

**Diagnostic de pannes dans les circuits logiques :  
Développement d'une méthode ciblant un ensemble  
élargi de modèles de fautes**

Alexandre Rousset

► **To cite this version:**

Alexandre Rousset. Diagnostic de pannes dans les circuits logiques : Développement d'une méthode ciblant un ensemble élargi de modèles de fautes. Micro et nanotechnologies/Microélectronique. Université Montpellier II - Sciences et Techniques du Languedoc, 2008. Français. tel-00282204

**HAL Id: tel-00282204**

**<https://tel.archives-ouvertes.fr/tel-00282204>**

Submitted on 26 May 2008

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

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

UNIVERSITE MONTPELLIER II  
SCIENCES ET TECHNIQUES DU LANGUEDOC

THÈSE

Pour obtenir le grade de  
DOCTEUR DE L'UNIVERSITE MONTPELLIER II  
*Discipline : Electronique, Optronique et Systèmes*  
*Ecole Doctorale : Information, Structure et Systèmes*

Présentée et soutenue publiquement

Par

**Alexandre Rousset**

Le 01 Avril 2008

Titre :

---

**Diagnostic de pannes dans les circuits logiques : Développement d'une  
méthode ciblant un ensemble élargi de modèles de fautes**

---

JURY

M. Pravossoudovitch Serge	Président de Jury
M. Girard Patrick	Directeur de thèse
Mme Robach Chantal	Rapporteur
M. Simeu Emmanuel	Rapporteur
M. Bouzaida Laroussi	Examineur
M. Bosio Alberto	Examineur



# Remerciements

Les travaux de recherche de ce mémoire ont été effectués au Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier (LIRMM) dirigé par Monsieur Michel Robert, Professeur à l'Université Montpellier II.

Je souhaite remercier également Monsieur Lionel Torres et Madame Marie-Lise Flottes, chefs du département Microélectronique du LIRMM de m'avoir accueilli au sein du département et pour toute l'aide qu'ils ont pu me fournir tout au long de ma thèse.

Je remercie mon équipe encadrante composée de mon directeur de thèse, Monsieur Patrick Girard (directeur de recherche CNRS), mon co-directeur de thèse, Monsieur Serge Pravossoudovitch (Professeur d'Université Montpellier II), ainsi que Monsieur Christian Landrault (Directeur de recherche CNRS) et Monsieur Arnaud Virazel (Maître de Conférence à l'Université Montpellier II) pour le rôle important d'encadrement qu'ils ont joué tout au long de ma thèse. Je tiens également à remercier Monsieur Alberto Bosio qui est arrivé durant ma deuxième année de thèse et qui a joué un rôle majeur dans l'avancée de celle-ci, autant par son soutien moral que technique.

Je remercie sincèrement Madame Chantal Robach (Professeur d'Université Valence) et Monsieur Emmanuel Simeu (Maître de Conférence Grenoble), qui m'ont fait l'honneur d'accepter d'être rapporteurs et membres du jury de cette thèse.

Tous mes remerciements également à Messieurs Laroussi Bouzaida (St Microelectronics) et Alberto Bosio (Maître de Conférence Montpellier), d'avoir bien voulu participer à mon jury de thèse en tant qu'examineur.

Je remercie enfin Monsieur Serge Pravossoudovitch d'avoir accepté de présider ce jury de thèse.

Une pensée amicale pour tous les thésards croisés durant ces quelques années passées au LIRMM qui ont joué un rôle de soutien, de confident ou technique pour moi, et tout particulièrement Messieurs Julien Dalmasso, Lionel Gouyet, Nicolas Houarche et Laurent Larguier,.

Toutes les personnes m'ayant permis de mener à bien ce travail sont assurées de ma

gratitude.

Mes remerciements vont tout droit à ma famille sans qui je ne serais jamais arrivé là où je suis maintenant, toujours là pour me soutenir, répondant à chaque appel.

Je remercie également affectueusement, Anaé, ma fiancée pour son soutien indéfectible, sa présence attentionnée, sa gentillesse, sa compréhension et pour son sourire angélique qui m'a apporté un bonheur de tous les instants même au moment les plus durs de ma vie.

Enfin, je tiens également à remercier la famille Brunner au grand complet pour tout leur soutien, leur joie de vivre qu'ils savent très bien communiquer et qui fait qu'on a toujours envie de donner le meilleur de soi même.

# *Table des matières*



# Table des matières

<b>INTRODUCTION GENERALE</b>	<b>13</b>
<b>I - Conditions de sensibilisation des pannes</b>	<b>17</b>
Introduction	19
I.1. Modèles de fautes considérés	21
I.1.1. Fautes de collage	22
I.1.2. Fautes de court-circuit	22
I.1.2.a. Strong Driver Wired Or	24
I.1.2.b. Strong Driver Wired And	25
I.1.2.c. Strong Driver Wired	25
I.1.2.d. Wired Or	26
I.1.2.e. Wired And	26
I.1.2.f. Byzantine	27
I.1.3. Court-circuit résistif	28
I.1.4. Fautes de circuit-ouvert	29
I.1.5. Fautes de circuit-ouvert résistif	30
I.1.6. Fautes de délai	30
I.1.7. Fautes de stuck-on/stuck-open	31
I.2. Synthèse	32
Conclusion	35
<b>II - Méthode de diagnostic développée</b>	<b>37</b>
Introduction	39
II.1. Principe de la méthode développée	41
II.2. Phase de localisation des sites potentiels de panne	44
II.2.1. Simulation multivaluée du circuit sain	44
II.2.1.a. Codage des signaux	45
II.2.1.b. Indicateur de valeur pour la prise en compte des courts-circuits	46
II.2.2. Traçage de chemins critiques ou sensibles	49
II.2.3. Processus d'intersection	51
II.2.3.a. Règles d'intersection	51
II.2.3.b. Implantation informatique	53
II.2.4. Prise en compte des sorties saines	55
II.3. Phase d'allocation des modèles de fautes	61
Conclusion	65



<b>III - Optimisations et adaptations de la méthode de diagnostic</b>	<b>67</b>
Introduction	69
III.1. Optimisation de la précision du diagnostic pour les modèles de court-circuit et de stuck-on/open	71
III.1.1. Principe	71
III.1.2. Cas des pannes de court circuit de type « Strong Driver Wired And/Or et Resistive Strong Driver Wired And/Or»	71
III.1.3. Cas des pannes de stuck-on et stuck-open	73
III.1.4. Exemple d'optimisation de la précision du diagnostic pour les modèles de court-circuit et de stuck-on/open	75
III.1.4.a. Amélioration du diagnostic pour les fautes de court-circuit	76
III.1.4.b. Amélioration du diagnostic pour les fautes de stuck-on/open	77
III.2. Adaptation pour prendre en compte des pannes dont l'effet n'affecte qu'une connexion mais pour deux polarités différentes (0/1 et 1/0)	78
III.2.1. Modification du processus d'intersection	79
III.2.1.a. Prise en compte des pannes de délais du type LàM et LàD et des circuits ouverts résistifs.	79
III.2.1.b. Prise en compte des pannes de court-circuit de type « strong driver wired »	81
III.2.2. Modification du tableau d'allocation des modèles de fautes	84
III.3. Adaptation de la méthode de diagnostic pour prendre en compte les pannes de court-circuit pouvant avoir des effets sur les deux connexions	85
III.3.1. Pannes dont l'effet se traduit sur une seule connexion mais qui peut varier en fonction du vecteur de sensibilisation	86
III.3.2. Pannes dont l'effet se traduit sur les deux connexions en court-circuit	88
III.3.2.a. Adaptation du processus de traçage de chemins critiques	88
III.3.2.b. Mise à jour du tableau d'allocation	90
Conclusion	91
<b>IV - Expérimentations</b>	<b>93</b>
Introduction	95
IV.1. Environnement expérimental	97
IV.2. Validation de la méthode de diagnostic développée	98
IV.2.1. Expérimentations réalisées sur le circuit C432 (ISCASS'85)	98
IV.2.2. Expérimentations réalisées sur un jeu de plus gros circuits	102
IV.3. Validation des adaptations de la méthode de diagnostic	102
IV.3.1. Expérimentations permettant d'apprécier l'amélioration de la précision du diagnostic	103
IV.3.2. Expérimentations réalisées avec des pannes induisant des erreurs de polarités multiples ou plusieurs erreurs.	106

Conclusion	111
<b>CONCLUSION GENERALE</b>	<b>115</b>
<b>REFERENCES BIBLIOGRAPHIQUES</b>	<b>121</b>
<b>BIBLIOGRAPHIE PERSONNELLE</b>	<b>127</b>



# *Introduction générale*



# INTRODUCTION GENERALE

L'analyse des défaillances observées en production peut conduire à réagir sur les processus de conception et/ou de fabrication d'un circuit intégré et ainsi permettre d'améliorer les rendements de fabrication. Dans un premier temps, cette opération d'analyse des défaillances s'appuie sur des informations produites par un processus de diagnostic logique dont l'objectif est d'identifier un sous ensemble de sites potentiellement fautifs justifiant les comportements erronés observés sur le Circuit Sous Test (CUT *Circuit Under Test*). Les informations fournies par le processus de diagnostic logique sont alors utilisées pour restreindre l'espace d'investigation et guider l'observation physique du circuit (à l'aide d'un Microscope Electronique à Balayage par exemple) durant l'analyse de défaillance proprement dite.

Les informations généralement disponibles et exploitées par les processus de diagnostic logique sont : (i) les vecteurs de test appliqués au circuit, (ii) les réponses du circuit à ces vecteurs de test (issues du testeur) et (iii) la description structurelle du circuit (généralement au niveau portes). Les algorithmes classiques de diagnostic sont basés soit sur une approche « *Cause à Effet* », soit sur une approche « *Effet à Cause* » [Abra90].

L'approche « *Cause à Effet* » s'appuie sur un processus de simulation de fautes. Il génère un dictionnaire de fautes contenant les réponses du circuit à une séquence de test en présence d'un ensemble donné de fautes [Pom97]. Le diagnostic est alors exécuté en comparant les réponses du circuit sous test avec celles stockées dans le dictionnaire. Bien qu'elle ait un certain nombre d'avantages, notamment le fait de pouvoir traiter des circuits purement séquentiels, cette approche « *Cause à Effet* » présente plusieurs inconvénients. Le premier d'entre eux est la nécessité de disposer de modèles de fautes représentant au mieux les comportements des défaillances pouvant affecter le circuit et de pouvoir faire exploiter ces modèles par un simulateur de fautes. Le second inconvénient est lié à la grande quantité de données qui doivent être générées par le simulateur de fautes, rendant cette approche inadaptée pour des circuits de grande complexité.

La seconde approche, dite « *Effet à Cause* », utilise un processus de recherche du site défaillant à partir des informations produites sur les sorties du circuit durant l'application du

test. Cette approche, basée sur un processus de traçage de chemins critiques (CPT *Critical Path Tracing*) a initialement été proposée dans [Abra90]. Ce processus démarre à partir d'une sortie fautive du circuit et, par une analyse systématique des possibilités de propagation des erreurs sur chacune des portes amont à cette sortie, en déduit les sites de pannes potentiels. Chaque ligne traversée par le processus de traçage est considérée comme une source possible de l'erreur observée en sortie. L'avantage principal de cette approche est la faible quantité d'informations manipulée puisque aucun dictionnaire de faute n'est nécessaire [ven01]. D'autre part, en n'utilisant pas de processus de simulation de fautes, les problèmes de modélisation de fautes sont à priori évités.

D'autres techniques de diagnostic récemment présentées dans la littérature combinent les deux approches « *Effet à Cause* » et « *Cause à Effet* » [Amy06][Liu05] afin de tirer parti des avantages de chacune d'elle. Par ailleurs, certains processus de diagnostic sont également basés sur la génération de nouveaux vecteurs de test appelés *vecteurs de distinction* [Lin07]. Dans ce cas, le résultat fourni par le diagnostic logique est utilisé comme entrée d'un processus de génération de vecteurs de test automatique (ATPG *Automatic Test Pattern Generation*) permettant de générer des vecteurs de test additionnels capable de discriminer les pannes potentielles et ainsi d'améliorer la résolution du diagnostic.

Une analyse détaillée de toutes les méthodes de diagnostic existantes nous a conduit à constater qu'une caractéristique commune à toutes ces méthodes est qu'elles ne manipulent généralement qu'un seul modèle de fautes à la fois ou tout au plus deux modèles de fautes (dans le cas où les effets induits sont identiques) [Eng03]. Malheureusement, avec l'avancée des technologies de fabrication et des stratégies de schéma d'horloge toujours plus agressives, de plus en plus de défaillances conduisent à des effets divers non modélisés par les pannes de collage classiques tels que des déviations temporelles des signaux ou des comportements paramétriques défaillants.

Dans certains cas, le concept de modèles de fautes composites a été adopté afin de résoudre partiellement cet inconvénient. On considère que le comportement de plusieurs modèles de fautes complexes (par exemple le modèle de faute de court-circuit) peut être modélisé par des fautes de collage multiples [Lav98]. Un exemple est l'approche SLAT (*Single Location At a Time*) [Hui04] et l'extension présentée dans [Hol07]. Celle-ci est basée sur un processus d'analyse de la propagation des erreurs effectué par l'intermédiaire d'un simulateur de fautes de collage. L'analyse de la propagation d'erreurs à partir de

chacune des lignes du circuit et la comparaison avec les réponses du circuit obtenues sur le testeur permettent de fournir une liste de lignes potentiellement fautives indépendamment du type de défaillance pouvant affecter cette ligne. Cette approche paraît efficace et très prometteuse pour traiter les divers types de défaillances pouvant affecter les circuits, qu'il s'agisse d'ailleurs de pannes simples ou multiples. Cependant, elle présente encore les inconvénients inhérents aux approches « *Cause à Effet* » (volume de données à manipuler) mais surtout ne permet pas d'identifier le type de défaillance pouvant être à l'origine de l'erreur produite sur les lignes trouvées comme étant potentiellement fautives.

En conséquence, nous avons proposé dans cette thèse une méthode basée principalement sur une approche « *Effet à Cause* » même si, comme nous le verrons dans ce document, une phase très particulière est réalisée par une approche « *Cause à Effet* » pour optimiser le traitement. Cette méthode, qui reprend le principe d'analyse de la propagation des erreurs potentielles évoqué précédemment, s'affranchit d'une manipulation explicite des différents modèles de fautes. Néanmoins elle permet, a posteriori, par une analyse spécifique des signaux et conditions de propagation des erreurs, de proposer un ensemble de modèles de défaillances affectant les sites incriminés.

Ce mémoire présente la méthode de diagnostic développée. La solution proposée consiste en une approche en deux phases : (i) « *la localisation de fautes* » et « *l'allocation de modèles de fautes* ». La phase de localisation de fautes est une phase d'analyse du circuit en fonction des réponses fournies par le testeur. Cette analyse a pour objectif d'obtenir l'ensemble des lignes qui peuvent être la source des erreurs observées (lignes suspectes). Durant cette phase, un ensemble d'informations est également collecté afin de permettre l'identification des modèles de fautes possibles. Cette identification fait l'objet de la deuxième phase du processus de diagnostic développé.

Ce mémoire sera donc organisé de la manière suivante. Dans le chapitre I, nous présenterons les différents modèles de fautes que nous considérerons, ainsi que les conditions de sensibilisation de chacun d'entre eux. Ces conditions sont un des points les plus importants à mettre en place, car toutes les hypothèses de simulation, d'intersection et d'allocation sont basées sur celles-ci. Puis nous détaillerons la méthode de diagnostic développée dans le chapitre II. Nous verrons les différentes étapes du processus de diagnostic allant de la simulation du circuit sain jusqu'à l'obtention d'une liste de lignes suspectes auxquelles sera associé un ensemble précis de modèles de fautes pouvant expliquer le



comportement fautif. Ensuite, nous aborderons le chapitre III, lequel est scindé en deux parties. La première partie montre comment optimiser le résultat de diagnostic obtenu dans le chapitre II. Ces optimisations concernent tout d'abord le modèle de court-circuit, et permettent de déterminer un ensemble d'agresseurs précis, mais également les modèles de stuck-on/open permettant de cibler ou discriminer les transistors liés à une/des porte(s) amont à une ligne suspecte. La seconde partie de ce chapitre III montrera les modifications à appliquer à notre méthode de diagnostic, afin de prendre en compte des modèles de fautes provoquant des erreurs doubles sur une connexion ou bien provoquant une erreur simple sur deux connexions. Enfin, le chapitre IV présentera un ensemble complet d'expérimentations menées afin de prouver l'efficacité et la précision de la méthode de diagnostic proposée. Ce chapitre sera structuré de façon à montrer la fiabilité et l'intérêt de notre méthode à chaque étape de son évolution. Nous terminerons ce mémoire par une conclusion générale qui résumera l'ensemble des différents chapitres et qui proposera des perspectives de travail.

# **I - Conditions de sensibilisation des pannes**



## **Introduction**

Pour qu'une panne produise une erreur observable en sortie, deux conditions doivent être remplies. La panne doit être sensibilisée par le vecteur (ou la séquence de vecteurs) de test et l'erreur ainsi produite doit être propagée jusqu'à une sortie au moins. L'aspect propagation, étant inhérent à la méthode de diagnostic proposée, sera considéré dans les chapitres suivants. Par contre, l'aspect sensibilisation requiert une étude préalable pour définir les paramètres et informations pertinentes qu'il faudra prendre en compte pour élaborer les procédures de diagnostic.

Ainsi, l'objectif de ce chapitre est de faire le point sur les conditions de sensibilisation des divers modèles de fautes considérés.



## I.1. Modèles de fautes considérés

Les défaillances affectant les circuits peuvent être de nature très diverses (défauts localisés ou répartis, variations des paramètres des transistors dues à des variations du processus de fabrication, ...) et entraîner des comportements également divers (erreurs logiques, retards, déviations de paramètres électriques, ...). Afin de pouvoir être traités par les outils de simulation de fautes, de génération de vecteurs de test, ou de diagnostic, les comportements engendrés par ces défauts sont généralement représentés par des modèles. Ces modèles traduisent des déviations du comportement du circuit qui peuvent être logiques (collages, courts-circuits, circuits ouverts), temporelles (pannes de délai de porte/chemin) ou paramétriques (courts-circuits résistifs, circuits ouvert résistifs). D'autre part, certaines défaillances affectent le comportement statique du circuit, d'autres affectent le comportement dynamique. Les pannes affectant le comportement dynamique telles que pannes temporelles ou pannes de type « transistor stuck-open » nécessitent l'application d'une séquence de vecteurs pour être détectées. Ainsi, afin de pouvoir réaliser un diagnostic le plus fiable et précis possible, il est fondamental de pouvoir considérer le plus large éventail possible de défaillances et d'effets induits par ces défaillances, et par conséquent de prendre en compte dans l'outil de diagnostic, un ensemble de modèles de fautes représentant au mieux tous ces comportements. Dans le cadre de cette étude, nous considérons l'ensemble des comportements représentés par les modèles de pannes les plus couramment utilisés [Abra83, Che06, Gho00], à savoir les modèles de :

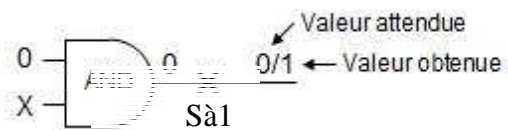
- Collages
- Courts-circuits
- Courts-circuits résistifs
- Circuits ouverts
- Circuits ouverts résistifs
- Délais de porte (de type lent-à-monter et/ou lent-à-descendre) et par extension, de chemins ou de segments de chemins
- Stuck-on et stuck-open (transistors N ou P)

Ces modèles peuvent affecter le circuit de manière plus ou moins différente et sont surtout sensibilisés par des conditions particulières. Dans la suite de ce chapitre, nous étudierons le comportement de ces modèles de fautes, et en particulier les conditions de sensibilisation de chacun d'entre eux.

### I.1.1. Fautes de collage

**Fautes de collage simple :** Une faute de collage simple affecte seulement l'interconnexion entre des portes logiques (Figure 1). Il existe deux types de collage, le collage à 0 et le collage à 1 (généralement écrits respectivement *s-a-0* et *s-a-1*). Ainsi, une ligne L avec un collage à 0 (1) aura toujours l'état logique 0 (1) quelle que soit la valeur logique imposée par la sortie de la porte liée à cette ligne L.

**Condition de sensibilisation :** La condition de sensibilisation d'une telle panne est l'application d'une valeur logique opposée au collage sur le site de la panne.



**Figure 1** : Exemple de collage à 1 d'une connexion

### I.1.2. Fautes de court-circuit

Plusieurs modèles de court-circuit ont été reportés dans la littérature (Strong Driver Wired OR/AND, Strong Driver Wired, Wired AND/OR, Byzantine, courts-circuits résistifs). Chacun de ces modèles représente un comportement particulier pouvant être induit par la mise en court-circuit de deux lignes. Parmi ces modèles nous pouvons différencier :

- Ceux qui provoquent une erreur sur une seule des deux lignes en court circuit (erreur simple) pour une condition de sensibilisation unique (Strong Driver Wired Or, Strong Driver Wired And). Dans ce cas, l'erreur produite sur la ligne est systématiquement soit 0/1, soit 1/0, où la première valeur représente la valeur attendue et la deuxième représente la valeur obtenue.

- Ceux qui provoquent une erreur sur une seule des deux lignes en court circuit (erreur simple) pour deux conditions de sensibilisation différentes (Strong Driver Wired). Dans ce cas l'erreur produite sur la ligne peut être 0/1 et 1/0.

- Ceux qui provoquent une erreur sur l'une ou l'autre des deux lignes en court circuit mais pas sur les deux (erreur simple) pour deux conditions de sensibilisation différentes (Wired Or, Wired And [Lo00]).

- Ceux qui provoquent une erreur sur les deux lignes en court circuit (erreur double) (Byzantine [Han03]).

Le Tableau 1 résume les conditions de sensibilisation ainsi que les effets de chacun des modèles de court-circuit cités précédemment. Ces conditions seront explicitées à la suite de ce tableau.

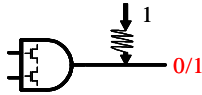
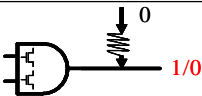
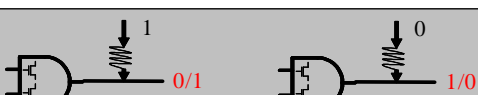
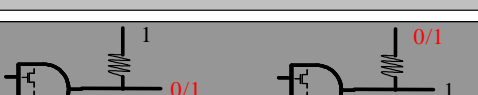
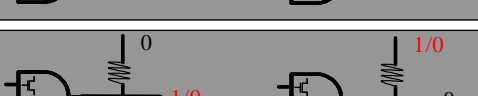
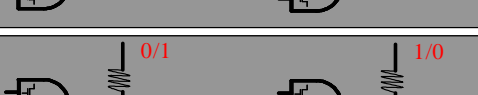
Strong Driver Wired Or		Erreur simple / 1 site 0/1
Strong Driver Wired And		Erreur simple / 1 site 1/0
Strong Driver Wired		Erreur double / 1 site 0/1 et 1/0
Wired Or		Erreur simple / 2 sites 0/1
Wired And		Erreur simple / 2 sites 1/0
Bizantine		Erreur double / 2 sites 0/1 and 1/0

Tableau 1 : Différents modèles de courts-circuits

Un modèle général de court-circuit permettant de préciser les conditions de sensibilisation ainsi que les effets induits (erreurs) par les courts-circuits est proposé sur la Figure 2.  $x$  et  $y$  représentent les valeurs respectives que devraient avoir les deux lignes en court circuit dans le circuit sain. La fonction réalisée par le court-circuit est notée  $Z(x,y)$ . Les



sorties ( $x'$  et  $y'$ ) de cette fonction représentent les valeurs effectives qu'ont les deux lignes en présence du court-circuit considéré.

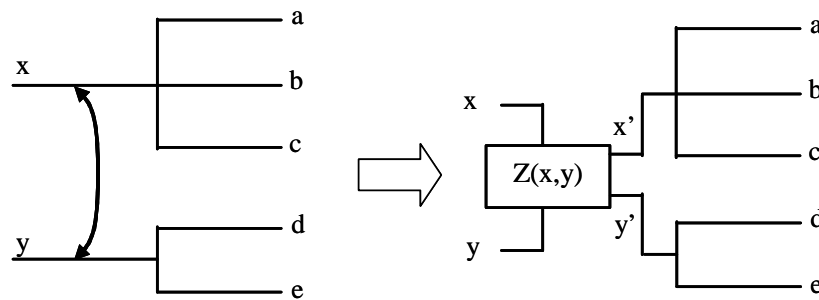


Figure 2 : Modèle général du court-circuit

### I.1.2.a. Strong Driver Wired Or

Un court-circuit de type **Strong Driver Wired Or** correspond à une panne telle qu'une ligne  $x$  positionnée à '1' impose sa valeur à une ligne  $y$  positionnée à '0' (Figure 3). On dit que la ligne  $x$  est l'*agresseur* et que la ligne  $y$  est la *victime*.

**Conditions de sensibilisation :** avoir des valeurs opposées entre l'agresseur et la victime. La ligne victime doit avoir la valeur logique '0' et la ligne agresseur la valeur logique '1'.

Les conditions de sensibilisation et le comportement de ce type de court circuit peuvent être modélisés de la manière suivante : soient  $x$  et  $y$ , deux lignes distinctes du circuit :

Si  $x='1'$  et  $y='0'$ , alors  $x'=y='1'$  sinon  $x'=x$  et  $y'=y$

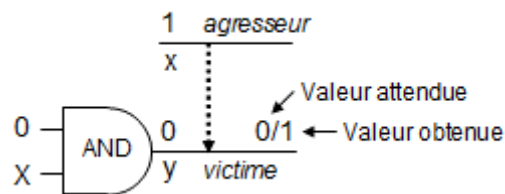


Figure 3 : Court-circuit de type Strong Driver Wired Or

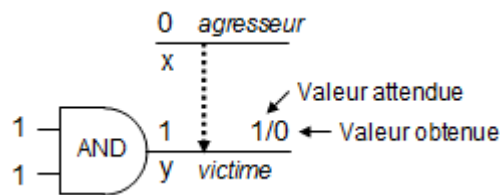
I.1.2.b. Strong Driver Wired And

Un court-circuit de type *Strong Driver Wired And* correspond à une panne telle qu'une ligne x positionnée à '0' impose sa valeur à une ligne y positionnée à '1' (Figure 4). On dit que la ligne x est l'*agresseur* et que la ligne y est la *victime*.

**Conditions de sensibilisation :** avoir des valeurs opposées entre l'agresseur et la victime. La ligne victime doit avoir la valeur logique '1' et la ligne agresseur la valeur logique '0'.

Les conditions de sensibilisation et le comportement de ce type de court circuit peuvent être modélisés de la manière suivante : soient x et y, deux lignes distinctes du circuit :

Si  $x='0'$  et  $y='1'$ , alors  $x'=y='0'$  sinon  $x'=x$  et  $y'=y$



**Figure 4** : Court-circuit de type Strong Driver Wired And

I.1.2.c. Strong Driver Wired

Le court-circuit de type *Strong Driver Wired (SDW)* (Figure 5) produit une erreur simple sur une seule des deux connexions en court-circuit mais l'erreur peut être de « polarité » différente (0/1 ou 1/0) en fonction des valeurs respectives des deux lignes. La valeur de la ligne *agresseur* est toujours imposée à la ligne *victime*.

**Condition de sensibilisation :** avoir des valeurs logiques opposées entre l'agresseur et la victime.

Les conditions de sensibilisation et le comportement de ce type de court circuit peuvent être modélisés de la manière suivante : soient x et y, deux lignes distinctes du circuit :

Si  $x \neq y$ , alors  $x'=y' = x$  sinon  $x'=x$  et  $y'=y$

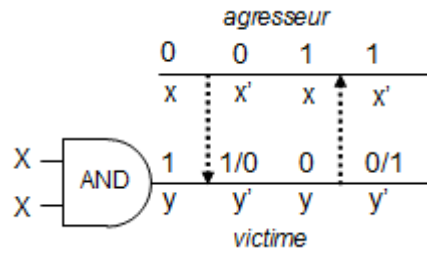


Figure 5 : Court-circuit de type Strong Driver Wired

I.1.2.d. Wired Or

Le court-circuit de type **Wired Or** (Figure 6) produit une erreur simple et sur une seule des deux connexions à la fois. Néanmoins, ce type de court-circuit peut affecter chacune des deux connexions. Dans ce cas là, il n'y a pas de lignes victime et agresseur. Si les deux lignes court-circuitées ont des valeurs logiques opposées, alors la ligne ayant la valeur logique '1' imposera sa valeur à la ligne ayant la valeur logique '0'.

**Condition de sensibilisation :** avoir des valeurs logiques opposées entre l'agresseur et la victime.

Les conditions de sensibilisation et le comportement de ce type de court circuit peuvent être modélisés de la manière suivante : soient x et y, deux lignes distinctes du circuit :

Si  $x=1'$  et  $y=0'$ , alors  $x'=y'=1'$  sinon  $x'=x$  et  $y'=y$

Ou si  $x=0'$  et  $y=1'$ , alors  $x'=y'=1'$  sinon  $x'=x$  et  $y'=y$

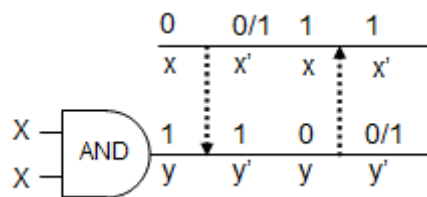


Figure 6 : Court-circuit de type Wired Or

I.1.2.e. Wired And

Le court-circuit de type **Wired And** (Figure 7) produit une erreur simple et sur une

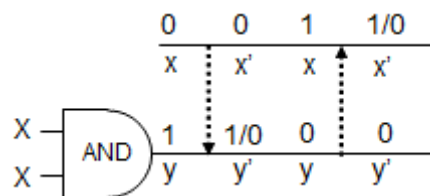
seule des deux connexions à la fois. Néanmoins ce type de court-circuit peut affecter chacune des deux connexions. Dans ce cas là, il n'y a pas de lignes victime et agresseur. Si les deux lignes court-circuitées ont des valeurs logiques opposées, alors la ligne ayant la valeur logique '0' imposera sa valeur à la ligne ayant la valeur logique '1'.

**Condition de sensibilisation :** avoir des valeurs logiques opposées entre l'agresseur et la victime.

Les conditions de sensibilisation et le comportement de ce type de court circuit peuvent être modélisés de la manière suivante : soient x et y, deux lignes distinctes du circuit :

Si  $x='1'$  et  $y='0'$ , alors  $x'=y='0'$  sinon  $x'=x$  et  $y'=y$

Ou si  $x='0'$  et  $y='1'$ , alors  $x'=y='0'$  sinon  $x'=x$  et  $y'=y$



**Figure 7** : Court-circuit de type Wired And

#### I.1.2.f. Byzantine

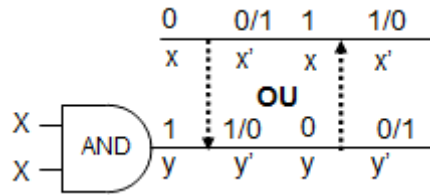
Le court-circuit de type *Byzantine* (Figure 8) produit une erreur double sur les deux lignes court-circuitées. Les valeurs logiques de chaque ligne court-circuitée sont respectivement inversées, c'est-à-dire que pour un même vecteur une des deux lignes prendra la valeur '0' (la ligne a par exemple) alors qu'à l'origine un '1' était attendu, et inversement pour la seconde ligne (la ligne b). Mais une analyse électrique de ce type de panne permet de montrer que cette panne n'est valable que dans un seul sens. Admettons maintenant que la ligne b ait la valeur '1' et la ligne a la valeur '0', aucune erreur n'apparaîtra sur ces lignes. La Figure 8 montre donc les 2 cas possibles.

**Condition de sensibilisation :** avoir des valeurs logiques opposées entre l'agresseur et la victime.

Les conditions de sensibilisation et le comportement de ce type de court circuit

peuvent être modélisés de la manière suivante : soient  $x$  et  $y$ , deux lignes distinctes du circuit :

Si  $x \neq y$ , alors  $x' = y$  et  $y' = x$  sinon  $x' = x$  et  $y' = y$



**Figure 8** : Court-circuit de type Byzantine

### I.1.3. Court-circuit résistif

De la même façon que les courts-circuits présentés précédemment, il existe plusieurs types de courts-circuits résistifs qui sont les suivants :

- Strong Driver Wired Or resistive (Figure 9.a)
- Strong Driver Wired And resistive (Figure 9.b)
- Strong Driver Wired resistive (Figure 9.c)
- Wired Or resistive (Figure 9.d)
- Wired And resistive (Figure 9.e)

Nous ne détaillerons pas l'ensemble de ces courts-circuits car leurs modes de fonctionnement sont identiques à ceux présentés précédemment (Strong Driver Wired Or resistive → Strong Driver Wired Or, Wired Or resistive → Wired Or, ...). La seule différence concerne la sensibilisation de la ligne «victime ». En effet, il doit y avoir une transition sur cette ligne afin de sensibiliser ce modèle de faute (transition descendante dans le cas du Strong Driver Wired Or resistive par exemple). Les différents cas sont représentés sur la Figure 9. Nous pouvons voir que ce sont les mêmes cas que pour le court-circuit simple à la seule différence que la ligne victime est sensibilisée par une transition et que l'observation de la panne se traduira par un retard.

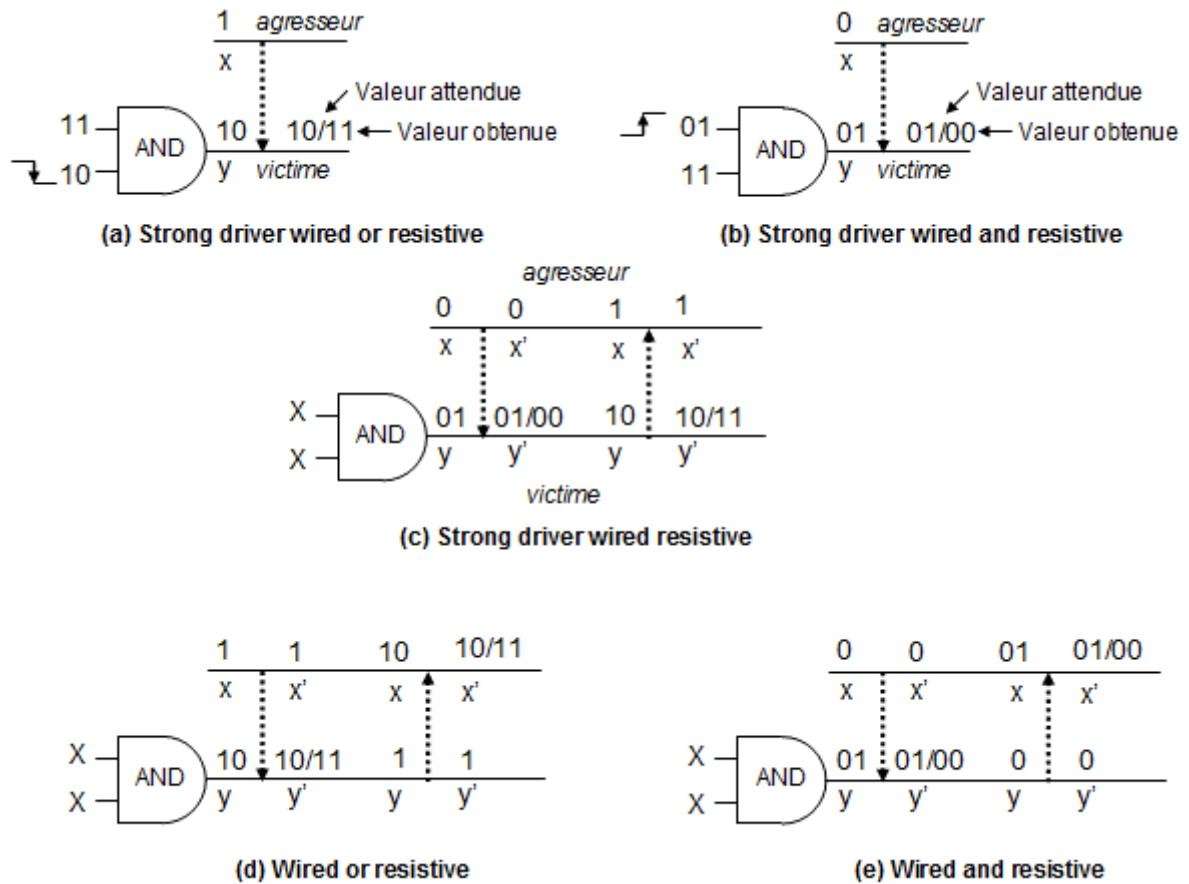


Figure 9 : Cas des courts-circuits résistifs

#### I.1.4. Fautes de circuit-ouvert

Les fautes de circuit ouvert sont assimilables aux fautes de collage en terme de comportement. En effet, une ligne ouverte est ramenée à un potentiel qui peut être la masse, équivalent à un '0' logique permanent, quelle que soit la valeur appliquée sur cette ligne, ou à Vdd, équivalent à un '1' logique permanent (Figure 10).

**Condition de sensibilisation :** application d'une valeur logique opposée à celle engendrée par la panne.

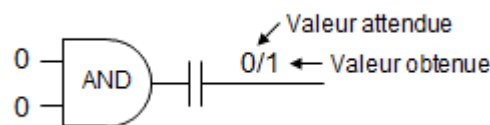
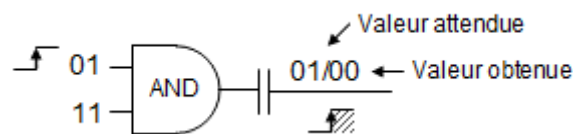


Figure 10 : Exemple d'un circuit ouvert à '1'

### I.1.5. Fautes de circuit-ouvert résistif

Le circuit ouvert résistif fonctionne sur le même principe que les fautes de délai. Une ligne ouverte peut être vue comme une résistance qui ralentit la transition. Ce type de faute peut affecter les transitions montantes (lent-à-monter) (Figure 11), les transitions descendantes (lent-à-descendre) ou les deux à la fois.

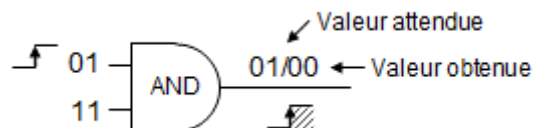
**Condition de sensibilisation :** application d'une paire de vecteurs provoquant une transition (montante pour le lent-à-monter, **ou/et** descendante pour le lent-à-descendre) sur le site de la panne.



**Figure 11** : Exemple de circuit ouvert résistif entraînant une transition lente-à-monter

### I.1.6. Fautes de délai

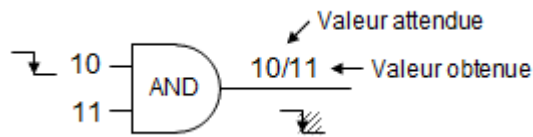
Lorsqu'une panne temporelle est à l'origine du mauvais fonctionnement d'un circuit, cette panne peut soit être issue d'un défaut physique localisé sur un site particulier, soit être la conséquence de plusieurs défauts accumulés le long d'un chemin de propagation. Par conséquent, la modélisation des défauts d'origine temporelle a donné lieu à une classification suivant l'aspect local ou global du défaut. Les deux modèles locaux sont le modèle de *faute de délai de porte* [Bar83, Car87, Hsi77, Kis86, Pra88], et le modèle de *faute de transition* [Lev86, Sch87, Wai86].



**Figure 12** : Exemple d'un lent-à-monter sur une porte ET

Le modèle global est le modèle de *faute de délai de chemin* [Smi85]. Quel que soit le modèle de faute de délai de chemin utilisé, il y a toujours deux fautes de délai associées à un

même site : une faute pour la transition montante (de type « *lent à monter* », Figure 12) et une autre sur la transition descendante (de type « *lent à descendre* », Figure 13).

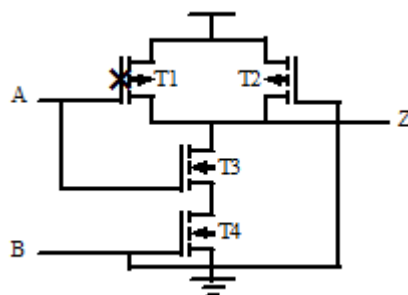


**Figure 13** : Exemple d'un lent-à-descendre sur une porte ET

**Condition de sensibilisation :** application d'une paire de vecteurs provoquant une transition (montante pour le lent-à-monter, ou/et descendante pour le lent-à-descendre) sur le site de la panne.

### I.1.7. Fautes de stuck-on/stuck-open

Le modèle de stuck-on/open suppose que l'un des transistors de la porte logique se trouve dans un état permanent passant (stuck-on) ou ouvert (stuck-open) [Meno93, Wood87]. La Figure 14 montre le cas d'une porte logique NAND dont le transistor T1 est stuck-open. Les fautes de stuck-on/open peuvent engendrer un comportement séquentiel et requièrent donc une séquence de vecteurs pour être détectées. Considérant le cas de la porte NAND dont le transistor T1 est stuck-open (Figure 14), une séquence de deux vecteurs doit être appliquée. En effet, considérons tous les vecteurs possibles qui peuvent être appliqués et qui sont donnés dans le Tableau 2.



**Figure 14** : Transistor T1 stuck-open



Vecteur	PréZ	A	B	ExpZ	ObsZ
1	0	0	0	1	1
2	0	0	1	1	0
3	0	1	0	1	1
4	0	1	1	0	0
5	1	0	0	1	1
6	1	0	1	1	1
7	1	1	0	1	1
8	1	1	1	0	0

**Tableau 2** : Comportement logique en présence d'un stuck-open

La colonne PréZ représente la valeur logique précédente de la sortie Z. La colonne ExpZ donne les valeurs logiques attendues à la sortie de la porte NAND, tandis que la colonne ObsZ fournit les valeurs logiques obtenues en réalité sur la sortie Z en présence de la faute. Ce tableau montre bien qu'une paire de vecteurs est nécessaire à la sensibilisation et à la propagation d'une faute de stuck-open puisque dans ce cas précis la valeur logique précédente du vecteur 2, PréZ(2), est nécessaire pour mettre en évidence le stuck-open du transistor T1. Par contre, si PréZ = 1, il n'y a aucun changement observé sur la sortie Z.

**Condition de sensibilisation :** Application d'une paire de vecteurs pour provoquer une transition sur la sortie (effet séquentiel).

## I.2. Synthèse

L'ensemble des modèles de fautes considérés montre bien que chaque modèle est sensibilisé de façon différente, en effet :

- Les modèles de fautes assimilables à des fautes de délai (court-circuit résistif, circuit ouvert résistif, délai) ou ayant un comportement séquentiel (stuck-on/open) nécessitent l'application d'une paire de vecteurs pour sensibiliser la faute.
- Les modèles statiques tels que court-circuit, collage, circuit ouvert, ne nécessitent qu'un seul vecteur pour être sensibilisés, mais parfois d'autres conditions sont nécessaires comme dans le cas des modèles de courts-circuits où l'agresseur doit avoir une certaine valeur logique.
- L'ensemble des modèles de courts-circuits étudié nécessite aussi d'avoir des

valeurs logiques qui soient opposées entre les deux lignes court-circuitées.

Dans l’hypothèse de faute unique que nous utiliserons pour la mise en place de notre approche de diagnostic, nous considèrerons uniquement les modèles de fautes provoquant une erreur sur une seule ligne du circuit. Dans ce cas, le Tableau 3 résume les modèles de fautes que nous allons considérer ainsi que la relation qui peut être faite entre les vecteurs (ligne 1) et les modèles de fautes considérés (ligne 2 à 15). Ainsi, nous pouvons voir qu’en fonction des vecteurs appliqués sur une ligne L, seuls certains modèles de fautes pourront être associés avec ceux-ci. En effet, comme nous avons pu le voir précédemment, les vecteurs appliqués sur le site d’une panne sensibiliseront une faute si et seulement si ils correspondent aux critères de sensibilisation de celle-ci (paire de vecteurs, valeur logique, transition, valeur logique correcte sur la ligne *agresseur* dans le cas des courts-circuits, ....).

modèles \ vecteurs	00	11	10	01	Ligne agresseur à '0'	Ligne agresseur à '1'
Collage à 0		x		x		
Collage à 1	x		x			
Tn Stuck-open	x		x			
Tn Stuck-on		x		x		
Tp Stuck-open		x		x		
Tp Stuck-on	x		x			
Circuit ouvert à 0		x		x		
Circuit ouvert à 1	x		x			
Strong Driver Wired Or	x		x			x
Strong Driver Wired And		x		x	x	
Strong Driver Wired Or resistive			x			x
Strong Driver Wired And resistive				x	x	
Delai (lent à descendre)			x			
Delai (lent à monter)				x		

Tableau 3 : Relation entre vecteurs et modèles de fautes



## Conclusion

Nous avons vu dans ce chapitre que nous considérons un ensemble de modèles de fautes complet (les plus souvent considérés dans la littérature). Ces modèles de fautes représentent des comportements de défaillances variés. La diversité de ces comportements implique un mode de sensibilisation approprié selon le modèle de faute considéré.

Nous avons pu également voir que certains modèles de fautes ne provoquent qu'une seule erreur dans le circuit, et n'affectent qu'une seule ligne (cas d'une panne simple), alors que d'autres peuvent soit provoquer plusieurs erreurs sur une même ligne, soit affecter plusieurs lignes du circuit (cas du multi erreurs).

Nous pouvons en conclure que lorsque une erreur est observée en sortie du circuit, cette erreur peut être assimilable à un certain nombre de modèles de fautes en fonction des conditions de sensibilisation de chaque ligne suspecte (transition, valeur constante), mais aussi des valeurs sur les autres lignes (agresseur pour les courts-circuits). Lors du processus de diagnostic, il sera donc important de tenir compte de toutes ces conditions de sensibilisation.



# **II - Méthode de diagnostic développée**



## Introduction

L'objectif de ce chapitre est de présenter la méthode de diagnostic que nous avons développée. Pour des raisons de clarté, nous ne considérerons dans ce chapitre que des pannes simples produisant des erreurs sur une seule connexion et ayant toujours la même polarité. Les autres cas seront abordés dans les chapitres suivants.

Dans ce chapitre, nous préciserons, dans un premier temps, les grands principes de l'approche de diagnostic adoptée. Nous décrirons par la suite les différentes étapes du processus de diagnostic. Nous verrons que ce processus comprend deux phases principales ; une phase de localisation des sites de pannes potentiels et une phase d'affectation de modèles de pannes. La phase de localisation des sites de pannes potentiels est conçue autour d'une approche de type « effet à cause » qui est complétée, dans une phase d'optimisation, par une procédure de type « cause à effet ». La phase d'affectation des modèles de pannes fait quant à elle appel à l'étude présentée dans le premier chapitre et notamment aux conditions de sensibilisation des pannes.



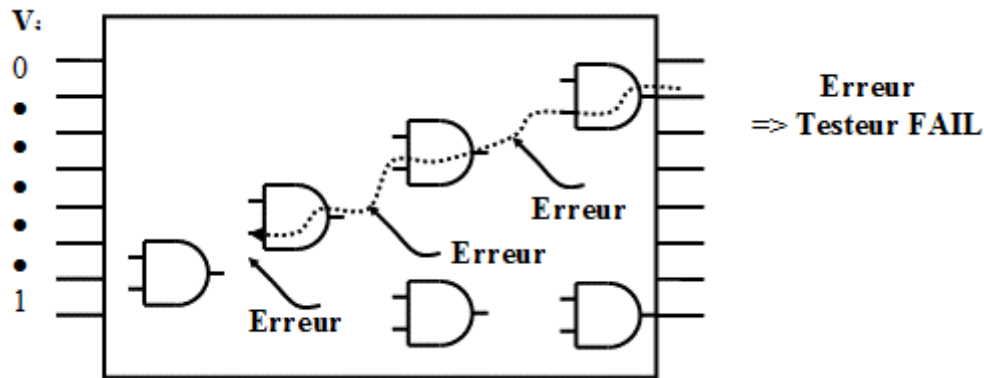


## II.1. Principe de la méthode développée

Les erreurs observées lors du test d'un circuit sont le résultat de défaillances physiques affectant certains composants du circuit. Comme indiqué dans le chapitre précédent, ces défaillances sont généralement représentées par divers modèles de fautes qui peuvent modéliser soit une déviation logique (collage, courts-circuits, circuits ouverts, transistors stuck-open), soit une déviation temporelle ou paramétrique (pannes de délais, courts-circuits résistifs, circuits ouverts résistifs, transistors stuck-on). Certaines fautes (collage, courts-circuits, circuits ouverts, transistors stuck-on) affectent le comportement statique du circuit, d'autres (transistors stuck-open, pannes de délais, courts circuits résistifs, circuits ouverts résistifs) modifient le comportement dynamique et par conséquent nécessitent une transition pour être sensibilisées et ainsi produire une erreur.

Quoi qu'il en soit, l'effet de ces défaillances se traduit, lors du test, par l'observation de valeurs logiques erronées sur les sorties du circuit. A partir de l'observation de ces valeurs erronées sur les sorties, le but est de remonter au site défaillant et éventuellement au modèle de défaillance. Lorsqu'une erreur est observée sur une sortie  $S_i$  du circuit, cette erreur provient soit d'une défaillance sur cette sortie, soit de la propagation d'une erreur provoquée par une défaillance sur une connexion située en amont. A partir de cette constatation, nous avons établi un processus de diagnostic scindé en deux phases.

La première appelée « phase de localisation » est une phase de recherche des sites potentiels d'erreurs ayant pu produire les erreurs observées en sortie. (Figure 15). Cette phase requiert la connaissance (i) de la structure du circuit exprimée au niveau porte, (ii) des vecteurs de test appliqués au circuit, (iii) des réponses du circuit à ces vecteurs de test (réponses fournies par le testeur). Elle fournit une liste de sites potentiels d'erreurs pouvant expliquer les erreurs observées en sorties ainsi que des informations concernant les valeurs logiques propagées dans le circuit (Fault evidence).



**Figure 15** : Sites potentiels d'erreurs ayant pu produire l'erreur observée en sortie

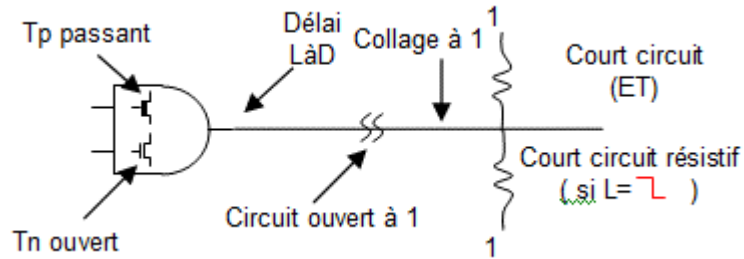
La seconde phase du processus de diagnostic développé appelée « phase d'affectation des modèles » est une phase de définition des modèles de pannes potentiels associés à ces sites. Lorsqu'une erreur potentielle est localisée sur une ligne L, cette erreur peut être engendrée par divers défauts qui peuvent être identifiés grâce à des informations telles que : (i) la valeur logique attendue sur la ligne L, (ii) les valeurs logiques attendues sur les autres lignes du circuit (pannes de court-circuit), (iii) les transitions produites sur la ligne L (pannes dynamiques telles que pannes de délai et stuck-open). Ainsi, il apparaît que ces modèles de fautes peuvent être déterminés à partir de la connaissance du fonctionnement du circuit sain.

A titre d'exemple, si un '1' est observé sur une ligne L alors qu'un '0' est attendu (Figure 16), cette erreur peut provenir :

- D'un collage à 1 de L,
- D'une coupure de type « Open 1 » de la ligne L
- D'un court-circuit (de type OR) entre L et toute ligne à 1
- D'un stuck-on sur un transistor p de la porte amont
- D'une panne de délai sur L (uniquement si une commutation se produit sur L)
- D'un court-circuit résistif entre L et toute ligne à 1 (si une commutation se produit sur L)
- D'un circuit ouvert résistif (Resistive Open) sur L (uniquement si une

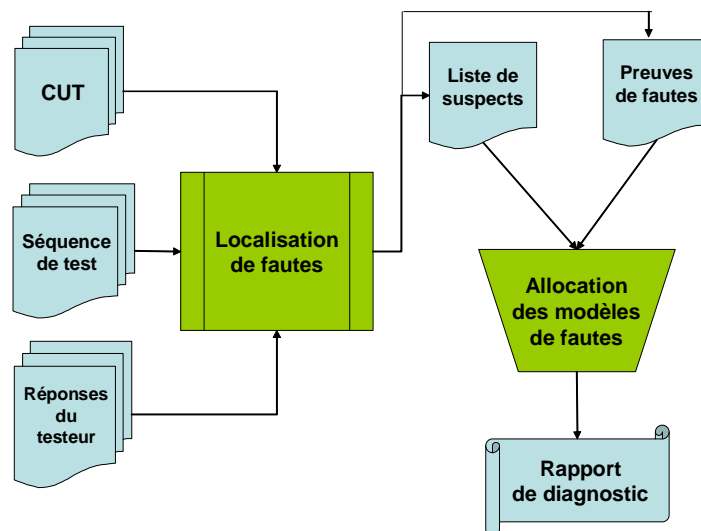
commutation se produit sur L)

- D'un stuck-open sur un transistor p de la porte amont



**Figure 16** : Défautes pouvant être à l'origine d'une erreur

Le synoptique du processus de diagnostic ainsi développé est présenté sur la Figure 17. La phase de « localisation » a pour entrée la description structurelle du circuit (CUT), la séquence de vecteurs de test et les réponses du circuit à ces vecteurs. Elle fournit une liste de suspects (lignes suspectées d'être à l'origine de l'erreur observée) ainsi que des informations spécifiques concernant les valeurs propagées dans le circuit. La liste de suspects et ces informations sont exploitées lors de la phase « d'affectation des modèles » qui produit le rapport de diagnostic final.



**Figure 17** : Synoptique du processus de diagnostic développé

## II.2. Phase de localisation des sites potentiels de panne

L'objectif de cette première phase du processus de diagnostic est de déterminer l'ensemble des sites de pannes potentiels c'est à dire l'ensemble des lignes potentiellement suspectes. Cette phase de localisation conjugue une approche « *Effet à Cause* » avec une approche « *Cause à Effet* ».

Une analyse « effet à cause » basée sur le principe de traçage de chemins critiques est réalisée dans un premier temps pour déterminer les sites de pannes potentiels. Cette analyse comporte les trois étapes suivantes :

- Simulation multivaluée du circuit sain,
- Traçage de chemins critiques (CPT) à partir des sorties déclarées fautives (fail) lors du test,
- Intersection des listes de suspects obtenues par le processus de traçage de chemins.

Une analyse « cause à effet » est alors réalisée pour optimiser la liste de sites potentiels de pannes obtenue précédemment. Cette analyse permet de prendre en compte certaines informations issues des sorties déclarées saines (pass) lors du test. Comme nous le verrons par la suite, il s'avère qu'une analyse de type « *Cause à Effet* » s'avère plus efficace dans ce cas qu'une analyse de type « effet à cause ».

### II.2.1. Simulation multivaluée du circuit sain

Cette phase de simulation permet de déterminer, pour chacun des vecteurs de test conduisant à l'observation d'une erreur en sortie, les informations logiques utiles au processus de traçage de chemins critiques. Afin de faire apparaître, outre les valeurs logiques, les commutations ou possibilités de commutations sur les diverses lignes du circuit, cette simulation met en œuvre une logique multivaluée [Gir92a].

II.2.1.a. Codage des signaux

Nous avons vu dans le chapitre précédent que les conditions de sensibilisation variaient en fonction du modèle de faute incriminé. En effet, pour les fautes statiques telles que le collage, court-circuit, circuit ouvert, un seul vecteur est nécessaire à la sensibilisation de la panne. Par contre, les fautes dynamiques (délai, court-circuit résistif, circuit ouvert résistif, stuck-open) ont besoin d'une paire de vecteurs pour les sensibiliser. L'algèbre à six valeurs (C0, C1, F0, R1, P0, P1) développé dans [Gira92a, Gira92b] permet de satisfaire toutes les exigences de sensibilisation précédemment citées. Dans le cadre de cette étude, nous utiliserons donc cette algèbre à six valeurs :

- C0 (C1) représente un signal stable à 0 (1) quels que soient les délais présents dans le circuit (délais inhérents au circuit ou délais provenant d'une panne).
- R1 (F0) modélise un signal dont la valeur initiale est 0 (1) et la valeur finale est 1 (0). Ce signal peut présenter plusieurs transitions, mais la dernière est nécessairement une transition montante (descendante).
- P0 (P1) représente un signal de valeur initiale et finale à 0 (1) mais présentant une possibilité d'aléas logiques à 1 (0) en fonction des valeurs des paramètres temporels du circuit ou des pannes temporelles affectant ce circuit.

Ce codage permet de représenter les valeurs logiques produites dans le circuit par le vecteur de test ayant produit l'erreur (0, 1), les transitions générées par ce vecteur (R1, F0) ou les possibilités de transitions (P0, P1), sans avoir à réaliser d'analyse temporelle du circuit. On peut remarquer qu'une algèbre à quatre valeurs (C0, C1, F0, R1) aurait pu être suffisante pour représenter les transitions. Néanmoins, disposer des possibilités de transition peut être intéressant notamment pour affiner le traitement des pannes de délai.

Les signaux à appliquer sur les entrées du circuit lors de la simulation sont déterminés à partir du vecteur de test  $V_i$  ayant produit l'erreur et du vecteur  $V_{i-1}$  appliqué sur le circuit immédiatement avant le vecteur  $V_i$ . Ainsi, pour une entrée E donnée, la valeur à appliquer est déterminée de la manière suivante :

$$V_{i-1}(E) = 0, V_i(E) = 0 \Rightarrow E = C0$$

$$V_{i-1}(E) = 1, V_i(E) = 1 \Rightarrow E = C1$$

$$V_{i-1}(E) = 0, V_i(E) = 1 \Rightarrow E = R1$$

$$V_{i-1}(E) = 1, V_i(E) = 0 \Rightarrow E = F0$$

Une fois les entrées déterminées, le processus de simulation revient à propager ces signaux dans le circuit sain en utilisant les tables de transfert associées aux portes logiques. Les tables de transfert des portes ET, OU, NOT sont données sur le Tableau 4. Celles associées aux autres types de portes se déduisent aisément de celles-ci.

<b>AND</b>	<b>C0</b>	<b>C1</b>	<b>F0</b>	<b>R1</b>	<b>P0</b>	<b>P1</b>
<b>C0</b>	C0	C0	C0	C0	C0	C0
<b>C1</b>	C0	C1	F0	R1	P0	P1
<b>F0</b>	C0	F0	F0	P0	P0	F0
<b>R1</b>	C0	R1	P0	R1	P0	R1
<b>P0</b>	C0	P0	P0	P0	P0	P0
<b>P1</b>	C0	P1	F0	R1	P0	P1

<b>OR</b>	<b>C0</b>	<b>C1</b>	<b>F0</b>	<b>R1</b>	<b>P0</b>	<b>P1</b>
<b>C0</b>	C0	C1	F0	R1	P0	P1
<b>C1</b>	C1	C1	C1	C1	C1	C1
<b>F0</b>	F0	C1	F0	P1	F0	P1
<b>R1</b>	R1	C1	P1	R1	R1	P1
<b>P0</b>	P0	C1	F0	R1	P0	P1
<b>P1</b>	P1	C1	P1	P1	P1	P1

<b>IN</b>	<b>OUT</b>
<b>C0</b>	C1
<b>C1</b>	C0
<b>F0</b>	R1
<b>R1</b>	F0
<b>P0</b>	P1
<b>P1</b>	P0

**Tableau 4** : Tables de transfert des portes élémentaires

### II.2.1.b. Indicateur de valeur pour la prise en compte des courts-circuits

Nous avons vu dans le chapitre précédent qu'une des conditions de sensibilisation des pannes de court-circuit était d'avoir des valeurs logiques opposées sur les deux lignes court-circuitées.

Outre la ligne portant l'erreur potentielle, il peut donc être intéressant de disposer

d'informations complémentaires sur les lignes potentiellement « agresseur ».

Nous savons par exemple que dans le cas du Strong Driver Wired Or, la ligne qui agresse la ligne victime doit porter la valeur logique '1' pour tous les vecteurs fautifs (pour que ce modèle puisse être incriminé). Dans ce cas, nous devons faire en sorte de déterminer quelles lignes conservent continuellement la valeur logique '1' pour l'ensemble des vecteurs fautifs de la séquence de test.

Pour disposer de ce type d'information, nous associerons une variable (flag) à chaque ligne du circuit qui sera mise à jour lors de la simulation du circuit sain. Cette variable permet de conserver l'historique des valeurs qui se sont succédées sur chaque ligne du circuit pour les vecteurs fautifs. Elle indique en fait si les valeurs logiques positionnées sur les lignes du circuit ont changé ou sont restées constantes pour chaque vecteur fautif. Cette variable codée sur deux bits a la signification suivante :

- '00': Invalide, utilisé seulement lors de l'initialisation de chaque ligne du circuit.
- '01': assigné si le symbole de la ligne associée appartient à l'ensemble  $=\{C0,F0,P0\}$
- '10': assigné si le symbole de la ligne associée appartient à l'ensemble  $=\{C1,R1,P1\}$
- '11': assigné si le symbole de la ligne associée appartient à l'ensemble  $=\{C0,F0,P0\} \cup \{C1,R1,P1\}$

Durant la simulation du circuit sain, à l'application de chaque nouveau vecteur fautif, la variable (*flag*) est mise à jour en fonction de la valeur courante du symbole associé à la ligne. Les règles suivantes sont utilisées pour la mise à jour :

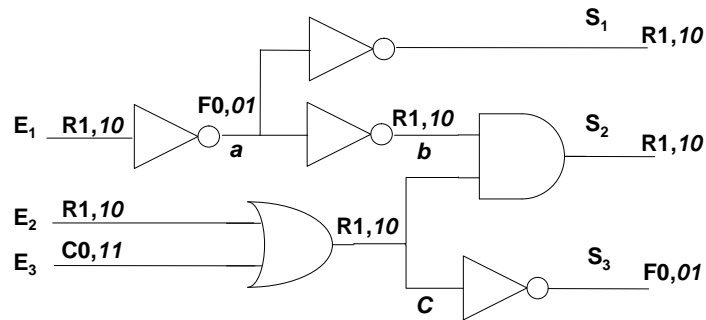
- $Flag(i) = Flag(i) + '01'$  si le symbole courant appartient à  $\{C0,F0,P0\}$
- $Flag(i) = Flag(i) + '10'$  si le symbole courant appartient à  $\{C1,F1,P1\}$

Où  $i$  est la  $i$  ème ligne du circuit,  $Flag(i)$  est le *flag* associé à la  $i$  ème ligne et '+' est l'opérateur logique OU. Ainsi, après application de l'ensemble des vecteurs fautifs,  $Flag(i)$  permet de savoir si les valeurs logiques qui ont été positionnées sur la ligne  $i$  ont toujours été des 0, toujours des 1 ou ont changé.



**Exemple :**

Considérons la Figure 18 qui comporte 3 entrées ( $E_1, E_2, E_3$ ) et trois sorties ( $S_1, S_2, S_3$ ) ainsi que six portes logiques.



**Figure 18** : Simulation du vecteur ( $E_1, E_2, E_3$ ) = (R1,R1,C0)

Une première paire de vecteurs est appliquée sur les entrées du circuit est  $V_1=(0,0,1)$ ,  $V_2=(1,1,1)$ , sachant que lors du test, c'est le vecteur  $V_2$  qui avait propagé des erreurs en sortie(s). D'après ce que nous avons vu précédemment, nous pouvons déduire que les vecteurs d'entrées codés sont  $V_{21}=(R1, R1, C1)$ , et les valeurs seront propagées selon les Tableau 4.

Une seconde paire de vecteurs est appliquée par la suite au circuit. La paire de vecteurs codée est  $V_{87}=(R1, R1, C0)$  (Figure 18). Cela signifie également, dans notre contexte de travail, que le vecteur  $V_8$  a propagé des erreurs sur au moins une sortie du circuit durant la phase de test.

Le couple de l'entrée  $E_3$ , après l'application de la seconde paire de vecteur  $V_{87}$  devient alors (C0, 11), ce qui signifie que le symbole est C0 et le symbole mis à jour est '11'. Précédemment, nous avons vu que pour le vecteur  $V_{21}$ , le flag était devenu '10'. Avec l'application de ce nouveau vecteur, le flag est devenu le résultat de l'opération logique '10' (valeur du flag précédent) + '01' (codage du symbole actuel) = '11'. Le flag rend donc bien compte de l'historique de la ligne d'entrée  $E_3$ , puisque cette ligne a porté une fois un symbole à valeur finale '1' (C1), puis un symbole à valeur finale '0' (C0).

### II.2.2. Traçage de chemins critiques ou sensibles

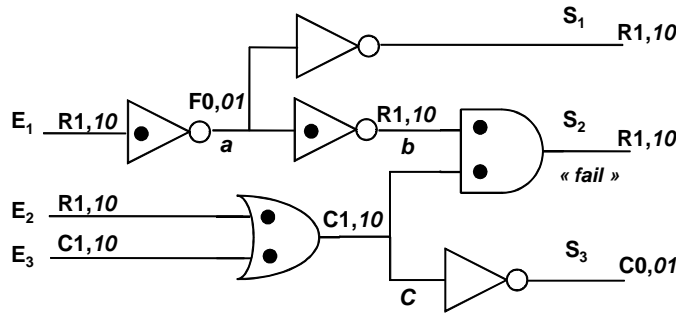
Le processus de traçage de chemins critiques consiste, pour chaque sortie fautive, à remonter dans le circuit au travers des portes logiques afin de déterminer les lignes susceptibles d'être à l'origine de l'erreur observée sur la sortie considérée. Cette remontée s'effectue en suivant les entrées sensibles des portes.

Le processus de traçage de chemins critiques développé s'inspire de ceux utilisés dans [Abra80] et [Gira92a] sans pour autant être parfaitement identique. En effet, dans ces deux cas, un modèle de faute précis était considéré ; le modèle de collage pour [Abra80] et le modèle de délai pour [Gira92a]. Bien que l'algèbre que nous utilisons ici soit celle développée dans [Gira92a], la notion d'entrée sensible qui est utilisée dans [Gira92a] pour assurer la remontée à travers le circuit n'est pas utilisable directement dans notre cas. En effet, dans [Gira92a], les entrées des portes ayant des valeurs stables (C0, C1) ne sont pas considérées comme sensibles puisque ces valeurs stables ne peuvent remplir les conditions de sensibilisation des fautes de délai. Les lignes portant ce type de symbole sont donc ignorées dans le processus de traçage de chemins critiques. Dans notre cas, nous cherchons à remonter une erreur logique, que cette erreur soit induite par des fautes statiques ou dynamiques. Comme nous avons à manipuler des erreurs logiques, nous pouvons par contre, en l'adaptant à la logique multivaluée utilisée, exploiter directement les principes développés dans [Abra80a]. Pour cela, il suffit de considérer la valeur finale de chaque symbole (correspondant au vecteur fautif) pour déterminer les entrées sensibles de chaque porte, et ainsi remonter dans le circuit.

Par ce processus, nous pouvons ainsi déterminer, pour chaque sortie fautive, une liste de lignes suspectées d'être à l'origine de l'erreur observée. D'autre part, à chaque ligne suspecte, nous associons le symbole porté par celle-ci au moment du traçage de chemins. Ainsi, le processus de traçage de chemin fournit des listes de suspects composées du couple (LC,S) où LC est l'identificateur de la ligne critique et S le signal porté par cette ligne pour le vecteur considéré. C'est l'association entre la ligne et le symbole qui nous permettra, lors de la dernière étape de ce processus de diagnostic, d'associer les modèles de fautes correspondant à chaque ligne critique.

**Exemple :**

Considérons à nouveau l'exemple précédent après la simulation de la paire de vecteurs  $V_{21}$ . Supposons maintenant que lors du test, l'application du vecteur  $V_2$  a provoqué une erreur sur la sortie  $S_2$  (Figure 19). Le processus de traçage de chemins critiques débute donc à partir de cette sortie.



**Figure 19** : Simulation du vecteur  $(E1,E2,E3) = (R1,R1,C1)$

La connexion de sortie  $S_2$ , pouvant être à l'origine de l'erreur observée, est directement ajoutée à la liste de suspects. La remontée à travers le circuit s'effectue ensuite à travers la porte logique ET. Les deux entrées de cette porte sont dites sensibles, car ces deux entrées ont pour valeur finale la valeur non prioritaire (1) de la porte ET. Autrement dit, la modification de la valeur finale d'une de ses deux entrées provoquerait un changement de la valeur de sortie. Ainsi, les lignes  $b$  et  $c$  doivent être ajoutées à la liste de suspects avec le symbole associé à chacune d'entre elles. Cette opération est répétée pour toutes les entrées sensibles des portes (entrées marquées par un point noir). La liste de suspects ainsi obtenue à partir de la sortie  $S_2$  est la suivante :

$$L_{S_2, V_{21}} = \{(E_1, R1), (E_2, R1), (E_3, C1), (a, F0), (b, R1), (c, C1), (S_2, R1)\}.$$

Cette première liste de suspects est composée de sept lignes pouvant être la cause de l'erreur observée sur la sortie  $S_2$ . Ce processus de traçage de chemins critiques est répété pour tous les vecteurs ayant propagé des erreurs en sortie (Tableau 5). Si on suppose que trois vecteurs ( $V_{21}$ ,  $V_{54}$ ,  $V_{87}$ ) ont généré des erreurs, toutes sur la sortie  $S_2$  ( $F \rightarrow \text{Fail}$ ), le processus de traçage de chemins critiques doit donc être répété pour les deux sorties fautives correspondantes aux vecteurs  $V_{54}$  et  $V_{87}$ .

Vf	E1	E2	E3	S1	S2	S3
V21	R1	R1	C1	P	F	P
V54	R1	F0	C1	P	F	P
V87	R1	R1	C0	P	F	P

**Tableau 5** : Vecteurs et réponses associées

Nous obtenons ainsi les deux listes de suspects supplémentaires suivantes :

$$L_{S_2-V54} = \{(E_1, R1), (E_3, C1), (a, F0), (b, R1), (c, C1), (S_2, R1)\}$$

$$L_{S_2-V87} = \{(E_1, R1), (E_2, R1), (a, F0), (b, R1), (c, R1), (S_2, R1)\}$$

A la fin de cette première étape, nous avons donc trois listes de suspects composées chacune de plusieurs lignes pouvant être la cause des erreurs observées sur la sortie  $S_2$ .

### II.2.3. Processus d'intersection

Le processus de traçage de chemins critiques est réalisé à partir de toutes les sorties déclarées « fail » lors du test. Ce processus fournit, pour chacune de ces sorties, une liste de lignes critiques auxquelles sont associées le symbole représentant le signal. Les sites de pannes ayant conduit aux erreurs observées sont nécessairement présents dans ces listes. Si l'on se place maintenant dans l'hypothèse de la panne simple, le site de la panne est nécessairement présent dans toutes les listes. Dans cette hypothèse, les lignes critiques du circuit peuvent ainsi être déterminées par intersection des listes produites par le processus de traçage de chemins critiques.

#### II.2.3.a. Règles d'intersection

Les listes générées par le processus de traçage de chemins critiques comprennent des éléments (couples) constitués d'un site de panne potentiel et de la valeur du signal associé à ce site. Ces deux paramètres sont à prendre en compte par le processus d'intersection.

Soient deux listes,  $L_1$  et  $L_2$ , produites par le processus de traçage de chemins critiques. Soit  $L_s = L_1 \cap L_2$ .  $L_s$  résulte de l'intersection entre chaque élément  $(LC_i, Si)_1$  de  $L_1$  et chaque

élément  $(LC_j, S_j)_2$  de  $L_2$ . L'intersection entre ces éléments est définie de la manière suivante :

- Si  $LC_i \neq LC_j$  alors  $(LC_i, S_i)_1 \cap (LC_j, S_j)_2 = \emptyset$
- Si  $\cap_s S_j$  non défini alors  $(LC_i, S_i)_1 \cap (LC_j, S_j)_2 = \emptyset$
- Sinon  $(LC, S_i)_1 \cap (LC, S_j)_2 = (LC, S_i \cap_s S_j)$

Où  $\cap_s$  est l'opération d'intersection portant sur les signaux associés à une ligne critique. Pour les pannes considérées dans ce chapitre (pannes simples produisant des erreurs sur une seule connexion et ayant toujours la même polarité), cette opération d'intersection est donnée par le Tableau 6.

$\cap_s$	C0	C1	F0	R1	P0	P1
C0	C0	-	C0	-	C0	-
C1	-	C1	-	C1	-	C1
F0	C0	-	F0	-	F0	-
R1	-	C1	-	R1	-	R1
P0	C0	-	F0	-	P0	-
P1	-	C1	-	R1	-	P1

**Tableau 6** : Opération d'intersection entre signaux

Le symbole « - » représente une intersection non définie. Si ce cas se produit, l'erreur sur la connexion ne peut en aucun cas venir d'une défaillance affectant la connexion en question. Elle vient nécessairement de la propagation d'une erreur amont. La défaillance ne pouvant être localisée sur la connexion en question, cette connexion est donc à exclure de la liste des lignes critiques. En effet, si nous considérons le symbole C0 qui sensibiliserait un collage à '1' et le symbole C1 qui sensibiliserait un collage à '0', l'intersection de ces deux symboles conduit à un ensemble vide (dans l'hypothèse de la panne simple produisant des erreurs sur une seule connexion et ayant toujours la même polarité). Dans cette hypothèse, ce cas est donc impossible.

De la même façon, l'intersection d'un symbole représentant une valeur stable (C0, C1) et d'un symbole représentant une commutation (F0, R1, P0, P1) (ou possibilité de commutation) à valeur finale identique  $\{(C1, R1), (C1, P1), (C0, F0), (C0, P0)\}$  donne comme résultat le symbole représentant une valeur stable (C0, C1). En effet, les symboles représentant une valeur stable ne sensibilisent que les fautes statiques (collage, circuit ouvert, ...) alors que les

symboles représentant une transition (ou possibilité de transition) sensibilisent à la fois les fautes statiques et dynamiques. Dans ce cas, le résultat de l'intersection est donc le symbole représentant la valeur stable (C0, C1).

**Exemple :**

Reprenons les trois listes de suspects de l'exemple précédent :

$$L_{S2\_v21} = \{(E_1, R1), (E_2, R1), (E_3, C1), (a, F0), (b, R1), (c, C1), (S_2, R1)\}.$$

$$L_{S2\_v54} = \{(E_1, R1), (E_3, C1), (a, F0), (b, R1), (c, C1), (S_2, R1)\}$$

$$L_{S2\_v87} = \{(E_1, R1), (E_2, R1), (a, F0), (b, R1), (c, R1), (S_2, R1)\}$$

Le résultat de l'intersection de ces trois listes est :

$$L_F = \{(E_1, R1), (a, F0), (b, R1), (c, C1), (S_2, R1)\}$$

Les lignes d'entrées  $E_2$  et  $E_3$  n'ont pas été conservées car elles n'étaient pas présentes dans les trois listes de suspects. Les lignes  $E_1$ ,  $a$ ,  $b$ ,  $S_2$  conservent les mêmes symboles car ils ne changent pas d'une liste de suspects à l'autre. Enfin, la ligne  $c$  est conservée car elle est présente dans l'ensemble des trois listes de suspects, mais l'intersection de ses symboles (C1, C1, R1) donne le symbole final C1 (cf. Tableau II.3).

II.2.3.b. Implantation informatique

L'implantation informatique du processus d'intersection est basée sur la manipulation de deux paramètres associés à chacune des lignes du circuit lors du processus de traçage de chemins critiques. Le premier paramètre (*CNT*) est en fait un compteur qui est incrémenté chaque fois que le processus de traçage de chemins critiques passe sur la ligne en question. Le second paramètre (*VAL*) est le résultat de l'intersection (cf Tableau 6) entre la valeur précédente de *VAL* et la valeur courante du signal porté par la ligne. Ainsi, le processus de traçage de chemins ne construit pas à proprement parler une liste comme précédemment illustré, mais pour chaque ligne rencontrée, met à jour ces indicateurs.

A la fin du processus de traçage de chemins critiques, dans l'hypothèse de la panne

simple, les connexions susceptibles de porter la panne sont uniquement les lignes dont le paramètre CNT est égal au nombre d'erreurs observées (nombre de fois où le processus de traçage de chemins a été activé).

Cette implantation informatique a l'avantage de ne pas mettre en œuvre de processus complexe d'intersection de liste, mais aussi, comme nous le verrons dans le chapitre suivant de pouvoir être utilisée en l'état pour traiter les pannes multiples ou pannes à effets multiples.

**Exemple :**

Reprenons le processus de traçage de chemins critiques et d'intersection tel qu'il a été implanté informatiquement (Tableau 7). Nous pouvons voir dans ce tableau, trois différentes parties (CPT L<sub>S2</sub>\_V21, CPT L<sub>S2</sub>\_V21, CPT L<sub>S2</sub>\_V21) qui correspondent à la mise à jour du compteur (CNT) et de la valeur (VAL) pour chaque nouvelle application du processus de traçage de chemins critiques à partir de la sortie S2. La première partie du tableau (CPT L<sub>S2</sub>\_V21) montre les lignes suspectes obtenues à partir de la sortie S2 et pour le vecteur V21. Sept lignes critiques ont été trouvées (E1, a, b, c, S2, E3, E2) puisque le compteur CNT de chacune de ces lignes a été incrémenté (à l'initialisation CNT=0). De plus, nous pouvons voir que la colonne VAL contient les symboles portés par ces lignes au moment de l'application de ce premier vecteur fautif. Les deux autres parties de ce tableau permettent de voir les mises à jour du CNT et de VAL pour le traçage de chemins critiques à partir des deux autres vecteurs fautifs (V45 et V78). Nous pouvons ainsi voir que pour le vecteur V45 ayant entraîné la sortie S2 fautive, six lignes ont été tracées puisque leur CNT associé a été mis à jour (CNT=2). De plus les symboles des lignes E3 et c ont été également mis à jour à partir du tableau d'intersection vu précédemment (Tableau 6). Enfin, la dernière partie de ce tableau donne la liste finale de suspects puisqu'elle correspond à la dernière application du processus de traçage, c'est-à-dire pour le dernier vecteur fautif V87. Le compteur CNT a à nouveau été mis à jour. Nous pouvons voir que cinq lignes ont un CNT=3 et deux lignes ont un CNT=2. Dans l'hypothèse de la panne simple, nous conserverons uniquement les lignes ayant un CNT égal au nombre de sorties fautives, c'est-à-dire CNT=3, puisque seule une de ces lignes peut être l'origine des erreurs observées sur toutes les sorties fautives. Dans ce cas précis, les cinq lignes conservées à la fin du processus d'intersection sont les lignes E1, a, b, c S2.

Lignes	CPT L <sub>s2-v21</sub>		CPT L <sub>s2-v54</sub>		Lf=CPT L <sub>s2-V87</sub>	
	CNT	VAL	CNT	VAL	CNT	VAL
E <sub>1</sub>	1	R1	2	R1	3	R1
a	1	F0	2	F0	3	F0
b	1	R1	2	R1	3	R1
c	1	R1	2	C1	3	C1
S <sub>2</sub>	1	R1	2	R1	3	R1
E <sub>3</sub>	1	R1	2	C1	2	C1
E <sub>2</sub>	1	R1	1	R1	2	R1

Tableau 7 : Implantation du processus d'intersection

II.2.4. Prise en compte des sorties saines

Jusqu'à présent nous n'avons considéré que les sorties fautives du circuit. Or les sorties saines peuvent être aussi utilisées afin de disculper certaines lignes suspectes déterminées lors du processus de traçage de chemins critiques [Abra83]. Dans [Abra80], de la même façon que pour les sorties fautives, les sorties saines peuvent être tracées et, si une ligne est commune à la liste de suspects et à la liste des lignes saines, alors cette ligne peut être retirée de la liste des suspects. Cette observation est illustrée sur la Figure 20. Lors du processus de traçage de chemins à partir de la sortie S<sub>2</sub>, la liste de suspects obtenue est {S<sub>2</sub>, E<sub>1</sub>, E<sub>2</sub>}. Avec l'utilisation de la sortie saine S<sub>1</sub>, la ligne E<sub>1</sub> peut être disculpée, car une panne sur cette ligne propagerait une erreur sur la sortie S<sub>1</sub> en plus de la sortie S<sub>2</sub>. La liste de suspects finale obtenue en considérant les sorties fautives et saines est donc {S<sub>2</sub>, E<sub>2</sub>}.

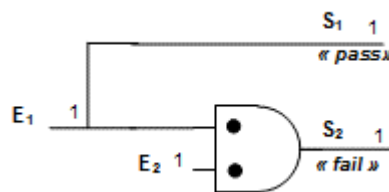


Figure 20 : Exemple de prise en compte des sorties saines

Les expérimentations que nous avons pu faire montrent qu'en règle générale, le nombre de sorties saines est beaucoup plus important que le nombre de sorties fautives. Le Tableau 8 présente un récapitulatif des simulations que nous avons menées sur le b17s de la famille des ITC'99. Ce circuit comporte 1.512 sorties. 1.931 vecteurs de test lui ont été appliqués (séquence déterminée par ATPG). Ce qui signifie que le nombre de sorties



observées est de l'ordre de 2.9 millions pour l'application complète de la séquence de test. Nous avons injecté aléatoirement (une par une) des fautes (colonne scénario) et noté le nombre de sorties fautives (colonne #S<sub>f</sub>) obtenu à la fin de l'application de la séquence de test. Exprimé en pourcentage (colonne %S<sub>f</sub>), nous pouvons voir que le nombre de sorties fautives est considérablement moins élevé que le nombre de sorties saines (colonne %S<sub>p</sub>).

Scenario	#S <sub>f</sub>	%S <sub>f</sub>	%S <sub>p</sub>
P1_U11977 - Sa1	41	0,0014	99,9986
P3_U9191 - Sa0	831	0,0285	99,9715
P3-U11173 => P1_U11264 AND Bridge	15	0,0005	99,9995
P2_U14376 => P2_U9555 OR BRIDGE	971	0,0333	99,9667
P3_INSTQUEUE_REG_15_3 => P2_U12213 AND Bridge	60	0,0021	99,9979
DATAI_31 - Sa0	38	0,0013	99,9987
P2_U10186 => PP1_R1068_U95 AND Bridge	393	0,0135	99,9865
P2_U14148 - Sa0	156	0,0053	99,9947
P3_U13901 - Sa1	52	0,0018	99,9982

**Tableau 8** : Comparaison entre le nombre de sorties fautives et le nombre de sorties saines

Nous faisons de la simulation de fautes car nous ne considérons pas toutes les fautes de collage mais uniquement celles qui sont dans la liste de suspects.

A la fin du processus de traçage de chemins, nous avons obtenu une liste de lignes suspectes ainsi que pour chaque ligne un indicateur (VAL) représentant la succession de valeurs logiques portées par la ligne. A partir de cette liste, nous générons une liste de fautes de collage qui représenteront les erreurs possibles sur ces lignes et que nous utiliserons dans le processus de simulation de faute afin d'analyser la propagation de ces erreurs. Une ligne suspecte avec un indicateur de valeur représentant une valeur finale à 0 (C0,F0,P0) donnera lieu à une panne de collage à 1. Une ligne suspecte avec un indicateur de valeur représentant une valeur finale à 1 (C1,R1,P1) donnera lieu à une panne de collage à 0.

Lors de la simulation de fautes, uniquement deux cas peuvent se présenter pour chacune des fautes simulées :

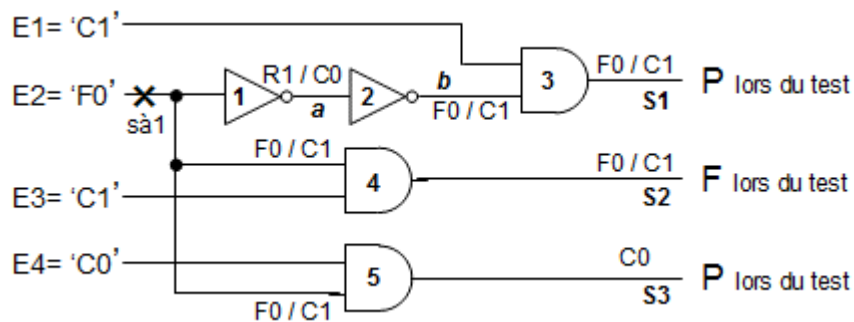
- La faute est observable uniquement sur toutes les sorties déclarées « fail » par le testeur.
- La faute est observable sur toutes les sorties déclarées « fail » par le testeur et sur certaines sorties déclarées « pass »

Le premier cas ne donne pas lieu à un traitement particulier et confirme simplement ce qui a été obtenu par le processus « *Effet à Cause* ». La liste de suspects obtenue initialement ne peut donc pas être réduite.

Par contre dans le second cas, nous devons analyser si les différences apparaissent durant la simulation de fautes avec un vecteur de test fautif ou durant la simulation de fautes avec un vecteur de test sain. Ces informations sont stockées comme « preuve de fautes » en utilisant deux marqueurs (\* et +).

Le **marqueur \*** pour une ligne suspecte (dont le symbole est alors codé par C0\*, C1\*, F0\*, R1\*, P0\* ou P1\*) indique que la simulation d'une faute de collage sur cette ligne a fourni des résultats qui diffèrent de ceux obtenus lors du test quand des vecteurs de test **fautifs** sont considérés. D'un autre côté, la simulation de fautes de collage avec des vecteurs de test sains a fourni les mêmes résultats que ceux fournis par le tester.

**Exemple :**



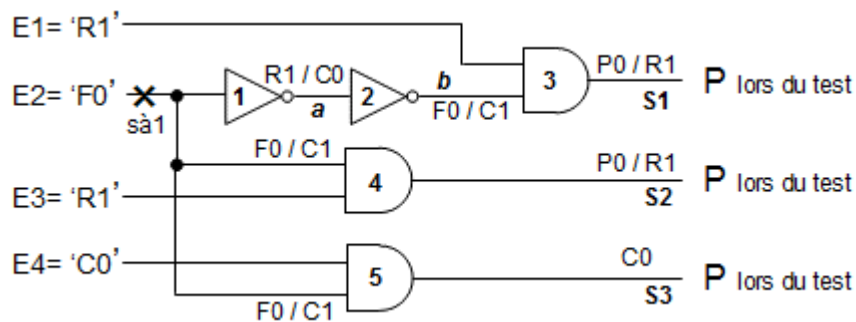
**Figure 21** : Simulation du circuit pour un vecteur fautif

La Figure 21 représente un circuit logique composé de cinq portes, quatre entrées et trois sorties. La paire de vecteur appliquée à ce circuit est représentée par les symboles (C1,F0,C1,C0). Supposons que lors de la phase de test, seule la sortie S<sub>2</sub> ait été déclarée fautive. Durant l'opération de traçage de chemins critiques à partir de la sortie S<sub>2</sub>, la liste de suspects obtenue sera {(S<sub>2</sub>,F0) et (E<sub>2</sub>,F0)}. Simulons maintenant la faute de collage à '1' sur la ligne E<sub>2</sub> avec ce même vecteur fautif. Nous pouvons voir que cette faute est observable sur la sortie S<sub>2</sub>, mais également sur la sortie S<sub>1</sub>. Nous pouvons donc en déduire qu'une faute statique (cà1) ne peut affecter la ligne E<sub>2</sub> (sinon elle aurait été observée sur S<sub>1</sub> et S<sub>2</sub> lors du

test). Seul un modèle de faute dynamique peut être la cause de l'erreur observée sur la sortie  $S_2$  durant le test. Nous mettrons donc à jour le symbole  $F_0$  associé à la ligne  $E_2$  avec le marqueur '\*' afin de prendre en considération ces observations lors de la phase d'allocation des modèles de fautes. La nouvelle liste de suspects est donc  $\{(S_2, F_0) \text{ et } (E_2, F_0^*)\}$ .

De la même façon, le **marqueur '+'** pour une ligne suspecte (dont le symbole est alors codé par  $C_0+$ ,  $C_1+$ ,  $F_0+$ ,  $R+$ ,  $P_0+$  ou  $P_1+$ ) indique que la simulation de fautes de collage sur cette ligne a fourni des résultats qui diffèrent de ceux obtenus à partir du testeur quand des vecteurs de test **sains** sont considérés. D'un autre côté, la simulation de fautes de collages avec des vecteurs de test fautifs a fourni les mêmes résultats que ceux fournis par le testeur.

**Exemple :**



**Figure 22** : Simulation du circuit pour un vecteur sain

Reprenons la figure précédente, et considérons le vecteur  $(R_1, F_0, R_1, C_0)$  déclaré sain lors du test (Figure 22). Supposons que la liste de suspects obtenue lors de l'opération de traçage de chemins critiques pour de précédents vecteurs fautifs avait donné  $\{(S_2, F_0) \text{ et } (E_2, F_0)\}$ . Simulons maintenant la faute de collage à '1' de la ligne  $E_2$  avec le vecteur sain. Comme nous pouvons le voir sur la Figure 22, la faute est observable sur les sorties  $S_1$  et  $S_2$ . Nous pouvons donc en déduire qu'une faute statique sur  $E_2$  (çà1) aurait rendu le vecteur fautif lors du test. Cela signifie que les modèles de fautes statiques ne peuvent pas être la cause des erreurs observées, excepté le modèle de court-circuit. En effet, pour ce modèle de faute, nous ne pouvons pas garantir que l'agresseur est sensibilisé et il n'est donc pas possible de disculper la ligne victime. Nous ne pouvons donc pas éliminer ce modèle de faute de ceux

associés à cette ligne ( $E_2$ ) lors de la phase d'allocation. Nous mettrons donc à jour le symbole  $F_0$  associé à la ligne  $E_2$  avec le marqueur '+' afin de prendre en considération ces observations lors de la phase d'allocation des modèles de fautes. La nouvelle liste de suspects est donc  $\{(S_2, F_0) \text{ et } (E_2, F_0+)\}$ .

En conclusion, une ligne suspecte marquée par '\*' indique que la ligne ne peut être affectée que par des fautes dynamiques si elle est à l'origine de l'erreur observée. Une ligne suspecte marquée par '+' indique que la ligne ne peut être affectée que par des fautes dynamiques ou court-circuit.

### **Implantation informatique :**

L'implantation de cette partie du processus de diagnostic a été faite en apportant une information supplémentaire au niveau des symboles. Nous considérerons une liste de suspects déjà établie lors du processus de traçage de chemins critiques. Les conditions seront les suivantes, chaque ligne de la liste de suspects est testée par simulation de fautes et si les réponses du circuit diffèrent de celles obtenues lors du test :

- Pour un vecteur fautif à partir d'une sortie saine et:
  - ⇒ Si son symbole représente une valeur stable ( $C_0$  ou  $C_1$ ), alors cette ligne peut être supprimée de la liste de suspects (car l'ensemble des modèles de fautes associés à ces deux symboles sont des fautes statiques qui peuvent donc être éliminés).
  - ⇒ Si son symbole représente une valeur de transition ( $F_0$  ;  $R_1$ ,  $P_0$ ,  $P_1$ ), alors on lui ajoute une étoile qui symbolise que l'ensemble des modèles de fautes statiques associés à cette ligne pourront être supprimés. Les nouveaux symboles sont alors ( $F_0^*$ ,  $R_1^*$ ,  $P_0^*$ ,  $P_1^*$ ).
- Pour un vecteur sain et donc une entrée saine :
  - ⇒ Si son symbole représente une valeur stable ( $C_0$  ou  $C_1$ ), alors on lui ajoute un '+' qui symbolise que l'ensemble des modèles de fautes statiques peuvent être supprimés excepté le modèle de court-circuit strong driver wired and ou or. Les

nouveaux symboles sont alors (C0+, C1+).

⇒ Si son symbole représente une valeur de transition (F0; R1, P0, P1), alors on lui ajoute également un ‘+’ qui symbolise que l’ensemble des modèles de fautes statiques peuvent être supprimés excepté le modèle de court-circuit strong driver wired and ou or. Les modèles de fautes dynamiques sont quand à eux conservés également. Les nouveaux symboles sont alors (F0+, R1+, P0+, P1+).

**Exemple :**

Reprenons la liste de suspects obtenue dans le paragraphe II.2.3.a, c’est à dire  $L_F = \{(E_1, R1), (a, F0), (b, R1), (c, C1), (S_2, R1)\}$ . La Figure 23 montre les lignes suspectes déterminées durant le processus de traçage de chemins (pointillés). Pour chacune de ces lignes, nous faisons la simulation de fautes pour l’ensemble de la séquence de vecteurs (fautifs et sains).

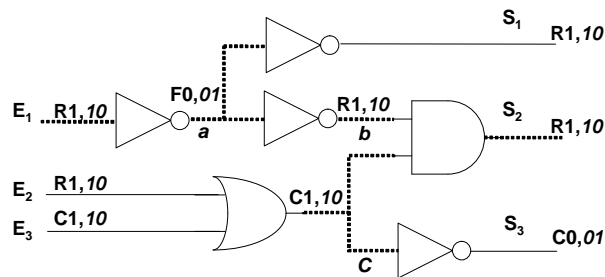


Figure 23 : Simulation de fautes

Vf	E1	E2	E3	S1	S2	S3
V21	R1	R1	C1	P	F	P
V54	R1	F0	C1	P	F	P
V87	R1	R1	C0	P	F	P

Tableau 9 : vecteurs et réponses associés

Le Tableau 9 rappelle les réponses fournies par le circuit fautif lors de la phase de test. Nous pouvons voir que seul la sortie S2 est déclarée fautive. Or lorsque nous avons vérifié la propagation de l’erreur pour ces vecteurs fautifs et pour les lignes E1 et a, une erreur a été observée sur la sortie S1 pourtant déclarée bonne durant le test. De la même façon, lors de la

simulation de fautes pour la ligne  $c$  la sortie  $S3$  a été déclarée fautive. Cela signifie que les modèles de fautes statiques ne peuvent pas être la cause des erreurs observées (puisque nous considérons, pour l'instant, uniquement des vecteurs fautifs). Nous pouvons donc associer les symboles des lignes  $E_1$ ,  $a$ , et  $c$  avec le marqueur '\*'. La nouvelle liste de suspects mise à jour est donc :

$$L_F = \{(E_1, R1^*), (a, F0^*), (b, R1), (c, C1^*), (S_2, R1)\}$$

La simulation de fautes n'ayant pas fourni de sorties fautives pour des vecteurs sains, aucune ligne voit son symbole associé avec le marqueur '+'.

### II.3. Phase d'allocation des modèles de fautes

Les modèles de fautes considérés dans cette première partie d'étude sont les fautes de collage, de transistors stuck-on, de transistors stuck-open, de circuit ouvert, de court circuit de type OU et ET (strong driver wired OR et AND), de court-circuit résistif de type OU et ET (strong driver wired resistive OR et AND), de délai de type L<sub>à</sub>M ou L<sub>à</sub>D.

Nous avons vu dans le chapitre précédent les conditions de sensibilisation de chaque modèle de faute considéré. A l'origine de la simulation du circuit, nous nous sommes servis de ces conditions pour justifier l'utilisation d'un codage à six valeurs. Nous pouvons donc lier le tableau créé dans le chapitre précédent en remplaçant les valeurs logiques par le symbole qu'elles représentent ( $C0='00'$ ,  $C1='11'$ ,  $F0='10'$ , ...).

En considérant également les informations supplémentaires apportées par l'ajout des signes '\*' et '+' dont nous avons expliqué le fonctionnement précédemment, nous avons pu créer le Tableau 10. Ce tableau regroupe dans la première colonne l'ensemble des modèles de fautes pris en compte dans ce chapitre (modèles de fautes n'engendrant que des pannes simples). Les colonnes suivantes fournissent les symboles manipulés ainsi que les modèles de fautes associés à ceux-ci (sélectionnés par une croix).

Les modèles de fautes associés avec une ligne suspecte peuvent être déterminés en fonction du symbole associé avec la ligne. Les lignes marquées avec un symbole stable ( $C0$  ou  $C1$ ) portent seulement des fautes statiques, puisqu' aucune transition n'est apparue sur la ligne et donc aucune faute dynamique a été sensibilisée. Les lignes marquées avec un symbole

représentant une transition (F0, R1, P0, P1) peuvent représenter aussi bien des fautes statiques que dynamiques (colonnes 2 à 5).

Maintenant, si le symbole associé avec une ligne suspecte est marquée avec une ‘\*’, cela signifie que les modèles de fautes sélectionnés ne peuvent pas être statiques. Comme le montre le Tableau 10 (colonnes 6 à 9), les lignes marquées avec C0\* et C1\* peuvent par conséquent être complètement retirées de la liste des lignes suspectes pendant que les modèles de fautes associés avec les symboles représentant des transitions (F0\*, R1\*, P0\*, P1\*) sont considérés comme dynamiques seulement (délai et court-circuit résistif).

Quand les symboles sont marqués avec un ‘+’, l’allocation des modèles de fautes est plus complexe. Tout d’abord, la ligne peut être innocentée d’être affectée seulement par des fautes statiques impliquant une ligne. Ensuite, puisque dans la simulation de fautes de collage des vecteurs de test sains, la condition d’activation sur la ligne agresseur (un ‘0’ ou ‘1’ logique dépendant du type de court-circuit (OU/ET)) ne peut être garanti, nous ne pouvons pas exclure ce type de fautes comme possible cause des erreurs observées (colonnes 10 à 13).

Nous pouvons voir par exemple que le symbole C0+ est associé avec un seul modèle de faute qui est le strong driver wired OR. Le symbole F0\* est associé aux modèles de fautes dynamiques tels que le circuit ouvert résistif, strong driver wired Or resistive, délai de type lād. Le symbole F0+ = F0\* av

ec le strong driver wired OR en plus.

Modèles de fautes	C0	C1	F0,P0	R1,P1	C0*	C1*	F0*,P0*	R1*,P1*	C0+	C1+	F0+,P0+	R1+,P1+
Collage à 0		x		x								
Collage à 1	x		x									
Tn Stuck-open	x		x									
Tn Stuck-on		x		x								
Tp Stuck-open		x		x								
Tp Stuck-on	x		x									
circuit ouvert à 0		x		x								
circuit ouvert à 1	x		x									
strong driver wired OR (avec une ligne à 1)	x		x						x		x	
strong driver wired AND (avec une ligne à 0)		x		x						x		x
Strong driver wired Or resistive			x				x				x	
Strong driver wired And resistive				x				x				x
Delai lent-à-descendre			x				x				x	
Delai lent-à-monter				x				x				x

Tableau 10 : Modèles de fautes associés avec les lignes critiques en fonction du symbole

**Exemple :**

Considérons l'exemple utilisé durant tout ce chapitre et dont la liste finale de suspects est représentée ci-dessous :

$$\mathbf{L}_F = \{(\mathbf{E}_1, \mathbf{R1}^*), (\mathbf{a}, \mathbf{F0}^*), (\mathbf{b}, \mathbf{R1}), (\mathbf{c}, \mathbf{C1}^*), (\mathbf{S}_2, \mathbf{R1})\}$$

Dans ce cas, l'association des modèles de fautes avec la liste de suspects et d'après le Tableau 10 est :

- Ligne  $\mathbf{E}_1$  : court-circuit résistif (avec une ligne à 0), délai (LàM).
- Ligne  $\mathbf{a}$  : court-circuit résistif (avec une ligne à 1), délai (LàD).
- Lignes  $\mathbf{b}$  et  $\mathbf{S}_2$  : Collage à 0, Tn stuck-on, Tp Stuck-open, circuit ouvert à 0, court-circuit (ET), court-circuit résistif, délai (LàM).
- Lignes  $\mathbf{c} = \emptyset$ , aucun modèle de fautes, la ligne est éliminée de la liste des suspects.

Cette seconde phase qui est l'association des modèles de fautes avec les lignes critiques termine le processus de diagnostic. Nous obtenons donc quatre lignes auxquelles sont associées un ensemble précis de modèles de fautes.





## Conclusion

Nous avons développé une méthode de diagnostic en deux phases. La première phase de recherche des lignes suspectes comprend une partie de simulation. Celle-ci est basée sur un codage multivalué qui remplit toutes les conditions de sensibilisation requises pour l'ensemble des modèles de fautes décrits dans le chapitre précédent. La simulation nous permet de connaître les valeurs portées par chaque ligne dans le circuit. A partir de là, le processus de traçage de chemin à partir des sorties fautives permet de constituer une première liste de lignes suspectes réduite. Toutes les lignes suspectes ont chacune deux paramètres associés, le symbole porté ainsi que le nombre de fois que chaque ligne a été tracée. Avec l'hypothèse de panne simple, seules les lignes suspectes avec un CNT égal au nombre de sorties fautives sont conservées. Enfin, afin de disculper certaines lignes ou certains modèles de fautes qui seront associés par la suite à ces lignes, une opération de simulation de fautes basée sur des fautes de collage est effectuée. Cette opération de simulation de fautes permet de vérifier la propagation de l'erreur à partir de chaque ligne suspecte (déterminées lors du processus de traçage de chemins) et ainsi de comparer les résultats obtenus avec ceux du testeur. Néanmoins, des observations ont été faites quand à la nature des modèles de fautes disculpés par cette opération de simulations de fautes (statique, dynamique, sensibilisation de l'agresseur dans le cas des courts-circuits).

Enfin, la seconde phase qui termine ce processus de diagnostic permet d'associer les modèles de fautes considérés dans cette partie avec les lignes finales suspectes déterminées à la fin du processus de simulation de fautes.



# **III - Optimisations et adaptations de la méthode de diagnostic**



## Introduction

Ce chapitre est consacré à la description des adaptations de la méthode de diagnostic permettant d'optimiser la précision ou de prise en compte de modèles de pannes particuliers. Il sera scindé en trois parties.

La première partie, est consacrée à une adaptation de la méthode de diagnostic qui permet d'optimiser la précision du diagnostic pour les modèles de fautes de stuck-on/open et de court-circuit. Ces pannes sont des pannes simples prises en compte par le processus présenté dans le chapitre précédent. Néanmoins, l'utilisation d'informations complémentaires produites lors de la simulation effectuée durant la phase de localisation des sites de panne peut permettre d'améliorer la précision du diagnostic.

Dans la seconde partie, nous présenterons les adaptations de la méthode de diagnostic qui nous permettront de prendre en compte des pannes dont l'effet n'affecte qu'une connexion mais qui peut être de deux polarités différentes (0/1 ou 1/0) selon le vecteur de sensibilisation. Ces pannes qui n'étaient pas traitées jusqu'à présent sont les pannes temporelles affectant les deux fronts montant et descendant (StR & StF) et les pannes de court circuit de type « Strong Driver Wired ».

Enfin, la troisième partie sera consacrée aux adaptations de la méthode de diagnostic permettant de prendre en compte les pannes de court-circuit dont l'effet peut affecter deux connexions. Nous verrons dans un premier temps le cas des pannes dont l'effet se traduit sur l'une ou l'autre des connexions en court-circuit en fonction du vecteur de sensibilisation (wired And/Or, wired And/Or resistive). Nous verrons ensuite, le cas des pannes pouvant affecter simultanément les deux connexions en court-circuit pour un même vecteur de sensibilisation (byzantine). Cette partie nous amènera par ailleurs à envisager le traitement d'erreurs multiples.



### **III.1. Optimisation de la précision du diagnostic pour les modèles de court-circuit et de stuck-on/open**

La méthode de diagnostic détaillée précédemment conduit à obtenir un résultat composé de lignes suspectes avec des modèles de fautes associés. Cette association de modèles de fautes est obtenue en fonction des observations faites sur les conditions de sensibilisation, c'est-à-dire en fonction du symbole porté par chaque ligne suspecte. La précision de ces résultats peut être optimisée en considérant d'autres paramètres. Ainsi, pour le modèle de court-circuit, nous pouvons chercher à déterminer quelles sont les lignes agresseur potentielles (dans le cas du Strong Driver Wired And/Or et du Strong Driver Wired And/Or resistive). Pour le modèle de stuck-on/open, nous pouvons également cibler plus précisément les transistors susceptibles d'être affectés par des pannes de stuck-on ou stuck-open.

#### **III.1.1.Principe**

Les informations utilisées pour optimiser le diagnostic sont les valeurs propagées sur chaque ligne du circuit durant l'application des vecteurs fautifs. Ces informations sont en fait représentées par une variable « flag » associée à chaque ligne du circuit lors de la simulation effectuée durant la phase de localisation des sites de panne. Nous avons vu dans le chapitre II (cf §II.2.1.b) comment était initialisée cette variable et comment elle était mise à jour en fonction des vecteurs appliqués au circuit. Notons ici que cette variable est mise à jour uniquement lors de l'application de vecteurs fautifs.

#### **III.1.2.Cas des pannes de court circuit de type « Strong Driver Wired And/Or et Resistive Strong Driver Wired And/Or»**

Nous considèrerons dans cette partie uniquement les courts-circuits qui provoquent une erreur sur une seule des deux lignes en court circuit (erreur simple) pour une condition de sensibilisation unique (Strong Driver Wired Or, Strong Driver Wired And, Resistive Strong Driver Wired Or, Resistive Strong Driver Wired And).



Lorsque le circuit est affecté par un de ces modèles de court-circuit, nous pouvons déterminer, parmi l'ensemble des lignes du circuit, un sous ensemble de lignes potentiellement agresseur. Deux cas sont à envisager :

- La ligne est affectée par un court-circuit de type Strong Driver Wired Or ou Resistive Strong Driver Wired Or. Dans ce cas, la ligne fautive a systématiquement une valeur finale à 0 pour les vecteurs fautifs (car nous avons vu dans le chapitre précédent que pour ce type de court-circuit la ligne victime doit avoir la valeur logique '0'). Les lignes potentiellement agresseur sont donc les lignes qui ont toujours la valeur logique '1' pour les vecteurs fautifs.

- La ligne est affectée par un court-circuit de type Strong Driver Wired And ou Resistive Strong Driver Wired And. Dans ce cas, la ligne fautive a systématiquement une valeur finale à 1 pour les vecteurs fautifs (car nous avons vu dans le chapitre précédent que pour ce type de court-circuit la ligne victime doit avoir la valeur logique '1'). Les lignes potentiellement agresseur sont donc les lignes qui ont toujours la valeur logique '0' pour les vecteurs fautifs.

Afin de réaliser cette opération de recherche de la ligne agresseur, nous allons nous appuyer sur la variable « flag » introduite dans le chapitre précédent (cf. §II.2.1.b). Cette variable permet de conserver l'historique des valeurs logiques portées par chaque ligne du circuit lors de l'application des vecteurs produisant une erreur sur le testeur (vecteurs fautifs). Comme nous l'avons vu dans le chapitre II, le « flag » associé à une ligne qui aura porté pour l'ensemble des vecteurs fautifs des symboles à valeur finale '1', aura pour valeur '10'. De la même façon, le « flag » associé à une ligne qui aura porté pour l'ensemble des vecteurs fautifs des symboles à valeur finale '0', aura pour valeur '01'.

Par conséquent, le processus de détection des pannes de type Strong Driver Wired Or, Strong Driver Wired And, Resistive Strong Driver Wired Or, Resistive Strong Driver Wired And est le suivant :

⇒ Si la ligne critique déterminée durant le processus de traçage de chemins porte le symbole C0, C0+, F0, F0+, P0, P0+, un des modèles de fautes pouvant être associé à cette ligne est le Strong Driver Wired OR avec comme agresseurs toutes les lignes du circuit dont le « flag » a pour valeur '10'.

⇒ Si la ligne critique déterminée durant le processus de traçage de chemins porte le symbole C1, C1+, R1, R1+; P1, P1+, un des modèles de fautes pouvant être associé à cette ligne est le Strong Driver Wired AND avec comme agresseurs toutes les lignes du circuit dont le « flag » a pour valeur '01'.

⇒ Si la ligne critique déterminée durant le processus de traçage de chemins porte le symbole F0, P0, F0\*, P0\*, F0+, P0+, un des modèles de fautes pouvant être associé à cette ligne est le Resistive Strong Driver Wired OR avec comme agresseurs toutes les lignes du circuit dont le « flag » a pour valeur '10'.

⇒ Si la ligne critique déterminée durant le processus de traçage de chemins porte le symbole R1, P1, R1\*, P1\*, R1+, P1+, un des modèles de fautes pouvant être associé à cette ligne est le Resistive Strong Driver Wired AND avec comme agresseurs toutes les lignes du circuit dont le « flag » a pour valeur '01'.

### III.1.3.Cas des pannes de stuck-on et stuck-open

Une des conditions de sensibilisation d'une panne de stuck-on sur un transistor est l'application de la valeur de blocage sur sa grille (0 pour un transistor N, 1 pour un P) et une des conditions de sensibilisation d'une panne de stuck-open sur un transistor est l'application de la valeur d'activation sur sa grille (1 pour un transistor N, 0 pour un P).

Ainsi, lorsqu'une erreur est observée en sortie du circuit, si la condition de sensibilisation de ce type de panne (stuck-on ou stuck-open) n'est pas remplie sur le transistor incriminé, la panne en question peut être disculpée. En complément au processus de diagnostic élaboré jusqu'à présent, nous pouvons donc chercher à disculper certaines pannes de stuck-on et stuck-open en étudiant les signaux portés par les entrées de la porte amont à la ligne suspecte.

A titre d'exemple, considérons une porte NAND vue au niveau transistor (Figure 24). Le Tableau 11 représente les différentes valeurs logiques pouvant être appliquées à cette porte (colonnes A et B) ainsi que la valeur de sortie attendue. Si la valeur logique obtenue en sortie n'est pas celle attendue, l'étude des valeurs d'entrées permet de cibler plus précisément la panne (colonne 4, failure transistor).

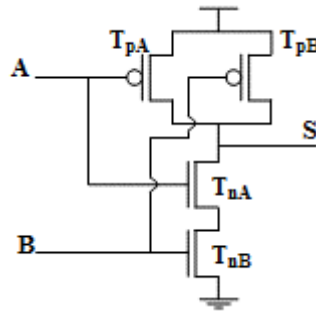


Figure 24 : Porte logique NAND au niveau transistor

A	B	S	failure transistor
0	0	1	impossible
0	1	1	TpA stuck-open or TnA stuck-on
1	0	1	TpB stuck-open or TnB stuck-on
1	1	0	TpA or TpB stuck-on or TnA or TnB stuck-open

Tableau 11 : Transistors défaillants en fonction des valeurs d'entrées

D'autre part, dans l'hypothèse de la panne simple, si une porte logique a un de ses transistors défaillant et que celui-ci est la cause des erreurs observées en sortie, alors les valeurs logiques d'entrée de la porte sont nécessairement identiques pour l'ensemble des vecteurs fautifs.

Ainsi, en fonction de la porte logique considérée (AND/OR/NOR/...), des valeurs logiques d'entrée et du fait que ces valeurs sont constantes ou pas pour l'ensemble des vecteurs fautifs, plusieurs cas sont à considérer :

- Pour chaque vecteur fautif, toutes les valeurs d'entrée de la porte sont constantes.

⇒ Si toutes les entrées sont à la valeur non prioritaire, tous les transistors restent la cause potentielle des erreurs observées (Stuck-on ou Stuck-open en fonction de la fonction réalisée par la porte). Pour une porte NAND il s'agit par exemple d'une panne de Stuck-open sur les transistors N ou Stuck-on sur les transistors P.

⇒ Si une seule entrée est à la valeur prioritaire, uniquement les 2 transistors correspondant à cette entrée peuvent être à l'origine de la panne observée (Stuck-on ou stuck-open en fonction de la fonction réalisée par la porte).

Pour une porte NAND il s'agit par exemple d'une panne de Stuck-on sur le transistor N ou Stuck-open sur le transistor P.

⇒ Si plusieurs entrées sont à la valeur prioritaire, aucune des pannes de stuck-on et stuck-open sur les transistors de cette porte ne peuvent être à l'origine des erreurs observées.

- Pour chaque vecteur fautif, une entrée au moins n'est pas constante.

⇒ Dans ce cas, aucune des pannes de stuck-on et stuck-open sur les transistors de cette porte ne peut être à l'origine des erreurs observées. En effet, si l'on considère par exemple le Tableau 11, nous pouvons voir que si l'entrée A est constante à '1' et que l'entrée B ne l'est pas (c'est-à-dire que la ligne B a porté des valeurs logiques '1' mais aussi '0' pour les vecteurs fautifs), alors aucun transistor ciblé n'est commun aux deux listes proposées. Dans l'hypothèse de la panne simple, les modèles de fautes de stuck-on et stuck-open peuvent donc être retirés de la liste des pannes suspectes

### III.1.4.Exemple d'optimisation de la précision du diagnostic pour les modèles de court-circuit et de stuck-on/open

Considérons le circuit de la Figure 25. Ce circuit composé de six entrées et trois sorties, correspond à l'état du circuit après l'application du dernier vecteur fautif. Ainsi, nous pouvons observer pour l'ensemble des lignes, l'historique des valeurs logiques appliquées à celles-ci au travers de la variable « flag ».

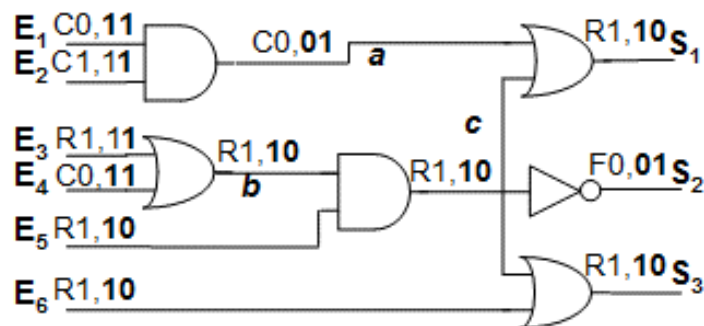


Figure 25 : Historique du circuit au travers du « flag »

En considérant la méthode de diagnostic telle qu'elle a été développée dans le chapitre précédent, la liste de suspects obtenue après l'étape de vérification de la propagation de l'erreur (Effet à Cause) serait la suivante :

$$L_F = \{(c,R1), (b,C1), (E5,R1)\}$$

L'ensemble des pannes potentielles est déterminé à partir de cette liste (cf § II.3). Le Tableau 12 présente le résultat du diagnostic avant prise en compte de la variable répercutant l'évolution des valeurs logiques sur les lignes du circuit (flag). La ligne suspecte est mentionnée dans la première colonne. La seconde colonne correspond au symbole associé à la ligne suspecte. Enfin la dernière colonne correspond aux modèles de fautes associés à chaque ligne suspecte.

L	S	Modèles de fautes
c	R1	Collage à 0, Tn stuck-on, Tp stuck-open, circuit ouvert à 0, strong driver wired AND, strong driver wired AND resistive, délai de type L&D
b	C1	Collage à 0, Tn stuck-on, Tp stuck-open, circuit ouvert à 0, strong driver wired AND
E5	R1	Collage à 0, circuit ouvert à 0, strong driver wired AND, strong driver wired AND resistive, délai de type L&D

**Tableau 12** : Modèles de fautes associés aux lignes suspectes

A ce stade, nous pouvons maintenant préciser :

- Les lignes qui peuvent effectivement produire les courts-circuits mentionnés.
- Les transistors potentiellement affectés par les pannes des stuck-on et stuck-open.

#### III.1.4.a. Amélioration du diagnostic pour les fautes de court-circuit

Nous avons vu lors du traitement des différents modèles de fautes que les courts-circuits dominants étaient composés d'une ligne *victime* (ligne qui va être retrouvée durant le processus de diagnostic, étant donné que c'est le site où l'erreur apparaît) et d'un *agresseur* qui impose sa valeur logique à la ligne victime. Nous allons donc chercher à déterminer tous les agresseurs possibles qui peuvent imposer leurs valeurs aux lignes *c*, *b* et *E5*. S'agissant de pannes de type « strong driver wired AND », uniquement les lignes ayant des

symboles à valeur finale ‘0’ durant l’application des vecteurs fautifs sont des lignes potentiellement agresseurs.

Ces lignes correspondent à celles dont la variable « flag » est à la valeur ‘01’. Dans le cas de cet exemple, ces lignes sont  $S_2$  et  $a$ . Ainsi, les deux seules lignes qui peuvent agresser les trois lignes suspectes sont les lignes  $S_2$  et  $a$ . Le Tableau 13 montre déjà l’amélioration de la précision du diagnostic avec l’ajout des lignes agresseur liées au modèle de court-circuit.

L	S	Modèles de fautes	agresseurs
c	R1	Collage à 0, Tn stuck-on, Tp stuck-open, circuit ouvert à 0, strong driver wired AND, strong driver wired AND resistive, délai de type LàD	S2 et a
b	C1	Collage à 0, Tn stuck-on, Tp stuck-open, circuit ouvert à 0, strong driver wired AND	
E5	R1	Collage à 0, Tn stuck-on, Tp stuck-open, circuit ouvert à 0, strong driver wired AND, strong driver wired AND resistive, délai de type LàD	

**Tableau 13** : Diagnostic affiné pour les courts-circuits

#### III.1.4.b. Amélioration du diagnostic pour les fautes de stuck-on/open

Deux lignes suspectes ( $c$  et  $b$ ) induisent des pannes de stuck-on et stuck open. Ces deux lignes sont respectivement des sorties de portes ET et OU.

Commençons dans un premier temps par le cas de la porte OU associée à la ligne  $b$ . Nous pouvons voir que les variables « flag » associées aux entrées de cette porte ont toutes les deux la valeur ‘11’. Cela signifie, d’après ce que nous avons vu précédemment, que sur chacune de ces deux lignes d’entrée, des symboles à valeur finale ‘0’ mais aussi ‘1’ ont été appliqués. D’un cas à l’autre, les transistors incriminés sont donc nécessairement différents. Dans l’hypothèse de la panne simple, il est donc impossible que les erreurs observées soient dues à un stuck-on ou à un stuck-open des transistors de cette porte OU. Nous pouvons donc retirer ces pannes des modèles de fautes à considérer pour la ligne  $b$ .

Dans le cas de la porte ET dont la sortie est la ligne  $c$ , les « flags » associés aux deux entrées ont une valeur ‘01’, ce qui reflète une valeur constante 1 présente systématiquement sur chacune de ces entrées pour les vecteurs fautifs. Nous sommes dans le cas où toutes les entrées sont constantes à la valeur prioritaire. Dans ce cas, s’agissant d’une porte ET, les pannes de stuck-open ne peuvent affecter que les transistors N et les pannes de stuck-on que

les transistors P.

Le rapport de diagnostic final est donc celui présenté sur le Tableau 14. Nous pouvons voir qu'ainsi, la précision du diagnostic a été améliorée. Nous avons en effet pu réduire le nombre de lignes agresseur potentielles dans le cas du court-circuit ( $S_2$  et  $a$ ), et nous avons pu éliminer le modèle de stuck-on/open pour la ligne  $b$  et cibler les transistors de la ligne  $c$ .

L	S	Modèles de fautes	agresseurs
c	R1	Collage à 0, <b>TpA or TpB stuck-on, TnA or TnB stuck-open</b> , circuit ouvert à 0, strong driver wired And, strong driver wired And resistive, délai de type LàD	<b>S2 et a</b>
b	C1	Collage à 0, circuit ouvert à 0, strong driver wired And	
E5	R1	Collage à 0, circuit ouvert à 0, strong driver wired And, strong driver wired And resistive	

**Tableau 14** : Rapport de diagnostic final

### III.2. Adaptation pour prendre en compte des pannes dont l'effet n'affecte qu'une connexion mais pour deux polarités différentes (0/1 et 1/0)

Parmi l'ensemble des modèles de fautes étudié dans le chapitre I, trois peuvent produire des effets qui n'affectent qu'une connexion mais qui sont tels que ces effets peuvent produire des erreurs de polarités différentes. Ces pannes, qui n'étaient pas prises en compte jusqu'à présent par le processus de diagnostic, sont :

- Les délais du type LàM et LàD
- Le circuit ouvert résistif
- Le court-circuit de type « Strong Driver Wired »

Pour prendre en compte ce type de panne, il faudra considérer lors du processus de localisation des sites potentiels de pannes, que l'erreur observée en sortie peut provenir d'une ligne sur laquelle les deux polarités d'erreur (1/0 et 0/1) peuvent se produire.

Ces observations impliquent des modifications dans le processus de diagnostic telles qu'il a été décrit dans le chapitre II. Ces modifications concernent principalement le processus d'intersection des symboles, et bien évidemment le processus d'allocation des modèles de fautes.

### III.2.1.Modification du processus d'intersection

Le processus d'intersection des symboles a été défini dans le chapitre précédent (cf §II.2.3.a). Le Tableau 15 rappelle les règles d'intersection définies à ce moment.

$\cap_s$	C0	C1	F0	R1	P0	P1
C0	C0	-	C0	-	C0	-
C1	-	C1	-	C1	-	C1
F0	C0	-	F0	-	F0	-
R1	-	C1	-	R1	-	R1
P0	C0	-	F0	-	P0	-
P1	-	C1	-	R1	-	P1

**Tableau 15** : Tableau d'intersection des symboles initial

Lors de ce processus d'intersection, les symboles à valeurs finales opposées, qu'ils représentent des valeurs constantes ou des transitions, donnaient un ensemble vide symbolisé par un tiret ("-"). La ligne associée à ces symboles pouvait alors être supprimée de la liste des lignes suspectes. En effet, ces lignes pouvaient être supprimées car aucun modèle de fautes considéré ne pouvait affecter une même ligne pour les deux polarités d'erreur (1/0 et 0/1). En considérant les trois nouveaux modèles de fautes précédemment cités, cette hypothèse ne tient plus. Pour prendre en compte ces phénomènes, nous avons introduit dans la table d'intersection deux nouveaux symboles. Ces symboles que nous détaillerons par la suite sont les symboles **D** et **SDW**.

#### III.2.1.a. Prise en compte des pannes de délais du type LàM et LàD et des circuits ouverts résistifs.

Les pannes de délai du type LàM et LàD et les circuits ouverts résistifs sont sensibilisés par des commutations qui peuvent être montantes ou descendantes. Selon le type de commutation, l'erreur produite sur la connexion peut ainsi être de « polarité différente (0/1 ou 1/0). Pour prendre en compte les effets de ce type de pannes, le symbole **D** a été introduit. Il traduit l'intersection entre des symboles représentant des transitions ou des possibilités de transition à valeurs finales opposées (F0, P0 et R1, P1). Avec l'ajout de ce symbole, la table d'intersection devient celle présentée sur le Tableau 16.

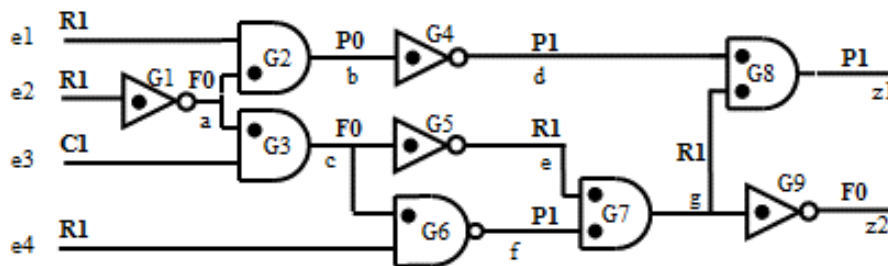


$\cap s$	C0	C1	F0	R1	P0	P1
C0	C0	-	C0	-	C0	-
C1	-	C1	-	C1	-	C1
F0	C0	-	F0	D	F0	D
R1	-	C1	D	R1	D	R1
P0	C0	-	F0	D	P0	D
P1	-	C1	D	R1	D	P1

**Tableau 16** : Tableau d'intersection mis à jour pour le symbole D

**Exemple :**

Considérons le circuit représenté sur la Figure 26 composé de quatre entrées (e1 à e4) et de deux sorties (z1 et z2). Considérons une paire de vecteurs V1 et V2 ayant produit des erreurs lors du test tels que V1 = (R1, R1, C1, R1) et V2 = (R1, C0, R1, R1).



**Figure 26** : Exemple de circuit affecté par une panne pouvant avoir les deux types de polarité (0/1 et 1/0)

Le premier vecteur V1 = (R1, R1, C1, R1) symbolisé sur la Figure 26 a propagé des erreurs sur les sorties z1 et z2. Les listes de suspects obtenues à partir de ces sorties sont les suivantes :

- $L_{1\_V1} = \{(e_2, R1), (a, F0), (b, P0), (c, F0), (d, P1), (e, R1), (f, P1), (g, R1), (z_1, P1)\}$
- $L_{2\_V1} = \{(e_2, R1), (a, F0), (c, F0), (e, R1), (f, P1), (g, R1), (z_2, F0)\}$

Dans l'hypothèse de la panne simple, nous pouvons faire l'intersection des deux listes de suspects obtenues en nous servant du Tableau 16. La nouvelle liste obtenue est la suivante :

- $L_{12} = \{(e_2, R1), (a, F0), (c, F0), (e, R1), (f, P1), (g, R1)\}$

Considérons maintenant le second vecteur fautif  $V2 = (R1, C0, R1, R1)$ . Ce vecteur propage une erreur sur la sortie  $z2$  du circuit. L'opération de traçage de chemins critiques permet d'obtenir la liste de suspects suivante :

- $L_{3\_V2} = \{(e2,C0),(e3,R1),(a,C1),(c,R1),(g,F0),(z2,R1)\}$

Nous devons maintenant renouveler le processus d'intersection entre la liste  $L_{12}$  et  $L_{3\_V2}$ . La liste de suspects obtenue à l'issue de cette intersection est la suivante :

$$L_{123} = \{(c, D), (g,D)\}$$

Les lignes qui n'étaient pas communes aux listes  $L_{12}$  et  $L_{3\_V2}$  (lignes  $z2$ ,  $e3$ ,  $e$ ,  $f$ ) n'ont pas été conservées (hypothèse de la panne simple). Les lignes à valeurs finales opposées mais dont les deux symboles ne représentaient pas chacun des transitions ou possibilités de transitions ont été éliminées (lignes  $e2$  et  $a$ ). Les deux lignes suspectes finales sont donc les lignes  $c$  et  $g$  qui ont chacune le symbole  $D$  associé.

### III.2.1.b.Prise en compte des pannes de court-circuit de type « strong driver wired »

Selon les valeurs logiques portées par deux lignes en court-circuit dû à un « strong driver wired », l'erreur produite sur chacune des deux connexions peut être de polarités différentes (0/1 ou 1/0).

Le symbole  $D$  introduit précédemment pour les pannes de délais permet de considérer ce phénomène lorsque des transitions apparaissent sur le site de la panne. Or, le modèle de court-circuit strong driver wired peut être sensibilisé sans qu'une transition n'apparaisse. Pour prendre en compte ce modèle, nous avons introduit un symbole supplémentaire ; le symbole **SDW** (pour Strong Driver Wired). Avec l'ajout de ce symbole, la table d'intersection devient celle présentée sur le Tableau 17.

		S					
		C0	C1	F0	R1	P0	P1
$S_i$	C0	C0	SDW	C0	SDW	C0	SDW
	C1	SDW	C1	SDW	C1	SDW	C1
	F0	C0	SDW	F0	D	F0	D
	R1	SDW	C1	D	R1	D	R1
	P0	C0	SDW	F0	D	P0	D
	P1	SDW	C1	D	R1	D	P1
	D	SDW	SDW	D	D	D	D
	SDW	SDW	SDW	SDW	SDW	SDW	SDW

**Tableau 17** : Table d'intersection

Chaque colonne dans le Tableau 17 représente le symbole associé avec une ligne  $L_k$  obtenu après la simulation du circuit sain. Chaque ligne représente le symbole stocké dans la liste de suspects  $L_F$  pour la même ligne  $L_k$ . Lorsque l'intersection est faite entre deux symboles représentant des transitions ou possibilités de transition, nous obtenons le nouveau symbole 'D'. Le symbole 'SDW' permet quant à lui de représenter une situation où le signal sur une ligne suspecte peut être stable et peut prendre des valeurs différentes pour différents vecteurs fautifs. Ceci se produit dans le cas de pannes de court-circuit de type « strong driver wired » où la valeur logique d'une ligne suspecte dépend directement de la valeur d'une autre ligne.

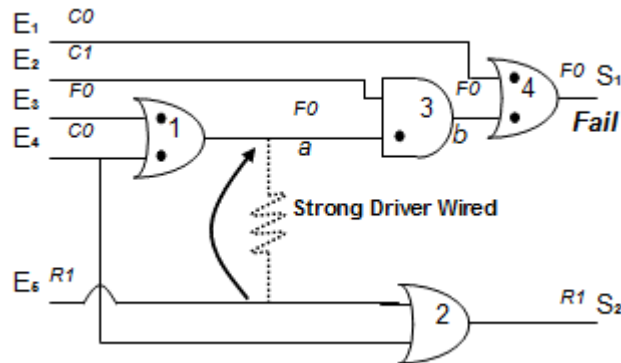
Notons que l'intersection du symbole D avec un symbole représentant une valeur stable donne le symbole SDW. En effet, le symbole D représente à l'origine l'intersection de symboles transitions à valeurs finales opposées, permettant de représenter des modèles de fautes provoquant des retards sur les fronts. Or si un symbole D associé à une ligne stockée dans la liste de suspects est intersectée avec un symbole représentant une valeur stable (C0 ou C1), cela signifie que le seul modèle de faute qui peut être associé à cette ligne est le Strong Driver Wired (seul modèle pouvant affecter une ligne et provoquer des erreurs quelles soient la nature du symbole et sa valeur finale).

Nous remarquerons également sur ce tableau d'intersection que, contrairement à celui vu précédemment, celui-ci ne permet plus de supprimer des lignes. Néanmoins, travaillant toujours dans l'hypothèse de la panne simple, le processus d'intersection des listes de suspects

peut être conservé et ainsi réduire le nombre de lignes suspectes à son minimum.

**Exemple :**

Considérons le circuit de la Figure 27 qui comporte cinq entrées ( $E_1$  à  $E_5$ ) et deux sorties ( $S_1$  et  $S_2$ ). Considérons deux paires de vecteurs  $V_1$  et  $V_2$  ayant produit des erreurs lors du test tels que  $V_1 = (C0,C1,F0,C0,R1)$  et  $V_2 = (C0,C1, C1, C0,C0)$ .



**Figure 27** : CPT à partir de la sortie  $S_1$

Lors du test, l'application du vecteur  $V_1=(C0,C1,F0,C0,R1)$  symbolisé sur la Figure 27 a propagé une erreur jusqu'à la sortie  $S_1$  qui a été déclarée fautive.

Le processus de traçage de chemins à partir de la sortie  $S_1$  donne la liste de suspects suivante :

- $L1 = \{(S1,F0),(b,F0),(E1,C0),(a,F0),(E3,F0),(E4,C0)\}$

Cette opération de traçage est répétée pour le vecteur  $V_2=(C0,C1, C1, C0,C0)$  qui a propagé une erreur sur la sortie  $S_1$  du circuit. La nouvelle liste de suspects obtenue est la suivante :

- $L2 = \{(S1,C1),(b,C1),(E2,C1),(a,C1),(E3, C1)\}$

L'intersection des deux listes de suspects en prenant en compte la nouvelle table d'intersection des symboles (Tableau 17) donne la liste finale suivante :

$$L12 = \{(S1,SDW),(b,SDW),(a,SDW),(E3,SDW)\}$$

### III.2.2.Modification du tableau d'allocation des modèles de fautes

L'ajout de ces trois nouveaux modèles de fautes dans le processus de diagnostic ainsi que la création des deux nouveaux symboles D et SDW entraînent par conséquent une modification du tableau d'allocation des modèles de fautes comme nous pouvons le voir sur le tableau III.8.

Le symbole 'D' représente la possibilité d'avoir une ligne qui est tracée pour des symboles représentant à la fois des transitions ou des possibilités de transitions montantes (R1 ou/et P1) et à la fois des transitions ou des possibilités de transitions descendantes (F0 ou/et P0). Or parmi l'ensemble des modèles de fautes considérés, seuls les modèles de court-circuit résistif, de délai de type L<sub>M</sub> & L<sub>D</sub> et de strong driver wired peuvent provoquer ce type d'erreur. Ce seront donc les seuls modèles associés avec le symbole 'D'.

Le symbole 'SDW' ne peut apparaître que durant l'opération de traçage de chemins critiques étant donné qu'il représente un modèle de court-circuit, et que nous avons vu auparavant que les conditions de sensibilisation de l'agresseur ne pouvaient être assurées que lors de l'application des vecteurs fautifs. Le symbole 'SDW' représente donc la possibilité d'avoir une ligne qui est tracée à la fois pour des symboles à valeur finale '0' (C0, F0, P0, C0+, F0+ ou P0+) et à la fois pour des symboles à valeur finale '1' (C1, F1, P1, C1+, F1+ ou P1+). Le seul modèle de faute ayant la particularité d'affecter une seule ligne en affectant les deux possibilités de valeurs logique sans tenir compte de la nature du symbole (transition, valeur constante, possibilité de transition) est le Strong Driver Wired. Ce sera donc le seul modèle de faute associé à ce symbole.

Ainsi, avec l'ajout de ces pannes et de ces symboles, la table d'allocation de pannes devient celle présentée sur leTableau 18.

Modèles de fautes	C0	C1	F0,P0	R1,P1	C0*	C1*	F0*,P0*	R1*,P1*	C0+	C1+	F0+,P0+	R1+,P1+	D	SDW
Collage à 0		x		x										
Collage à 1	x		x											
Tn Stuck-open	x		x											
Tn Stuck-on		x		x										
Tp Stuck-open		x		x										
Tp Stuck-on	x		x											
Circuit ouvert à 0		x		x										
Circuit ouvert à 1	x		x											
Circuit ouvert résistif			x	x			x	x			x	x	x	
Strong driver wired OR	x		x						x		x			
Strong driver wired AND		x		x						x		x		
Strong driver wired OR resistive			x				x				x			
Strong driver wired AND resistive				x				x				x		
Strong driver wired	x	x	x	x					x	x	x	x	x	x
Délai lent-à-descendre			x				x				x			
Délai lent-à-monter				x				x				x		
Délai de type LâM & LâD													x	

Tableau 18 : Mise à jour du tableau d’association avec les SDW, délai et court-circuit résistif

### III.3. Adaptation de la méthode de diagnostic pour prendre en compte les pannes de court-circuit pouvant avoir des effets sur les deux connexions

Deux types de court-circuit peuvent produire une erreur sur deux connexions en court-circuit.

Le premier qui comprend les pannes de type « Wired And/Or » et « Wired And/Or resistive » peut affecter les deux connexions mais pas simultanément. Ainsi, selon le vecteur de sensibilisation appliqué, c’est l’une ou l’autre des connexions en court-circuit qui est affectée par l’erreur. D’autre part, pour ce type de pannes, l’erreur a nécessairement la même polarité. Cette polarité dépend du fait qu’il s’agisse d’une panne de type « Wired And ou Resistive Wired And» ou d’une panne de type « Wired Or ou Resistive Wired Or ».

Un second type de court-circuit peut affecter deux connexions simultanément, c’est à dire pour le même vecteur. Dans ce cas, les valeurs logiques opposées portées par chaque ligne sont « échangées ». Le modèle de court-circuit pouvant provoquer ce type de panne est le modèle « byzantine ».

Dans la suite de ce chapitre, nous étudierons ces deux types de courts-circuits et adapterons le processus de diagnostic afin de les prendre en compte.

### **III.3.1.Pannes dont l'effet se traduit sur une seule connexion mais qui peut varier en fonction du vecteur de sensibilisation**

Le processus d'intersection tel qu'il a été proposé n'est valable que pour des pannes simples n'affectant qu'une connexion. Sans remettre en cause le processus de traçage de chemins critiques, le traitement des modèles de fautes de type « Wired And/Or » et « Wired And/Or resistive » dont l'effet se traduit sur une seule connexion mais qui peut varier en fonction du vecteur de sensibilisation implique de repenser ce processus.

Jusqu'à présent nous avons supposé que la ligne suspecte se trouvait forcément dans toutes les listes de suspects. Nous pouvions donc faire l'intersection de l'ensemble de ces listes de suspects (même si nous avons vu que l'implantation informatique de cette intersection pouvait se faire par un processus de comptage, cf. §II.2.3.c). Ce n'est plus le cas avec les pannes de type « Wired And/Or » et « Wired And/Or resistive ». En effet, dans ces cas, chacune des listes de suspects obtenue par traçage de chemins critiques contient bien la ligne sur laquelle s'est produite l'erreur initiale ayant provoquée l'erreur observée en sortie. Par contre, comme cette erreur peut affecter l'une ou l'autre des connexions en court-circuit, la même connexion ne se retrouve pas nécessairement dans toutes les listes de suspects. Il n'est donc plus possible d'effectuer l'intersection de celles-ci comme cela avait été fait précédemment. En fait, avec ce type de pannes, nous commençons à aborder les problèmes induits par les pannes multiples.

Une solution permettant de prendre en compte ce type de pannes, est de raisonner en s'appuyant sur le processus informatique développé pour effectuer l'intersection des listes (cf § II.2.3.b). Dans ce processus, l'intersection des listes n'était pas réalisée en tant que telle, mais deux paramètres étaient associés à chaque ligne lors du processus de traçage de chemin critique : un compteur (CNT) et un symbole (VAL). Le compteur CNT est incrémenté chaque fois qu'une ligne est rencontrée par le processus de traçage de chemins critiques et le symbole VAL est mis à jour en utilisant la table d'intersection des symboles (Tableau 17).

Précédemment, dans l'hypothèse de la panne simple, les lignes suspectes étaient celles dont la valeur du compteur était égale au nombre de fois que le processus de traçage de chemin avait été lancé (nombre de sortie produisant une erreur lors de l'application de la

séquence de test). Pour prendre en compte les pannes considérées ici, l'analyse devient plus complexe. En effet, chacune des lignes dont la valeur du compteur CNT est supérieure à 0 doit être considérée comme étant potentiellement une des sources possibles des erreurs observées en sortie du circuit. Néanmoins, l'information fournie par la valeur du compteur associé à chacune de ces lignes, peut donner une indication sur la probabilité que la panne vienne de telle ou telle connexion. En effet, il paraît plus probable que la panne provienne d'une ligne qui aurait été rencontrée autant de fois qu'il y a de sorties fautives plutôt que d'une ligne qui aurait été rencontrée une seule fois.

Ces nouveaux modèles de courts-circuits doivent maintenant être considérés dans le processus d'allocation des modèles de fautes. Ainsi, le tableau d'allocation devient celui présenté sur le Tableau 19.

Modèles de fautes	C0	C1	F0,P0	R1,P1	C0*	C1*	F0*,P0*	R1*,P1*	C0+	C1+	F0+,P0+	R1+,P1+	D	SDW
Collage à 0		*		*										
Collage à 1	*		*											
Tn Stuck-open	*		*											
Tn Stuck-on		*		*										
Tp Stuck-open		*		*										
Tp Stuck-on	*		*											
Circuit ouvert à 0		*		*										
Circuit ouvert à 1	*		*											
Circuit ouvert résistif													*	
Strong driver wired OR	*		*						*		*			
Strong driver wired AND		*		*						*		*		
Strong driver wired OR resistive			*			*					*			
Strong driver wired AND resistive				*			*					*		
wired AND		*		*					*		*		*	
wired OR	*		*					*	*	*	*			
Wired And resistive				*			*					*		
Wired OR resistive			*			*					*			
Strong driver wired	*	*	*	*					*	*	*	*	*	*
Dé la i lent-à- de scendre			*			*					*			
Dé la i lent-à- monter				*			*					*		
Dé la i de type LâM & LâD													*	

Tableau 19 : Mise à jour du tableau d'association avec les Wired And/Or et resistive

Les modèles de courts-circuits de type wired And et Or sont associés aux mêmes symboles que le strong driver wired And et Or respectivement. Cette observation est tout à fait justifiée puisque les modes de sensibilisation et de fonctionnement de ces deux types de courts-circuits sont similaires. La seule différence réside dans le fait que le wired And ou Or peut apparaître indifféremment sur l'une ou l'autre des deux lignes court-circuitées. Il est donc normal que ces deux types de courts-circuits soient associés aux mêmes symboles. Cette remarque s'applique également aux cas des wired And/Or resistive et des strong driver wired And/Or resistive.



### III.3.2.Pannes dont l'effet se traduit sur les deux connexions en court-circuit

Le modèle de faute de type byzantine est le plus complexe des courts-circuits. En effet, comme nous l'avons vu dans le chapitre I, cette faute produit une erreur sur chacune des connexions en court-circuit et ces deux erreurs sont de polarité opposée (0/1 sur une ligne et 1/0 sur l'autre). Notons que, comme indiqué dans le chapitre I, il n'est pas envisageable électriquement parlant, qu'une faute byzantine soit sensibilisée par les deux configurations de valeur sur les connexions mises en jeu (0/1 et 1/0). Dans cette hypothèse, lorsqu'elle est sensibilisée, une faute byzantine produit donc un effet qui peut être assimilé à un double collage (collage à 0 sur une des deux lignes, et collage à 1 sur l'autre). Dans cette hypothèse, et en considérant que le processus d'intersection est remplacé par un processus de comptage (cf § III.3.1), les modifications à apporter au processus de diagnostic pour prendre en compte ces pannes portent sur le processus de traçage de chemins critiques et bien évidemment également sur l'allocation des modèles de fautes.

#### III.3.2.a. Adaptation du processus de traçage de chemins critiques

Le processus de traçage de chemins critiques développé jusqu'à présent supposait un effet de faute simple, c'est-à-dire sur une seule connexion du circuit. La prise en compte des fautes byzantines entraîne quelques modifications lors du traçage de chemins critiques lui-même. En effet, jusqu'à présent, lorsqu'une porte *non sensible* était rencontrée, le processus de traçage de chemins critiques stoppait et un pied de divergence était recherché [Abra80]. Deux cas étaient alors possibles :

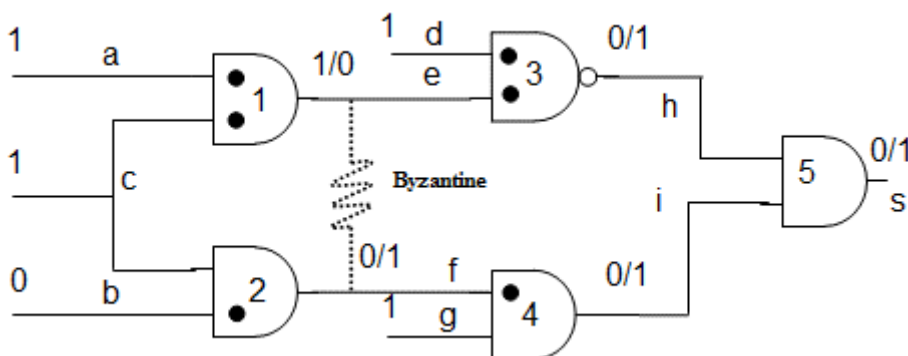
- Soit aucun pied de divergence n'était trouvé et alors le processus de traçage de chemins critiques s'arrêtait à la porte non sensible.
- Soit un pied de divergence était déterminé et alors le processus de traçage de chemins critiques reprenait à partir de ce pied de divergence. Les portes et lignes se trouvant entre cette porte non sensible et le pied de divergence étaient alors ignorées.

Dans le cadre d'un diagnostic adapté au cas de la byzantine, il n'est plus possible d'ignorer les portes logiques se trouvant dans une boucle de reconvergence. En effet, avec ce type de pannes, il est possible qu'au même moment, deux erreurs apparaissent dans une boucle de reconvergence et se propagent au-delà de la porte non sensible, puis

jusqu'aux sorties primaires du circuit. Ainsi, durant le processus de traçage de chemins critiques, si une porte de reconvergence non sensible est trouvée, les lignes en amont de celle-ci devront être considérées comme si elles étaient des lignes sensibles. Dans tous les autres cas, c'est à dire présence uniquement d'entrées sensibles ou présence d'entrées sensibles et d'autres non sensibles, le processus de traçage n'est pas modifié.

**Exemple :**

Considérons la Figure 28. Ce circuit a trois entrées (a,c,b) et une sortie (s). Le pied de divergence est la ligne *c*. La porte de reconvergence non sensible est la *porte 5*. Lors du test avec le vecteur (1,1,0) correspondant respectivement aux entrées (a,c,b), la sortie *s* a été déclarée fautive, la valeur logique '0' était attendue, et nous avons obtenu un '1'. Dans l'hypothèse de la panne simple et avec le processus de traçage de chemins utilisé jusqu'à présent, nous nous serions arrêtés à la porte 5 non sensible et aurions repris le processus de traçage de chemins critiques à partir du pied de divergence *c*, en ignorant les lignes et portes intermédiaires (*h, i, porte3, porte4, d, e, f, g, porte1, porte2*). Or nous pouvons voir dans cet exemple que la cause des erreurs est le modèle de faute byzantine. Ce modèle provoque une erreur simultanément sur deux lignes (*e* et *g*). Ces deux erreurs se propagent via les entrées sensibles des portes 3 et 4, atteignant la porte 5. Ces erreurs provoquent alors simultanément un changement de valeur sur les entrées *h* et *i* de la porte non sensible 5, changeant ainsi la valeur de sortie de celle-ci. Il est donc impératif, maintenant avec la prise en compte des possibilités de multi erreurs, de considérer les chemins internes aux boucles de reconvergence.



**Figure 28** : Exemple de portes non sensibles

III.3.2.b.Mise à jour du tableau d'allocation

Comme précédemment, il est maintenant nécessaire de mettre à jour la table d'allocation des modèles de fautes en rajoutant le modèle de faute byzantine (Tableau 20). Nous pouvons remarquer que ce modèle de court-circuit est associé à la plupart des symboles à l'exception de ceux ayant un marqueur '+'. En effet, comme pour les autres courts-circuits, si une erreur sur une ligne suspecte produit (en simulation de fautes), pour un vecteur fautif, une erreur sur une sortie déclarée saine, alors le court-circuit de type byzantine peut être disculpé et donc retiré de la liste des modèles de fautes associés à cette ligne.

De plus, la byzantine n'est pas associée aux symboles D et SDW car ces symboles sont associés à des connexions qui peuvent avoir les deux possibilités de changement de polarité (1/0 et 0/1). Or, le court-circuit de type byzantine tel qu'il est considéré affecte deux connexions en même temps, mais ne peut affecter qu'une seule des deux polarité sur chaque ligne. Admettons qu'une ligne *a* ait pour valeur logique d'origine '1' et une ligne *b* la valeur logique '0', alors avec le court-circuit de type byzantine, la ligne *a* se retrouvera avec la valeur '0' et la ligne *b* la valeur '1'. Mais l'inverse n'est pas possible, si *a* = '0' et *b* = '1' à l'origine, il n'y aura pas de modification de leurs valeurs logiques. La byzantine ne « marche » que dans un seul sens.

Le Tableau 20 représente la table finale d'allocation des modèles de fautes de cette étude. Tous les modèles de fautes considérés et décrits dans le chapitre I s'y retrouvent.

Modèles de fautes	C0	C1	F0,P0	R1,P1	C0*	C1*	F0*,P0*	R1*,P1*	C0+	C1+	F0+,P0+	R1+,P1+	D	SDW
Collage à 0		x		x										
Collage à 1	x		x											
Tn Stuck-open	x		x											
Tn Stuck-on		x		x										
Tp Stuck-open		x		x										
Tp Stuck-on	x		x											
Circuit ouvert à 0		x		x										
Circuit ouvert à 1	x		x											
Circuit ouvert résistif													x	
Strong driver wired OR	x		x						x		x			
Strong driver wired AND		x		x						x		x		
Strong driver wired OR resistive			x				x				x			
Strong driver wired AND resistive				x				x					x	
wired AND		x		x						x		x		
wired OR	x		x						x		x			
Wired And resistive				x				x					x	
Wired OR resistive			x								x			
Strong driver wired	x	x	x	x					x	x	x	x	x	x
Byzantine	x	x	x	x					x	x	x	x		
Délai lent-à-descendre			x				x				x			
Délai lent-à-monter				x				x					x	
Délai de type LâM & LâD														x

Tableau 20 : Tableau d'association des modèles de fautes complet

## Conclusion

L'ensemble de ce chapitre a présenté les adaptations apportées à la méthode de diagnostic initiale, d'une part pour améliorer la précision du diagnostic et d'autre part, pour permettre la prise en compte de l'ensemble des modèles de fautes considérés au chapitre I.

La première partie de ce chapitre était plus particulièrement consacrée aux adaptations permettant d'améliorer la précision du diagnostic. Ces améliorations sont apportées par la prise en compte de l'historique des valeurs logiques ayant transité dans le circuit (variable « flag » introduite dans le chapitre II). Ces adaptations permettent de cibler un ensemble de lignes agresseur dans le cas des courts-circuits simples (SDW And/Or et SDW And/Or resistive) mais également de disculper des transistors dans le cas des pannes de stuck-on et stuck-open. Ces précisions pourront être un atout important lors de l'analyse de défaillance.

Les deuxième et troisième parties ont permis de mettre en évidence les modifications à apporter à la méthode de diagnostic initiale pour permettre la prise en compte d'une part, des modèles de fautes pouvant engendrer sur une même connexion des erreurs de polarités différentes et, d'autre part, les modèles de courts-circuits pouvant engendrer des erreurs sur les deux lignes en court-circuit. Ces modifications telles qu'elles ont été développées conduisent, à terme, à obtenir un processus de diagnostic complet applicable quelle que soit la faute qui affecte le circuit.

Même si pour certaines, elles peuvent induire des effets multiples, les pannes considérées dans cette étude sont des pannes simples. Les adaptations mises en place pour traiter les pannes à effets multiples laissent à penser que cette méthode de diagnostic, associée à l'utilisation d'outils statistiques, serait parfaitement utilisable ou tout au moins adaptable au cas des pannes multiples.



# **IV - Expérimentations**



## Introduction

Dans ce dernier chapitre, nous présentons les expérimentations que nous avons réalisées afin de valider chacune des étapes de notre méthode de diagnostic. Nous présenterons dans un premier temps l'environnement de travail sur lequel reposent ces expérimentations.

Nous détaillerons par la suite les résultats obtenus en considérant uniquement les modèles de fautes ne provoquant qu'une erreur simple sur un seul site dans le circuit sous test (chapitre II). En la comparant à un outil de référence industriel, nous montrerons ainsi l'intérêt de notre méthode de diagnostic en terme de précision.

Nous présenterons également les résultats obtenus en considérant les améliorations apportées par la prise en compte de l'historique des valeurs s'étant propagées dans le circuit (flag). Ainsi, nous pourrions voir que cette amélioration dans le processus de diagnostic permettra de cibler un ensemble limité de transistors pouvant être affectés par une panne de stuck-on ou de stuck-open ou un ensemble limité d'agresseurs dans le cas des courts-circuits simples tels que le strong driver wired or/and et le strong driver wired or/and resistive.

Enfin, nous présenterons les résultats obtenus lorsque les pannes induisant des erreurs de polarités multiples ou des erreurs sur plusieurs sites sont considérées.





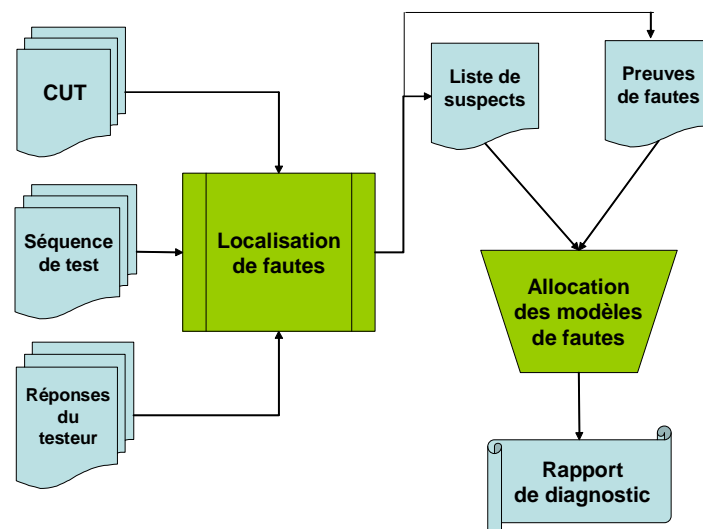
## IV.1. Environnement expérimental

La méthode de diagnostic développée a été implantée avec à peu près 5000 lignes de code C++. La complexité de cette approche dépend de trois facteurs:

- Le nombre de portes :  $n$
- Le nombre de vecteurs fautifs: #FP
- Le nombre de sorties primaires fautives: #FPO

Cette complexité, qui peut s'exprimer de la manière suivante :  $O((\#FP \times n) + (\#FPO \times n)) = O((\#FP + \#FPO) \times n)$  est linéaire en terme de nombre de portes.

Les différentes phases et les entrées/sorties de ce processus de diagnostic sont rappelées sur la Figure 29. La phase de « localisation » a pour entrées la description structurale du circuit (CUT), la séquence de vecteurs de test et les réponses du circuit à ces vecteurs. Elle fournit une liste de suspects (lignes suspectées d'être à l'origine de l'erreur observée) ainsi que des informations spécifiques concernant les valeurs propagées dans le circuit. La liste de suspects et ces informations sont exploitées lors de la phase « d'affectation des modèles » qui produit le rapport de diagnostic final.



**Figure 29** : Synoptique du processus de diagnostic développé.

Les expérimentations ont été effectuées sur des stations SUN.

Les circuits considérés sont ceux des familles ISCAS'85, ISCAS'89 et ITC'99.

Les séquences de vecteurs de test appliquées au circuit ont été générées en utilisant un outil commercial de génération de vecteurs de test automatique (ATPG de Tetramax, ©Synopsys).

Pour simuler le comportement des circuits fautifs, c'est-à-dire obtenir les vecteurs fautifs ainsi que les sorties fautives correspondantes, nous avons injecté des fautes dans la description structurelle du circuit et simulé celui-ci avec un simulateur logique (Tetramax, ©Synopsys). Les réponses de sorties comparées à celles obtenues avec la simulation du circuit sain fournissent l'équivalent de ce qu'aurait fourni un testeur.

## IV.2. Validation de la méthode de diagnostic développée

Cette partie a pour objectif de présenter les expérimentations réalisées pour valider la méthode proposée et de montrer l'intérêt d'une approche permettant de considérer un ensemble élargi de modèles de fautes

Le premier jeu d'expérimentations présenté est relativement détaillé. Il a été réalisé sur un circuit de taille modeste : le circuit C432 de la famille des ISCAS'85. Des résultats synthétiques obtenus sur un jeu de circuits de taille plus conséquente seront présentés dans un deuxième temps.

### IV.2.1. Expérimentations réalisées sur le circuit C432 (ISCASS'85)

Le circuit C432 de la famille ISCAS'85 comporte 36 entrées et 7 sorties. La séquence de test appliquée à ce circuit est celle générée par l'ATPG (Tetramax, ©Synopsys) pour des pannes de collage. Cette séquence comporte 76 vecteurs.

Le Tableau 21 résume les expérimentations réalisées sur ce circuit. Chacune des lignes du tableau correspond à une opération de diagnostic. Le « scénario » précisé dans la première colonne, indique la ligne fautive et le modèle de fautes injecté dans le circuit. Les modèles de fautes utilisés ici sont ceux exploitables par le simulateur qui nous a servi à simuler le comportement du testeur, à savoir : les pannes de collage à 0 et à 1, les pannes de

délai de type LàM ou LàD et les pannes de court-circuit de type strong driver wired And et strong driver wired Or. Dans le cas des fautes de court-circuit, la notation L1 => L2 indique que la faute est injectée entre L1 et L2, et que L1 (agresseur) *attaque* L2 (victime). La colonne 2 donne le nombre de vecteurs fautifs (#FP) obtenu par simulation et la colonne 3 donne le nombre de sorties fautives (#FPO). La colonne 4 fournit le nombre de lignes suspectes identifiées par un outil de diagnostic commercial qui nous a servi de référence (Ref Tool). Cet outil de référence, basé sur une approche « Cause à Effet », c'est-à-dire sur de la simulation de fautes, ne considère à priori que les pannes de collages. Enfin la dernière colonne indique le nombre de lignes identifiées par notre méthode de diagnostic (NOD, Notre Outil de Diagnostic).

Scenario	#FP	#FPO	Ref Tool	NOD
L164 – Cà0	19	32	3	4
L96 – Cà1	2	5	1	9
0_3 – Cà0	39	39	2	4
L94 => L123 strong driver wired Or	23	76	1	1
L140 => L163 strong driver wired Or	30	62	1	3
<b>L_2 =&gt; L100 strong driver wired AND</b>	<b>3</b>	<b>7</b>	<b>1 error</b>	<b>21</b>
L52 => L84 strong driver wired AND	22	61	2	2
Gate_90 - LàM	3	4	8	8
Gate_140 - LàD	7	16	5	5
<b>Gate_46 – LàD</b>	<b>15</b>	<b>62</b>	<b>Ø</b>	<b>1</b>
<b>Gate_72 - LàD</b>	<b>4</b>	<b>8</b>	<b>Ø</b>	<b>7</b>

**Tableau 21** : Résultats expérimentaux obtenus sur le C432 (ISCAS'85)

Avant de commenter ce tableau, notons que nous avons bien évidemment vérifié qu'avec la méthode proposée, la ligne sur laquelle nous avons injecté la panne se retrouvait bien dans la liste des lignes suspectes.

Comparons maintenant les résultats de diagnostic fournis par la méthode proposée avec ceux obtenus en utilisant l'outil qui nous a servi de référence. Un premier commentaire sur les résultats obtenus est que notre approche fournit généralement plus de lignes suspectes que l'outil de référence. Cette différence est expliquée par le fait que l'outil de référence considère seulement les modèles de fautes de collage durant son analyse. Il ne fournit donc que des candidats (lignes suspectes) pouvant être liées à des fautes de collage. Comme notre approche considère un ensemble plus complet de modèles de fautes, elle ne peut éliminer certaines lignes suspectes qui pourraient être à l'origine des erreurs observées. C'est par exemple le cas des lignes qui sont éliminées par l'outil de référence quand il existe un chemin

sensible vers une sortie déclarée bonne. Dans notre cas, ces lignes ne peuvent être systématiquement éliminées notamment par exemple pour la prise en compte de pannes de délai, qui en fonction du retard introduit par la panne ne produisent pas nécessairement une erreur sur chaque sortie sensibilisée.

Pour mieux expliquer cette situation, le Tableau 22 montre le résultat de diagnostic obtenu avec le scénario reporté sur la seconde ligne du Tableau 21 (L96 – Ca1). Dans ce cas, l'outil de référence reporte seulement un suspect alors que notre approche en donne neuf (Tableau 21). Notons que la ligne L96 à laquelle est associée une liste de fautes qui comprend la faute injectée (Ca1) est la seule ligne comprenant un collage. Le cas contraire aurait été incohérent avec le résultat obtenu par l'outil de référence. D'autre part, nous remarquerons qu'à toutes les autres lignes suspectes est associé un symbole avec le marqueur « + ». Ceci indique que la réponse obtenue en simulation de fautes pour des pannes de collages sur ces lignes (procédure de prise en compte des sorties saines) n'est pas identique à celle obtenue sur le testeur. Dans ce cas, des vecteurs ne produisant pas d'erreur sur le testeur en produisent en simulation. Cet état de fait provient du fait que la panne de collage est systématiquement sensibilisée et observée s'il existe un chemin sensible alors que ce n'est pas le cas pour d'autres et notamment celles mentionnées dans le tableau. En effet, un court-circuit de type Strong wired and n'est par exemple pas sensibilisé si la connexion « agresseur » n'est pas à la valeur 0. Dans ce cas elle ne produit pas d'erreur alors que le collage sur la même connexion en produit. De la même manière, une panne de délai sur une telle connexion n'est pas observée en sortie si le délai introduit par la panne n'est pas suffisant.

LC	CNT	S	Modèles de fautes
L96	5	F0	Cà1, Tn stuck-open, Tp stuck-on, circuit ouvert à 1, strong driver wired Or, strong driver wired or resistive, Byzantine, strong driver wired, wired or, court circuit résistif, délai de type LàD
I_10	5	C0+	Strong driver wired or, Byzantine, strong driver wired, wired or
I_14	5	C0+	Strong driver wired or, Byzantine, strong driver wired, wired or
I_26	5	F0+	Strong driver wired or, Byzantine, strong driver wired, wired or, circuit ouvert résistif, délai de type LàD
I_28	5	C1+	Strong driver wired and, Byzantine, strong driver wired, wired and
L31	5	C0+	Strong driver wired or, Byzantine, strong driver wired, wired or
L49	5	C1+	Strong driver wired and, Byzantine, strong driver wired, wired and
L83	5	F0+	Strong driver wired or, Byzantine, strong driver wired, wired or, circuit ouvert résistif, délai de type LàD
L84	5	R1+	Strong driver wired and, Byzantine, strong driver wired, wired and, circuit ouvert résistif, délai de type LàM

**Tableau 22** : Résultat de diagnostic

Un second commentaire sur les résultats présentés sur le Tableau 21 concerne le sixième scénario (injection d'une faute de Strong Driver Wired And I\_2=>L100). L'outil de référence a trouvé une ligne suspecte, ce qui en soi pourrait être un bon résultat, mais cette ligne n'est pas la ligne où la faute a été injectée.

Un autre problème apparaît pour les deux dernières fautes injectées, c'est-à-dire Gate\_46 LàD et Gate\_72 LàD (deux dernières lignes du Tableau 21). Dans ces deux cas, l'outil de référence échoue pour trouver une ligne suspecte alors que notre approche fournit une liste contenant bien le défaut injecté. En fait, comme les fautes injectées dans ces cas sont des fautes de délai, l'outil de référence est incapable de fournir un rapport de diagnostic correct. Ces expérimentations sur le circuit C432 montrent que l'outil de référence n'est utilisable que lorsque les défauts peuvent être modélisés par des collages. Lorsque les défauts qui affectent le circuit ne se comportent pas comme des collages, il arrive régulièrement que l'outil de référence ne puisse pas donner de résultats ou alors que les résultats donnés soient erronés. Par contre, la méthodologie de diagnostic que nous proposons fournit toujours un résultat de diagnostic cohérent au détriment d'un ensemble de suspects plus important.

### IV.2.2. Expérimentations réalisées sur un jeu de plus gros circuits

Les expérimentations présentées dans cette partie visent à montrer l'applicabilité de la méthode de diagnostic proposée sur un jeu de circuits de tailles plus conséquentes. Ces expérimentations ont été réalisées sur les circuits b17, b20, b21 et b22 de la famille ITC'99. Sur ces circuits, 100 scénarios (pannes injectées) différents ont été appliqués. Les résultats présentés sont une moyenne des résultats obtenus sur ces 100 scénarios

Le Tableau 23 présente ces résultats. La colonne 1 indique le nom du circuit, la colonne 2 son nombre d'entrées, la colonne 3 son nombre de sorties et la colonne 4 son nombre de portes. La colonne 5 donne la longueur de la séquence de test appliquée au circuit (#VT). Cette séquence est générée par ATPG (Tetramax, ©Synopsys). Les colonnes 6 et 7 présentent respectivement la moyenne du nombre de vecteurs fautifs (#FP) et la moyenne du nombre de sorties fautives (#FPO). La moyenne du nombre de lignes suspectes identifiées par notre méthode est donné dans la colonne 8 (#suspects).

Comme nous pouvons le voir le nombre de lignes suspectes obtenues est généralement assez faible (33 dans le pire cas). Ceci dit, ce paramètre n'est pas nécessairement un bon indicateur de la qualité de la méthode de diagnostic puisqu'il dépend d'autres données et notamment de la séquence de test appliquée au circuit. Par ailleurs, nous avons bien évidemment vérifié que la faute injectée se retrouvait bien dans la liste des suspects obtenue.

Circuit	#Entrées	#Sorties	#Portes	#VT	#FP	#FPO	#Suspects
b17	37	97	32326	1931	360	463	28.5
b20	32	22	20226	1246	165	245	33.2
b21	32	22	20571	1162	215	326	20.5
b22	32	22	29951	1699	199	307	28.4

**Tableau 23** : Résultats obtenus sur des circuits benchmark ITC'99

### IV.3. Validation des adaptations de la méthode de diagnostic

Dans cette partie, nous présenterons les expérimentations réalisées pour valider les adaptations de la méthode de diagnostic décrites au chapitre III.

Nous présenterons dans un premier temps un jeu d'expérimentations qui nous permet d'apprécier l'amélioration de la précision du diagnostic amenée par la prise en compte des valeurs s'étant propagées dans le circuit. Nous présenterons dans un deuxième temps des expérimentations réalisées avec des pannes induisant des erreurs de polarités multiples ou plusieurs erreurs.

#### **IV.3.1. Expérimentations permettant d'apprécier l'amélioration de la précision du diagnostic**

Les expérimentations présentées ici ont été réalisées sur deux des circuits de la famille des ITC'99 (b21, b22). Les caractéristiques de ces deux circuits ainsi que le nombre de vecteurs des séquences de test qui leur ont été appliqués (générées par ATPG) sont présentés sur le Tableau 23.

Une première expérimentation réalisée sur le circuit b21 est présentée ici. Dans ce circuit une panne aléatoire a été introduite et le processus de diagnostic a été lancé avec et sans prise en compte des valeurs s'étant propagées dans le circuit (flag). Le Tableau 24 présente les résultats obtenus dans les deux cas. La colonne 1 donne les lignes suspectes et la colonne 2 les symboles associés à ces lignes. Les trois colonnes suivantes précisent les paramètres associés à la porte amont aux lignes suspectes à savoir : le type de porte (colonne 3), les entrées de cette porte (colonne 4) et les « flags » associés à ces entrées (colonne 5). Les colonnes suivantes présentent le résultat du diagnostic lorsque les valeurs s'étant propagées dans le circuit (paramètre flag) ne sont pas prises en compte (colonne 6) et lorsqu'elles le sont (colonne 7).

Nous pouvons ainsi remarquer en analysant la ligne P1-U6581, que dans les deux cas celle-ci est suspectée d'être affectée par un court-circuit de type strong driver wired and, wired And, strong driver wired, byzantine, un collage à 0 ou un circuit ouvert à 0. Par contre, en considérant les valeurs propagées dans le circuit nous pouvons observer que les pannes de stuck-on et de stuck-open n'apparaissent plus.

L'amélioration de la précision du diagnostic par la prise en compte des valeurs s'étant propagées dans le circuit (flag) concernent les pannes de stuck-on et stuck-open mais également les pannes de court-circuit de type strong driver wired or/and, strong driver wired or/and resistive en limitant le nombre d'agresseurs potentiels. Ces résultats ne sont pas



explicites dans ce tableau.

lignes suspectes	Symbole	porte précédente	entrées	flag	modèles de fautes sans considérer le flag	modèles de fautes en considérant le flag
P1_U6079	D	AND	P1_U6458	10	Délai LàM& LàD, circuit ouvert résistif, strong driver wired	Délai LàM& LàD, circuit ouvert résistif, strong driver wired
			P1_U7996	10		
P1_U6581	C1	NAND	P1_U8631	11	Collage à 0, Tp stuck-open, Tn stuck-on, circuit ouvert à 0, strong driver wired and, wired and, strong driver wired, byzantine	Collage à 0, circuit ouvert à 0, strong driver wired and, wired and, strong driver wired, byzantine
			P1_U8630	10		
P1_R414_U255	C0	NOT	P1_U6581	10	Collage à 1, circuit ouvert à 1, Tn Stuck open, Tp Stuck on, strong driver wired or, wired or, strong driver wired, byzantine	Collage à 1, circuit ouvert à 1, Tn Stuck open, Tp Stuck on, strong driver wired or, wired or, strong driver wired, byzantine
P2_U5985	D	NAND	F2_U7374	01	Délai LàM& LàD, circuit ouvert résistif, strong driver wired	Délai LàM& LàD, circuit ouvert résistif, strong driver wired
			F2_U7373	10		
			F2_U7375	10		
			F2_U7377	10		

**Tableau 24** : Résultats du diagnostic

Cette expérimentation a été reconduite sur les circuits b21 et b22 (ITC'99) pour plusieurs scénarios, c'est-à-dire plusieurs sites et types de pannes injectées. Le Tableau 25 et le Tableau 26 présentent les résultats obtenus.

La colonne 1 donne le site de la faute, c'est-à-dire la ligne sur laquelle la faute a été injectée, ainsi que le modèle de fautes injectées sur celle-ci. Les colonnes 2 et 3 donnent respectivement le nombre de vecteurs fautifs (#Fp) et le nombre de sorties primaires fautives (#Fo). La colonne 4 (#suspects) donne le nombre de lignes suspectes identifiées. Les colonnes suivantes montrent comment la prise en compte des valeurs propagées dans le circuit (flag) permet d'améliorer l'allocation des modèles de fautes. Ainsi, la colonne 5 (SO) donne le nombre de lignes où les modèles de stuck-on/open sont éliminés par la prise en compte des valeurs propagées dans le circuit. La colonne 6 (Agg) donne le nombre total de lignes potentiellement agresseurs des courts-circuits suspectés. La colonne 7 (%LA) indique le pourcentage d'agresseurs possibles en fonction du nombre total de lignes dans le circuit (nombres d'agresseurs possibles / nombre de lignes total \*100). Enfin la dernière colonne (%DR) montre la résolution du diagnostic exprimée en pourcentage (nombre de lignes suspectes / nombre de lignes total).

Site de faute (b21)	# Fp	#Fo	#suspects	SO	Agg	% LA	% DR
P1_DATAO_REG_23 - AND - PPO_204	3	3	13	8	1654	15,37	0,121
P1_IR_REG_12_ - Sa1	432	1742	2	0	-	-	0,019
P1_R199_U859 - Sa1	40	40	5	3	-	-	0,046
P1_U6224 - OR - P1_ADDR_REG_3_	319	680	2	0	189	1,756	0,019
P1_U6247 - OR - P1_REG3_REG_21_	286	290	2	0	350	3,252	0,019
P1_U6252 - OR - P1_REG3_REG_3_	267	950	2	0	274	2,546	0,019
P1_U6329 - Sa1	121	165	4	3	-	-	0,037
P1_U6403 - AND - P1_U8135	375	822	3	2	17	0,157	0,028
P1_U6817 - OR - P1_U6677	37	339	2	1	1215	11,29	0,019
P2_R370_U289 - AND - P2_R370_U222	78	78	11	7	63	0,585	0,102
P2_U5907_U755 - OR - P2_U5907_U754	55	55	4	2	849	7,889	0,037
P2_U5986 - AND - P2_U7897	41	41	11	5	111	1,031	0,102
P2_U6495 - OR - P2_U8552	9	9	15	1	2408	22,377	0,139
P2_U6604 - AND - P2_U7249	7	7	10	6	352	3,271	0,093
P2_U7062 - Sa0	762	1106	1	1	-	-	0,009
U192 - Sa0	234	370	3	2	-	-	0,028
P1_U6010 - Delay	4	4	33	3	-	-	0,307
P1_U6079 - Delay	27	27	1	0	-	-	0,009
P2_U5907_U839 - Delay	7	7	17	5	-	-	0,158
P2_U7871 - Delay	13	13	30	0	-	-	0,279
U65 - Delay	143	143	1	1	-	-	0,009

**Tableau 25** : Expérimentations menées sur le b21

Un premier commentaire est que nous avons dans chaque cas bien évidemment vérifié que la ligne sur laquelle nous avons injecté la faute se retrouvait bien dans la liste de suspects et qu'il en était de même pour le modèle de la faute injectée.

A partir de ces résultats, nous pouvons voir que la résolution du diagnostic exprimée en pourcentage (%DR) est très bonne. En effet, la résolution de diagnostic moyenne est d'environ 0.076%. Etant entendu que cette résolution ne dépend pas uniquement de la méthode de diagnostic, mais également d'autres paramètres tels que notamment la séquence de test appliquée nous pouvons néanmoins déduire de ces résultats que la localisation de la panne est en règle générale relativement précise (#suspects).

Ces expérimentations montrent également l'intérêt de prendre en compte les valeurs propagées dans le circuit (flag). Pour le modèle de fautes de stuck-on/open, ceci permet effectivement de supprimer en moyenne 30% des sites de pannes (transistors) potentiels. Dans le cas des fautes de court-circuit de type « strong driver wired and/or et resistive strong driver wired and/or », les résultats montrent qu'en considérant les valeurs propagées dans le circuit

(flag) un nombre réduit d’agresseurs potentiels peut être identifié (Agg). Nous remarquerons cependant une grosse différence dans les pourcentages (%LA) obtenus dans le cas des courts-circuits de type ET et dans le cas des courts-circuits de type OU. En présence d’un court-circuit de type ET, le nombre d’agresseurs associés est bien plus petit. Le pourcentage (colonne 7) est effectivement d’environ 1%. En présence d’un court-circuit de type OU, ce pourcentage monte jusqu’à 24% du nombre total de lignes du circuit sous test. Cela peut s’expliquer par la structure des circuits considérés. En effet, les deux circuits b21 et b22 étant composés de 60% de portes NAND, la probabilité d’avoir ‘1’ sur les connexion est plus importante que celle d’avoir 0. Ceci augmente le nombre d’agresseurs possible pour les courts-circuits de type OU.

Site de faute (b22)	# Fp	#Fo	#suspects	SO	Agg	% LA	% DR
P1_R746_U137 - Sa1	33	33	1	1	-	-	0,006
P1_U3342 - AND - P1_U4216	99	2224	12	4	102	0,62	0,073
P1_U3593 - Sa1	800	800	4	2	-	-	0,024
P1_U4269 - OR - P1_U4268	8	8	21	2	3890	23,55	0,127
P1_U4489 - Sa0	106	106	5	1	-	-	0,030
P1_U5313 - Sa1	23	44	3	0	-	-	0,018
P2_DATAO_REG_17_ - OR - PPO_443	15	15	6	0	2902	17,57	0,036
P2_U3087 - OR - P2_U5117	88	88	5	2	-	-	0,030
P2_U4600 - Sa1	25	25	7	0	-	-	0,042
P2_U5888 - OR - P2_U5889	33	33	5	2	2683	16,24	0,030
P2_U5907_U138 - Sa1	54	101	4	3	-	-	0,024
P2_U5907_U376 - AND - P2_U5907_U378	21	45	1	0	185	1,12	0,006
P3_R337_U415 - AND - P3_R337_U418	19	19	5	3	252	1,53	0,030
P3_U4817 - Sa1	35	25	4	1	-	-	0,024
P3_U5125 - OR - P3_U5124	32	65	8	0	1834	11,10	0,048
SUB_1605_U330 - OR - SUB_1605_U97	39	41	1	1	1583	9,58	0,006
U109 - Sa0	457	522	1	1	-	-	0,006
P1_U4029 - Delay	78	78	2	1	-	-	0,012
P2_R433_U105 - Delay	2	2	28	0	-	-	0,169
P2_U3218 - Delay	708	708	1	1	-	-	0,006
P3_U5612 - Delay	2	2	30	2	-	-	0,182

Tableau 26 : Expérimentations menées sur le b22

### IV.3.2. Expérimentations réalisées avec des pannes induisant des erreurs de polarités multiples ou plusieurs erreurs.

Finalement, nous avons testé la capacité de notre outil à traiter les pannes induisant

des erreurs de polarités multiples ou plusieurs erreurs. Pour cela, nous avons injecté des modèles de fautes pouvant entraîner une erreur toujours de même polarité sur une seule ligne (strong driver wired and/or), une erreur sur une seule ligne mais pouvant avoir des polarités différentes (strong driver wired), ou deux erreurs sur deux lignes différentes (wired and/or, byzantine).

Le Tableau 27 présente le résumé des expérimentations menées sur des circuits de référence de la famille des ISCAS'89 et ITC'99. La colonne 1 précise le nom du circuit. La colonne 2 donne le scénario, c'est-à-dire le modèle de faute injecté. Pour chaque modèle de fautes, nous effectuons 100 injections sur différents sites choisis aléatoirement. Les colonnes 3 et 4 donnent respectivement le nombre moyen de vecteurs fautifs (#FP) et le nombre moyen de sorties fautives (#FPO). La colonne 5 (#lignes critiques) donne le nombre moyen de lignes suspectes rencontrées au moins une fois pendant le processus de traçage de chemins (CNT > 0). La colonne 6 (#suspects) fournit le nombre moyen de lignes suspectes ayant été rencontrées lors de chaque processus de traçage de chemins **et** donc ayant un CNT égal au nombre de sorties fautives. Ces lignes sont les seules que nous aurions à considérer si nous nous situons dans l'hypothèse de la panne simple (panne engendrant une erreur toujours de même polarité). L'avant dernière colonne (Ligne) donne la position moyenne de la ligne où nous avons injecté la faute dans la liste de suspects ordonnancée par rapport à la valeur du paramètre CNT associé. Enfin la dernière colonne donne la résolution de diagnostic obtenue (%DR). Cette résolution correspond au (nombre de lignes critiques / nombre de lignes totales) \* 100.

Encore une fois, nous avons vérifié que la faute injectée se retrouvait bien dans la liste de fautes suspectes.

Circuit	Scenario	#FP	#FPO	#lignes critiques	#suspects	Ligne	%DR
s38584	SDW	590	685	283	7	4	13.94
	W-AND	285	340	235	13	7	13.12
	W-OR	315	387	172	7	4	11.57
	Byzantine	550	641	294	6	5	14.18
	Dom AND	95	110	312	9	7	15.07
	Dom OR	398	410	215	7	4	12.98
b20	SDW	380	464	1261	20	12	12.15
	W-AND	350	339	1325	7	8	12.77
	W-OR	478	559	1358	23	14	13.09
	Byzantine	315	641	294	6	5	2.83
	Dom AND	274	355	513	12	8	4.94
	Dom OR	240	295	540	10	5	5.20
b21	SDW	952	396	1224	16	11	11.37
	W-AND	110	249	1496	10	10	13.90
	W-OR	271	542	1660	12	9	15.43
	Byzantine	752	1441	2512	2	7	23.34
	Dom AND	745	821	1312	8	3	12.19
	Dom OR	319	680	1423	15	2	13.22
b22	SDW	411	560	1050	29	19	6.36
	W-AND	277	435	1304	7	8	7.89
	W-OR	396	483	1318	8	12	7.98
	Byzantine	741	1063	2239	3	7	13.55
	Dom AND	479	744	939	18	12	5.8
	Dom OR	741	764	1251	12	8	7.57

**Tableau 27** : ISCAS'89 and ITC'99 results summary

Un commentaire concerne le nombre de suspects fournis par notre outil (# lignes critiques). Dans ce cas (prise en compte de pannes multiples) toutes les lignes tracées au moins une fois durant le processus de traçage de chemins (CNT>0) sont comptabilisées. Même dans ce cas, nous pouvons voir que la résolution du diagnostic reste exploitable puisque moins d'un quart des lignes totales du circuit sont conservées.

Un second commentaire concerne la position de la ligne sur laquelle a été injectée la faute dans la liste de suspects ordonnancée (suivant le CNT). Nous pouvons voir dans le Tableau 27 que la ligne (ou au moins une des deux lignes dans le cas des courts-circuits affectant deux connexions) est toujours classée parmi celles ayant un des plus grand CNT (en moyenne entre la position 2 et 19 comme montré dans la colonne 7). Dans le cas des courts-circuits de type strong driver wired and/or et strong driver wired, l'effet de la faute affecte seulement une ligne. Nous nous attendons par conséquent à ce que la ligne ayant le plus haut score dans la liste ordonnée de suspects soit toujours la victime. Comme prévu, la ligne victime a toujours un CNT égal au nombre de sorties primaires fautives et donc obtient le plus

haut rang dans la liste de suspects. Dans le cas des modèles provoquant plusieurs erreurs dans le circuit tels que les byzantines ou les wired and/or, au moins une des deux lignes est toujours dans le début de la liste des suspects.

Dans le cas des fautes de court-circuit, la résolution du diagnostic peut être améliorée en prenant en compte des informations telles que la description physique du circuit sous test. Ces informations liées aux résultats du diagnostic (victime et agresseur) peuvent permettre de discriminer certaines possibilités de lignes suspectes.



## Conclusion

Les expérimentations menées ont permis de valider toutes les étapes de développement de notre méthode de diagnostic. Dans un premier temps, nous avons pu constater l'utilité de développer une méthode de diagnostic basée sur un ensemble complet de modèles de fautes, car là où un outil industriel de référence a échoué, notre méthode contenait toujours le bon site avec le bon modèle de faute associé (vu pour les fautes de délai et de court-circuit). Un autre outil industriel aurait sûrement permis de trouver le bon site, mais en ne connaissant à priori pas le modèle de faute qui représentait au mieux la défaillance, il aurait fallu répéter cette opération de diagnostic autant de fois que le nombre de modèles de fautes que nous aurions voulu considérer.

Dans un second temps, nous avons validé la méthode telle qu'elle a été développée dans le chapitre II sur de plus gros circuits de référence de la famille ITC et ISCAS. Nous avons pu voir que, malgré l'ensemble complet de modèles de fautes considérés, le nombre de lignes suspectes obtenues restait exploitable. Surtout que généralement la ligne réellement cause des erreurs observées en sortie reste dans le premier tiers de chaque liste de suspects constitués. Ce qui fait de notre méthode de diagnostic un outil fiable.

Puis, les expérimentations faites afin de valider les optimisations apportées par le flag ont permis de montrer que notre outil de diagnostic pouvait aller encore plus loin en terme de précision donnée sur le résultat de diagnostic. Nous avons constaté que nous pouvions cibler ou supprimer, dans 30% des cas, les modèles de stuck-on/stuck-open. Ces informations sont importantes, surtout lorsque nous passerons par la suite à l'analyse de défaillance. De la même façon, nous pouvons cibler un ensemble de ligne « agresseurs » dans le cas des courts-circuits simples qui amènent une information qui sera également très utile lors de l'analyse de défaillance.

Enfin, nous avons pu voir l'adaptabilité de notre méthode, qui moyennant quelques modifications, nous a permis de réaliser des expérimentations sur l'hypothèse du multi erreurs. Encore une fois, quel que soit le modèle de faute considéré, nous avons remarqué l'efficacité de la méthode en terme de résultat de diagnostic, puisque dans tous les cas, la ligne réellement cause des erreurs observées en sortie se trouvait dans le début de la liste ordonnancée (via le CNT) de suspects.





# *Conclusion générale*



## CONCLUSION GENERALE

Avec l'évolution de la complexité et des performances des circuits intégrés actuels, nous avons vu apparaître de nouvelles défaillances difficilement caractérisables ou assimilables (en terme de comportement) à un modèle simple de collage. Or, la plupart des méthodes de diagnostic développées à ce jour se basent sur un voire deux modèles de fautes si ceux-ci sont équivalents en terme d'erreur. Le problème est qu'avant de débiter le processus de diagnostic, nous ne connaissons pas a priori, la défaillance, et donc a fortiori le modèle de faute permettant de représenter le mieux. Nous ne pouvons donc pas utiliser une méthode de diagnostic basée sur un modèle de faute précis. La solution que nous avons proposé a donc été de développer une méthode de diagnostic basée sur un ensemble de modèles de fautes complet (le plus généralement utilisés dans la littérature).

Nous avons, dans le chapitre I, présenté l'ensemble des modèles de fautes considérés tout au long de ce mémoire. Nous avons fait une analyse complète de ceux-ci afin de déterminer les conditions de sensibilisation de chacun d'entre eux. Nous avons vu que les conditions d'activation des modèles de fautes que nous considérons étaient très diverses. Pour les fautes à effet dynamique (délai, circuit ouvert résistif, ...) ou à effet séquentiel (stuck-on/open), nous avons besoin d'une paire de vecteurs pour les sensibiliser. Pour les fautes statiques (collage, court-circuit, ...), un seul vecteur était nécessaire mais d'autres conditions de sensibilisation pouvaient être nécessaires (condition spécifique sur la ligne agresseur dans le cas du court-circuit *et* valeurs logiques opposées entre la ligne agresseur et la ligne victime).

Le second chapitre a présenté la méthode de diagnostic développée. Cette méthode est divisée en deux étapes. La première, l'étape de *localisation de fautes* permet de déterminer un ensemble de lignes suspectes qui peuvent être à l'origine des erreurs observées. Cette étape conjugue une approche « effet à cause » avec une approche « cause à effet ». L'analyse « effet à cause » basée sur le principe de traçage de chemins critiques est réalisée dans un premier temps pour déterminer les sites de pannes potentiels. Cette analyse comporte trois phases. La première, la *simulation multivaluée* du circuit permet de simuler le circuit en tenant compte de toutes les conditions de sensibilisation décrites dans le chapitre I, et ainsi de connaître les valeurs sur toutes les lignes du circuit. La seconde phase, le *traçage de chemins*

*critiques* (CPT) fournit des listes de lignes suspectes construites à partir de chaque sortie du circuit déclarée fautive durant la phase de test (testeur). Ensuite, avec l'hypothèse de la panne simple, nous pouvons faire l'*intersection des listes de suspects* obtenues durant le processus de traçage de chemins critiques. Cette opération d'intersection des lignes et des symboles permet d'obtenir une liste unique de suspects comprenant l'ensemble des lignes qui peuvent être la cause de toutes les erreurs observées.

L'analyse « cause à effet », quand à elle, est alors réalisée pour optimiser la liste de sites potentiels de pannes obtenues précédemment. Cette analyse permet de prendre en compte certaines informations issues des sorties déclarées saines (pass) lors du test. A la fin de cette seconde analyse, nous obtenons une liste de suspects réduite compte tenu des différentes conditions de sensibilisation. Cette liste est composée de lignes suspectes associées avec des symboles et des marqueurs si nécessaire (\*' et '+').

Jusqu'à cet instant, nous n'avons pas manipulé de modèles de fautes explicitement, mais uniquement des lignes et des symboles représentant des valeurs logiques. La seconde étape de ce processus de diagnostic est l'*allocation des modèles de fautes*. En fonction des symboles associés à chaque ligne suspecte et des conditions de sensibilisation décrites dans le chapitre I, il est alors possible de lier chaque ligne suspecte avec un ensemble particulier de modèles de fautes.

L'objectif du diagnostic n'est pas seulement de déterminer un ensemble de lignes critiques, mais également que ce résultat soit le plus précis possible. Nous avons donc présenté, dans la première partie du chapitre III, un moyen d'optimiser le résultat du diagnostic pour deux modèles de fautes, le court-circuit et le stuck-on/open. Grâce à la prise en compte d'une nouvelle variable appelée *flag* dans le processus de diagnostic, nous avons pu, grâce aux informations qu'elle nous a fournies, déterminer précisément les lignes agresseurs dans le cas du court-circuit, et cibler ou supprimer des transistors dans le cas du stuck-on/open. Cette amélioration dans le résultat du diagnostic a deux avantages. Le premier, dans le cas du stuck-on/open, est de pouvoir cibler/éliminer ce modèle de fautes de la liste des modèles associés à une ligne si les conditions de sensibilisation ne sont pas respectées, et donc d'augmenter la précision du diagnostic. Le second avantage est de pouvoir cibler plus précisément ces deux modèles de fautes (agresseur pour les courts-circuits et transistors pour les stuck-on/open). Ces informations supplémentaires sont un atout important qui pourra être

utilisé, par la suite, lors de l'analyse de défaillance.

La seconde partie, de ce chapitre a permis de montrer que la méthode de diagnostic était adaptable au cas du multi erreurs moyennant quelques modifications lors du processus de traçage de chemins, d'intersection et d'allocation des modèles de fautes. Ceci permet de garantir l'efficacité de la méthode, puisqu'avant même de traiter le cas du multi fautes, nous devions garantir de pouvoir traiter la cas des fautes à erreurs multiples (wired or/and, resistive wired or/and, ...). Nous avons pu également voir que ces adaptations permettaient de garantir un résultat de diagnostic même dans le cas du multi erreurs, laissant ouvert la possibilité de faire du traitement multi fautes.

Enfin, dans le chapitre IV, les aspects de validation de la méthode de diagnostic ont été abordés au travers de plusieurs expérimentations. Ces expérimentations ont permis dans un premier temps de démontrer l'utilité d'une méthode de diagnostic basée sur un ensemble complet de modèles de fautes par comparaison avec un outil industriel. Nous avons vu par la suite que la méthode de diagnostic développée ainsi que les optimisations apportées rendaient la méthode précise et fiable. La ligne suspecte et le modèle réellement cause des erreurs observées ont toujours été trouvés.

Le bilan général de cette étude ainsi que l'ensemble des expérimentations effectuées nous permettent de conclure sur l'intérêt de la méthode de diagnostic développée. L'utilisation d'une telle méthode de diagnostic permet d'obtenir un résultat qui auparavant nous aurait demandé l'utilisation de plusieurs outils, tout en conservant un résultat fiable et précis.

La prospective que nous pouvons dresser de notre travail peut se diviser principalement en deux parties : une partie valorisation de l'innovation et une partie amélioration de l'innovation.

Pour ce qui concerne la valorisation, le LIRMM a d'ores et déjà une collaboration avec la société ST Microelectronics avec pour buts i) d'implanter la méthode de diagnostic que nous avons développée en milieu industriel, ii) de comparer les résultats obtenus avec ceux fournis par la chaîne de diagnostic en cours d'utilisation dans cette société. Ces travaux s'effectuent dans le cadre d'une thèse CIFRE qui a démarré à la fin de l'année 2006.

Pour la partie amélioration nous pensons coupler notre outil avec des outils statistiques

permettant d'améliorer la finesse du diagnostic et donc de rendre plus efficace la phase suivante de caractérisation. Des études ont permis de montrer que les portes logiques pouvaient être classées selon leurs probabilités de défaillance. Ce classement est effectué en fonction de la nature de la porte (nand, nor, ouex, ...) mais également en fonction du nombre d'entrée de ces portes. Ainsi, en faisant une corrélation entre les résultats obtenus avec notre outil de diagnostic et cet outil statistique, nous pourrions obtenir un résultat plus précis et plus juste. Les mêmes observations peuvent être faites pour les modèles de court-circuit. Notre méthode de diagnostic fournit un résultat d'agresseurs potentiels sans tenir compte du layout du circuit. Avec la connaissance du layout, nous pourrions discriminer certains courts-circuits, et établir des probabilités d'apparitions de court-circuit pour les lignes restantes (en fonction de leurs positions, de leurs distances, ...).

*Références  
bibliographiques*





## Références bibliographiques

- [Abra80] M. Abramovici et al., "Multiple Fault Diagnosis in Combinational Circuits Based on an Effect-Cause Analysis", *IEEE Transactions on Computer*, vol. c-29, n°6, June 1980, pp. 451-460.
- [Abr83] M. Abramovici, P.R. Menon, D.T. Miller, "Critical Path Tracing - an Alternative to Fault Simulation", *Proceedings of the 20th conference on Design automation*, p. 214-220, 1983.
- [Abra90] M. Abramovici et al., "Digital System Testing and Testable Design", *IEEE Press*, 1990.
- [Amy06] M.E. Amyeen et al., "Improving Precision Using Mixed-level Fault Diagnosis", *Proceedings IEEE Int.Test Conf.*, p. 1-10, 2006.
- [Bar83] Z. Barzilai, B.K. Rosen, "Comparison of AC Self-Testing Procedures", *Proceedings of the International Test Conference*, p. 89-94, 1983.
- [Car87] J.L. Carter, V.S. Iyengar, B.K. Rosen, "Efficient Test Coverage Determination for Delay Faults", *Proceedings of the International Test Conference*, p. 418-427, 1987.
- [Che06] Y.Y. Chen, J.J. Liou, "Enhancing Diagnosis Resolution For Delay Faults By Path Extension Method", *Proceedings of the 21st IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT)*, p. 428-438, 2006.
- [Eng03] P. Engelke, I. Polian, M Renovell, B Becker, "Simulating Resistive Bridging and Stuck-At Faults", *Proceedings of the International Test Conference*, p. 1051-1059, 2003.
- [Gho00] J. Ghosh-Dastidar, N. A. Touba, "Diagnosing Resistive Bridges Using Adaptive Techniques", *Proceedings of the IEEE Custom Integrated Circuits Conference*, p. 79-82, 2000.
- [Gir92a] P. Girard, C. Landrault, S. Pravossoudovitch, " Delay-Fault Diagnosis Based on Critical Path Tracing from Symbolic Simulation" *Proceedings of the IEEE International Symposium on Circuits and Systems*, vol. 3, p. 1133-1136, 1992.
- [Gir92b] P. Girard, C. Landrault, S. Pravossoudovitch, "Delay-Fault Diagnosis by Critical-Path Tracing", *Proceedings of the IEEE Design & Test of Computers*,

vol. 9, issue 4, p. 27-32, 1992.

- [Han03] Shi Yu Hang, "A Symbolic Inject-and-Evaluate Paradigm for Byzantine Fault Diagnosis", *Journal of Electronic Testing: Theory and Applications*, vol. 19, issue 2, p. 161-172, 2003.
- [Hol07] S. Holst, H.-J. Wunderlich, "Adaptive Debug and Diagnosis without Fault Dictionaries", *Proceedings of the 12<sup>th</sup> IEEE European Test Symposium*, p. 7-12, 2007.
- [Hsi77] E.P. Hsieh, R. Rasmussen, L. Vidunas, W. Davis, "Delay Test Generation", *Proceedings of the Design Automation Conference*, p. 486-492, 1977.
- [Hui04] L. M. Huisman, "Diagnosing Arbitrary Defects in Logic Designs Using the Single Location At a Time (SLAT)", *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems*, vol. 23; No 1, 2004, pp.91-101.
- [Kis86] K. Kishida, F. Shirotori, Y. Ikemoto, S. Ishiyama, T. Hayashi, "A Delay Test System for High Speed Logic LSI's", *Proceedings of the Design Automation Conference*, p. 786-790, 1986.
- [Lav98] D.B. Lavo et al. , "Diagnosing Realistic Bridging Faults with Single Stuck-At Information", *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems*, vol.17, No 3, p. 255-267 1998.
- [Lev86] Y. Leventel, P.R. Menon, "Transition Faults in Combinational Circuit: Input Transition Test Generation and Fault Simulation", *Proceedings of the Fault-Tolerant Computing Symposium*, p. 278-283, 1986.
- [Lin07] Yung-Chieh Lin, Feng Lu, Kwang-Ting Cheng, "Multiple-Fault Diagnosis Based On Adaptive Diagnostic Test Pattern Generation", *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems*, vol. 26; No 5, p. 932-942 2007.
- [Liu05] J.B. Liu, A. Veneris, "Incremental Fault Diagnosis", *IEEE Transaction on Computer Aided Design of Integrated Circuits and Systems*, vol. 24, n°2, p. 240-251, 2005.
- [Lo00] C.-K. Lo, P. C. H. Chan. "An Efficient Structural Approach to Board Interconnect Diagnosis", *International journal of electronics*, vol. 87, n°10, pp. 1239-1255, 2000.
- [Meno93] S.M. Menon, et al., "Testable Design of BiCMOS Circuits for Stuck-Open Fault Detection using Single Patterns", *Proceedings of the VLSI Test Symposium*, p. 296-302, 1993.

- [Pom97] I. Pomeraz et al., "On Dictionnary-Based Fault Location in Digital Logic Circuits", *IEEE Transaction on Computers*, vol. 46, n° 1, p. 48-59, 1997.
- [Pra88] A.K. Pramanick, S.M. Reddy, "On the Detection of Delay Faults", *Proceedings of the International Test Conference*, p. 845-856, 1988.
- [Sch87] M.H. Schulz, F. Brglez, "Accelerated Transition Fault Simulation", *Proceedings of the Design Automation Conference*, p. 237-243, 1987.
- [Smi85] G.L. Smith, "Model for Delay Faults Based upon Paths", *Proceedings of the International test Conference*, p. 342-349, 1985.
- [Ven97] S. Venkataraman, W. K. Fuchs, "A Deductive Technique for Diagnosis of Bridging Faults", *Proceedings of the Proceedings of the IEEE/ACM international conference on Computer-aided design*, p. 562-567, 1997.
- [Ven01] S. Venkataraman, S.B. Drummonds, "Poirot: applications of a logic fault diagnosis tool", *IEEE Design & Test of Computer*, vol. 18, No 1, pp. 19-30, 2001.
- [Wai86] J.A. Waicukausky, E. Lindbloom, B. Rosen, V. Iyendar, "Transition Fault Simulation by Parallel Pattern Single Fault Propagation", *Proceedings of the International Test Conference*, p. 542-549, 1986.
- [Wood87] B. W. Woodhall, et al, "Empirical Results on Undetected CMOS Stuck-Open Failures", *Proceedings of the International Test Conference*, p. 166-170, 1987.



# *Bibliographie personnelle*



# Bibliographie personnelle

## Conférences avec actes et comité de lecture

A. Rousset, A. Bosio, P. Girard, C. Landrault, S. Pravossoudovitch, A. Virazel, "Improving Diagnosis Resolution without Physical Information", *IEEE International Symposium on Electronic Design, Test & Applications*, Hong Kong, January 23-25, 2008 (accepté).

A. Rousset, P. Girard, S. Pravossoudovitch, C. Landrault, A. Virazel, "Unified Framework for Logic Diagnosis", *Proceedings of IEEE East-West Design & Test Workshop (EWDTW'06)*, p. 47-52, 2006.

A. Rousset, A. Bosio, P. Girard, C. Landrault, S. Pravossoudovitch, A. Virazel, "A Mixed Approach for Unified Logic Diagnosis", *IEEE Workshop on Design and Diagnostics of Electronic Circuits and Systems (DDEC'07)*, p. 239-242, 2007.

A. Rousset, A. Bosio, P. Girard, C. Landrault, S. Pravossoudovitch, A. Virazel, "DERRIC: A Tool for Unified Logic Diagnosis", *12th IEEE European Test Symposium (ETS'07)*, p. 13-20, 2007.

A. Rousset, A. Bosio, P. Girard, C. Landrault, S. Pravossoudovitch, A. Virazel, "Fast Bridging Fault Diagnosis using Logic Information", *16<sup>th</sup> IEEE Asian Test Symposium (ATS'07)*, p. 33-38, 2007.

## Colloques sans actes ou avec actes à diffusion restreinte

A. Rousset, P. Girard, S. Pravossoudovitch, C. Landrault, A. Virazel, "Unified Diagnostic Method Targeting Several Fault Models"; *VLSI PhD forum 2007* (Poster).

A. Rousset, P. Girard, S. Pravossoudovitch, C. Landrault, A. Virazel, "State-of-the-art Diagnosis of Delay Faults in Scan Environment", SETS'2005, South European Test Seminar, St Leonhard, Tyrol, Autriche.

A. Rousset, P. Girard, S. Pravossoudovitch, C. Landrault, A. Virazel, "Unified Diagnostic Method Focusing Several Fault Models", SETS'2006, South European Test Seminar, Neustift, Autriche.

A. Rousset, A. Bosio, P. Girard, S. Pravossoudovitch, C. Landrault, A. Virazel, "A Comprehensive Diagnosis Methodology for Several Fault Models", SETS'2007, South European Test Seminar, Sestriere, Italie.

A. Rousset, P. Girard, S. Pravossoudovitch, C. Landrault, A. Virazel, "Méthode de diagnostic



unifiée ciblant plusieurs modèles de fautes", Majestic 2006, Lorient, France.

A. Rousset, P. Girard, S. Pravossoudovitch, C. Landrault, A. Virazel, "Diagnostic unifié ciblant plusieurs modèles de fautes", Journées Nationales du Réseau Doctoral de Microélectronique (JNRDM) 2006, Rennes, France.

A. Rousset, P. Girard, S. Pravossoudovitch, C. Landrault, A. Virazel, "Méthode de diagnostic unifiée pour circuits intégrés numériques", GDR Soc-Sip, Paris, France, 2007.

# *Liste des figures*



# Listes des figures

<b>FIGURE 1</b>	: EXEMPLE DE COLLAGE A 1 D'UNE CONNEXION	22
<b>FIGURE 2</b>	: MODELE GENERAL DU COURT-CIRCUIT	24
<b>FIGURE 3</b>	: COURT-CIRCUIT DE TYPE STRONG DRIVER WIRED OR	24
<b>FIGURE 4</b>	: COURT-CIRCUIT DE TYPE STRONG DRIVER WIRED AND	25
<b>FIGURE 5</b>	: COURT-CIRCUIT DE TYPE STRONG DRIVER WIRED	26
<b>FIGURE 6</b>	: COURT-CIRCUIT DE TYPE WIRED OR	26
<b>FIGURE 7</b>	: COURT-CIRCUIT DE TYPE WIRED AND	27
<b>FIGURE 8</b>	: COURT-CIRCUIT DE TYPE BYZANTINE	28
<b>FIGURE 9</b>	: CAS DES COURTS-CIRCUITS RESISTIFS	29
<b>FIGURE 10</b>	: EXEMPLE D'UN CIRCUIT OUVERT A '1'	29
<b>FIGURE 11</b>	: EXEMPLE DE CIRCUIT OUVERT RESISTIF ENTRAINANT UNE TRANSITION LENTE-A-MONTER	30
<b>FIGURE 12</b>	: EXEMPLE D'UN LENT-A-MONTER SUR UNE PORTE ET	30
<b>FIGURE 13</b>	: EXEMPLE D'UN LENT-A-DESCENDRE SUR UNE PORTE ET	31
<b>FIGURE 14</b>	: TRANSISTOR T1 STUCK-OPEN	31
<b>FIGURE 15</b>	: SITES POTENTIELS D'ERREURS AYANT PU PRODUIRE L'ERREUR OBSERVEE EN SORTIE	42
<b>FIGURE 16</b>	: DEFAUTS POUVANT ETRE A L'ORIGINE D'UNE ERREUR	43
<b>FIGURE 17</b>	: SYNOPTIQUE DU PROCESSUS DE DIAGNOSTIC DEVELOPPE	43
<b>FIGURE 18</b>	: SIMULATION DU VECTEUR $(E1,E2,E3) = (R1,R1,C0)$	48
<b>FIGURE 19</b>	: SIMULATION DU VECTEUR $(E1,E2,E3) = (R1,R1,C1)$	50
<b>FIGURE 20</b>	: EXEMPLE DE PRISE EN COMPTE DES SORTIES SAINES	55
<b>FIGURE 21</b>	: SIMULATION DU CIRCUIT POUR UN VECTEUR FAUTIF	57
<b>FIGURE 22</b>	: SIMULATION DU CIRCUIT POUR UN VECTEUR SAIN	58
<b>FIGURE 23</b>	: SIMULATION DE FAUTES	60
<b>FIGURE 24</b>	: PORTE LOGIQUE NAND AU NIVEAU TRANSISTOR	74
<b>FIGURE 25</b>	: HISTORIQUE DU CIRCUIT AU TRAVERS DU « FLAG »	75
<b>FIGURE 26</b>	: EXEMPLE DE CIRCUIT AFFECTE PAR UNE PANNE POUVANT AVOIR LES DEUX TYPES DE POLARITE (0/1 ET 1/0)	80
<b>FIGURE 27</b>	: CPT A PARTIR DE LA SORTIE $S_1$	83
<b>FIGURE 28</b>	: EXEMPLE DE PORTES NON SENSIBLES	89
<b>FIGURE 29</b>	: SYNOPTIQUE DU PROCESSUS DE DIAGNOSTIC DEVELOPPE.	97



# *Liste des tableaux*



# Liste des tableaux

<b>TABLEAU 1</b>	: DIFFERENTS MODELES DE COURTS-CIRCUITS	23
<b>TABLEAU 2</b>	: COMPORTEMENT LOGIQUE EN PRESENCE D'UN STUCK-OPEN	32
<b>TABLEAU 3</b>	: RELATION ENTRE VECTEURS ET MODELES DE FAUTES	33
<b>TABLEAU 4</b>	: TABLES DE TRANSFERT DES PORTES ELEMENTAIRES	46
<b>TABLEAU 5</b>	: VECTEURS ET REPONSES ASSOCIEES	51
<b>TABLEAU 6</b>	: OPERATION D'INTERSECTION ENTRE SIGNAUX	52
<b>TABLEAU 7</b>	: IMPLANTATION DU PROCESSUS D'INTERSECTION	55
<b>TABLEAU 8</b>	: COMPARAISON ENTRE LE NOMBRE DE SORTIES FAUTIVES ET LE NOMBRE DE SORTIES SAINES	56
<b>TABLEAU 9</b>	: VECTEURS ET REPONSES ASSOCIES	60
<b>TABLEAU 10</b>	: MODELES DE FAUTES ASSOCIES AVEC LES LIGNES CRITIQUES EN FONCTION DU SYMBOLE	62
<b>TABLEAU 11</b>	: TRANSISTORS DEFAILLANTS EN FONCTION DES VALEURS D'ENTREES	74
<b>TABLEAU 12</b>	: MODELES DE FAUTES ASSOCIES AUX LIGNES SUSPECTES	76
<b>TABLEAU 13</b>	: DIAGNOSTIC AFFINE POUR LES COURTS-CIRCUITS	77
<b>TABLEAU 14</b>	: RAPPORT DE DIAGNOSTIC FINAL	78
<b>TABLEAU 15</b>	: TABLEAU D'INTERSECTION DES SYMBOLES INITIAL	79
<b>TABLEAU 16</b>	: TABLEAU D'INTERSECTION MIS A JOUR POUR LE SYMBOLE D	80
<b>TABLEAU 17</b>	: TABLE D'INTERSECTION	82
<b>TABLEAU 18</b>	: MISE A JOUR DU TABLEAU D'ASSOCIATION AVEC LES SDW, DELAI ET COURT-CIRCUIT RESISTIF	85
<b>TABLEAU 19</b>	: MISE A JOUR DU TABLEAU D'ASSOCIATION AVEC LES WIRED AND/OR ET RESISTIVE	87
<b>TABLEAU 20</b>	: TABLEAU D'ASSOCIATION DES MODELES DE FAUTES COMPLET	90
<b>TABLEAU 21</b>	: RESULTATS EXPERIMENTAUX OBTENUS SUR LE C432 (ISCAS'85)	99
<b>TABLEAU 22</b>	: RESULTAT DE DIAGNOSTIC	101
<b>TABLEAU 23</b>	: RESULTATS OBTENUS SUR DES CIRCUITS BENCHMARK ITC'99	102
<b>TABLEAU 24</b>	: RESULTATS DU DIAGNOSTIC	104
<b>TABLEAU 25</b>	: EXPERIMENTATIONS MENEES SUR LE B21	105
<b>TABLEAU 26</b>	: EXPERIMENTATIONS MENEES SUR LE B22	106
<b>TABLEAU 27</b>	: ISCAS'89 AND ITC'99 RESULTS SUMMARY	108







---

**TITRE : Diagnostic de pannes dans les circuits logiques : Développement d'une méthode ciblant un ensemble élargi de modèles de fautes**

---

**RESUME :** Avec l'évolution de la complexité et des performances des circuits intégrés, l'occurrence de défaillances non modélisables par de simples collages devient importante et même prépondérante. Ces effets ne sont généralement pas pris en compte par les méthodes classiques de diagnostic. Cette thèse a pour objectif le développement d'une méthode de diagnostic ciblant un ensemble élargi de modèles de fautes.

La méthode de diagnostic développée est présentée dans ce manuscrit de manière progressive. Dans un premier temps, les modèles de fautes considérés sont analysés afin de dégager les conditions de sensibilisation. La deuxième partie est consacrée à la présentation globale de la méthode de diagnostic développée. Cette méthode utilise principalement une approche « Effet à Cause » basée sur le traçage de chemins critiques. La troisième partie présente l'amélioration de cette méthode pour la prise en compte de pannes à effets spécifiques. La dernière partie est consacrée à la validation de chaque étape de l'évolution de la méthode de diagnostic au travers de diverses expérimentations.

---

**MOTS CLES :** Diagnostic de circuits digitaux, analyse d'effet à cause, traçage de chemins critiques, modèles de défaillance divers

---

**DISCIPLINE :** Microélectronique

---

**TITLE : Failure diagnosis in the logic circuits : Development of a method targeting an enlarged set of fault models**

---

**ABSTRACT :** With the evolution of the complexity and performances of digital circuits, the occurrence of failures which can not be modeled by simple stuck-at faults becomes important and even preponderant. These effects are generally not taken into account by classical diagnosis methods. The purpose of this thesis is to develop a diagnosis method targeting an enlarged set of fault models.

In this manuscript, the developed diagnosis method is presented in a progressive way. First, the considered fault models are analyzed in order to show the sensitization conditions. The second part is dedicated to the whole presentation of the proposed diagnosis method. This method mainly uses an "effect-cause" approach based on critical path tracing. The third part presents the improvements of this method to take into account failures which have specific effects. The last part is dedicated to the validation of the diagnosis method throughout many experimentations on benchmark circuits.

---

**KEYWORDS :** Diagnosis of digital circuits, effect-cause analysis, critical path tracing, various fault models

---

Université de Montpellier II: Sciences et Techniques du Languedoc

LIRMM : Laboratoire d'Informatique de Robotique et de Microélectronique de  
Montpellier

161 Rue Ada - 34392 Montpellier Cedex 5