n'importe quelle entrée  $A_i$  ou  $B_i$  vers la sortie correspondante  $S_i$ , est celui de 2 portes XOR seulement, pour tout  $i$ .

Le tableau 4.6 représente les pourcentages d'augmentation du coût en surface pour divers additionneurs.

|                             | CP (ns) | TAO(%)        |              |              |            |         |
|-----------------------------|---------|---------------|--------------|--------------|------------|---------|
|                             |         | $\delta=0.46$ | $\delta=0.6$ | $\delta=0.8$ | $\delta=1$ | 20%Dmax |
| <b>Brent &amp; Kung 8</b>   | 3       | 39.25         | 48.54        | 59.54        | 62.42      | 48.54   |
| <b>Brent &amp; Kung 16</b>  | 4.2     | 34.99         | 45.46        | 52.43        | 58.10      | 53.4    |
| <b>Brent &amp; Kung 32</b>  | 5.1     | 32.55         | 42.54        | 50.03        | 55.05      | 55.05   |
| <b>Han &amp; Carlson 8</b>  | 2.8     | 37.53         | 47.3         | 53.6         | 70.2       | 47.3    |
| <b>Han &amp; Carlson 16</b> | 3.6     | 31.71         | 40.60        | 48.92        | 51.68      | 45.8    |
| <b>Han &amp; Carlson 32</b> | 4.2     | 27.69         | 36.8         | 41.33        | 48.39      | 42.1    |
| <b>Kogge &amp; Stone 8</b>  | 2.2     | 31.72         | 33.75        | 38.47        | 46.84      | 31.72   |
| <b>Kogge &amp; Stone 16</b> | 2.5     | 26.6          | 27.4         | 31.21        | 37.52      | 27.0    |
| <b>Kogge &amp; Stone 32</b> | 3.1     | 22.06         | 22.82        | 26.53        | 31.14      | 22.82   |
| <b>Sklanski 8</b>           | 2.3     | 34.97         | 38.0         | 44.66        | 54.39      | 34.97   |
| <b>Sklanski 16</b>          | 2.5     | 29.35         | 32.5         | 37.66        | 44.1       | 30.6    |
| <b>Sklanski 32</b>          | 4.9     | 26.7          | 29.29        | 33.48        | 38.65      | 37.1    |
| <b>CLA 8</b>                | 3.1     | 37.62         | 43.21        | 53.66        | 70.21      | 43.21   |
| <b>CLA 16</b>               | 3.6     | 37.74         | 44.1         | 50.10        | 67.49      | 50.10   |
| <b>CLA 32</b>               | 5.2     | 35.97         | 42.65        | 46.98        | 65.48      | 68.8    |

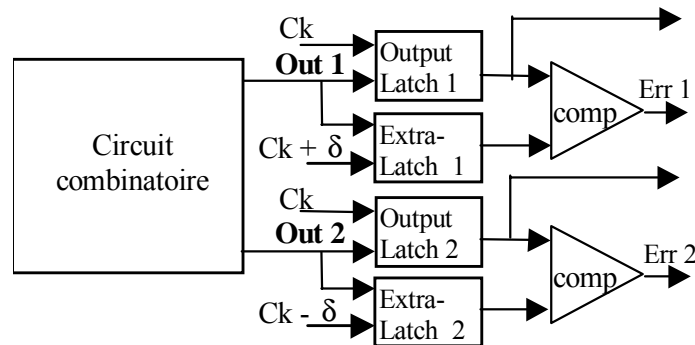
**Table 4.6. Surcoût en surface pour différentes structures et tailles d'additionneurs.**

On observe que le surcoût en surface est beaucoup plus grand que pour des multiplicateurs, ce qui réduit l'intérêt d'utiliser cette méthode pour les additionneurs. Dans la section suivante on propose quelques méthodes de réduction du coût matériel.

#### 4.3.4. Méthodes de réduction du coût matériel.

##### Implémentation

1) La technique développée antérieurement détecte les fautes transitoires en comparant la valeur présente à la sortie du circuit à deux instants différents, la différence entre les deux instants étant égale à  $\delta$ . Cependant, la même capacité de détection est acquise si la deuxième instance de temps est fixée à  $t_0 - \delta$  au lieu de  $t_0 + \delta$ . On peut alors modifier le schéma originel de la figure 4.15. Dans cette figure on observe qu'une partie des sorties est organisée d'après le principe présenté dans la figure 4.13.b (avec les horloges  $Ck$  and  $Ck+\delta$ ), tandis que l'autre partie des sorties utilise le principe modifié (avec les horloges  $Ck$  and  $Ck-\delta$ ).



**Figure 4.15. Méthode de réduction du coût matériel.**

Le principe modifié est essentiellement utilisé pour les sorties qui sont stables suffisamment à l'avance (les sorties qui ont un retard de propagation maximal égal à  $D_{\max} - \delta$ ).

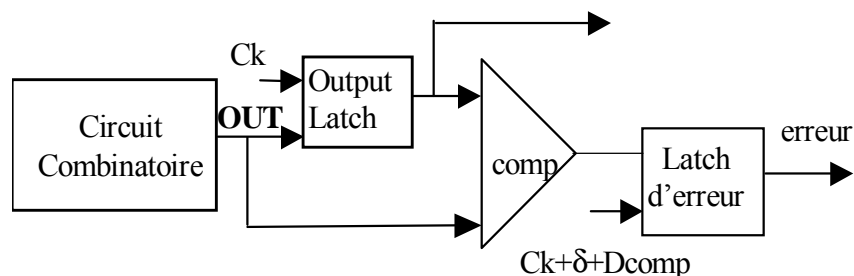
Ce principe est très intéressant car il nous permet d'éliminer le surcoût correspondant à la contrainte  $D_{\min} > \delta$ , pour les sorties du deuxième groupe, Out2. Etant donné que ce groupe contient les sorties dont les retards ne dépassent pas  $D_{\max} - \delta$  (Out2), il peut être utilisé pour la réduction du surcoût matériel. Ces sorties sont susceptibles de contenir le plus grand nombre de chemins à retard de propagation faible. Ainsi, l'élimination de la contrainte  $D_{\min} > \delta$ , pour ce groupe de sortie, devra amener une réduction appréciable du surcoût matériel.

L'efficacité de détection reste inchangée. En effet, puisque les instances de capture de la sortie Out2 par les deux latches diffèrent d'une valeur  $\delta$ , la technique détecte toutes les défauts transitoires ayant une durée  $D_{TR} < \delta - D_{setup} - D_{hold}$ .

En ce qui concerne la détection d'erreurs de timing ayant des retards plus courts ou égaux au  $\delta$ , elles seront détectées si elles affectent les sorties Out1. Pour celles qui affectent les sorties Out2, étant donné que le temps maximal de propagation des chemins reliant les entrées aux sorties Out 2 est plus petit que  $D_{max} - \delta$ , en ajoutant sur ces chemins un retard  $\delta$  égal à la valeur de l'erreur de timing, on obtient des sorties retardées d'un temps de propagation plus court que  $D_{max}$ . Alors, les fautes de timing de retard plus court ou égal à  $\delta$  ne peuvent pas affecter la valeur capturée par les latches Output-Latch 2.

Considérons maintenant l'implémentation des signaux d'horloge car nous avons intérêt à réduire le nombre de signaux d'horloge. Tout d'abord, pour les latches Extra Latch 1, on peut utiliser le signal d'horloge Ck après avoir introduit un retard à l'entrée de ce latch, et après avoir inversé le signal Ck comme nous l'avons décrit dans la figure 4.14. On peut ainsi utiliser le signal normal de l'horloge Ck, pour capturer les entrées des latches Extra-Latch 2 au même instant que pour les latches Output-Latch (1,2), à la condition qu'un bloc de retard de durée  $\delta$  soit ajouté à l'entrée de latch Extra Latch 2. Dans ce cas, la valeur actuelle capturée dans les latches Extra-Latch 2 est, en fait, la valeur présente à la sortie Out2 à l'instant  $t_0 - \delta$ .

2) Un deuxième principe de réduction du coût matériel est présenté dans la figure 4.16. Ce principe est utilisé pour les circuits ayant une *APO* réduite, parce que, dans ce cas, le latch Extra-Latch représente un surcoût très important.



**Figure 4.14. Méthode de réduction de surcoût matériel avec élimination de l'Extra-Latch**



Le principe de fonctionnement de la technique présentée dans la figure 4.14 est le suivant :

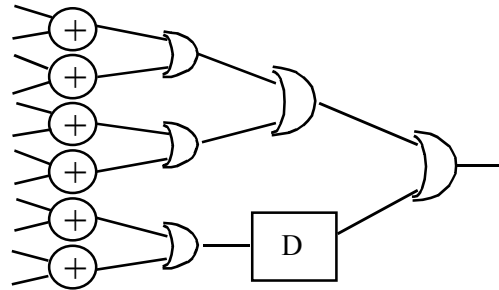
- On considère le temps de propagation du comparateur  $D_{comp}$ . On suppose que les sorties du circuit combinatoire sont stables et peuvent être capturées par le Output Latch à l'instant  $t_0$ . Alors, la valeur fournie par le comparateur est capturée par un latch supplémentaire Error-Latch, à l'instant  $t_0 + \delta + D_{comp}$ . Cette valeur est le résultat de comparaison des valeurs présentes aux entrées du comparateur à l'instant  $t_0 + \delta$ . Ces valeurs sont, d'un part, le contenu du Output-Latch détenant la valeur présentée à la sortie OUT à l'instant  $t_0$ , et d'autre part la valeur présente à la sortie OUT à l'instant  $t_0 + \delta$ . Donc, on a comparé les valeurs présentes sur la sortie OUT aux instants  $t_0$  et  $t_0 + \delta$ . Le fonctionnement du schéma présenté dans la figure 4.16. est équivalent au fonctionnement du schéma présenté dans la figure 4.13.b.

Le schéma utilise deux signaux d'horloge  $C_k$  et  $C_k + \delta + D_{comp}$ , où  $D_{comp}$  est le temps de propagation du comparateur. Pour simplifier encore plus le schéma, on peut utiliser un seul signal d'horloge  $C_k$ , pour les deux latches, Output-Latch et le Error Latch. Pour l'implémentation, on va utiliser le front descendant de l'horloge  $C_k$  pour capturer les valeurs de sortie du comparateur dans Error Latch. De plus, un bloc de retard  $T_1 - \delta - D_{comp}$  sera ajouté à la deuxième entrée du comparateur, où  $T_1$  est la durée du niveau logique '1' du signal d'horloge  $C_k$ .

Une autre possibilité est l'utilisation du front montant de l'horloge comme front actif pour le Error Latch. Dans ce cas, le retard ajouté à l'entrée du comparateur sera  $T - \delta - D_{comp}$ , où  $T$  est la période de l'horloge  $C_k$ .

Un inconvénient à cette implémentation est qu'elle nécessite que tous les chemins de propagation entre les entrées et les sorties du comparateur aient des temps de propagation similaires. Pour les comparateurs ayant un nombre d'entrées de type  $2^k$ , ce principe est toujours valable. Pour les autres cas, il est nécessaire d'ajouter des éléments de retard. Ce principe est présenté dans la figure 4.17, où est illustré un comparateur à 6 paires

d'entrées. Un élément de retard d'une valeur égale à la durée de propagation d'une porte logique 'OU' est ajouté pour équilibrer les délais de propagation du comparateur.



**Figure 4.17. Comparateur à temps de propagation équilibrés.**

Les deux techniques de détection de fautes présentées dans cette section assurent un surcoût d'implémentation minimale. La première technique sera utilisée pour les circuits qui présentent un grand nombre de chemins ayant des temps de propagation dispersés (cas des multiplicateurs Brown et des additionneurs), ou dans le cas d'une valeur élevée pour le  $\delta$  (cas de fautes de timing). La deuxième technique sera intéressante pour les circuits dans lesquels le rapport *APO* est grand (additionneurs). Par contre, la deuxième technique risque de réduire l'efficacité de détection d'erreurs produites par les particules frappant les latches Output Latch 1. Ceci peut arriver si l'erreur se produit à la fin du cycle d'horloge.

### Evaluation du surcoût d'implémentation

Les deux principes de réduction de coût matériel ont été vérifiés dans des multiplicateurs conçus pour tolérer les erreurs de timing ( $\delta = 20\%$  de  $D_{max}$ ) et des additionneurs. La dernière colonne du tableau 4.5 montre les résultats pour des multiplicateurs configurés conformément à la figure 4.13. On vérifie que le coût supplémentaire est réduit de façon significative.

Les tableaux 4.7 et 4.8 montrent l'implémentation des principes présentés dans les figures 4.15 et 4.16., respectivement, pour des additionneurs.

|                             | CP(ns) | TAO(%)          |                |                |              |         |
|-----------------------------|--------|-----------------|----------------|----------------|--------------|---------|
|                             |        | $\delta=0.46ns$ | $\delta=0.6ns$ | $\delta=0.8ns$ | $\delta=1ns$ | 20%Dmax |
| <b>Brent &amp; Kung 8</b>   | 3      | 25.41           | 31.50          | 34.98          | 39.33        | 31.50   |
| <b>Brent &amp; Kung 16</b>  | 4.2    | 24.60           | 25.24          | 29.38          | 39.37        | 32.2    |
| <b>Brent &amp; Kung 32</b>  | 5.1    | 22.02           | 23.01          | 26.97          | 35.84        | 35.84   |
| <b>Han &amp; Carlson 8</b>  | 2.8    | 24.29           | 28.31          | 34.65          | 45.87        | 26.16   |
| <b>Han &amp; Carlson16</b>  | 3.6    | 22.01           | 26             | 31.63          | 37.66        | 25.8    |
| <b>Han &amp; Carlson32</b>  | 4.2    | 18.64           | 20.02          | 23.50          | 33.71        | 25.2    |
| <b>Kogge &amp; Stone 8</b>  | 2.2    | 19.89           | 22.12          | 25.25          | 28.76        | 19.89   |
| <b>Kogge &amp; Stone 16</b> | 2.5    | 19.16           | 20.1           | 23.66          | 27.84        | 19.8    |
| <b>Kogge &amp; Stone 32</b> | 3.1    | 14.83           | 15.83          | 17.95          | 22.91        | 15.83   |
| <b>Sklanski 8</b>           | 2.3    | 22.16           | 25.41          | 30.48          | 38.41        | 22.16   |
| <b>Sklanski 16</b>          | 2.5    | 21.06           | 23.89          | 27.54          | 32.93        | 22      |
| <b>Sklanski 32</b>          | 4.9    | 17.98           | 19.96          | 21.89          | 25.84        | 21.9    |
| <b>CLA 8</b>                | 3.1    | 23.88           | 26.51          | 30.90          | 43.63        | 26.51   |
| <b>CLA 16</b>               | 3.6    | 26.97           | 30.64          | 33.31          | 35.76        | 33.31   |
| <b>CLA 32</b>               | 5.2    | 24.87           | 26.24          | 27.07          | 38.84        | 40.2    |

Tableau 4.7. Surcoût matériel pour l'implémentation de la figure 4.15.

On observe une réduction très importante du coût matériel supplémentaire par comparaison aux résultats du tableau 4.5. Les meilleurs résultats sont obtenus pour l'implémentation du principe présenté par la figure 4.16 et montrés dans le tableau 4.8 - à l'exception de 14 cas d'additionneurs (en grisé dans le tableau 4.7).

|                             | CP (ns) | TAO(%)        |              |              |            |         |
|-----------------------------|---------|---------------|--------------|--------------|------------|---------|
|                             |         | $\delta=0.46$ | $\delta=0.6$ | $\delta=0.8$ | $\delta=1$ | 20%Dmax |
| <b>Brent &amp; Kung 8</b>   | 3       | 16.79         | 26.19        | 36.37        | 38.55      | 26.19   |
| <b>Brent &amp; Kung 16</b>  | 4.2     | 15.46         | 25.98        | 32.57        | 37.88      | 35.1    |
| <b>Brent &amp; Kung 32</b>  | 5.1     | 14.56         | 24.57        | 31.87        | 36.72      | 36.72   |
| <b>Han &amp; Carlson 8</b>  | 2.8     | 15.7          | 25.64        | 33.47        | 37.91      | 21.19   |
| <b>Han &amp; Carlson 16</b> | 3.6     | 13.85         | 22.79        | 30.76        | 33.19      | 24.5    |
| <b>Han &amp; Carlson 32</b> | 4.2     | 12.31         | 21.44        | 25.81        | 32.72      | 27.1    |
| <b>Kogge &amp; Stone 8</b>  | 2.2     | 12.7          | 14.82        | 18.87        | 26.62      | 12.7    |
| <b>Kogge &amp; Stone 16</b> | 2.5     | 11.84         | 12.73        | 16.18        | 22.21      | 12.2    |
| <b>Kogge &amp; Stone 32</b> | 3.1     | 9.84          | 10.62        | 14.19        | 18.69      | 10.62   |
| <b>Sklanski 8</b>           | 2.3     | 14.01         | 17.20        | 23.07        | 32.11      | 14.01   |
| <b>Sklanski 16</b>          | 2.5     | 12.58         | 15.77        | 20.61        | 26.81      | 14.9    |
| <b>Sklanski 32</b>          | 4.9     | 11.88         | 14.5         | 18.54        | 23.56      | 18.8    |
| <b>CLA 8</b>                | 3.1     | 14.96         | 20.67        | 30.33        | 46.13      | 20.67   |
| <b>CLA 16</b>               | 3.6     | 17.3          | 23.75        | 29.31        | 46.33      | 29.31   |
| <b>CLA 32</b>               | 5.2     | 17            | 23.7         | 27.83        | 46.15      | 48.6    |

Table 4.8. Surcoût matériel des additionneurs implémentés conformément à la figure 4.16.

La colonne d'intérêt pratique est celle associée aux valeurs d'impulsion transitoire  $\delta$  de 0,46ns (à gauche ; les autres colonnes considèrent des impulsions trop longues par rapport à la réalité). La colonne correspond à une durée de 20% du temps de propagation maximale ( $D_{max}$ ), car elle peut être utilisée pour la détection d'erreurs de timing. Pour  $\delta = 20\%$  de  $D_{max}$ , on observe que le surcoût varie de 10,62% pour l'additionneur à 32 bits Kogge & Stone, à 40,2% pour l'additionneur à 32 bits CLA. Le coût est assez bas pour l'additionneur Kogge & Stone, avec des valeurs qui passent de 10,62% à 12,7% (colonne  $\delta = 20\%$  de  $D_{max}$ ), et de 14,19% à 18,87% (colonne  $\delta = 0,8$  ns). Ces résultats sont encore plus intéressants parce qu'ils concernent les additionneurs les plus rapides (voir la colonne CP : le chemin de propagation critique).

#### **4.3.5. Evaluation de l'efficacité de détection des fautes transitoires et des erreurs de timing**

L'efficacité de détection des fautes transitoires a été évaluée par la simulation de défauts, en utilisant le simulateur décrit dans le chapitre 2. Comme l'espace de la simulation est énorme (produit du nombre de nœuds par le nombre d'instants d'injection par le nombre de vecteurs d'entrée du circuit), il est impossible de le couvrir exhaustivement. Néanmoins, grâce à la rapidité du simulateur, nous allons étudier un nombre de fautes transitoires très élevé pour augmenter l'efficacité de notre évaluation. Nous avons choisi pour ces expérimentations les additionneurs et multiplieurs de 16 bits.

Pour chaque circuit, les nœuds affectés lors de l'injection de fautes ont été choisis aléatoirement, et la durée de l'impulsion transitoire a été fixée à 0,4 ns. Pour chaque point d'injection, on calcule le rapport entre le nombre d'erreurs détectées et le nombre d'erreurs affectant la sortie du circuit (l'efficacité de détection calculée au niveau du nœud). L'efficacité globale de la méthode de tolérance aux fautes est obtenue par un calcul de moyenne. Les résultats obtenus sont présentés dans le tableau 4.9.

|                      | <b>Fonct.<br/>Correct %</b> | <b>Efficacité<br/>%</b> | <b>Sensibilité %</b> |
|----------------------|-----------------------------|-------------------------|----------------------|
| <b>BK16x16</b>       | 97.46                       | 96.46                   | 2.24                 |
| <b>CLA16x16</b>      | 97.51                       | 91.64                   | 2                    |
| <b>HC16x16</b>       | 97.53                       | 98.21                   | 2.22                 |
| <b>KS16x16</b>       | 95.13                       | 92.29                   | 3.43                 |
| <b>SKL16x16</b>      | 95.02                       | 91.90                   | 4.11                 |
| <b>Wallace</b>       | 92.14                       | 89.89                   | 1.97                 |
| <b>Booth-Wallace</b> | 92.9                        | 90 .1                   | 9.89                 |

**Tableau 4.9 Efficacité de détection du schéma présenté dans la figure 4.16.**

Les injections de fautes transitoires sont faites dans 50% des nœuds de chaque circuit, et un nombre très important de vecteurs de test a été appliqué. La probabilité de fonctionnement correct est très élevée, surtout pour les additionneurs. La probabilité de détection correspond à la sortie du comparateur positionnée à la valeur '1' quand les sorties du latch de sortie sont affectées. Les non-détections correspondent à une valeur erronée sur les sorties du circuit qui n'est pas signalée par le comparateur. Etant donné que les circuits simulés ont été conçus de façon à détecter toute impulsion d'une durée inférieure ou égale à 0,4 ns, affectant une ou plusieurs sorties du circuit, une erreur pourra ne pas être détectée uniquement dans le cas d'une amplification de l'impulsion originale dû aux chemins reconvergeants ayant des retards différents de propagation.

La dernière colonne du tableau 4.9. présente la sensibilité des circuits en absence d'une méthode de détection d'erreur. La sensibilité est définie comme le rapport entre le nombre d'erreurs capturées par les latch de sorties et le nombre total de fautes injectés dans le circuit. Par exemple, pour l'additionneur Brent & Kung, pour 1 million de fautes injectées, la sortie sera erronée 22400 fois. Ce circuit était évalué à 100 MHz. Ceci signifie que pour un circuit similaire, fonctionnant à 1 GHz, la sortie sera erronée 224000 fois par injection du même nombre de fautes. L'efficacité de détection (colonne efficacité %) varie de 90% à 98% selon le type de circuit. Puisque les cas de non-détection sont dus à l'amplification de la durée de l'impulsion par les chemins du circuit (qui pourra ainsi dépasser la valeur de  $\delta - D_{\text{setup}} - D_{\text{hold}}$ ), on devrait améliorer cette efficacité en augmentant  $\delta$ . Et, en effet, on constate une très forte influence de ce paramètre. En

augmentant  $\delta$  de 0,1 ns, on constate une diminution des cas d'erreurs non-détectées de moitié.

Concernant l'efficacité de détection pour les fautes de timing, on observe qu'un retard supplémentaire  $\delta$  distribué sur un ou plusieurs chemins du circuit ajoute le même  $\delta$  sur le retard total du chemin. C'est-à-dire que, pour les fautes de retard, il n'y a pas d'amplification de  $\delta$ . Ainsi, le schéma utilisant un décalage  $\delta$  entre les horloges de Output Latch et de l'Extra Latch détecte tous les fautes de timing introduisant un retard plus petit ou égal à  $\delta$ . Nous avons donc une efficacité de détection de 100% pour ces fautes, et il n'est pas besoin d'effectuer la simulation de fautes.

En conclusion, le schéma de détection proposé assure un niveau d'efficacité de détection élevé, pour un coût d'implémentation très faible. De plus, le circuit n'est pas affecté en termes de vitesse de propagation. Cette technique s'avère très intéressante pour les circuits logiques se trouvant dans le chemin de propagation critique d'un système.

## **Conclusion**

La nouvelle technique de tolérance aux fautes combinant la redondance temporelle avec des circuits de préservation de l'état correct offre une très bonne efficacité de correction de fautes tout en ayant un surcoût de surface très faible par rapport aux autres techniques analysées. Par contre, l'impact sur les performances en vitesse du circuit est significatif. Ainsi, les solutions des figures 4.4 et 4.10 peuvent être utilisées dans les parties critiques d'un circuit si la vitesse de fonctionnement n'est pas un paramètre critique. La solution 4.10 ayant un très faible coût matériel sera utilisée quand un allongement de la période d'horloge de l'ordre de  $2D_{TR}$  est acceptable. Mais si cette perte de vitesse est inacceptable, on devra payer un coût matériel significatif pour implémenter la solution de la figure 4.4. afin de diviser par deux la perte en vitesse. Les choses sont plus faciles si on doit traiter des parties n'appartenant pas au chemin critique du circuit. Dans ce cas, le retard réduit du circuit pourra permettre l'implémentation de la technique sans être obligé d'allonger la période d'horloge. Selon les marges de retard dont on dispose, on choisira la solution de la figure 4.10, si les marges de retard sont suffisamment larges, ou celle de la

figure 4.4, si les marges de délai sont plus courtes. Finalement, on pourra aussi traiter les parties appartenant aux chemins critiques du circuit, sans allonger la période d'horloge (en employant la solution de la figure 4.4), ou par un allongement de l'ordre de  $D_{TR}$  (en employant la solution de figure 4.10). Cette diminution de l'allongement de la période d'horloge aura pour conséquence une réduction de l'efficacité de la tolérance aux fautes. Ainsi, à chaque fois qu'une sortie est calculée en activant des chemins critiques, la sortie ne sera pas protégée. Mais, dans les cycles d'horloge qui calculent la sortie en activant des chemins plus courts, la sortie se trouvera protégée. La protection sera plus efficace quand la différence entre le délai du chemin critique et le délai des chemins activés sera plus grande. Cette solution sera très intéressante là où les chemins de chaque sortie seront activés pendant une petite fraction du temps. Il en est ainsi, par exemple, des circuits dont la majorité des chemins sont plus courts que le chemin critique.





## Conclusions et perspectives

Dans cette étude, nous avons analysé divers aspects de l'évolution des technologies CMOS vers le domaine nanométrique, qui nous ont amené à la conclusion qu'il y a un renversement de tendances concernant la fiabilité des circuits intégrés. Si dans le passé, les technologies des circuits intégrés ont permis une amélioration importante de la fiabilité des systèmes électroniques, nous sommes entrés actuellement dans une phase de dégradation accélérée des circuits intégrés. Les raisons sont diverses.

Tout d'abord, l'augmentation de la vitesse, la réduction du niveau d'alimentation, plusieurs effets de deuxième ordre qui étaient négligeables dans les technologies micrométriques, prennent de l'ampleur et, dans certains cas, dominent le comportement du circuit. Ces phénomènes très complexes ne sont pas modélisés par les outils de CAO existants, laissant le concepteur démuni devant leur émergence. Des outils modélisant ces phénomènes de façon plus ou moins précise vont apparaître, mais même avec des outils adéquats, la maîtrise de ces phénomènes nécessitera un rallongement considérable du temps de conception. Ceci est incompatible avec la tendance du marché actuel qui impose la réduction du temps de mise sur le marché d'un nouveau produit, et augmente le risque de voir des produits défectueux pour cause d'erreurs de conception (voir le retrait récent du processeur Pentium III, 1.13 GHz). Cependant, même pour les systèmes parfaitement conçus, la probabilité d'introduire sur le marché des produits contenant des défauts de fabrication augmente, car l'impact des défauts sur le comportement des circuits devient très complexe. Ainsi, le temps CPU pour la génération de vecteurs de test, la simulation de fautes et la longueur du test deviennent irréalistes.

Dans le même temps, et pour les mêmes causes (réduction des grandeurs géométriques, augmentation de la vitesse et réduction des niveaux d'alimentation), on observe une augmentation de la sensibilité des circuits aux diverses sources de perturbation, et notamment aux particules alpha et aux neutrons atmosphériques.

On observe donc, d'une part, une aggravation de trois facteurs de défaillances dans les circuits intégrés : les erreurs de conception, les défauts de fabrication échappant aux tests de fabrication et la sensibilité aux fautes transitoires. D'autre part, le seul moyen permettant d'arrêter cette aggravation consiste à bloquer la progression vers les technologies nanométriques. En se basant sur cette analyse, nous avons conclu que la tolérance aux fautes devra s'imposer comme la seule technologie capable d'arrêter cette dégradation de fiabilité, afin de permettre la poursuite du progrès technologique (continuation de la loi de Moore).

Mais le coût élevé de la tolérance aux fautes traditionnelle rend cette approche inacceptable pour les produits orientés « grand public ». Nous avons donc entrepris l'étude de techniques innovantes capables d'implémenter la tolérance aux fautes à faible coût.

Dans un premier temps, nous avons étudié le comportement des circuits auto-contrôlables face aux fautes de timing. En montrant que les implémentations classiques des circuits auto-contrôlables développés pour les collages logiques sont inefficaces pour ces fautes, nous avons proposé par la suite des solutions offrant une meilleure efficacité de détection et nécessitant un coût d'implémentation plus faible.

Par la suite, dans le but d'une réduction du coût matériel plus poussée, nous avons étudié des approches utilisant la redondance temporelle. On montre qu'un circuit logique tolérant les fautes transitoires peut être implémenté à faible coût grâce à cette approche, mais le concepteur devant payer une réduction de vitesse non négligeable. L'approche redondance matérielle/redondance temporelle divise par deux cette réduction de vitesse, mais au prix de la duplication du circuit. Ces techniques sont intéressantes pour les parties qui se trouvent en dehors des chemins critiques du circuit. Pour les autres parties, des techniques qui n'affectent pas la vitesse du circuit sont nécessaires. La détection de fautes basée sur la redondance temporelle est montrée comme la solution idéale dans ce contexte. Elle offre une efficacité de détection élevée pour les fautes transitoires et les fautes de timing au prix d'un faible coût matériel, et sans réduction de la vitesse.

Notons au passage que ces analyses ont été rendues possibles grâce à un outil de simulation de fautes transitoires que nous avons mis au point. Il nous a permis d'évaluer les différentes techniques et par l'intermédiaire de ces évaluations, de faire le choix d'améliorer leur efficacité. Ces techniques offrent un large choix de solutions qui devraient faciliter la conception des circuits intégrés tolérants les fautes temporaires à faible coût matériel et sans impact sur la vitesse de fonctionnement.

Ces techniques constituent une première base pour développer un environnement de conception pour les prochaines générations de circuits intégrés. Dans cette perspective, plusieurs voies sont à explorer, tels que le développement d'outils permettant l'automatisation de ces techniques, le développement d'outils de simulation au niveau physique et électrique permettant de simuler les phénomènes qui apparaissent entre l'impact de la particule et la création de l'impulsion transitoire (où commence le domaine d'application de notre simulateur), le développement des techniques et architectures permettant la reprise au niveau circuit intégré, ... , l'expérimentation de ces techniques dans le cadre de conception des circuits d'intérêt commercial, et la validation de ces techniques par des tests sous radiations.

Notons à cette occasion que les techniques présentées dans ce manuscrit ont été utilisées dans un projet industriel qui concerne l'implémentation d'un processeur compatible SPARC, tolérant les fautes transitoires. Le faible coût matériel, et le maintien de la vitesse du circuit, obtenus dans ce projet, montrent que nos conclusions basées sur des circuits combinatoires isolés restent valables dans le cadre des circuits intégrés de complexité industrielle.

Tels développements sont très importants, car ils ont pour but de lever une barrière technologique majeure, selon Intel, les erreurs transitoires sont le plus grand défi pour les circuits sous-microniques, après l'augmentation des courants de fuite.



## REFERENCES

- [AITK 99] AITKEN R., "Nanometer Technology Effects on Fault Models for IC Testing", *Computer Design & Test, November 1999, pg. 46-51.*
- [ALEX 00] ALEXANDRESCU D., "Méthodes et outils pour la simulation des fautes transitoires", *Rapport DEA, Juin 2000.*
- [ALZ 99] ALZAHER I., NICOLAIDIS M., "A Tool for Automatic Generation of Self-Checking Multipliers Based on Residu Arithmetic Codes, "Design, Automation and Test in Europe Conference, March Munich, Germany, 1999.
- [AND 71] ANDERSON D.A., "Design of Self-Checking Digital Networks Using Coding Techniques. Coordinates. Sciences Laboratory", *Report R/527. University of Illinois, Urbana, September 1971.*
- [AND 73] ANDERSON D.A , METZ G., "Design of Totally Self-Checking Check Circuits for m-out-of-n Codes", *IEEE Trans. on Comput., Vol. C-22, pp.263-269, March 1973.*
- [ANG 00a] ANGHEL L., NICOLAIDIS M. "Cost Reduction and Evaluation of a Temporary Fault Detection Technique », *Proceedings of DATE'00., pp. , Paris, March 2000.*
- [ANG 00b] ANGHEL L., NICOLAIDIS M., ALZAHER I. "Self Checking Circuits versus Realistic Faults in Very Deep Submicron", *Proceedings of VTS'00, pp.263-269, Montreal, May 2000.*
- [ANG 00c] ANGHEL L., ALEXANDRESCU D., NICOLAIDIS M. "Self Checking Circuits versus Realistic Faults in Very Deep Submicron", *Proceedings of SBCCI'00, pp.237-242, Manaus, Brasil, September 2000.*

- [ASH 77] **ASHJAE M.J., REDDY S.M.**, "On-Totally Self-Checking Checkers for Separable Codes", *IEEE Trans. on Comp.*, vol. C-26, pp. 737-744, Aug. 1977.
- [AVI 73] **AVIZIENIS A.**, "Arithmetic Algorithms for Error-Coded Operands" *IEEE Trans. on Comput.*, Vol. C-22, No. 6, pp.567-572, June 1973.
- [BAUM 99] **BAUMANN R.**, "Neutron Induced Boron Fission as a Major Source of Soft Errors in Deep Submicron SRAM Devices"
- [BAZ 97] **BAZE M., BUCHNER S.**, "Attenuation of Single Event Induced Pulses in CMOS Combinational Logic", *IEEE Trans. on Nuclear Science*, Vol. 44, No 6, December 1997.
- [BEN 75] **BENOWITZ N., et al.** , "An Advanced Fault Isolation System for Digital Logic", *IEEE Trans. On Computers*, vol. C-24, No. 5, pp. 489-497, May 1975
- [BES'94] **D. BESSOT, R. VELAZCO**, « Design of SEU Hardened CMOS Memory Cell : The HIT Cell », Proceedings 1994, RADECS Conference, pp. 563-570.
- [BON 79] **BONEY J., RUPP E.** , "Let Your Next Microcomputer Check Itself and Cut Down Your testing Overhear", *Electronic Design*, pp. 101-106, 1979
- [BOUR 91] **BOURRIEAU J.**, "L'environnement spatial : flux, dose, blindage, effets des ions lourds", *Tutorial Short Course, RADECS'91, 1991.*
- [BRE 82] **BRENT R., KUNG H.**, "A regular layout for parallel adders", *IEEE Transactions on Computers*, vol. C-31, n 3, pp. 260-264, March 1982.
- [CALI 96] **CALIN T. et al.**"Upset Hardened Memory Design for Submicron CMOS Technology", *IEEE Transaction on Nuclear Science*, vol. 43, pp. 2874-2878, 1996.
- [CAR 68] **CARTER W.C., SCHNEIDER P.R.**, "Design of dynamically checked computers", *Proc.4<sup>th</sup> Congress IFIP*, vol.2, Edinburgh, Scotland, Aug. 5-10 1968, pp. 878-883.
- [CAT 99] **CATALDO A.**, "Intel Scans for Soft Errors in Processor Designs", *EETimes*, 06/16/99
- [CHA 93] **CHA H., PATEL J.**, "A logic-level model for alpha-particle hits in CMOS circuits", *Proc. Int. Conf. Computer Design*, Oct. 1993, pp. 538-542.
- [COH 99] **COHEN N.** " Soft Error Considerations for DSM CMOS Circuit Applications", 1999

- [DE 94] NATARAJAN K. De. C., NAIR D., and BANERJEE P., "RSYN: A System for Automated Synthesis of Reliable Multilevel Circuits, *IEEE Transactions on VLSI Systems*, Volume 2, number 2, June 1994, pp. 186-195.
- [DELL 99] DELL T., "" EDTN, June 1999.
- [DET 96] DETCHEVERRY C. et al. "SEU Critical and Sensitive Area in a Submicron CMOS Technology", *IEEE Transaction on Nuclear Science*, vol. 44, no. 6, decembre 1996.
- [DIEH 83] DIEHL S.E., et al. "Consideration for Single Event Upset Immune VLSI Logic", *IEEE Transactions on Nuclear Science*, vol. 30, pp. 4501-4508, 1983.
- [FAS 82] FASANG P.P. "A Fault Detection and Isolation technique for Microprocessors", *Digest of papers 1982 Intn'l Test Conf.* pp. 214-219, November 1982
- [FUJ 83] FUJIWARA H, and SHIMONO T., "On the Acceleration of Test Generation Algorithm", *IEEE Trans. On Computers*, Vol. C-32, No. 12, pp. 1137-1144, December, 1983
- [FUN 75] FUNATSU S., WAKATSUKI N. and YAMADA A., "Designing Digital Circuits with Easy Testable Considerations", *Proc. Test Conf.* , pp. 98-102, September, 1978.
- [GEL 87] GELSINGER P.P., "Design and test of the 80386", *IEEE Design & Test of Computers*, vol. 4, No. 3, pp. 42-50, June 1987.
- [GOEL 81] GOEL P., "An Implicit Enumeration Algorithm to Generate Tests for Combinational Logic Circuits", *IEEE Trans. On Computers*, Vol. C-30, No.3, pp. 215-222, March, 1981
- [GREE 98] GREEN L. "Addressing the Effects of Signal Integrity in Deep-Submicron Design", *ISD Mag, Cover Story*, June 1998
- [HAN 87] HAN T., CARLSON D., "Fast area efficient VLSI Adders", *8<sup>th</sup> Symposium on Computer Arithmetic*, pp. 49-56, May 1987.
- [HWA 79] HWANG K., "Computer Arithmetic, Principles, Architectures and Design", *Jhon Wiley and Sons, New York*, 1979
- [JHA 93] JHA N.K., WANG S-J, "Design and Synthesis of Self-Checking VLSI Circuits", *IEEE Trans. Comp. Aided Des.*, vol. 12pp. 878-887, June 1993.

- [KAR 89] KARPOVSKY M.G., NAGVAJARA P., "Design of Self-Diagnostic Boards by Signature Analysis", *IEEE Trans. On Industrial Electronics*, vol. 36, No. 2, pp. 241-245, May 1989.
- [KATZ 95] KATZ I., "What System Designers Need to Know about Deep-submicron ASICs", *EDN Mag*, 16 february 1995
- [KOB 68] KOBAYASKI T., MATSUE T., SHIBA H. "Flip-Flop Circuit with FLT Capability", *Proc. IECEO Conf.* p. 692, 1968.
- [KOG 73] KOGGE P.M., STONE H., "A Parallel algorithm for efficient solution of a general class of recurrence equations", *IEEE Transactions on Computers*, vol. C-22, n 8, pp. 786-792, August 1973
- [LAMM 98] LAMMERS D., GOERING R., "Compiled ASIC Libraries Ease Path to 0.25-Micron Technologies", 23 february 1998, *EE Times*.
- [MA 89] MA T.P., DUSSENDORFER P., "Ionizing Radiation Effects in Mos Devices and Circuits", *Willey, New York*, 1989.
- [MAY 78] MAY T.C., WOODS M., "A New Physical Mechanism for Soft Errors in Dynamic memories", *Proc. 16 Int'l Reliability Physics Symposium*, pp. 33, April 1978.
- [McC 71] MCCLUSKEY E.J., CLEGG F.W. "Fault equivalence in combinational logic networks", *IEEE Trans. Comp.*, Vol. C-20, pp. 1286-1293, Nov. 1971.
- [MUSS 96] MUSSEAU O., "Single Event Effects in SOI technologies and Devices", *IEEE Transactions on Nuclear Science*, vol. 43, pp. 603-613, 1996.
- [NIC 86] NICOLAIDIS M., COURTOIS B., "Design of Self-Checking Circuits Using Unidirectional Error Detecting Codes". In *Proc. of the 16<sup>th</sup> FTCS Vienna, July 1986, Austria*.
- [NIC 87] NICOLAIDIS M., "Shorts in Self-Checking circuits", *Proc. Int. Test Conf., Washington, D.C., Sept. 1987*
- [NIC 91] NICOLAIDIS M., BOUDJIT M. "New Implementations, Tools and Experiments for Decreasing Self-Checking PLAs Area Overhead"*IEEE International Conference on Computer Design (ICCD), Cambridge, Massachusetts, October, 1991*.
- [NIC 93] NICOLAIDIS M. "Efficient Implementations of Self-Checking Adders and ALUs"*In Proc. 23th IEEE International Symposium on Fault Tolerant Computing, Toulouse, France, June 1993*.



- [NIC 97] **NICOLAIDIS M., DUARTE R.O., MANICH S., FIGUERAS J.** "Achieving Fault Secureness in Parity Prediction Arithmetic Operators", *IEEE Design & Test of Comp.*, April-June 1997.
- [NIC 99] **NICOLAIDIS M.**, "Time Redundancy Based Soft-Error Tolerance to Rescue Nanometer Technologies", *In proceedings VTS 99*.
- [NIC 99] **NICOLAIDIS M., DUARTE R.O.**, "Design of Fault-Secure Parity-Prediction Booth Multipliers", *IEEE Design & Test of Comp.*, July-Sept. 1999.
- [NIC 00] **NICOLAIDIS M.**, " Carry Checking/Parity Prediction Adders and ALUs", *to appear IEEE Trans. on VLSI Systems*.
- [NORM 96] **NORMAND E.** "Single Event Upset at Ground Level", *IEEE Transactions on Nuclear Science*, vol. 43, no. 6, December 1996.
- [PET 58] **PETERSON W.W.**"On checking an Adder", *IBM J. Res. Develop.* 2, pp.166-168, April 1958.
- [PET 72] **PETERSON W.W., WELDON E.J.**, "Error-Correcting Codes", *second Ed., The MIT press, Cambridge, Massachusetts, 1972*
- [PET 83] **PETERSEN E.L.**, "Single Event Upsets in Space : Basic Concepts", *Tutorial Short Course, IEEE NSREC'83, 1983*.
- [PICK 83] **PICKEL J.K.**, "Single Events Upsets Mechanisms and Predictions", *Tutorial Short Course, IEEE NSREC'83, 1983*.
- [PRAD 86] **PRADHAN D.** "Fault Tolerant Computing : Theory and Techniques", *Prentice Hall 1986*.
- [ROC 88] **ROCKETT L.**, " An SEU Hardened CMOS Data Latch Design ", *IEEE Trans. On Nuclear Sc. Vol. NS-35, n. 6, Dec. 1988, pp. 1682-1687*.
- [ROTH 66] **ROTH J.P.** "Diagnosis of Automata Failure : A Calcul and a Method", *IBM Journal of Research and Development, Vol. 10, No. 4, pp. 278-291, July, 1966*
- [ROTH 67] **ROTH J.P., BOURICIUS W.G. and SCHNEIDER P.R.**, "Programmed Algorithm to Compute Tests to Detect and Distinguish Between Failure in Logic Circuits", *IEEE Trans. On Electronic Computer, Vol. EC-16, No.10, pp.567-579, October 1967*
- [SAP 96] **SAPOSHNIKOV V.V., SAPOSHNIKOV VI.V., MOROSOV A., GOSSEL M.**, "Design of Self-Checking Unidirectional Combinational Circuits with Low Area Overhead", *2<sup>nd</sup> IEEE Intl. On-Line Testing Workshop, Saint Jean de Luz, Biarritz, France, July 1996*.

- [SCI 00] ACTUALITÉS, Intel, Science et Avenir, octobre 2000.
- [SEM 98] SEMATECH "Quality and Reliability Issues in the National Technology Roadmap for Semiconductors", 1998
- [SEXT 89] SEXTON F.W., "SEU Characterisation of Hardware CMOS 64K and 256K SRAM", *IEEE Transactions on Nuclear Science*, vol. 36, pp. 2311-2319, 1989.
- [SIEW 92] SIEWIOREK, SWARZ, "Reliable Computer Systems, Design and Evaluation", *second edition*, 1992
- [SILB 84] SILBERBERG R. et al."Neutron Generated Single Event Upset in the Atmosphere", *IEEE Transactions on Nuclear Science*, NS -31, December 1984.
- [SKL 60] SKLANSKI J., "Conditional-sum addition logic", *IRE Transaction on Electronic Computers*, vol. EC-9, n 2, pp. 226-231, June 1960.
- [SMI 77] SMITH J.E, METZE G."The Design of Totally Self-Checking Combinatorials Circuits", *Proc. 7<sup>th</sup> Fault Tolerant Computing Symposium, Loss Angeles, USA, June 1997.*
- [SMI 78] SMITH J.E., METZE G. "Strongly Fault Secure logic networks", *IEEE Trans. on Comp. Vol. C-27 N°6, June 1978.*
- [SYL 99] SYLVESTER D., KEUTZER K., "Rethinking Deep-Submicron Circuit Design", *Computer Design & Test*, November 1999, pg. 25-33.
- [TAB 92] TABER A. et NORMAND E., "Investigations and Characterisation on SEU Effects and Hardening Strategies in Avionics", *IBM Report 92-L75-020-2, August 1992, republished as DNA-Report DNA-TR-94-123, feb. 1995.*
- [TOU 94] TOUBA N. A., MCCLUSKEY E.J., "Logic Synthesis Techniques for Reduced Area Implementation of Multilevel Circuits with Concurrent Error Detection", *Proc. of Int. Conf. on Comp. Aided Design*, 1994.
- [WES 94] WESTE N.H.E. et al. "Principles of CMOS VLSI Design- A System Perspective- second edition, *Addison-Wesley Publishing Co., 1994.*
- [WHI 91] WHITAKER S., CANARIS J. et al. « SEU Hardened Memory Cell for a CCSDS Reed Solomon Encoder », *IEEE. Trans. On Nuclear Sc. Vol. NS-38, n.6, Dec. 1991, pp. 1471-1477.*
- [WILL 99] WILLIAMS T., KAPUR R., "Tough Challenges as Design and Test Go Nanometer", *Computer Design & Test*, November 1999.
- [ZIEG 96] ZIEGLER J.F., "Terrestrial Cosmic Rays", *IBM 1996.*

## **RESUME EN FRANÇAIS :**

Les technologies de silicium s'approchent de leurs limites physiques en termes de réduction de tailles des transistors, et de la tension d'alimentation ( $V_{DD}$ ), d'augmentation de la vitesse de fonctionnement et du nombre de dispositifs intégrés dans une puce. En s'approchant de ces limites, les circuits deviennent de plus en plus sensibles à toute source de bruit (telles que les couplages capacitifs ou «cross-talks», l'influence électro-magnétique, le bruit sur les lignes d'alimentation «ground-bounce»), ainsi qu'aux phénomènes radiatifs (particules alpha et neutrons atmosphériques).

Ainsi, le taux d'erreurs du fonctionnement causées par l'impact des particules ionisantes (erreurs soft) ou par des défauts difficiles à détecter échappant ainsi au test de fabrication (par ex. fautes temporelles), se voit augmenté de façon radicale. Dans cette thèse, nous analysons dans un premier temps ces problèmes et nous concluons que tout circuit doit être conçu en utilisant des techniques de tolérance aux fautes afin de pouvoir maintenir des niveaux de fiabilité acceptables pour les prochaines générations de circuits nanométriques. Cette analyse montre que les parties logiques tendent de devenir aussi sensibles aux erreurs soft que les mémoires, nécessitant ainsi le même niveau de protection. Les techniques traditionnelles de tolérance aux fautes (TMR, duplication) étant trop coûteuses, ne sont pas acceptables pour les applications à faible valeur ajoutée (ex. produits grand public). Le caractère temporel des fautes transitoires et de timing est exploité afin de proposer des solutions efficaces utilisant des structures self-checking, ainsi que des techniques de redondance temporelle. Ces techniques minimisent le coût matériel et ont un faible impact sur la vitesse de fonctionnement du circuit. Nous avons aussi développé une méthodologie de simulation de fautes transitoires, qui nous a permis d'évaluer de façon précise l'efficacité de protection obtenue par ces techniques.

## **TITRE en anglais :**

### **FAULT TOLERANCE VERSUS TECHNOLOGICAL LIMITATIONS OF SILICON**

## **RESUME EN ANGLAIS :**

Integrated circuit technology is approaching the ultimate limits of silicon in terms of geometry shrinking, power supply level, speed and density. By approaching these limits, circuits are becoming increasingly sensitive to any noise source (such as cross-talks, electromagnetic influence, noise on the power line, ground bounce) as well as radiative phenomena (e.g. alpha particles and atmospheric neutrons).

Thus, the error rate due of the impact of ionizing particles (soft errors) or by the defects difficult to detect that may escape fabrication testing (e.g. timing faults) is drastically increased. In this thesis, we address these problems and we conclude that future integrated circuits have to be designed by using fault tolerance techniques, in order to maintain acceptable reliability levels. This analyze shows that logic parts are becoming as sensitive to soft errors as memories and therefore they need to be protected. Traditional fault tolerance techniques (e.g. TMR, duplication) are of a high cost, they are not acceptable for low added value applications (for example commercial products). The temporal nature of the transient and timing faults is exploited in order to obtain efficient solutions by using self-checking structures as well as time redundancy techniques. These techniques decrease the hardware cost and have a small impact on the circuit performances. We have also developed a transient fault simulation methodology, which has allowed us to evaluate the efficiency of these methods with a very good accuracy.

**Spécialité: MICROELECTRONIQUE**

**Mots-clés : TOLERANCE AUX FAUTES, FAUTES TRANSITOIRES, FAUTES DE TIMING, CIRCUITS AUTO-CONTROLABLES, TECHNOLOGIES SOUSMICRONIQUES, REDONDANCES TEMPORELLES**

**ISBN 2\_913329\_54\_3 broché**

**ISBN 2\_913329\_55\_1 format électronique**