ParisTech
INSTITUT DES SCIENCES ET TECHNOLOGIES
PARIS INSTITUTE OF TECHNOLOGY

TELECOM ParisTech

2012-ENST-0038

EDITE - ED 130

Doctorat ParisTech

THÈSE

pour obtenir le grade de docteur délivré par

TELECOM ParisTech

Spécialité « Communications et Electronique »

présentée et soutenue publiquement par

Tian BAN

le 4 septembre 2012

Méthodes et Architectures basées sur la Redondance Modulaire pour Circuits Combinatoires Tolérants aux Fautes

THÈSE

Directrice de thèse : **Mme. Lirida NAVINER**

Jury

M. Patrick Girard, Directeur de Recherche, CNRS, LIRMM	Rapporteur
M. Emmanuel Casseau, Professeur, ENSSAT, INRIA/IRISA	Rapporteur
M. Habib Mehrez, Professeur, UPMC, LIP6	Examinateur
M. Costin Anghel, Professeur associé, ISEP	Examinateur
Mme. Lirida Naviner, Professeur, Télécom ParisTech	Directrice de thèse

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

École Doctorale
d'Informatique,
Télécommunications
et Électronique de Paris

Thèse

Methods and Architectures based on Modular Redundancy for Fault-Tolerant Combinational Circuits

Tian BAN

Directrice de thèse Pr. Dr. Lirida NAVINER

« To
My Father
Mr. BAN Tong
and
My Mother
Mrs. ZHU Zhaoyun »

Acknowledgments

Tout d'abord, je tiens à remercier ma directrice de thèse, Mme. Lirida Naviner, qui a dirigé mes travaux de recherche avec toujours beaucoup de patience et disponibilité. En particulier, j'ai beaucoup apprécié la liberté qu'elle m'a accordée pour étudier différentes pistes (qu'elles soient incertaines ou de sentiers plus cadrés) toujours en assurant mon encadrement et avec le souci de me faire acquérir une vision globale de mon travail.

Je remercie M. Patrick Girard, M. Emmanuel Casseau, M. Habib Mehrez et M. Costin Anghel pour avoir accepté d' être membres de mon jury de thèse, en accordant leur temps et leur expérience.

Merci également à Dr. Jean-François Naviner, Dr. Hervé Petit et Dr. Philippe Matherat. Vos idées et suggestions ont renforcé le contenu de ce travail. Je voudrais aussi remercier tous les membres de notre équipe de recherche NanoElec (l'équipe "Nouvelles Problématiques de l'Electronique"), pour leur soutien pendant ma thèse.

Merci à ceux qui ont contribué, directement ou indirectement, à la réussite de cette thèse. Mon professeur de Français Langue Etragère, Julie Montagu, qui m'a appris à aimer cette belle langue. Les personnels administratifs de Télécom ParisTech, Florence Besnard, Martha Dwyer, Fabienne Lassausaie et Nazha Essakkaki, qui enlèvent les tracas de différents types de bureaucraties.

Un grand merci aux personnels administratifs du département COMELEC, en particulier Chantal Cadiat, Zouina Sahnoune et Yvonne Bansimba, qui ont fourni diverses aides pour mon travail et ma vie à l'école.

J'ai eu le plaisir de collaborer avec tous mes collègues venus de divers horizons. Etudier, travailler et déjeuner avec vous m'a offert une excellente occasion de mieux connaître différentes cultures et m'a fait voir le monde différemment.

Mes amis chinois dans le monde entier, je vous remercie pour votre disponibilité immédiate chaque fois j'ai eu des difficultés. Des remerciements particuliers à Feng Yan, Jian Wang et Kun Lee, en France, aux Etats-Unis et la Chine, respectivement. Je me permets de vous décrire en ramassant un mot technique dans ma thèse, fiable.

Un grand merci tout particulier à mes parents. Le talent, l'amour et la compréhension dont vous avez fait toujours preuve sont indispensables pour chacune de mes réalisations. La fin de ma scolarité vous semblait parfois lointaine... elle a pris presqu'un quart de siécle, mais voici qu'elle arrive !

Enfin, je tiens à remercier le CSC (China Scholarship Council) et le projet STIC AmSud Nanoradio pour leur soutien financier au cours de mon travail de recherche.

At the end of my thesis, I would like to thank my advisor, Mrs. Lirida Naviner, for her excellent guidance, patience and assistance while I pursued this research. In particular, thank you for ensuring that I saw the big picture and allowing me the absolute freedom to study the different tracks, sometimes uncertain paths even occasional stray.

An additional thank you to Mr. Patrick Girard, Mr. Emmanuel Casseau, Mr. Habib Mehrez and Mr. Costin Anghel for serving on my committee. Thank you for your time and experience on this thesis.

The same acknowledgments are due to Dr. Jean-François Naviner, Dr. Hervé Petit and Dr. Philippe Matherat. Your insights and suggestions have strengthened the content of this work. I would like to say thank you to all the former and present members in our research team NanoElec (Team of new issues in electronics), for their support during my thesis.

Thank you for those who contributed, directly or indirectly, to the success of this thesis. To my french teacher, Julie Montagu, who taught me to love this beautiful language. To the administrative staffs of Télécom ParisTech, Florence Besnard, Martha Dwyer, Fabienne Lassausaie and Nazha Essakkaki, who remove the hassles of different kinds of bureaucracies.

To the administrative staffs of COMELEC department, particularly Chantal Cadiat, Zouina Sahnoune and Yvonne Bansimba : thank you for providing miscellaneous assistances as I used and abused during my work and life on campus.

I am very happy to work with my colleagues from various backgrounds. Studying, working, lunch and coffee break with you offers me an excellent opportunity to learn about different cultures and made me see the world differently.

My Chinese friends all around the world, thank you for your immediate availability each time I am in difficulty. Particular thanks to Feng Yan, Jian Wang and Kun Lee, in France, United States and China respectively. Please allow me to describe you by picking up a technical word in my thesis, reliable.

The most special thanks are undoubtedly due to my parents. Talent, love and understanding derived from you enabled each achievement from kindergarten to graduate school. While you may complain sometimes that the end of my schooling seems to be unforeseeable, now it is the end, yet a bit long - nearly a quarter of century.

Lastly, I would like to thank CSC (China Scholarship Council) and STIC AmSud Nanoradio Project for their financial support during my research work.

Contents

List of Figures

List of Tables

Symbols and Abbreviations

A	Area (area cost)
C	Cost (overall cost)
ε_i	Error probability of inputs/outputs
ε_g	Gate error probability
$e(i)$	Eligibility of block
p_i	Error probabilities of inputs/outputs
q	Reliability of block
μ,$\mu(t)$	Repair rate
λ,$\lambda(t)$	Failure rate
$R, R(t)$	Reliability
ADC	Analog to digital converter
ADD	Algebraic decision diagram
ASIC	Application-specific integrated circuit
BDD	Binary decision diagram
BDEC	Boolean difference-based error calculator
BN	Bayesian network
BSC	Binary symmetric channel
CED	Concurrent error detection
CMOS	Complementary metal oxide semiconductor
DSM	Deep submicron
DSP	Digital signal processing
ECC	Error correcting code
EDA	Electronic design automation
EDC	Error detection code
FET	Field-effect transistor
FIFA	Fault-injection-fault-analysis
FPGA	Field-programmable gate array
HDL	Hardware description language
IC	Integrated circuits

IP	Intellectual property
ISCAS	International symposium on circuits and systems
ITM	Ideal transfer matrix
ITRS	International technology roadmap for semiconductors
LUT	Look-up table
MC	Monte carlo
MOS	Metal oxide semiconductor
MOSFET	Metal oxide semiconductor FET
MRF	Markov random field
MTBF	Mean time between failure
MTTF	Mean time to failure
MTTR	Mean time to repair
NIR	N-tuple interwoven redundancy
NMR	N-tuple modular redundancy
PBR	Probabilistic binomial reliability model
PGM	Probabilistic gate model
PLA	Programmable logic array
PLD	Programmable logic device
PMR	Progressive modular redundancy
PMMR	Progressive mixed modular redundancy
PTM	Probabilistic transfer matrix
PTMR	Progressive modular redundancy
QCA	Quantum cellular automaton
RE	Relative error
RTD	Resonant tunneling devices
RTL	Register-transfer level
SCM	Stochastic computational model
SER	Soft error rate
SET	Single-electron transistor
SET	Single event transient
SEU	Single event upset
SFT	Selective fault tolerance
SPR	Signal probability reliability model
SRAM	Static random access memory
TIR	Triple interwoven redundancy
TMR	Triple modular redundancy
VHDL	Very high speed integrated circuit hardware description language
VLSI	Very large scale integration

Abstract

Human have to ponder on the reliability of each tool they have created, and cope with the consequences of the failure caused by these tools. Electronic devices are not exceptions. Downscaling of technology brings with the invention of devices into deep submicrion (DSM) CMOS and even non-CMOS nanometer dimensions. While there are a lot of advantages (smaller size, higher speed, lower power consumption as well as better performance in function diversity), downscaling of geometries also has some drawbacks affecting the system reliability.

Under the circumstances of miniaturization, Single-Event Transient (SET) is much easier to produce unexpected values called *soft errors*. Historically, soft errors were often masked before reaching an output or a storage element. However, aforementioned technological trends such as faster clock rates, smaller device sizes, lower supply voltages, and shallower logic depths are drastically reducing SET masking, and thus are reducing reliability of digital IPs dramatically. International Technology Roadmap for Semiconductors (ITRS) depicts that reliability is emerging as a main threat to the future electronic systems and it should be considered as a very important parameter in the phase of predesign.

Under the foreseeable fact that probability of failure is getting higher, reliable digital IPs will be made of unreliable components. As a matter of fact, a 100% reliability of digital IPs is not only very consuming in performances but also might be impractical. Consequently, what we are pursuing is to maintain a high reliability of digital IPs while keeping extra expense accepted. This aim fosters several issues related to fault-tolerant designs in this thesis.

The most common approach for fault tolerance consists of incorporating redundancy, either static or dynamic or hybrid of the both. Aiming at economical design, a lot of work have been studied in recent years to find a good trade-off between high reliability and better performance. In this context, we try to find feasible methods that could build fault-tolerant digital IPs based on smaller redundancy factors.

Inspired by Pareto principle, identifying and classifying critical constituent blocks of digital IPs should be very meaningful. This thesis presents two new classification criteria regarding the significance of a block with respect to the reliability of a circuit. *Sensitivity* gives the criticality of each block for the circuit reliability and *Eligibility* indicates which priority should be given to each block in a process of adding redundancy. How to acquire the ranking depends on the property of the combinational circuit (structure, individual reliability, etc.). The proposed two concepts provide key information for the designer who looks for efficient fault-tolerant designs.

The aforementioned block grading based on *Sensitivity* and *Eligibility* brings in a straightforward efficient method to select the best subset: Progressive Module Redundancy (PMR). This method proposes to build a fault-tolerant system step by step, that is in a *progressive* way.

It is obvious that we could have different redundant configurations under the same or similar redundant factor. This progressive module redundancy presents a shortcut by avoiding analyses of all the possible redundant architectures exhaustively. It points out a new direction of economical redundant fault-tolerant designs and it is applicable at all the hierarchical design levels (logic gate, arithmetic and processor).

In this thesis, we mainly take into account the representative technique Triple Module Redundancy (TMR) as the reliability improvement technique. A voter is an necessary element in this kind of fault-tolerant architectures. The importance of reliability in majority voter is due to its application in both conventional fault-tolerant design and novel nanoelectronic systems. The property of a voter is therefore a bottleneck since it directly determines the whole performance of a redundant fault-tolerant digital IP (such as a TMR configuration).

Obviously, the efficacy of TMR is to increase the reliability of digital IP. However, TMR sometimes could result in worse reliability than a simplex function module could. A better understanding of functional and signal reliability characteristics of a 3-input majority voter (majority voting in TMR) is studied. We analyze them by utilizing signal probability and boolean difference. It is well known that the acquisition of output signal probabilities is much easier compared with the obtention of output reliability. The results derived in this thesis proclaim the signal probability requirements for inputs of majority voter, and thereby reveal the conditions that TMR technique requires. This study shows the critical importance of error characteristics of majority voter, as used in fault-tolerant designs.

As the flawlessness of majority voter in TMR is not true, we also proposed a fault-tolerant and simple 2-level majority voter structure for TMR. This alternative architecture for majority voter is useful in TMR schemes. The proposed solution is robust to single fault and exceeds those previous ones in terms of reliability. Furthermore, it saves area, power dissipation and propagation delays.

With novel techniques emerging in the future, new algorithms and architectures for reliable digital IPs could be envisaged. More reliable nanoelectronic systems or quantum computers need researches from multidisciplines, such as physics, mathematical modeling, electronic engineering as well as computer architecture, and maybe even more.

French Summary

Introduction

La réduction des dimensions des dispositifs semi-conducteurs selon la Loi de Moore [6] a apporté beaucoup de bénéfices : les systèmes électroniques sont plus petits, plus rapides, moins consommateurs et plus performants. Néanmoins, des phénomènes associés à cette miniaturisation conduisent à une réduction de la fiabilité des circuits. En particulier, le nombre de Single-Event Transient (SET) augmente, augmentant en même temps la probabilité que les circuits produisent des valeurs erronées appelées *soft errors*. Les derniers rapports annuels de l'International Technology Roadmap for Semiconductors (ITRS) montrent bien que la fiabilité est devenue un paramètre très important pour la conception des systèmes électroniques dans les technologies avancées.

Dans cette thèse, nous nous intéressons à la recherche d'architectures fiables pour les circuits logiques. Par "fiable", nous entendons des architectures permettant le masquage des fautes et les rendant de ce fait "tolérantes" à ces fautes.

Les solutions pour la tolérance aux fautes sont basées sur la redondance, d'où le surcoût qui y est associé. La redondance peut être mise en oeuvre de différentes manières : statique ou dynamique, spatiale ou temporelle. Nous menons cette recherche en essayant de minimiser tant que possible le surcoût matériel engendré par le mécanisme de tolérance aux fautes.

Le travail porte principalement sur les solutions de redondance modulaire, mais certaines études développées sont beaucoup plus générales (cf. chapitre 3).

Organisation

Ce document est structuré de la manière suivante :

- Le chapitre 1 donne le cadre général de cette thèse. Nous y trouvons notamment la motivation pour ce travail et le rappel de ses objectifs, à savoir, développer de nouvelles méthodes et de nouvelles architectures pour circuits logiques tolérants aux fautes.
- Le chapitre 2 présente une introduction générale au problème de l'amélioration de la fiabilité des circuits logiques. Nous y trouverons notamment les définitions et métriques de la fiabilité, la motivation pour améliorer la fiabilité ainsi que des méthodes d'estimation de la fiabilité.

- Le chapitre 3 est dédié à l'étude de l'importance qu'un sous-bloc du circuit peut avoir vis-à-vis de la fiabilité du circuit global. Cette étude nous conduit à proposer deux métriques : la sensibilité et l'éligibilité. Ces métriques d'importance de chaque sous-bloc permet de mettre en oeuvre un classement des sous-blocs. Ce classement est la clé de l'efficacité d'une stratégie de durcissement sélectif.
- Le chapitre 4 décrit l'approche d'amélioration de fiabilité proposée dans cette thèse. Afin de maximiser le gain en fiabilité tout en minimisant les surcoûts liés au durcissement, nous proposons un ajout de redondance progressif. L'algorithme d'insertion de redondance ainsi que des études de cas sont décrits en détails dans ce chapitre.
- Le chapitre 5 porte sur la fiabilité de l'arbitre dans un schéma TMR. Nous utilisons les différences booléennes pour analyser le comportement d'un arbitre sujet aux fautes et l'impact que ces fautes peuvent avoir sur l'efficacité de la solution TMR. Les études permettent d'établir les conditions nécessaires à imposer sur la fiabilité de l'arbitre pour que la solution TMR soit source d'amélioration de la fiabilité. Nous présentons également une architecture nouvelle et tolérance aux fautes pour la mise en oeuvre d'une stratégie de vote par majorité.

Analyse de la fiabilité

Introduction

La définition classique de la fiabilité, notée $R(t)$ est la probabilité (en fonction du temps t) que le système fonctionne conformément à ses spécifications pendant une période spécifiée $[t_0, t]$. Elle est déterminée par un paramétre important appelé *taux de défaillance* d'un composant (transistor, porte, bloc, module, etc.), généralement noté λ.

La relation entre la fiabilité $R(t)$, taux de défaillance λ et temps t peut être exprimée par :

$$R(t) = e^{-\lambda t}. \tag{1}$$

Plusieurs métriques peuvent être utilisées pour qualifier la fiabilité des circuits combinatoires.

- Métrique 1 : Probabilité de fonctionnement
  On définit la fiabilité d'un composant (transistor, bloc, module, système, etc) comme étant la probabilité que le composant réalise la fonction souhaitée. La probabilité de fonctionnement est également connue comme *fiabilité fonctionnelle* [7,8].

$$Prob_{comp} = Prob(working) = 1 - Prob(failing) \tag{2}$$

- Métrique 2 : Probabilité de sortie exacte
  On définit la fiabilité d'un circuit par la probabilité que la sortie du circuit soit un bit-vecteur contenant seulement des 0's et 1's corrects. Cette métrique est également connue sous le nom

de *fiabilité du signal* [7, 12].

Dans cette thèse, nous considérons la métrique "fiabilité du signal" et nous nous appuyons sur les méthodes d'analyses décrites en [12].

Supposons qu'un signal binaire x peut véhiculer des informations éventuellement incorrectes. Cela est équivalent à supposer que ce signal peut prendre quatre valeurs logiques différentes : zéro correct(0_c), un correct(1_c), zéro incorrect(0_i), et un incorrect(1_i). Les probabilités d'occurrence de chacune de ces quatre valeurs peuvent être représentées dans une matrice probabilité du signal [**?**] :

$$\begin{bmatrix} P(x = 0_c) & P(x = 1_i) \\ P(x = 0_i) & P(x = 1_c) \end{bmatrix} = \begin{bmatrix} x_0 & x_1 \\ x_2 & x_3 \end{bmatrix} \tag{3}$$

La fiabilité du signal pour x, noté R_x, vient directement de l'expression (4), où $P(.)$ est la fonction de probabilité :

$$R_x = P(x = 0_c) + P(x = 1_c) = x_0 + x_3 \tag{4}$$

Considérons maintenant un bloc numérique effectuant un traitement sur un signal d'entrée x afin de produire un signal de sortie y (cf. Figure 2.5).

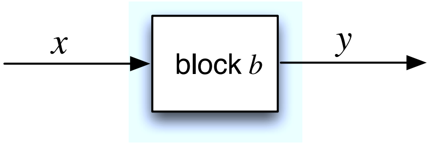


FIGURE 1 – Génération des signaux de sortie y du signal d'entrée x traitée par l'opérateur **block b**.

Supposons que p est la probabilité de défaillance de cet opérateur et que $q = (1 - p)$ est la probabilité qu'il fonctionne correctement. La fiabilité du signal de sortie y peut être facilement obtenue comme suit :

$$R_y = (x_0 + x_3).q + (x_1 + x_2).p \tag{5}$$

L'équation (2.12) montre que lorsque le signal d'entrée est fiable ($x_1 + x_2 = 0$), la fiabilité du signal de sortie est donnée par q, ce qui représente la probabilité de succès du bloc numérique. Ceci signifie que, dans le cas d'entrées exemptes de fautes, la fiabilité du signal de sortie est donnée par la fiabilité intrinsèque de l'opérateur de calcul qui produit ce signal.

L'évaluation de la fiabilité des circuits logiques

Pour les circuits numériques, l'évaluation de la fiabilité est le processus de l'analyse de l'effet et de la propagation des erreurs sur les blocs de base du circuit.

Il existe différentes méthodes d'analyse de la fiabilité reportées dans la littérature [10–17]. Le Tableau 1 présente une synthèse comparative des propriétés de ces techniques en termes de vitesse, précision, besoins en mémoire et l'adaptabilité à des circuits de grande taille (scalability).

TABLE 1 – La comparaison des méthodes représentatives d'évaluation de fiabilité.

Méthodes	Vitesse	Précision	Mémoire	Scalability
PTM	–	précise	trés haute	non
BN	moyenne	précise	haute	non
PGM	moyenne	haute	basse	non
SPR_{MP}	basse	précise	basse	non
PBR	basse	précise	basse	non
SCM	moyenen	adaptative	basse	oui
$BDEC$	moyenne	haute	basse	oui

Amélioration de la fiabilité

La tolérance aux fautes basée sur la redondance

Il existe plusieurs formes de redondance en vue de la tolérance aux fautes [18–20]. La figure 2 montre une synthèse des principales formes de redondance et leur applicabilité aux différents niveaux d'abstraction et granularité de la représentation du système.

Dans ce travail, nous nous intéressons particulièrement à la redondance matérielle mise en oeuvre par réplication de N modules. Cette redondance est connue sous le nom de ”redondance modulaire” ou NMR (N Modular Redundancy).

Dans un schéma NMR, il existe $N + 1$ éléments : N répliques du module et un arbitre (le **VOTER**) qui détermine la valeur de la sortie en fonction de chacune des N sorties produites par les différents modules.

La figure 3 présente un schéma général NMR. Le vote par majorité est l'une des stratégies les plus courantes pour le **VOTER**. Avec l'algorithme de la vote par majorité, N est un nombre impair et la sortie choisie est celle produite par plus de la moitié des répliques, c'est-à-dire, au moins $\frac{N+1}{2}$ modules.

Sous l'hypothèse d'un arbitre idéal (c'est-à-dire, exempt de fautes), la fiabilité d'un système NMR peut être donnée par l'expression (6), où q_M est la fiabilité de chaque module et $\binom{N}{i} = \frac{N!}{i!(N-i)!}$.

$$R_{NMR-IV} = \sum_{i=\lceil N/2 \rceil}^{N} \binom{N}{i} q_M^i \left(1 - q_M\right)^{N-i} \tag{6}$$

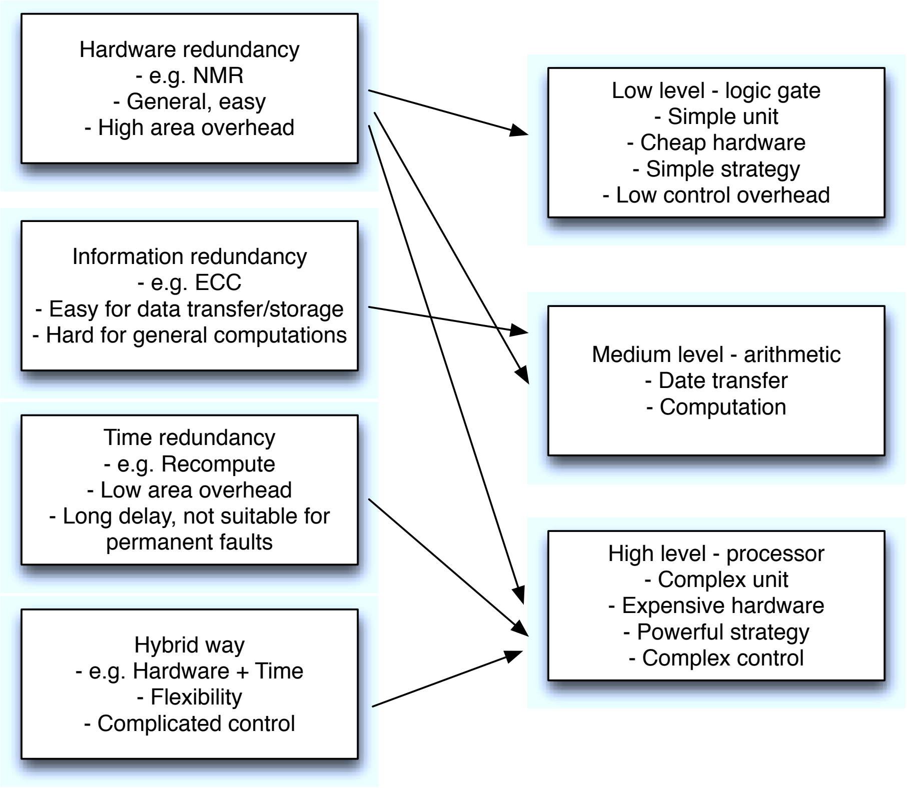


FIGURE 2 – Techniques de tolérance aux fautes [1, 2]).

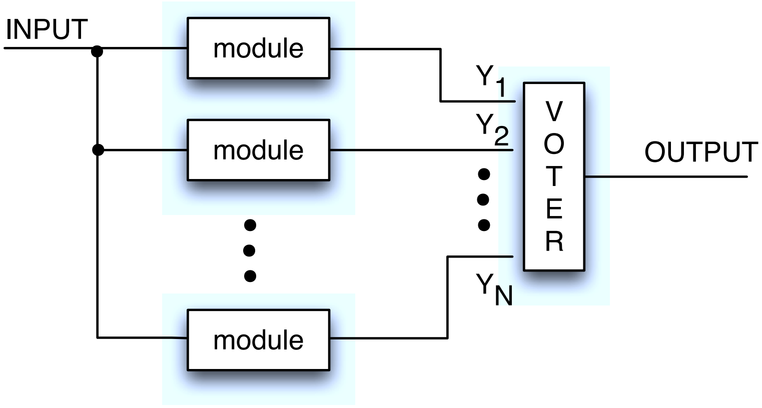


FIGURE 3 – Schéma général pour la NMR.

Dans le cas particulier de la redondance modulaire triple (TMR), la fiabilité est donnée par l'expression (7).

$$R_{TMR-IV} = \left(3q_M^2 - 2q_M^3\right) \tag{7}$$

La figure 4 montre le comportement de la fiabilité d'un système sans redondance et avec TMR conformément à l'expression (7).

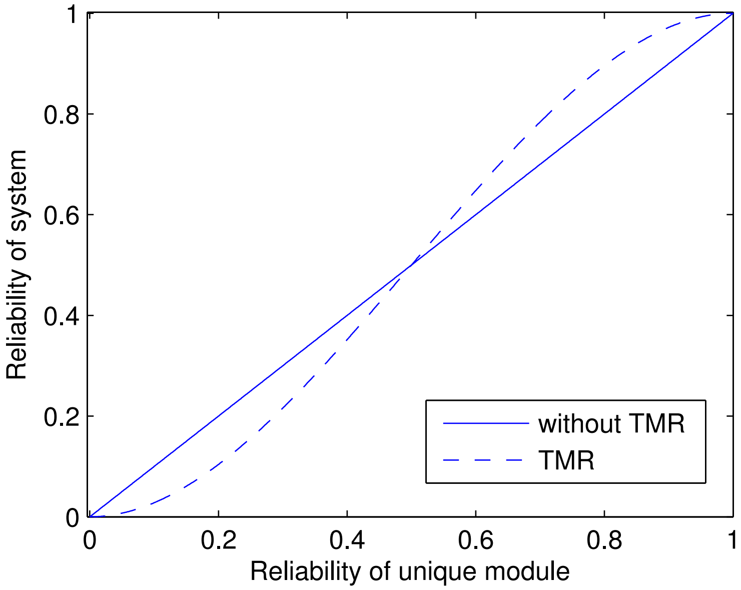


FIGURE 4 – Fiabilité pour deux stratégies de mise en oeuvre : module unique (ligne continue) et TMR (ligne pointillée).

Durcissement sélectif

Dans un circuit numériques constitué de différents sous-blocs numériques, la fiabilité est fonction des fiabilités intrinsèques de chacun des blocs individuels ainsi que de la manière selon laquelle ces sous-blocs sont interconnectés. Il est important de savoir quel rôle chacun des sous-blocs joue dans le circuit vis-à-vis de la fiabilité. Cette information est essentielle pour la mise en oeuvre d'une stratégie de durcissement sélectif. Dans ce sens, nous définissons les concepts de "'sensibilité" et 'éligibilité".

Sensibilité et éligibilité

Considérons un circuit $\mathcal{C}$ avec fiabilité R et constitué de K blocs indépendants b_i. Soit $B = \{b_1, b_2, \cdots, b_K\}$ l'ensemble de tous les blocs K dans $\mathcal{C}$ et $Q = \{q_1, q_2, \cdots, q_K\}$ l'ensemble de leurs fiabilités respectives (q_i est la fiabilité de b_i).

Nous définissons la **sensibilité** de la fiabilité du circuit $\mathcal{C}$ par rapport au bloc b_i dans l'expression (8). Cela correspond à la dérivée partielle de la fonction R par rapport à la variable q_i.

$$s(b_i) = \left| \frac{\partial R}{\partial q_i} \right| \tag{8}$$

Notons $\Theta = \{\theta_1, \theta_2, \cdots, \theta_K\}$ l'ensemble des blocs classés en fonction des valeurs de sensibilité de telle sorte que θ_1 (resp. θ_K) est le bloc dont la sensibilité est maximale (resp. minimale).

Nous définissons l'**éligibilité** d'un bloc b_i, notée $e(b_i)$, comme l'indicateur de l'impact positif qu'a l'amélioration de la fiabilité de ce bloc sur la fiabilité globale du circuit. Notons $\Delta_i = |R_i^* - R|$ le changement de la fiabilité du circuit qui résulte de l'amélioration de la fiabilité q_i.

Les valeurs d'éligibilité de deux blocs b_i, b_j sont alors définies de telle sorte qu'elles respectent $R_i^* > R_j^* \Rightarrow e(b_i) > e(b_j)$. Les valeurs $e(b_i)$ sont des entiers $[1, K]$, où 1 et K représentent, respectivement, les blocs les moins et les plus éligibles.

Structures en cascade

Nous considérons d'abord les blocs $b_i \in B$ tels que définis précédemment. Supposons que ces blocs sont assemblés en une structure en cascade de telle sorte que l'entrée du bloc b_i est donnée par la sortie du bloc b_{i-1} (cf. Figure 5).

Nous pouvons voir que la contribution de la fiabilité individuelle q_i à la fiabilité globale R dépend de la position du bloc b_i dans la structure. La sensibilité $s(b_i)$ dépend aussi de la position du bloc i comme suit :

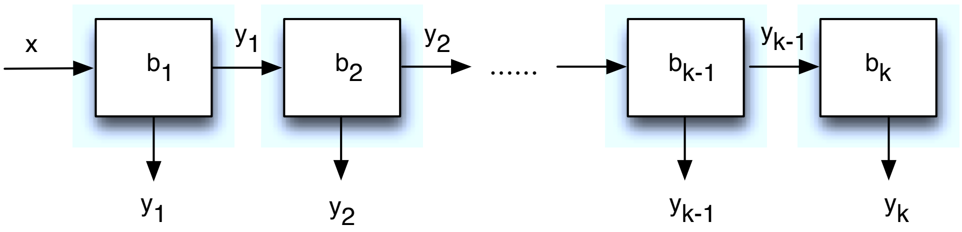


FIGURE 5 – Exemple de structure en cascade.

$$
\begin{aligned}
\left|\frac{\partial R}{\partial q_i}\right| &= (K-i+1)q_i^{K-i}\prod_{i=1}^{K}\prod_{j=1,j\neq i}^{i} q_j \\
&= \frac{K-i+1}{q_i}R \qquad (9)
\end{aligned}
$$

$$
s(b_i) > s(b_j) \Leftrightarrow \frac{K-i+1}{K-j+1} > \frac{q_i}{q_j} \qquad (10)
$$

Structures génériques

Dans le cas de structures génériques, l'expression de la fiabilité du circuit n'est pas évidente. Ainsi, nous avons déterminé la sensibilité et l'éligibilité à partir de simulations.

Une plate-forme de simulation est utilisée pour injecter des fautes [13, 80]. La figure 6 illustre le processus d'injection de fautes et d'analyse. Les résultats produits par le circuit de référence (exempt de fautes) et le circuit dans lequel les fautes ont été injectées sont comparés. Si ces résultats sont différents, il est conclu que les effets de la faute ont été propagés vers les sorties. Sinon, il est conclu que la faute a été masquée.

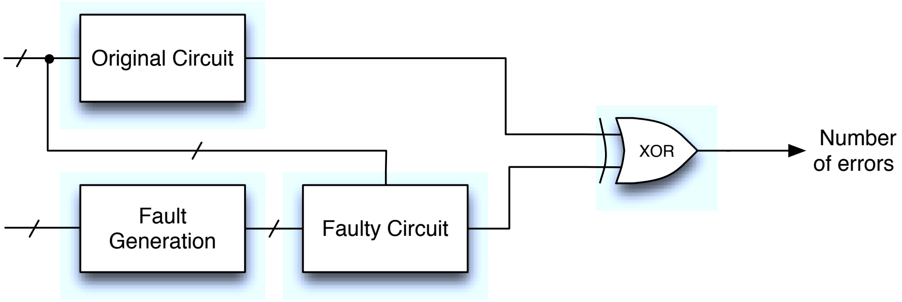


FIGURE 6 – Schéma général pour l'injection de fautes et l'analyse des masquage.

Chaque bloc b_i contribue à un certain nombre de masquages de fautes, noté m_i. Le nombre total de masquages m est la somme des m_i, $m = \sum_{i=1}^{K} m_i$. Chaque m_i est obtenu en calculant le nombre de masquages après l'injection d'une faute simple sur le bloc correspondant b_i. En d'autres termes, en considérant que tous les autres blocs b_j $(1 \leq j \leq K, j \neq i)$ fonctionnent correctement. De cette façon, chaque m_i est directement lié à la sensibilité du bloc b_i. Le classement des blocs en fonction de leurs sensibilités est défini à partir de la relation :

$$s(b_i) > s(b_j) \Leftrightarrow m_i < m_j \tag{11}$$

La figure 7 présente un exemple de circuit benchmark (le circuit C17, de la collection ISCAS-85 [77]). Chaque porte NAND est considérée comme un bloc. Il y a 32 combinaisons logiques possibles pour les entrées et deux configurations considérées pour chaque bloc (sans faute et avec faute). Parmi les 192 configurations qui en découlent, les résultats de m_i sont donnés dans le Tableau 2. Notez que $m5 = m6 = 0$, ce qui signifie que ces deux blocs ne peuvent pas masquer les défauts. En effet, leurs sorties sont également sorties primaires du circuit et nous considérons que seule faute.

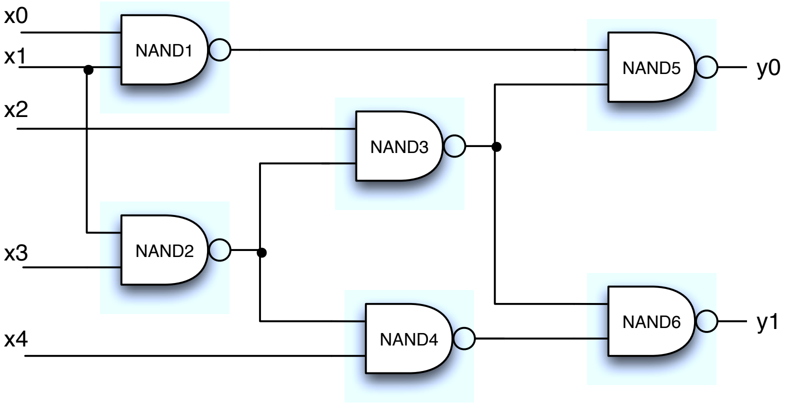


FIGURE 7 – Circuit C17.

TABLE 2 – Classement des sous-blocs du circuit C17 en fonction de la sensibilité.

$b(i)$	m_i
$NAND_1$	12
$NAND_2$	8
$NAND_3$	2
$NAND_4$	12
$NAND_5$	0
$NAND_6$	0

Considérons le même circuit de C17 pour illustrer le classement de blocs en fonction de leurs éligibilités. Sans perte de généralité, supposons que chaque porte NAND est censée avoir $q_i = q = 0.99$ (99%). De même, considérons que l'amélioration de la fiabilité est mise en oeuvre par la technique TMR (Triple Modular Redundancy). Le résultat de l'application de TMR sur chaque porte NAND est présenté dans le Tableau 3.

TABLE 3 – Classement des sous-blocs du circuit C17 en fonction de l'éligibilité.

$b(i)$	R_i^+	$e(i)$
$NAND_1$	95.765%	2
$NAND_2$	95.882%	3
$NAND_3$	96.056%	4
$NAND_4$	95.763%	1
$NAND_5$	96.117%	6
$NAND_6$	96.114%	5

Approche progressive pour la redondance modulaire

Principe général

Nous avons proposé une méthode de durcissement progressif basée sur le classement des blocs dans le circuit. Cette méthode, nommée *Progressive Modular Redundancy*(PMR), a pour objectif de trouver un compromis entre l'amélioration de la fiabilité ΔR et le surcoût matériel ΔC engendré par cette amélioration de la fiabilité. Selon les contraintes de la conception, l'analyse du problème se traduit par :

- déterminer l'architecture de moindre coût qui peut satisfaire une contrainte de fiabilité. Cela signifie réduire ΔC, tout en respectant $R \geq R_{min}$.
- déterminer l'architecture de meilleure fiabilité ne dépassant pas un surcoût matériel donné. Cela signifie maximiser ΔR, tout en respectant $\Delta C \leq \Delta C_{max}$.

Quel que soit le cas considéré ci-dessus, nous proposons d'agir progressivement sur les blocs, en commençant par l'amélioration de la fiabilité d'un seul bloc, puis deux blocs, et ainsi de suite jusqu'à couvrir tous les blocs ou atteindre le coût maximal autorisé. La méthode proposée est décrite dans la Figure 8.

Le Tableau 4 montre comment de nouvelles architectures sont produites selon cette méthode progressive. Le circuit est censé avoir K blocs. La redondance est effectuée en ajoutant une redondance modulaire triple (TMR ou 3MR), ce que nous nommons *Progressive TMR* (PTMR).

Dans ce tableau, les blocs sont classés en fonction de leur poids (w_i) et m désigne les étapes d'exécution de la méthode. Les valeurs dans les cellules représentent les degrés de redondance (1 = pas de redondance, 3 = triple redondance modulaire). L'architecture du circuit correspondant à l'étape

m est obtenue en utilisant TMR sur les blocs de m pour lesquels $w_i \in [K-m+1, K]$.

TABLE 4 – Exécution de l'algorithme PTMR.

	w_i							
m	K	$K-1$	$K-2$	$\cdots$	$K-m$	$\cdots$	2	1
0	1	1	1	1	1	1	1	1
1	3	1	1	1	1	1	1	1
2	3	3	1	1	1	1	1	1
3	3	3	3	1	1	1	1	1
4	3	3	3	3	1	1	1	1
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
K	3	3	3	3	3	3	3	3

L'application de l'approche PTMR au circuit de la Figure 7 produit le Tableau 3. Chaque porte NAND est censée avoir fiabilité $q_i = q = 0.99$ (99%) et coût S_{NAND}. Les contraintes de conception considérées sont : fiabilité minimale requise $R_{req} = 0.97$ (97%) et surcoût en surface maximal accepté $4 \times S_{NAND}$. Bien que le nombre total d'architectures générant la même surface soit $C_6^2 = 15$, cette méthode permet d'identifier le meilleure d'entre elles sans avoir à toutes les tester.

TABLE 5 – Résultats de l'application de l'approche *PTMR* sur le circuit C17.

Steps (m)	Architecture	Reliability	Area Cost
0	1-1-1-1-1-1	95.20%	$6S_{NAND}$
1	3-1-1-1-1-1	96.12%	$8S_{NAND} + S_V$
2	3-3-1-1-1-1	97.05%	$10S_{NAND} + 2S_V$

Fiabilité de l'arbitre dans un schéma TMR

Dans les schémas TMR, l'arbitre est souvent supposé parfait, c'est-à-dire, exempt de fautes. Or, la prise en compte de la fiabilité réelle de ce composant est essentielle pour déterminer l'efficacité de cette approche. Dans le cas de vote par majorité, si nous considérons que l'arbitre a une fiabilité R_v, l'équation (7) est modifiée pour donner :

$$R_{NMR-FV} = R_v \cdot \sum_{i=\lceil N/2 \rceil}^{N} \binom{N}{i} q_M^i \left(1-q_M\right)^{N-i} \tag{12}$$

Avec un arbitre imparfait, la condition pour que le système TMR apporte un gain en fiabilité se traduit par :

$$R_{NMR-FV} > q_M \quad or \quad \frac{R_{NMR-FV}}{q_M} > 1 \tag{13}$$

Ainsi, la fiabilité minimale pour l'arbitre est :

$$R_{V_{min}} = \frac{1}{\sum_{i=\lceil N/2 \rceil}^{N} \binom{N}{i} q_M^{i-1} \left(1 - q_M\right)^{N-i}} \tag{14}$$

La figure 9 montre l'exigence de fiabilité minimum pour un arbitre dans un schéma de redondance modulaire d'ordre 3 (TMR) et 5 (5MR).

Arbitre tolérant aux fautes

Toutes les combinaisons possibles d'entrées dans un arbitre basé sur le vote majoritaire sont présentées dans le Tableau 6. L'expression booléenne correspondante est donnée dans (15). La forme simplifiée de cette équation, donnée dans l'équation 16, permet d'obtenir différentes structures de mise en oeuvre, comme nous pouvons voir dans les Figures 10(a) et 10(b).

TABLE 6 – Table de vérité pour un arbitre basé sur le vote majoritaire.

A	B	C	V
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

$$\begin{aligned} V &= ABC + AB\overline{C} + \overline{A}BC + A\overline{B}C && (15) \\ &= AB + BC + AC && (16) \end{aligned}$$

Considérons la structure d'arbitre AND-OR dans la Figure 10(a). Nous pouvons constater qu'une faute dans une porte **AND** ou **OR** peut générer une valeur erronée dans la sortie de cet arbitre. Par exemple, si $A = B = C = 0$ et une faute simple a lieu dans $S1$, la sortie sera $V = 1$, alors que la valeur attendue est $V = 0$. Par conséquent, cette structure n'est pas tolérante aux fautes.

Dans [92], Kshirsagar a proposé une structure d'arbitre tolérante aux fautes (NTFVC) (cf. Figure 11).

Nous proposons une structure nouvelle pour arbitre dans schéma TMR avec stratégie de vote par majorité capable de tolérer des fautes (cf. Figure 12) et plus compacte que celle de Kshirsagar. Les résultats d'analyse de fiabilité obtenus avec l'algorithme SPR [12, 46] sont présentés dans la Figure 14. L'arbitre proposé a été utilisé dans plusieurs circuits (cf. exemple de half-adder dans la Figure 13). Les résultats des comparaisons par rapport à des paramétres différents en fonction de différents électeurs de la technique TMR sont présentés dans le Tableau 7.

TABLE 7 – Résultats synthétisés dans ASIC (RTL Compiler).

Comparison	Classic	NFTVC	Proposed
Instances	25	31	25
Area	31	40	32
Power μw	2.35	3.75	2.85

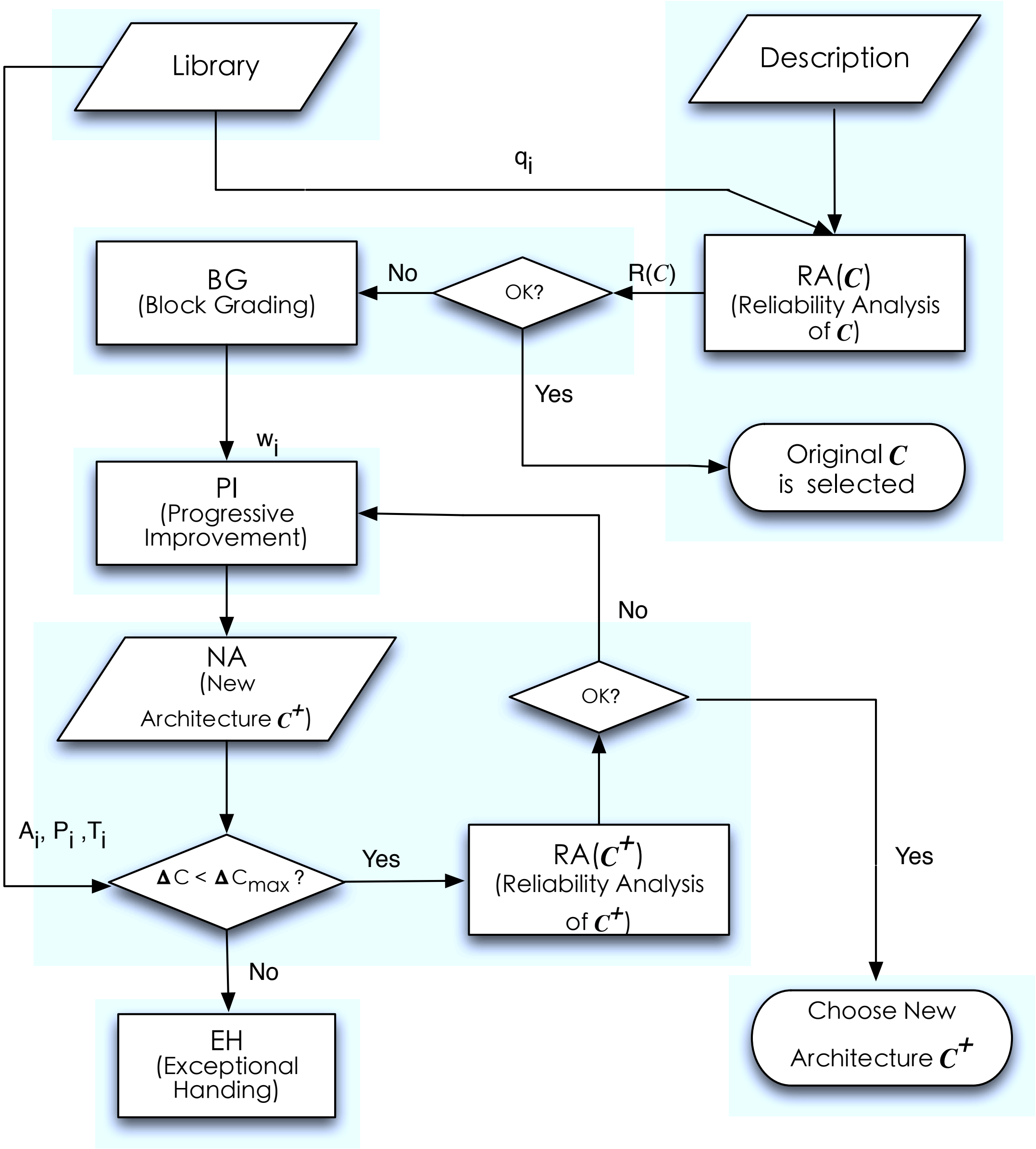


FIGURE 8 – Workflow pour la méthode proposée.

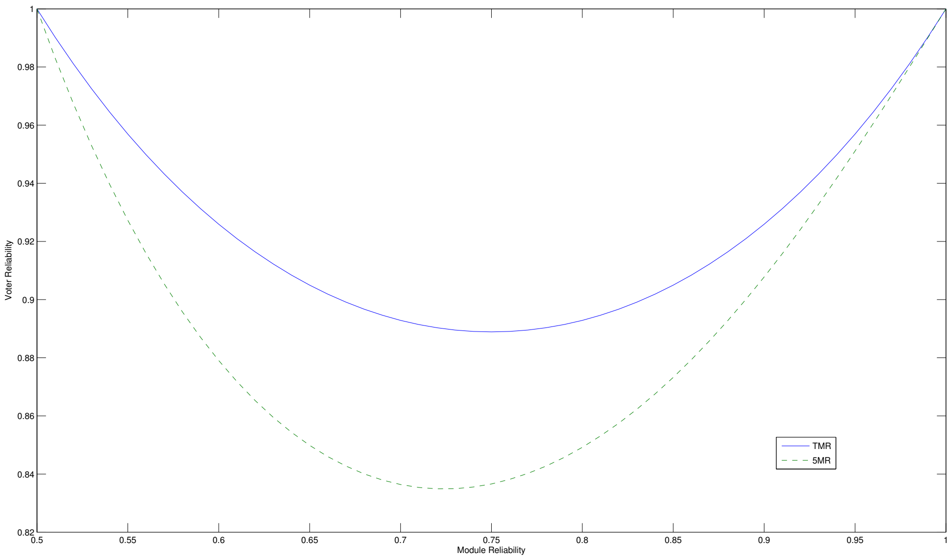
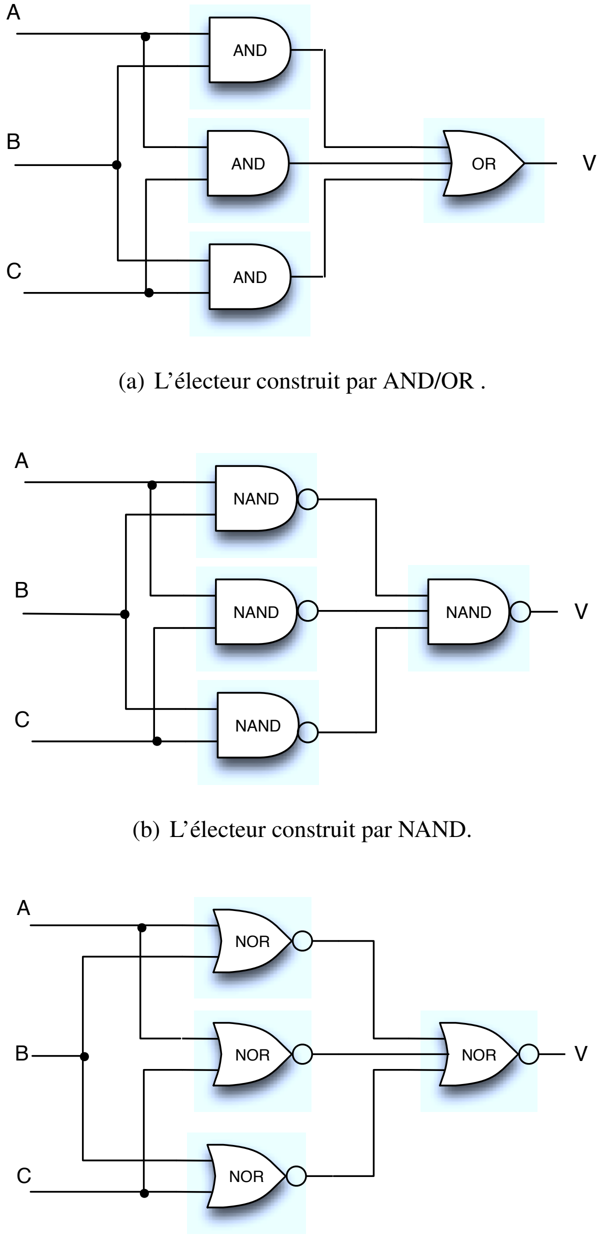


FIGURE 9 – La fiabilité des électeurs requis dans le systéme de NMR.



(a) L'électeur construit par AND/OR .

(b) L'électeur construit par NAND.

(c) L'électeur construit par NOR.

FIGURE 10 – Schémas classiques pour arbitre dans schéma TMR avec stratégie de vote par majorité.

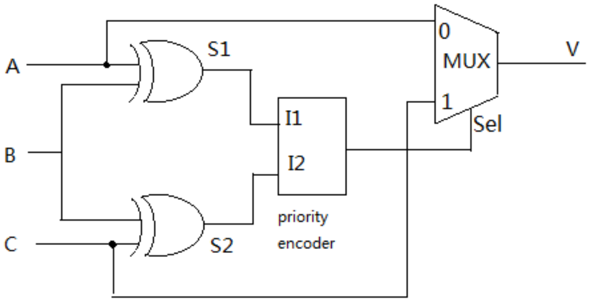


FIGURE 11 – Structure d'arbitre tolérant aux fautes (NFTVC) [92].

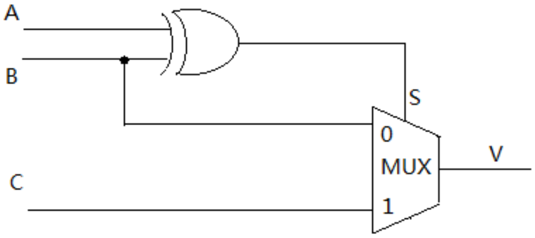


FIGURE 12 – Schéma proposé pour l'arbitre tolérant aux fautes.

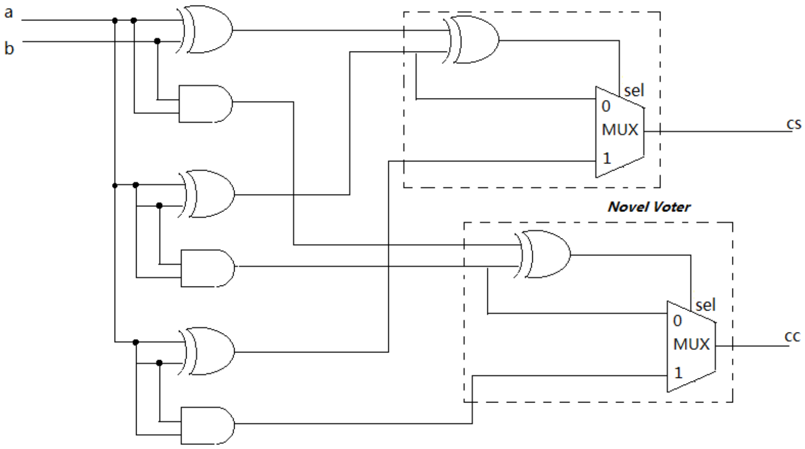


FIGURE 13 – Exemple d'utilisation de l'arbitre tolérant aux fautes dans un demi-additionneur.

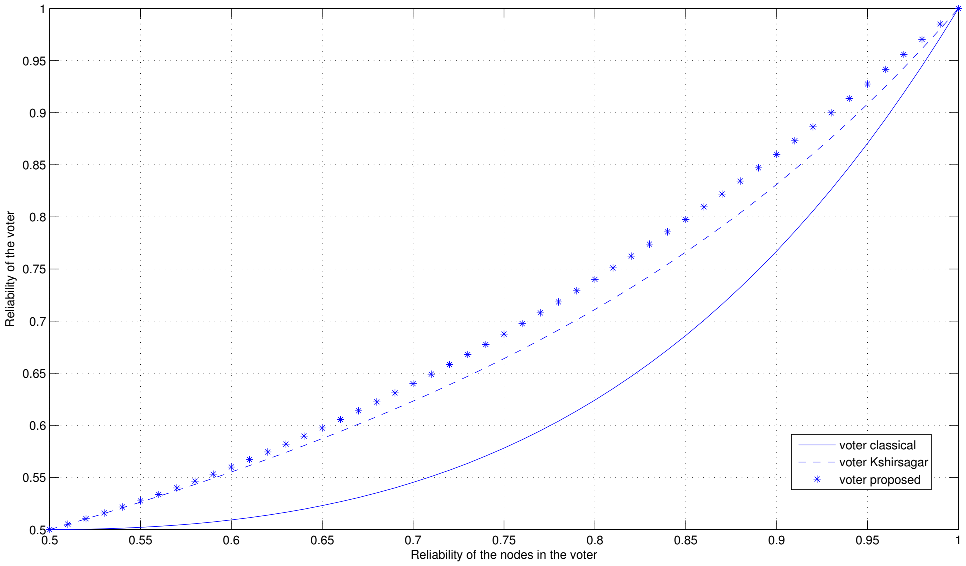


FIGURE 14 – Courbes de fiabilité pour les classiques (ligne continue), NFTC (ligne pointillée) et l'électeur proposée (ligne pointillée).

Chapter 1

Introduction

1.1 Motivations

Intellectual property (IPs) in semiconductors refers to pre-designed function modules which are used in Application- Specific Integrated Circuit (ASIC), Field-Programmable Gate Array (FPGA) or Programmable Logic Device (PLD), etc. In accordance with Moore's Law, the number of transistors per chip should double every 18 to 24 months. The smaller dimensions enabled the development of faster, more compact, and more powerful electronic devices [21]. However, the complexity to design and to produce such devices has also grown over the years, emerging as a drawback to maintain their current evolution rate. The operation frequency is expected to increase up to 12 GHz, and a single chip will contain over 12 billion transistors in 2020 according to ITRS [22]. Therefore, transistors are now implemented into the nanoscale and known as nano electronic devices.

As a result of scaling, the amount of defects as well as the number of transient errors in electronic circuits are expected to increase, and it becomes a major concern in Deep-SubMicron technologies (DSM). Some of these are manufacturing imprecision, improved susceptibility to environmental factors and physical parameters variability [23]. This caused two significant challenges. The first is related to the yield that can be achieved during the manufacturing process of digital IPs, known as design-for-yield which is an entire research field. The second challenge is that although digital IPs have been proved to work correctly, they must continue to operate reliably even they are increasingly sensitive to various kinds of perturbations, which are known as SEUs (Single-event Upsets). Reduction in the reliability of digital IPs is one of the main threats in semiconductors industry [24], which leads to paradigm shift toward design-for-reliability [25]. Since reliability is a crucial feature of designs, new techniques of reliability improvement need to be developed to overcome the expected unreliability of IPs implemented on such new DSM technologies.

SEUs in digital IPs have been ignored for a long time because SETs (Single-event transient, also known as soft errors [21,26]) had only minor influence in earlier $0.35\mu m$ and larger technologies [27]. On the other hand, there are three intrinsic properties of logic circuit itself: First, *logical masking*.

A glitch might not propagate to a memory cell because of a gate on the path not being sensitized to facilitate glitch propagation. Second, *electrical masking*. A generated glitch may get attenuated because of the delays of the gates on the path to the output [28]. Third, *latching-window masking*. A glitch that reaches the primary output might not still cause an error because of the latch not being open.

However, in Deep SubMicron (DSM) technologies, due to the decreasing number of gates in a pipeline stage, logical masking as well as electrical masking have been decreasing for new technology generations. Electrical masking has also been decreasing due to the reduction in node capacitances and supply voltages every generation. Furthermore, increasing clock frequencies have reduced the time window in which latches are not accepting data, thereby also reduced latch-widow masking. As above statement, the soft error rate (SER) in digital IPs raised 9 orders of magnitude from 1992 to 2011, when it is equal to the SER of unprotected memory elements [29]. Needless to say, under the DSM technologies, soft errors in digital IPs are not optimistic and are becoming dominant in the overall soft error rate.

There are various fault-tolerant techniques to increase the robustness of the circuits, which are mostly based on the concept of redundancy [9, 30–32]. These techniques have been applied at different levels of granularity, such as gate level, logic block level, logic function level, unit level, etc. Moreover, these techniques were historically targeted to mission critical systems, for example, in medical, spatial and military applications. With the expected reduction in the reliability of electronic devices, they have been considered in many general applications, such as consumer electronics.

The usage of aforementioned fault-tolerant techniques can help to enhance reliability of circuits, but they normally result in great area, time or power overheads. Moreover, too much redundancy may reduce the yield since a larger-area circuit is expected to have a larger number of defects. *Pareto principle* [33] states that, for many events, roughly 80% of the effects come from 20% of the causes. Consequently, successful designs must have the optimal amount of redundancy to be added. In order to develop this optimal redundancy configuration, we try to implement a partial redundancy where the most eligible parts of the circuits are protected, meanwhile the best improvement in reliability can be achieved.

1.2 Objectives

The objective of this thesis is to develop new methods and architectures for logic function IPs bringing with optimized trade-offs which can improve reliability and give consideration to classical performance parameters (power consumption, area and time) simultaneously. Consequently, the question we try to answer in this thesis is how to make a judicious redundancy configuration that could result in optimal reliability improvements. It includes research on finding most suitable extent and location of implementing the hardware redundancy under a given design criteria.

This dissertation mainly addresses the following issues.

- As it is possible to build systems with unreliable components, how to fabricate more reliable systems with the probabilistic nature of the component's individual reliability?
- Propose and verify new fault-tolerant architectures that enable improvement in reliability, with respect to those existing ones.
- As a realistic fault model is relevant to further evaluation of reliability, fault models and reliability evaluation methods need to be analyzed and compared.
- Since the common approach for constructing fault-tolerant architectures consists of incorporating redundancy, the feasibility of designing reliable architectures by using economical/small redundancy factors is worthy to be discussed.
- The importance of reliability in majority voter is due to its application in fault-tolerant design. Error characteristics of majority voter are discussed to reveal the conditions that Triple Module Redundancy (TMR) technique requires.

1.3 Organization of the thesis

The dissertation is organized as follows:

- Chapter 2 introduces the basic concepts related to reliability improvement in logic circuits: definition and metrics of reliability, motivations for enhancing reliability, ways and methods of evaluating reliability, etc. A brief review of fault-tolerant techniques is presented, including types of redundancy and general methods of fault-tolerant system design. Reliability improvement of digital IPs by hardware redundancy is mainly discussed, concerning module redundancy (especially TMR), trade-offs between reliability requirements and performance degradation etc. As TMR becomes a favorable solution because of its straightforward implementation and also reliability gains, state of the art in alleviating the performance degradation of TMR designs are addressed, including Selective TMR, Partitioning TMR, Non-uniform TMR, Automatic-insertion of TMR, Selective fault tolerance, etc.
- Chapter 3 is about two proposed concepts that describe the significance of the constituent blocks of digital IPs: sensitivity and eligibility. This inherent significance can identify the most critical/eligible blocks of digital IPs according to different design criteria. Afterwards, reliability analysis and improvement based on these two concepts are discussed. The concepts are firstly implemented on the cascade structure mathematically and then on generalized circuit structures by simulations. Applications based on these concepts are also presented.
- Chapter 4 presents reliability improvement by redundancy management. In order to maximize the reliability while minimizing redundancy adding, the progressive manner of redundancy addition (Progressive Modular Redundancy, PMR) is proposed. The algorithm, workflow and methodology are presented in details. Furthermore, considering that not all bits of information

have the same degree of importance, a more aggressive way of fault-tolerance is proposed in some practical applications with related to the bit significance. Finally, results obtained from the proposed method are compared with the state-of-the-art techniques.

– Chapter 5 analyzes the error characteristics of majority voter. The importance of reliability in majority voter is due to its application in both conventional fault-tolerant design and novel nanoelectronic systems. A better understanding of signal probability, functional/signal reliability and error bound of majority voter is discussed here. These parameters are analyzed by boolean difference. The equations derived here present the characteristics of error propagations in majority voter, and reveal the conditions that TMR technique requires. The results show the critical importance of error characteristics of majority voter, as used in fault-tolerant designs. Finally, a simple fault-tolerant voter structure is also proposed which avoids voter introducing new locations wherein faults may occur.
– Chapter 6 presents concluding remarks, reviews of this dissertation and further perspectives are also discussed.

Chapter 2

Reliability Improvement Techniques

2.1 Introduction

Reliability is actually as important as other factors such as power consumption, area overhead and speed in nanometer electronic designs. The ITRS explicitly calls for a fresh look on nano-architecture with emphasis on fault and defect tolerance [34]. With technology scaling and complexity of the designs increasing, digital IPs become more fault-prone and there is a serious menace to the continuous development of the integrated circuits industry. These problems have motivated a lot of researches on reliability improvement. However, cost penalty concomitant with the reliability improvement makes the task not an easy work.

This chapter presents general concepts related to reliability and its enhancement in digital IPs and the scope of the work is also defined. Preliminaries and prevalent models for reliability evaluation are introduced. Most recent progresses about reliability evaluation methods are reviewed. Comparison and comments about them are also presented. An overview of redundancy-based fault-tolerant schemes regarding to nano-electronic technologies is presented as well as a detailed discussion and comparison concerning the methodologies for reliability improvement designs.

2.2 Preliminaries of reliability

2.2.1 Defect, fault, error and failure

An electronic system can be in one of the five states as shown in Figure 2.1: Ideal, Defective, Faulty, Erroneous, or Failed [35]. These states are explained as below:

- Defect: A defect is any imperfection on the wafer.
- Fault: A fault is an erroneous state of the system, either hardware of software.
- Error: An error is the manifestation of a fault.
- Failure: A failure causes the system performance deviate from its specified performance.

Some distinctions between these definitions are as follows:

- Defects and Faults: A defect is any imperfection on the wafer, but only those defects that actually affect the circuit operation are called faults.
- Faults and Errors: If a fault is actually exercised, it may contaminate the data flowing within the system, causing errors, but not necessarily.
- Errors and Failures: Errors may or may not cause the affected circuits to failures. It doesn't necessarily have a catastrophe: when an error is encountered during the operation of a system, it will cause a failure.

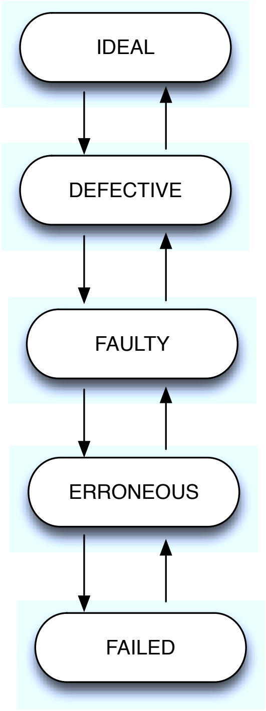


Figure 2.1: System States and state transitions in the multi-level model of reliability.

The circuit systems move from one state to another as a result of deviations and remedies. Deviations are the events that take the system to a less desirable state, while remedies are measures that enable a system to make the transition to a higher state. In addition, faults are usually characterized based on its duration. That is, a fault is said to be *permanent* if it continues to exist until it can be repaired. An *intermittent fault* is one that happens and ends at a frequency that can be characterized. A *transient fault*, which is mainly focused on in this thesis, is the fault that occurs and disappears at an unknown frequency. It is caused by alpha or neutron particles, electrostatic discharge, thermal noise, crosstalk, quantum mechanical effects, etc.

2.2.1.1 SEUs and SETs under DSM technologies

The transient faults we are interested in this thesis are single event upsets (SEUs), in which radiated particles cause the state of a storage element changed. They can occur in storage elements of digital IPs like latches, flip-flops as well as memory cells (SRAM, DRAM). SEUs may happen in three ways, as shown in Figure 2.2.

- Case 1: A particle attacks an internal node of a latch or flip-flop directly. In this case, it may produce an inversion of the element state.
- Case 2: A particle hits the combinational parts of a digital circuit, for example a logic gate. In this case, the particle strike causes a glitch (glitches) in the output voltage of a logic gate, which is called single event transients (SETs). SETs may propagate through the combinational part to a latch or a flip-flop, thus turn into a SEU.
- Case 3: A particle attacks the control signal such as clock signal. This will result in early or late edge in the clock signal and data will not be latched correctly.

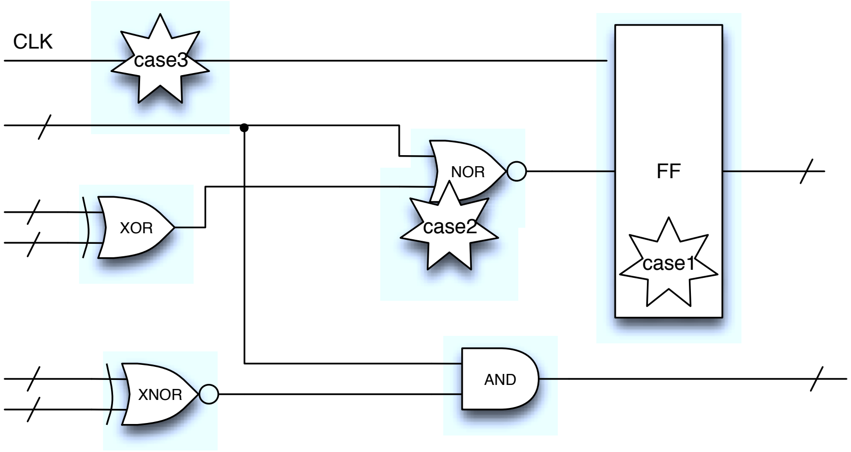


Figure 2.2: Mechanism of SEUs in digital IPs.

As technology size shrinks, digital IPs are becoming more susceptible to SEUs and SETs. SEUs and SETs therefore become the major reliability concerns in deep submicron technologies. Since SETs depend on the propagation time and also combinational logic, the probability to latch a SET can only be evaluated very late in the design process. However, it is necessary to know the potential impact of SEUs and SETs at earlier stage in design flow. Consequently, it requires a model to make it possible to analyze the faults.

2.2.1.2 Modeling of faults

To deal with faults, a model is needed to simulate their effects. A fault model is a logical abstraction that describes the functional effect of the physical defect. Fault modeling can be made at different levels, from the lowest physical geometrical level and then gate level and to the highest which is the functional level. The lower the fault models are, the more accuracy could be obtained while the computation is more complex. For example, Figure 2.3 shows the double exponential model for current pulses in analog domain which is proposed in [36]. Although this fault model remains at low level, it can be used to implement fault injections on target nodes in the high-level description of analog blocks. In order to simplify the simulations and reduce the fault injection experiment duration, a more practical model is then proposed in [37] with more parameters that derived from the classical double exponential model.

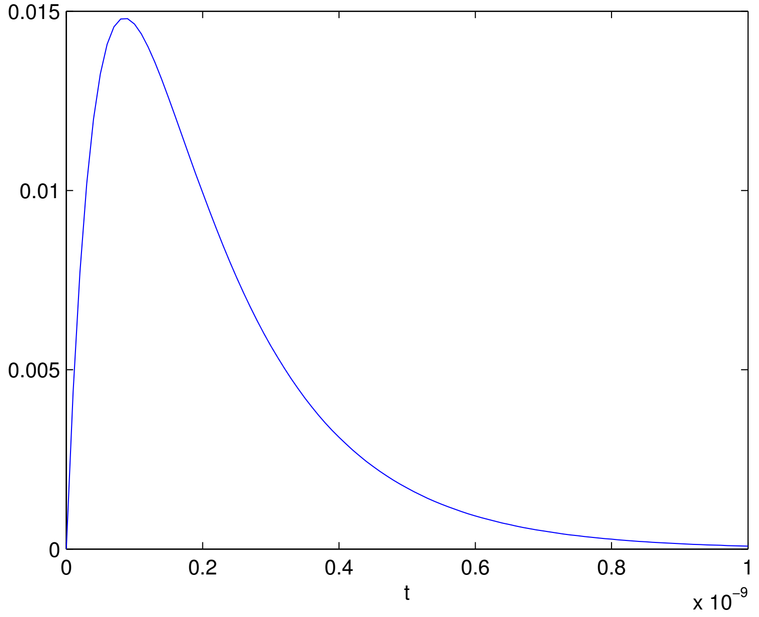


Figure 2.3: Double exponential fault model.

Fault injection techniques can be classified into three main categories [38].

- *physical fault injection*: it introduces faults directly to the hardware of the target system by distributing the working environment of hardware. Like electromagnetic interferences, heavy ion radiation, etc. It is most close to the real fault environment but the device for injection is expensive. And it also needs long design cycles.
- *software fault injection*: it refers to changing the memory and registers to emulate the consequences of hardware faults. The flaw is that injection locations are limited and time resolution is poor.
- *simulated fault injection*: It imitates the faults of the system based on the use of hardware description languages. It is favorable because it can provide check results at an early time in the design process and simulations could be implemented on RTL as well as gate level.

Simulation-based fault injection is widely adopted for its flexibility, visibility and nonintrusiveness. In digital domains, the consequences of SEUs and SETs could be modeled by one of more bit-flip(s) at the functional level. As stated above, it is realized by modifying the initial description of circuits, i.e., HDL codes. To make these modifications, there are two approaches named *saboteurs* and *mutants* respectively. The first is to add blocks (the saboteurs) between the existing blocks. Such modifications are easy both in concept and implementation and are adopted by a lot of applications. In this thesis, we will use this fault model for injecting faults to digital IPs and then analyze the circuit reliability.

If we need to inject higher-level errors that are behavioral errors, we need to modify signals within the initial blocks, for example, values of memorized signals or variables. In such cases, the modified description of the block is called *mutant*. The injections of faults in the high-level is more difficult but more powerful. Examples of sabotaged and mutated VHDL codes could be found in [39] and [40]. A complete framework for Verilog-based fault injection and evaluation is presented in [38]. A novel simulation fault injection method for the dependability analysis of complex SoCs using 32-nm technology is proposed in [41] (named SyFI, which augmented the SystemC simulator kernel so that fault injection experiments can be performed conveniently). It is applied at the system description level, as opposed to the lower, flattened RT level, in order to reduce simulation time and storage space.

2.2.2 Measures and metrics related to reliability

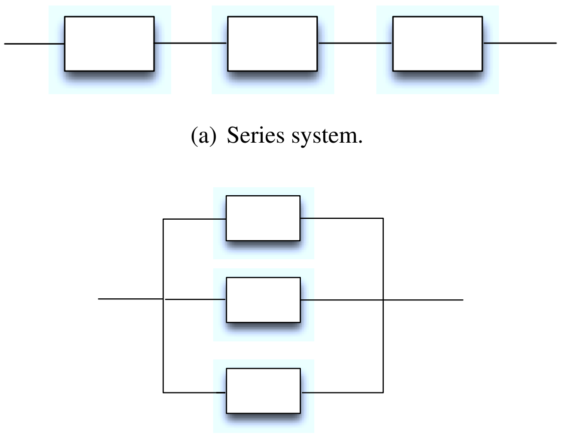
A reliability measure is a mathematical representation of the circuit reliability characteristics. The conventional definition of reliability, denoted by $R(t)$, is the probability (as a function of the time t) that the system will execute its specified function continuously in a given time interval $[t_0, t]$. It is determined by an important parameter called the *failure rate* of a component (transistor, gate, block, module, etc.). Failure rate (usually denoted by λ), is the rate at which an individual component suffers faults. This parameter depends on the current age of the component, any physical shocks it suffers and the technology. As we stated in the first chapter, combinational circuits have logical, electrical and temporal masking properties. Therefore, *failure rate* in combinational circuits is also dependent on its capacity to mask these faults. This ability reduces the probability of propagation and further storage of the faults in the sequential elements of the circuit, characterizing a failure.

The relationship between reliability $R(t)$ and time t that has been widely accepted is the exponential function, i.e.,

$$R(t) = e^{-\lambda t}. \tag{2.1}$$

Now we consider reliability of a digital IPs in most canonical structures, series and parallel, as shown in Figure 2.4.

A system consists of N *series* components, wherein the failure of a component will cause the system failure, reliability is given by



(a) Series system.

(b) Parallel system.

Figure 2.4: Series and parallel systems

$$R_s(t) = R_1(t) \cdot R_2(t) \cdot R_3(t) \cdots R_N(t) \tag{2.2}$$

where $R(t)$ is the system reliability and $R_i(t)$, $i = 1, 2, ...N$, are the component reliabilities. If we consider the components have the exponential failure densities, then

$$R_s(t) = exp\left(-\sum_{i=1}^{N} \lambda_i t\right) \tag{2.3}$$

A system consists of N *parallel* components, wherein a system failure requires the fails of all the constituent modules, reliability is given by

$$R_p(t) = 1 - \prod_{i=1}^{N}(1 - R_i(t)) \tag{2.4}$$

Since $R(t)$ depends on the time of system or component operation and thus is variable, it is not suitable in practical use. A metric very closely related to reliability is known as *Mean time to failure (MTTF)*. It is defined as the expected time that a system is functional until the first failure occurs. As certain types of components suffer no aging and have a failure rate that is constant over time, in most calculations of reliability, we also use a constant failure rate because of the simplified derivations. With $\lambda(t) = \lambda$, *MTTF* can be derived as follow.

$$MTTF = \int_0^\infty \quad e^{-\lambda t} dt = \frac{1}{\lambda}. \tag{2.5}$$

Mean Time Between Failure, MTBF is the average time between two consecutive failures. It has the same value as *MTTF* if the *Mean time to Repair, (MTTR)* is the same. Here *MTTR* denotes the average time needed to repair the first failure. If we use μ to express the repair rate, then *MTTR* is defined as below.

$$MTTR = \int_0^{\infty} \quad e^{-\mu t} dt = \frac{1}{\mu}. \tag{2.6}$$

Therefore, we have

$$MTBF = MTTF + MTTR \tag{2.7}$$

Furthermore, *Availability* is the probability of a system operates correctly at the instant t. It means a system can be either available or unavailable (i.e., in repair) and it is expressed as

$$A(t) = \frac{MTTF}{MTBF} \tag{2.8}$$

Several metrics can be used to measure the reliability of combinational circuits. Which metric is chosen for reliability evaluation is closely related with the specific applications.

– Metric 1: Probability of working
  We define the reliability of a component (transistor, gate, block, module, system etc.) as the probability that the component realizes the desired function, i.e., it has a correct output for normal inputs. It is also known as *functional reliability* [7, 8].

$$Prob_{comp} = Prob(working) = 1 - Prob(failing) \tag{2.9}$$

– Metric 2: Probability of the output being correct
  We define the reliability of a given circuit as the probability of the circuit's output is correct even in presence of errors. That is, the output is a bit-vector containing only correct 0's and correct 1's. It is also known as *signal reliability* which means the probability that a given signal carries a correct value [7, 12]. Notice that for a particular circuits, signal reliability is always greater or equal to the functional reliability.
  Assume that a binary signal x can carry incorrect information is equivalent to assume that it can take four different values: correct zero (0_c), correct one (1_c), incorrect zero (0_i), and incorrect one (1_i). Then, the probabilities for occurrence of each one of these four values are represented in matrices as shown bellow:

$$\begin{bmatrix} P(x = 0_c) & P(x = 1_i) \\ P(x = 0_i) & P(x = 1_c) \end{bmatrix} = \begin{bmatrix} x_0 & x_1 \\ x_2 & x_3 \end{bmatrix} \tag{2.10}$$

  The signal reliability for x, noted R_x, comes directly from expression (2.11), where $P(.)$ stands for the probability function:

$$R_x = P(x = 0_c) + P(x = 1_c) = x_0 + x_3 \tag{2.11}$$

Now let us consider a digital block performing a function on a signal x in order to produce a signal y. Assume that p is the probability that this operator fails, and that $q = (1 - p)$ is the probability it works correctly. We can easily obtain the reliability of y as:

$$R_y = (x_0 + x_3).q + (x_1 + x_2).p \tag{2.12}$$

Equation (2.12) shows that when the input signal is reliable, the output signal reliability is given by q which represents the probability of success in the digital block. It is due to $x_1 + x_2 = 0$. This implies that for fault-free inputs, the reliability of the output signal is given by the inherent reliability of the block that produces this signal.

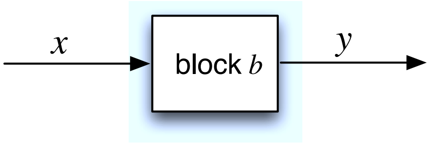


Figure 2.5: Generation of the output signal y from the input signal x processed by the digital block b.

- Metric 3: Relative Error

  In definition 1 and 2, nothing is considered related to weights, i.e. an erroneous value in any bit (from LSB to MSB) of the output is equally weighted. However, some faults may result in larger errors in a most significant bit. It is therefore common to use another metric for reliability measures by using *relative error* (RE). Relative error is widely used in reliability evaluation of Analog-Digital-Converters (ADCs) [42].

$$RE = \frac{expected \quad output \quad - \quad actual \quad output}{expected \quad output} \tag{2.13}$$

- Metric 4: Effective reliability

  Based on the concept of *relative error*, a more practical metric is proposed in [43, 44]. which analyzes the circuit reliability not only circuit structure but also the target application.

  Let $\mathbf{y}_i = b_{M-1}b_{M-2}\cdots b_1b_0$ be defined as a vector of M bits that represents the output of a circuit. In this case, bit b_0 stands for the LSB (Least Significant Bit). Also, let us define the reliability of bit b_i as q_i. For a such circuit, the nominal reliability R_{nom} can be evaluated by :

$$R_{nom} = \prod_{i=0}^{M-1} q_i \tag{2.14}$$

For a large number of applications, consequences of some errors can be more critical than others, but this phenomenon is not included in (2.14).

Unlike nominal reliability, the concept of effective reliability given in (2.15) allows to classify errors into two categories: critical and noncritical errors.

$$R_{eff} \quad = \quad R_{nom} + R_{ack} \tag{2.15}$$

Noncritical errors are defined as errors that do not compromise the circuit performance, i.e., errors that can be acceptable by the system. The first term in (2.15) stands for the nominal reliability concept and the second one, R_{ack}, stands for the probability of the faulty output be considered noncritical. The value of R_{ack} could be evaluated according to different quality metrics as stated in [43, 44].

2.3 Reliability evaluation of logic circuits

As stated before, the effects of noise at the electrical level can usually be modeled by a probability distribution about the nominal voltage, instead of a single number. A Gaussian distribution have been widely adopted to approximate the noise distribution. However, reliability analysis of combinational circuits is a problem in the digital domain, while the noise distribution modeled by Gaussian distribution is in the continuous domain and is therefore not suitable. Without loss of generality, the average probability that the low and high voltages exceeds the noise margin is used to estimate the gate failure probability, expressed by ϵ.

For digital IPs, reliability evaluation refers to the process of analyzing the effect and propagation of errors on the basic blocks of the circuit. The classical model for errors due to noise in digital IPs was first introduced by von Neumann in 1956 [9]. Noise at a logic gate is modeled as a binary symmetric channel (BSC), with a crossover probability ϵ, i.e., the gate output to toggle is symmetrical from 0 to 1 or 1 to 0 with the same probability ϵ. Here $\epsilon \in [0, 0.5]$ because it is unrealistic for a gate have $\epsilon > 0.5$, it would mean that the output of the gate is more likely to be faulty than correct. In such a case, adding a NOT gate at the output would make the combination of the two gates more reliable. Note that gates are assumed to fail independently to each other. Although it might be an ideal assumption, since effects of noise are potentially and randomly localized and correlated, this assumption helps to simplify reliability analysis while still provides meaningful insights on circuit reliability analysis.

The increased importance of reliability strongly suggests that reliability needs to be included as the fourth optimization pillar of forthcoming electronic design automation (EDA) tools, along with the well-known triplet parameters area, power and speed. However, reliability analysis is a NP-hard problem since the combinations of input signals increase exponentially.

The methods to evaluate reliability of digital IPs can generally be divided into three groups:

- analytical based.
- simulation based.

– hybrid of simulation and analytical ways.

2.3.1 Analytical-based way

A number of recent works proposed analytical approaches for evaluating circuit reliability. In [10], Probabilistic transfer matrices (PTM) is proposed to model the soft errors in the form of matrixes, and then combine the PTMs of gates to form an overall PTM of the circuit. Output probabilities could be extracted by the circuit PTM. But it doesn't solve the problem of runtime complexity and memory storage in the case of a large circuit PTM. For a circuit with m inputs and n outputs, it suffers from the computation complexity of $O(2^{m+n})$. Even the author has compressed circuit-matrices using ADDs (algebraic decision diagrams) [45] in [11], it is intractable to store the circuit PTM for large circuits. To be more efficient in runtime or memory storage, several methodologies are proposed but generally they give the approximated results of reliability.

In [46], signal probability reliability (SPR) considers that each internal node has four values (correct 0, correct 1, incorrect 0 and incorrect 1), and each node in the circuit is characterized by its probability of having a correct value. The reliability calculation is based on the cumulative effect of errors in the signals of the circuit. The advantage of SPR over PTM is the linear complexity while it suffers from its disadvantage as accuracy decrease in case of nodes recovergents.

How to deal with re-convergence sources is always a bottleneck in reliability evaluation of combinational circuits. For example, in the *SPR* method, each signal has four states. If there are M re-convergence sources in a given circuit, the computational complexity will be increased to a factor of 4^M. SPR in multi-pass (SPR_{MP}) presents a solution but is time-consuming. A method is therefore proposed in [47] to de-correlate the correlated signals by using conditional probabilities. It obtains a tradeoff between accuracy and time complexity.

In [16], algorithms called observability-based reliability analysis are proposed. They are based on the concept of observability which means that a failure at a gate close to the primary output has a greater probability of propagating to primary output than a gate several levels far away. A closed-form expression for reliability of the outputs is derived and it is fast and simple for reliability evaluation, while the problem is that it lacks high accuracy.

Authors in [48] proposed probabilistic treatment of signals in combinational digital IPs. By taking into account this methodology, in [49], the the probabilistic gate model (PGM) is proposed for unreliable logic gates and then PGMs are used as models to analyze the circuit reliability. At first, PGM-based method also suffers from runtime problems. The authors therefore improved their model in [15] where PGMs are used in a hierarchy way. Large circuits are decomposed into their smaller components and then reliability estimation could be obtained by recursively use of PGM model. This improvement presents a tradeoff between accuracy and complexity.

Boolean Difference was firstly used in error detection of digital IPs in [50]. In [14], a model was proposed by using Boolean Difference calculus, which is applied to probabilistic analysis of logic circuits. This error/reliability calculator *Boolean Difference-based Error Calculator* (BDEC), takes signal and error probabilities of primary inputs, gate error probabilities and then computes the reliability of the circuit. It benefits from a linear-time complexity with the number of gates in the circuit with very high accuracy.

In complex systems in which constant failure rates are assumed but combinatorial arguments are insufficient for analyzing the reliability of the system, we can use Markov models for deriving expressions for the system reliability. In addition, Markov models provide a structured approach for the derivation of reliabilities of systems that may configured with hardware redundancy and a repair process. In Section 2.4.3.2 we will present an example of calculation of MTTF values in majority voting redundant systems with Markov Models.

A probabilistic design methodology based on Markov Random Fields (MRF) is presented in [51] that uses the Gibbs distribution to characterize reliability in terms of entropy and noise in terms of thermal energy. Evaluating reliability using this technique becomes computationally intensive for arbitrary multilevel logic circuits because it involves minimization of a complex Gibbs distribution function with a large number of variables. This technique is more suitable for evaluating reliability of regular redundancy architectures like triple modular redundancy. And in [17], an approximate inference scheme is proposed for the handling of large circuits using a probabilistic model based on Bayesian networks (BNs). Without exception, these two approaches suffer from the problem of scalability.

An exact analysis method at logic-level was proposed in [52] which takes advantage of circuit transformations to calculate the signal probabilities of all internal nodes of logic circuits. At the same time, the authors proposed a hybrid measure to estimate reliability of digital IPs that provides a trade-off between scalability and accuracy. However, rewiring the signals connecting to less reliable nodes brings with some over-cost.

2.3.2 Simulation-based way

Standard techniques for reliability analysis consist of faults into internal nodes of the circuits and then simulate the circuit for different input vectors in a Monte Carlo (MC) framework. The main drawback of simulation is that numerous pseudo-random numbers need to be generated thus a large number of simulation runs must be executed to reach a stable output, so making this way very time-consuming for large circuits or circuits with large number of inputs. However, simulation has still been considered as an alternative in industry field when analytical approach becomes intractable [51, 53–55].

2.3.3 Hybrid way

Recently, some approaches making use of both analytical and simulation-based methods have been proposed.

A probabilistic binomial model for reliability calculation (PBR) was presented in [13]. It takes into account the logical masking properties of the circuit structure, and gives accurate results. This approach can easily be integrated in the design flow of the target systems. Determination of reliability depends on an exhaustive fault simulation of the target circuit.

Based on the PGM model, in [56], a computational approach using the stochastic computational models (SCMs) accurately determines the reliability of a circuits with its precision only limited by random fluctuations inherent in the representation of random binary bit streams. The SCM approach has a linear computational complexity and thus it is scalable for large circuits. However, since it is modeled under a stochastic circumstance, shortcomings like random permutation and random fluctuation are inevitable. It is thus another aspect of trade-off between accuracy and scalability.

As in SCM, gate failure is assumed as a constant probability which is not desirable in physical basis (because faults actually affect devices such as transistors). A more practical approach is proposed in [57]. It utilizes a transistor-level stochastic analysis for digital fault modeling. i.e., a simple equation is used to relate the reliability of the transistors to that of a gate, $\epsilon_{gate} = (1 - \epsilon_{transistor})^n$, where n is the number of transistors in the gate.

2.3.4 Comparison of existing techniques

Analytical approaches can handle reliability assessment of small and simple circuits without loss of accuracy, buy the efficiency and accuracy becomes difficult for VLSI circuits. Therefore, a trade-off is usually made for accuracy versus the complexity, speed and memory. These trade-offs must consider *signal correlation*. It refers to the number of recovergent fanouts in combinational circuits, and the number of feedbacks in sequential circuits.

In this case, to allow the analysis of larger size circuits, each approach has its way to make the compromise. For example, SPR_{MP} and $PGM_{Accurate}$ could consider less reconvergent fanouts, PBR could make reductions of fault emulations and SCM could reduce the length of input stochastic signals.

Table 2.1 presents some quantitative comparison with related to speed, accuracy, memory and scalability of the representative methods. This comparison derives from statements references in [10–17] etc. Since the limitation on the size of matrices that is necessary for PTM, its usage is very restricted. The speed of PTM could not be found in literature.

Table 2.1: Comparison of representative reliability evaluation methods.

Methods	Speed	Accuracy	Memory	Scalability
PTM	–	exact	very high	no
BN	medium	exact	high	no
PGM	medium	low	low	no
SPR_{MP}	slow	exact	low	no
PBR	slow	exact	low	no
SCM	medium	adaptive	low	yes
$BDEC$	medium	high	low	yes

2.3.5 Comments on PGM approach

The PGM approach produces the gate's PGM according to (2.16), where ϵ is the gate failure probability, *P* is the probability of the output of the gate being stimulated, and p_i is the probability that a fault-free gate will produce logic-1 at its output. This is reasonable. However, given an output's signal probability $P(1)$ and $P(0) = 1 - P(1)$, the author defined that reliability of the circuit is calculated according to (2.17), where $P_e(1)$ and $P_e(0)$ are the probabilities that the output is expected to be "1" and "0" respectively, under the condition that the gates are fault-free [15, 49, 58]. And they declared that reliability evaluation could be transformed into the problem of evaluation of signal probability of the output, which is easy to solve.

$$\mathbf{P} = \begin{bmatrix} p_i & 1-p_i \end{bmatrix} \cdot \begin{bmatrix} 1-\epsilon \\ \epsilon \end{bmatrix} \tag{2.16}$$

$$R = P(1)P_e(1) + P(0)P_e(0) \tag{2.17}$$

Although the author presented the reliability accuracy is high, the reliability evaluation formula is worthy to be discussed here. For example, we consider a NAND gate with gate failure rate $\epsilon = 0.1$. As shown in Figure 2.6, signal probabilities of primary inputs of a NAND gate are $P(A) = P(B) = 0.5$. According to (2.16), output signal probability is calculated as in (2.18) and reliability is calculated as in (2.19). The result 0.6 obtained by *PGM* is contradictory with *PTM*, *SPR*, etc., which is 0.9.

$$\mathbf{P}(\mathbf{Z}) = \begin{bmatrix} 1-P(A)P(B) & P(A)P(B) \end{bmatrix} \cdot \begin{bmatrix} 1-\epsilon \\ \epsilon \end{bmatrix} = 0.7 \tag{2.18}$$

$$R = P(1)P_e(1) + P(0)P_e(0) = 0.75 \cdot 0.7 + 0.25 \cdot 0.1 = 0.6 \tag{2.19}$$

Another example is addressed here. We compare the reliability results of *PGM* and *SPR*. Consider a simple OR gate with failure rate $\epsilon = 0.05$ as shown in Fig. 2.7. Input signal probabilities are shown

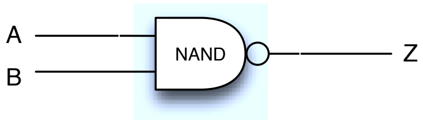


Figure 2.6: A NAND Gate.

in (2.20) and (2.21), so we have $P(A) = 0.3$ and $P(B) = 0.5$.

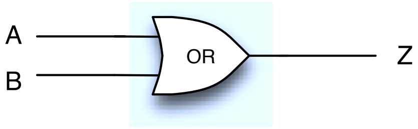


Figure 2.7: An OR Gate.

$$\begin{bmatrix} P(A=0_c) & P(A=1_i) \\ P(A=0_i) & P(A=1_c) \end{bmatrix} = \begin{bmatrix} 0.675 & 0.075 \\ 0.025 & 0.225 \end{bmatrix} \tag{2.20}$$

$$\begin{bmatrix} P(B=0_c) & P(B=1_i) \\ P(B=0_i) & P(B=1_c) \end{bmatrix} = \begin{bmatrix} 0.49 & 0.01 \\ 0.01 & 0.49 \end{bmatrix} \tag{2.21}$$

According to (2.16), signal probability is derived as in (2.22) which is 0.6350. Then reliability evaluation is done according to (2.17), that is shown in (2.23). The result obtained here have a huge error compared to 0.89285 which is obtained by *SPR*.

$$\begin{aligned} \mathbf{P(Z)} &= \begin{bmatrix} P(A)+P(B)-P(A)P(B) & 1-P(A)-P(B)+P(A)P(B) \end{bmatrix} \\ &\cdot \begin{bmatrix} 1-\epsilon \\ \epsilon \end{bmatrix} \\ &= 0.6350 \end{aligned} \tag{2.22}$$

$$R = P(1)P_e(1) + P(0)P_e(0) = 0.635 \cdot 0.65 + 0.3650 \cdot 0.35 = 0.5450 \tag{2.23}$$

Apparently, there is no link between the single probability and signal reliability of output. Therefore it is very hard to find the accuracy of the model *PGM* based on the reliability evaluation formula in (2.19). Furthermore, the authors reported different reliability results of the same benchmark circuits by using the same method. For example, for ISCAS85 benchmark circuit C17, the reliability value was 0.7840 in [49] compared to 0.7582 in [15] respectively, by using the same method *PTM* which is the representative reliability evaluation method. For the proposed *PGM*, the reliability value was 0.7620 in [49] compared to 0.7582in [15].

2.4 Reliability Improvement based on redundancy

In order to guarantee the circuit reliability against SEUs and SETs, several mitigation techniques have been proposed in literature during the last few years. Mitigation techniques against soft errors in logic circuit include system-level schemes, such as back up a same system and run them simultaneously. This method is mostly used in real time or critical space applications, and the hardware cost is too expensive to apply it on daily application. Representative logic-level methods includes Triple Module Redundancy (TMR), Error detection and correction coding (EDC), and this kind of methods are now proved to be the most effective and feasible.

On the other hand, these mitigation techniques could be classified into fabrication-based, design-based and recovery-based. Fabrication-based techniques are mainly concerned in reduction of radiation effects. Recovery-based methods try to recover the initial programmed information after an upset. This dissertation deals with design-based techniques at logic-level and it mainly concentrates on module redundancy techniques.

2.4.1 Types of redundancy

Redundancy resources are needed in fault-tolerant designs. There are four forms of redundancy: hardware, software, information and time. Redundancy techniques can include any or a composite of these four forms [18–20].

- Hardware redundancy is realized by adding extra hardware into the design to either detect or mask the errors of a failed component. For example, we could use two or three units performing the same function instead of utilizing a single unit. In this way, errors could be detected by the two function units; and for three unit, the majority output can override the wrong output of the single function unit. Hardware redundancy is normally reserved for critical systems where area overheads could be ignored compared with reliability requirements because of its high overheads. However, since its high reliability and straightforward structure, it is still preferred in many designs. Thus methodologies for alleviating the performance degradation by hardware redundancy have been proposed in recent years to make hardware redundancy more feasible in most applications.
- Information redundancy refers to the addition of extra information to data in order to allow fault detection, masking or tolerance. It is known as its application like error detection and correction coding. They are widely used to protect data transmitting in noisy channels. By the way, these error codes also require hardware to process the redundant data.

– Time redundancy provides repetition of a given program on the same hardware in a number of times. Then the comparison of results will determine if a discrepancy exists. It has much lower hardware overhead but causes a high performance penalty like latency.
– Software redundancy is used to deal with software failures. Similar to hardware redundancy, multiple versions of the program can be executed concurrently. In fact, it will cause hardware and time redundancy as well.

Not all fault-tolerant techniques are applicable at all design levels. Therefore the applicability level is an important property of fault-tolerant techniques. Figure 2.8 shows the applicability of different fault-tolerant techniques.

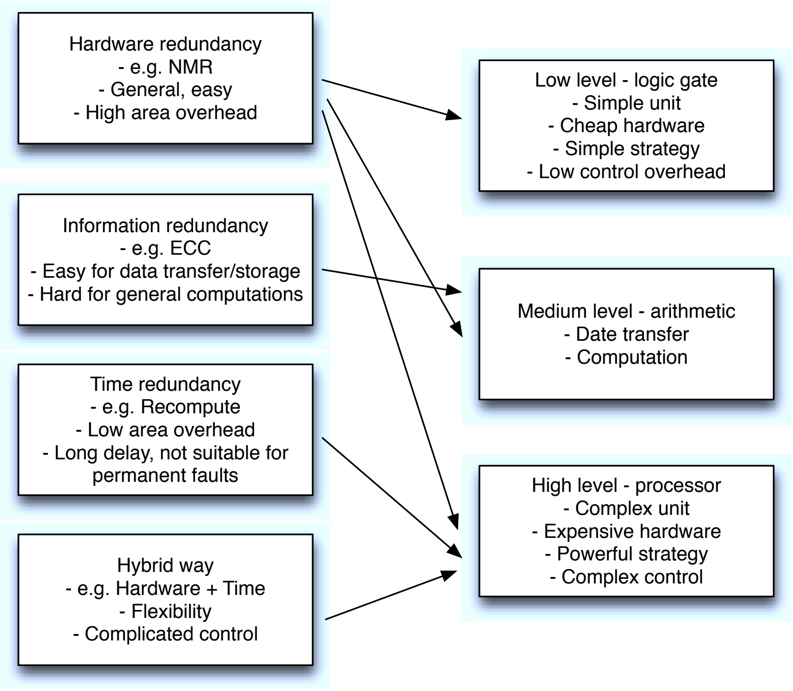


Figure 2.8: Fault-tolerant technique and its applicability (adapted from [1, 2]).

2.4.2 Hardware fault tolerance

Hardware fault tolerance is the most popular and mature area in the field of fault tolerant computing and architecture. Many hardware fault-tolerant techniques have been developed and implemented in practice both in critical applications ranging from space missions to medical instruments and in consumed electronics like display techniques [18, 21, 59–61]. Concurrent computations are implemented on the extra hardware redundancy. The outputs are fed to a voter so that errors could masked, and the reliability of the circuit is therefore enhanced. Or the duplicate (spare) hardware can be switched automatically to replace failed components.

Hardware redundancy can be characterized by passive (static), active (dynamic) and hybrid [2, 18, 19]. The first is related with fault masking, in the applications of reliability improvement. The second is mainly concerned with fault detection and replacement of faulty modules thus it is suitable in the situation of monitoring, for example. Hybrid redundancy requires both fault masking and replacement.

2.4.3 Reliability of M-of-N systems

An M-of-N system is a system which consists of N modules and requires at least M of them for correct operation. In other words, the system fails when less than M modules are functional. It is a representative technique of passive (static) hardware redundancy.

The well-known example is triplex, which consists of three identical modules whose outputs are voted after. This is the situation when $N = 3$ and $M = 2$. As long as a majority (2 or 3) modular functional, the system will have an expected output.

We now consider the reliability of an M-of-N system which could be expressed as follows,

$$R_{M-of-N} = \sum_{i=M}^{N} \binom{N}{i} R^i (1-R)^{N-i} \tag{2.24}$$

The corresponding diagram of Markov model is shown in Figure 2.9. This state diagram is not hard to develop for reliability computation.

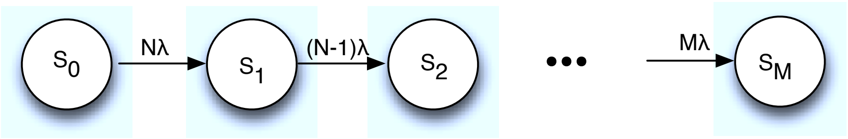


Figure 2.9: Markov diagram for M-of-N configuration.

2.4.3.1 N-tuple Module Redundancy

An M-of-N system becomes an N-tuple Module Redundancy (NMR) structure when $N = odd$ and $M = \lceil N/2 \rceil$. Figure 2.10 presents a general scheme for NMR based on majority voter. There are $N + 1$ components: N replicas of the module **M** and a voter **MAJ**. With the majority voter algorithm, the output chosen to be the correct result is that which was produced by more than half of the replicated modules.

The reliability of a NMR system based on ideal majority voter can be given by the expression (2.25), where q_M is the reliability of each module and $\binom{N}{i} = \frac{N!}{i!(N-i)!}$ is the number of i-element subsets of an N-element set.

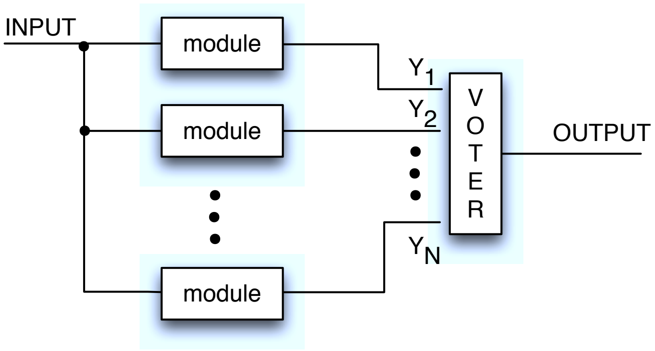


Figure 2.10: General scheme for NMR based on majority voter.

$$R_{NMR-IV} = \sum_{i=\lceil N/2 \rceil}^{N} \binom{N}{i} q_M^i \left(1 - q_M\right)^{N-i} \tag{2.25}$$

Notice that q_M gives the probability that a module generates correct outputs and $\binom{N}{i}$ represents the amount of possible combinations with i correct modules and $N - i$ faulty modules. Therefore, (2.25) gives the probability of a majority module working correctly. In the special case of triple modular redundancy, this derives the value given in (2.26).

$$R_{TMR-IV} \quad = \quad \left(3q_M^2 - 2q_M^3\right) \tag{2.26}$$

Figure 3.6 compares the reliability of a system implementation based on a single module with one based on the modules replication according to equation (2.26). As the reliability of a simplex (a single module) decreases, the advantages of redundancy become less marked; until for q_M < 0.5, redundancy actually becomes a disadvantage, with the simplex being more reliable than either of the redundant arrangements. We observe that:

- if $q_M < 0.5$, then $R_{TMR} < q_M$, that is to say TMR strategy is invalid in improving the reliability compared with the original module without replica.
- if $q_M > 0.5$, then $R_{TMR} > q_M$, so demonstrating the TMR strategy carries to an enhancement of the reliability.

In fact, this is also reflected in the value of $MTTF_{TMR}$, which (for $R_{voter(t)} = 1$ and $R(t) = e^{\lambda t}$) can be calculated as below.

$$MTTF_{TMR} = \int_0^\infty \quad 3R(t)^2 - 2R(t)^3 dt = \frac{5}{6\lambda} < \frac{1}{\lambda} = MTTF_{Simplex} \tag{2.27}$$

In most applications, however, $R(t) > 0.5$ is realistic and the system is repaired or replaced long before $R(t) < 0.5$, so a triplex arrangement does offer significant reliability gains.

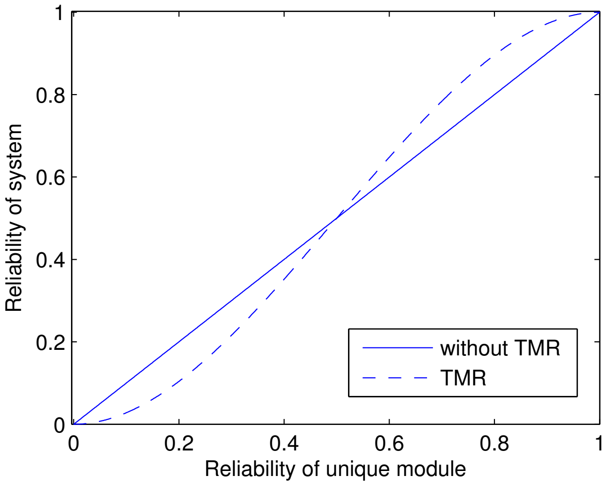


Figure 2.11: Reliability curves for two implementations strategies: unique module (continuous line) and TMR (dashed line).

2.4.3.2 TMR reliability with repair

As stated before, as long as the operating circuit processes redundancy, sometimes we need some additions of repair or replacement to raise the circuit reliability. We have analyzed the reliability of a TMR system without repair in the last section. Here we consider the reliability of a TMR system by using Markov model. If we consider the voter will not fail, the markov system states are in Table 2.2 and the diagram of system states is shown in Figure 2.12.

Table 2.2: System states of a markov model for a TMR.

State	Description
S0	Three components functional, zero failure
S1	Two component functional, one failure
S2	System failure, two or three failures

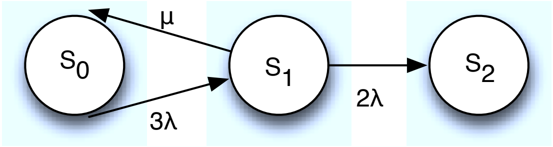


Figure 2.12: A Markov model for a TMR system with repair [3].

In the TMR model of Figure 2.12, we illustrate the use of Markov reliability model as follows:

$$\frac{\mathrm{d}P_{s0}(t)}{\mathrm{d}t} = -3\lambda P_{s0}(t) + \mu P_{s1}(t) \tag{2.28}$$

$$\frac{\mathrm{d}P_{s1}(t)}{\mathrm{d}t} = 3\lambda P_{s0}(t) - (2\lambda + \mu) P_{s1}(t) \tag{2.29}$$

$$\frac{\mathrm{d}P_{s2}(t)}{\mathrm{d}t} = 2\lambda P_{s1}(t) \tag{2.30}$$

where μ is repair rate, as defined in (2.6).

Assuming that both systems are initially good, the initial conditions are

$$P_{s0}(0) = 1, \qquad P_{s1}(0) = P_{s2}(0) = 0 \tag{2.31}$$

Then we transform the the set of equations into the Laplace domain, yielding:

$$P_{s0}(s) = \frac{s + 2\lambda + \mu}{s^2 + (5\lambda + \mu)s + 6\lambda^2} \tag{2.32}$$

$$P_{s1}(s) = \frac{3\lambda}{s^2 + (5\lambda + \mu)\, s + 6\lambda^2} \tag{2.33}$$

$$P_{s0}(s) = \frac{6\lambda}{s\left[s^2 + (5\lambda + \mu)\, s + 6\lambda^2\right]} \tag{2.34}$$

Therefore, the reliability of a TMR system with repair in *S* domain could be yielded as the sum of P_{s0} and P_{s1}, i.e.,

$$\begin{aligned} R_{TMR}(s) &= P_{s0}(s) + P_{s1}(s) \\ &= \frac{s + 5\lambda + \mu}{s^2 + (5\lambda + \mu)\, s + 6\lambda^2} \\ &= \frac{\frac{3\lambda+\mu}{\lambda}}{s + 2\lambda} - \frac{\frac{2\lambda+\mu}{\lambda}}{s + 3\lambda} \end{aligned} \tag{2.35}$$

We can also obtain the time function by Inverse Laplace transform, so yielding the reliability of a TMR system with repair in time domain:

$$R_{TMR}(t) = \left(3 + \frac{\mu}{\lambda}\right) e^{-2\lambda t} - \left(2 + \frac{\mu}{\lambda}\right) e^{-3\lambda t} \tag{2.36}$$

Here we find if $\mu = 0$, i.e., without repair, which yields

$$R_{TMR}(t) = 3e^{-2\lambda t} - 2e^{-3\lambda t} \tag{2.37}$$

and if we define the reliability of single module as $q_M = e^{-\lambda t}$, this becomes equation (2.26) as presented in Section 2.4.3.1, and thus it is coincident with the results previously computed.

$$R_{TMR} = \left(3q_M^2 - 2q_M^3\right) \tag{2.38}$$

2.5 Trade-offs between reliability improvement and performance degradation

As any fault tolerant approach, soft error mitigation may impact significantly area, power and performance. It is therefore necessary to find an effective logic fault architecture which has a better trade-off among these key parameters [62]. Although N-modular redundancy, especially TMR, has been the most prevalent and representative technique in reliability improvement, a full NMR is extremely expensive in terms of performance.

To alleviate the degradation of the performance, the straightforward method is to protect only the critical parts of digital IPs. Selective TMR is therefore proposed to harden a design against SEUs by selectively inserting TMR in those sensitive gates determined by the input environment [63–67]. In [63, 64], the sensitivity of a gate is determined by the signal probabilities at the primary inputs of a circuit. A gate is identified as a sensitive one if an SEU on any one of the inputs is likely to propagate to the final output. The authors in [63, 64] defined that a gate is sensitive when the probability that only one of the inputs to the gate has a dominant logic value or all of the inputs to the gate have non-dominant logic values is greater than the threshold probability specified by the user. If the threshold is set as 0, Selective TMR becomes a full TMR. According to this definition, the gate XOR, XNOR and NOT are obviously always sensitive. However, this method has its shortcomings. Firstly, the threshold which decides the sensitivity of the gates is a subjective parameter and there is a lack of such most appropriate threshold. Secondly, reliability of the circuits not only depends on the logic masking ability of each independent logic gates but also highly relates with the inherent combinational logic.

A logic-level soft error mitigation methodology for digital circuits is proposed in [4] which adds functionally redundant wires selectively to combinational logic of a circuit. It can prevent distorted signal from propagating to an output or a storage element. Another good advantage lies in its slight overhead in hardware, power or delay. This method is based on logic implications in digital IPs. For the circuit example shown in Figure 2.13, one such implication would be $e = 1 \Rightarrow G_8 = 0$. Because if $e = 1$, then $G_3 = 1$, and $G_8 = 0$. Therefore, given $e = 1$, we expect that $G_8 = 0$. If a SET changes the value on the path (including G_3 and G_8), the output of G_8 will become an erroneous value of 1. The proposed method thus added a *redundant function wire* (dotted in Figure 2.13) from e to G_{10} to mask the SET before reaching G_{10}. This method could bring a significant SER reduction while commensurate with the incurred hardware, power and delay overhead. However, to identify the candidate redundant wires by analyzing logic implication is not an easy work. On the other hand, the algorithm to choose the most suitable redundant wires among the candidate subset is complicated.

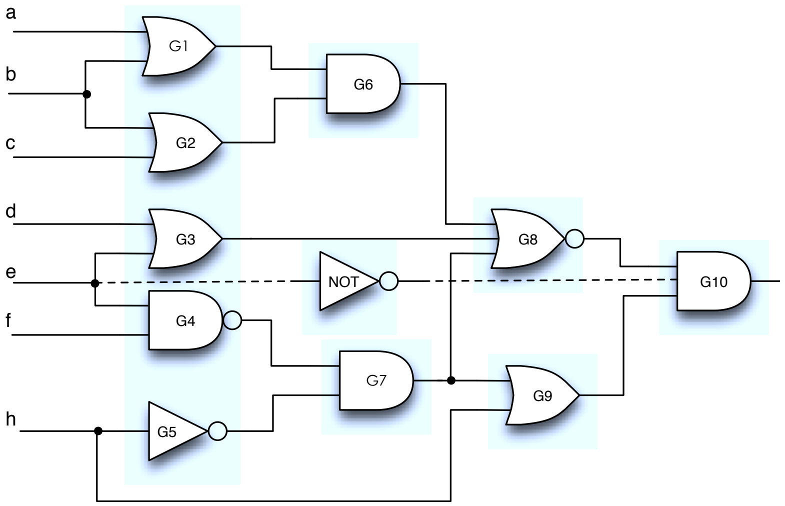


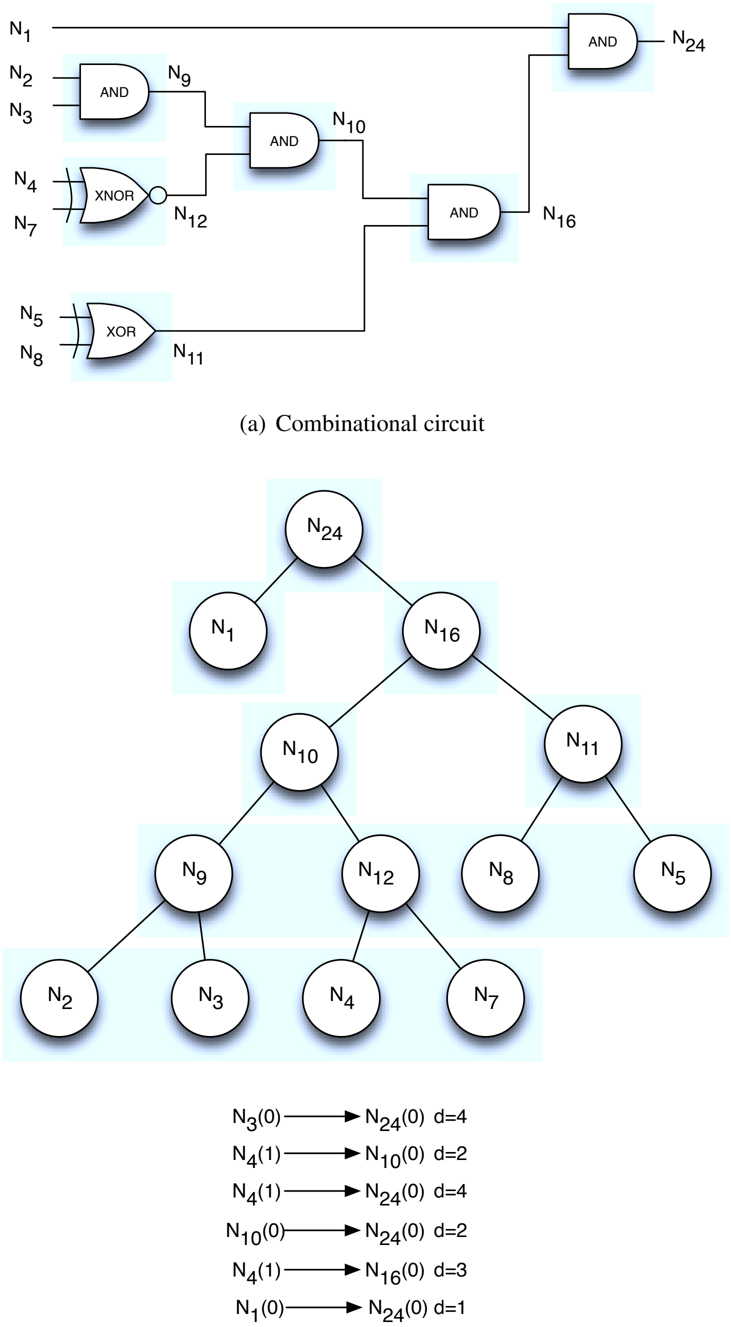
Figure 2.13: Example circuit of adding redundant wires [4].

The difficulty in [4] is to find all the implications in a circuit which is a daunting job. A cost effective approach for online error detection was proposed in [5] to solve this problem. It describes the combinational circuits in graphical way (see Fig: 2.14), and identifies these implications automatically without requiring any knowledge about the high-level circuit behavior. It also avoids the re-synthesis of the logic being protected.

Based on the concept that systems can allow a fixed error rate in a lot of practical applications, a methodology for automatic insertion of TMR is proposed in [68]. It considers TMR insertion as a graph partitioning problem which is included in the category of NP-complete problem. It partitions all registers of the initial circuit into two sub-graphs. One stores the registers that have been chosen to protect with TMR, and the other stores the remaining registers without TMR. In this methodology, the initial partition is important because it will change the convergence iterations (explored partial solutions) a lot. In order to have a better initial partition, the following criterions are proposed and validated in [68].

– A flip-flop with a higher fan-in should not be tripled.
– A flip-flop with a larger fan-out is susceptible to be tripled.
– A flip-flop which is far from the primary output is not convenient to be tripled.
– A flip-flop with feedback loops is more convenient to be tripled.

On the other hand, in [69], a selective fault tolerance (SFT) is described as a modification of TMR by protecting special input signals. It is more useful in practical applications because the designer could choose the subset of inputs according to real needs. This method doesn't choose parts of the circuit to be protected but provides fault tolerance for selected input signals. Reduced area overhead


N_1
N_2
N_3
AND
N_9
AND
N_{10}
N_4
XNOR
N_7
N_{12}
AND
N_{16}
AND
N_{24}
N_5
XOR
N_8
N_{11}
N_{24}
N_1
N_{16}
N_{10}
N_{11}
N_9
N_{12}
N_8
N_5
N_2
N_3
N_4
N_7
$N_3(0) \rightarrow N_{24}(0)$ d=4
$N_4(1) \rightarrow N_{10}(0)$ d=2
$N_4(1) \rightarrow N_{24}(0)$ d=4
$N_{10}(0) \rightarrow N_{24}(0)$ d=2
$N_4(1) \rightarrow N_{16}(0)$ d=3
$N_1(0) \rightarrow N_{24}(0)$ d=1

(a) Combinational circuit

(b) Graphical representation

Figure 2.14: Combinational circuit and its graphical representation [5].

as well as power consumption is demonstrated by validation examples. A subset $\boldsymbol{X_1}$ of all possible inputs $\boldsymbol{X}$ is chosen by the designer in accordance with the real needs of fault tolerance in the concrete circumstance. Then fault tolerance is guaranteed to the chosen subset whereas for the remaining inputs no fault tolerance is demanded. Suppose we have the original combinational system S_1 with m binary inputs $x = (x_1, x_2, \cdots, x_m)$ and a single output y implementing an m-ary Boolean function $y = S_1(x_1, x_2, \cdots, x_m) = S_1(x)$. Different from traditional TMR, the additional systems $S_2(x)$ and $S_3(x)$ implementing the m-ary Boolean functions $S_2(x)$ and $S_3(x)$ are constructed as follows.

The Boolean functions of $S_2(x)$ and $S_3(x)$ are defined as:

$$S_2(x) = \begin{cases} S_1(x) & \text{for } x \in X_1 \\ - & \text{don't care otherwise} \end{cases}$$

$$S_3(x) = \begin{cases} S_1(x) & \text{for } x \in X_1 \\ S_1(x) & \text{for } S_1(x) \neq S_2(x) \\ - & \text{don't care otherwise} \end{cases}$$

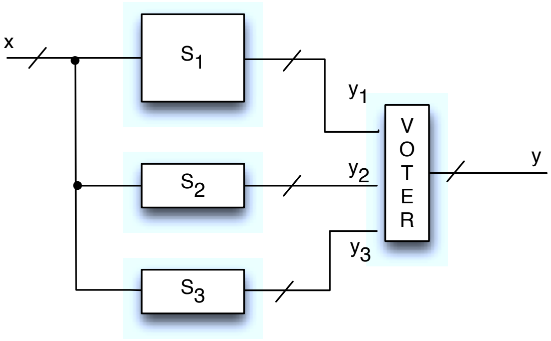


Figure 2.15: Scheme of selective fault tolerance.

A framework for NMR-based fault tolerance is investigated in [70] to perform dynamic tradeoffs between power consumption and reliability. This *fluid* approach is based on the concept of application's error tolerance. The voting strategies and number of modules to be used for voting dynamically switch based on an application's error tolerance. A famous face detection algorithm [71] was used as an example application to demonstrate the proposed fluid NMR framework can produce significant power and performance benefits over traditional NMR. As a large class of emerging applications have algorithmic and cognitive error tolerance, this dynamic concept is useful in the future work.

In [72], a partitioned TMR design based on FPGA is proposed by partitioning the digital IPs into three levels: maximum, medium and minimum logic partitioning. By using maximum logic partition, each combinational logic component was triplicated and majority voters were inserted in their outputs. On the contrary, the minimum partition only triplicate the whole circuit and only a majority voter is placed in the outermost output. The medium could choose a repeated logic function block as a unit to be triplicated and voters were placed in the output of each logic partition. Of course, the maximum logic partitioning TMR implementation is the best solution in reliability improvement. However, this work is based on a full TMR solution. In other words, all components are triplicated among either of the three configurations, and differences only lie in the number of majority voters. While in the nano-electronical era, novel devices like quantum-dot cellular automata (QCA), single electron tunneling (SET) make the implementation of majority voters in a single gate [73], so the overheads based on voters are not very significant as before.

In addition, redundancy based fault tolerance can be implemented at different levels of granularity, such as gate level, logic block level, logic function level, etc. A tool for evaluating granularity and reliability trade-offs in nano-architectures was proposed in [74]. Another software-supported methodology to alleviate the performance degradation of TMR solutions was presented in [75]. It removes redundancy from nonsensitive sub-circuits by extracting the temperature profile across the FPGA since the failure probability for a device region increases with temperature. Another method improves the circuit reliability by using *non-uniform redundancy*. It is a dynamic programming algorithm that leverages that circuits have non-uniform vulnerability and non-uniform observability when sub-circuits to implement redundancy. Besides, the triple interwoven redundancy (TIR) and its extended higher orders (N-tuple interwoven redundancy, NIR) are proposed to achieve higher system reliability [76]. This structure is based on TMR and NMR but implemented with random interconnections. This TIR/NIR implementation is in particular suitable for molecular nano-computers, which are fabricated by a manufacturing process of stochastically chemical assembly.

In conclusion, all of the techniques and methodologies focused on soft error mitigation and reliability improvement are trying to find a better trade-off between reliability and cost penalty. It is also a motivation of our investigation on this purpose.

Chapter 3

Reliability Enhancement based on Significance

3.1 Introduction

The objective of this dissertation is to develop new methods and architectures for logic function IPs bringing with optimized trade-offs which can improve reliability and give consideration to classical performance parameters.

As digital IPs consist of a series of sub digital blocks, the reliability depends on the reliabilities of these individual blocks, as well as the combinational logic. As noted in [77], many large circuits contain a limited number of simple logic components that are used repeatedly throughout a design. Meanwhile, motivated by the *Pareto principle*, it is very interesting to find the best candidates whose reliabilities play important roles in the overall reliability. This chapter presents several works concerning significance of constituent blocks of digital IPs related to reliability. We proposed two concepts *sensitivity* and *eligibility* that succeed to identify most significant blocks according to different design criteria. To validate the feasibility and efficiency of the proposed concepts, implementations are presented to illustrate how they help to accomplish block grading.

3.2 Identification of critical blocks

The reliability of a circuit consisting of several blocks depends on the reliabilities of these individual blocks. This is shown in equation (3.1) for a circuit consisting of K blocks, where R is the circuit's reliability and q_i, q_j stand for the reliabilities of the blocks b_i, b_j respectively ($1 \leq i, j \leq K$).

$$R = f(q_1, q_2, ...q_i, ...q_j, ...q_K) \tag{3.1}$$

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

3.2.1 A simple example

Let us start from a simple example. Suppose a circuit has the structure in Figure **??** and each block b_i has the same reliability q. Suppose we use QMR(5MR) once and TMR once respectively in this circuit. In this case, let $R_{5,3}$ stands for the reliability of the structure that put QMR in b_1 and TMR in b_2. And $R_{3,5}$ represents the opposite one. We could calculate reliability of the circuit by $R = \prod_{i=1}^{K} \prod_{j=1}^{i} q_j$, so yields (3.2) and (3.3)

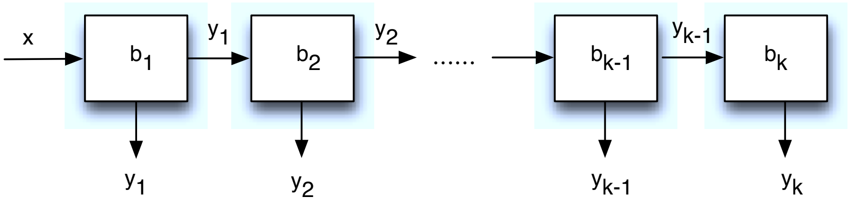


Figure 3.1: Cascade structure.

$$R_{5,3} = \prod_{i=1}^{K} r_i = q_5^k \cdot q_3^{k-1} \cdot q^{(1+2+\cdots+k-2)} \tag{3.2}$$

$$R_{3,5} = \prod_{i=1}^{K} r_i = q_3^k \cdot q_5^{k-1} \cdot q^{(1+2+\cdots+k-2)} \tag{3.3}$$

In (3.2) and (3.3), q_5 stands for block has the improved reliability with configuration QMR. And q_3 is improved reliability by TMR. q_5 and q_3 are given according to equation (3.4) , when N = 5 and N = 3 respectively.

$$R_{NMR} = \sum_{i=\lceil N/2 \rceil}^{N} \binom{N}{i} q^i \left(1-q\right)^{N-i} \tag{3.4}$$

Obviously, the relation of $R_{5,3}$ and $R_{3,5}$ is verified as follows.

$$\frac{R_{5,3}}{R_{3,5}} = \frac{q_5}{q_3} > 1 \tag{3.5}$$

Thus it reveals the first way of redundancy configuration produces the higher reliability. This example is very simple but it implies that we could add redundancy more efficiently according to the concept of *block significance*. Although it is straightforward, this example demonstrates that in cascade circuits, the importance of block is monotone decreasing from inputs to outputs.

In a more general way, we will consider that a reliability change of a single block b_i brings in its new reliability q_i^*, the circuit's reliability becomes R_i^*. Since different blocks b_i and b_j make different contributions to the circuit's reliability, changes of different blocks may produce different values R_i^* and R_j^*.

We propose to design a fault tolerant system in the following progressive steps.

1. Grading constituent sub-blocks of a combinational circuit based on a criterion. (For instance, sensitivity to faults is a kind of criterion)
2. Implementing the fault tolerant techniques progressively on these ranked blocks. (For instance, modular redundancy is a kind of fault tolerant techinque)

The next section presents *sensitivity* and *eligibility* concepts, they help to identify what are the more important blocks in different digital IPs.

3.2.2 Sensitivity and eligibility concept

The concept of sensitivity has been studied in mixed signal applications from a layout point of view [78,79]. For example, drain and source of a MOS transistor are considered as sensitive parts of a circuit node [78]. Previous works have also demonstrated fault sensitivity estimation in fault-tolerant design has a great effect to power consumption [79]. However, very little has been done to analyze the importance of the constituent blocks of digital IPs at logic level. Therefore, we try to explore concepts and methods to classify constituent blocks with respect to reliability.

Consider a circuit $\mathcal{C}$ with reliability R and being constituted of K independent blocks b_i. Let $B = \{b_1, b_2, \cdots, b_K\}$ be the set of all K blocks in $\mathcal{C}$ and $Q = \{q_1, q_2, \cdots, q_K\}$ is the set of their respective reliabilities (i.e, q_i is the reliability of b_i). We define the **sensitivity** of $\mathcal{C}$'s reliability with respect to b_i's reliability as a metric giving the impact on R by changing q_i. This is expressed as (3.6) that stands for partial derivative of function R with respect to variable q_i.

$$s(b_i) = \left|\frac{\partial R}{\partial q_i}\right| \tag{3.6}$$

We denote $\Theta = \{\theta_1, \theta_2, \cdots, \theta_K\}$ as the set of blocks ordered according to sensitivity values in such a way that θ_1 (resp. θ_K) stands for the block whose sensitivity is maximal (resp. minimal).

We define the **eligibility** of a block b_i, noted $e(b_i)$, as the metric expressing how reliability improvement of this block is meaningful. Let us denote $\Delta_i = |R_i^* - R|$ as the reliability change resulting from q_i improvement. The eligibilities of two blocks b_i, b_j are then defined such that they satisfy $R_i^* > R_j^* \Rightarrow e(b_i) > e(b_j)$. We denote $\Lambda = \{\lambda_1, \lambda_2, \cdots, \lambda_K\}$ the set of blocks ordered according to eligibility values, where λ_1 (resp. λ_K) represents the block has the highest (resp. the least) priority for reliability improvement. For a circuit such as $\mathcal{C}$, $e(b_i)$ are integers in $[1, K]$, where 1 and K represent the less and the most eligible blocks, respectively.

Note that eligibility values depend on the techniques adopted to improve the reliability of the blocks. Consequently, the ordering of blocks according to their eligibilities is not necessarily the one given by the set Θ.

Consequently, the sensitivity and eligibility concepts propose an implication which reflects relations between reliability improvement and block grading based on their weights, and reveals opportunities for adding redundancy more efficiently.

Sensitivity and eligibility are two concepts closely related, but addressing different analysis objectives. Sensitivity is useful to determine which blocks are critical in $\mathcal{C}$, that is blocks whose reliability degradation would lead to significant degradation of $\mathcal{C}$'s reliability. Eligibility determines the order in which blocks should have their reliability improved to obtain the best gain in $\mathcal{C}$'s reliability.

3.2.3 Block grading according to design requirement

Let a circuit be constituted of a certain number K of blocks b_1, b_2, ..., b_K. Denote the reliability of each block b_i by q_i. The reliability R of such a circuit has a close relation with the reliabilities of its constituent blocks. We can rank these blocks according to the design objective.

For example, if we focus on reliability improvement, we can improve the reliability of a single block b_i as in (3.7), the circuit's reliability will be improved as in (3.8).

$$q_i^+ = q_i + \Delta q_i \tag{3.7}$$

$$R_i^+ = R + \Delta R_i \tag{3.8}$$

The value Δq_i stands for the individual reliability improvement of block b_i. Similarly, the value ΔR_i expresses the global reliability increase due to reliability improvement of block b_i. Indeed, Δq_i depends on the technique (noted t_i) used for reliability improvement of b_i while ΔR_i scales with the weight (noted w_i) of Δq_i in the new global reliability calculation.

$$\Delta q_i = f_1(q_i, t_i) \tag{3.9}$$

$$\Delta R_i = f_2(q_i, w_i, t_i) \tag{3.10}$$

The object of reliability improvement is to obtain a best gain of the overall reliability by harden its constituent blocks. This global reliability increase is given by (3.11), where K is the number of blocks in the circuit.

$$\Delta R \quad = \quad R^+ - R = f_3(\Delta R_{i,i=1,\cdots,K}) \tag{3.11}$$

This implication reflects relations between reliability improvement and block grading based on their weights, and reveals opportunities for adding redundancy more efficiently. The greater w_i is, the greater the ΔR_i will be. In the proposed method, the w_i values are integers in the range $[1, K]$, and they can be obtained from the set Θ decided by *eligibility* concept.

By the way, improving reliability of a block may result from actions at different levels of abstraction such as technological or architectural. If the variation of reliability is due to a change in technology, the independence among the blocks is not necessarily.

However, if reliability change of a given block is obtained by changing the logic configuration structure of this block, this will not bring in any changes in the reliabilities of the other blocks. In other words, we can assume that the reliabilities of the blocks are independent of each other.

In this thesis, we consider only logic changes for reliability improvement. Therefore, improving the reliability of the blocks in the circuit could be expresses as below.

$$R^+ = R + \sum_{i=1}^{K} \Delta R_i \tag{3.12}$$

$$\Delta R = \sum_{i=1}^{K} \Delta R_i \tag{3.13}$$

As stated in the last chapter, reliability improvement at logic level is carried out by adding redundancy and therefore the cost penalty is inevitable. Depending on different techniques taken into account, overhead can be expressed in terms of area (A), power consumption (P) or time delay (T):

$$\Delta C = f_4(A, P, T) \tag{3.14}$$

According to the target application and the design constraints, the expression (3.14) must be defined in an appropriate manner to each target application. This means giving more or less importance to A, P or T, according to the constraints of the project.

3.3 Block Grading

In the following sections, we consider a circuit $\mathcal{C}$ with reliability R and being constituted of K independent blocks b_i. Let $B = \{b_1, b_2, \cdots, b_K\}$ be the set of all K blocks in $\mathcal{C}$ and $Q = \{q_1, q_2, \cdots, q_K\}$ is the set of their respective reliabilities (i.e, q_i is the reliability of b_i).

3.3.1 Block grading in cascade structures

Since cascade structures are widely used in digital systems, we first consider blocks $b_i \in B$ as defined before. Assume that these blocks are assembled in a cascade structure such that the input of block b_i is given by the output of block b_{i-1} (see Figure 3.2).

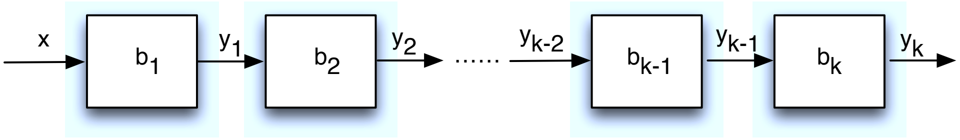


Figure 3.2: A kind of cascade structure.

In order to determine sensitivity and eligibility values, let us define

- $\Theta_i = \{\theta_1, \theta_2, \cdots, \theta_i\}$, the set of the first i elements in Θ set.
- $Q_i = \{q(\theta_1), q(\theta_2), \cdots, q(\theta_i)\}$ the set of the reliabilities of blocks in Θ_i set.
- *Case 1:* $y = y_K$ *and* $R = \prod_{j=1}^{K} q_j$

It can be derived from the sensitivity definition that $s(b_i) > s(b_j) \Leftrightarrow q_i < q_j$, where we can see the smaller q_i is, the higher $\mathcal{C}$'s sensitivity with respect to b_i will be. The ordered set Θ is obtained according to (3.15), where $\Theta_0 = Q_0 = \{\varnothing\}$ and $B - \Theta_{i-1}$ (resp. $Q - Q_{i-1}$) stands for the set of elements in B (resp. Q) not belonging to Θ_{i-1} (resp. Q_{i-1}). Observe that the sensitivity is inversely proportional to the reliability of the block. Furthermore, when all blocks have the same reliability, sensitivity doesn't allow ordering with respect to sensitivity. Ordering of eligibilities is as (3.16).

$$\begin{aligned} \theta_i &= b_j \in B - \Theta_{i-1} | q_j \\ &= \min\{Q - Q_{i-1}\} \end{aligned} \qquad (3.15)$$

$$\begin{aligned} R_i^* > R_j^* &\Leftrightarrow s(b_i)q_i^* > s(b_j)q_j^* \\ &\Leftrightarrow \frac{q_i^*}{q_i} > \frac{q_j^*}{q_j} \end{aligned} \qquad (3.16)$$

- *Case 2:* $y = y_K y_{K-1} \cdots y_2 y_1$ *and* $R = \prod_{i=1}^{K} \prod_{j=1}^{i} q_j$

We can see that the contribution of the individual reliability q_i to the global reliability R depends on the position of the block b_i in the structure. The sensitivity $s(b_i)$ also depends on the block's position i as follows:

$$\begin{aligned} \left|\frac{\partial R}{\partial q_i}\right| &= (K-i+1) q_i^{K-i} \prod_{i=1}^{K} \prod_{j=1, j\neq i}^{i} q_j \\ &= \frac{K-i+1}{q_i} R \end{aligned} \qquad (3.17)$$

$$s(b_i) > s(b_j) \Leftrightarrow \frac{K-i+1}{K-j+1} > \frac{q_i}{q_j} \tag{3.18}$$

The ordered set Θ can be obtained from (3.18). Unlike *Case 1*, the positions of the elements in Θ are no longer arbitrary when they have the same reliability value. In fact, they must satisfy $\theta_i = b_{K-i+1}$. Ordering of eligibilities is as (3.19).

$$R_i^* > R_j^* \Leftrightarrow \frac{(q_i^*)^{K-i+1}}{(q_i)^{K-i+1}} > \frac{(q_j^*)^{K-j+1}}{(q_j)^{K-j+1}} \tag{3.19}$$

In order to illustrate the presented concepts, consider the circuit of Figure 3.2 where $K = 8$. We assume the technique used for reliability improvement is TMR with majority voting mechanism. The most eligible blocks are then tripled and voters are placed at the outputs to identify the correct value (Figure 3.3).

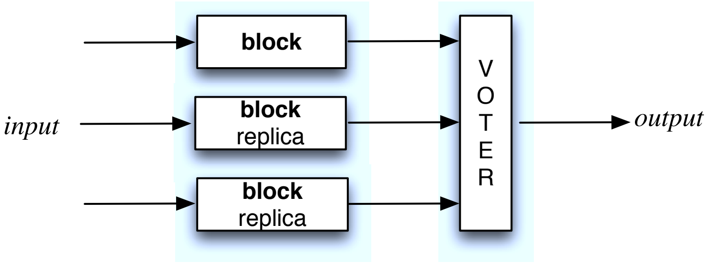


Figure 3.3: TMR Principle.

Implementing TMR with majority voter technique on a block with reliability q_i makes the reliability become q_i^* given by (3.20), where $q_i^* > q_i$ if $q_i > 0.5$.

$$q_i^* = 3q_i^2 - 2q_i^3 \tag{3.20}$$

As explained before, for $\mathcal{C}$ with $y = y_8$ and $q_i = q$, $\forall i$, sensitivity and eligibility don't allow to define the sets Θ and Λ for cascade structures. In other words, all blocks could be treated equally.

For $\mathcal{C}$ with $y = y_8$ and different values of q_i, the ordered set Θ is directly obtained from the first case and eligibility for each block comes from:

$$\begin{aligned} R_i^* > R_j^* \quad &\Leftrightarrow \quad \frac{3q_i^2 - 2q_i^3}{q_i} > \frac{3q_j^2 - 2q_j^3}{q_j} \\ &\Leftrightarrow \quad 3q_i - 2q_i^2 > 3q_j - 2q_j^2 \end{aligned} \tag{3.21}$$

This is equivalent to analyze the monotonicity of the function $f(q) = 3q - 2q^2$. This function $f(q)$ has two monotone parts: it increases when $q < 0.75$ and decreases when $q > 0.75$. Then, the ordered set Λ is easily obtained according to it. Table 3.1 gives the results for two sets of reliabilities Q.

Table 3.1: Results for $\mathcal{C}$ with $y = y_8$ and different values of q_i.

i	1	2	3	4	5	6	7	8
$q_i \in Q$	0.8	0.99	0.94	0.85	0.83	0.87	0.96	0.98
$\theta_i \in \Theta$	b_1	b_5	b_4	b_6	b_3	b_7	b_8	b_2
$\lambda_i \in \Lambda$	b_1	b_5	b_4	b_6	b_3	b_7	b_8	b_2

(a)

i	1	2	3	4	5	6	7	8
$q_i \in Q$	0.68	0.7	0.74	0.85	0.83	0.87	0.96	0.98
$\theta_i \in \Theta$	b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8
$\lambda_i \in \Lambda$	b_3	b_2	b_1	b_4	b_5	b_6	b_7	b_8

(b)

In this case of $\mathcal{C}$ with $y = y_8y_7\cdots y_2y_1$ and $q_i = q, \forall i$, expression (3.19) becomes expression (3.22). Given monotonicity properties of the function $f(q) = 3q - 2q^2$, there are two solutions for Λ set, depending on whether the value q is greater or less than 0.75, as shown in Table 3.2. When $q = 0.75$, all blocks are equally eligible.

$$R_i^* > R_j^* \Leftrightarrow (3q - 2q^2)^i > (3q - 2q^2)^j \tag{3.22}$$

Table 3.2: Results for $\mathcal{C}$ with $y = y_8y_7\cdots y_2y_1$ and $q_i = q, \forall i$.

i	1	2	3	4	5	6	7	8
$q_i \in Q$	q	q	q	q	q	q	q	q
$\theta_i \in \Theta$	b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8
$\lambda_i \in \Lambda$	b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8

$(a)q < 0.75$

i	1	2	3	4	5	6	7	8
$q_i \in Q$	q	q	q	q	q	q	q	q
$\theta_i \in \Theta$	b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8
$\lambda_i \in \Lambda$	b_8	b_7	b_6	b_5	b_4	b_3	b_2	b_1

$(b)q > 0.75$

For $\mathcal{C}$ with $y = y_8y_7\cdots y_2y_1$ and different values of q_i, ordering of the blocks according to sensitivities and eligibilities comes from expressions (3.18) and (3.19). In this case, both position i and individual reliability q_i must be taken into account. Table 3.3 summarizes the results for two examples of Q sets.

Table 3.3: Results for $\mathcal{C}$ with $y = y_8y_7 \cdots y_2y_1$ and different values of q_i.

i	1	2	3	4	5	6	7	8
$q_i \in Q$	0.8	0.99	0.94	0.85	0.83	0.87	0.96	0.98
$\theta_i \in \Theta$	b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8
$\lambda_i \in \Lambda$	b_1	b_4	b_5	b_6	b_3	b_7	b_2	b_8

(a)

i	1	2	3	4	5	6	7	8
$q_i \in Q$	0.68	0.7	0.74	0.85	0.83	0.87	0.96	0.98
$\theta_i \in \Theta$	b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8
$\lambda_i \in \Lambda$	b_3	b_2	b_1	b_4	b_5	b_6	b_7	b_8

(b)

3.3.2 Block grading in general cases

In more general cases, the reliability of the circuit can not be expressed by closed-form equations as in the previous section. Therefore, sensitivity and eligibility analysis are realized by simulations.

3.3.2.1 Experimental metric for sensitivity classification

In order to get the set Θ which characterizes the sensitivity of the blocks, a fault simulation platform is used to inject faults due to Single Event Upsets (SEUs) [13, 80]. Figure 3.4 shows the process of fault injection and analyzing. The circuit is faulted by inserting a single fault to different internal nodes. The results produced by the original and the faulty circuits are compared. If these results are different, it is concluded that the effects of the fault injected have been propagated to the outputs. On the contrary, it is concluded that the fault has been masked (there is a "mask").

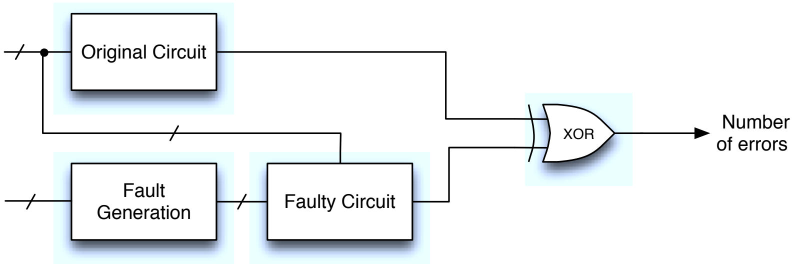


Figure 3.4: Fault injection and masks analysis.

Table 3.4 lists types of faults could be injected. In this table, $V(t)$ is the original value and $F(t)$

stands for the value of injected value. $t1$ and $t2$ are the beginning and end time of fault injection respectively. When the fault is permanent, we can realize it by setting $t2$ as the end of simulation. In this thesis, the fault type we inject is bit-flip, which is also known as von Neumann error [9, 15, 56].

Table 3.4: Fault Models

Fault type	Fault value ($t1 < t < t2$)
$Stuck - at - 0$	$F(t) = 0$
$Stuck - at - 1$	$F(t) = 1$
$Bit - flip$	$F(t) = \ (V(t1))$
$Toggling - bit - flip$	$F(t) = \ (V(t))$
$Indetermination$	$F(t) = random(t)$

We still consider a circuit $\mathcal{C}$ being constituted of K independent blocks b_i. Each b_i contributes to a number of masks, noted as m_i, and the total number of masks m is the sum of m_i, $m = \sum_{i=1}^{K} m_i$.

Each m_i is obtained by calculating the number of masks after fault injection based on the hypothesis that only the corresponding block b_i is faulty. In other words, all the other blocks b_j $(1 \leq j \leq K, j \neq i)$ are fault-free. In this way, each m_i is directly related to the sensitivity of block b_i. The block corresponding to the minimum number of masks is defined as the most sensitive block, i.e,

$$s(b_i) > s(b_j) \Leftrightarrow m_i < m_j \tag{3.23}$$

An illustrated example is the implementation of the ISCAS 85' benchmark C17 [77] as shown in Figure 3.5. In this case, each NAND gate is considered as a block. There are 32 possible input logic combinations and two considered configurations for each block (fault-free and fault-prone). Among the consequent 192 faulty configurations, the results of m_i are given in Table 3.5. Notice that $m5 = m6 = 0$, which means that these two blocks can not mask any faults. Indeed, their outputs are also primary outputs of the circuit and we consider only single fault.

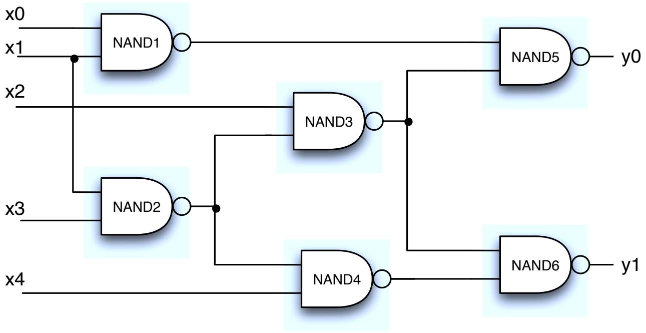


Figure 3.5: An illustrated example: Circuit C17.

Table 3.5: Results of sensitivity rank of C17.

$b(i)$	m_i
$NAND_1$	12
$NAND_2$	8
$NAND_3$	2
$NAND_4$	12
$NAND_5$	0
$NAND_6$	0

3.3.2.2 Experimental metric for eligibility classification

In another aspect, the values λ_i which illustrate the eligibility of the constituent blocks can be obtained by comparing the results of $\Delta_i = |R_i^* - R|$ for each block b_i considering the same technique, as illustrated before. The highest Δ_i value indicates that the block b_i has the highest eligibility, that is, $e_i = K$.

We still use circuit C17 from ISCAS85' benchmark for illustration. Each NAND gate is supposed to have $q_i = q = 0.99$ (99%). After TMR implementation on each NAND gate, reliability improvement results are those shown in Table 3.6, thus $e(i)$ is also decided here.

Table 3.6: Results of eligibility rank of C17.

$b(i)$	R_i^+	$e(i)$
$NAND_1$	95.765%	2
$NAND_2$	95.882%	3
$NAND_3$	96.056%	4
$NAND_4$	95.763%	1
$NAND_5$	96.117%	6
$NAND_6$	96.114%	5

Notice here that eligibility gives the similar order of the blocks as sensitivity since all blocks have the same reliability q_i. However, if each block b_i has a different reliability value q_i, eligibility analysis could produce a different rank of blocks. This is due to the fact that reliability growth of each block is not a linear process by utilizing reliability improvement techniques such as TMR. Indeed, in this case, it is related with the original reliability of each block, as shown in Figure 3.6.

3.3.2.3 Implementation on 74283 fast adder

To further explain the proposed concepts, we consider a more complicated circuit 74283 fast adder. It is constituted of 40 independent gates g_i $(i \in [0, 39])$ as labeled in the gate-level schematic (see Figure 3.7). Similar to the example in Section 3.3.2.1, there are 2^9 possible input logic values and

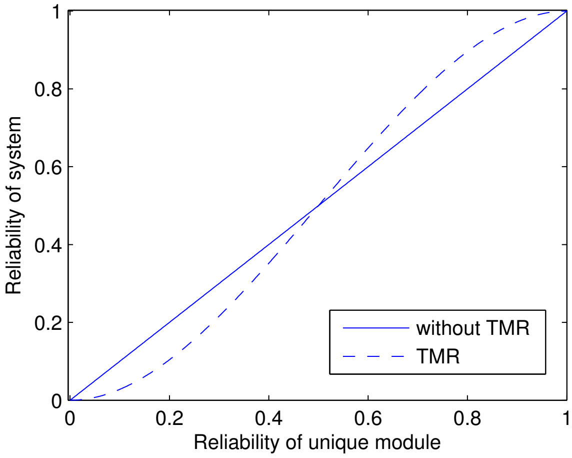


Figure 3.6: Nonlinear Reliability Growth based on TMR Techinque.

two considered configurations for each block (fault-free and faulty). Among the consequent $2^9 \times 40$ faulty configurations, the results of m_i are given in Table 3.7.

Table 3.7: Results for sensitivity factor of 74283 benchmark circuit.

q_i	0	1	2	3	4	5	6	7
m_i	128	0	128	0	128	0	128	0
q_i	8	9	10	11	12	13	14	15
m_i	0	192	224	240	248	240	0	128
q_i	16	17	18	19	20	21	22	23
m_i	192	224	240	224	0	128	192	224
q_i	24	25	26	27	28	29	30	31
m_i	192	0	128	192	128	0	128	0
q_i	32	33	34	35	36	37	38	39
m_i	0	0	0	0	0	0	0	0

According to the number of masks in Table 3.7, blocks are ordered by their sensitivity factors. If number of masks equals, blocks that are nearer the primary outputs are considered as more sensitive ones. If the numbers of masks and the distance to the primary outputs are both identical, the block which has more re-convergent fan-outs is considered more sensitive. If these three parameters are the same, they will be considered to have the same sensitive factor.

On the other hand, to classify the blocks according to the eligibility is obvious. It is based on TMR implementation on each gate. Afterwards, e_i is decided by each reliability improvement value $\Delta_i = |R_i^* - R|$.

Under the digital application circumstances, sometimes we have to take into consideration the real usage of the results of an output of a digital circuit. Output bits that are considered crucial to a

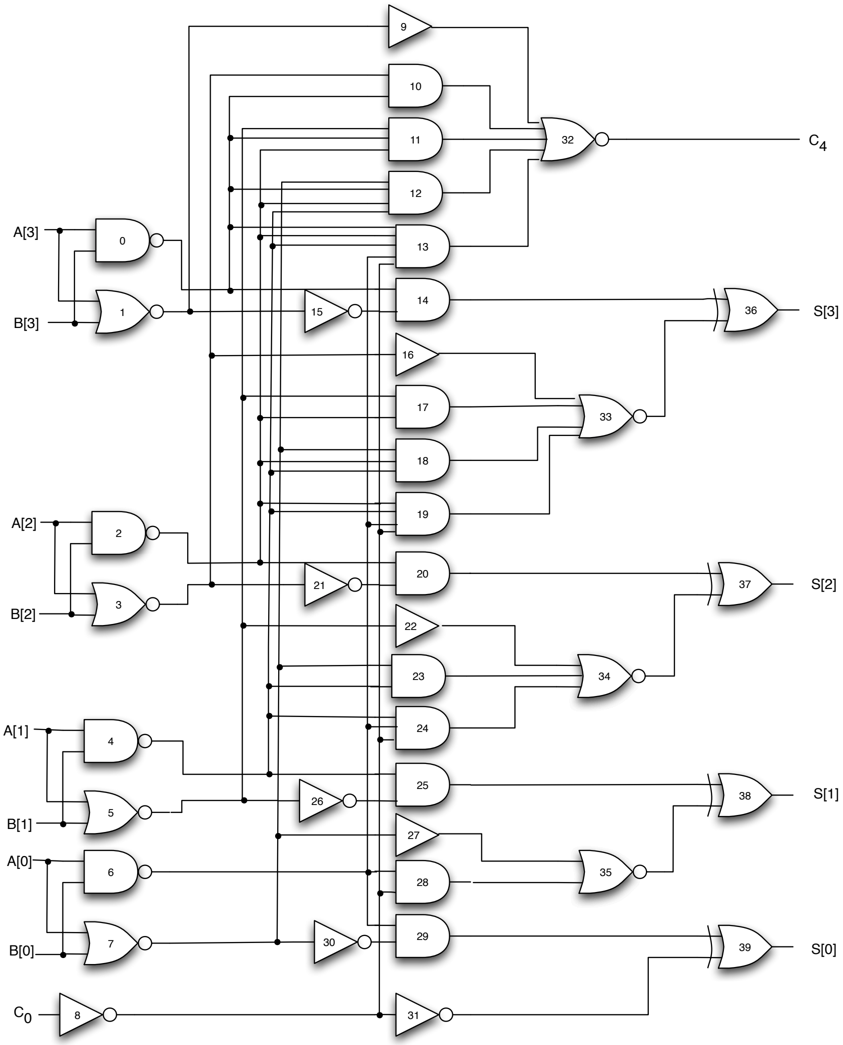


Figure 3.7: 74283 gate-level schematic.

system have higher priorities to be protected, reducing the occurrence of critical errors. This implies a weighted way for grading constituent blocks. The next chapter will present how to grade blocks and configure redundancy (like TMR) by considering the bit significance in order to avoid critical errors in digital IPs.

3.4 Practical Applications of Block Significance

3.4.1 Using Sensitivity for Monitoring

Monitoring mechanisms allow to detect the imminence of a possible failure and then to take appropriate measures to prevent the occurrence of failure. The ability of preventing failures is an essential requirement in circuits used for critical applications as avionics, nuclear, etc. Monitoring the reliability of the blocks in a circuit provides useful information for monitoring the reliability of the circuit and, in consequence, to prevent failures.

Monitoring mechanisms include many aspects such as instrumentation, data collection, data evaluation, data management, and a response plan.

From a design standpoint, the implementation of these mechanisms usually results in increased cost (additional resources for the observability) or decreased performance (implementation of observation itself). Hence, even if it would be interesting to monitor the reliability of all blocks composing a given circuit, it is often necessary to limit the amount of monitored blocks in the circuit.

The proposed concept of sensitivity can be used to guide the designer in the choice of blocks to be monitored, as shown in the workflow of Figure 3.8.

For example, according to Table 3.1.b, the order in which the blocks should be monitored is given by the set:

$$\boldsymbol{\Theta} = \{b_1, b_5, b_4, b_6, b_3, b_7, b_8, b_2\}$$

If only one block should be monitored, the mechanisms of monitoring should be applied to the most sensitive block. Indeed, this is the block whose reliability is the most significant in the sense that a small modification of its reliability will cause a significant change in the reliability of $\mathcal{C}$. Based on Θ set and target application constraints, the designer is able to define trade-offs.

3.4.2 Using Eligibility for Reliability Improvement

In general, the problem of improving reliability is reflected in search of a minimum level of reliability required by the target application.

Reliability improvement can be achieved by acting at different abstraction levels and granularities. Addition of spatial or temporal redundancy inherent in the process of increasing reliability at logic level generates extra costs of which minimization is important. In other words, the designer's challenge is how to achieve the reliability goal while adding as less overhead as possible.

The proposed concept of eligibility helps the designer to explore the space of solutions for reliability improvement. Indeed, this space is potentially very broad and in some cases it would be unrealistic to explore it exhaustively. The workflow of Figure 3.8 shows how reliability significance analysis can be used to reliability improvement.

The Λ set provides the classification of blocks according to eligibility. More eligible blocks should then be considered to have priorities to implement reliability enhancement techniques such as adding redundancy (TMR).

As in the previous section, assume the results in Table 3.1.b which produces Λ set below.

$$\mathbf{\Lambda} = \{b_1, b_4, b_5, b_6, b_3, b_7, b_2, b_8\}$$

We can see that reliability improvement should begin by reliability improvement of block b_1. If the resulting reliability (R_1^*) is inferior to those required (R_{req}), the reliability improvement procedure should be iterated by involving more and more blocks (firstly, b_4, then b_5, and so on). The last block to be taken into account should be b_8.

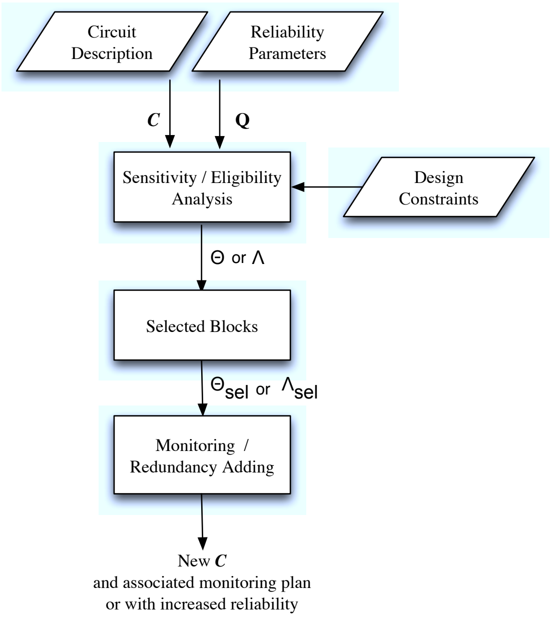

Circuit Description
Reliability Parameters
C
Q
Sensitivity / Eligibility Analysis
Design Constraints
Θ or Λ
Selected Blocks
Θsel or Λsel
Monitoring / Redundancy Adding
New C
and associated monitoring plan
or with increased reliability

Figure 3.8: Workflow of reliability significance concept applied to monitoring and reliability improvement

In the next chapter, we are going to propose the Progressive Modular Redundancy (PMR) method which makes use of the block grading concept.

Chapter 4

Progressive Approach for Modular Redundancy

4.1 Introduction

Full hardware redundancy implies huge area overheads. Take the TMR technique for example, it results in 3X overhead in area. Motivated by the need of economical designs, designers are constantly looking for ways and methods to alleviate the area overhead while keep the reliability improvement at an accepted level. What we are pursuing is to maximize the reliability by introducing as less modular redundancy as possible. Prior works like [65, 81] proposed methods to establish possible candidate redundant structures by TMR under a given design parameter (like reliability requirement and cost constraint). It has to traverse all possible TMR architectures under a given cost constraint and then choose the architectures that meet the reliability requirements. And the aforementioned Selective TMR in [63, 64] defines the sensitive logic gates under given criteria and equip them with TMR to realize partition protection. It is realized by utilizing characteristics of logic gates but it missed the combinational logic masking ability. Automatic TMR [68] considers this problem as a graph partitioning problem. It divides the original circuit into two parts, one with TMR and the other without. And then the partition adjusts according to some criteria until it converges. However, the convergence is time-consuming and unforeseeable. The Selective Fault Tolerance in [69] which protects special input signals is efficient but it needs to select the inputs to be protected. Furthermore, Functional redundant wires proposed in [4] solves the problem very economically by digging out logic implications in digital IPs. However, it needs to analyze all logic implications which is a tough journey and it highly depends on how frequently the logic implications happen.

Based on the block grading concept addressed in the last chapter, the *Progressive Module Redundancy* (PMR) method is proposed and evaluated in this chapter. Fault-tolerant architectures and design workflow based on proposed methodology are then addressed in details. Considering the reliability improvement techniques, this method is not constrained on TMR but extends to QMR (5MR). Experiment results based on a series of circuits demonstrate its advantages in efficiency, reliability and cost. The proposed method points out a new direction of economical redundant fault-tolerant designs. Comparison with the state of the art, advantages and limitations are discussed in the end.

4.2 Progressive modular redundancy approach

The proposed method tries to find a balance between the reliability improvement ΔR and the overhead ΔC generated by this reliability improvement. The analysis of this problem depends on the improvement objective:

– determine the architecture that can satisfy a constraint of reliability, while minimizing the overhead. It means minimizing of ΔC, while respecting $R \geq R_{min}$.

– determine the best gain in reliability due to a maximum-tolerant additional cost. It means maximizing of ΔR, while respecting $\Delta C \leq \Delta C_{max}$.

Whatever the considered cases above, we propose to solve this problem step by step, that is in a *progressive* way. The basic idea is to act progressively on the blocks:

– starting with improving the reliability of a single block, then two blocks, and so on until cover all the blocks.

– starting with improving the reliability of the blocks with inexpensive technique, then gradually move to techniques increasingly expensive (and so increasingly efficient).

The order of which improvement techniques will be adopted comes from equation (3.14). This could be relatively easy to determine, especially if only one parameter (A:area) is considered due to independence property of area cost. On the other hand, determine the order for executing in blocks is possible with their weights.

For easy illustration of the proposed method, we still consider a circuit $\mathcal{C}$ with K blocks (b_1, b_2, ... , b_K). Assume the design criteria is to reach a required reliability R_{req}, while respecting the overhead should be $\Delta C \leq \Delta C_{max}$. The workflow corresponding to the proposed method is described in Figure 4.1.

Description defines how the blocks b_i are connected together to form the circuit. This can be a structural HDL description of $\mathcal{C}$. *Library* provides useful parameters of each block $b_i \in \mathcal{C}$ like area A_i, power consumption P_i, delay T_i and reliability q_i. Main procedures are explained below.

– $\mathbf{RA}(\mathcal{C})$ refers to reliability analysis of the original circuit $\mathcal{C}$. It is done to check whether $\mathcal{C}$ already meets the constraints. If $R(\mathcal{C}) \geq R_{req}$, the original $\mathcal{C}$ is selected.

– $\mathbf{BG}$ grades the blocks according to their different weights. All the constituent blocks are then donated with the parameter (weight) $w_i \in [1, K]$ to decide their priorities when adding redundancy.

– $\mathbf{PI}$ stands for reliability improvement with progressive modular redundancy. Improvements

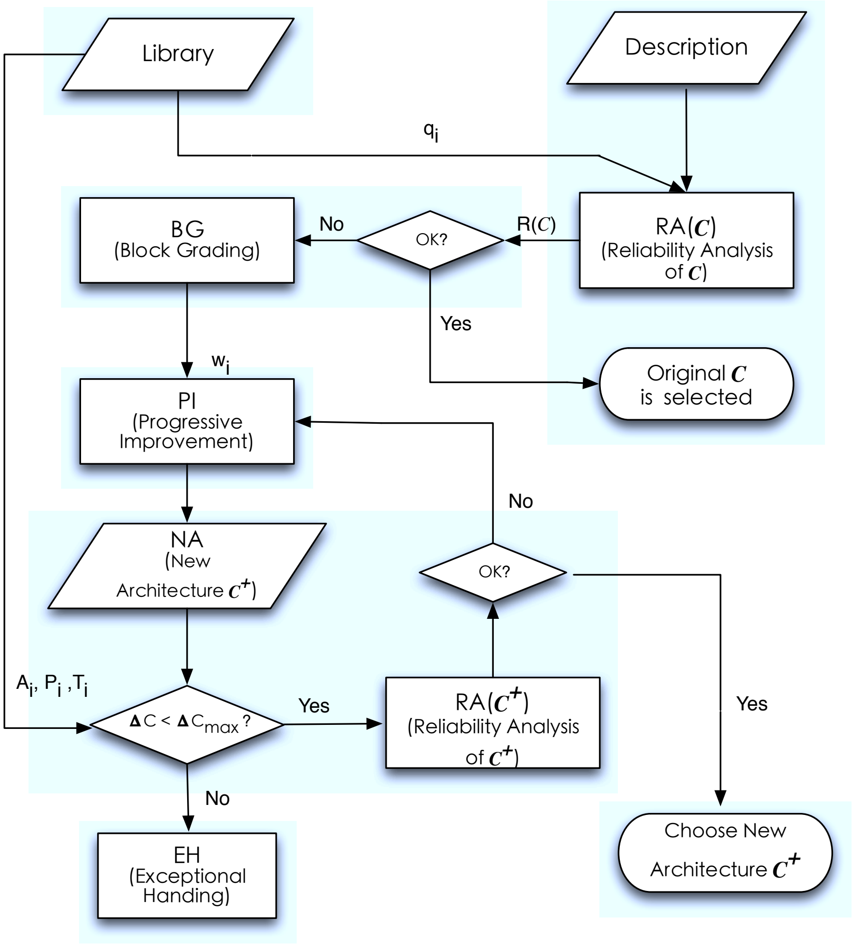


Figure 4.1: Workflow for the proposed method.

are implemented in an iterative way as shown in Table 4.1 and Table 4.2.

- **NA** are new versions of $\mathcal{C}$ related to use of progressive *TMR* ($\mathcal{C}_{PTMR}$) or progressive *MMR* ($\mathcal{C}_{PMMR}$).
- **RA**$(\mathcal{C}^+)$ refers to reliability analysis of the new architectures that have passed cost constraint checking.
- **EH** functions when both of the new architectures $\mathcal{C}_{PTMR}$ and $\mathcal{C}_{PMMR}$ exceed the area cost limit. It either relaxes ΔC_{max} or selects the architecture of the last iteration.

4.2.1 Progressive triple module redundancy: PTMR

Now we present the method for the procedure **PI** in Figure 4.1. We first consider representative technique TMR in a progressive manner. The idea is to add redundancy first on the blocks that have higher weights (because they conduce to higher reliability improvements) and then (if necessary) on the blocks with lower weights.

The values w_i which are required for the proposed method can be obtained by comparing the results of equation (3.8) for each block b_i under the same technique. Notice that sometimes w_i come directly from circuit's topology, as illustrated in Section 3.2.1.

According to the proposed method, only the block with the highest w_i benefits from redundancy adding in the first step. If the obtained reliability improvement is considered insufficient compared with the requirements, a second block is considered for redundancy adding (this is the block with the second highest w_i value). The procedure is carried on until the reliability requirement is satisfied or maximum redundancy is used. Table 4.1 shows how new architectures are produced under this progressive method. The circuit is supposed to have K blocks. Redundancy adding is performed with triple modular redundancy (TMR or 3MR), so it brings in the progressive TMR (*PTMR*) approach. In this table, the blocks are ordered according to their weights (w_i) and m denotes the execution steps of the method. The values in the cells represent degrees of redundancy (1= no redundancy, 3 = triple modular redundancy). The circuit architecture corresponding to step m is obtained by using TMR on the m blocks for which $w_i \in [K - m + 1, K]$.

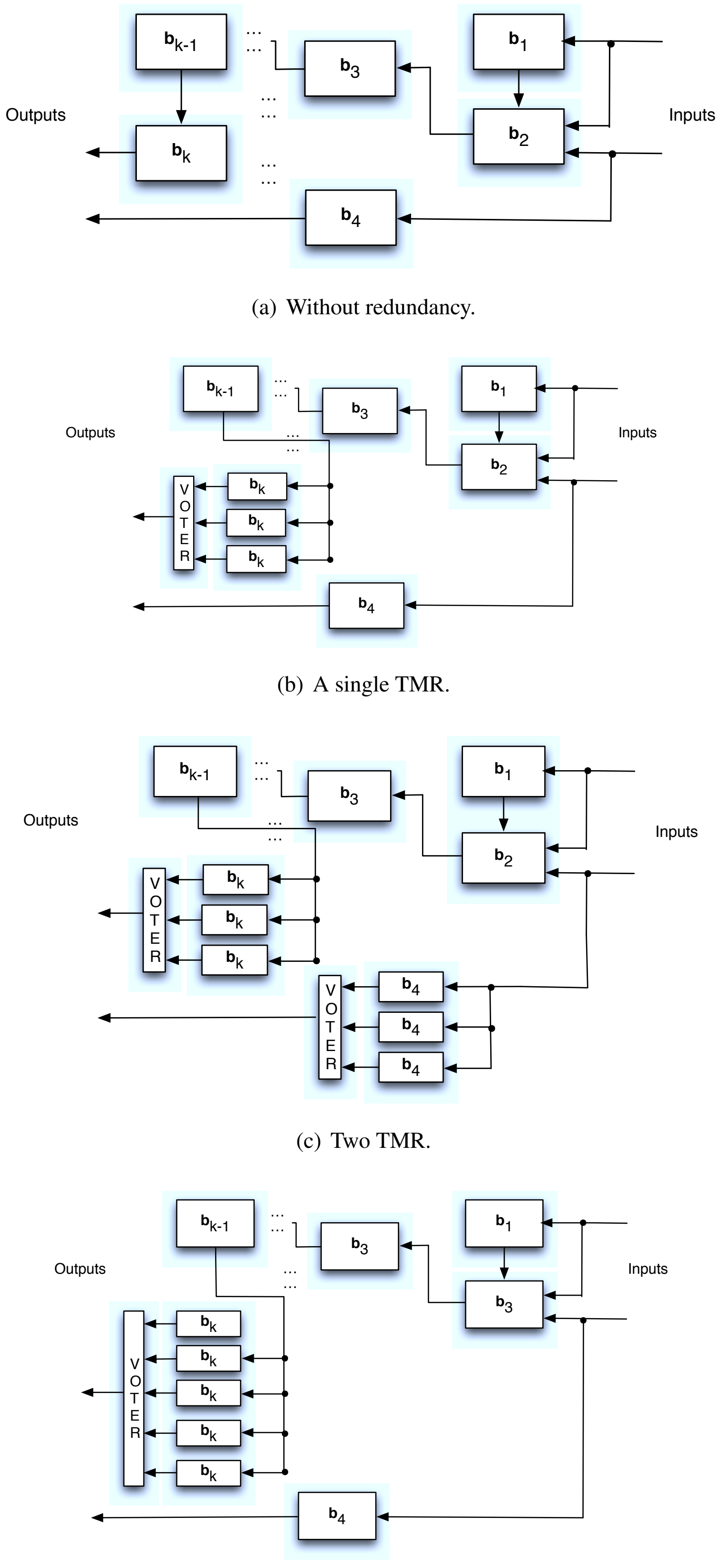
Table 4.1: Algorithm execution for PTMR.

	w_i							
m	K	$K-1$	$K-2$	$\cdots$	$K-m$	$\cdots$	2	1
0	1	1	1	1	1	1	1	1
1	3	1	1	1	1	1	1	1
2	3	3	1	1	1	1	1	1
3	3	3	3	1	1	1	1	1
4	3	3	3	3	1	1	1	1
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
K	3	3	3	3	3	3	3	3

4.2.2 Progressive mixed module redundancy: PMMR

We have also explored the use of combined triple and quintuple modular redundancy with progressive redundancy adding. This involves the progressive mixed modular redundancy (*PMMR*) which is an extension of *PTMR*. Execution of *PMMR* method is illustrated in Table 4.2 for $K = 8$ where two new architectures are generated iteratively for comparison. This time, there are three redundancy degrees taken into account (1= no redundancy, 3 = triple modular redundancy, 5 = quintuple modular redundancy).

Considering the ordered set of blocks, the first one has weight K and the last one has weight 1. In addition to architecture based on *PTMR*, *PMMR* approach produces a new architecture as follows. When m is odd, this comes from QMR on the first p blocks ($p = \lfloor \frac{m}{2} \rfloor$) and TMR on the next r blocks ($r = m - p$). When m is even, it results in only QMR on the first p blocks.



(a) Without redundancy.

(b) A single TMR.

(c) Two TMR.

(d) Single QMR.

Figure 4.2: Different redundancy architectures of generalized circuits

Table 4.2: Algorithm execution for PMMR where $K = 8$.

	w_i							
m	8	7	6	5	4	3	2	1
0	1	1	1	1	1	1	1	1
1	3	1	1	1	1	1	1	1
2	3	3	1	1	1	1	1	1
	5	1	1	1	1	1	1	1
3	3	3	3	1	1	1	1	1
	5	3	1	1	1	1	1	1
4	3	3	3	3	1	1	1	1
	5	5	1	1	1	1	1	1
5	3	3	3	3	3	1	1	1
	5	5	3	1	1	1	1	1
6	3	3	3	3	3	3	1	1
	5	5	5	1	1	1	1	1
7	3	3	3	3	3	3	3	1
	5	5	5	3	1	1	1	1
8	3	3	3	3	3	3	3	3
	5	5	5	5	1	1	1	1
...	...	...	...	...	...	...	...	...
16	5	5	5	5	5	5	5	5

Figure 4.2 shows some of the candidate redundancy structures based on the proposed method. In this figure, we suppose b_m is the block with the highest weight and b_4 is the second one. Figure 4.2(a), Figure 4.2(b), Figure 4.2(c) and Figure 4.2(d) correspond to the strategies Non Redundancy, Single TMR, Two TMR and Single QMR, respectively.

4.3 Validation of progressive redundancy adding: PTMR and PMMR

According to specified design requirements, we can rank the constituent blocks based on the proposed sensibility or eligibility concepts. With these ranked blocks, we could go further into the progressive approach in reliability improvement.

Here we present two implementations of the proposed progressive redundancy adding method. They have been chosen for their simplicity but the method can be applied to any circuit. Also, for easier illustration and due to the independence property, we consider only one parameter for cost analysis, the area A. Recall the workflow (see Figure 4.1), reliability analysis $\mathbf{RA}(\mathcal{C})$ or $\mathbf{RA}(\mathcal{C}^+)$ are implemented with the SPR_{MP} algorithm described in [46].

4.3.1 PTMR implementation

The PTMR approach has been applied on circuit C17 (Figure 3.5) from ISCAS'85 benchmark [77]. Each NAND gate is supposed to have reliability value $q_i = q = 0.99$ (99%) and the area cost S_{NAND}. Requirements of the project are as follows: minimum required reliability is $R_{req} = 0.97$ (97%) and maximum redundancy area overhead is set as $4 \times S_{NAND}$.

After TMR implementation on each NAND gate, block grading results are those shown in Table 3.6. As one can see in Table 4.3 for $m = 0$, $\mathbf{RA}(\mathcal{C}) = 95.2\% \leq R_{req}$. In the same way, implementation of TMR only on the block NAND5 ($w_5 = 6$) is still insufficient. Finally, for $m = 2$ and TMR on NAND5 and NAND6, $\mathbf{RA}(\mathcal{C}^+) = 97.047\%$ which satisfies the design requirements.

Although the total number of candidate architectures under the same area cost is $C_6^2 = 15$, we find a shortcut to reach the best one.

Table 4.3: Results of *PTMR* approach on C17.

Steps (m)	Architecture	Reliability	Area Cost
0	1-1-1-1-1-1	95.20%	$6S_{NAND}$
1	3-1-1-1-1-1	96.12%	$8S_{NAND} + S_V$
2	3-3-1-1-1-1	97.05%	$10S_{NAND} + 2S_V$

4.3.2 PMMR implementation

We implemented the PMMR approach on the 8-bit ripple carry adder (RCA-8) shown in Figure 4.3. The basic blocks (FA) are supposed to have reliability value $q_i = q = 0.999$ (99.9%) and the area cost S_{FA}. Requirements of the project are as follows: minimum required reliability is $R_{req} = 0.955$ (95.5%) and maximum area overhead is set as $4 \times S_{FA}$. The results are presented in Table 4.4.

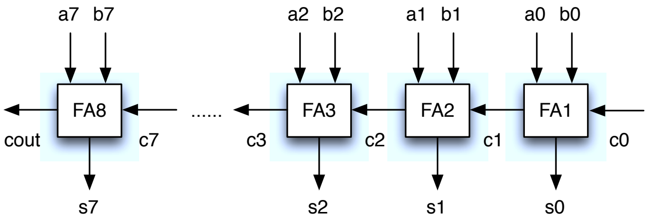


Figure 4.3: A 8-bits carry ripple adder.

Before executing the algorithm of PMMR, we discuss about the reliability evaluation in a ripple carry adder as shown in Figure 4.3. Equation 4.1 shows the probabilistic transfer matrix (PTM) of a full adder model (see Figure 4.4), where p is the failure rate and $p = 1 - q$. Equation 4.2 shows the input probabilities of a given FA with inputs a, b and c in each stage, as shown in Figure 4.4.

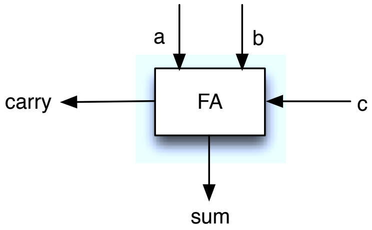


Figure 4.4: A full adder block model.

$$\begin{array}{cc} Input\ cases & Output\ cases\ (carry, sum): \\ (b,a,c): & \begin{array}{cccc} 00 & 01 & 10 & 11 \end{array} \\ \begin{array}{c} 000 \\ 001 \\ 010 \\ 011 \\ 100 \\ 101 \\ 110 \\ 111 \end{array} & \begin{bmatrix} 1-p & p/3 & p/3 & p/3 \\ p/3 & 1-p & p/3 & p/3 \\ p/3 & 1-p & p/3 & p/3 \\ p/3 & p/3 & 1-p & p/3 \\ p/3 & 1-p & p/3 & p/3 \\ p/3 & p/3 & 1-p & p/3 \\ p/3 & p/3 & 1-p & p/3 \\ p/3 & p/3 & p/3 & 1-p \end{bmatrix} \end{array} \tag{4.1}$$

$$INPUT = \begin{bmatrix} a_0 & a_1 \\ a_2 & a_3 \end{bmatrix} \otimes \begin{bmatrix} b_0 & b_1 \\ b_2 & b_3 \end{bmatrix} \otimes \begin{bmatrix} c_0 & c_1 \\ c_2 & c_3 \end{bmatrix} \tag{4.2}$$

In each stage, output probabilities of sum and carry could be computed as follows. By using the block matrix multiplication, Input probabilities must be multiplied by the PTM of full adder, and then we have the matrix as shown in Figure 4.5.

$$\begin{bmatrix} a_0b_0c_0(1-p)+a_0b_0c_1p/3 & a_0b_0c_0p/3+a_0b_0c_1(1-p) & a_0b_0c_0p/3+a_0b_0c_1p/3 & a_0b_0c_0p/3+a_0b_0c_1p/3 \\ a_0b_0c_2(1-p)+a_0b_0c_3p/3 & a_0b_0c_2(1-p)+a_0b_0c_3p/3 & a_0b_0c_2p/3)+a_0b_0c_3p/3 & a_0b_0c_2p/3+a_0b_0c_3p/3 \\ a_0b_3c_0p/3+a_0b_3c_1(1-p) & a_0b_3c_0(1-p)+a_0b_3c_1p/3 & a_0b_3c_0p/3+a_0b_3c_1(1-p) & a_0b_3c_0p/3+a_0b_3c_1p/3 \\ a_0b_3c_2p/3+a_0b_3c_3p/3 & a_0b_3c_2(1-p)+a_0b_3c_3p/3 & a_0b_3c_2(1-p)+a_0b_3c_3p/3 & a_0b_3c_2p/3+a_0b_3c_2p/3 \\ a_3b_0c_0p/3+a_3b_0c_1p/3 & a_3b_0c_0(1-p)+a_3b_0c_1p/3 & a_3b_0c_0p/3+a_3b_0c_1(1-p) & a_3b_0c_0p/3+a_3b_0c_1p/3 \\ a_3b_0c_2p/3+a_3b_0c_3p/3 & a_3b_0c_2(1-p)+a_3b_0c_3p/3 & a_3b_0c_2(1-p)+a_3b_0c_3p/3 & a_3b_0c_2p/3+a_3b_0c_3p/3 \\ a_3b_3c_0p/3+a_3b_3c_1p/3 & a_3b_3c_0p/3+a_3b_3c_1p/3 & a_3b_3c_0(1-p)+a_3b_3c_1p/3 & a_3b_3c_0p/3+a_3b_3c_1(1-p) \\ a_3b_3c_2p/3+a_3b_3c_3p/3 & a_3b_3c_2p/3+a_3b_3c_3p/3 & a_3b_3c_2(1-p)+a_3b_3c_3p/3 & a_3b_3c_2p/3+a_3b_3c_3(1-p) \end{bmatrix}$$

Figure 4.5: Inputs multiply by PTM of a full adder.

According to Figure (4.5), in each stage, we can yield the probabilities of sum and carry as below:

$$p(sum) = p(carry) = [(c_0 + c_3)(a_0b_0 + a_3b_3) + (c_0 + c_2)(a_0b_3 + a_3b_0)](1 - p) \\ +[(c_1 + c_2)(a_0b_0 + a_3b_3) + (c_1 + c_3)(a_0b_3 + a_3b_0)]p/3 \tag{4.3}$$

The purpose of presenting the reliability evaluation of the adder is to remind that, it is a cascade structure but it is not exactly the same as the examples we discussed in Chapter 3. It should be noted that here, not all outputs of the previous stage are considered as inputs of the following stage. This results in the reliability evaluation a bit more complicated. By the way, based on the PTM method stated above, a probabilistic analysis of a multi-bit ripple carry adder which is implemented on quantum-dot cellular automata nanotechnology is presented in [82] where a second degree polynomial approximation of the RCA reliability is extracted based on PTM.

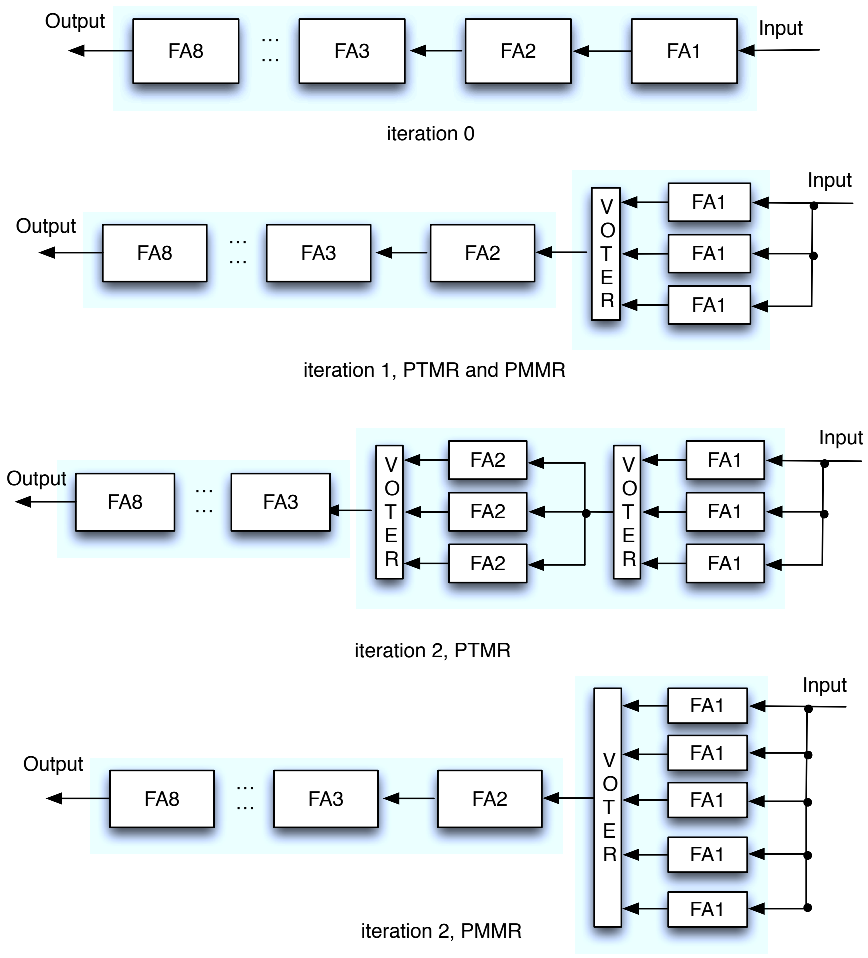
Now let us see how the workflow works on this ripple carry adder. In the first stage (i.e. FA1), a, b and c are all fault-free. Afterwards, starts from the second stage, one of the input c becomes fault-prone but we assume inputs a and b are always fault-free. If we implement NMR in some of the full adder models (FA_i where $i \in [1, 8]$), the p changes according to the block reliability q_i. For example, in iteration 2 of algorithm PMMR as shown in Figure 4.6, there will be two architectures generated there. If we put TMR in FA1 and FA2, then $q_3 = 3q^2 - 2q^3$ and thus $p = 1 - q_3$ in PTMs of FA1 and FA2 when we calculate the probability of the corresponding signals. If we put 5MR in FA1, then $q_5 = 6q^5 - 15q^4 + 10q^3$, so brings $p = 1 - q_5$ in the PTM of FA1.

Because RCA-8 is a cascade structure and all blocks have the same reliability, there is no need to analyze equation (3.8). In fact, block weights are given as $w(FA_i) = 8 - i + 1$ since it is a cascade circuit structure.

For $m = 0$, $\mathbf{RA}(\mathcal{C}) = 94.06\% \leq R_{req}$. Redundancy adding is performed according to Table 4.2. The first iteration generates one architecture with triple modular redundancy at $FA1$. The reliability of this new architecture is $R = 94.85\% \leq R_{req}$. The second iteration generates one architecture with triple modular redundancy at two blocks ($FA1$, $FA2$), and another architecture with quintuple modular redundancy at block $FA1$. Both architectures satisfy reliability requirements, so we select the architecture which results in less area cost ($S = 12 * S_{FA} + S_V$). Notice that mixed modular redundancy is a better solution than triple modular redundancy in this case. MMR brings into higher reliability and less area cost. Therefore, if R_{req} is set higher, for example, 96%, we don't need further iterations, either. By the way, further steps or procedure **EH** may be required if R_{req} is set higher or ΔC_{max} is set smaller.

4.3.3 Remarks

A decision element is needed in NMR circuits. A design of a fault-tolerant majority voter has been proposed in [83]. Although it is favorable for applications based on transistors, it should be noted that



iteration 0

iteration 1, PTMR and PMMR

iteration 2, PTMR

iteration 2, PMMR

Figure 4.6: Iterations of PMMR implemented on RCA8.

Table 4.4: Results of *PMMR* approach on RCA-8.

Steps (m)	Architecture(s)	Reliability	Area Cost
0	1-1-1-1-1-1-1-1	94.06%	$8S_{FA}$
1	3-1-1-1-1-1-1-1	94.85%	$10S_{FA} + S_V$
2	3-3-1-1-1-1-1-1	95.64%	$12S_{FA} + 2S_V$
	5-1-1-1-1-1-1-1	97.14%	$12S_{FA} + S_V$

the performance gain by the NMR is possible to be degraded by the complexity of the majority voter circuit, especially at the high degree of redundancy factor [84]. To avoid this degradation, a simple structure of a single majority gate has been proposed for possible implementations in nanoelectronic regime [73, 84]. With the use of single majority voters in our work, an improved circuit reliability is obtained, and furthermore, complexity of the voter is kept similar for TMR and 5MR, i.e., area cost for voters in TMR or 5MR are the same: $S_V(TMR) = S_V(5MR) = S_V$.

Here is a remark about the simplification of PMMR in the circuits with a cascade structure in all the blocks are identical, such as the n-bit ripple carry adder aforementioned. In this case, we only need to generate one new architecture in each step, except step 2. This is because comparison of $\mathcal{C}_{PTMR}$ with $\mathcal{C}_{PMMR}$ in step 2 could be used as a criterion in the following steps. Notice that $\mathcal{C}_{PTMR}$ and $\mathcal{C}_{PMMR}$ have the same area cost in each step (the voter adopted is a single gate described in [73] and therefore is not taken into account), so we make the choice according to reliability which is the unique parameter. In step 2, $\mathcal{C}_{PTMR} = \mathcal{C}_{2TMR}$ while $\mathcal{C}_{PMMR} = \mathcal{C}_{5MR}$. After comparing the reliabilities of them, we can generate only one new version in the following steps (either $\mathcal{C}_{PTMR}$ or $\mathcal{C}_{PMMR}$). Since we can find that in step m where m is even and $m \geq 2$, the comparison of PTMR and PMMR is equal to compare $\frac{m}{2}$ pairs of $\mathcal{C}_{2TMR}$ and $\mathcal{C}_{5MR}$. Similarly, in m-th step where m is odd and $m \geq 3$, the comparison of PTMR and PMMR equals to compare $\lfloor \frac{m}{2} \rfloor$ pairs of $\mathcal{C}_{2TMR}$ and $\mathcal{C}_{5MR}$. Consequently, circuits in this kind of structure imply an efficient path.

4.3.4 More examples and comparisons

In order to limit the area overhead due to fault tolerance improvement with TMR, a selective TMR (STMR) technique has been proposed in [64]. This method is based on the logic masking analysis of each gate. Then, selective insertion of TMR is made in the gates that are more vulnerable to faults. As a matter of fact, the reliability of the circuit not only depends on the logic masking of the gates but also relates with the combinational logic. The sensitivity and eligibility analysis in this work take combinational masking into account and show that it is more efficient in fault-tolerant designs.

For example, in the circuit C17, Selective TMR only considered gate properties [64] and decided to implement TMR on gates NAND3, NAND4, NAND5 and NAND6. However, under the same area cost, the proposed method [66] gave us another solution. It chooses gates NAND5, NAND6, NAND3 and NAND2 to implement TMR. This alternative increases the overall reliability compared to the STMR solution. The reliability calculation used here is based on the method SPR_{MP} [46].

We compare the proposed method with anther similar technique in [68]. The ISCAS circuit 74283 fast adder is taken as an example. If the redundant factor is 5 (5 blocks could be configured with TMR), gates g_{32}, g_1, g_3, g_0 and g_9 are selected by the proposed method PTMR as the five candidates to be hardened. The method presented in [68], under the same area overhead constraint, applies TMR in gates g_{32}, g_{36}, g_{37}, g_{38} and g_{39}. The proposed method presented a better reliability gain.

Comparison results related to the above statements are shown in Table 4.5, where TC, IC OR, R stand for Total Cells, Improved Cells, Original Reliability and Reliability, respectively.

Table 4.5: Results and comparisons of Progressive Modular Redundancy.

Circuit	TC	IC	OR	R by [64]	R by [4]	R by PMR
C17	6	4	95.2%	98.6%	–	98.8%
74283	40	5	68.9%	–	72.2%	72.6%

4.4 Critical errors mitigation in digital IPs

In a lot of digital applications, not every bit carries the information of the same degree of importance. A more aggressive fault-tolerance improvement is proposed in practical applications with related to the bit significance. A novel TMR configuration is proposed to prevent the critical errors degrading the performance of the digital circuits. With this configuration and a more practical metric in reliability evaluation, we have proved this proposed method provides better solutions than the recent similar proposals.

4.4.1 Cost function and TMR configuration

In previous works, errors at the outputs are considered equally in reliability evaluation, such as in [10, 12, 13, 49, 56]. In other words, number and position of errors have the same extend of cost. It equals to consider such a cost function in mathematical optimization.

Suppose the output of a M-bit digital IP is $y = y_0y_1 \cdots y_{M-1}$ which is the value under the fault-free condition, a deterministic value. If faults and errors exist, the experimental value is $Y = Y_0Y_1 \cdots Y_{M-1}$.

If we don't consider any bit significance in digital applications, the cost function could be defined as in (4.4).

$$C_{nominal}(Y, y) = 1 - \prod_{i=0}^{M-1} \delta(Y_i - y_i) = \begin{cases} 0 & \text{for all } Y_i = y_i \\ 1 & \text{any } Y_i \neq y_i \end{cases} \tag{4.4}$$

This cost function (4.4) assigns equal cost to all the output bits and it is not desirable. Ideally, a desirable cost function should assign progressively larger cost to MSB(s). In this case, we define the weighted cost function as in (4.5).

$$C_{weighted} = \sum_{n=0}^{M-1} 2^n \cdot C_n(Y, y) \qquad C_n(Y, y) = 1 - \prod_{i=0}^{M-1} \delta(Y_i - y_i) \tag{4.5}$$

In some situations, we have some dispensable bits, so we could also define a truncated cost function as in (4.6).

$$C_{truncated} = \sum_{n=k}^{M-1} \cdot C_n(Y, y) \tag{4.6}$$

where k is the coarse scale.

Truncated cost function could also be combined with the weighted one in some applications that include both binary bits as well as nonsignificant bits. That is expressed in (4.7).

$$C_{hybrid} = \sum_{n=k}^{M-1} 2^n \cdot C_n(Y, y) \tag{4.7}$$

4.4.2 Practical metric for reliability evaluation

According to the aforementioned discussion about cost functions, Practical reliability is proposed as in (4.8) based on cost function (4.5). It is a metric that can take into account the importance of each output bit when analyzing the reliability of a circuit.

$$R_{practical} = \prod_{i=0}^{M-1} R_i^{k_i} \tag{4.8}$$

$$k_i = \frac{1}{2^{(M-1)-i}} \tag{4.9}$$

The weight factor k_i allows the designer to control the importance of a specific output bit b_i to the output of the circuit. Notice that if $k_i = 1$ for all $0 \leq i \leq M - 1$, the practical reliability expression (4.8) becomes the nominal reliability expression (4.10) which corresponds to cost function (4.4). In this thesis, a standard binary representation is considered so that k_i is calculated as shown in (4.9). Note that (4.8) can also be related to the probability that an error will cause a significant disparity on the output of a circuit (a critical error).

$$R_{nominal} = \prod_{i=0}^{M-1} R_i \tag{4.10}$$

Here is an example for better understanding the practical reliability. Let us suppose that a designer obtained three different architectures for a 4-bit adder in which the output is coded using a binary scheme. Besides, he has to select one among them taking into account the reliability of the output. The reliabilities for the output bits of such architectures are presented in Table 4.6.

Analyzing the nominal reliability values for the obtained architectures, *Architecture 1* and *Architecture 2* are selected as the best solutions. Indeed, no distinction can be made between these two architectures regarding the nominal reliability value. However, as the output of this circuit is coded

Table 4.6: Reliability for the output bits of three structures for a 4-bit adder

Architecture	b_3	b_2	b_1	b_0	$R_{nominal}$	$R_{practical}$
1	99%	99%	99%	95%	92.18%	97.63%
2	95%	99%	99%	99%	92.18%	94.17%
3	98%	99%	99%	95%	91.25%	96.64%

using a binary scheme, the first architecture will provide better results (smaller disparities) than the second. Ideally, a more desirable analysis should take into account the amount of information each bit of an output carries (or its importance) in order to assign progressively great costs to them. In order to tackle this problem, a new metric to analyze the reliability of a circuit with a multiple-bit output is presented in next section.

4.4.3 Selectively applying TMR based on bit significance

The previous sections addressed the Block grading concept and Progressive Modular Redundancy approach where the constituent blocks are ranked by their different weights. Here the critical gate is proposed which takes into account not only the probability that an error will be introduced by a gate, but also how critical this error will be for the target application.

We illustrate the problem by a familiar example, 74283 fast adder circuit. The fault injection emulation is performed in order to detect the critical factors. The idea is to inject a single fault in a gate g_i and analyze the output for all the possible input vectors. Then, for each output bit b_z, the number of errors S_z related to a single fault in g_i is evaluated (see Table 4.7). The columns S_{z_w} correspond to weighted versions of S_z. In our case study, as a standard binary representation is considered, S_{z_w} is obtained as shown in (4.11). Notice that there are 2^9 possible input logic values for each faulty gate. All the simulation results are shown in Table 4.7.

$$S_{w_z} = 2^z \cdot S_z \tag{4.11}$$

The **critical gates** are detected according to the results presented in Table 4.7. The more critical the gates are, the higher priorities they receive to be protected (in this case using TMR). Configuration of TMR based on this principle is more efficient in practical applications as will be shown after.

In fact, critical factors are assigned to the gates according to the number of weighted errors in Table 4.7. If the number of weighted errors equals, gates that are nearer the primary outputs receive higher priorities. If the numbers of weighted errors and the distance to the primary outputs are both identical, gates presenting more reconvergent fanouts are considered more critical. Gates for which these three parameters are equal receive the same critical factor. Notice that the rightmost column in Table 4.7 gives the critical factor for a gate g_i. The higher the factor number is, the more critical the

Table 4.7: Error analysis for the gates of 74283 fast adder

g_i	S_0	S_{0_w}	S_1	S_{1_w}	S_2	S_{2_w}	S_3	S_{3_w}	C_4	C_{4_w}	$\sum errors_{weighted}$	$CriticalFactor$
0	0	0	0	0	0	0	384	3072	192	3072	6144	36
1	0	0	0	0	0	0	384	3072	320	5120	8192	38
2	0	0	0	0	384	1536	192	1536	96	1536	4608	33
3	0	0	0	0	384	1536	320	2560	160	2560	6656	37
4	0	0	384	768	192	768	96	768	48	768	3072	25
5	0	0	384	768	320	1280	160	1280	80	1280	4608	32
6	384	384	192	384	96	384	48	384	24	384	1920	14
7	384	384	320	640	160	640	80	640	40	640	2944	23
8	512	512	256	512	128	512	64	512	32	512	2560	22
9	0	0	0	0	0	0	0	0	320	5120	5120	35
10	0	0	0	0	0	0	0	0	288	4608	4608	34
11	0	0	0	0	0	0	0	0	272	4352	4352	31
12	0	0	0	0	0	0	0	0	264	4224	4224	29
13	0	0	0	0	0	0	0	0	272	4352	4352	31
14	0	0	0	0	0	0	512	4096	0	0	4096	27
15	0	0	0	0	0	0	384	3072	0	0	3072	24
16	0	0	0	0	0	0	320	2560	0	0	2560	21
17	0	0	0	0	0	0	288	2304	0	0	2304	20
18	0	0	0	0	0	0	272	2176	0	0	2176	18
19	0	0	0	0	0	0	288	2304	0	0	2304	20
20	0	0	0	0	512	2048	0	0	0	0	2048	17
21	0	0	0	0	384	1536	0	0	0	0	1536	13
22	0	0	0	0	320	1280	0	0	0	0	1280	12
23	0	0	0	0	288	1152	0	0	0	0	1152	10
24	0	0	0	0	320	1280	0	0	0	0	1280	12
25	0	0	512	1024	0	0	0	0	0	0	1024	7
26	0	0	384	768	0	0	0	0	0	0	768	6
27	0	0	320	640	0	0	0	0	0	0	640	4
28	0	0	384	768	0	0	0	0	0	0	768	6
29	512	512	0	0	0	0	0	0	0	0	512	2
30	384	384	0	0	0	0	0	0	0	0	384	0
31	512	512	0	0	0	0	0	0	0	0	512	1
32	0	0	0	0	0	0	0	0	512	8192	8192	39
33	0	0	0	0	0	0	512	4096	0	0	4096	27
34	0	0	0	0	512	2048	0	0	0	0	2048	15
35	0	0	512	1024	0	0	0	0	0	0	1024	8
36	0	0	0	0	0	0	512	4096	0	0	4096	28
37	0	0	0	0	512	2048	0	0	0	0	2048	16
38	0	0	512	1024	0	0	0	0	0	0	1024	9
39	512	512	0	0	0	0	0	0	0	0	512	3

gate will be. In this work critical factors are assigned as integers $\in [0, 39]$.

4.4.4 Comparison

In this section, implementation of block grading takes into account not only the probability of error occurrence, but also the impact of such error on the system. Compared with the automatic insertion of TMR [68] as shown in Table 4.8, the reliability enhancement is more meaningful and the weighted approach is of great use for practical applications.

It can be also noted that, under the same area overhead, the nominal reliability increases by almost the same amount with both methods (see Figure 4.7). In fact, nominal reliability assigns equal reliability costs to the output bits of the ISCAS 74283. This means that the output bits are considered as having the same importance to the system, so that the nominal reliability value does not distinguish in

Table 4.8: Reliability comparison of 74283 fast adder.

Reliability	No hardening	Method in [68]	Proposed
S_0	94.07%	94.97%	94.07%
S_1	92.39%	93.26%	92.39%
S_2	91.80%	92.65%	92.43%
S_3	91.33%	92.17%	93.07%
S_4	94.60%	95.51%	97.15%
$R_{nominal}$	68.93%	72.24%	72.63%
$R_{practical}$	87.29%	88.89%	90.65%

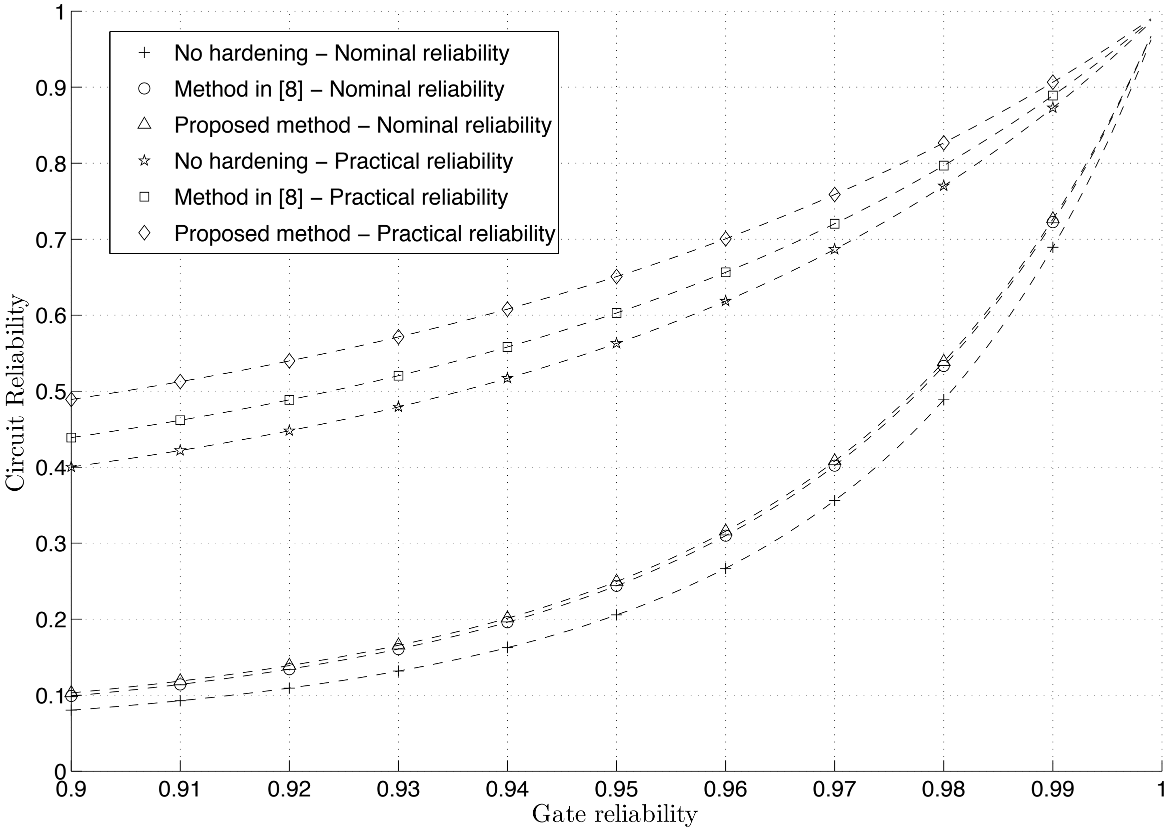


Figure 4.7: Simulation results for the ISCAS 74283

which bit the reliability was actually increased. In spite of that, practical reliability results can handle this problem, and can indeed provide a sharper distinction between this two hardened architectures as shown in Figure 4.7.

Chapter 5

Error Characteristics of Majority Voter

5.1 Introduction

Voters are frequently used in fault-tolerant designs. This chapter presents an overview of voters. It starts by introduction of voting mechanisms and voter structures. As majority voter plays an important role in both conventional fault-tolerant design and novel nanoelectronic systems, a better understanding of signal probability, functional/signal reliability and error bound of majority voter is discussed in this chapter. We analyze these parameters by boolean difference [14]. The result derived shows the characteristics of error propagations in majority voter. More importantly, it reveals the conditions that TMR technique requires. The results show the critical importance of error characteristics of majority voter, as used in fault-tolerant designs.

5.2 Voters

5.2.1 Introduction of voters

A voter receives inputs from an M-of-N system and generates a representative output. It works in a lot of applications where fault tolerance is required. In digital applications, the simplest voter is the one that does a *bit-by-bit* comparison of the outputs. The *bit-by-bit* comparison has some restrictions. It only works when every functional module generates an output that matches the output of every other functional module, also bit by bit. It requires all the replicated modules have mutually synchronized clocks.

If the modules are the same in function but different in hardware of software, we can use the concept called *practically identical* for voting [18]. In other words, what we compare is *word-by-word*. For example, if we define two outputs y_1 and y_2 are *practically identical* if they satisfy $|y_1 - y_2| < \delta$, where δ is a specified threshold. Note that *practically identical* is not transitive.

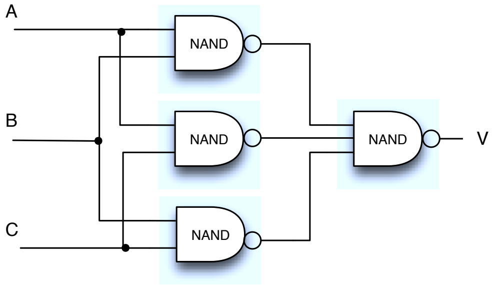


Figure 5.1: Designs of 2-level majority voter.

5.2.2 Voting mechanisms for digital circuits

The most common voting strategies are majority, median, mean (average), and plurality. We explain these voting strategies here.

– Majority voting: It chooses the output as the one produced by more than half of the process.

– Median voting: It chooses the output as this one corresponding to the median value from a collection of results.

– Mean voting: It computes the average of the values produced by the process and then chooses the output as the value closest to this. In digital applications, we could adopt this mechanism only when we do a *word-by-word* comparison.

– Plurality voting: It chooses the output as the value occurs most frequently. A $k - plurality$ voter looks for a set at least k practically identical outputs and picks any of them as the representative.

More details of voting algorithms, voter characters and comparison of voting strategies can be found in [85–87] where how to choose the appropriate voting mechanism under different circumstances are explicated mathematically.

5.2.3 Majority voting configurations

Figure 5.1 shows designs of a 2-level majority voter consisting of four NAND gates or other basic logic gates. This kind of voter structure is favorable for applications based on transistors.

As an example, consider using this voter in a NMR system for example, we can calculate the number of NAND gates in NMR voter for $N = 3, 5, 7, 9$ are $4, 11, 36$ and 127, respectively. It means that voter size grows faster than the redundancy factor. This implies that the performance gain by a higher degree of redundancy may be degraded by an increased complexity of an NMR voter. Therefore, if the redundancy factor $N > 7$, it is not recommended to use this kind of majority voter design [84]. Since if this kind of voter is adopted by a NMR system at a higher redundancy factor situation, we will find that the higher the redundancy factor is, the lower the reliability will be [84].

A recent work proposed a concept of communication grid with the majority property for interconnects to implement a robust NMR system design [88]. This grid adapts *CDMA*, traditionally used in wireless communication, to a nanoscale VLSI chip environment. Using the CDMA mechanism and grid structure, an NMR system without an explicit voter unit is realized. This communication grid is scalable and thus is an option for voter solution in NMR at high redundancy factor.

5.2.4 Single-gate voter structure in Nanoelectronics

In quantum and nanoelectronic circumstances, the implementation of majority logic could be greatly simplified. A simple structure of a single majority gate, as schemed in Figure 5.2. It is a single electron tunneling device as shown in Figure 5.3, and has been adopted for possible implementations in nanoelectronic circuits [73].

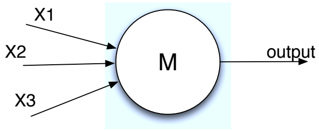


Figure 5.2: Scheme of a single gate majority voter.

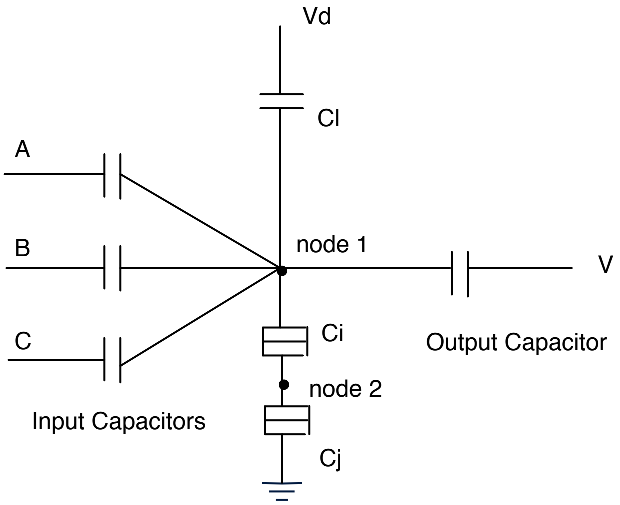


Figure 5.3: Architecture of a single gate majority voter.

With the usage of this single majority voter, if the voter is reliable, the reliability will increase with the redundancy factor.

5.3 Analysis of voter reliability

5.3.1 Motivation

Besides the inherent mechanism of TMR, the efficacy of TMR highly depends on the voter reliability. An imperfect voter could harm system reliability in fault tolerant designs. It is worthy to find the conditions for TMR implementations by considering the error propagation in voters.

Error propagation of combinational circuits is widely analyzed these years, several methods have been proposed such as Probabilistic Transfer Matrix (PTM) [11], Signal Probability (SPR) [12], Information theoretic way [89]. However, PTM and PTM-like methods (for example, SPR) are unable to evaluate the reliability of majority voter in a straightforward way. For example, if we use PTM of the majority logic function to evaluate its signal reliability, the maximum value of signal reliability of the output will be the same as that of the input (under the condition that the majority voter is fault-free, i.e. PTM is an ITM).

We prove the above conclusion in the following.

Equation (5.1) shows the probabilistic transfer matrix (PTM) of a majority voter, where p is the failure rate and $p = 1 - q$. Equation (5.2) shows the probabilities of the 3 inputs a, b and c.

$$\begin{array}{cc} (a,b,c) & \begin{array}{cc} \quad 0 & \quad\quad 1 \end{array} \\ \begin{array}{c} 000 \\ 001 \\ 010 \\ 011 \\ 100 \\ 101 \\ 110 \\ 111 \end{array} & \left[\begin{array}{cc} 1-p & p \\ 1-p & p \\ 1-p & p \\ p & 1-p \\ 1-p & p \\ p & 1-p \\ p & 1-p \\ p & 1-p \end{array}\right] \end{array} \qquad (5.1)$$

$$INPUT = \begin{bmatrix} a_0 & a_1 \\ a_2 & a_3 \end{bmatrix} \otimes \begin{bmatrix} b_0 & b_1 \\ b_2 & b_3 \end{bmatrix} \otimes \begin{bmatrix} c_0 & c_1 \\ c_2 & c_3 \end{bmatrix} \qquad (5.2)$$

Reliability of the output is calculated according to Figure (5.4).

It is obvious that reliability will not greater than the signal reliability of input, the maximum value is when $p = 0$, output reliability equals the input reliability.

5.3.2 Preliminaries

Before our derivation of voter reliability, some preliminaries that will be used later are presented firstly.

$$\begin{bmatrix} a_0b_0c_0 & a_0b_0c_1 & a_0b_0c_0 & a_0b_0c_1 & a_0b_0c_0 & a_0b_0c_1 & a_0b_0c_0 & a_0b_0c_1 \\ a_0b_0c_2 & a_0b_0c_3 & a_0b_0c_2 & a_0b_0c_3 & a_0b_0c_2 & a_0b_0c_3 & a_0b_0c_2 & a_0b_0c_3 \\ a_0b_3c_0 & a_0b_3c_1 & a_0b_3c_0 & a_0b_3c_1 & a_0b_3c_0 & a_0b_3c_1 & a_0b_3c_0 & a_0b_3c_1 \\ a_0b_3c_2 & a_0b_3c_3 & a_0b_3c_2 & a_0b_3c_3 & a_0b_3c_2 & a_0b_3c_3 & a_0b_3c_2 & a_0b_3c_2 \\ a_3b_0c_0 & a_3b_0c_1 & a_3b_0c_0 & a_3b_0c_1 & a_3b_0c_0 & a_3b_0c_1 & a_3b_0c_0 & a_3b_0c_1 \\ a_3b_0c_2 & a_3b_0c_3 & a_3b_0c_2 & a_3b_0c_3 & a_3b_0c_2 & a_3b_0c_3 & a_3b_0c_2 & a_3b_0c_3 \\ a_3b_3c_0 & a_3b_3c_1 & a_3b_3c_0 & a_3b_3c_1 & a_3b_3c_0 & a_3b_3c_1 & a_3b_3c_0 & a_3b_3c_1 \\ a_3b_3c_2 & a_3b_3c_3 & a_3b_3c_2 & a_3b_3c_3 & a_3b_3c_2 & a_3b_3c_3 & a_3b_3c_2 & a_3b_3c_3 \end{bmatrix} \cdot \begin{bmatrix} 1-p & p \\ 1-p & p \\ 1-p & p \\ p & 1-p \\ 1-p & p \\ p & 1-p \\ p & 1-p \\ p & 1-p \end{bmatrix}$$

Figure 5.4: Inputs multiply by PTM of a majority voter.

5.3.2.1 Signal probability

The signal probability of an output is defined as the probability that the output attains a specified value, and similarly, an input probability is the probability that a given input has the specified value. The circuit function effectively provides a transform from input to output probabilities. Indeed, it will be seen that when all input probabilities are set to be either extreme (1 or 0), the probability function of the circuit produces identically the same result as the Boolean function of the circuit. Without loss of generality, signal probability of an input and output of a logic gate is defined as the probability that the signal is logical 1.

Kenneth et al. proposed algorithms for general combinational logic circuits which allow the formulation of an output probability by the given input probabilities [48]. Given independent inputs, Boolean functions could be mapped into algebraic expressions of signal probabilities according to three definitions.

– Definition 1: Boolean "NOT", i.e., $B = \bar{A}$, corresponds to $b = 1 - a$.

– Definiiton 2: Boolean "AND", i.e., $C = AB$, corresponds to $c = a \cdot b$.

– Definition 3: Boolean "OR", i.e., $C = A + B$, corresponds to $c = a + b - a \cdot b$.

By using these above definitions, all types of Boolean logic can be mapped to arithmetic equations of signal probabilities. These are very useful to the practical problems like fault detection and reliability evaluation of logic circuits [49].

5.3.2.2 Errors in the gate and in the signal

Soft errors in a logic gate are classically modeled as a binary symmetric channel (BSC), with a crossover probability ε_g. In other words, following the computation at the gate, the BSC can cause the gate output to toggle symmetrically (from $0 \to 1$ or $1 \to 0$) with the same probability of error ε_g. Each gate has an $\varepsilon_g \in [0, 0.5]$.

Fig. 5.5 shows an arbitrary gate realizing Boolean function *f*. To avoid ambiguity, we use ε_g to express the gate error probability (corresponding to functional reliability). And we denote the

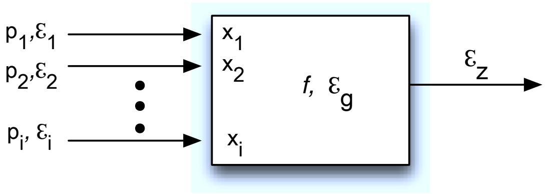


Figure 5.5: Signal and error probability of inputs, error probability of gate and in the outputs.

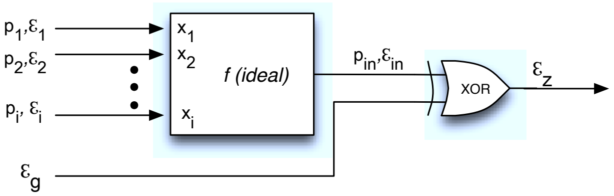


Figure 5.6: Faulty gate model equivalent to Fig. 5.5.

error probability on signal line x_i as ε_i. As stated before, signal probability is defined as $p_i = Pr\{x_i = 1\}$. Therefore, in Fig. 5.5, signal probabilities of the inputs are, $p_1, p_2, \cdots p_i$ while the input error probabilities are $\varepsilon_1, \varepsilon_2, \cdots \varepsilon_i$, the output error probability is ε_z. It should be noticed that, in each input x_i, $p_i \neq 1 - \varepsilon_i$.

Each faulty gate in Fig. 5.5 could be modeled as an ideal gate ($\varepsilon_g = 0$) with the same functionality and another ideal XOR gate as shown in Fig. 5.6. Based on this model, error appears at the output when the input is erroneous while the gate is not, or when the gate is erroneous but the input is not. Therefore, the output error probability is expressed as

$$\varepsilon_z = \varepsilon_{in}(1 - \varepsilon_g) + \varepsilon_g(1 - \varepsilon_{in}) = \varepsilon_g + (1 - 2\varepsilon_g)\varepsilon_{in} \tag{5.3}$$

5.3.3 Minimum functional reliability required for majority voter

In the analysis presented so far, we have assumed that the voter itself cannot fail while it is sometimes untrue. In fact, the voter's unreliability will wipe out the gains of the redundancy scheme. If we add the reliability of the voter, R_v, the system reliability is modified to yield:

$$R_{NMR-FV} = R_v \cdot \sum_{i=\lceil N/2 \rceil}^{N} \binom{N}{i} q_M^i \left(1 - q_M\right)^{N-i} \tag{5.4}$$

To achieve an overall gain, the voting scheme with the imperfect voter must be better than a single element, that is:

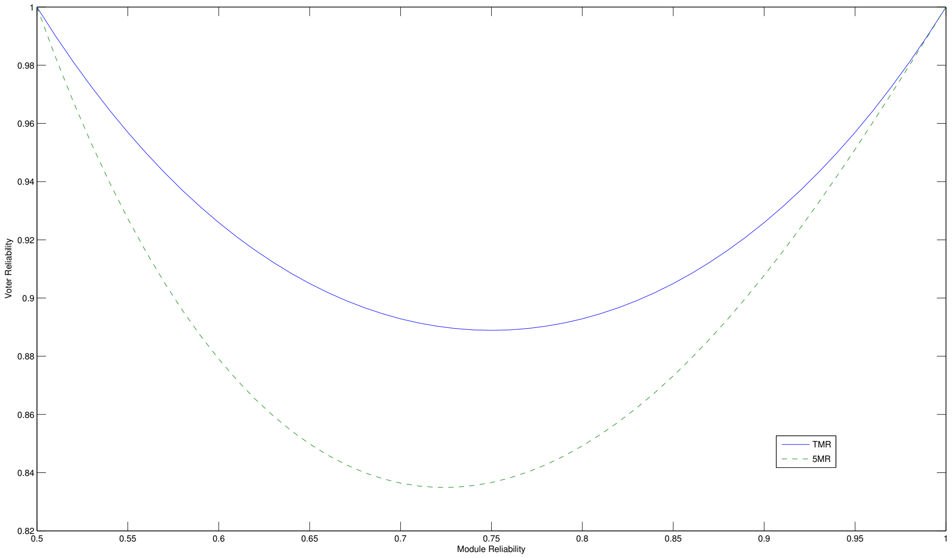


Figure 5.7: Voter reliability required in NMR system.

$$R_{NMR-FV} > q_M \quad or \quad \frac{R_{NMR-FV}}{q_M} > 1 \tag{5.5}$$

This yields the minimum voter reliability required as follow,

$$R_{V_{min}} = \frac{1}{\sum_{i=\lceil N/2 \rceil}^{N} \binom{N}{i} q_M^{i-1} (1 - q_M)^{N-i}} \tag{5.6}$$

Figure 5.7 shows the minimum reliability requirement for a voter in TMR and 5MR. We can also find that a very high voter reliability, close to 1, is needed for small and large module reliability q_M.

Examining Figure 5.7, we could check the minimum value of R_v. Take TMR as an example, R_v will be obtained when the expression $3q_M - 2q_M^2$ reaches its maximum value. Differentiating with respect to q_M and equating to zero yields $q_M = 3/4$, which agrees with Figure 5.7. Substituting this value of q_M into $R_v \cdot q_M(3 - 2q_M) = 1$ yields $R_v = 8/9 = 0.889$. This result has been generalized for N-voter redundancy, and the results are shown in Table 5.1. This table provides lower bounds on voter reliability that are useful during design. It demonstrates that a high reliability of voters needs to be guaranteed by using reliable components or a fault-tolerant design.

Table 5.1: Minimum Voter Reliability.

Redundant factor	3	5	7	9	11	∞
Minimum voter reliability	0.889	0.837	0.807	0.789	0.777	0.75

5.3.4 Signal probability limitation and upper error bound

Combinational circuits have logical, electrical and temporal masking properties, which reduce the probability of propagation and further storage of errors. Functional reliability required by majority voter is therefore a lower bound. Von Neumman has shown that the error bound for a three-input majority logic is $\frac{1}{6}$, i.e. an improvement of reliability could not be achieved through the majority gates when $\varepsilon_g \geq \frac{1}{6}$ [9]. The derivation was based on (5.7). However, it was under his “special case": the probabilities of the inputs (ε_i) are independent and all inputs are considered to be in the state of either logical 1 or logical 0 ($p_i = 0$ or $p_i = 1$).

$$\varepsilon_z = \varepsilon_g + (1 - 2\varepsilon_g)(3\varepsilon^2 - 2\varepsilon^3) \tag{5.7}$$

Indeed, input signal probabilities of the majority voter can not be always expected as being “1" or “0". We then consider the general case by using boolean difference which was first used to analyze errors in logic circuits by Frederick et al. [50].

We assume that there is a combinational logic circuit with functionality *f*. It has n inputs $x_1, x_2, \cdots, x_n$ and the output could be expressed as $f(x_1, x_2, \cdots, x_n)$. Boolean difference is defined as

$$\frac{df}{dx_i} = f(x_1, \cdots x_i, \cdots, x_n) \oplus f(x_1, \cdots \bar{x_i}, \cdots, x_n) \tag{5.8}$$

Then, a necessary and sufficient condition for a function f to be independent of input x_i is that $\frac{df}{dx_i} = 0$, i.e. an error in x_i does not cause an error in the output. For the TMR voting system, we have

$$f = x_1x_2 + x_2x_3 + x_1x_3 \tag{5.9}$$

$$\frac{df}{dx_1} = x_2\bar{x_3} + x_3\bar{x_2}. \tag{5.10}$$

and $\frac{df}{dx_1} = 0$ unless $x_2 \neq x_3$, which is impossible if only x_1 is in error. In other words, all single errors could be masked.

As boolean difference is very useful for analyzing the input sensitivity, a tool based on C language is developed to realize this function. Details about this tool will be presented in the section of *Appendix*.

For multiple errors, total boolean difference is defined in [14]. Now we use total boolean difference as well as the faulty gate model shown in Fig. 5.6 to calculate the output error probabilities of the majority voter.

According to (5.3), $\varepsilon_z = \varepsilon_g + (1 - 2\varepsilon_g)\varepsilon_{in}$,

$$\begin{aligned}
\varepsilon_{in} &= \varepsilon_1(1-\varepsilon_2-\varepsilon_3+\varepsilon_2\varepsilon_3)Pr\left\{\frac{df}{dx_1}\right\} \\
&+ \varepsilon_2(1-\varepsilon_1-\varepsilon_3+\varepsilon_1\varepsilon_3)Pr\left\{\frac{df}{dx_2}\right\} \\
&+ \varepsilon_3(1-\varepsilon_1-\varepsilon_2+\varepsilon_1\varepsilon_2)Pr\left\{\frac{df}{dx_3}\right\} \\
&+ \varepsilon_1\varepsilon_2(1-\varepsilon_3)Pr\left\{\frac{df}{d(x_1x_2)}\right\} \\
&+ \varepsilon_2\varepsilon_3(1-\varepsilon_1)Pr\left\{\frac{df}{d(x_2x_3)}\right\} \\
&+ \varepsilon_1\varepsilon_3(1-\varepsilon_2)Pr\left\{\frac{df}{d(x_1x_3)}\right\} \\
&+ \varepsilon_1\varepsilon_2\varepsilon_3Pr\left\{\frac{df}{d(x_1x_2x_3)}\right\} \qquad (5.11)
\end{aligned}$$

where $Pr\{\cdot\}$ stands for the signal probability function and returns the probability of its boolean argument to be 1.

For majority voter, $f = x_1x_2 + x_2x_3 + x_1x_3$,

$$\begin{aligned}
&\frac{df}{dx_i} = x_j\bar{x_k} + x_k\bar{x_j}, i \neq j \neq k \in \{1,2,3\} \\
&\frac{df}{d(x_ix_j)} = x_ix_j + \bar{x_i}\bar{x_j}, i \neq j \in \{1,2,3\} \\
&\frac{df}{d(x_1x_2x_3)} = 1, \qquad (5.12)
\end{aligned}$$

As Han et al. have proved, when the nominal inputs to majority gate are expected to be different, its output is less reliable than its inputs for any $\varepsilon_g \in (0, 0.5)$ [90]. Therefore we consider the case that three inputs have the same $p_i = p$ and $e_i = e$, so yields,

$$\begin{aligned}
\varepsilon_{in} = \varepsilon^3 + 3\varepsilon^2\varepsilon[p^2 + (1-p)^2] + 6p(1-p)\varepsilon(1-\varepsilon)^2 \\
= 2(6p - 6p^2 - 1)\varepsilon^3 + 3(6p^2 - 6p + 1)\varepsilon^2 + (6p - 6p^2)\varepsilon
\end{aligned} \qquad (5.13)$$

Substituting ε_{in} into (5.3), we have

$$\begin{aligned}
\varepsilon_z = \varepsilon_g + (1-2\varepsilon_g)\cdot\varepsilon_{in} = \varepsilon_g + (1-2\varepsilon_g) \\
\cdot[2(6p - 6p^2 - 1)\varepsilon^3 + 3(6p^2 - 6p + 1)\varepsilon^2 + (6p - 6p^2)\varepsilon]
\end{aligned} \qquad (5.14)$$

Notice that (5.14) obtained here is equivalent to (5.7) by von Neumman in [9] under the condition

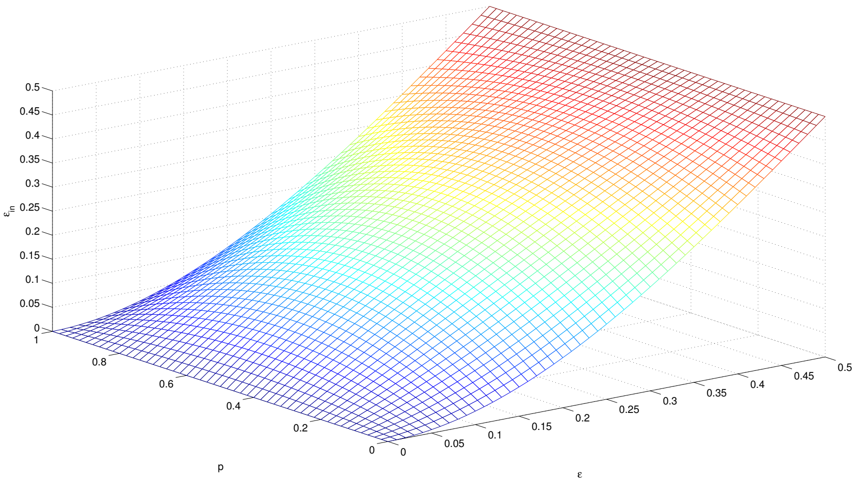


Figure 5.8: ε_{in} varies with ε and p.

$p = 0$ or $p = 1$, which is exactly the assumption that von Neumman stated in [9].

The efficacy of TMR is to decrease the error probability, so output error probability ε_z should satisfy $\varepsilon_z < \varepsilon$. It yields (5.15), which derives (5.16) and then we have (5.17). By solving (5.17) we find the limitation of signal probabilities of majority voter in TMR as in (5.18).

$$\varepsilon_g + (1 - 2\varepsilon_g)\varepsilon_{in} < \varepsilon \tag{5.15}$$

$$(1 - 2\varepsilon_{in})\varepsilon_g < \varepsilon - \varepsilon_{in} \tag{5.16}$$

$$\varepsilon_{in} < \varepsilon < \frac{1}{2} \quad and \quad \varepsilon_g < \frac{\varepsilon - \varepsilon_{in}}{1 - 2\varepsilon_{in}} \tag{5.17}$$

$$1 > p > \frac{3 + \sqrt{3}}{6} \qquad or \qquad 0 < p < \frac{3 - \sqrt{3}}{6} \tag{5.18}$$

Fig. 5.8 shows how ε_{in} varies with parameters ε and p, and Fig. 5.9 describes how ε_{in} varies with ε under some given values of p. We find that $\varepsilon_{in} > \varepsilon$ which makes the TMR invalid if p exceeds the bound in (5.18).

Furthermore, under the condition that p satisfies (5.17), we derive that the maximum value of ε_g that is $\frac{1}{6}$, the same as in [90, 91].

5.3.5 Application

As majority voter is widely used in both the conventional and the emerging nanoelectronic fault-tolerant systems, it is very important to carefully verify the efficacy of TMR for certain applications, since sometimes TMR could bring in worse reliability than a simplex function module could.

The acquisition of output signal probabilities is much more easier compared with the obtention

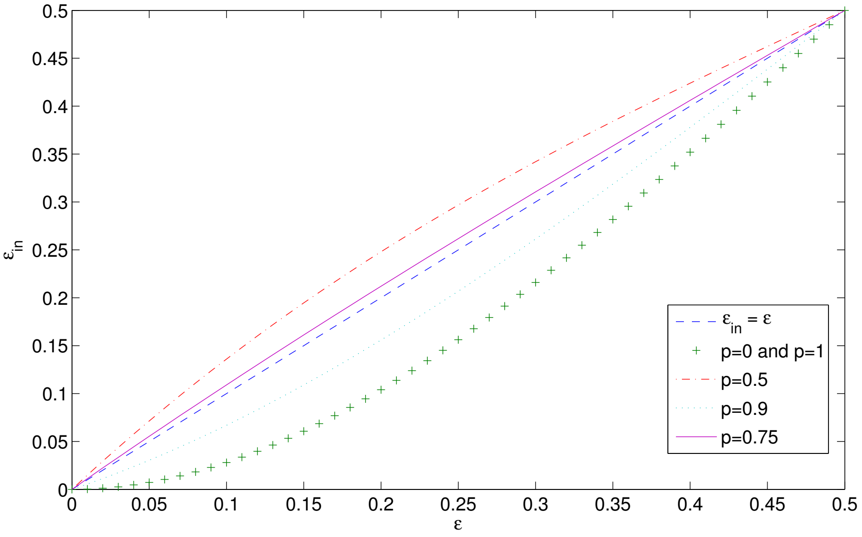


Figure 5.9: ε_{in} varies with ε under given p.

of output reliability. For example, we consider the function module in Fig. 5.10 which is a candidate module for implementing TMR. Let the input test vectors $P(A) = P(B) = \cdots = P(E) = q$, which is the case when tests are implemented from a large set of random numbers. If we consider the three gates are fault-free, the output probability is $z = q^2 + q^3 - q^5$. When $q = 0.5$, $z = 0.34375$. As $\frac{3-\sqrt{3}}{6} < 0.34375 < \frac{3+\sqrt{3}}{6}$, TMR is not suitable for this function module. If the gates are with error probability ε_g which is a more general case, we could use probabilistic gate model (PGM) [49] to obtain the output signal probability easily. When $q = 0.5$, $\varepsilon_g = 0.9$, we have $z = 0.452$.

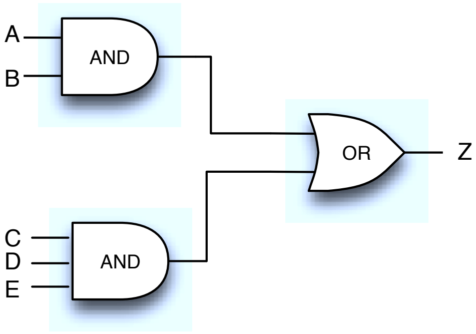


Figure 5.10: An illustrative function module.

5.4 A simple fault-tolerant voter structure in TMR

5.4.1 TMR based on imperfect majority voter

If we consider that voter may fail, which is a more realistic hypothesis, the probability of correctness at the output of the system will decrease according to the reliability $R_{MAJ} = q_{MAJ}$ of the voter, as shown in (5.19).

$$R_{TMR-FV} = R_{MAJ} \times R_{TMR-IV} \tag{5.19}$$

Figure 5.11 shows the impact of the reliability of the voter on the global reliability of the system. It considers $q_{MAJ} = q_M$.

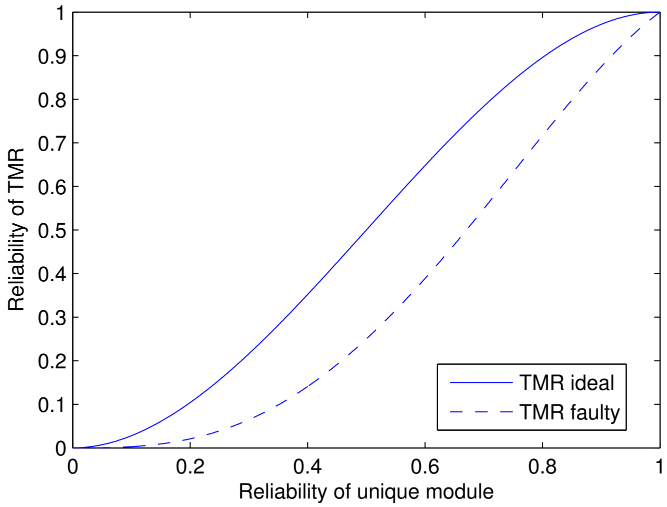


Figure 5.11: Reliability curves for TMR with ideal (continuous line) and faulty (dashed line) voter.

All possible combinations of inputs in a majority voter are shown in Table 5.2. The corresponding canonic boolean expression is given in (5.20). A direct approach of designing a majority voter is to include the terms in equation 5.23. Such a circuit is easy to realize with basic logic gates as shown in Figure 5.12(a), where three AND gates plus one OR gate is used, and in Figure 5.12(b), where four NAND gates are used. The voter in Figure 5.12(a) can be seen as equivalent to that in Figure 5.12(b) if one examines the output and applies DeMorgan's theorem.

$$\begin{aligned} V &= ABC + AB\overline{C} + \overline{A}BC + A\overline{B}C && (5.20) \\ &= AB + BC + AC && (5.21) \end{aligned}$$

$$V = \overline{\overline{AB} \cdot \overline{BC} \cdot \overline{AC}} \quad (5.22)$$

$$= AB + BC + AC \quad (5.23)$$

TMR architectures are designed under the hypothesis of single-fault models. It means that only one fault can occur at a time. This hypothesis is considered in this section. Among the 4 modules (including the voter), only one can be faulty, the others are fault-free.

Recent researches on TMR still utilize this structure in the majority voter, such as [68]. However, this structure masks the single-fault only if it occurs on the module **M**. Take the AND-OR voter structure in Figure 5.12(a) as an example, if **AND** or **OR** gates in the voter fail, the output maybe a faulty value. For example, if $A = B = C = 0$ and there is an unique fault in $S1$, the output will be $V = 1$ which is an incorrect value.

Table 5.2: True table for majority voter.

A	B	C	V
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

5.4.2 Alternative majority schemes

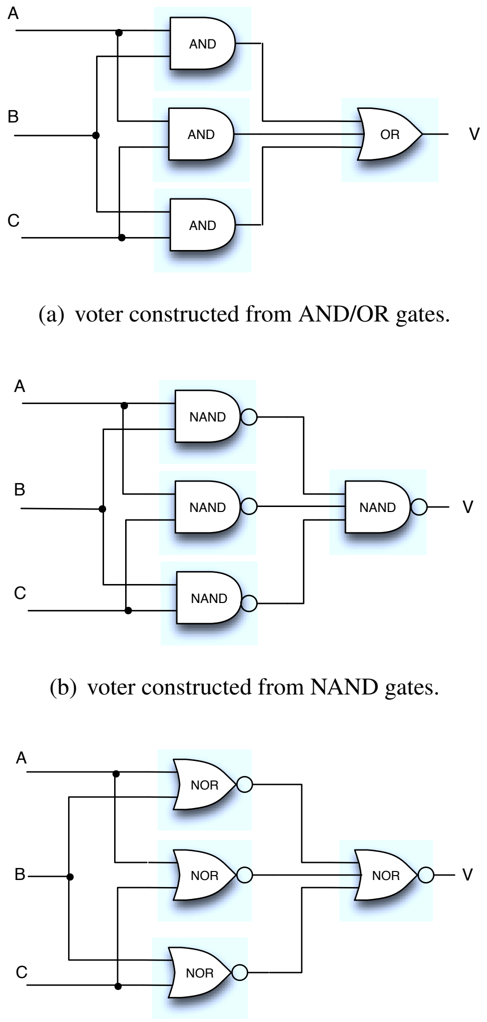
As we have discussed, imperfect voter can influence the circuit reliability a lot and a high reliability of voter should be guaranteed by using reliable components or a fault-tolerant design. In [92], Kshirsagar proposed the fault-tolerant voter (NTFVC) as shown in Figure 5.13. The circuit contains a priority encoder designed to output a selecting signal for the multiplexer.

– if $I1 = 0$, then $sel = 0$ and A (equals to B) is going to be selected as the output;

– if $I2 = 0$, then $sel = 1$ and C (equals to B) is the selected output.

Simulations of a full adder with TMR structure based on this voter proved that this structure brought the robustness to the classical TMR system and has a better performance in the parameters of delay, power, area, etc.

He also illustrated the voter is fault-tolerant. For example, with $A = B = C = 0$, if any of the nodes in the voter, i.e. either $S1$ or $S2$ is stuck to 1, the circuit will still produce a fault-free output as 0. One can show that if the inputs are correct then the outputs will always be correct, irrespective of



(a) voter constructed from AND/OR gates.

(b) voter constructed from NAND gates.

(c) voter constructed from NOR gates.

Figure 5.12: Conventional schemes for majority voter in TMR.

faults in the voter circuit.

5.4.3 A novel simple majority voter

In this work we propose the new scheme for majority voting presented in Figure 5.14. This circuit is based on few logic blocks (just an exclusive-or gate and a multiplexer). We can see that this structure is fault-tolerant following the analysis below:

A fault occurs on the voter It means that the unique internal node S is stuck to a fault. Due to the single-fault model, the output is correct, independent of the value S $(A = B = C)$

A fault occurs on one of the modules M

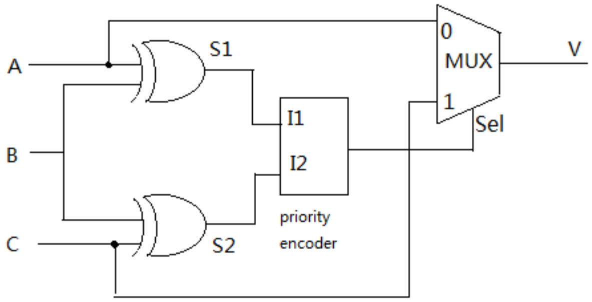


Figure 5.13: Kshirasgar's scheme for the majority voter (NFTVC).

- Case 1 (C is faulty): the logic signals A and B are the same and $S = A \oplus B = 0$. So, the multiplexer's output will be signal B which represents the majority value.
- Case 2 (A or B is faulty): the logic signals A and B are different and the majority must be the logic value given by $AC + BC$, which corresponds to logic value C. According to the scheme, this relation is respected because $S = A \oplus B = 1$.

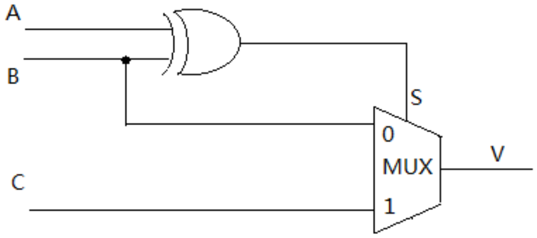


Figure 5.14: Proposed scheme for the majority voter.

5.4.4 Analysis and Results

The proposed voter structure shows a better performance than the previous schemes. Compared to the classical or the Kshirsagar's voter (NFTVC), it saves area together with power dissipation and also there is less delay. These features and improvements come directly from the simplicity of the architecture. Concerning single fault events, we have also proved in Section 5.4.3 that our scheme is as robust as that of Kshirsagar's NFTVC (and therefore better than the conventional solution).

The results obtained with SPR algorithm [12, 46] are shown in Figure 5.16. We consider that probabilities of error for components in the voter are independent and they vary from 0 to 0.5 (so the reliabilities of the nodes in the voter vary from 0.5 to 1). We notice that the proposed voter produces the best reliability among the three solutions.

The proposed voter is then put into a real context such as a half adder in Figure 5.15. Comparison results with respect to different parameters based on different voters in TMR technique are shown in Table 5.3.

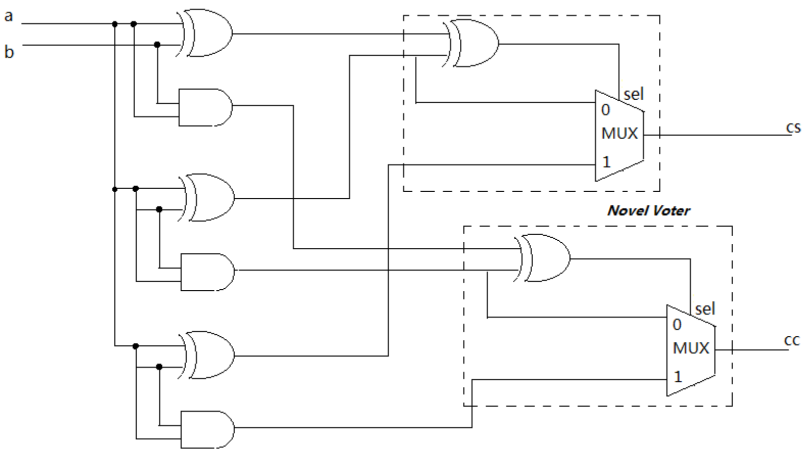


Figure 5.15: Half adder in TMR using proposed voter.

Table 5.3: Results synthesized in ASIC (RTL Compiler).

Comparison	Classic	NFTVC	Proposed
Instances	25	31	25
Area	31	40	32
Power μw	2.35	3.75	2.85

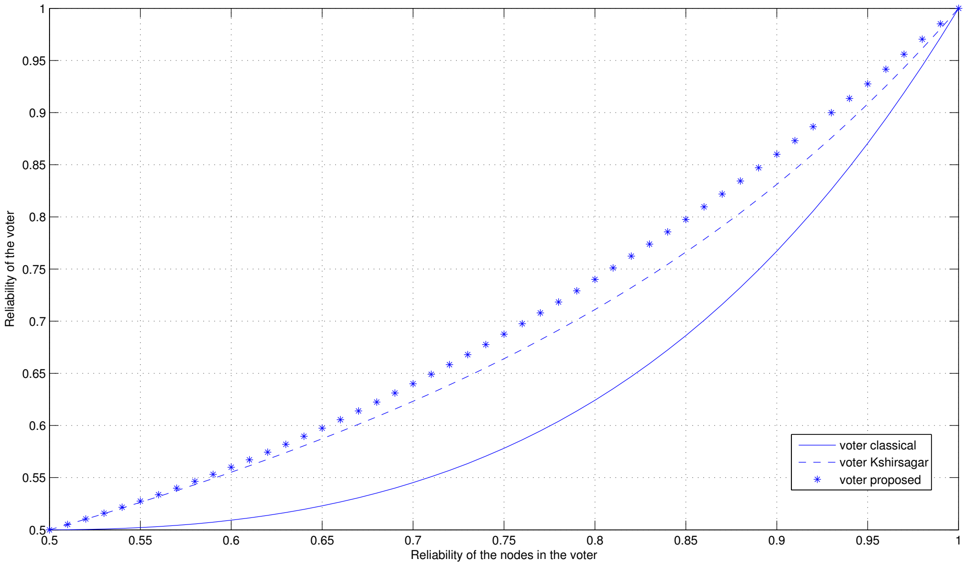


Figure 5.16: Reliability curves for conventional (continuous line), NFTVC (dashed line) and the proposed voter (dotted line).

Conclusion

Concluding remarks

Digital IPs in nanometric technologies are increasingly sensitive to various kinds of perturbations, leading to paradigm shift toward design-for-reliability. Facing the problems within, the present dissertation specified solutions about trade-offs between the new design criterion reliability and hardware redundancy.

As reliability improvement is generally achieved by adding redundancy, identify and classify critical blocks of a circuit is a major concern. Motivated by the need of economical designs, we presented two new classification methods regarding the significance of a block with respect to the reliability of a circuit. One gives the criticality of each block for the circuit reliability and the other indicates which priority should be given to each block. Thus, the proposed concepts provide key information for the designer who is looking for efficient solutions of reliability monitoring or reliability improvement.

Based on these concepts, we then presented an efficient method to select the best subset among possible modular redundant architectures. It builds upon the progressive module redundancy (PMR) technique we proposed. Efficiency is achieved by taking into account the grades of the blocks with respect to reliability, by adding redundancy progressively and by considering mixed modular redundancy. The proposed method presents a shortcut by avoiding analyses of all the possible redundant architectures exhaustively. The PMR method points out a new direction of economical redundant fault-tolerant designs for nanoelectronics.

Experimental results and comparison with the state-of-the-art similar technique have shown the feasibility and efficiency of the proposed method. Furthermore, compared with previous works such as selective module redundant techniques described in [64], the approach described in this thesis does not need to analyze the logic implications of each logic gates, and it could be combined with other soft error mitigation techniques at logic and circuit level.

Voter mechanism is always accompanying with hardware redundancy. Majority voter is very important for its applications in fault-tolerant designs. Individual reliability of a majority voter plays a critical role in the overall reliability of predesigned fault-tolerant systems. Under this background, error propagation characteristics of majority voter is worthy to be discussed. We presented the functional reliability bounds of majority vote, derived the signal probability requirements for its inputs, and also demonstrated its upper error bound. The results obtained are immediately applicable to the practical problems of fault-tolerant designs.

Many researches before were based on the assumption that the voter is perfect while this is not true. We therefore proposed a fault-tolerant and simple majority voter architecture for TMR schemes. The novel voter presents some meaningful features. Given its succinct architecture, it saves area, power dissipation and propagation delays. This solution is robust to single fault and exceeds over those previous ones in terms of reliability.

Scientific products

The studies developed along the dissertations originated several publications, that are cited below:

- "Progressive module redundancy for fault-tolerant designs in nanoelectronics", published at *Microelectronics Reliability - Elsevier*, October 2011, vol. 51, no. 9-11, pp. 1489-1492. [66]
- "A Simple Fault-tolerant Digital Voter Circuit in TMR Nanoarchitectures", presented at *IEEE International New Circuits and Systems Conference (NEWCAS)*, Montreal, Canada, June 2010, pp. 269-272. [83]
- "Optimized Robust Digital Voter in TMR Designs", presented at *Colloque National GdR SoC-SiP*, Lyon, France, June 2011. [93]
- "Reliability analysis based on significance", presented at *IEEE Conference on Micro-nanoelectronics, Technology and Applications (CMTA)*, Buenos Aires, Argentina, August 2011, pp. 1-7. [67]
- "Progressive module redundancy for fault-tolerant designs in nanoelectronics", presented at *European Symposium on Reliability of Electron Devices, Failure Physics and Analysis (ES-REF)*, Bordeaux, France, October 2011. [94]
- "A Progressive Approach for Fault Tolerance Improvement in Digital IPs", presented at *South Symposium on Microelectronics (SIM)*, RS, Brazil, April 2012. [95]
- "Majority Voter: Signal Probability, Reliability and Error Bound Characteristics", presented at *IEEE International Midwest Symposium on Circuits and Systems (MWSCAS)*, Boise, United States, August 2012. [96]
- "Reliability Analysis of a Reed-Solomon Decoder", presented at *IEEE International Midwest Symposium on Circuits and Systems (MWSCAS)*, Boise, United States, August 2012. [97]

And beyond this thesis, the contents and ideas that form these scientific papers have followed up by several new studies on fault tolerant design based on redundancy such as selective hardening by considering trade-offs between power and reliability [98], selective hardening under multiple faults [99] and selective hardening by proposing more practical trade-off parameters [100].

Limitations and future work

The examples used for implementation have some limitations. For the benchmark circuit like C17, it is composed by the same logic gate (NAND). It is reasonable and practical to consider each gate has the same reliability. For the 8-bit ripple carry adder, each block (FA) consists of the same elements, it is therefore also appropriate to consider they are with the same failure rate. However, for other circuits under implementation, we also consider in this way for simplification. It is not desirable and lack of physical basis that we assume a constant reliability of different gate since usually faults and defects attack individual devices such as transistors [101, 102]. Besides individual reliability, each logic gate has different area [57] as they are composed of different numbers of transistors. This point we didn't take into account in the simulation neither.

In this case, a more desirable and accurate metric for evaluating the trade-offs between reliability and area cost should take into account the physical basis as stated before. Here we define C_i (first letter of comprise) as the trade-offs evaluation parameter.

$$C_i = \frac{\Delta R_i}{\Delta A_i} \tag{5.24}$$

where

$$\Delta R_i = \frac{|R_i^* - R|}{R} \tag{5.25}$$

$$\Delta A_i = \frac{|A_i^* - A|}{A} \tag{5.26}$$

they stand for variation percentage of reliability and area cost, respectively. In this way, the units of measure keep the same in both numerator and denominator.

We will have a rank of each constituent block of digital IPs, block grading based on C_i in (5.24) should be more meaningful as it is acquired under physical basis for its applicability. This idea could be realized and we may find better reliability improvements after comparisons with previous ways.

Furthermore, fault injection method presented in this thesis are demonstrated with single error. As the defect density is expected to increase, simulations capable of several failures should be taken into account by using FIFA platform [80]. At the same time, more benchmark circuits should be tested after we conquered the speed of FIFA platform.

Perspectives

The concluding remarks concludes this report, while it could be considered as “early days" for reliability issues of nano electronic systems.

The technology is still scaling down. New circuit structures are emerging, such as typical nanoelectronics devices such as single-electron transistors (SETs), resonant tunneling devices (RTDs), quantum cellular automata (QCA), one-dimensional (1D) devices. CMOS-modecular electronics (CMOL) and other nanoelectronic devices.

Researches on future nanoelectronic systems face challenges, envisioned now as a lack of accurate fault modeling (at different levels) to make more successful reliability evaluation. Reliability under these new emerging technologies should be guaranteed at a relative high level while the redundancy (no matter what kind or a hybrid way) factor should be kept at a very low level.

Under the pressure that failure rates as well as sensitivities to voice and variations of nanodevices are expected higher and higher, reliability will be considered as one of the most threats in the future digital IPs. Fault-tolerant designs should be taken into account at very important phases.

Appendix A

Appendix

A.1 Tools for Inputs Sensitivity Analysis by Boolean Difference

A.1.1 Introduction

In chapter 5, we analyze the boolean difference based on a majority voter. As boolean difference is very useful for analyzing the input sensitivity, it could be very interesting to have a tool for implementing this function. A tool based on C language is developed to realize this function.

For example, we have a circuit as shown in Figure A.1. It has its boolean logic function $y = x_1 \wedge x_3 \vee x_1 x_2 x_3$.

Input sensitivity could be calculated by boolean difference as follows.

$$\frac{dy}{dx_i} = 1 \tag{A.1}$$

$$\frac{df}{dx_1} = x_1 x_3 \tag{A.2}$$

$$\frac{df}{dx_3} = 1 \tag{A.3}$$

It reveals that input x_1 and x_3 are more sensitive since error in these input signals will be definitely propagated to the outputs.

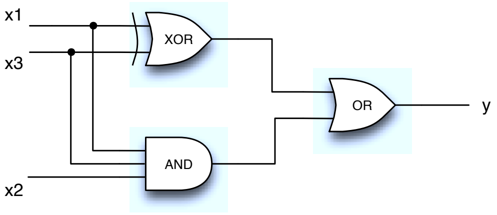


Figure A.1: An illustrative function module.

A.1.2 General usage of the tool

The tool includes help function, help information will be obtained by commands like $help$ or h or ?.

**********************help**********************

The min element number is 2

The max element number is 20

The format of input equation is like this:

Two elements: $y = x0x1 + (x0)x1$

Three elements: $y = x0x1x2 + (x0)(x1)x2 + x0x1(x2)$

Four elements: $y = x0x1x2(x3) + (x0)(x1)x2x3 + x0x1(x2)(x3) + (x0)(x1)(x2)(x3)$

and so on.

****

The tool first verifies input string to exclude the invalid input string. Input should be as follows. $y = x0x1x2(x3) + (x0)(x1)x2x3 + x0x1(x2)(x3) + (x0)(x1)(x2)(x3)$,

where we define:

– element: x0 and x1are elements, an element represents a inpunt.

– nominal: x0x1(x2)(x3), a nominal consists of several elements.

– logic not: parentheses represents logic not. $(x2)$ = $\bar{x}2$.

A.1.3 Variables

Global variables are listed as follows.

– $g\_CurrentElementIndex[MAX_ELEMENT_NUM]$: subscripts of elements.

– $g\_CurrentElementNum : numberofelements.$

– $g\_CurrentItemNum : numberofnominals.$

Local variables are listed as follows:

– element0: Transition array for describing the boolean logic function when $x_i = 0$

– element1: Transition array for describing the boolean logic function when $x_i = 1$

Rows of the transition array represents nominals of boolean logic functions.

A.1.4 Algorithms

For any valid input polynomials, we first derive the transition arrays. Then we compare combinations of all elements to transition arrays exhaustively. The corresponded nominals is abandoned if results derived from the two transition arrays are identical. Otherwise, this nominal is one of the nominals of the output boolean function.

A.1.5 Function descriptions and flow charts

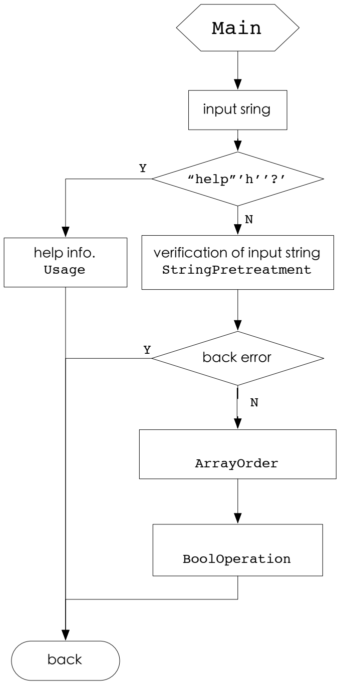


Figure A.2: Flow chat of main function.

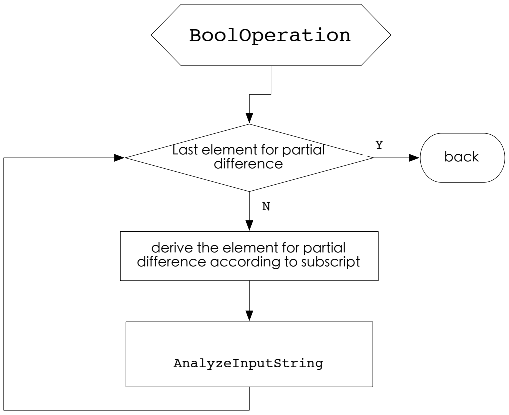

BoolOperation
Last element for partial difference
Y
back
N
derive the element for partial difference according to subscript
AnalyzeInputString

Figure A.3: Flow chat of BoolOperation function.

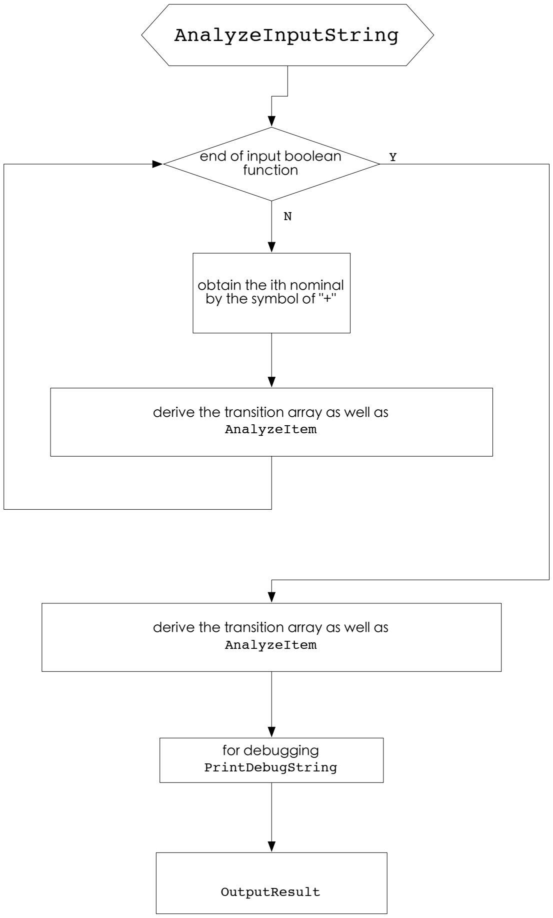

AnalyzeInputString
end of input boolean function
Y
N
obtain the ith nominal by the symbol of "+"
derive the transition array as well as AnalyzeItem
derive the transition array as well as AnalyzeItem
for debugging PrintDebugString
OutputResult

Figure A.4: Flow chat of AnalyzeInputString function.

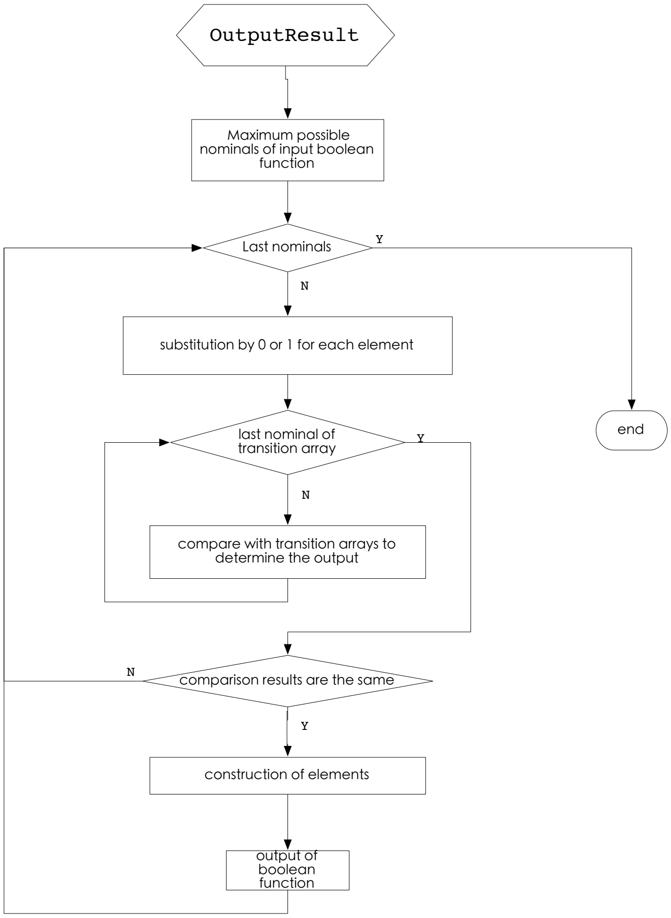

OutputResult
Maximum possible nominals of input boolean function
Last nominals
Y
N
substitution by 0 or 1 for each element
end
last nominal of transition array
Y
N
compare with transition arrays to determine the output
N
comparison results are the same
Y
construction of elements
output of boolean function

Figure A.5: Flow chat of OutputResult function.

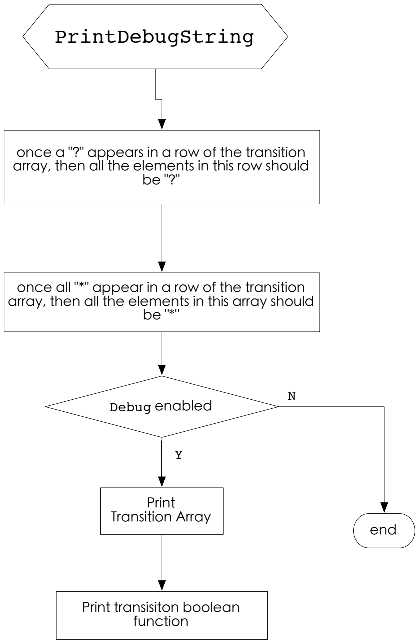

PrintDebugString
once a "?" appears in a row of the transition array, then all the elements in this row should be "?"
once all "*" appear in a row of the transition array, then all the elements in this array should be "*"
Debug enabled
N
Y
Print Transition Array
end
Print transisiton boolean function

Figure A.6: Flow chat of PrintDebugString function.

Bibliography

[1] R. I. Bahar, D. Hammerstrom, J. Harlow, W. H. Joyner Jr., C. Lau, D. Marculescu, A. Orailoglu, and M. Pedram, “Architectures for silicon nanoelectronics and beyond,” *Computer*, vol. 40, pp. 25–33, Jan. 2007.

[2] M. Stanisavljevic, M. Schmid, and Y. Leblebici, *Reliability of Nanoscale Circuits and Systems*. New York, NY, USA: Springer., 2011.

[3] P. Pukite and J. Pukite, *Markov Modeling for Reliability Analysis*. Wiley-IEEE Press, 1st ed., 1998.

[4] S. Almukhaizim and Y. Makris, “Soft error mitigation through selective addition of functionally redundant wires,” *Reliability, IEEE Transactions on*, vol. 57, pp. 23 –31, march 2008.

[5] N. Alves, A. Buben, K. Nepal, J. Dworak, and R. Bahar, “A cost effective approach for online error detection using invariant relationships,” *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 29, pp. 788 –801, may 2010.

[6] E. Mollick, “Establishing moore’s law,” *Annals of the History of Computing, IEEE*, vol. 28, pp. 62 –75, july-sept. 2006.

[7] R. Ogus, “The probability of a correct output from a combinational circuit,” *Computers, IEEE Transactions on*, vol. C-24, pp. 534 – 544, may 1975.

[8] D. P. Siewiorek, “An improved reliability model for NMR,” *Technical report of Stanford University, Computer Systems Laboratory*, vol. 24, dec 1971.

[9] J. von Neumann, “Probabilistic logics and synthesis of reliable organisms from unreliable components,” in *Automata Studies*, pp. 43–98, Princeton University Press, 1956.

[10] H. J. P. Pantel Ketan N, Markov Igor L, “Evaluating circuit reliability under probabilistic gate-level fault models,” in *Proceedings of the twelfth international workshop on logic and synthesis (IWLS 2003)*, pp. 59 – 64, May 2003.

[11] S. Krishnaswamy, G. Viamontes, I. Markov, and J. Hayes, “Accurate reliability evaluation and enhancement via probabilistic transfer matrices,” in *Design, Automation and Test in Europe, 2005. Proceedings*, pp. 282 – 287 Vol. 1, march 2005.

[12] D. Franco, M. Vasconcelos, L. Naviner, and J.-F. Naviner, "Reliability of logic circuits under multiple simultaneous faults," in *Circuits and Systems, 51st Midwest Symposium on*, pp. 265 –268, aug. 2008.

[13] M. de Vasconcelos, D. Franco, L. de B. Naviner, and J.-F. Naviner, "Reliability analysis of combinational circuits based on a probabilistic binomial model," in *Circuits and Systems and TAISA Conference, 2008. NEWCAS-TAISA 2008. 2008 Joint 6th International IEEE Northeast Workshop on*, pp. 310 –313, 2008.

[14] N. Mohyuddin, E. Pakbaznia, and M. Pedram, "Probabilistic error propagation in logic circuits using the boolean difference calculus," in *Computer Design, 2008. ICCD 2008. IEEE International Conference on*, pp. 7 –13, oct. 2008.

[15] J. Han, H. Chen, E. Boykin, and J. Fortes, "Reliability evaluation of logic circuits using probabilistic gate models," *Microelectronics Reliability*, vol. 51, no. 2, pp. 468 – 476, 2011. 2010 Reliability of Compound Semiconductors (ROCS) Workshop; Prognostics and Health Management.

[16] M. Choudhury and K. Mohanram, "Reliability analysis of logic circuits," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 28, no. 3, pp. 392 –405, 2009.

[17] T. Rejimon and S. Bhanja, "Scalable probabilistic computing models using bayesian networks," in *Circuits and Systems, 2005. 48th Midwest Symposium on*, pp. 712 –715 Vol. 1, 2005.

[18] I. Koren and C. M. Krishna, *Fault Tolerant Systems*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2007.

[19] M. L. Shooman, *Reliability of Computer Systems and Networks: Fault Tolerance, Analysis, and Design*. New York, NY, USA: John Wiley & Sons, Inc., 2002.

[20] L. Anghel, D. Alexandrescu, and M. Nicolaidis, "Evaluation of a soft error tolerance technique based on time and/or space redundancy," in *Integrated Circuits and Systems Design, Proceedings. 13th Symposium on*, pp. 237 –242, 2000.

[21] M. Nicolaidis, "Design for soft error mitigation," *Device and Materials Reliability, IEEE Transactions on*, vol. 5, no. 3, pp. 405 – 418, 2005.

[22] "International technology roadmap for semiconductors," 2007.

[23] H. Yamauchi, "A discussion on sram circuit design trend in deeper nanometer-scale technologies," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 18, pp. 763 –774, may 2010.

[24] S. Roy and V. Beiu, "Majority multiplexing-economical redundant fault-tolerant designs for nanoarchitectures," *Nanotechnology, IEEE Transactions on*, vol. 4, pp. 441 – 451, july 2005.

[25] S. S. Mahdavi and K. Mohammadi, "Reliability enhancement of digital combinational circuits based on evolutionary approach," *Microelectronics Reliability*, vol. 50, no. 3, pp. 415 – 423, 2010.

[26] K. Mohanram and N. Touba, "Partial error masking to reduce soft error failure rate in logic circuits," in *Defect and Fault Tolerance in VLSI Systems, 2003. Proceedings. 18th IEEE International Symposium on*, pp. 433 – 440, nov. 2003.

[27] H. Yu, X. Fan, and M. Nicolaidis, "Design trends and challenges of logic soft errors in future nanotechnologies circuits reliability," in *Solid-State and Integrated-Circuit Technology, 2008. ICSICT 2008. 9th International Conference on*, pp. 651 –654, 2008.

[28] M. Baze and S. Buchner, "Attenuation of single event induced pulses in CMOS combinational logic," *Nuclear Science, IEEE Transactions on*, vol. 44, pp. 2217 –2223, dec 1997.

[29] R. Baumann, "Radiation-induced soft errors in advanced semiconductor technologies," *Device and Materials Reliability, IEEE Transactions on*, vol. 5, pp. 305 – 316, sept. 2005.

[30] D. P. Siewiorek and R. S. Swarz, *Reliable computer systems - design and evaluation (2. ed.).* 1992.

[31] K. Nikolic, A. Sadek, and M. Forshaw, "Fault tolerant techniques for nanocomputers," *Nanotechnology*, vol. 13, no. 3, pp. 357–362, 2002.

[32] D. Teixeira Franco, J.-F. Naviner, and L. Naviner, "Yield and reliability issues in nanoelectronic technologies," *Annals of Telecommunications*, vol. 61, pp. 1422–1457, 2006. 10.1007/BF03219903.

[33] K. Woo and M. Guthaus, "Fault-tolerant synthesis using non-uniform redundancy," in *Computer Design, 2009. ICCD 2009. IEEE International Conference on*, pp. 213 –218, oct. 2009.

[34] *The International Technology Roadmap for Semiconductors: 2005 Update*, http://www.itrs.net/Links/2005ITRS/ERD2005.pdf.

[35] B. Parhami, "Defect, fault, error,..., or failure?," *Reliability, IEEE Transactions on*, vol. 46, pp. 450 –451, Dec 1997.

[36] G. C. Messenger, "Collection of charge on junction nodes from ion tracks," *Nuclear Science, IEEE Transactions on*, vol. 29, pp. 2024 –2031, dec. 1982.

[37] R. Leveugle and A. Ammari, "Early SEU fault injection in digital, analog and mixed signal circuits: a global flow," in *Design, Automation and Test in Europe Conference and Exhibition, 2004. Proceedings*, vol. 1, pp. 590 – 595 Vol.1, feb. 2004.

[38] D. Kammler, J. Guan, G. Ascheid, R. Leupers, and H. Meyr, "A fast and flexible platform for fault injection and evaluation in verilog-based simulations," in *Secure Software Integration and Reliability Improvement, 2009. SSIRI 2009. Third IEEE International Conference on*, pp. 309 –314, july 2009.

[39] J.-C. Baraza, J. Gracia, S. Blanc, D. Gil, and P.-J. Gil, “Enhancement of fault injection techniques based on the modification of VHDL code,” *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 16, pp. 693 –706, june 2008.

[40] R. Leveugle and K. Hadjiat, “Multi-level fault injections in VHDL descriptions: Alternative approaches and experiments,” *Journal of Electronic Testing*, vol. 19, pp. 559–575, 2003. 10.1023/A:1025178014797.

[41] D. Lee and J. Na, “A novel simulation fault injection method for dependability analysis,” *Design Test of Computers, IEEE*, vol. 26, pp. 50 –61, nov.-dec. 2009.

[42] A. Namazi, S. Askari, and M. Nourani, “Highly reliable A/D converter using analog voting,” in *Computer Design, 2008. ICCD 2008. IEEE International Conference on*, pp. 334 –339, 2008.

[43] E. Marques, G. Junior, L. Naviner, and J.-F. Naviner, “Effective metrics for reliability analysis,” in *Circuits and Systems (MWSCAS), 53rd IEEE International Midwest Symposium on*, pp. 237 –240, 2010.

[44] G. dos Santos, E. Marques, L. de B. Naviner, and J.-F. Naviner, “Using error tolerance of target application for efficient reliability improvement of digital circuits,” *Microelectronics Reliability*, vol. 50, no. 9-11, pp. 1219 – 1222, 2010.

[45] R. Bahar, E. Frohm, C. Gaona, G. Hachtel, E. Macii, A. Pardo, and F. Somenzi, “Algebraic decision diagrams and their applications,” in *Computer-Aided Design, 1993. ICCAD-93. Digest of Technical Papers., 1993 IEEE/ACM International Conference on*, pp. 188 –191, nov 1993.

[46] D. T. Franco, M. C. Vasconcelos, L. Naviner, and J.-F. Naviner, “Signal probability for reliability evaluation of logic circuits,” *Microelectronics Reliability*, vol. 48, no. 8-9, pp. 1586 – 1591, 2008.

[47] J. T. Flaquer, J.-M. Daveau, L. A. B. Naviner, and P. Roche, “Fast reliability analysis of combinatorial logic circuits using conditional probabilities.,” *Microelectronics Reliability*, vol. 50, no. 9-11, pp. 1215–1218, 2010.

[48] K. Parker and E. McCluskey, “Probabilistic treatment of general combinational networks,” *Computers, IEEE Transactions on*, vol. C-24, pp. 668 – 670, june 1975.

[49] J. Han and P. Jonker, “A system architecture solution for unreliable nanoelectronic devices,” *Nanotechnology, IEEE Transactions on*, vol. 1, pp. 201 – 208, dec 2002.

[50] J. Sellers, F.F., M. Hsiao, and L. Bearnson, “Analyzing errors with the boolean difference,” *Computers, IEEE Transactions on*, vol. C-17, pp. 676 – 683, july 1968.

[51] D. Bhaduri and S. Shukla, “Nanolab-a tool for evaluating reliability of defect-tolerant nanoarchitectures,” *Nanotechnology, IEEE Transactions on*, vol. 4, no. 4, pp. 381 – 394, 2005.

[52] S. Sivaswamy, K. Bazargan, and M. Riedel, “Estimation and optimization of reliability of noisy digital circuits,” in *Quality of Electronic Design, 2009. ISQED 2009. Quality Electronic Design*, pp. 213 –219, march 2009.

[53] P. C. Murley and G. R. Srinivasan, “Soft-error monte carlo modeling program, SEMM,” *IBM Journal of Research and Development*, vol. 40, pp. 109 –118, jan. 1996.

[54] M. Boyd and S. Bavuso, “Simulation modeling for long duration spacecraft control systems,” in *Reliability and Maintainability Symposium, 1993. Proceedings., Annual*, pp. 106 –113, jan 1993.

[55] A. Singhee and R. Rutenbar, “Statistical blockade: A novel method for very fast monte carlo simulation of rare circuit events, and its application,” in *Design, Automation Test in Europe Conference Exhibition, 2007. DATE ’07*, pp. 1 –6, april 2007.

[56] H. Chen and J. Han, “Stochastic computational models for accurate reliability evaluation of logic circuits,” in *Proceedings of the 20th symposium on Great lakes symposium on VLSI*, GLSVLSI ’10, (New York, NY, USA), pp. 61–66, ACM, 2010.

[57] H. Chen, J. Han, and F. Lombardi, “A transistor-level stochastic approach for evaluating the reliability of digital nanometric cmos circuits,” in *Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT), IEEE International Symposium on*, pp. 60 –67, oct. 2011.

[58] E. Taylor, J. Han, and J. Fortes, “Towards accurate and efficient reliability modeling of nanoelectronic circuits,” in *Nanotechnology, 2006. IEEE-NANO 2006. Sixth IEEE Conference on*, vol. 1, pp. 395 – 398, june 2006.

[59] J. Vial, A. Bosio, P. Girard, C. Landrault, S. Pravossoudovitch, and A. Virazel, “Using TMR architectures for yield improvement,” in *Defect and Fault Tolerance of VLSI Systems, 2008. DFTVS ’08. IEEE International Symposium on*, pp. 7 –15, 2008.

[60] B. Bosio, P. Girard, S. Pravossoudovitch, and A. Virazel, “A comprehensive framework for logic diagnosis of arbitrary defects,” *Computers, IEEE Transactions on*, vol. 59, pp. 289 –300, march 2010.

[61] J. Vial, A. Bosio, P. Girard, C. Landrault, S. Pravossoudovitch, and A. Virazel, “Soc yield improvement: Redundant architectures to the rescue?,” in *Test Conference, 2008. ITC 2008. IEEE International*, p. 1, oct. 2008.

[62] L. Anghel and M. Nicolaidis, “Cost reduction and evaluation of a temporary faults detecting technique,” in *Design, Automation and Test in Europe Conference and Exhibition 2000. Proceedings*, pp. 591 –598, 2000.

[63] P. Samudrala, J. Ramos, and S. Katkoori, “Selective triple modular redundancy (STMR) based single-event upset (SEU) tolerant synthesis for FPGAs,” *Nuclear Science, IEEE Transactions on*, vol. 51, no. 5, pp. 2957 – 2969, 2004.

[64] X. She and P. Samudrala, “Selective triple modular redundancy for single event upset (SEU) mitigation,” in *Adaptive Hardware and Systems, 2009. AHS 2009. NASA/ESA Conference on*, pp. 344 –350, 2009.

[65] E. C. Marques, L. A. de Barros Naviner, and J.-F. Naviner, “An efficient tool for reliability improvement based on TMR,” *Microelectronics Reliability*, vol. 50, no. 9-11, pp. 1247 – 1250, 2010.

[66] T. Ban and L. Naviner, “Progressive module redundancy for fault-tolerant designs in nanoelectronics,” *Microelectronics Reliability*, vol. 51, no. 9-11, pp. 1489 – 1492, 2011.

[67] L. de B Naviner, J.-F. Naviner, T. Ban, and G. Gutemberg, “Reliability analysis based on significance,” in *IEEE Conference of Micro-Nanoelectronics Technology and Applications (CMTA),*, pp. 1 –7, aug. 2011.

[68] O. Ruano, J. Maestro, and P. Reviriego, “A methodology for automatic insertion of selective TMR in digital circuits affected by SEUs,” *Nuclear Science, IEEE Transactions on*, vol. 56, no. 4, pp. 2091 –2102, 2009.

[69] M. Augustin, M. Gö andssel, and R. Kraemer, “Reducing the area overhead of TMR-systems by protecting specific signals,” in *On-Line Testing Symposium (IOLTS), 2010 IEEE 16th International*, pp. 268 –273, 2010.

[70] J. Satori, J. Sloan, and R. Kumar, “Fluid NMR - performing power/reliability tradeoffs for applications with error tolerance,” in *Workshop on Power Aware Computing and Systems*, march 2009.

[71] P. Viola and M. Jones, “Robust real-time object detection,” in *International Journal of Computer Vision*, 2001.

[72] X. Wang, K. E. Holbert, and L. T. Clark, “Using TMR to mitigate SEUs for digital instrumentation and control in nuclear power plants,” *Interface*, pp. 925–934, 2010.

[73] T. Oya, T. Asai, T. Fukui, and Y. Amemiya, “A majority-logic device using an irreversible single-electron box,” *Nanotechnology, IEEE Transactions on*, vol. 2, pp. 15 – 22, Mar. 2003.

[74] D. Bhaduri and S. Shukla, “Nanoprism: A tool for evaluating granularity vs. reliability tradeoffs in nano architectures,” in *In Proc. 14th Great Lakes Symposium on VLSI (GLSVLSI’04), ACM.*, apr. 2004.

[75] K. Siozios and D. Soudris, “A methodology for alleviating the performance degradation of TMR solutions,” *Embedded Systems Letters, IEEE*, vol. 2, no. 4, pp. 111 –114, 2010.

[76] J. Han, J. Gao, P. Jonker, Y. Qi, and J. Fortes, “Toward hardware-redundant, fault-tolerant logic for nanoelectronics,” *Design Test of Computers, IEEE*, vol. 22, no. 4, pp. 328 – 339, 2005.

[77] M. Hansen, H. Yalcin, and J. Hayes, “Unveiling the iscas-85 benchmarks: a case study in reverse engineering,” *Design Test of Computers, IEEE*, vol. 16, no. 3, pp. 72 –80, 1999.

[78] M. Singh and I. Koren, “Fault-sensitivity analysis and reliability enhancement of analog-to-digital converters,” *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 11, no. 5, pp. 839 – 852, 2003.

[79] A. Maheshwari, W. Burleson, and R. Tessier, “Trading off transient fault tolerance and power consumption in deep submicron (DSM) VLSI circuits,” *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 12, pp. 299 –311, march 2004.

[80] L. Naviner, J.-F. Naviner, G. dos Santos Jr., E. Marques, and N. P. Jr., “Fifa: A fault-injection fault-analysis-based tool for reliability assessment at RTL level,” *Microelectronics Reliability*, vol. 51, no. 9-11, pp. 1459 – 1463, 2011.

[81] E. Marques, L. Naviner, and J.-F. Naviner, “A method for efficient implementation of reliable processors,” in *Circuits and Systems (MWSCAS), 53rd IEEE International Midwest Symposium on*, pp. 1250 –1253, 2010.

[82] I. Hanninen and J. Takala, “Reliability of n-bit nanotechnology adder,” in *Symposium on VLSI, IEEE Computer Society Annual*, pp. 34 –39, 2008.

[83] T. Ban and L. de Barros Naviner, “A simple fault-tolerant digital voter circuit in TMR nanoarchitectures,” in *NEWCAS Conference (NEWCAS), 8th IEEE International*, pp. 269 –272, june 2010.

[84] J. Han and P. Jonker, “From massively parallel image processors to fault-tolerant nanocomputers,” in *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, vol. 3, pp. 2 – 7 Vol.3, 2004.

[85] D. Blough and G. Sullivan, “A comparison of voting strategies for fault-tolerant distributed systems,” in *Reliable Distributed Systems, 1990. Proceedings., Ninth Symposium on*, pp. 136 –145, Oct. 1990.

[86] B. Parhami, “Voting networks,” *Reliability, IEEE Transactions on*, vol. 40, pp. 380 –394, aug 1991.

[87] B. Parhami, “Voting algorithms,” *Reliability, IEEE Transactions on*, vol. 43, pp. 617 –629, dec 1994.

[88] A. Namazi, M. Nourani, and M. Saquib, “A fault-tolerant interconnect mechanism for nmr nanoarchitectures,” *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 18, pp. 1433 –1446, oct. 2010.

[89] W. Evans and L. Schulman, “Signal propagation and noisy circuits,” *Information Theory, IEEE Transactions on*, vol. 45, pp. 2367 –2373, nov 1999.

[90] J. Han, E. Boykin, H. Chen, J. Liang, and J. Fortes, “On the reliability of computational structures using majority logic,” *Nanotechnology, IEEE Transactions on*, vol. 10, pp. 1099 –1112, sept. 2011.

[91] W. Evans and L. Schulman, “Signal propagation and noisy circuits,” *Information Theory, IEEE Transactions on*, vol. 45, pp. 2367 –2373, nov 1999.

[92] R. Kshirsagar and R. Patrikar, “Design of a novel fault-tolerant voter circuit for TMR implementation to improve reliability in digital circuits,” *Microelectronics Reliability*, vol. 49, no. 12, pp. 1573 – 1577, 2009.

[93] T. Ban and L. Naviner, “Optimized robust digital voter in tmr designs,” in *Colloque National GdR SoC-SiP, 2011*, june 2011.

[94] T. Ban and L. Naviner, “Progressive module redundancy for fault-tolerant designs in nanoelectronics,” in *European Symposium on Reliability of Electron Devices, Failure Physics and Analysis, 2011*, oct. 2011.

[95] T. Ban and L. Naviner, “Fault tolerant architectures in nanoelectronics: The progressive approach,” in *27th Southern Simposium of Microelectronics, 2012*, apr. 2012.

[96] T. Ban and L. Naviner, “Majority voter: Signal probability, reliability and error bound characteristics,” in *Circuits and Systems, 55th IEEE International Midwest Symposium on*, aug. 2012.

[97] K. Liu, T. Ban, L. Naviner, and J.-F. Naviner, “Reliability analysis of a reed-solomon decoder,” in *Circuits and Systems, 55th IEEE International Midwest Symposium on*, aug. 2012.

[98] S. N. Pagliarini, L. A. de Barros Naviner, and J.-F. Naviner, “Selective hardening methodology for combinational logic,” in *Latin-American Test Workshop, 2012. LATW 2012. IEEE*, p. 6, apr. 2012.

[99] S. N. Pagliarini, L. A. de Barros Naviner, and J.-F. Naviner, “Selective hardening methodology concerning multiple faults,” in *Nuclear and Space Radiation Effects Conference, 2012. IEEE*, july 2012.

[100] S. N. Pagliarini, G. G. dos Santos Jr, L. A. de Barros Naviner, and J.-F. Naviner, “Exploring the feasibility of selective hardening for combinational logic,” in *European Symposium on Reliability of Electron Devices, Failure Physics and Analysis*, oct. 2012.

[101] W. Ibrahim and V. Beiu, “Reliability of NAND-2 CMOS gates from threshold voltage variations,” in *Innovations in Information Technology, 2009. IIT ’09. International Conference on*, pp. 135 –139, dec. 2009.

[102] P. Zarkesh-Ha and A. Shahi, “Logic gate failure characterization for nanoelectronic EDA tools,” in *Defect and Fault Tolerance in VLSI Systems (DFT), 2010 IEEE 25th International Symposium on*, pp. 16 –23, oct. 2010.