



**HAL**  
open science

# Généralisation des méthodes de scan pour le test des circuits intégrés complexes et application à des circuits critiques en vitesse

Joseph Bulone

► **To cite this version:**

Joseph Bulone. Généralisation des méthodes de scan pour le test des circuits intégrés complexes et application à des circuits critiques en vitesse. Modélisation et simulation. Institut National Polytechnique de Grenoble - INPG, 1994. Français. NNT: . tel-00344980

**HAL Id: tel-00344980**

**<https://theses.hal.science/tel-00344980>**

Submitted on 8 Dec 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.

Tu 21890

**THESE**

présentée par

**Joseph BULONE**

pour obtenir le titre de **DOCTEUR**

de l'**INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE**

(arrêté ministériel du 30 mars 1992)

**Spécialité : Microélectronique (Conception)**



**GENERALISATION DES METHODES DE SCAN POUR LE TEST DES  
CIRCUITS INTEGRES COMPLEXES ET APPLICATION A DES CIRCUITS  
CRITIQUES EN VITESSE**



Date de soutenance : 2 décembre 1994

Composition du jury :

Monsieur :	René DAVID	Président
Mesdames :	Gabrièle SAUCIER	
	Pascale THEVENOD	
Messieurs :	Yves BERTRAND	Rapporteur
	Joseph BOREL	
	Habib MEHREZ	Rapporteur



*[...] Chacun appelle idées claires celles qui sont au même  
degré de confusion que les siennes propres.*

**Marcel Proust**

*A l'ombre des jeunes filles en fleurs*

*Au nom de la grâce qui m'a été donnée,  
je vous le dis à tous et à chacun :  
ne vous surestimez pas plus qu'il ne faut vous estimer,  
mais gardez de vous une sage estime,  
chacun selon le degré de foi que Dieu lui a départi.*

*Epître aux Romains, XII, 3*

à Géraldine, sans qui ...



## Remerciements

Je tiens à remercier,

M. René DAVID, directeur de recherche au CNRS, qui me fait l'honneur de présider le jury,

Mme Gabrièle SAUCIER, directrice du laboratoire Conception de Systèmes Intégrés, qui a été mon directeur de thèse,

Mme Pascale THEVENOD, directeur de recherche au CNRS pour avoir accepté d'être membre du jury,

M. Yves BERTRAND, maître de conférences au LIRMM, pour ses conseils, et pour avoir accepté d'être rapporteur de ce travail et membre du jury,

M. Joseph BOREL, vice-président de la centrale R&D de SGS-TOMSON, qui m'a accueilli au sein de la R&D dans le cadre de ma CIFRE, et qui a accepté de faire partie du jury.

M. Habib MEHREZ, maître de conférences à l'université Pierre et Marie Curie, pour ses conseils, et pour avoir accepté d'être rapporteur de ce travail et membre du jury.

Je remercie également

Régis Leveugle, Michel Crastes de Paulet et Mihail Nicolaïdis pour leur relecture et pour leurs remarques et critiques constructives,

Mario Diaz Nava, et toute l'équipe qui a collaboré au projet LT et qui m'ont encouragé dans ce travail.

Enfin, que les personnes du CSI, de ST et d'ailleurs, que je n'ai pas pu citer et qui ont contribué de près ou de loin aux travaux de cette thèse, en soient ici remerciées.



## **Résumé**

Cette thèse propose une extension des méthodes classiques de chemins de "scan". On utilise des opérateurs combinatoires plus généraux à la place des multiplexeurs à une seule sortie. Ils peuvent comporter des entrées et des sorties multiples. Ils peuvent boucler sur eux-mêmes par l'intermédiaire d'une ou plusieurs bascules. Lorsqu'ils vérifient certaines propriétés de bijectivité et qu'ils forment une structure propageant de l'information, alors cette structure est aussi utile que les chaînes du "scan" complet et s'utilise de manière semblable. Elle permet aussi une approche hiérarchique du test des circuits. On montre comment tirer profit de cette méthode plus générale pour réduire l'impact de la méthode de "scan" complet sur les performances de circuits complexes implantant des fonctions mathématiques courantes ou des séquenceurs. Des résultats sont donnés pour le cas réel d'un circuit CMOS, très rapide, spécifique pour le réseau numérique large bande et pour lequel les contraintes en vitesse étaient primordiales.

## **Mots clés**

Chemin de scan généralisé, scan partiel, scan complet, circuits VLSI, génération de test, simulation symbolique, modèle fonctionnel, test de chemins de données, test de séquenceurs, conception de circuits très rapides.





**Abstract**

This thesis proposes an extension of the classical scan path methods. More general combinational operators are used instead of the single-output multiplexers. They may have multiple inputs and outputs. They may be in self-loops through one or more flip-flops. When they satisfy some bijectivity properties, and when they form a structure which propagates information, this structure is as useful as the full scan chains and is usable in a similar manner. It allows also a hierarchical approach of circuit test. It is shown how take advantage of this more general method, to reduce the impact of the full scan method on the performances of complex circuits implementing usual mathematical functions or sequential controllers. Results are given in a real case of a high speed, CMOS circuit dedicated to the broadband digital network, for which speed constraints were essential.

**Key words**

Generalised scan path, partial scan, full scan, VLSI circuits, test generation, symbolic simulation, functional model, data path test, sequential controller test, design of high speed circuits.



# **Table des matières**



<b>Table des matières</b> .....	11
<b>Introduction</b> .....	21
<b>Chapitre 1 :</b>	
<b>Etat de l'art sur le scan partiel</b> .....	27
<b>I. <u>Approches utilisant des mesures de testabilité</u></b> .....	29
<b>I.1 Introduction</b> .....	29
<b>I.2 L'apport de la testabilité</b> .....	30
<b>I.3 Approche avec des contraintes de coût pour le dessin</b> .....	30
<b>I.3.1 La fonction de coût</b> .....	30
<b>I.3.2 Les critères de testabilité</b> .....	31
<b>I.3.3 Le processus d'optimisation</b> .....	32
<b>I.3.4 Application au premier critère de testabilité</b> .....	33
<b>I.3.5 Application au deuxième critère de testabilité</b> .....	34
<b>I.3.6 Application au troisième critère de testabilité</b> .....	34
<b>I.3.7 Analyse et résultats</b> .....	35
<b>I.4 Approche par sélections et combinaisons d'heuristiques</b> .....	37
<b>I.4.1 Les cinq heuristiques de base</b> .....	37
<b>I.4.2 Les heuristiques de combinaison</b> .....	39
<b>I.4.3 Algorithme de sélection des heuristiques</b> .....	41
<b>I.4.4 Analyse et résultats</b> .....	42
<b>I.5 Approche par testabilité empirique</b> .....	43
<b>I.5.1 Différence de testabilité d'un circuit</b> .....	43
<b>I.5.2 Mesures de testabilité empirique</b> .....	44
<b>I.5.3 Algorithme par différence de testabilité empirique</b> .....	46

I.5.4	Algorithme par différence de testabilité empirique renversée .....	47
I.5.5	Accélération par sélection multiple de bascules .....	48
I.5.6	Analyse et résultats .....	48
<b>II.</b>	<b><u>Approches utilisant des générateurs automatiques de vecteurs de test</u></b> .....	<b>50</b>
II.1	Introduction .....	50
II.2	Approche par calcul des non contrôlabilités et des non observabilités ...	50
II.2.1	Description .....	50
II.2.2	Analyse et résultats .....	51
II.3	Approche en fréquence et en distance .....	51
II.3.1	Description .....	51
II.3.2	Analyse et résultats .....	53
II.4	Approche par énumération d'états et par utilisation d'un graphe de transitions .....	53
II.4.1	Principes .....	53
II.4.2	Sélection des bascules à mettre en scan .....	53
II.4.3	Analyse et résultats .....	54
<b>III.</b>	<b><u>Approches basées sur le graphe séquentiel ou topologique du circuit</u></b> .....	<b>55</b>
III.1	Introduction .....	55
III.2	Equilibrage des chemins séquentiels : BALLAST .....	55
III.2.1	Description .....	55
III.2.2	Analyse et résultats .....	56
III.3	Suppression de cycles ou diminution de la profondeur séquentielle .....	57
III.3.1	Suppression des cycles les plus longs .....	57
III.3.2	Description .....	57
III.3.3	Analyse et résultats .....	59

<b>IV. <u>Autres approches symboliques ou de haut niveau</u></b> .....	60
<b>IV.1 Introduction</b> .....	60
<b>IV.2 Les principales méthodes</b> .....	60
<b>IV.2.1 Méthodes statistiques ou probabilistes</b> .....	60
<b>IV.2.2 Méthodes structurelles et symboliques</b> .....	61
<b>V. <u>Synthèse et conclusion</u></b> .....	63
 <b>Chapitre 2 :</b>	
<b>Proposition, formalisation et mise en œuvre d'une méthode de scan généralisée</b> .....	
	67
<b>I. <u>Introduction</u></b> .....	69
<b>II. <u>Modes locaux combinatoires</u></b> .....	74
<b>II.1 Préliminaires</b> .....	74
<b>II.1.1 Modèles utilisés pour le circuit</b> .....	74
<b>II.1.2 Conventions d'écriture</b> .....	75
<b>II.2 Description d'un mode local combinatoire</b> .....	75
<b>II.2.1 Modélisation</b> .....	76
<b>II.2.2 Représentation graphique</b> .....	77
<b>II.2.3 Exemples et modes particuliers</b> .....	78
<b>II.3 Propriétés associées aux modes locaux</b> .....	79
<b>II.3.1 Surjectivité, injectivité et bijectivité</b> .....	79
<b>II.3.2 Disjonction</b> .....	80
<b>III. <u>Modes globaux</u></b> .....	81
<b>III.1 Définitions préliminaires</b> .....	81
<b>III.2 Propriétés associées aux modes globaux</b> .....	81



III.2.1 Surjectivité et injectivité .....	81
III.2.2 Disjonction .....	81
III.2.3 Bascules isolées en sortie et en entrée, chaînage avant et arrière	82
III.2.4 Cyclicité .....	83
III.2.5 Dépendance temporelle .....	85
III.2.6 Propriétés de scan étendues .....	87
<b>IV. <u>Théorèmes sur les propriétés de scan</u> .....</b>	<b>88</b>
IV.1 Enoncés .....	88
IV.2 Démonstrations .....	88
IV.2.1 Notations et définitions préliminaires .....	88
IV.2.2 Démonstration du théorème de contrôlabilité faible .....	90
IV.2.3 Démonstration du théorème d'observabilité faible .....	96
IV.2.4 Démonstration du théorème de contrôlabilité forte .....	100
IV.2.5 Démonstration du théorème d'observabilité forte .....	100
IV.2.6 Démonstration du théorème d'accessibilité faible .....	100
IV.2.7 Démonstration du théorème d'accessibilité forte .....	100
IV.2.8 Démonstration du théorème de contrôlabilité continue .....	101
<b>V. <u>Application au test structurel</u> .....</b>	<b>104</b>
V.1 Schéma d'utilisation des théorèmes sur les propriétés de scan .....	104
V.2 Mise en œuvre des contrôles et de l'observation pour le mode global .....	104
V.2.1 Algorithme de mise en œuvre du contrôle instantané .....	104
V.2.2 Exemple d'application .....	107
V.2.3 Algorithme de mise en œuvre du contrôle continu .....	110
V.2.4 Exemple d'application .....	113
V.2.5 Algorithme de mise en œuvre de l'observation .....	115

V.2.6 Exemple d'application .....	116
V.2.7 Améliorations des mises en œuvre .....	117
V.2.8 Evaluation de la complexité .....	120
V.3 Traitement des bascules de rebouclage seul et limitations potentielles ....	122
V.3.1 Contrôle des points seulement rebouclants .....	122
V.3.2 Observation des points seulement rebouclants .....	125
V.3.3 Limitations dues à l'utilisation d'un modèle fonctionnel .....	127
<b>VI. <u>Méthodes d'obtention et de sélection de modes globaux adaptés</u></b> .....	129
VI.1 Introduction .....	129
VI.2 Obtention des modes locaux .....	129
VI.2.1 Utilisation de blocs fonctionnels du circuit .....	129
1 - Opérateurs unaires .....	130
2 - Multiplexeur .....	130
3 - Additionneur ou soustracteur .....	131
4 - Multiplicateur .....	133
5 - Opérateurs galoisiens .....	133
6 - Composition d'opérateurs .....	134
VI.2.2 Constructions de modes locaux optimaux .....	135
1 - Registre d'accumulation .....	135
2 - Multiplicateur .....	136
3 - Séquenceur .....	137
VI.3 Obtention des modes globaux .....	139
VI.3.1 Formulation du problème .....	139
1 - Principes .....	139
2 - Obtention du graphe d'incompatibilité des modes locaux pour la	

disjonction .....	140
3 - Obtention du graphe maximal des modes locaux .....	141
VI.3.2 Réductions du problème .....	141
VI.3.3 Heuristique proposée .....	142
VI.3.4 Chaînage des modes locaux .....	143
VI.3.5 Construction et conception hiérarchique .....	145
VI.3.6 Extensions envisageables .....	146
<b>Chapitre 3 :</b>	
<b>Application à des circuits de terminaison de ligne pour l'ATM .....</b>	<b>149</b>
<b>I. <u>Introduction</u> .....</b>	<b>151</b>
<b>II. <u>Spécifications du LTP</u> .....</b>	<b>154</b>
<b>II.1 Structure des données utilisées .....</b>	<b>154</b>
<b>II.2 Fonctions d'émission .....</b>	<b>155</b>
<b>II.3 Fonctions de réception .....</b>	<b>156</b>
<b>II.4 Fonctions d'interfaçage avec un microprocesseur .....</b>	<b>157</b>
<b>II.5 Résumé des caractéristiques .....</b>	<b>157</b>
<b>III. <u>Une conception orientée test et vitesse</u> .....</b>	<b>159</b>
<b>III.1 Une méthode de conception descendante .....</b>	<b>159</b>
<b>III.2 Obtention des modes locaux .....</b>	<b>160</b>
<b>III.2.1 Bascules des pipelines .....</b>	<b>160</b>
<b>III.2.2 Bascules de calcul de syndrome et de champ HEC .....</b>	<b>161</b>
<b>III.2.3 Bascules des générateurs de séquences pseudo-aléatoires .....</b>	<b>161</b>
<b>III.2.4 Bascules des séquenceurs .....</b>	<b>162</b>
<b>III.3 Obtention et activation des modes globaux .....</b>	<b>162</b>

III.4 Génération des vecteurs de test et résultats .....	163
IV. <u>Autres emplois des modes globaux</u> .....	166
<b>Conclusion</b> .....	169
<b>Références bibliographiques</b> .....	173
<b>Annexe A : Eléments de la théorie des graphes</b> .....	187
<b>Annexe B : Fonctions de coûts tirées du dessin des masques</b> .....	193
<b>Annexe C : Calcul des mesures de SCOAP</b> .....	197
<b>Annexe D : Principes de l'algorithme "0/1 knapsack"</b> .....	201
<b>Annexe E : Une heuristique de suppression de cycles d'un graphe</b> .....	207
<b>Annexe F : Algorithme de génération du nuage d'influence</b> .....	211
<b>Annexe G : Calcul des non détectabilités</b> .....	215
<b>Annexe H : Principes de l'ATPG STALLION</b> .....	219
<b>Annexe I : Algorithmes de sélection des lignes d'état</b> .....	223
<b>Annexe J : Estimation du taux de couverture pour les approches en fréquence et en distance</b> .....	227
<b>Annexe K : Inversion des opérateurs galoisiens</b> .....	231
<b>Liste des figures</b> .....	237



# **Introduction**



Etant donné un circuit et un modèle de fautes, l'objectif des méthodes de test est de déterminer les vecteurs de test qui permettent d'atteindre une couverture maximale de fautes, et s'il le faut, de modifier le circuit en conservant sa fonctionnalité pour améliorer cette couverture. On pourra se référer à [ABRA92] pour la définition du taux de couverture<sup>1</sup> et sa signification réelle. Le fait d'avoir un circuit séquentiel rend le problème du test beaucoup plus complexe [BOUR71] [BREU76] [HENN64] [MICZ83], même dans le cas d'un modèle<sup>2</sup> simple comme le collage à 1 ou à 0 (hypothèse de l'existence d'un nœud d'une description au niveau porte ou inférieur qui est toujours collé à 1 ou toujours collé à 0). Pour pallier cette difficulté, en général on a cherché à ramener le problème du test d'un circuit séquentiel à celui d'un circuit combinatoire plus facilement traitable. La méthode de scan<sup>3</sup> complet [EICH77] [AGRA84] [CLUS84] apporte alors l'une des solutions les plus efficaces en terme de taux de couverture de fautes. Elle est aussi l'une des plus utilisées car elle est en soit facile à mettre en œuvre. Elle consiste à interconnecter toutes les bascules du circuit, par l'intermédiaire d'un multiplexeur placé devant chaque bascule, pour former ainsi une seule chaîne, dite de scan. Dans le mode dit de scan, le circuit se comporte alors comme un registre à décalage. Cependant, différents coûts<sup>4</sup> sont induits par cette méthode : des coûts en surface pour convertir les bascules ou ajouter des multiplexeurs, des coûts en vitesse du circuit, des coûts en terme de longueur des séquences de test. C'est pourquoi, il a été fait appel à des méthodes de scan partiel, en tant que méthodes efficaces au niveau des coûts. Dans ces méthodes, seule une partie des bascules du circuit est mise dans une ou plusieurs chaînes de scan, ce qui permet de réduire les coûts. Cependant, la plupart de ces méthodes nécessitent a priori des générateurs automatiques de vecteurs de test (ATPG<sup>5</sup>) séquentiels et de ce fait sont aussi mal adaptées aux approches hiérarchiques.

Dans une première partie, on présentera l'état de l'art sur le scan partiel. On exposera les différentes approches et les différents algorithmes employés. Pour les classer, on a utilisé le critère dominant ou essentiel pour chacune des approches. Ces critères peu-

1. Abramovici, dans [ABRA93], fait le lien avec le rendement, et des recommandations sur son emploi.

2. Les modèles de fautes de collage sont de moins en moins suffisants car ils offrent moins de garantie face aux nouvelles technologies [MAXW93]. Il faut donc identifier et s'intéresser aux autres types de défauts plus proches de la réalité des processus de fabrication [SODE93].

3. Le terme anglais scan couramment utilisé, sera employé au lieu du terme français balayage ou scrutation

4. On trouvera dans [LEVI89] une étude sur l'aspect économique de l'emploi du scan (rendements, réponse au marché, ...). Dans [TURI93], on trouvera une brève présentation sur la façon de gérer les problèmes de test dans un projet de conception d'un circuit.

5. On utilisera l'abréviation anglaise ATPG pour "Automatic Test Pattern Generator", couramment employée au lieu de GAVT en français.



vent être aussi bien les types de données ou les types d'outils utilisés. Bien que ces choix soient personnels, on a cependant essayé de respecter la philosophie des auteurs qui pouvait apparaître dans les articles. Il va de soit que certaines approches peuvent être considérées comme appartenant à plusieurs de ces catégories. Pour synthétiser, sont récapitulées dans un tableau (au V page 63) la plupart des méthodes citées en indiquant pour chacune d'entre elles les outils et les niveaux de description exacts utilisés.

Dans une deuxième partie, une nouvelle approche de scan partiel est proposée. Elle vise à obtenir un taux de couverture maximal comme pour le scan complet. Elle n'est pas liée à un ATPG séquentiel, un ATPG combinatoire suffit. Le principe consiste à mieux exploiter les fonctionnalités de certaines parties du circuit. Pour cela de nouvelles structures de test ont été créées. Elles étendent les notions de chaînes de scan. Ces structures ont été modélisées de manière formelle et appelées modes globaux. De même des propriétés suffisantes pour qu'elles puissent remplacer les chaînes de scan dans le cadre du test structurel d'un circuit ont été formulées et démontrées mathématiquement. Elles permettent l'utilisation non seulement de fonctions combinatoires beaucoup plus complexes que celles des multiplexeurs, mais encore d'ensembles de bascules qui rebouclent sur eux-mêmes. C'est-à-dire, lorsque les propriétés précédentes sont vérifiées, elles garantissent que les boucles et les fonctions combinatoires utilisées n'affectent pas les propriétés de contrôle et d'observation qui sont attribuées aux bascules. En revanche, spécialement à cause de ces boucles, la mise en œuvre dans un cas général est rendue beaucoup plus complexe. Il n'y a pas d'a priori sur le niveau exact de description. L'important est de pouvoir simuler les états des bascules du circuit quelles soient vues isolément ou à l'intérieur de registres et de pouvoir réaliser des inversions de fonctions afin de simuler avec un sens du temps renversé.

Les fonctions combinatoires existant dans les circuits et les plus appropriées pour ce genre d'utilisation sont les fonctions ayant certaines propriétés associées à la bijectivité. Elles peuvent être parfois utilisées telles quelles, et parfois il faut modifier leur comportement pour les modes de test. Ces modifications sont toujours faites dans l'optique d'avoir un impact minimum d'abord sur le chemin critique et ensuite sur la surface de silicium supplémentaire. En particulier, si le critère de vitesse est primordial, alors des modifications pourront entraîner des affaiblissements dans les propriétés de contrôle et d'observation. Dans ces cas seulement, il faudra recourir à un simulateur de fautes ou à un ATPG séquentiel pour générer les vecteurs de test pour les parties concernées. Des séquenceurs sont aussi rendus testables à faible coût en vitesse et en surface par optimisation du codage et par emploi du scan partiel. Cette méthode permet aussi des approches hiérarchiques de test.

Dans une troisième partie, une application à un circuit réel de télécommunication est présentée. Pour être compatible avec des recommandations internationales pour les réseaux numériques à large bande, ce circuit implanté avec une technologie CMOS  $0.7\mu\text{m}$  avait de très fortes contraintes de vitesse. Il devait fonctionner à 78 MHz en pire cas. Cela a été réalisé en utilisant à la fois une méthodologie de conception descendante à l'aide d'outils de synthèse, tout en lui conférant une testabilité maximale.



# **Chapitre 1 : Etat de l'art sur le scan partiel**

Ce chapitre présente les principales méthodes existantes pour implanter du scan partiel. Certes une analyse comparant quantitativement ces méthodes serait idéale, mais elle n'est pas facilement réalisable en pratique étant donné la diversité et les spécificités des méthodes et de leurs variantes. De plus les résultats dépendraient des caractéristiques des circuits de test utilisés. C'est pourquoi, sont avant tout analysés et comparés les différents principes et concepts employés dans les algorithmes pour repérer à la fois les avantages, les difficultés et les résultats auxquels on peut s'attendre. En synthèse, on insistera en particulier sur les limitations dues à l'emploi de générateurs automatiques de vecteurs de test séquentiels et aux contraintes en vitesse imposées pour la conception.



## **I. Approches utilisant des mesures de testabilité**

### **I.1 Introduction**

Les mesures de testabilité, sont des méthodes de calcul sur des circuits pour évaluer la probabilité pour qu'un ensemble de fautes puisse être détecté. Elles aident aussi à estimer la difficulté du test d'un circuit, et donc la longueur des séquences de test qu'il faudra utiliser pour obtenir un taux de couverture donné. Les algorithmes d'analyse de testabilité se différencient fortement de ceux de génération de vecteurs de test en ce sens qu'ils réclament beaucoup moins de puissance de calcul. Leurs buts sont d'être les plus précis possibles dans leurs estimations pour pouvoir aider au mieux le concepteur dans la détection des problèmes de test potentiels du circuit dès les premières étapes de conception. Cependant, les mesures de testabilité gardent leur caractère purement statistique : leurs résultats sur la testabilité finale peuvent varier d'un circuit à l'autre, et elles ne fournissent que peu d'informations sur le fait qu'une faute sera détectée ou non par un générateur de vecteurs de test, mais donnent quelques indications sur la probabilité de détection d'une faute. V. D. Agrawal, dans [AGRA82] présente un outil statistique d'évaluation de la qualité d'information extraite par les mesures de testabilité. Ces mesures de testabilité sont donc naturellement utilisées pour choisir où insérer des points de test [SEIS91], ou pour choisir quelles bascules sont à mettre dans une chaîne de scan (on utilisera par la suite l'expression "mettre en scan") dans le cadre d'une méthode de scan partiel. Ces méthodes s'intéressent en général beaucoup plus au problème du choix des bascules qu'à celui de la génération des vecteurs de test effectifs. Le processus de génération des vecteurs de test peut être ensuite utilisé pour vérifier l'efficacité des heuristiques de choix des bascules. La notion de mesure de testabilité est en soi une notion très large. Dans les mesures de testabilité, on peut tenir compte des portes combinatoires en tant que fonctions booléennes ou non. Si on ne tient pas compte de leur fonction, seules les interconnexions des portes prennent une signification. Dans les méthodes présentées, celles qui sont exclusivement rattachées au graphe séquentiel du circuit, ont été mises de préférence au III page 55 traitant des "Approches basées sur le graphe séquentiel ou topologique du circuit"<sup>1</sup>. On y retrouvera en particulier les principales méthodes qui cherchent à supprimer des cycles dans un graphe<sup>2</sup> ou à diminuer sa profondeur séquentielle.

<sup>1</sup>. Précisons que le graphe séquentiel du circuit est le graphe dont les sommets sont les bascules du circuit et dont les arcs sont les liens combinatoires entre ces entrées et sorties de bascules. Il est parfois aussi appelé graphe de dépendance. Le graphe topologique sera défini au III.

<sup>2</sup>. Pour toutes les notions concernant la théorie des graphes, on pourra voir l'annexe A, page 187: "Éléments de la théorie des graphes".

## **I.2 L'apport de la testabilité**

Trischler dans [TRIS80] présente des règles et recommandations pour l'implantation d'une chaîne de scan incomplète c'est-à-dire pour l'implantation d'un scan partiel. Différentes méthodes de génération des vecteurs de test seront utilisées en fonction de l'implantation choisie. Son objectif est avant tout de diminuer de manière significative le coût de la génération des vecteurs de test, de la simulation de fautes et la longueur de la séquence de test, ceci sans utiliser une méthode de scan complet parce que le circuit ne le permet pas ou parce que le coût en surface de silicium est trop élevé. Les règles et recommandations présentées sont d'ordre plutôt qualitatif et pas toujours facilement mesurables. Elles sont basées sur l'appréciation du concepteur qui détermine manuellement les bascules facilement accessibles à partir des entrées et sortie primaires. Les principales règles peuvent se résumer de la façon suivante : les bascules trouvées n'ont pas alors besoin d'être mises en scan. Ces règles ont aussi pour but la simplification de la simulation et la génération des vecteurs de test dans la partie séquentielle du circuit non concernée par le scan. Trischler a proposé d'investiguer l'utilisation de méthodes de mesures de testabilité comme celle fournie dans [GOLD79] pour savoir s'il faut mettre ou non les bascules en scan. Cela a été fait et amélioré par de nombreux auteurs. Miczo, dans [MICZ86], souligne le fait que les mesures de testabilité permettent de trouver les nœuds qui sont difficiles à tester et qui souvent ne sont pas facilement identifiables.

## **I.3 Approche avec des contraintes de coût pour le dessin**

Dans le cadre de méthodes de conception utilisant des bibliothèques de cellules précaractérisées, on désire lors de l'insertion d'un scan partiel avoir un impact minimum sur le coût en surface du circuit ou sur ses performances en vitesse. Pour cela, Chickermane, dans [CHIC90], propose de sélectionner les bascules en maximisant des critères de testabilité sous la contrainte d'une fonction de coût qui doit rester bornée par une valeur préalablement définie.

### **I.3.1 La fonction de coût**

Le coût de conversion d'une bascule en une bascule de scan (bascule qui intègre un multiplexeur pour une méthode de scan) est évalué comme la somme de trois coûts :

- Le coût de congestion d'un canal de routage, i.e. l'augmentation en hauteur d'une rangée de cellules précaractérisées. Il est noté  $w_c$ .
- Le coût en largeur, i.e. le dépassement en largeur de la rangée de cellules précaractérisées la plus large. Il est noté  $w_s$ .

- Le coût en longueur des fils, utilisé pour estimer le coût en vitesse. Il est noté  $w_d$ .

Ces coûts sont calculés à partir d'un placement et d'un routage donnés. Dans cette méthode, on peut a priori pondérer les différents coûts ou même utiliser d'autres fonctions de coût qui correspondent à des contraintes "plus fines" ou autres que l'on veut imposer à la solution du problème de sélection de bascules à mettre en scan. Cependant, pour conserver la facilité de calcul, il vaut mieux avoir des coûts additifs.

### **1.3.2 Les critères de testabilité**

On recherche à sélectionner les bascules qui ont le plus d'influence sur la testabilité du circuit. Donc, à partir d'une mesure de testabilité  $T$  ( $T$  est définie de telle façon que plus  $T$  est petit, plus les nœuds du circuit sont accessibles), on emploie une fonction de gain qui est définie comme la diminution de  $T$  lorsque l'on met un ensemble de bascules en scan. L'objectif est de maximiser cette fonction de gain. Trois critères de testabilité ont été utilisés par Chickermane.

- **Critère de testabilité 1** : il est basé sur des analyses structurelles du circuit, dérivées du graphe séquentiel du circuit, en ce sens qu'il détermine le nombre de cycles que comporte le graphe après suppression des bascules mises en scan. Dans un deuxième temps si on a supprimé tous les cycles sous la contrainte de coût donnée, alors il s'intéresse à augmenter le nombre de bascules à mettre en scan de façon à diminuer la profondeur séquentielle du graphe sans les bascules mises en scan.
- **Critère de testabilité 2** : il est basé sur les mesures de contrôlabilité et d'observabilité appelées SCOAP pour les circuits séquentiels et décrites par Goldstein dans [GOLD79]. SCOAP définit six fonctions, divisées en deux classes : combinatoires et séquentielles (voir annexe C, page 197 : "Calcul des mesures de SCOAP"). Dans chaque classe, trois entiers positifs sont affectés à chaque nœud, représentant la 1-contrôlabilité, la 0-contrôlabilité, et l'observabilité. Ici, on ne considère que les trois mesures séquentielles parce que ce sont principalement elles qui sont influencées lorsqu'on fait des transformations de bascules en bascules de type scan. Les contrôlabilités séquentielles donnent une mesure du nombre de coups d'horloge requis pour contrôler un nœud donné. En effet, elles estiment le nombre minimum de nœuds séquentiels qui doivent être fixés à des valeurs logiques spécifiées afin de justifier un 0 ou un 1 sur ce nœud. De manière similaire, l'observabilité séquentielle d'un nœud est liée au nombre de coups d'horloge requis pour propager une valeur sur cette ligne à une sortie primaire, ainsi qu'au nombre de nœuds séquentiels à être contrôlés pour assurer la propagation. Ces mesures ne tiennent pas compte des corrélations possibles entre les signaux du circuit, néanmoins elles restent intéressantes pour comparer la testabilité de deux circuits.



- **Critère de testabilité 3** : il est aussi basé sur des analyses structurelles du circuit, il consiste en un calcul théorique d'un majorant des longueurs des séquences de test. La longueur d'une séquence de test désigne celle de la plus longue séquence pour détecter une faute dans un circuit séquentiel. La procédure complète est donnée dans [HUDL89], elle démarre à partir du graphe séquentiel associé à un circuit. Le principe consiste à partitionner et à réduire le graphe de façon à pouvoir considérer les sous-graphes du partitionnement comme des sous-machines mises en série et en parallèle. Ces machines sont traitées de manière individuelle pour calculer des majorants des longueurs de séquences de test. Soient deux machines M1 et M2 associées à B1 et B2, des majorants respectifs des longueurs des séquences de test, alors la connexion en série de M1 et M2 aura un majorant pour les séquences de test de B1+ B2, et la connexion en parallèle donnera le majorant B1 \* B2.

### 1.3.3 Le processus d'optimisation

Le problème d'optimisation peut être formalisé comme suit. Soit n le nombre de bascules dans le circuit séquentiel donné. On décrit une configuration des bascules par le n-uplet  $(x_1, x_2, x_3 \dots x_n)$  avec  $x_i \in \{0,1\}$ , la valeur 0 pour  $x_i$  dénote que la ième bascule n'a pas été mise en scan et la valeur 1, qu'elle a été mise en scan. Soit  $w_i$  le coût pour convertir la ième bascule en scan. Soit  $\Delta T(x_1, x_2, x_3 \dots x_n)$  la fonction de gain en testabilité, i.e. l'amélioration de la testabilité T dans la configuration  $(x_1, x_2, x_3 \dots x_n)$ . Le problème se résume alors à maximiser  $\Delta T(x_1, x_2, x_3 \dots x_n)$  sous la contrainte que

$$\sum_{1 \leq i \leq n} x_i \cdot w_i \leq \text{Coût\_maximum}, \text{ où Coût\_maximum est une limite fixée au départ.}$$

L'espace de recherche a  $2^n$  éléments, sa taille croît exponentiellement avec le nombre de bascules. Il faut éviter d'explorer tout l'arbre des solutions. Donc quand on a une solution partielle, on évalue une fonction de majoration  $\beta T(x_1, x_2, x_3 \dots x_n)$  pour prédire si les développements futurs de la solution partielle fourniront de meilleurs résultats. Ce type de problème, appelé problème "0/1 knapsack" (problème de remplissage d'un sac à dos pour optimiser un gain avec une contrainte de poids) est NP-complet. La même structure de base pour l'algorithme est utilisée pour les trois mesures de testabilité traitées, elle est décrite dans l'annexe D, page 201 : "Principes de l'algorithme "0/1 knapsack"". Les algorithmes vont différer par les deux fonctions calcul\_du\_gain et calcul\_majorant propres à chaque critère de testabilité.

### **I.3.4 Application au premier critère de testabilité**

Supprimer les cycles et réduire la profondeur séquentielle du graphe séquentiel permet de diminuer la complexité de la génération des vecteurs de test qui dépend exponentiellement de la longueur des cycles du graphe [WUND89] [CHENG90]. Deux problèmes d'optimisation se posent.

Le premier est de sélectionner un ensemble S1 de bascules sous la contrainte d'un coût inférieur à coût\_maximum de façon à minimiser le nombre de cycles restant dans le circuit après suppression des bascules de S1 mises en scan.

Le deuxième problème, si le premier a été résolu et a donné comme résultat un nombre nul de cycles, est de sélectionner un ensemble S2 supplémentaire de bascules sous la contrainte d'un coût pour S2 inférieur à (coût\_maximum - coût(S1)), de façon à minimiser la profondeur séquentielle du circuit après suppression des bascules de S1 et de S2 mises en scan.

Dans l'annexe E, page 207: "Une heuristique de suppression de cycles d'un graphe", on décrit comment la technique "knapsack" est appliquée au graphe séquentiel après des simplifications lorsque le circuit a de nombreuses bascules. Le gain pour la mise en scan de la bascule k est  $p_k$ , le nombre de cycles auxquels appartient la bascule k. Le gain d'un ensemble S, ou gain(S) est le nombre de cycles qui passent par au moins l'un des éléments de S. Finalement une fonction de majoration est donnée par la relation évidente :  $\text{gain}(S \cup \{k\}) \leq \text{gain}(S) + p_k$ .

Le problème de minimisation de la profondeur séquentielle d'un graphe (ou d'un circuit) par suppression de sommets (ou de bascules) est NP-dur. Soit seq\_dist[i] la longueur du chemin le plus long de la bascule i à l'entrée primaire ou à la sortie primaire la plus éloignée. Elle est calculée facilement par une recherche qui se fait prioritairement en largeur à partir des entrées et sorties primaires. Le fait de mettre k en scan réduit la profondeur pour les autres bascules du circuit. Une définition formelle de  $p_k$  associé deuxième problème peut être donnée :

$$p_k = \left\{ \sum_{i \neq k} \text{seq\_dist}[i] \right\}_{\text{graphe du circuit sans scan}} - \left\{ \sum_{i \neq k} \text{seq\_dist}[i] \right\}_{\text{graphe du circuit avec la bascule k seule mise en scan}}$$

cule k seule mise en scan.

Pour le calcul du gain pour un ensemble S solution intermédiaire au problème, il faut réévaluer les seq\_dist[i] en fonction des bascules qui sont mises en scan. Le gain pour S est alors donné par :

$$\text{gain}(S) = \left\{ \sum_{i \neq k} \text{seq\_dist}[i] \right\}_{\text{graphe du circuit sans scan}} - \left\{ \sum_{i \neq k} \text{seq\_dist}[i] \right\}_{\text{graphe du circuit avec les bascules de } S \text{ seules mises en scan}}$$

Une fonction de majoration provient de l'inégalité :  $\text{gain}(S \cup \{k\}) \leq \text{gain}(S) + p_k$ .

### I.3.5 Application au deuxième critère de testabilité

On utilise les mesures séquentielles de testabilité de SCOAP. Pour une bascule  $i$ , soient  $SC^0[i]$  et  $SC^1[i]$ , respectivement la contrôlabilité séquentielle à 0 et à 1 de sa sortie. Soit  $O[i]$  l'observabilité séquentielle de son entrée. Lorsqu'une bascule  $k$  est mise en scan, les observabilités et les contrôlabilités séquentielles changent. En effet, on considère alors la sortie de la bascule comme une entrée secondaire et son entrée comme une sortie secondaire. Le gain  $p_k$  pour la mise en scan de la bascule  $k$  seule, est donné par :

$$p_k = \left\{ \sum_{i \neq k} SC^0[i] + SC^1[i] + O[i] \right\}_{\text{circuit sans scan}} - \left\{ \sum_{i \neq k} SC^0[i] + SC^1[i] + O[i] \right\}_{\text{circuit avec la bascule } k \text{ seule mise en scan}}$$

Le gain en testabilité pour la mise en scan des bascules de l'ensemble  $S$  est donné par :

$$\text{gain}(S) = \left\{ \sum_{i \neq k} SC^0[i] + SC^1[i] + O[i] \right\}_{\text{circuit sans scan}} - \left\{ \sum_{i \neq k} SC^0[i] + SC^1[i] + O[i] \right\}_{\text{circuit avec les bascules de } S \text{ seules mises en scan}}$$

Finalement une fonction de majoration est donnée par la relation simple :  $\text{gain}(S \cup \{k\}) \leq \text{gain}(S) + p_k$ .

Le critère utilisé est globalement facilement calculable et mesure de manière inhérente la notion de profondeur séquentielle, ainsi que la difficulté relative à forcer la valeur logique 0 ou 1 aux entrées des bascules et à observer leurs sorties.

### I.3.6 Application au troisième critère de testabilité

Voici les principales étapes utilisées pour calculer et ensuite minimiser les bornes supérieures des longueurs de séquences de test :

**Etape 1 :** trouver les composantes fortement connexes (CFCs) du graphe séquentiel  $G$  associé au circuit.

**Etape 2 :** générer  $G^S$ , le graphe réduit de  $G$  où chaque sommet de  $G^S$  représente une CFC de  $G$ , et où chaque arc de  $G$  reliant deux composantes distinctes est reporté au niveau de  $G^S$  entre les deux sommets correspondants.

**Etape 3 :** emploi de techniques de regroupement pour simplifier  $G^S$ . La borne supé-

rieure pour chacune des sous-machines est alors prise comme étant  $3^n$  (où  $n$  est nombre de sommets ou bascules).

**Etape 4 : dérivation d'une machine équivalente sous la forme série-parallèle.**

Si les bornes supérieures sont très élevées, on peut alors les réduire avant l'étape 2 en procédant sur les CFCs comme dans le cas de la minimisation du nombre de cycles.

On utilise  $T(G^S)$ , la borne supérieure des longueurs des séquences de test, comme mesure de testabilité.  $T$  est une expression polynomiale dont les variables sont les bornes pour chacune des sous-machines. Par construction (mise en parallèle et en série),  $T$  est de degré au plus 1 par rapport à chacune de ces variables. Donc pour une sous-machine  $M$ , quand on gagne  $\delta(M)$  sur sa borne associée, alors en fonction des mises en parallèle de  $M$  dans  $G^S$  et des bornes associées aux autres sous-machines, il y a un gain global qui peut se noter  $\frac{\partial T}{\partial M} \times \delta(M)$ , et où  $\frac{\partial T}{\partial M}$  ne dépend pas la borne associée à  $M$ . Si on considère une bascule  $k$  qui se trouve dans  $M$  et qui apporte une diminution de  $\delta(M, k)$  de la borne pour la sous-machine  $M$ ,  $p_k$  est donné par :

$$p_k = \frac{\partial T}{\partial M} \times \delta(M, k).$$

Pour un ensemble  $S$  de bascules mises en scan, on évalue  $T(G^S)$  en réappiquant les étapes 2 et 3. Cette opération est en  $O(n^2)$ . La diminution de  $T(G^S)$  par rapport au circuit sans chaîne de scan donne le gain de  $S$ . La fonction de majoration est donnée par la relation :  $\text{gain}(S \cup \{k\}) \leq \text{gain}(S) + p_k$ .

### **I.3.7 Analyse et résultats**

Le fait de partir du dessin des masques donne a priori des fonctions de coûts pour la surface et la vitesse les plus proches des coûts réels lors de l'implantation finale. Cependant les trois fonctions de coûts présentées sont très dépendantes du placement et du routage utilisés. Des permutations simples de portes peuvent facilement reporter les coûts de certaines bascules sur d'autres (par exemple pour les bascules dont la densité locale du canal est proche de la capacité du canal, ou pour les bascules qui se trouvent dans des rangées très pleines contenant peu de bascules, ou encore pour des bascules qui sont topologiquement éloignées des cellules qui y sont connectées sans pour autant se trouver sur un chemin critique), tout en conservant approximativement la surface occupée par le dessin et la même vitesse de fonctionnement. Il faudrait étudier la dispersion de ces coûts en fonction de différents placements et routages jugés équivalents.

Cela pourrait donner des résultats plus indépendants des placements et routages initiaux, et éviterait donc de donner par exemple un coût très important à une bascule qui est essentielle pour améliorer le critère de testabilité. Si on est dans des situations où le coût de chaque bascule est négligeable par rapport au coût maximum autorisé, ces phénomènes deviennent plus marginaux en ce sens qu'ils peuvent avoir un impact faible à cause d'un nivellement statistique. La pondération des coûts n'est pas forcément évidente à évaluer, et n'est pas explicitée. En effet un problème se pose automatiquement : comment comparer ce qui représente un chemin critique et ce qui représente une surface. Une façon plus naturelle de procéder consisterait à considérer un vecteur de coût dont les composantes sont les coûts partiels d'un certain type, et à prendre donc des limites pour chacune des composantes. Cela ne changerait pas les algorithmes, sauf pour déterminer si on est encore dans un domaine de coût autorisé. Au niveau du coût en surface, si aucun facteur de forme très strict n'intervient, alors le vrai coût est celui dû à l'accroissement global de la surface que cela provienne de l'allongement d'une rangée ou de l'élargissement des canaux par exemple. En prenant un coût moyen en surface par bascule, on conserve la simplicité des calculs de coût (coûts additifs et un coût par bascule). On pourrait par exemple prendre la surface ajoutée lorsque l'on met toutes les bascules en scan et diviser par le nombre de bascules du circuit. Si on divise encore par la surface initiale, et si on suppose que les rangées de cellules dans le dessin des masques ont à peu près toutes la même longueur, alors on peut considérer qu'on a une assez bonne approximation du coût en pourcentage de surface pour l'ajout d'une bascule. Cela justifie les nombreux algorithmes qui ne prennent en compte qu'un seul critère à la fois à savoir celui de la surface et qui attribuent implicitement un coût identique à chacune des bascules en prenant comme point d'arrêt une limite supérieure sur le nombre de bascules mises en scan.

Les calculs des coûts en surface effectués supposent que l'on ait déjà le dessin des masques, or cela fait en général partie des étapes finales de la conception. Pour les gros circuits, on travaille de manière hiérarchique, on peut avoir des préplacements de blocs et des estimations de place occupée. Etant donné les coûts des opérations de placement et de routage, il est difficilement envisageable de réaliser de nombreux essais de dessins des masques en fonction des choix des bascules à mettre en scan.

Dans cet algorithme un coût pour chaque bascule est calculé une seule fois au départ. Ces coûts sont additifs et sont sommés pour calculer le coût du circuit comportant des bascules en scan. Pour la testabilité, on calcule aussi la testabilité pour chaque bascule mais ces testabilités finalement varient en fonction de l'ensemble des bascules choisies pour être mises en scan. L'addition de ces mesures de testabilité donnent une mesure

globale de la testabilité du circuit. Ce sont ces deux valeurs globales qui sont alors utilisées comme critères de sélection des bascules. Cette approche par fonctions de coût a l'avantage de fournir une solution algorithmique générale pour réaliser des compromis et des choix entre les performances du circuit et sa testabilité. Les performances (vitesse, surface, ...) sont vues comme des contraintes du dessin dans un processus d'optimisation de critères de testabilité.

#### **I.4 Approche par sélections et combinaisons d'heuristiques**

Il existe de nombreuses autres façons de mesurer la testabilité d'un circuit indépendamment d'un ATPG. Chacune de ces mesures peut servir à sélectionner les bascules à mettre en scan. Il suffit par exemple de prendre une heuristique de type "glouton", qui consiste à sélectionner la bascule dont le gain en testabilité sera le plus fort étant donné l'ensemble courant des bascules déjà choisies pour la chaîne de scan. A chaque mesure correspond donc une heuristique, toutes ces mesures ne sont pas équivalentes, et donc on cherche à sélectionner la meilleure. La meilleure est celle qui offre le meilleur taux de couverture pour un pourcentage donné de bascules mises en scan. Dans [PRAD91], cinq heuristiques de base différentes (chacune étant liée à la topologie du circuit) sont présentées. Pradhan et al. ont combiné ces heuristiques pour améliorer leurs performances mesurées à l'aide d'un ATPG et d'un simulateur de fautes. Ensuite, de façon à obtenir une méthodologie automatique, simple et efficace de sélection des bascules à mettre en scan d'un circuit, ils ont proposé un algorithme de sélection parmi ces cinq heuristiques de base et trois de leurs combinaisons déjà validées pour leur performance.

##### **I.4.1 Les cinq heuristiques de base**

Les heuristiques de base génèrent une liste ordonnée de bascules à inclure dans la chaîne de scan. Elles utilisent les métriques suivantes :

- la sortance des bascules
- la distance d'une bascule aux entrées primaires
- la contrôlabilité des sorties d'une bascule
- l'observabilité des entrées d'une bascule
- le nombre de chemins de sortie d'une bascule qui reconvergent.

**L'heuristique de sortance** : elle donne une priorité aux bascules qui ont le plus grand nombre de portes connectées directement à sa (ou ses) sortie(s) appelé aussi sortance. Lors de la mise en scan d'une bascule, l'excitation et la sensibilisation pour la détection de fautes diverses dans la partie du circuit qui la suit, peuvent être accomplies facile-

ment. Donc, dans le processus de génération des vecteurs de test, le tracé arrière est requis seulement jusqu'aux entrées de type scan (les sorties des bascules mises en scan), au lieu d'aller jusqu'aux entrées primaires. Aussi, plus la sortance est grande, plus on pourra justifier un nombre de lignes importants via les entrées de scan. Le comptage de la sortance pour chaque bascule est obtenu en additionnant les sortances individuelles de ses nœuds de sortie. Ce comptage n'inclut pas bien sûr le nombre de branches de sortie qui sont des sorties primaires, ou qui alimentent directement des bascules déjà en scan (car elles sont déjà complètement observables). Par conséquent, les comptages sont mis à jour à chaque fois qu'une bascule est choisie pour la mise en scan, on réalise alors un nouvel ordonnancement des bascules.

**L'heuristique de distance** : elle est basée sur la distance de chaque nœud du circuit aux entrées primaires appelée aussi niveau. Pour un circuit combinatoire, les niveaux des nœuds peuvent être calculés comme suit : d'abord on assigne 0 aux entrées primaires, ensuite de proche en proche, aux sorties des portes on assigne 1 de plus que le niveau maximum sur les entrées de la porte considérée (remarque : pour une porte à une seule entrée on pourrait reporter la valeur en entrée). Ce processus a un sens et converge dans le cas où il n'y a pas de boucles combinatoires. Le niveau d'un nœud exprime alors le nombre maximum de portes traversées dans un chemin qui va d'une entrée primaire à ce nœud. Dans les circuits séquentiels, les niveaux des nœuds sont calculés en tenant compte du fait que les nœuds peuvent se trouver dans des boucles séquentielles ou être des sorties de bascules en scan. Le processus de calcul des niveaux commence par couper tous les nœuds de sortie de bascules en deux nœuds, à savoir le nœud de sortie lui-même et sa pseudo-entrée. Ils rendent compte respectivement pour l'état présent et pour l'état suivant. Après on calcule le niveau de chacun des nœuds en utilisant l'algorithme de calcul de niveau suivant :

**Etape 1** : initialiser les niveaux de toutes les entrées primaires et de toutes les pseudo-entrées à 0.

**Etape 2** : assigner les niveaux comme décrit au ci-auparavant pour les parties combinatoires jusqu'à atteindre les sorties primaires ou les sorties des bascules. Assigner le niveau du nœud de sortie des bascules à leur pseudo-entrée si la bascule n'est pas en scan, sinon assigner 0.

**Etape 3** : former une liste ordonnée par niveaux des bascules qui ne sont pas en scan.

**Etape 4** : comparer la liste ordonnée avec la liste générée dans l'itération précédente. Si elle est différente, assigner le niveau 0 aux entrées primaires et retenir les autres valeurs de niveaux et reprendre à l'étape 2, sinon c'est terminé.

Plus le niveau d'une sortie d'une bascule est élevé, plus grande sera la priorité de mise en scan. Ce processus assure que l'on assigne des valeurs plus élevées aux bascules qui sont dans des boucles. Une bascule qui reboucle directement sur elle-même a un niveau qui s'incrémente régulièrement lors de son calcul. Par contre, les niveaux des sorties des bascules qui ne sont dans aucune boucle vont être constants après la première affectation. Les bascules qui sont dans des boucles sont plus difficiles à appréhender pendant la génération des vecteurs de test que celles qui n'y sont pas. Après un nombre suffisant d'itérations, la liste ordonnée des bascules va se stabiliser. Ce processus d'attribution de niveaux et donc de priorités doit être répété à chaque fois que l'on met une bascule en scan, car la présence de bascules en scan intervient dans l'algorithme. En effet, les sorties de ces bascules agissent comme des entrées primaires et leurs entrées comme des sorties primaires, et donc elles doivent être traitées comme telles.

**Les heuristiques de contrôlabilité et d'observabilité :** elles sont basées sur le calcul des contrôlabilités et observabilités des nœuds données par l'algorithme STAFAN [JAIN84]. Comme vecteurs d'entrée de STAFAN, on utilise les vecteurs de test obtenus par l'ATPG séquentiel CONTEST [AGRA88] sur le circuit sans chaîne de scan ou avec du scan partiel. Les bascules dont les sorties sont difficiles à contrôler (faible contrôlabilité) et/ou dont les entrées sont difficiles à observer (faible observabilité) sont mises en priorité en scan. Le processus de sélection des bascules à insérer commence par le calcul des contrôlabilités des nœuds de sortie des bascules, et par le calcul des observabilités des nœuds d'entrée des bascules. Les bascules sont triées en fonction du minimum entre la contrôlabilité à 1 et 0 de leur sortie Q. La bascule dont l'une des contrôlabilités est la plus faible est mise en scan. De même les bascules dont l'observabilité est en dessous d'un certain seuil sont mises en scan. Pour la nouvelle itération de STAFAN, le circuit est modifié pour transformer les sorties de cette bascule en entrées primaires, et ses entrées en sorties primaires. Le processus continue jusqu'à obtenir le taux de couverture désiré.

**L'heuristique de sortance reconvergente :** elle choisit la bascule dont le nombre de chemins qui sortent de celle-ci et qui reconvergent est maximum, pour la mettre en scan. Ce nombre de chemins est appelé sa sortance reconvergente. Cette heuristique a pour but d'accroître la testabilité en cassant les chemins sortant des bascules et qui reconvergent.

#### **I.4.2 Les heuristiques de combinaison**

Les cinq heuristiques de base qui ont été présentées peuvent toutes être utilisées indépendamment ou en combinaison. Pradhan et al. ont réalisé un ensemble d'expérimen-



tations sur différents circuits de test. L'effet de la sélection des bascules sur la génération des vecteurs de test a été observé du point de vue de l'augmentation du taux de couverture, de la taille de l'ensemble des vecteurs de test et de la durée de génération des vecteurs de test. Il a été constaté que les heuristiques de sortance et de contrôlabilité étaient les plus efficaces. Par conséquent, des combinaisons de ces heuristiques incluant les deux plus performantes se sont imposées si l'on veut avoir une bonne stratégie de sélection. Dans une combinaison d'heuristiques, après mise à l'échelle des valeurs de priorité des heuristiques de la combinaison, on calcule pour chaque bascule, la somme pondérée  $H$  de ces valeurs.  $H$  devient alors la nouvelle valeur de priorité. L'utilisateur choisit un certain pourcentage de bascules à mettre en scan, et répète l'opération de calcul de  $H$  et de sélection pour chacune des insertions d'une bascule.  $H$  peut en effet varier beaucoup entre deux sélections car une bascule proche d'une bascule qui vient d'être sélectionnée a une priorité qui diminue fortement. Trois heuristiques de combinaisons ont été utilisées en raison de leur efficacité dans l'algorithme de sélection des heuristiques : X2, X3, X4.

X2 est la combinaison des heuristiques de distance, de contrôlabilité et d'observabilité. Elle donne un plus grand poids au critère observabilité quand la bascule a un niveau faible et un plus grand poids au critère contrôlabilité quand la bascule a un niveau élevé.  $H$  est donné par la formule :

$$H = \text{niveau} (1 - C_{\text{norm}}) + (\text{niveau\_maximum} - \text{niveau}) (1 - O_{\text{norm}})$$

où  $\text{niveau}$  vaut le niveau de la bascule considérée, et où  $\text{niveau\_maximum}$  vaut le niveau maximum de toutes les bascules non encore choisies dans le circuit donné.  $C_{\text{norm}}$  et  $O_{\text{norm}}$  désignent, pour une bascule, respectivement sa contrôlabilité normalisée et son observabilité normalisée.  $C_{\text{norm}}$  est obtenue en prenant le quotient de sa contrôlabilité et du maximum des valeurs des contrôlabilités de toutes les bascules non encore choisies dans le circuit donné.  $O_{\text{norm}}$  est obtenue en prenant le quotient de son observabilité et du maximum des valeurs des observabilités de toutes les bascules non encore choisies dans le circuit donné.

X3 est la combinaison des heuristiques de contrôlabilité et de sortance. Le poids  $W_{\text{sort}}$  est attribué à l'heuristique de sortance, et  $W_{\text{cont}}$  à l'heuristique de contrôlabilité. Sans études complémentaires sur la priorité à donner à l'une ou à l'autre de ces deux heuristiques, on prend  $W_{\text{cont}}$  et  $W_{\text{sort}}$  tous deux égaux à 0,5.  $H$  est donné par la formule :

$$H = W_{\text{sort}} (S_{\text{norm}}) + W_{\text{cont}} (1 - C_{\text{norm}})$$

où  $S_{\text{norm}}$  est, pour une bascule, sa sortance normalisée.  $S_{\text{norm}}$  est obtenue en prenant le quotient de sa sortance et du maximum des valeurs des sortances de toutes les bascules

non encore choisies dans le circuit donné.

X4 est la combinaison des heuristiques de contrôlabilité et de sortance reconvergente.

H est donné par la formule :

$$H = 0,5 (SR_{norm}) + 0,5 (1 - C_{norm})$$

où  $SR_{norm}$  est, pour une bascule, sa sortance reconvergente normalisée.  $SR_{norm}$  est obtenue en prenant le quotient de sa sortance reconvergente et du maximum des valeurs des sortances reconvergentes de toutes les bascules non encore choisies dans le circuit donné.

### **I.4.3 Algorithme de sélection des heuristiques**

Voici la procédure de sélection des heuristiques proposée par Pradhan :

**Si** “la plupart des bascules ont un niveau élevé”

**Alors**

{

**Si** “il y a des bascules dont la sortance reconvergente prédomine” ou “il y a des bascules dont la sortance prédomine”

**Alors**

{

**Si** “le nombre de bascules à être choisies < max(nombre de bascules dont la sortance reconvergente prédomine, nombre de bascules dont la sortance prédomine)”

**Alors**

{

**Si** “nombre de bascules dont la sortance reconvergente prédomine > nombre de bascules dont la sortance prédomine”

**Alors** utiliser l’heuristique de sortance reconvergente

**Sinon** utiliser l’heuristique de sortance

}

**Sinon**

{

Si "il y a un nombre important de bascules avec un niveau faible"

Alors utiliser l'heuristique X2

Sinon utiliser l'heuristique de contrôlabilité

}

}

**Sinon**

{

Si "il y a un nombre important de bascules avec un niveau faible"

Alors utiliser l'heuristique X2

Sinon utiliser les deux heuristiques X3 et X4

}

}

**Sinon** utiliser l'heuristique d'observabilité.

Explications :

"La plupart des bascules ont un niveau élevé" signifie que plus de la moitié des bascules ont un niveau supérieur à la moitié du niveau maximum de toutes les bascules. "La prédominance des sortances ou des sortances reconvergentes" peut être vue par exemple comme le fait que la sortance ou la sortance reconvergente soit supérieure à un certain seuil. "Un nombre important de bascules avec un niveau faible" peut se référer aussi à un certain seuil.

#### **I.4.4 Analyse et résultats**

Cet algorithme provient principalement de constatations expérimentales réalisées sur six "petits" circuits ayant 6 à 21 bascules internes. Les constatations tirées peuvent s'étayer par des considérations théoriques ou intuitives. Dans les expérimentations, à chaque fois qu'une bascule a été sélectionnée, des vecteurs de test ont été générés en utilisant CONTEST [AGRA88], et le taux de couverture, le nombre de vecteurs de test et le temps CPU utilisé ont été déterminés. L'effet d'une heuristique particulière ou d'une combinaison particulière des heuristiques dans le processus de sélection des bascules à mettre en scan a alors été testé. On est passé à un niveau de complexité supérieur dans l'automatisation par rapport à celui qui consiste à prendre une heuristique donnée

simple, c'est-à-dire que l'heuristique est maintenant choisie de manière algorithmique et le processus de sélection est donc plus complexe. Cette approche fait bien apparaître la difficulté d'avoir une méthodologie universelle efficace. En effet, on se rend compte déjà sur des circuits relativement petits qu'une mesure de testabilité n'est a priori pas privilégiée dans tous les cas par rapport à une autre. De plus même appliquée à un seul circuit, l'ordre des performances des heuristiques peut varier selon le pourcentage de bascules que l'on désire mettre en scan. Un autre point difficile à analyser est la mesure de la performance qui est basée sur un ATPG, lui-même basé sur des algorithmes non optimaux. Il pourrait donc aussi y avoir des phénomènes du type de compensation de défauts ; c'est-à-dire par exemple pour un générateur qui traiterai bien les problèmes de contrôlabilité et moins bien ceux d'observabilité, réduire en priorité le nombre de bascules mal observables serait plus profitable que réduire en priorité le nombre de bascules mal contrôlables.

### **I.5 Approche par testabilité empirique**

En général, les mesures de testabilité évaluent la capacité intrinsèque d'un circuit à être testé. Malheureusement, a priori un ATPG n'exploite pas complètement cette capacité intrinsèque pour générer les vecteurs de test. De même, il peut employer d'autres mesures ou méthodes qui lui sont propres. Donc si on utilise une mesure de testabilité particulière pour choisir un ensemble de bascules à mettre en scan, le taux de couverture final va dépendre de manière importante de l'ATPG utilisé. En prenant des mesures de testabilité liées aux performances d'un ATPG donné, on peut choisir des bascules de manière plus spécifique et liées aux résultats attendus lors de la génération des vecteurs de test par cet outil. Kim dans [KIM90] et [KIM93] propose donc pour la sélection des bascules à mettre en scan d'utiliser des mesures de testabilité empirique des nœuds du circuit, c'est-à-dire des mesures qui combinent à la fois les temps de convergence d'un générateur automatique de vecteurs de test et les mesures de testabilité de SCOAP. La sélection finale des bascules fait appel à un critère global sur le circuit appelé différence de testabilité.

#### **I.5.1 Différence de testabilité d'un circuit**

Soit un circuit séquentiel ayant  $n$  bascules et  $m$  fautes nommées respectivement  $FF_i$  à  $FF_n$  et  $FLT_1$  à  $FLT_m$ . Dans les mesures employées ici, plus les valeurs de contrôlabilité, d'observabilité et de testabilité sont petites, plus le test est facile.

La différence de testabilité pour une bascule donnée :  $TD(FF_i)$  mesure l'amélioration globale au circuit, en terme de testabilité, de la mise en scan de la bascule  $FF_i$ . Elle peut se décomposer en deux termes :  $SCD(FF_i)$  et  $SOD(FF_i)$ , qui représentent les améliora-

tions en terme de contrôlabilité et d'observabilité séquentielles.

On a donc :  $TD(FFi) = SCD(FFi) + SOD(FFi)$ .

$SCD(FFi)$  est la différence de contrôlabilité séquentielle associée à la bascule  $FFi$ . C'est une mesure globale (c'est-à-dire portant sur toutes les fautes considérées du circuit) qui comptabilise les différences de contrôlabilité séquentielle de chacune des fautes quand  $FFi$  n'est pas en scan et quand  $FFi$  est mise en scan. Soit  $FLTj$  une faute de collage, sa contrôlabilité séquentielle,  $SC(FLTj)$ , est définie comme sa contrôlabilité séquentielle à 1 pour le nœud correspondant si  $FLTj$  est un collage à 0, et comme sa contrôlabilité séquentielle à 0 pour le nœud correspondant si  $FLTj$  est un collage à 1.  $SCi(FLTj)$  est définie comme cette même contrôlabilité quand la bascule  $i$  a été mise en scan. Par définition, on a alors l'égalité :

$$SCD(FFi) = \sum_{j=1}^m \{ SC(FLTj) - SCi(FLTj) \}$$

De la même façon,  $SOD(FFi)$  est la différence de contrôlabilité séquentielle associée à la bascule  $FFi$ . C'est aussi une mesure globale qui comptabilise les différences d'observabilité séquentielle de chacune des fautes quand  $FFi$  n'est pas en scan et quand  $FFi$  est mise en scan. Soit  $FLTj$  une faute de collage, son observabilité séquentielle,  $SO(FLTj)$ , est définie comme son observabilité séquentielle à 1 pour le nœud correspondant.  $SOi(FLTj)$  est définie comme cette même observabilité quand la bascule  $i$  a été mise en scan. Par définition, on a alors l'égalité :

$$SOD(FFi) = \sum_{j=1}^m \{ SO(FLTj) - SOi(FLTj) \}$$

Le critère TD est déterminé à partir de mesures de testabilité : SC et SO, et sa signification et sa précision leur sont directement liées. Il permet d'extraire globalement une donnée de gain en testabilité pour un circuit. Dans le cas de l'emploi des mesures de testabilité empirique on parlera de différence de testabilité empirique ou ETD.

### **I.5.2 Mesures de testabilité empirique**

Les mesures de testabilité empirique sont des mesures de contrôlabilité et d'observabilité semblables à celles de SCOAP [GOLD79][GOLD80], mais qu'on lie à un simulateur de fautes par mise à l'échelle et combinaison du temps CPU pour la génération de séquences de test pour certaines fautes données. En fait, l'objectif est d'affiner les mesures de SCOAP non seulement pour avoir des données intrinsèques au circuit, mais aussi pour obtenir des données sur les difficultés de génération de vecteurs de test avec

l'ATPG utilisé. Dans SCOAP, trois mesures séquentielles dont deux de contrôlabilité et une d'observabilité sont définies pour un nœud du circuit : SC0, SC1, SO. Soit N un nœud, SC0(N) et SC1(N) mesurent respectivement le nombre minimum de coups d'horloges qu'il faut donner pour justifier la valeur 0 et 1 en N. Et SO(N) évalue le nombre minimum de coups d'horloge qu'il faut donner pour propager la valeur qui est en N. La méthode de calcul est présentée dans l'annexe C, page 197 : "Calcul des mesures de SCOAP".

Ces valeurs ne dépendent pas d'un ATPG particulier. La mesure du temps CPU pour la production des vecteurs de test pour une faute donnée estime, par rapport à l'outil utilisé, la difficulté à générer ses vecteurs de test. La testabilité empirique prend en compte ces deux types de mesures. Pour chaque nœud des nouvelles définitions de SC0, SC1 et SO sont calculées. La méthode de calcul utilisée commence par la mesure empirique pour chaque bascule d'entrée D et de sortie Q, du temps CPU nécessaire pour avoir une séquence de test pour :

- D collé à 1 et D vu comme une sortie primaire
- D collé à 0 et D vu comme une sortie primaire
- Q collé à 0 ou 1 et Q vu comme une entrée primaire.

Dans le cas où il n'y a pas de résultat, on attribue temporairement à ces mesures la valeur -1, et on dit alors qu'elles sont inconnues. Ces mesures vont être par la suite approximées. Pour les nœuds combinatoires, et les nœuds des bascules de mesures inconnues, on calcule ces valeurs comme dans SCOAP après avoir rendu compatibles les valeurs données par les temps CPU et les valeurs séquentielles de SCOAP. La compatibilité des temps CPU est obtenue par leur mise à l'échelle de manière linéaire. On procède afin que la plus grande et la plus petite valeur calculée par SCOAP pour les valeurs positives des mesures précédentes soit respectivement associée à la faute de plus grand et de plus petit temps CPU de la mesure empirique. Les valeurs positives données par l'ATPG remplacent ensuite celles données par SCOAP.

La détermination des valeurs de testabilité empirique pour chaque bascule candidate pour la mise en scan est trop coûteuse en terme de temps CPU pour être réalisée. Donc, on traite l'effet de la mise en scan d'une bascule à partir de l'estimation des modifications de ces mesures de testabilité obtenues pour le circuit dont les chaînes de scan ne contiennent pas cette bascule. La mise en scan d'une bascule se traduit par le fait que les contrôlabilités de sa ou ses sorties deviennent nulles ainsi que l'observabilité de son entrée. Ces valeurs sont alors propagées aux nœuds combinatoires, et aux nœuds des bascules de mesures inconnues. L'ajustement des contrôlabilités des sorties des bascu-

les obtenues directement par mesure empirique se fait en multipliant l'ancienne valeur de contrôlabilité de la sortie par le rapport de la nouvelle valeur sur l'ancienne valeur de la contrôlabilité de l'entrée correspondante. On procède de façon similaire pour l'observabilité. Ainsi, pour toute bascule donnée, on a des mesures de testabilité empirique pour le circuit ayant ou non cette bascule dans une chaîne de scan.

### **I.5.3 Algorithme par différence de testabilité empirique**

La sélection des bascules se fait à partir de la différence de testabilité dérivée des mesures de testabilité empirique après suppression des fautes déjà couvertes par des vecteurs de test fonctionnels. Les étapes sont les suivantes :

**Etape 1** : on évalue les contrôlabilités séquentielles à 0 et à 1, et les observabilités séquentielles de toutes les bascules qui ne sont pas déjà dans une chaîne de scan. On utilise pour ce faire un ATPG.

**Etape 2** : à partir des testabilités séquentielles calculées dans l'étape 1, on utilise une approche de type SCOAP pour évaluer les contrôlabilités et observabilités séquentielles de toutes les fautes dans la liste de fautes courante.

**Etape 3** : calcul de l'ETD.

**Etape 4** : on sélectionne la bascule dont l'ETD est le plus grand pour qu'elle soit mise en scan et on modifie le circuit en conséquence.

**Etape 5** : on répète les étapes 1 à 4 jusqu'à ce qu'une des conditions de sortie décrites ci-après soit satisfaite.

**Etape 6** : on applique l'ATPG pour toutes les fautes non détectées. Les entrées et les sorties des bascules mises en scan sont traitées respectivement comme des sorties et des entrées primaires pour l'outil de génération.

**Etape 7** : si on n'obtient pas un taux adéquat de couverture de fautes, alors on alterne la sélection d'autres bascules à mettre nécessairement en scan avec l'application de l'ATPG sur les fautes restantes non détectées. Cette étape est expliquée plus loin dans ce paragraphe.

Voici la liste des conditions de sortie pour passer à l'étape 6 :

**S1** : la limite, fixée par l'utilisateur, du nombre maximum de bascules à mettre en scan est atteinte.

**S2** : les valeurs de ETD pour toutes les bascules du circuit sont passées sous un seuil donné.

**S3** : les valeurs de testabilité pour un pourcentage spécifié (généralement élevé) de nœuds du circuit est passé sous un seuil donné.

**S4** : un pourcentage spécifié (généralement élevé) des valeurs de testabilité pour les bascules peut être généré de manière empirique.

Après l'exécution de l'étape 6, il reste une liste de fautes difficiles à détecter qui peut servir d'ensemble restreint de fautes. On met alors l'accent sur cet ensemble de fautes au lieu de prendre l'ensemble complet des fautes restantes. En sortie de l'étape 6, on calcule désormais ETD sur l'ensemble restreint et on reprend les étapes 1 à 4. Puis l'ATPG est appliqué sur ce même ensemble. Ceci est répété jusqu'à ce que soit atteint soit le taux de couverture de fautes désiré, soit la limite fixée par l'utilisateur du nombre maximum de bascules de scan. On notera que dans ce processus final on utilise intensivement l'ATPG entre chaque sélection de bascule à mettre en scan. Le temps CPU global va donc augmenter de manière significative, mais c'est le prix à payer pour l'amélioration de la sélection des bascules à insérer.

#### **I.5.4 Algorithme par différence de testabilité empirique renversée**

L'approche précédente peut être qualifiée d'approche par ajout, car on sélectionne la bascule à ajouter dans la chaîne de scan. Une autre façon de procéder consiste à calculer une différence de testabilité empirique pour la suppression d'une bascule de la chaîne de scan, et à sélectionner celle qui a l'effet minimum sur la testabilité globale du circuit. Cette approche est qualifiée d'approche renversée ou par suppression. L'algorithme global est alors :

**Etape 1** : commencer avec un circuit en scan complet et réaliser l'évaluation empirique pour la suppression de chaque bascule individuelle de la chaîne de scan.

**Etape 2** : sélectionner une ou plusieurs bascules ; celles qui ont le moins d'intérêt pour la mise en scan et les supprimer de l'ensemble des points de scan.

**Etape 3** : répéter l'étape 2 jusqu'à ce que les conditions de changement de phase décrites plus loin soient satisfaites.

**Etape 4** : utiliser l'ATPG sur l'ensemble réduit de fautes. Si le taux de couverture désiré n'est pas atteint, sélectionner et ajouter des bascules de scan en utilisant cet ensemble réduit de fautes jusqu'à parvenir au taux de couverture désiré ou jusqu'à atteindre le nombre maximum de bascules de scan autorisées.

Pour les conditions de changement de phase, des critères similaires à ceux de l'approche par ajout sont utilisés, en particulier, l'étape 2 est répétée jusqu'à obtenir moins de bascules mises en scan que le nombre maximum de bascules autorisées par l'utilisateur.



Pour accélérer le processus de sélection, on peut traiter les bascules par groupes, c'est-à-dire au lieu de supprimer une seule bascule à chaque itération, on supprime aussi les bascules qui lui sont "proches".

### **I.5.5 Accélération par sélection multiple de bascules**

La sélection multiple de bascules est basée sur des distances entre les bascules. On définit une observabilité séquentielle relative (RSO) d'une bascule "j" à une bascule "i" comme l'observabilité séquentielle de la sortie de la bascule "j" vue à l'entrée de la bascule "i". Son calcul se fait simultanément pour l'ensemble des bascules différentes de la bascule "i". On commence par assigner d'une part 0 pour la valeur de l'observabilité séquentielle de l'entrée de la bascule "i" et l'infini pour les valeurs de l'observabilité séquentielle de toutes les sorties primaires. On calcule ensuite les observabilités séquentielles pour les autres portes du circuit en utilisant les mêmes formules locales que celles de SCOAP. Des mesures empiriques peuvent aussi être adoptées. La RSO calculée est un moyen de mesurer combien deux bascules sont proches l'une de l'autre. Cette notion de proximité traduit avec quelle facilité une bascule permet d'observer une autre bascule, qui en retour contrôle implicitement la première. Le calcul des valeurs de RSO pour chacun des couples de bascules pouvant être très coûteux en temps CPU, on commence par calculer les valeurs des RSO de base, c'est-à-dire entre deux bascules adjacentes (des bascules qui sont connectées directement par un chemin purement combinatoire). Une fois ces valeurs de base calculées, on construit un graphe dont les sommets sont les bascules et dont les arcs relient les bascules adjacentes. A ces arcs sont associées des poids : les valeurs de RSO calculées précédemment. La valeur de RSO pour un couple de bascules, est alors la somme minimale des poids des arcs des chemins reliant ces deux bascules. Ce qui est, en fait recherché pour une bascule ce sont seulement les bascules qui lui sont proches. Donc pour l'efficacité de l'algorithme, partant d'une bascule, on recherche seulement son nuage d'influence (voir annexe F, page 211 : "Algorithme de génération du nuage d'influence"), c'est-à-dire l'ensemble des bascules de distance en termes de RSO inférieure à un certain seuil. Dans l'approche renversée, à chaque itération, la bascule la plus indésirable est sélectionnée. Elle est alors exclue des chaînes de scan ainsi que toutes celles de son nuage d'influence que l'on calcule. On continue jusqu'à atteindre le nombre maximum de bascules à être mises en scan ou jusqu'à recouvrir toutes les bascules restantes par le dernier nuage d'influence.

### **I.5.6 Analyse et résultats**

L'intérêt principal de l'utilisation de mesures de testabilité empirique par rapport aux

autres mesures de testabilité classiques est l'amélioration des performances, mais à un coût qui peut être important en temps CPU. En effet, ces mesures prennent en compte les spécificités de l'ATPG utilisé pour la génération des vecteurs de test. Ce lien entre la sélection des bascules et l'ATPG utilisé est très important. En effet, la testabilité d'une faute peut être évaluée bonne, mais la génération de la séquence de test par un ATPG donné peut être très coûteuse. Donc si les mesures de testabilité classiques donnent une approximation du caractère détectable d'une faute, les mesures de testabilité empirique rapportent cette mesure à un ATPG donné, et c'est cela qui est finalement utile. La méthode par approche renversée possède un certain nombre d'avantages par rapport à l'approche par ajout. En effet, la génération des vecteurs de test est initialement beaucoup plus facile et utilise moins de temps CPU puisque la plupart des bascules sont déjà en scan. Donc, pour certains circuits difficiles à tester, cela prend moins de temps CPU puisque la sélection initiale du circuit en scan complet est plus proche de la solution finale que la sélection initiale sans bascules en scan. Dans le cas de circuits comportant de nombreuses bascules, une méthode de sélection multiple est utilisable dans les deux phases de suppression et d'ajout de bascules. Elle permet de réduire les coûts de recalcul des testabilités. De plus, la valeur du seuil pour les nuages peut être ajustée en fonction de l'ATPG employé. En effet, pour un ATPG, on détermine, de manière empirique ou à partir de ses algorithmes, le nombre de coups d'horloge qu'il autorise dans le calcul des séquences de test. Le seuil s'obtient alors en rapportant ce nombre aux mesures de RSO qui représentent aussi un nombre de coups d'horloge.

Des comparaisons ont été faites avec des circuits tirés de ISCAS89 [BRGL89] entre les méthodes de testabilité empirique (renversée et non renversée), SCOAP et un algorithme de suppression de cycles et de diminution de la profondeur séquentielle tiré de [LEED90]. Les résultats restent du même ordre de grandeur. Leur mise en œuvre souligne les différents facteurs qui influent sur la qualité finale du résultat et les difficultés de comparaison : les ATPGs, les choix des conditions d'arrêt et des options dans les algorithmes, les limitations dues au temps CPU ou à des capacités de mémoire des machines. Dans [PARI93], un algorithme de structure tout à fait semblable qui peut servir de compromis temps CPU/taux de couverture, est présenté. Il emploie une mesure de testabilité appelée CoPS qui ne nécessite pas d'ATPG. Les valeurs des mesures se calculent de façon semblable à celles de SCOAP, mais avec des règles locales de calcul de contrôlabilité et d'observabilité différentes. L'objectif est de mieux représenter le nombre de coups d'horloge nécessaires pour contrôler ou observer une bascule. Cet algorithme permet aussi de tenir compte de fautes déjà détectées.

## II. Approches utilisant des générateurs automatiques de vecteurs de test

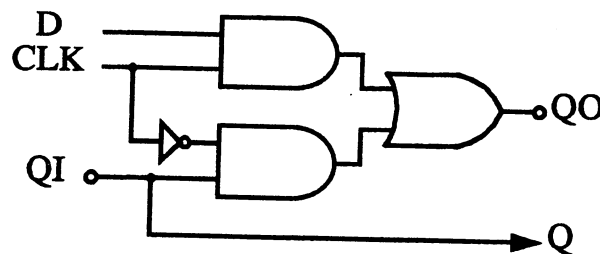
### II.1 Introduction

On a vu précédemment que les ATPGs pouvaient être utilisés pour faire des mesures empiriques. N'importe quel type d'ATPG pouvait a priori être considéré. Les méthodes précédentes sont des méthodes de sélection de bascules à mettre en scan qui ont été modifiées pour exploiter les résultats d'ATPGs. Des approches spécifiques à des ATPGs ont été aussi proposées. Ces méthodes utilisent les spécificités des résultats de certains ATPGs. En ce sens, on peut considérer que ces méthodes sont des ATPGs modifiés pour la sélection des bascules à mettre en scan. Au II.2 et au II.3 des ATPGs combinatoires sont utilisés, et au II.4 un ATPG séquentiel est utilisé.

### II.2 Approche par calcul des non contrôlabilités et des non observabilités

#### II.2.1 Description

Abramovici et al. ont proposé, dans [ABRA91], un algorithme d'analyse de scan partiel appelé PASCAL (PARTIAL SCAN ANALYSIS). PASCAL part d'un circuit séquentiel, il considère d'abord ce circuit avec toutes les bascules mises en scan et remplace alors le modèle séquentiel des bascules de scan par un modèle combinatoire. Ainsi, il peut cal-



**Figure 1** : modèle combinatoire de la bascule de scan

culer sur ce modèle les fautes qui ne sont plus détectables lorsqu'on supprime une ou plusieurs bascules de la chaîne de scan. Par exemple, pour une bascule de type D, on ne tient pas compte des signaux mise à 1 ou à zéro qui peuvent exister. D et CLK sont respectivement les entrées de la donnée et de l'horloge. Q est la sortie de la bascule. QI est une entrée primaire supplémentaire qui représente la valeur d'entrée amenée dans le mode de scan. QO est une sortie primaire supplémentaire qui représente la valeur de sortie récupérée dans le mode de scan. Ce modèle peut être vu de la façon suivante :

- La sortie Q reçoit la valeur qui entre en mode scan ( $Q = QI$ ).
- Une horloge active ( $CLK = 1$ ) capture la valeur de D pour l'amener sur la chaîne de scan ( $QO = D$ ).

- Une horloge inactive ( $CLK=0$ ) propage la valeur QI d'entrée de scan sur la chaîne de scan ( $QO = QI$ ).

On notera que ce modèle permet l'utilisation de logique de contrôle sur l'horloge.

Supprimer une bascule de la chaîne de scan signifie que son port QI devient non contrôlable et que son port QO devient non observable, et que certaines fautes deviennent par conséquent non détectables. Le but est alors de déterminer le nombre de ces fautes qui deviennent non détectables. Ce calcul s'effectue par propagation des non contrôlabilités et des non observabilités lorsque des points d'observation et de contrôle sont supprimés. Les principes de ce calcul sont présentés dans l'annexe G, page 215: "Calcul des non détectabilités".

Finalement, les bascules sont sélectionnées par un algorithme de type "glouton". A chaque étape on détermine quelle bascule choisie parmi celles qui sont encore en scan, donnera la plus faible augmentation du nombre de fautes non détectable quand elle sera mise hors de la chaîne de scan. Cette bascule est alors choisie pour être sortie de la chaîne de scan. On continue ainsi jusqu'à épuiser toutes les bascules du circuit.

### **II.2.2 Analyse et résultats**

L'algorithme PASCAL reflète en terme de taux de couverture maximum envisageable (calculé à partir du nombre de fautes non détectables) l'effet cumulatif de la suppression de bascules d'une chaîne de scan. Sa complexité relativement faible, en  $O(n^2)$ , où  $n$  est le nombre de bascules, permet d'appliquer PASCAL à de gros circuits. PASCAL permet avant tout d'avoir une vision globale de l'effet du scan partiel sur le taux de couverture final. Deux ajouts simples permettent d'accroître la précision des résultats. Le premier consiste à supprimer de la liste des fautes, dès le départ, les fautes couvertes par des tests fonctionnels. Le deuxième consiste à supprimer aussi de la liste des fautes, dès le départ, les fautes prouvées combinatoirement non détectables. De par sa conception, pour déterminer les fautes qui ne sont pas détectables, PASCAL ne prend pas en compte les phénomènes séquentiels du circuit autres que les décalages dans les registres de scan. Il ignore totalement les possibilités de justification et de propagation par des bascules qui ne seraient pas en scan.

## **II.3 Approche en fréquence et en distance**

### **II.3.1 Description**

Agrawal et al. dans [AGRA87] proposent deux approches pour l'inclusion de bascules de scan : une approche en distance et une approche en fréquence qui toutes deux partent

d'algorithmes de type PODEM [GOEL81]. La première étape consiste toujours à éliminer les fautes qui sont déjà détectées par des vecteurs fonctionnels.

Dans l'approche en fréquence, à chaque faute restante, on considère la partie combinatoire où elle est impliquée, et on recherche l'ensemble des vecteurs de test combinatoires qui la détectent. Le résultat est un tableau d'options qui donne pour chaque faute les différents vecteurs de test possibles et les bascules à utiliser dans chaque cas considéré. Cette méthode nécessite une amélioration de l'algorithme de PODEM de façon à générer différents vecteurs de test pour une même faute. C'est pourquoi afin d'éviter ce coût important de calcul de vecteurs de test, une autre approche utilisant une notion de distance est aussi proposée. Dans ce cas on ne génère qu'un seul vecteur de test par faute considérée en utilisant toujours un algorithme de type PODEM. Cette méthode intervient en modifiant la notion de distance utilisée dans PODEM. La distance dans PODEM est le nombre de portes sur le chemin de propagation ou de justification, qui est pris comme objectif. Pour les chemins dont l'entrée part d'une sortie de bascule ou dont la sortie se termine sur une entrée de bascule, on rajoute une grande valeur : 100. On ne rajoute rien si on aboutit à une entrée primaire ou à une sortie primaire. Comme pour l'approche en fréquence, un tableau d'options est aussi généré de la manière suivante. PODEM modifié est appliqué sur la partie combinatoire pour chacune des fautes non encore détectées. A chaque étape, une simulation de fautes est réalisée avec le vecteur de test trouvé, afin de supprimer toutes les fautes qu'il couvre de la liste des fautes encore à traiter.

Une fois le tableau d'options obtenu, le problème de sélection se résume alors à un problème de minimisation qui est NP-complet. Une méthode simple est proposée pour choisir dans un certain ordre de priorité les bascules du tableau.

**Etape 1 :** considérer les fautes pour lesquelles il n'y a qu'un vecteur de test, et mettre en scan les bascules associées à ce vecteur.

**Etape 2 :** éliminer les fautes qui peuvent être couvertes par cet ensemble de bascules. Estimer le taux de couverture comme dans l'annexe J, page 227 : "Estimation du taux de couverture pour les approches en fréquence et en distance" (cette estimation n'utilise pas de simulateur de fautes).

**Etape 3 :** pour les fautes restantes, estimer la fréquence d'usage de chaque bascule. la fréquence d'une bascule est le nombre de fautes qui ont au moins un vecteur de test qui utilise cette bascule.

**Etape 4 :** ajouter la bascule ayant la plus grande fréquence et retourner à l'étape 2.

Le résultat final est un taux de couverture estimé en fonction du nombre de bascules mises en scan. On remarquera que la formule d'estimation qui est employée est toujours pessimiste, car elle ne prend pas en compte du test aléatoire introduit par les vecteurs de test utilisés.

### **II.3.2 Analyse et résultats**

D'après leurs résultats expérimentaux sur de vrais circuits dont l'un de plus de 13000 portes logiques, les auteurs estiment que en incluant moins de 65% des bascules dans la chaîne de scan, on peut facilement obtenir des taux de couverture supérieurs à 95%. Sur les plus gros circuits, ils ont constaté que par rapport au scan complet les temps d'application des vecteurs de test étaient réduits de manière significative (plus de la moitié en se plaçant dans une optique de 95% de taux de couverture), et que dans l'approche en distance, l'ordre dans lequel sont choisies les fautes influe peu sur les résultats. On tirera d'autant plus profit de ces méthodes de scan partiel que les vecteurs de test fonctionnels sont eux-mêmes bons.

## **II.4 Approche par énumération d'états et par utilisation d'un graphe de transitions**

### **II.4.1 Principes**

Ma dans [MA88] propose pour la sélection des bascules à mettre en scan d'utiliser un ATPG nommé STALLION [MA87]. Cette sélection se fait alors en prenant un ensemble minimum de bascules dont le générateur aurait eu besoin de contrôler ou d'observer pour améliorer le taux de couverture. STALLION considère le circuit comme une seule machine à états finis, et utilise alors un graphe partiel de transition d'états, STG (state transition graph), qu'il extrait à partir d'une énumération d'états. Cet ATPG utilise une extension de l'algorithme implicite d'énumération de PODEM [GOEL81]. Les principes des algorithmes utilisés dans STALLION sont décrits dans l'annexe H, page 219: "Principes de l'ATPG STALLION". La méthode de sélection des bascules est en fait dérivée des méthodes et des résultats de STALLION.

### **II.4.2 Sélection des bascules à mettre en scan**

Etant donné un circuit  $S$ , et un ensemble de fautes  $F$  à détecter, on recherche un ensemble  $M$  de bascules à mettre dans la chaîne de scan pour que STALLION détecte les fautes dans  $F$ .

Etape 1 : pour chaque faute  $f$  de  $F$ , grâce à STALLION, rechercher un état initial  $S0_f$  et une séquence de propagation  $T1_f$ . Pour les fautes  $f$  qui ne sont pas propagées aux sor-

ties primaires, on définit  $NS_f$ , l'ensemble des lignes d'état qui propagent l'effet de la faute  $f$  sur l'état suivant. L'ensemble de ces fautes est appelé  $F^{NS}$  : l'ensemble des fautes qui ne peuvent être propagées sur les sorties primaires mais seulement sur les états suivants. Les états  $SO_f$  peuvent contenir des lignes d'états de valeurs indifférentes.

**Etape 2** : soit  $K$  l'ensemble des vecteurs  $SO_f$  pour tout  $f$ .

**Etape 3** : grâce au STG, générer pour  $i$  variant de 1 à  $NIVEAU\_MAX$  les ensembles d'états  $Q_i$  atteignables à partir de l'état de remise à zéro en  $i$  coups d'horloge. On limite le nombre d'états dans  $Q_i$  par la variable  $ETATS\_MAX$ .

**Etape 4** : pour chaque  $k$  de  $K$  et chaque état  $q$  de  $Q_i$ , trouver les bascules à mettre en scan afin de passer de l'état  $q$  à l'état  $k$ . Pour chaque  $k$  de  $K$ , générer  $CHOIX\_MAX$ , les meilleurs choix (le moins de lignes d'états) pour les ensembles de lignes qui permettent de couvrir  $f$ .

**Etape 5** : étant donnés  $K$ , les choix  $CHOIX\_MAX$  d'ensembles de lignes pour chaque  $k$  de  $K$ , et l'ensemble des lignes d'états  $NS_f$  pour chaque  $f$  de  $FNS$ , sélectionner un ensemble de lignes pour chaque  $k$  et une seule ligne tirée de  $NS_f$  de façon à minimiser le nombre de lignes d'état distinctes devant être mises en scan. Deux algorithmes de sélection sont détaillés dans l'annexe I, page 223 : "Algorithmes de sélection des lignes d'état"

### **II.4.3 Analyse et résultats**

Les algorithmes de sélection des lignes d'états permettent de faire des compromis sur sa complexité lors de la résolution. En effet, l'algorithme de type "glouton" utilisé est paramétrable. On peut soit faire des explorations exhaustives pour un seul choix, ou faire des explorations exhaustives sur plusieurs choix consécutifs. Le résultat sera potentiellement d'autant meilleur que la profondeur de l'espace d'exploration est élevée. A l'extrême, si cette profondeur correspond au nombre de fautes considérées, alors on obtient une solution optimale, mais aussi un coût en temps CPU maximum. Les auteurs préconisent comme compromis une profondeur égale à 3, afin notamment d'être proche de la solution optimale. On remarquera aussi que cette méthode est difficile à employer pour les gros circuits.

### **III. Approches basées sur le graphe séquentiel ou topologique du circuit**

#### **III.1 Introduction**

Ces approches sont uniquement basées sur des propriétés structurelles du circuit, elles ne tiennent pas compte des fonctions des blocs combinatoires, si ce n'est initialement pour déterminer des relations de dépendance, entre les sorties et les entrées de bascules.

On présente dans un premier temps au III.2, une méthode partant d'une description RTL. Elle est basée sur la notion d'équilibrage de graphe. Puis, dans un deuxième temps au III.3, on présente des méthodes travaillant en général au niveau bascule. Elle sont basées sur des notions de coupure de cycles et de diminution de la profondeur séquentielle d'un graphe.

#### **III.2 Equilibrage des chemins séquentiels : BALLAST**

##### **III.2.1 Description**

Une méthode de scan partiel nommée BALLAST (BALANCED Structure Scan Test) a été proposée dans [GUPS89]. Elle modélise un circuit à l'aide de son graphe topologique défini ci-après.

Les sommets (ou nœuds) du graphe, appelés aussi nuages, s'obtiennent en utilisant les règles de construction suivantes : A et B, 2 blocs combinatoires sont placés dans le même nuage si l'un précède directement l'autre, ou si l'un est précédé directement par la sortie Q d'une bascule et l'autre par la sortie  $\bar{Q}$  de la même bascule, ou encore s'ils ont un signal d'entrée commun. On considère des nuages vides qui ne contiennent pas de logique combinatoire. Dans ce cas, il s'agit soit de la connexion d'une sortie de registre à une entrée de registre, soit de la connexion d'une entrée primaire à une entrée de registre, soit de la connexion d'une sortie de registre à une sortie primaire. Les arcs du graphe sont donnés par les registres qui relient deux nuages entre eux. Si on considère le graphe adjoint (ou graphe aux arêtes) du graphe topologique, on obtient le graphe séquentiel où les bascules ont été regroupées en registres.

BALLAST s'intéresse au graphe topologique du noyau d'un circuit. Le noyau d'un circuit, est ce qui reste du circuit quand on a supprimé les registres en scan et les parties combinatoires entourées exclusivement de registres en scan ou d'entrées et sorties primaires. BALLAST cherche à obtenir des noyaux dont le graphe topologique est équilibré. Un graphe est équilibré<sup>3</sup> si tous les chemins qui ont la même origine et la même extrémité, ont la même longueur. Cela implique en particulier que le graphe est acyclique. La propriété fondamentale associée à un noyau équilibré d'un circuit (dont le



graphe topologique associé est équilibré) et démontrée formellement dans [GUPS88], est la suivante :

toutes les fautes séquentiellement détectables de ce noyau peuvent être détectées en appliquant sur plusieurs coups d'horloge des vecteurs de test de l'équivalent combinatoire du noyau. La durée de maintien de ces vecteurs de test est donnée par la profondeur séquentielle du graphe topologique du noyau. L'équivalent combinatoire du noyau s'obtient en remplaçant les bascules du noyau par un fil entre la sortie Q et l'entrée D et par un inverseur entre la sortie  $\bar{Q}$  et l'entrée D. Le maintien des vecteurs de test en entrée du noyau se fait en ajoutant des modes de maintien sur les registres en scan connectés aux entrées du noyau. On peut résumer la méthodologie BALLAST de la façon suivante :

**Etape 1 :** construire le graphe topologique G du circuit.

**Etape 2 :** sélectionner un ensemble minimal d'arcs R, à enlever de G de façon à ce que le graphe topologique restant soit équilibré. Ce graphe représente alors le noyau du circuit.

**Etape 3 :** déterminer l'équivalent combinatoire du noyau du circuit, par ATPG combinatoire classique déterminer un ensemble de vecteurs de test

**Etape 4 :** construire le chemin de scan à partir de R, et ajouter des modes de maintien.

La sélection de R peut se faire en cherchant à minimiser une fonction de coût comme la somme des tailles de registres de R (la taille d'un registre est le nombre de bascules qui le composent). Les heuristiques proposées par Gupta font appel en particulier à des algorithmes de flot maximum (à chaque arc est associée une capacité, la taille du registre correspondant). La vérification qu'un graphe est équilibré est de complexité  $O(n)$ , où n est la somme du nombre d'arcs et de sommets du graphe.

### III.2.2 Analyse et résultats

Gupta, dans [GUPS90], propose de supprimer certains modes de maintien que l'on avait ajoutés auparavant. Pour cela, les longueurs des chemins entre les sources et les puits du graphe équilibré sont analysés. Ainsi, les bascules qui nécessitent vraiment un mode de maintien sont déterminées. Un classement fournit aussi l'ordre dans lequel il faut mettre les registres dans la chaîne de scan. Cette technique influe aussi sur la façon d'appliquer la séquence de test en entrée de la chaîne de scan, car il faut alors ajouter

<sup>3</sup>. On pourrait aussi tenir compte d'un ensemble particulier H de sommets qui correspondrait à des bascules qui ont dès le départ un mode de maintien. Mais cela rendrait les explications plus complexes. Gupta parle aussi de B-structure pour les graphes équilibrés.

dans la séquence des bits fantômes de retardement. Le but de cette technique est de diminuer les coûts en surface. Dans le cas où le mode de maintien est réalisé par de la logique intervenant sur l'horloge, en supprimant ces modes de maintien, on supprime des problèmes de routage et de gigue. BALLAST réduit le temps de test quand la diminution du nombre de cycles par vecteur de test est, en proportion, plus importante que le nombre de vecteurs de test.

Mais cependant, des méthodes de diminution du temps de test basées sur le séquençage optimum des registres dans les chaînes de scan, comme celle proposée dans [NARA92a], ne sont plus directement applicables. En effet, il faut tenir compte de l'ordre préfixé des registres pour supprimer les modes de maintien.

BALLAST ne dépend que des propriétés structurelles du circuit, il ne tient pas compte des fonctions des blocs combinatoires. Il est intéressant pour les circuits pipelinés avec un nombre limité de cycles et de reconvergences. Il est peu efficace quand il y a peu de nuages ou quand chaque élément de stockage possède une boucle combinatoire sur lui-même. Quand il n'y a qu'un seul nuage BALLAST se réduit à du scan complet. En revanche, BALLAST assure un taux de couverture de 100% (pour les fautes séquentiellement détectables) et ne requiert que l'utilisation d'un ATPG combinatoire.

D'autres développements autour de BALLAST ont été réalisés. ACYST, une amélioration de BALLAST, analyse plus finement les déséquilibres des chemins et permet de réduire le nombre de registres mis en scan. Il fournit toujours des noyaux dont le graphe est acyclique, mais plus forcément équilibré. Ces outils ainsi que celui précédemment cité pour la réduction des temps de test ([NARA92a]) ont été intégrés dans un même système appelé SIESTA [NARA92b].

### **III.3 Suppression de cycles ou diminution de la profondeur séquentielle**

#### **III.3.1 Suppression des cycles les plus longs**

#### **III.3.2 Description**

Le problème qui consiste à trouver un ensemble minimum de sommets qui lorsqu'il est supprimé rend le graphe acyclique, est un problème NP-complet.

De nombreuses méthodes partant du graphe séquentiel ont été proposées. Cheng dans [CHENG89] propose de supprimer les cycles de longueur supérieure à 2 (inclus). L'algorithme commence par supprimer les arcs qui correspondent à une boucle d'un sommet sur lui-même. Il sélectionne ensuite un sommet qui appartient au plus grand nombre de cycles de longueur supérieure à 2, l'ajoute à l'ensemble des sommets déjà trouvé, et répète l'opération jusqu'à ce que le graphe soit acyclique. Le calcul pour cha-

que sommet du nombre de cycles auxquels il appartient est très coûteux. C'est pourquoi, seulement un nombre fixé de cycles sont énumérés pour faire un compromis potentiel sur la qualité finale de la solution. Il se peut que dans la sélection finale certains sommets soient inutiles parce qu'ils appartiennent à des cycles qui sont cassés par des sommets choisis ultérieurement. Donc une amélioration, non proposée par Cheng, consiste à scruter la sélection finale pour détecter si le retrait d'un sommet de l'ensemble solution ne donne pas aussi une solution.

De même dans le système de génération de vecteurs de test, PASCANT [BHAW91], un algorithme de mise de bascules en scan est implanté. Son premier objectif est aussi de casser les cycles de longueur supérieure à 2. Au lieu de faire une énumération des cycles, une estimation du sommet par où passe le maximum de cycles est faite. Pour chaque sommet dans une composante fortement connexe du graphe, trois formules d'estimation simples sont proposées pour un sommet. Elles ne dépendent que du demi degré interne et du demi degré externe de chaque sommet. Ce sont soit leur valeur maximale, soit leur somme, soit leur produit. A chaque étape, les sommets qui maximisent cette valeur sont pris, et le graphe est redécomposé en ses composantes fortement connexes. L'opération continue jusqu'à ce qu'on obtienne que des composantes fortement connexes à un seul sommet. Le deuxième objectif est de sélectionner les sommets de façon à minimiser la longueur des chemins élémentaires comportant uniquement des sommets appartenant à des cycles de longueur 1. Pour atteindre plus rapidement le premier objectif, Bhawmik propose aussi de sélectionner dès le départ les sommets de demi degré extérieur les plus élevés.

Dans [LEED90], différentes heuristiques utilisant des techniques de réduction sont présentées et comparées notamment avec un algorithme donnant les solutions optimales. De même, les cycles de longueur 1 n'ont pas été pris en considération sur les exemples traités et tirés de ISCAS. L'étude est aussi prolongée au problème NP-dur qui consiste à diminuer la profondeur séquentielle d'un graphe.

Dans [PARK92], un autre algorithme est présenté, il est basé sur la recherche de cycles essentiels, c'est-à-dire des cycles dont les sommets ne supportent pas d'autres cycles plus petits. Les sommets qui appartiennent au maximum de cycles essentiels sont alors choisis en priorité.

Récemment, de nouveaux algorithmes ont été présentés. Dans [CHAK94a], un algorithme trouvant une solution optimale au problème est proposé. Il utilise différentes techniques : des transformations de graphes, du partitionnement à chaque itération d'une méthode de branchement et de majoration, et de la suppression de solutions grâce

à des majorants obtenus à partir de formulations de programmation linéaire entière du problème. Sur les résultats comparatifs présentés pour les circuits de ISCAS, cet algorithme apparaît comme particulièrement efficace (en temps CPU) sur les gros circuits (plus d'un millier de bascules). Ces résultats sont confirmés sur les trois circuits industriels présentés. Dans [ASHA94], un autre algorithme trouvant aussi une solution optimale au problème est présenté. A la différence de l'algorithme précédent, il peut fournir l'ensemble de toutes les solutions au problème. Donc, il peut être facilement utilisé pour tenir compte de coûts particuliers associés aux bascules comme cela est fait dans [CHIC90]. Cet algorithme utilise aussi des techniques de partitionnement et de réduction de graphes. Il représente les cycles présents dans le graphe à l'aide d'une fonction booléenne. Dans l'implantation, la résolution du problème est faite sur une représentation de la fonction booléenne par un diagramme de décision binaire.

### **III.3.3 Analyse et résultats**

Avec les progrès effectués sur ces algorithmes, on peut au lieu de considérer le graphe séquentiel du circuit, considérer le graphe formé par les portes composant le circuit. Alors l'ensemble minimal des portes qui permet de rendre ce graphe acyclique est exploitable pour de la resynthèse [CHAK94b]. La resynthèse va consister à obtenir un circuit équivalent pour lequel les sommets solutions se trouveront en entrée d'une ou plusieurs bascules. L'intérêt de cette méthode est que le nombre de bascules finales trouvées (et à mettre en scan) est alors inférieur à celui qui serait obtenu en prenant directement le graphe séquentiel. Cependant, cette approche originale, permet difficilement d'estimer par avance la variation de coût en surface du circuit resynthétisé. Les nombres de bascules et de littéraux peuvent aussi bien diminuer qu'augmenter.

Ce critère de sélection des bascules à mettre en scan ne nécessite pas d'ATPG, et est simple à formaliser. De plus, les résultats obtenus sont des circuits acycliques. Donc, pour la génération des vecteurs de test, on peut utiliser des ATPGs spécialisés pour les circuits acycliques comme celui présenté dans [MIN92]. Ce critère ne permet pas d'utiliser l'information donnée par un ensemble de fautes déjà couvertes. Son calcul est rapide et facilement applicable aux gros circuits moyennant de bonnes heuristiques. De plus, l'emploi de critères de sélection de bascules directement liés à une notion courante de la théorie des graphes, permet de bénéficier des avancées des algorithmes utilisés dans le domaine de la théorie des graphes.

## **IV. Autres approches symboliques ou de haut niveau**

### **IV.1 Introduction**

A un haut niveau, on retrouve certaines méthodes et principes utilisés à un bas niveau comme le niveau porte. On définit des mesures de testabilité, on utilise des notions de graphes, on exploite aussi des propriétés de transparence de certains blocs (combinatoire ou non) du circuit. L'intervention du concepteur est beaucoup plus grande, c'est lui qui peut aussi guider les choix de l'outil qu'il utilise (voir par exemple [SAMA89] pour de l'insertion de multiplexeurs, ou [BEEN89] pour la stratégie générale adoptée). Ces méthodes vont être surtout bien adaptées aux circuits contenant des chemins de données.

### **IV.2 Les principales méthodes**

#### **IV.2.1 Méthodes statistiques ou probabilistes**

Dans [THEA89], deux méthodes de mesure de l'information véhiculée à un haut niveau sont présentées. Elles fournissent des mesures de contrôlabilité et d'observabilité permettant de partitionner un circuit. La première mesure s'obtient par des échantillonnages de résultats obtenus lors de la simulation de vecteurs aléatoires. La deuxième mesure s'obtient par des combinaisons sur le circuit des coefficients de transmission d'information, ITC (Information Transmission Coefficient). Ces coefficients s'appliquent à des opérateurs du type addition, soustraction, négation, division par deux, "et", "ou", retard, multiplexeur, ...

Dans [CHEN89], une mesure de testabilité est proposée pour les machines d'états finis. Les équations booléennes sont représentées par des diagrammes de décision binaire, ce qui revient à faire une implantation à l'aide uniquement de multiplexeurs à deux entrées et des constantes 0 et 1. Les mesures obtenues s'interprètent alors directement en termes probabilistes. Pour les parties séquentielles, la mesure tient compte des probabilités de transition entre différents états, et donne des informations probabilistes plus précises que SCOAP. Cependant, ces derniers calculs nécessitent en particulier la résolution de systèmes d'équations linéaires dont la taille est égale au nombre d'états de la machine d'états finis. Cette résolution est très coûteuse, car le nombre d'états possibles dépend exponentiellement du nombre de variables d'états, et le coût d'inversion d'une matrice  $n \times n$  est de complexité supérieure à  $O(n^2)$  et inférieure à  $O(n^{2,495548})$  d'après un résultat de Winograd cité avec d'autres dans [FAUR89].

De par leurs natures, ces deux mesures ne permettent pas directement de tenir compte

de fautes déjà couvertes pour améliorer les sélections des bascules à mettre en scan.

#### **IV.2.2 Méthodes structurelles et symboliques**

Dans [STEE93b], une description RTL d'un circuit est vue sous forme de graphe. Les sommets de ce graphe sont les modules élémentaires de la description (registres, ALU, ...). La première étape consiste à rendre le graphe acyclique par un algorithme de branchement et de majoration associé à des techniques d'élagage. Le caractère acyclique peut être obtenu, soit en mettant certains registres en scan, soit en ajoutant des registres de scan transparents pour le circuit en mode normal. Steensma fait ensuite appel à un générateur de vecteurs de test symboliques qui permet de modéliser des délais dans un chemin de données [STEE93a].

Les méthodes de test symbolique en général recherchent des chemins de justification et de propagation symboliques. Ces chemins peuvent être des chemins identités appelés I-Paths, définis dans [ABAD85], comme préservant les valeurs transportées. Elles recherchent alors à partir de chemins identités locaux quels blocs vont être atteignables, et quelles séquences d'entrée il faut appliquer aux signaux de contrôle du chemin de données [BLAT93]. Les difficultés de justification de valeurs symboliques proviennent en général de la non existence de fonctions inverses (non surjectivité) des blocs que l'on cherche à traverser [FREE88]. Les difficultés de propagation des valeurs symboliques proviennent en général de la non injectivité des blocs que l'on cherche à traverser [FREE88]. Cela induit des phénomènes de masquage lorsqu'on passe aux vecteurs de test réels. En agissant sur des modes de maintien de certains registres, on a pu améliorer ces méthodes. Murray, dans [MURR91], propose une méthode levant des ambiguïtés qui apparaissent lors de la propagation de valeurs aux sorties primaires à travers des fonctions non injectives. Le principe est qu'une fonction a des entrées qui doivent être propagées et d'autres qui servent de paramètres. Les entrées propagées sont maintenues pendant plusieurs coups d'horloge alors que les paramètres sont modifiés de façon à avoir unicité des valeurs propagées pour plusieurs observations consécutives des sorties de la fonction. Dans [KARA91a], une approche hiérarchisée est proposée. Dans cette approche, les modes de maintien servent à retarder des données sur les chemins de test symboliques pour éviter des conflits apparaissant sur une arborescence de propagation. On trouvera dans [KARA91b] de nombreuses références sur ces types d'approche.

Dans cette méthode de haut niveau [STEE93b], on n'est pas toujours sûr d'avoir un taux de couverture de 100%. Cette méthode de haut niveau permet de prendre en considération des notions de chemins équilibrés définis pour BALLAST en utilisant ou

non des modes de maintien. Elle nécessite que les signaux de contrôle des chemins de données et des modes de maintien soient complètement contrôlables. Dans la génération des vecteurs de test, on pourrait aussi tenir compte de certains modules déjà testés par ailleurs.

## **V. Synthèse et conclusion**

Dans le domaine du scan partiel, il y a une très grande diversité d'algorithmes, et d'années en années des améliorations et des sophistications sont proposées. Pour les principaux algorithmes, on extrait des caractéristiques essentielles, qui influent sur le coût de mise en œuvre, et l'efficacité des résultats qu'on peut attendre.

La caractéristique "structurelle niveau porte" notée SNP, est celle d'utiliser des mesures dérivées des interconnexions entre les portes du circuit.

La caractéristique "structurelle niveau bascule" notée SNB, est celle d'utiliser des mesures dérivées du graphe séquentiel du circuit. La caractéristique SNB est en soit un cas particulier de la caractéristique SNP.

La caractéristique "structurelle niveau porte et équation" notée SNPE, est celle d'utiliser des mesures dérivées à la fois des interconnexions entre les portes du circuit et des équations logiques de ces portes. La caractéristique SNP est en soit un cas particulier de la caractéristique SNPE.

La caractéristique "structurelle niveau registre" notée SNR, est celle d'utiliser des mesures dérivées des interconnexions entre les registres du circuit. La caractéristique SNB est en soit un cas particulier de la caractéristique SNR où tous les registres ne comportent qu'une seule bascule.

La caractéristique "structurelle niveau transfert de registres" notée SRTL, est celle d'utiliser des mesures dérivées à la fois des interconnexions entre les registres du circuit et des équations de transfert. La caractéristique SNPE est en soit un cas particulier de la caractéristique SRTL.

La caractéristique "considération des fautes" notée CF, est celle de faire des choix de bascules à insérer dans les chaînes de scan à partir de critères liés à un ensemble de fautes considéré. Cette caractéristique permet en particulier de pouvoir tenir compte de savoir que certaines fautes sont déjà couvertes par des vecteurs fonctionnels.

La caractéristique "générateur de vecteurs de test séquentiel" notée TPGS, est celle d'utiliser un générateur de vecteurs de test séquentiel.

La caractéristique "générateur de vecteurs de test combinatoire" notée TPGC, est celle d'utiliser un générateur de vecteurs de test combinatoire. On peut considérer que la caractéristique TPGC est un cas particulier de TPGS.

La caractéristique "ATPG final séquentiel" notée SEQ, est celle d'avoir besoin d'un ATPG séquentiel pour assurer un taux de couverture maximal après avoir sélectionné



les bascules à mettre en scan.

Les algorithmes proposés dans [CHIC90] sont regroupés sous trois rubriques a, b et c pour représenter respectivement les trois critères : nombre de cycles et profondeur séquentiel, SCOAP séquentiel, et longueur théorique des séquences de test.

Pour [PRAD91], on regroupe les heuristiques de sortance, sortance reconvergente et distance aux entrées primaires sous la rubrique a. La contrôlabilité et l'observabilité sont regroupées sous la rubrique b, et les combinaisons X2, X3, X4 sont regroupées sous la rubrique c.

	SNB	SNP	SNPE	SNR	SRTL	CF	TPGC	TPGS	SEQ
[CHIC90]a	X								X
[CHIC90]b	-	-	X						X
[CHIC90]c	X								X
[PRAD91]a	-	X							X
[PRAD91]b	-	-	X			X <sup>a</sup>	-	CONTEST	X
[PRAD91]c	-	-	X			X <sup>b</sup>	-	CONTEST	X
[KIM90]	-	-	X			X	-	FASTEST	X
[KIM93]	-	-	X			X	-	FASTEST	X
[PARI93]	-	-	X			X			X
[ABRA91]	-	-	X			X	X <sup>c</sup>		X
[MA88]	-	-	-			X	-	STALLION	X
[AGRA87]	-	-	-			X	X		X
[GUPS89] [GUPS90]				X					
[CHENG89] [CHENG90]	X								X
[LEED90]	X								X
[BHAW91]	X								X
[PARK92]	X								X
[CHAK94a]	X								X
[ASHA94]	X								X
[CHAK94b]	-	-	X						X
[CHEN89]				-	X				X

	SNB	SNP	SNPE	SNR	SRTL	CF	TPGC	TPGS	SEQ
[THEA89]				-	X				X
[STEE93b]				-	X				X <sup>d</sup>

- a. Cela est potentiellement faisable dans le calcul des contrôlabilités et des observabilités.  
 b. Idem note a.  
 c. La propagation des non-observabilités et des non-contrôlabilités est en fait seulement une partie d'un ATPG combinatoire.  
 d. Un ATPG séquentiel est nécessaire quand l'ATPG symbolique n'a pas donné de résultats satisfaisants.

On constate que la plupart de ces méthodes nécessitent des ATPGs séquentiels pour la génération des vecteurs de test finaux. De ce fait, elles sont mal adaptées aux approches hiérarchiques. De plus les ATPGs séquentiels sont en général coûteux en temps de calcul et ne fournissent pas forcément des taux de couverture maximum. Or, les critères de qualité d'un circuit, par leurs exigences demandent de s'assurer un taux de couverture maximum des fautes [CLUS93], c'est-à-dire 100% au moins dans le cas des fautes de type collage à 1 et à 0.

Dans ces méthodes, le critère surface est le premier pris en compte. Les résultats sont la plupart du temps donnés sous la forme d'un taux de couverture obtenu en fonction d'un nombre de bascules mises en scan. Les vrais coûts en surface sont rarement donnés. Ils sont en général inférieurs à 15% de la surface du circuit. Ils peuvent varier énormément en fonction de la proportion de bascules par rapport aux autres éléments du circuits. Ce critère est donc à relativiser vis à vis d'autres critères comme la qualité du circuit, la facilité de mise en œuvre du test ou les performances du circuit. Cela fait dire à Varma dans [VARM93] que le scan partiel peut actuellement apparaître dans un bilan final comme plus cher que le scan complet. Bennets et Beenker, dans [BENN93], résumant bien les tenants et aboutissants du scan partiel ou complet. Ils s'intéressent en particulier aux différentes formes du test, et à ce qui peut jouer en faveur ou en défaveur du scan partiel et du scan complet. Ils arrivent tous trois aux conclusions que l'intérêt principal du scan partiel réside en la potentialité qu'il offre : obtenir un impact minimum sur le chemin critique et donc sur les performances et les coûts économiques.

Dans la méthode proposée dans le cadre de cette thèse, on assure un taux de couverture maximal. Un ATPG combinatoire suffit pour la génération des vecteurs de test. Cette méthode peut être utilisée de manière hiérarchique. Elle peut combiner des descriptions au niveau portes et au niveau RTL. Elle est surtout centrée sur une utilisation pour réduire les coûts sur le chemin critique. Des solutions sont proposées dans les cas où une contrainte sur un chemin critique ne permet plus de garantir un taux de couverture de 100%. Dans ces cas seulement, il faudra utiliser un ATPG séquentiel ou simulateur

de fautes sur une faible partie du circuit.

## **Chapitre 2 : Proposition, formalisation et mise en œuvre d'une méthode de scan généralisée**

Des principes d'utilisation de nouvelles structures de test, plus complexes et plus générales que les chaînes de scan, sont présentés ici. On formalise mathématiquement ces structures et des propriétés structurelles et fonctionnelles que l'on peut leur attacher. Suivant les propriétés effectivement associées aux structures employées, on démontre qu'à des degrés plus ou moins élevés, elles assurent à la fois contrôlabilité et observabilité. Des algorithmes décrivent alors les étapes à mettre en œuvre pour contrôler et observer les bascules de ces structures. Dans le cas le plus favorable, à savoir quand un ATPG combinatoire sera suffisant pour générer les vecteurs de test finaux, le taux de couverture sera maximal comme dans le cas du scan complet. Sinon il faudra employer un ATPG séquentiel ou employer localement des techniques d'ATPGs séquentiels pour traiter les bascules posant des problèmes de testabilité. Enfin, l'obtention de ces structures est abordée à partir de différents types de blocs fonctionnels qui existent dans un circuit, en attachant une attention toute particulière aux problèmes de vitesse.



## I. Introduction

Une chaîne scan est un objet qui a un intérêt indéniable pour le test. Elle permet de justifier un état quelconque des bascules qui la composent, par ce qui est appelé le décalage en entrée. Les données proviennent de l'entrée de la chaîne de scan (entrée de scan), qui est une entrée primaire du circuit. Elle permet de déterminer un état quelconque des bascules qui la composent, par ce qui est appelé le décalage en sortie. Les données sont observées à la sortie de la chaîne de scan (sortie de scan), qui est une sortie primaire du circuit. Utilisée pour le test, la chaîne de scan sert à amener les vecteurs de test locaux combinatoires et à propager le résultat de leur application. Trois étapes résument l'emploi des chaînes de scan pour le test dans le cas du scan complet : décalage en entrée des vecteurs de test, application des vecteurs de test, décalage en sortie des vecteurs de test. Le décalage en entrée et en sortie sont en soi des modes de fonctionnement identiques, mais c'est l'interprétation qu'on en fait qui diffère. Ce mode de fonctionnement est appelé mode de scan. Un ATPG utilise ces mêmes caractéristiques des chaînes pour générer des vecteurs de test en présence de scan partiel. Les chaînes de scan au sens strict ne sont pas les seuls objets qui permettent ce genre d'emploi pour le test. De nombreuses extensions sont possibles. Dans ce travail, on a cherché à réaliser des extensions assez générales pour qu'elles puissent être exploitées en substitution au scan classique complet ou partiel dans le cadre de l'amélioration des performances des circuits en conservant comme objectif d'avoir un taux de couverture maximal. On a formalisé et combiné de nombreuses et nouvelles extensions possibles de la notion de scan. On décrit ici rapidement les idées qui ont poussé à formaliser cette généralisation de la méthode de scan.

L'extension la plus simple consiste à autoriser des inversions de la donnée le long de la chaîne de scan. D'autres extensions, s'obtiennent en combinant plusieurs chaînes de scan. Par exemple, dans la figure 2 page 69 les données entre deux chaînes sont com-

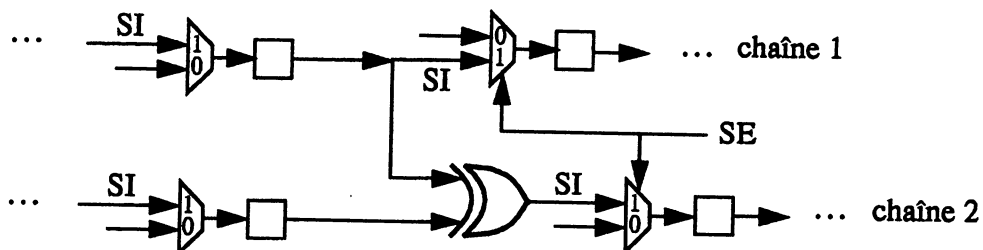


Figure 2 : combinaison de deux chaînes de scan par un "ou exclusif"

binées. La chaîne 1, conserve de manière naturelle ses propriétés de décalage en entrée et en sortie. La chaîne 2 pourra être contrôlée à un état quelconque à partir de son entrée de scan si on connaît aussi les valeurs qui ont été entrées pour la chaîne 1. De même,

on pourra déterminer un état quelconque de la chaîne 2 en observant sa sortie de scan si on observe aussi la sortie de scan de la chaîne 1. Cette notion est généralisée à un nombre quelconque de chaînes de scan. Elle fait intervenir des notions de bijectivité. On constate déjà que la mise en œuvre de l'équivalent du décalage en entrée sera plus complexe car il faudra réaliser des inversions de fonctions.

Dans les circuits, on trouve couramment des ensembles de bascules qui rebouclent sur eux-mêmes par l'intermédiaire de chemins combinatoires. Si on se place au niveau des bascules individuelles du graphe séquentiel du circuit, cela correspond en général à des cycles de longueur strictement supérieure à 1. Ce sont particulièrement les cycles qui créent la difficulté du test des circuits séquentiels. La figure 3 page 70 montre une bas-

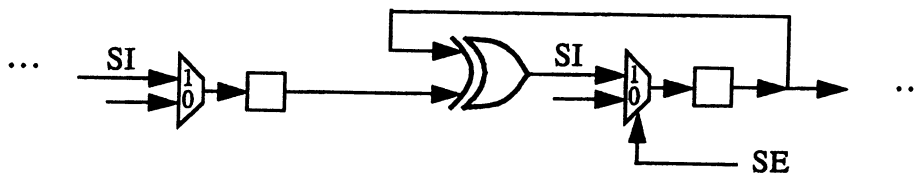


Figure 3 : rebouclage dans un chaîne de scan par un "ou exclusif"

cule insérée dans une chaîne de scan, qui, en mode de scan, combine la valeur de son entrée de scan avec sa valeur de sortie courante. On la note par exemple RT, R pour signifier ce rebouclage, et T pour signifier transmission en tant que bascule d'une chaîne de scan. On pourra déterminer l'état de toutes les bascules dans cette chaîne par l'étape de décalage en sortie. Cependant, la valeur d'une bascule qui est placée avant RT dans la chaîne de scan, sera combinée avec celle de RT à son passage. La valeur de RT à l'instant  $t$ , une fois observée en sortie permettra de calculer de proche en proche les valeurs de RT aux instants supérieurs à  $t$  ainsi que les valeurs des autres bascules à l'instant  $t$ . De même si on connaît l'état initial des bascules placées avant RT dans la chaîne de scan (par une observation préalable de la remise à zéro par exemple), alors on pourra justifier un état quelconque pour les bascules de cette chaîne de scan. Il faudra cependant réaliser des inversions de la fonction "ou exclusif" lorsque l'une des valeurs en entrée est connue. Dans l'extension proposée, on considère des boucles pouvant toucher des ensembles quelconques de bascules utilisées pour propager de l'information. Les fonctions décrivant ces boucles devront vérifier certaines propriétés de bijectivité partielle. On obtiendra alors des résultats pour le test identiques à ceux du scan classique avec cependant des mises en œuvre plus complexes.

On emploiera une nouvelle terminologie adaptée pour parler des objets utilisés dans ces extensions. On parlera de mode local pour désigner l'équivalent du multiplexeur utilisé dans le scan classique (intégré ou non à une bascule ou à de la logique préexistante du circuit). On parlera de mode global pour désigner l'équivalent d'un ensemble de chaî-

nes de scan. On parlera d'activation d'un mode global ou local pour désigner le fonctionnement correspondant au mode de scan. Ces nouveaux objets seront définis formellement au II pour les modes locaux, et au III pour les modes globaux. Seront aussi définies des propriétés identiques à celles du scan. La propriété principale sera la bijectivité, elle correspond à une extension de l'égalité entre l'entrée de scan du multiplexeur de scan et sa sortie. La propriété de disjonction correspond à une extension du fait qu'un multiplexeur ne sert qu'à une seule chaîne de scan à la fois. La propriété d'être acyclique correspond à une extension du fait que par construction les chaînes de scan sont acycliques. La propriété de chaînage avant correspond à une extension du fait que les multiplexeurs sont connectés en sortie soit à une sortie primaire, soit à une bascule elle-même connectée à l'entrée d'un autre multiplexeur. Cette propriété permet aux modes locaux d'un mode global de propager à des modes locaux successeurs l'information présente en entrée de ce mode. La propriété de chaînage arrière correspond à une extension du fait que les multiplexeurs sont connectés en entrée soit à une entrée primaire, soit à une bascule elle-même connectée à la sortie d'un autre multiplexeur. Cette propriété permet aux modes locaux d'un mode global de récupérer l'information en provenance des modes locaux prédécesseurs. Les bascules utilisées pour les interconnexions entre les modes locaux sont appelées les bascules de transmission. La figure 4 page 71 donne un exemple de structure d'un mode global, les points représen-

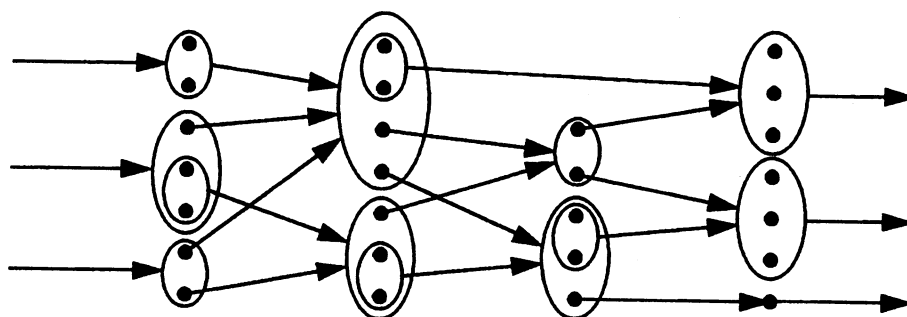


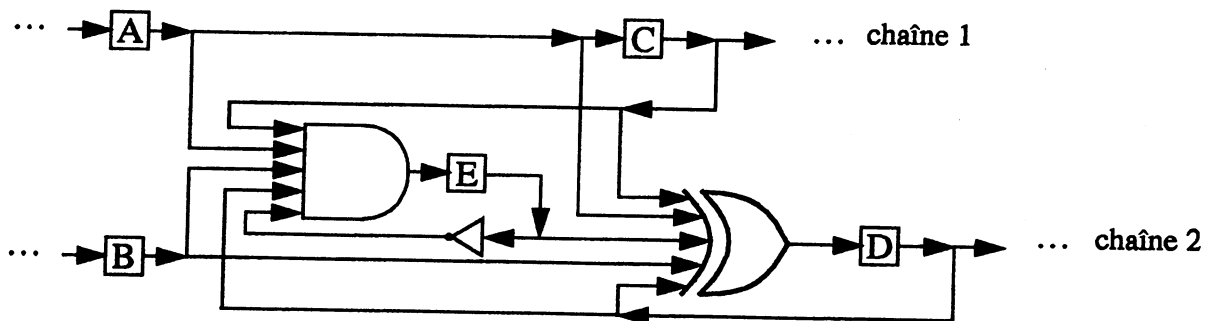
Figure 4 : exemple de structure d'un mode global

tent les bascules de transmission et les flèches représentent les relations de dépendance. Lorsque des bascules sont considérées de manière globale, on les a entourées. Lorsqu'un ensemble de flèches arrive sur un même ensemble de bascules, les entrées de ces bascules doivent être des fonctions bijectives des sorties des bascules aux origines des flèches. Sur cette figure, ne sont pas représentées les relations de dépendance possibles entre les sorties d'un ensemble de bascules et ses entrées. Ces sorties servent alors de paramètres pour ces dépendances bijectives.

Dans la méthode proposée, les principaux critères pris en compte sont dans l'ordre de priorité : le coût en vitesse, le taux de couverture final, la possibilité d'une approche



hiérarchique et le coût en surface. Partant d'un circuit séquentiel, on cherche d'abord à identifier les parties du circuit qui peuvent servir comme mode local. Cela concernera principalement des parties combinatoires implantant des fonctions mathématiques. Cependant, on n'est pas sûr de couvrir ainsi toutes les parties combinatoires où il y a des chemins critiques. On propose donc des constructions de modes locaux bijectifs à partir de la fonctionnalité avec un coût minimum sur le chemin critique. Ce coût doit être au moins inférieur à celui de l'insertion d'un multiplexeur de scan qui reste toujours possible. Cependant, dans certains cas, si on cherche à transmettre par toutes les bascules liées à des chemins critiques, des problèmes de bijectivité de fonctions peuvent apparaître. C'est pourquoi, pour ne pas affecter ces chemins critiques, on a encore étendu la méthode précédente à des cas présentant des bascules qui ne servent plus directement à transmettre des données d'un mode local à ses successeurs. Elles n'auront pas la contrôlabilité et l'observabilité des bascules de transmission. Ces bascules peuvent intervenir dans une boucle au niveau d'un mode local. On parlera de bascules de rebouclage seul. Les valeurs de sortie de ces bascules servent alors comme paramètres pour les fonctions bijectives associées aux modes locaux. La figure 5 page 72 montre le fonctionnement en mode activé d'un exemple avec une bascule de rebou-



**Figure 5** : exemple d'une combinaison de deux chaînes de scan par une bascule de rebouclage seul. E est la bascule de rebouclage seul. Les entrées de C et D dépendent bijectivement des sorties de A et B, lorsque les sorties de C, D, E sont fixées. Les multiplexeurs ou tout autre logique pour activer ce mode ne sont pas représentés dans cette figure.

L'extension du scan proposée prend en compte un nombre quelconque de ces bascules de rebouclage seul. L'introduction de ces bascules de rebouclage seul dans un mode local, affaiblit les propriétés de test associables aux modes globaux. Des notions de dépendance temporelle entre des modes locaux ou entre des entrées de bascules de transmission sont définies (III.2.5) pour analyser ou résoudre des problèmes de contrôle et d'observation liés à ces bascules. Au IV, différents théorèmes sont énoncés puis démontrés. Ils relient les propriétés structurelles et fonctionnelles des modes locaux à

des propriétés de contrôle et d'observation pour le test. La mise en œuvre du test structurel par l'intermédiaire du scan généralisé<sup>1</sup> devient beaucoup plus complexe que par l'intermédiaire du scan complet. Elle est décrite au V. Sont traités aussi les problèmes posés par les bascules de rebouclage seul pour le test structurel. Des méthodes pour l'obtention et la sélection de ces modes globaux particuliers sont explicitées au VI. Elles s'appliquent à un circuit séquentiel quelconque. Elles seront surtout adaptées aux circuits qui implantent des fonctions mathématiques ayant des propriétés de bijectivité ou des propriétés très proches. Elles permettent aussi de traiter le test de séquenceurs en modifiant le codage utilisé afin de diminuer (quand cela est critique) l'impact du test en terme de vitesse et de surface. Dans un mode global solution on intègre aussi les bascules qui servent pour l'activation des modes locaux (en particulier les bascules à l'origine des points de contrôle d'un chemin de données), en réutilisant les notions d'indépendance temporelle afin de réduire les coûts en vitesse et en surface. Finalement, des utilisations hiérarchiques des modes globaux sont décrites.

---

<sup>1</sup> On pourrait parler de scan fonctionnel pour deux raisons : il utilise des fonctions booléennes plus générales que dans le cas du scan classique, et il contribue à mieux exploiter la fonctionnalité du circuit.

## **II. Modes locaux combinatoires**

### **II.1 Préliminaires**

#### **II.1.1 Modèles utilisés pour le circuit**

Dans ce travail, on s'intéressera plus particulièrement aux circuits ou portions de circuits séquentiels synchrones. Le type de modélisation et de représentation utilisée se décompose en un ensemble de points de mémorisation (que l'on suppose être modélisables à l'aide de bascule D), en un ensemble d'entrées primaires du circuit, en un ensemble de sorties primaires du circuit et en un ensemble de fonctions booléennes. Ces dernières donnent les valeurs aux entrées des bascules et aux sorties primaires en fonction des valeurs aux entrées primaires et en sortie des bascules. Ces informations permettent de réaliser des simulations équivalentes à celle du niveau transfert de registres (RTL) : un registre est assimilé à un groupement de plusieurs bascules. En particulier la discrétisation du temps permet des représentations mathématiques et informatiques simples. On travaille au niveau des bascules et de regroupements de bascules ou registres. Lorsqu'on utilise des registres, on indexe généralement les bascules qui les composent, et les relations entre les registres sont alors liées à l'indexation utilisée. Ici, les regroupements de bascules sont traités comme des ensembles, et les relations entre des groupements de bascules ou de points (i.e. des entrées primaires, des sorties primaires, des entrées de bascules ou encore des sorties de bascules), ne sont pas liées à une indexation.

Les langages universels du type VHDL et VERILOG, largement employés dans l'industrie, fournissent de nombreuses descriptions d'un même circuit suivant certaines syntaxes et sémantiques. Ils permettent le passage à la description précédente. Si on les associe à des bibliothèques de cellules prédéfinies, ils permettent de décrire un circuit au niveau structurel comme dans le format EDIF. On peut alors aussi remonter à la description précédente grâce à la définition des bibliothèques. Les points sont aussi appelés nœuds car ils réalisent des interconnexions entre les éléments des bibliothèques. Dans les exemples employés, on considère des fautes sur les nœuds en entrée et en sortie de ces éléments de bibliothèques. Cependant les méthodes de contrôle et d'observation des fautes proposées ne font pas d'hypothèses a priori sur le modèle de faute utilisé, si ce n'est qu'elles puissent être mises en évidence par ses effets au niveau de description de type RTL. Cela ne concerne pas par exemple des fautes de type dynamique comme celles qui agissent sur le chemin critique. En revanche, on s'intéressera au chemin critique pour choisir quelles structures de test insérer, ou quelles structures existantes utiliser de façon à pénaliser le moins possible la vitesse de fonctionnement

du circuit. De même, il faut faire quelques liens avec des niveaux de description plus bas quand on s'intéresse aux coûts en surface.

### **II.1.2 Conventions d'écriture**

On emploiera dans ce travail la notion de famille indexée définie ci-après. Soient  $I$  et  $E$  deux ensembles donnés. Si à chaque  $i \in I$ , on associe un objet de  $E$  noté  $a_i$ , cela définit ce qui est appelé une famille d'objets (d'éléments) de  $E$ , indexés par  $I$  ( $I$  est appelé l'ensemble des indices). Cette famille est généralement notée  $(a_i)_{i \in I}$ , pour simplifier la notation on notera aussi cette famille :  $\hat{a}(I)$ .

Lorsqu'on utilisera des indices pour indiquer des instants, ils seront placés en position supérieure. On réservera la notation avec la lettre "x" pour le cas des variables booléennes (à valeurs dans  $\{0, 1\}$ ). Si  $N$  est un ensemble de nœuds,  $\hat{x}(N)$  est une variable booléenne généralisée indexée par un ensemble de nœuds, c'est un n-uplet de variables booléennes, elle est donc à valeurs dans  $\{0, 1\}^n$  avec  $n = \text{cardinal}(N)$ .

### **II.2 Description d'un mode local combinatoire**

Les modes locaux combinatoires représentent des modes de fonctionnement particuliers en certaines parties combinatoires du circuit. Ces modes locaux (le terme combinatoire est désormais omis par défaut) possèdent donc des signaux d'activation, qui mettent en œuvre ces fonctionnements. Ces fonctionnements sont utilisés pour le test, mais ils peuvent aussi correspondre à des fonctionnements normaux du circuit. On parlera donc de mode actif quand le mode est effectivement mis en œuvre et inactif sinon. Cette notion est la généralisation de mode de scan.

La localisation du mode va être donnée par l'ensemble des points d'entrée et de sortie du mode. Les points d'entrée sont soit des entrées primaires du circuit soit des sorties de bascules, et les points de sortie sont soit des sorties primaires du circuit, soit des entrées de bascules.

La fonctionnalité d'un mode local actif est déterminée par les fonctions booléennes des points de sortie du mode dont les arguments sont les valeurs des points d'entrée du mode. A chaque coup d'horloge, les bascules qui sont associées à des points de sortie prennent la valeur de ces derniers. Pour simplifier on remplacera toute bascule associée à un point d'entrée ayant deux sorties conjuguées, par une bascule à une seule sortie suivie d'un inverseur. Dès lors, on pourra considérer que l'on n'a affaire qu'à des bascules avec une seule sortie.

Pour pouvoir utiliser ces modes locaux afin de généraliser les modes de fonctionnement de type scan, qui permettent un test facile des chaînes de scan, ainsi que la contrôlabilité et l'observabilité simultanées de toutes les bascules qui se trouvent dans les chaînes, il faut que ces modes vérifient certaines propriétés. On commencera par formaliser la description de ces modes locaux et par définir leurs différentes propriétés. Ces propriétés seront utilisées par la suite pour démontrer que certaines classes de modes locaux utilisés dans les modes globaux (définis au III page 81) offrent les mêmes potentialités en terme de testabilité et de test que les multiplexeurs utilisés dans le scan classique.

### II.2.1 Modélisation

Soient E l'ensemble des entrées primaires d'un circuit, et S l'ensemble de ses sorties primaires. Un mode local combinatoire M va être caractérisé par :

- l'ensemble A des couples d'activation (*signal d'activation*, valeur d'activation) qui détermine comment activer ce mode. On notera PA l'ensemble des signaux d'activation appelés aussi *points d'activation*. On dira que le mode local est activé si  $\hat{x}(PA)$  correspond à A.
- l'ensemble PE des *points d'entrée*. PE se décompose en PEB, (les points qui sont des sorties de bascules, appelés *points d'entrée de type bascule*) et en PEP (les points qui sont dans E, appelés *points d'entrée primaire*). On a forcément PEB et PEP disjoints. On notera BE l'ensemble des bascules associé à PEB. Les bascules de BE sont appelées bascules d'entrée.
- l'ensemble PS des *points de sortie*. PS se décompose en PSB, (les points qui sont des entrées de bascules, appelés *points de sortie de type bascule*) et en PSP (les points qui sont dans S, appelés *points de sortie primaire*). Initialement, s'il y a recoupement entre PSB et PSP, il faudra choisir pour chaque point dans l'intersection si le point est vu comme un élément de PSB ou de PSP tout particulièrement pour toutes les propriétés<sup>2</sup> où on différencie les entrées de bascules et les sorties primaires. Certains points peuvent être utilisés comme des sorties primaires même s'ils sont aussi des entrées de bascules, et réciproquement. On notera BS l'ensemble des bascules associé à PSB. Les bascules de BS sont appelées les *bascules de sortie*.
- le système F de fonctions booléennes qui caractérisent les valeurs des points de sortie d'après les valeurs des points d'entrée. Cela peut se noter :  $\hat{x}(PS) = F_{\hat{x}(PA)}(\hat{x}(PE))$  ou

<sup>2</sup>. Dans la définition des points et bascules de transmission (voir II.3.1) et en conséquence pour la notion de chaînage avant et arrière (voir III.2.3).

plus simplement  $\hat{x}(PS) = F(\hat{x}(PE))$  puisque  $\hat{x}(PA)$  est fixé quand le mode est activé. Le système du mode activé est appelé *système des fonctions d'activation*. On suppose que  $PA \cap PE = \emptyset$ .

$BR = BS \cap BE$ , sera appelé ensemble des *bascules dites de rebouclage*. On appellera *points de sortie (ou d'entrée) rebouclants* du mode local, les entrées (ou les sorties) de ces bascules. Ces deux ensembles de points seront notés respectivement PSR et PER.

## II.2.2 Représentation graphique

Voici un schéma qui représente les différents objets définissant un mode local. Ce

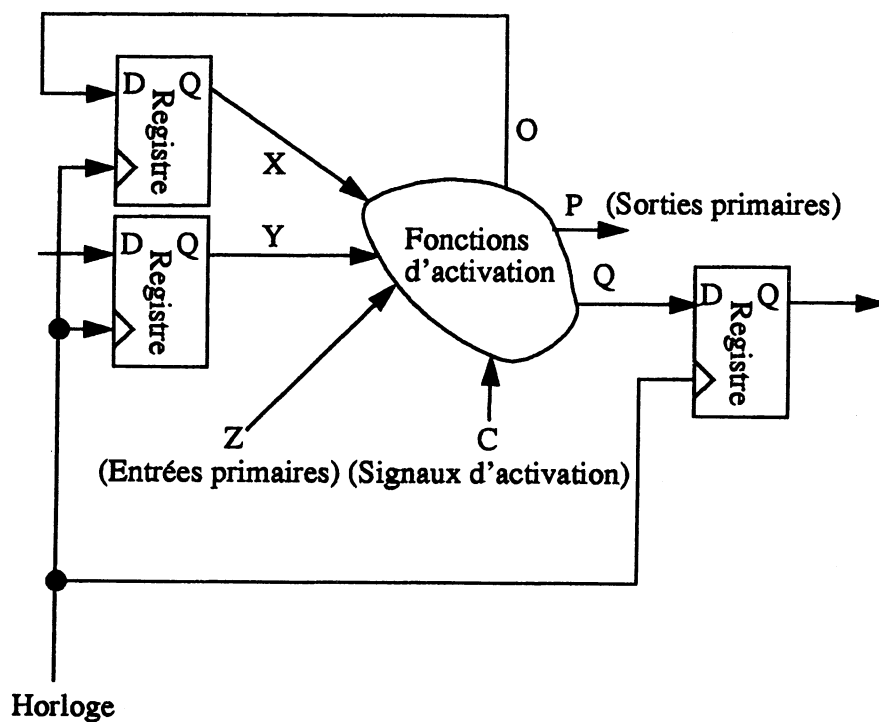


Figure 6 : représentation graphique d'un mode local

schéma est générique,  $X, Y, Z, O, P, Q, C$  peuvent ne contenir aucun nœud. L'indication registre représente un ensemble de bascules, et on aurait pu aussi décomposer ces ensembles en sous-ensembles.

L'horloge est celle qui cadence le circuit ou la portion de circuit, elle sera omise par la suite dans les schémas. Les registres seront toujours représentés par des figures rectangulaires.

PE est donné par  $X \cup Y \cup Z$ .

PEP est donné par  $Z$ .

PEB est donné par  $X \cup Y$ .

PS est donné par  $O \cup P \cup Q$ .

PSP est donné par P.

PSB est donné par  $O \cup Q$ .

PSR est donné par O.

PER est donné par X.

Le reste de la figure symbolise le système de fonctions F décrivant le fonctionnement du mode et les relations liant les divers groupements des nœuds :  $O = o_C(X,Y,Z)$ ,  $P = p_C(X,Y,Z)$ , et  $Q = q_C(X,Y,Z)$ .

On adoptera les conventions suivantes, si N est un ensemble de nœuds en entrée de bascules, on notera  $\widehat{N}$  l'ensemble des nœuds en sortie des bascules correspondantes, et si N est un ensemble de nœuds en sortie de bascules, on notera  $\underline{N}$  l'ensemble des nœuds en entrée des bascules correspondantes. Par exemple, on a ici :  $X = \widehat{O}$  et  $O = \underline{X}$ .

A chaque configuration prise par les éléments de C (qui définit PA), correspond un mode local. Cependant, si plusieurs configurations de C fournissent les mêmes fonctions d'activation, alors on est en droit de considérer que l'on a affaire au même mode local et on aura plus de choix pour activer le mode local.

### II.2.3 Exemples et modes particuliers

On définit les *modes rebouclants* comme les modes où il y a des bascules qui sont à la fois des bascules d'entrée et des bascules de sortie, c'est-à-dire les modes M tels que :

$BR(M) \neq \emptyset$ , ou ce qui est équivalent :  $PER(M) = PE(M) \cap \overbrace{PSB(M)} \neq \emptyset$ . En voici un exemple simple :

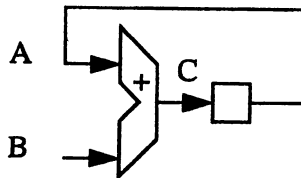


Figure 7 : exemple d'un mode local rebouclant

Des modes rebouclants peuvent être utilisés par exemple dans des implantations de BALLAST comme des modes locaux de maintien (hold) à savoir des modes qui conservent les valeurs en sortie de registre à chaque coup d'horloge. Cela peut être implanté par des registres dont les sorties rebouclent sur leurs entrées de manière à faire reprendre la même valeur aux sorties des registres au prochain coup d'horloge.

La méthode de scan implante un mode simple non rebouclant appelé mode de scan local ou mode transparent local. Dans ce mode, la première entrée d'un multiplexeur à deux entrées est connectée à une entrée primaire ou à une sortie de bascule. La sortie de ce multiplexeur est connectée à une entrée de bascule (distincte de la précédente) ou à une sortie primaire. La deuxième entrée du multiplexeur est connectée au nœud précédemment connecté à la bascule avant insertion du multiplexeur. Ce mode permet de propager des données d'un registre à l'autre sans en modifier les valeurs.

## II.3 Propriétés associées aux modes locaux

### II.3.1 Surjectivité, injectivité et bijectivité

Les propriétés d'injectivité et de surjectivité d'un mode local sont liées aux propriétés d'injectivité et surjectivité des fonctions booléennes associées aux points d'entrée et de sortie du mode local.

Soient  $M$  un mode local,  $PT(M)$  un ensemble de points dits de transmission tel que :  
 $(PS(M) - PSR(M)) \subset PT(M) \subset PS(M)$ .

$PT(M)$  se décompose en l'union disjointe de deux ensembles :

$PT(M) = PTP(M) \cup PTB(M)$  où  $PTP(M) = PT(M) \cap PSP(M)$  (l'ensemble des points de transmission primaire) et  $PTB(M) = PT(M) \cap PSB(M)$  (l'ensemble des points de transmission de type bascule). Les bascules associées aux points de transmission de type bascule seront appelées *bascules de transmission*, et cet ensemble de bascules sera noté  $BT(M)$ .

Soit  $F$  son système de fonctions booléennes d'activation :  $\hat{x}(PS(M)) = F(\hat{x}(PE(M)))$ .

$M$  est dit *injectif* par rapport à  $PT(M)$  si et seulement si la restriction de  $F$  obtenue en fixant  $\hat{x}(PER(M))$  et en ne considérant que les points de  $PT(M)$  :

$\hat{x}(PT(M)) = F(\hat{x}(PE(M) - PER(M)))$  est injective.

$M$  est dit *surjectif* par rapport à  $PT(M)$  si et seulement si la restriction de  $F$  obtenue en fixant  $\hat{x}(PER(M))$  et en ne considérant que les points de  $PT(M)$  :

$\hat{x}(PT(M)) = F(\hat{x}(PE(M) - PER(M)))$  est surjective<sup>3</sup>.

On parlera aussi de *bijectivité* dans le cas où on rencontre à la fois l'injectivité et la surjectivité.

<sup>3</sup>. La surjectivité sert avant tout à contrôler des bascules, on pourrait donc se limiter aux points de transmission de type bascule. Ce n'est pas fait dans cette présentation car la surjectivité pour l'ensemble des points est nécessaire dans une approche hiérarchique.



Dans le cas où  $M$  est un mode non rebouclant,  $PE(M) \cap PER(M)$  est vide, donc l'injectivité (ou la surjectivité) de  $M$  équivaut à l'injectivité (ou la surjectivité) de  $F$  :  $\hat{x}(PS(M)) = F(\hat{x}(PE(M)))$ . Par la suite, pour parler d'un mode local injectif ou surjectif, on omettra de dire par rapport à l'ensemble des points de transmission.

Ces propriétés sont importantes car l'injectivité va servir à assurer l'observabilité des bascules de transmission et la surjectivité à assurer la contrôlabilité de ces bascules. Ces propriétés entraînent des relations évidentes entre les cardinaux des ensembles associés à  $M$ .

$PS(M) - PSR(M) \subset PT(M) \subset PS(M)$ , et  $PSR(M) \subset PS(M)$  donc par passage aux cardinaux :  $|PS(M)| - |PSR(M)| = |PS(M) - PSR(M)| \leq |PT(M)| \leq |PS(M)|$ .

$$[\text{injectivité}] \Rightarrow [|\text{PE}(M)| - |\text{PER}(M)| \leq |\text{PT}(M)|]$$

$$[\text{surjectivité}] \Rightarrow [|\text{PE}(M)| - |\text{PER}(M)| \geq |\text{PT}(M)|]$$

$$[\text{bijectivité}] \Rightarrow [|\text{PE}(M)| - |\text{PER}(M)| = |\text{PT}(M)|].$$

On sait aussi par définition des points et bascules rebouclants que :

$$|\text{BR}(M)| = |\text{PER}(M)| = |\text{PSR}(M)|.$$

$|\text{PT}(M)| + |\text{PSR}(M)| = |\text{PS}(M)| + |\text{PT}(M) \cap \text{PSR}(M)|$  donc plus généralement la bijectivité donne :  $|\text{PS}(M)| + |\text{BT}(M) \cap \text{BR}(M)| = |\text{PE}(M)| = |\text{PT}(M)| + |\text{BR}(M)|$ .

### II.3.2 Disjonction

Deux modes locaux  $M$  et  $M'$  seront dits *disjoints en sortie* si et seulement si  $PS(M) \cap PS(M') = \emptyset$ .

Deux modes locaux  $M$  et  $M'$  seront dits *disjoints en entrée simple* si et seulement si  $PE(M) \cap PE(M') = \emptyset$ .

Deux modes locaux  $M$  et  $M'$  seront dits *disjoints en entrée non rebouclante* si et seulement si  $(PE(M) - \overbrace{PSB(M)}) \cap (PE(M') - \overbrace{PSB(M')}) = \emptyset$ .

Deux modes locaux seront dits *disjoints* si et seulement s'ils sont disjoints à la fois en entrée non rebouclante et en sortie.

### **III. Modes globaux**

#### **III.1 Définitions préliminaires**

Un mode global est défini comme un ensemble de modes locaux pour lesquels on a associé à chacun d'entre eux un unique ensemble de points de transmission. Un mode global est dit activable si on peut activer en même temps tous les modes locaux qui le composent en fixant les états des points d'activation (c'est-à-dire il existe un état de l'ensemble des points d'activation qui active tous les modes locaux). Ce qui va nous intéresser, ce sont les modes globaux activables ou rendus activables.

Un mode global sera dit non rebouclant si tous ses modes locaux sont non rebouclants.

Un mode global sera dit partiel s'il y a des bascules dans le circuit qui ne font pas partie de l'ensemble des bascules de sortie et d'entrée de tous les modes locaux le composant.

#### **III.2 Propriétés associées aux modes globaux**

##### **III.2.1 Surjectivité et injectivité**

Un mode global sera dit *surjectif* si et seulement si tous les modes locaux qui le composent sont surjectifs.

Un mode global sera dit *injectif* si et seulement si tous les modes locaux qui le composent sont injectifs.

Un mode global sera dit *bijectif* si et seulement si tous les modes locaux qui le composent sont bijectifs.

Le fait que la composition de fonctions surjectives est surjective permet principalement de démontrer des propriétés de contrôlabilité globale. De même, le fait que la composition de fonctions injectives est injective permet principalement de démontrer des propriétés d'observabilité globale

##### **III.2.2 Disjonction**

Un mode global sera dit *disjoint en sortie* si et seulement si tous les modes locaux qui le composent sont deux à deux disjoints en sortie.

Lorsqu'un mode global est disjoint en sortie, toute entrée de bascule et toute sortie primaire du circuit appartiennent à au plus l'un des ensembles de points de sortie de l'un des modes locaux du mode global.

Un mode global sera dit *disjoint en entrée* si et seulement si tous les modes locaux qui le composent sont deux à deux disjoints en entrée non rebouclante.

Lorsqu'un mode global est disjoint en entrée, toute entrée primaire du circuit appartient à au plus l'un des ensembles de points d'entrée de l'un des modes locaux, et toute sortie de bascule appartient à au plus deux de ces ensembles de points d'entrée.

Un mode global sera dit *disjoint* si et seulement s'il est à la fois disjoint en entrée et en sortie.

Lorsqu'un mode global est disjoint, toute bascule du circuit appartient à au plus l'un des ensembles de bascules de sortie de l'un des modes locaux  $M$  du mode global et à au plus l'un des ensembles de bascules d'entrée d'un autre mode local différent de  $M$ .

### III.2.3 Bascules isolées en sortie et en entrée, chaînage avant et arrière

Soient  $M$  un mode local du mode global et  $b$  une bascule de  $BT(M)$ . La bascule  $b$  est dite *isolée en sortie* si et seulement si  $\forall M'$  mode local du mode global tel que :  $M' \neq M$ , on a :  $b \notin BE(M')-BR(M')$ .

Dans un tel cas, on ne peut pas propager directement la valeur de sortie de  $b$  aux sorties primaires via le mode global. On dira alors que le mode global a des bascules isolées en sortie. Ce type de configuration pose un problème d'observabilité de  $b$  via le mode global, et donc aussi un problème de propagation de l'état du mode global aux sorties primaires. Si au contraire un tel  $M'$  existe,  $M'$  est alors appelé un *mode successeur de  $M$  associé à  $b$* .

Soient  $M$  un mode local du mode global et  $b$  une bascule de  $BE(M)-BR(M)$ . La bascule  $b$  est dite *isolée en entrée* si et seulement si  $\forall M'$  mode local du mode global, on a :  $b \notin BT(M')$ .

Dans un tel cas, on ne peut pas amener directement une valeur quelconque en entrée de  $b$  à partir des entrées primaires via le mode global. On dira alors que le mode global a des bascules isolées en entrée. Ce type de configuration pose un problème de contrôlabilité de  $b$  via le mode global, et donc aussi un problème de justification de l'état du mode global à partir des entrées primaires. Si au contraire un tel  $M'$  existe,  $M'$  est alors appelé un *mode prédécesseur de  $M$  associé à  $b$* .

On dira qu'un mode global a la *propriété de chaînage avant* ou qu'il est *sans bascule isolée en sortie* si et seulement si :

$\forall M$  mode local du mode global,  $\forall b \in BT(M)$ ,  $\exists M'$  mode local du mode global tel que :  $M' \neq M$  et  $b \in (BE(M')-BR(M'))$ .

On dira qu'un mode global a la *propriété de chaînage arrière* ou qu'il est *sans bascule*

*isolée en entrée* si et seulement si :

$\forall M$  mode local du mode global,  $\forall b \in (BE(M)-BR(M))$ ,  $\exists M'$  mode local du mode global tel que :  $M' \neq M$  et  $b \in BT(M')$ .

Plus généralement, si  $M' \neq M$ , on dira que  $M'$  est un successeur de  $M$  ou que  $M$  est un prédécesseur de  $M'$ , si  $BS(M) \cap BE(M') \neq \emptyset$ . L'ensemble des successeurs d'un mode  $M$  sera noté :  $\text{succ}(M)$ , et l'ensemble de ses prédécesseurs sera noté  $\text{pred}(M)$ .

Si le mode global est disjoint alors tout mode successeur  $M' \neq M$  associé à une bascule  $b$  de  $BT(M)$  est forcément unique et appartient à  $\text{succ}(M)$ . Si le mode global est disjoint en sortie alors tout mode prédécesseur  $M' \neq M$  associé à une bascule  $b$  de  $BE(M)-BR(M)$  est forcément unique et appartient à  $\text{pred}(M)$ .

On définit le *graphe des modes locaux* associé à un mode global comme le graphe orienté dont les sommets sont les modes locaux du mode global et dont les arêtes sont définies par la relation entre les modes locaux : "est un successeur de". Ce graphe se rapproche du graphe topologique du circuit (voir III.2 page 55 dans le chapitre "Etat de l'art sur le scan partiel") lorsqu'on considère des parties combinatoires restreintes à celles associées à un mode local activé. Dans les prochaines définitions, on emploiera parfois des termes simples de la théorie des graphes, qui peuvent se comprendre par eux-mêmes, et qui sont définis formellement dans l'annexe A, page 187: "Eléments de la théorie des graphes".

Si le mode global est sans bascule isolée en entrée, alors tout mode  $M$  source du graphe des modes locaux vérifie :  $(BE(M)-BR(M)) = \emptyset$

Si le mode global est sans bascule isolée en sortie, alors tout mode  $M$  puits du graphe des modes locaux vérifie :  $BT(M) = \emptyset$

### III.2.4 Cyclicité

Le mode global est dit *acyclique* si le graphe des modes locaux est acyclique, sinon il est dit cyclique.

Pour un graphe acyclique, nous allons numéroter les modes locaux du mode global. Une numérotation croissante (ou décroissante)  $f$  est définie comme une application  $f$  de l'ensemble des sommets du graphe dans  $\mathbb{N}$  (l'ensemble des entiers naturels) vérifiant pour tout arc d'origine  $o$  et d'extrémité  $e$  :  $f(o) < f(e)$  (ou  $f(o) > f(e)$ ).

On trouvera dans l'annexe sur la théorie des graphes la démonstration de l'équivalence entre l'existence d'une numérotation croissante (ou décroissante) et le fait que le

graphe soit acyclique.

Une numérotation croissante, qu'on appellera *distance aux entrées* notée DE est définie ci-après.

Pour tout mode  $m_l$ , source pour le graphe des modes on pose  $DE(m_l) = 0$ .

Les valeurs de DE sont ensuite définies de proche en proche. Pour tout mode  $m_l$  pour lequel DE a déjà été défini pour tous ses prédécesseurs, on pose :

$$DE(m_l) = 1 + \max_{m_l' \in \text{pred}(m_l)} (DE(m_l')).$$

On vérifie aisément que  $DE(m_l)$  correspond à la longueur du plus long chemin qui a pour origine une source quelconque du graphe et pour extrémité  $m_l$ .

Une numérotation décroissante qu'on appellera *distance aux sorties* et notée DS est définie ci-après.

Pour tout mode  $m_l$ , puits pour le graphe des modes on pose  $DS(m_l) = 0$ .

Les valeurs de DS sont ensuite définies de proche en proche. Pour tout mode  $m_l$  pour lequel DS a déjà été défini pour tous ses successeurs, on pose :

$$DS(m_l) = 1 + \max_{m_l' \in \text{succ}(m_l)} (DS(m_l')).$$

On vérifie aisément que  $DS(m_l)$  correspond à la longueur du plus long chemin qui a pour origine  $m_l$  et pour extrémité un puits quelconque du graphe.

Voici un schéma de graphe où sont indiqués les valeurs de DE et DS pour chaque som-

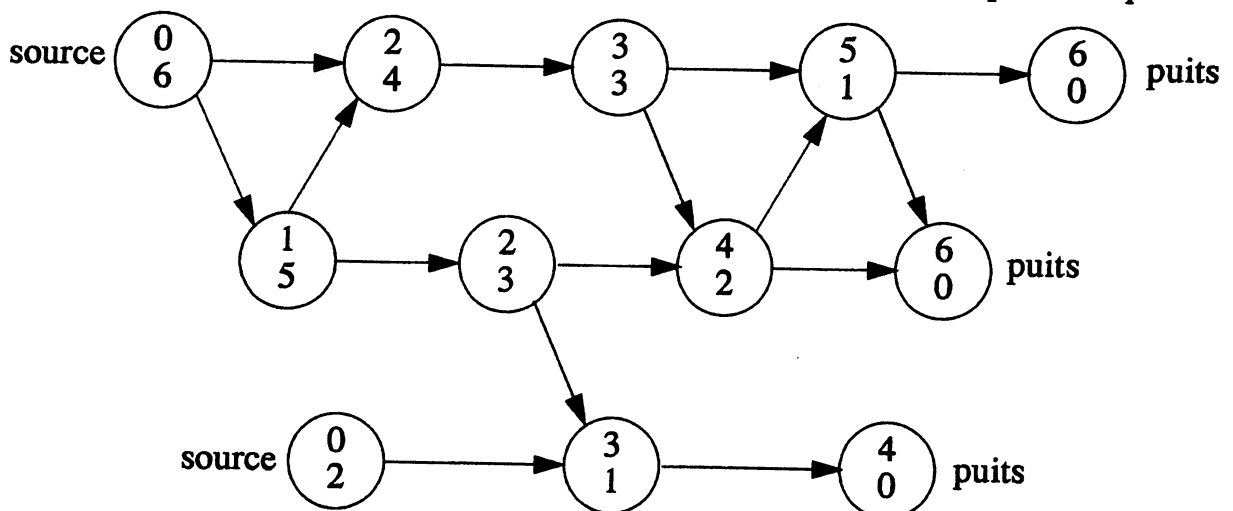


Figure 8 : exemple d'un graphe de modes locaux

met du graphe (représenté par un disque). La valeur de DE est inscrite dans la partie supérieure du disque, et celle de DS est inscrite dans la partie inférieure.

### III.2.5 Dépendance temporelle

Pour un mode global donné, on définit pour tout mode local  $ml$  de ce mode global, et pour tout  $n \geq 1$ , l'ensemble de ses *successeurs d'ordre inférieur à  $n$*  noté  $\text{succ}_n(ml)$  et

donné par :  $\text{succ}_n(ml) = \bigcup_{1 \leq i \leq n} \text{succ}^i(\{ml\})$ , où  $\text{succ}^i$  est la  $i$ ème itération de

l'opérateur  $\text{succ}$  sur un ensemble de mode locaux. L'ensemble de ses successeurs est donc aussi l'ensemble de ses successeurs d'ordre 1.

$\text{succ}_\infty(ml) = \bigcup_{1 \leq i \leq \infty} \text{succ}^i(\{ml\})$  est appelé l'ensemble de ses *successeurs d'ordre*

*quelconque*. De manière analogue, on définit l'ensemble de ses *prédécesseurs d'ordre*

*inférieur à  $n$*  par :  $\text{pred}_n(ml) = \bigcup_{1 \leq i \leq n} \text{pred}^i(\{ml\})$  et l'ensemble de ses *prédeces-*

*seurs d'ordre quelconque* par :  $\text{pred}_\infty(ml) = \bigcup_{1 \leq i \leq \infty} \text{pred}^i(\{ml\})$ . On remarquera

que  $ml$  appartient à un cycle du graphe des modes locaux si et seulement si  $ml \in \text{pred}_\infty(ml)$ , ou encore si et seulement si  $ml \in \text{succ}_\infty(ml)$ .

A chaque coup d'horloge, pour un mode local donné, les données qui se trouvent sur les points de transmission de type bascule vont passer sur les points d'entrée (de type bascule) de leur(s) mode(s) successeur(s). Ainsi, si le mode global est activé, les données présentées en entrée et en sortie d'un mode local à un instant donné vont dépendre des données qui ont été présentes aux instants précédents sur leurs modes prédécesseurs d'ordre quelconque.

Soient  $ml$  et  $ml'$  deux modes locaux distincts, on dira qu'ils sont *temporellement indépendants* si et seulement si ( $ml \notin \text{pred}_\infty(ml')$  et  $ml' \notin \text{pred}_\infty(ml)$ ), ou encore ce qui est équivalent ( $ml \notin \text{succ}_\infty(ml')$  et  $ml' \notin \text{succ}_\infty(ml)$ ).

Plus généralement pour un ensemble  $A$  de modes locaux du mode global, on dira qu'il est un *ensemble de modes temporellement indépendants* si et seulement si (pour tout mode  $ml$  de  $A$  et tout mode  $ml'$  de  $A$ , tels que  $ml'$  distinct de  $ml$ , alors  $ml$  et  $ml'$  sont temporellement indépendants).

On étend la notion d'indépendance temporelle à des ensembles de points de sortie. Pour simplifier, on supposera que le mode global est disjoint en sortie et qu'il a la propriété de chaînage arrière. Soit  $t$  un point de sortie du mode global, on définit son *mode support* noté :  $\text{supp}(t)$ , comme l'unique mode local dont  $t$  est l'un des points de sortie,

Pour tout ensemble  $T$  de points de sortie, on en induit la définition de  $\text{supp}(T)$ , l'ensemble des modes support de  $T$ . On appelle ensemble des *prédécesseurs de transmission d'un point de sortie*  $t$ , l'ensemble  $\underbrace{\text{PEB}(\text{supp}(t))}_{\text{noté } \text{predt}(t)}$ , noté  $\text{predt}(t)$ . Cet ensemble est un ensemble de points de transmission (propriété de chaînage arrière) et donc aussi de points de sortie. On définit aussi pour tout point de sortie  $t$ , et pour tout  $n \geq 1$ , l'ensemble de ses *prédécesseurs de transmission d'ordre inférieur à  $n$*  noté  $\text{predt}_n(t)$ , et donné par :  $\text{predt}_n(t) = \bigcup_{1 \leq i \leq n} \text{predt}^i(\{t\})$ , où  $\text{predt}^i$  est la  $i$ ème itération de l'opérateur  $\text{predt}$  sur un ensemble de points de sortie.  $\text{predt}_\infty(t) = \bigcup_{1 \leq i \leq \infty} \text{predt}^i(\{t\})$  est appelé l'ensemble de ses *prédécesseurs de transmission d'ordre quelconque*.

Soient  $t$  et  $t'$  deux points de transmission distincts, on dira qu'ils sont *temporellement indépendants* si et seulement si ( $t \notin \text{predt}_\infty(t')$  et  $t' \notin \text{predt}_\infty(t)$ ).

Plus généralement pour un ensemble  $T$  de points de transmission du mode global, on dira qu'il est un *ensemble de points de transmission temporellement indépendants* si et seulement si (pour tout point  $t$  de  $T$  et tout point  $t'$  de  $T$ , tels que  $t' \neq t$ , alors  $t$  et  $t'$  sont temporellement indépendants).

Pour tout ensemble  $T$  de points de sortie du mode global, on définit l'*ensemble de ses entrées de dépendance* notée  $\text{ED}(T)$  par :

$$\text{ED}(T) = \text{PE}\left(\bigcup_{t \in T} (\text{predt}_\infty(\text{supp}(t)) \cup \text{supp}(t))\right), \text{ ou ce qui est équivalent :}$$

$$\text{ED}(T) = \text{PE}\left(\text{supp}\left(\bigcup_{t \in T} (\text{predt}_\infty(t) \cup \{t\})\right)\right).$$

On définit aussi la *latence par rapport aux entrées* notée  $L(T)$  par :

$$L(T) = \max_{ml \in \text{supp}\left(\bigcup_{t \in T} (\text{predt}_\infty(t) \cup \{t\})\right)} (\text{DE}(ml)).$$

On vérifie de manière triviale que (  $A$  est un ensemble de modes locaux temporellement indépendants ) équivaut à (  $\text{PT}(A)$  est un ensemble de points de transmission temporellement indépendants ). La vérification que  $T$  est un ensemble de points de transmission temporellement indépendants ou que  $A$  est un ensemble de modes locaux temporellement indépendants peut se réaliser à partir d'algorithmes de marquage par parcours du graphe des modes locaux (on marque les modes locaux et les points de transmission). La notion d'indépendance temporelle est importante, car elle est à l'origine du contrôle complet dans le temps des modes locaux et des points de transmission.

### III.2.6 Propriétés de scan étendues

Par analogie avec la méthode de scan, et par extension, on dira qu'un mode global activable a la propriété d'*accessibilité faible* s'il vérifie lorsqu'il est activé les deux caractéristiques suivantes d'accessibilité pour le test :

- partant d'un état connu de toutes les bascules de sortie du mode global, on peut imposer<sup>4</sup> une configuration quelconque pour l'ensemble des bascules de transmission (propriété de *contrôlabilité faible des bascules de transmission*)

- par observation des sorties primaires, partant d'un état connu de toutes les bascules de rebouclage du mode global qui ne sont pas des bascules de transmission, on peut déterminer<sup>5</sup> de manière unique la configuration de départ pour l'ensemble des bascules de transmission (propriété d'*observabilité faible des bascules de transmission*).

On parlera de *contrôlabilité forte* si on n'a pas besoin de connaître l'état de départ des bascules de sortie du mode global pour pouvoir contrôler les bascules de transmission. De même, si pour déterminer l'état de départ des bascules de transmission, on n'a pas besoin de connaître l'état de départ des bascules de sortie qui ne sont pas des bascules de transmission, alors on parlera d'*observabilité forte*. Si on vérifie à la fois les propriétés d'observabilité et de contrôlabilité fortes, alors on parlera de propriété d'*accessibilité forte*.

Si toutes les bascules de sortie du mode global sont aussi des bascules de transmission, ce qui est par exemple le cas pour un mode global non rebouclant, alors il n'y a plus de différence entre l'observabilité forte et l'observabilité faible.

On dira qu'un mode global a la propriété de *contrôlabilité continue* pour un ensemble N de nœuds du circuit, si partant d'un état connu de toutes les bascules de sortie du mode global, on peut imposer une séquence de valeurs quelconque sur l'ensemble des nœuds de N après un certain temps fini de latence indépendant de l'état initial. De la même façon, on qualifiera les contrôlabilités faibles et fortes précédentes d'*instantanées* parce qu'elles ne durent que pendant un instant. (On pourrait aussi parler de contrôlabilité continue faible ou forte selon si on considère ou non un état connu au départ de toutes les bascules de sorties du mode global).

<sup>4</sup>. Cela se fait forcément en un nombre fini de coups d'horloge étant donné que les modèles de circuits utilisés comportent des nombre finis d'états et de transitions d'états possibles.

<sup>5</sup>. Cf 4.



## **IV. Théorèmes sur les propriétés de scan**

### **IV.1 Enoncés**

**Théorème de contrôlabilité faible :** tout mode global activable, disjoint, acyclique, surjectif, et qui a la propriété de chaînage arrière, possède alors la propriété de contrôlabilité faible.

**Théorème d'observabilité faible :** tout mode global activable, acyclique, injectif, et qui a la propriété de chaînage avant, possède alors la propriété d'observabilité faible.

**Théorème de contrôlabilité forte :** tout mode global activable, disjoint, acyclique, non rebouclant, surjectif, et qui a la propriété de chaînage arrière, possède alors la propriété de contrôlabilité forte.

**Théorème d'observabilité forte :** tout mode global activable, acyclique, sans bascule de rebouclage qui ne soit pas de transmission, injectif, et qui a la propriété de chaînage avant, possède alors la propriété d'observabilité forte.

**Théorème d'accessibilité faible :** tout mode global activable, disjoint, acyclique, bijectif, et qui a les propriétés de chaînage avant et arrière, possède alors la propriété d'accessibilité faible.

**Théorème d'accessibilité forte :** tout mode global activable, disjoint, acyclique, sans bascule de rebouclage qui ne soit pas de transmission, bijectif, et qui a les propriétés de chaînage avant et arrière, possède alors la propriété d'accessibilité forte.

**Théorème de contrôlabilité continue :** tout mode global activable, disjoint, acyclique, surjectif, et qui a la propriété de chaînage arrière, possède alors la propriété de contrôlabilité continue pour tout ensemble de points de transmission temporellement indépendants du mode global.

### **IV.2 Démonstrations**

#### **IV.2.1 Notations et définitions préliminaires**

Soient

- ML l'ensemble des modes locaux présents dans le mode global
- SB l'ensemble des points de sortie de type bascule du mode global :

$$SB = \bigcup_{ml \in ML} PSB(ml)$$

- TB l'ensemble des points de transmission de type bascule du mode global :

$$TB = \bigcup_{ml \in ML} PTB(ml)$$

- EB l'ensemble des points d'entrée de type bascule du mode global :

$$EB = \bigcup_{ml \in ML} PEB(ml)$$

- SP l'ensemble des sorties primaires du mode global :  $SP = \bigcup_{ml \in ML} PSP(ml)$

- TP l'ensemble des points de transmission primaire du mode global :

$$TP = \bigcup_{ml \in ML} PTP(ml)$$

- EP l'ensemble des entrées primaires du mode global :  $EP = \bigcup_{ml \in ML} PEP(ml)$ .

On suppose désormais que le graphe est acyclique pour que DE et DS soient définis.

Pour tout  $i \geq 0$ , on définit :  $E_i = \bigcup_{\substack{ml \in ML \\ DE(ml) = i}} PEP(ml)$  et  $S_i = \bigcup_{\substack{ml \in ML \\ DS(ml) = i}} PTP(ml)$ .

Pour un mode global, on définit aussi :

- l'ensemble de contrôlabilité d'ordre  $i$  noté  $C_i$  par :  $C_i = \bigcup_{\substack{ml \in ML \\ DE(ml) \leq i}} PT(ml)$

- l'ensemble d'observabilité d'ordre  $i$  noté  $O_i$  par :  $O_i = \bigcup_{\substack{ml \in ML \\ DS(ml) \leq i}} PE(ml)$

- l'ensemble des paramètres d'ordre  $i$  noté  $P_i$  par :

$$P_i = \bigcup_{\substack{ml \in ML \\ DS(ml) \leq i}} \overbrace{PER(ml) - PTB(ml)}$$

- l'ensemble des points d'entrée rebouclants d'ordre  $i$  noté  $R_i$  par :

$$R_i = \bigcup_{\substack{ml \in ML \\ DS(ml) = i}} PER(ml)$$

- l'ensemble frontière d'observabilité d'ordre  $i$  noté  $FO_i$  défini par :

$$FO_i = \bigcup_{\substack{ml \in ML \\ DS(ml) = i}} PE(ml).$$

### IV.2.2 Démonstration du théorème de contrôlabilité faible

Puisque la fonction DE est majorée, tout mode global vérifie de manière triviale pour tout majorant  $m$  de la fonction DE :  $C_m = \bigcup_{ml \in ML} PT(ml)$ . Pour démontrer l'implication de contrôlabilité faible, il suffit de démontrer en partant des hypothèses du théorème que la proposition P(j) suivante est vraie pour tout  $j \geq 0$  :

$\forall \hat{c}^j(C_j)$ , famille de valeurs booléennes ( $c_i^j \in \{0,1\}$ ).

$\exists \hat{c}^j(\bigcup_{j \geq f \geq 0} E_f)$ ,  $\hat{c}^{j-1}(\bigcup_{j-1 \geq f \geq 0} E_f)$ , ...,  $\hat{c}^0(E_0)$  (fonction de l'état initial de  $\widehat{SB}$ ) telles que si on impose aux entrées primaires pour tout  $e$  vérifiant  $j \geq e \geq 0$  :

$$\left( \begin{array}{l} \hat{x}^e(\bigcup_{f > e} E_f) = 0 \\ \hat{x}^e(\bigcup_{e \geq f \geq 0} E_f) = \hat{c}^e(\bigcup_{e \geq f \geq 0} E_f) \end{array} \right), \text{ on a alors l'égalité suivante : } \hat{x}^j(C_j) = \hat{c}^j(C_j).$$

La démonstration s'effectue par récurrence sur  $j$ , après avoir démontré le lemme 1 sous les mêmes hypothèses que le théorème.

**Lemme 1 :**  $\forall k \geq 0$ ,  $\forall \hat{d}^k(C_0)$ ,  $\hat{d}^{k-1}(C_0)$ , ...,  $\hat{d}^0(C_0)$ , familles de valeurs booléennes,  $\exists \hat{d}^k(E_0)$ ,  $\hat{d}^{k-1}(E_0)$ , ...,  $\hat{d}^0(E_0)$  telles que si on impose aux entrées primaires pour tout  $e$  vérifiant  $k \geq e \geq 0$  :  $\hat{x}^e(E_0) = \hat{d}^e(E_0)$ , on a alors les égalités suivantes :

$$\hat{x}^e(C_0) = \hat{d}^e(C_0).$$

Démonstration du lemme 1 :

$C_0 = \bigcup_{\substack{ml \in ML \\ DE(ml) = 0}} PT(ml)$  est une union disjointe puisque le mode est disjoint en sortie.

Par définition de DE et puisque le mode global est sans bascule isolée en entrée,

$E_0 = \bigcup_{\substack{ml \in ML \\ DE(ml) = 0}} (PE(ml) - PER(ml))$ . Cette union est disjointe car le mode est disjoint en entrée.

Donc les systèmes des fonctions d'activation de ces modes locaux ( $DE(ml) = 0$ ) utilisent des variables d'entrée distinctes entre chaque système, ainsi que pour les variables de sortie. De plus comme on a la surjectivité locale pour chacun d'entre eux (mode global surjectif), cela se traduit par :

A  $\hat{x}(\bigcup_{ml \in ML} PER(ml))$  donné,  $\hat{x}(\bigcup_{ml \in ML} PT(ml))$  dépend surjectivement de  $\hat{x}(\bigcup_{ml \in ML} (PE(ml) - PER(ml)))$ . Soit plus simplement,  $C_0$  dépend surjective-

ment de  $E_0$ . Donc le lemme 1 est vérifié pour  $k = 0$  et les choix possibles pour  $\hat{d}^0(E_0)$  sont déterminés par la valeur du paramètre  $\hat{x}^0(\bigcup_{ml \in ML} PER(ml))$  qui est induit par l'état du mode global à l'instant 0 supposé connu,  $\hat{x}^0(\widetilde{SB})$ .

Une fois ce choix fait,  $\hat{x}^1(\bigcup_{ml \in ML} PER(ml))$  soit  $\hat{x}^0(\bigcup_{ml \in ML} PSR(ml))$  est

déterminée par les fonctions d'activation des modes locaux comme fonction de  $\hat{x}^0(\bigcup_{ml \in ML} PE(ml))$ . Par récurrence immédiate sur  $k$ , on vérifie le lemme 1 en uti-

lisant le même argument de surjectivité et le fait que  $\hat{x}(\bigcup_{ml \in ML} PER(ml))$  est

déterminée à chaque étape. Si de plus si on a l'injectivité du mode global, alors le choix de  $\hat{d}(E_0)$  est unique pour  $\hat{x}(\bigcup_{ml \in ML} PER(ml))$  donné, et donc :

$\hat{d}^{k-1}(E_0), \dots, \hat{d}^{k-2}(E_0), \dots, \hat{d}^0(E_0)$  sont déterminées de manière unique en fonction de  $\hat{x}^0(\bigcup_{ml \in ML} PER(ml))$  (et a fortiori en fonction de  $\hat{x}^0(\widetilde{SB})$ ).

Démonstration de P(0) :

P(0) s'écrit :  $\forall \hat{c}^0(C_0), \exists \hat{c}^0(E_0)$  telle que si on impose aux entrées primaires du

circuit :  $\begin{cases} \hat{x}^0(\bigcup_{f>0} E_f) = 0 \\ \hat{x}^0(E_0) = \hat{c}^0(E_0) \end{cases}$ , on a alors l'égalité suivante :  $\hat{x}^0(C_0) = \hat{c}^0(C_0)$ .

$P(0)$  est une conséquence directe du lemme 1 appliqué à l'ordre 0 où on impose en plus  $\hat{x}^0(\bigcup_{f>0} E_f) = 0$ .

On suppose que la proposition  $P(j)$  est vraie, on va alors montrer que  $P(j+1)$  est vraie.

On considère un nouveau mode global où on a supprimé les modes locaux du mode global initial dont la distance aux entrées (DE) valait 0, ainsi que leurs bascules de sortie (cela est possible car il y a disjonction en sortie). Les sorties de leurs bascules de transmission sont désormais assimilées à des entrées primaires pour ce nouveau mode. De manière immédiate, les propriétés afférentes au mode global : être activable, disjonction, être acyclique, surjectivité, injectivité (s'il y a lieu), chaînage arrière et avant (s'il y a lieu), sont alors conservées. On en déduit que  $P(j)$  est aussi vérifiée pour ce nouveau mode. On utilisera la notation ' (prime) pour dénoter tout ce qui concerne ce nouveau mode global. Les relations entre les nouveaux et anciens ensembles définis pour un mode global découlent directement de ces égalités :

$$\forall ml \in ML', DE'(ml) = DE(ml) - 1.$$

On prend des origines des temps différentes pour ces deux modes en posant  $\hat{x}'^n = \hat{x}^{n+1}$ . L'état initial du nouveau mode global correspond à celui de l'instant 1 de l'ancien mode global :  $\hat{x}'^0(\widetilde{SB}') = \hat{x}^1(\widetilde{SB}')$ .

$$\widetilde{SB}' = \bigcup_{\substack{ml \in ML \\ DE(ml) \geq 1}} \overbrace{PSB(ml)}^{\text{traversée des bascules}}, \text{ donc } \hat{x}'^0(\widetilde{SB}') \text{ est complètement déterminée par}$$

$$\hat{x}^0\left(\bigcup_{\substack{ml \in ML \\ DE(ml) \geq 1}} PSB(ml)\right) \text{ (traversée des bascules), soit encore à partir des fonctions}$$

d'activation des modes locaux  $\hat{x}'^0(\widetilde{SB}')$  est complètement déterminée par

$$\hat{x}^0\left(\bigcup_{\substack{ml \in ML \\ DE(ml) \geq 1}} PE(ml)\right). \text{ C'est-à-dire } \hat{x}'^0(\widetilde{SB}') \text{ est complètement déterminée par}$$

$$\hat{x}^0\left(\bigcup_{i>0} E_i\right) \text{ et } \hat{x}^0\left(\bigcup_{\substack{ml \in ML \\ DE(ml) \geq 1}} PEB(ml)\right).$$

$PEB(ml) = (PEB(ml) - PER(ml)) \cup (PEB(ml) \cap PER(ml))$ , or aucune bascule n'est isolée en entrée et  $(PEB(ml) \cap PER(ml)) \subset \overbrace{PSB(ml)}$ , donc

$$\bigcup_{\substack{ml \in ML \\ (DE(ml) \geq 1)}} PEB(ml) \subset \bigcup_{\substack{ml \in ML \\ (DE(ml) \geq 0)}} \overbrace{PTB(ml)} \cup \bigcup_{\substack{ml \in ML \\ (DE(ml) \geq 1)}} \overbrace{PSB(ml)} \subset \widehat{SB}.$$

$\hat{x}^0(\widehat{SB})$  est supposée donnée et on peut imposer arbitrairement  $\hat{x}^0(\bigcup_{i>0} E_i) = 0$ , par

conséquent,  $\hat{x}'^0(\widehat{SB}')$  est prédéterminable indépendamment des valeurs que l'on va imposer à  $\hat{x}^0(E_0)$ .

$\forall \hat{c}^{j+1}(C_{j+1})$ , famille de valeurs booléennes,

$$\text{soit } \hat{c}^j(C'_j) = \hat{c}^j(C_{j+1} - C_0) = \hat{c}^{j+1}(C_{j+1} - C_0).$$

$P(j)$  est vraie pour le nouveau mode global donc :

$\exists \hat{c}^j(\bigcup_{j \geq f \geq 0} E'_f), \hat{c}^{j-1}(\bigcup_{j-1 \geq f \geq 0} E'_f), \dots, \hat{c}^0(E'_0)$  telles que si on impose aux entrées

$$\text{primaires pour tout } e' \text{ vérifiant } j \geq e' \geq 0 : \begin{cases} \hat{x}^{e'}(\bigcup_{f > e'} E'_f) = 0 \\ \hat{x}^{e'}(\bigcup_{e' \geq f \geq 0} E'_f) = \hat{c}^{e'}(\bigcup_{e' \geq f \geq 0} E'_f) \end{cases} \quad (i)$$

on a alors l'égalité suivante :  $\hat{x}^j(C'_j) = \hat{c}^j(C'_j)$ , soit encore

$$\hat{x}^{j+1}(C_{j+1} - C_0) = \hat{c}^{j+1}(C_{j+1} - C_0) \quad (ii)$$

Soit  $BC_0 = C_0 \cap \bigcup_{\substack{ml \in ML \\ (DE(ml) \geq 1)}} PEB(ml)$ , on cherche d'abord à vérifier (i) pour les

variables indexées par des points de  $\overline{BC_0}$ .

Pour tout  $i \geq 0$  :  $E'_i = E_{i+1} \cup \left( \overline{BC_0} \cap \bigcup_{\substack{ml \in ML \\ (DE(ml) = i+1)}} PEB(ml) \right)$ , et

$E_{i+1} \cap \overline{BC_0} = \emptyset$ , donc pour tout  $e$  vérifiant  $j \geq e \geq 0$ , on a :

$$\bigcup_{e \geq f \geq 0} E'_f \cap \overbrace{BC_0} = \overbrace{BC_0} \cap \begin{cases} \bigcup_{ml \in ML} PEB(ml), \text{ soit encore} \\ e \geq f \geq 0 \\ DE(ml) = f + 1 \end{cases}$$

$$\bigcup_{e \geq f \geq 0} E'_f \cap \overbrace{BC_0} = \overbrace{BC_0} \cap \begin{cases} \bigcup_{ml \in ML} PEB(ml). \\ e + 1 \geq DE(ml) \geq 1 \end{cases}$$

On pose alors :

$$\hat{d}^e(BC_0 \cap \begin{cases} \bigcup_{ml \in ML} PEB(ml) \\ e \geq DE(ml) \geq 1 \end{cases}) = \hat{c}^e(\overbrace{BC_0} \cap \begin{cases} \bigcup_{ml \in ML} PEB(ml) \\ e \geq DE(ml) \geq 1 \end{cases}),$$

la correspondance entre les éléments de ces deux familles ne peut pas être faite en utilisant l'égalité des indices dans deux familles, car ils sont tous distincts, mais elle est faite en utilisant la relation  $\text{indice}_{\text{famille1}} = \text{indice}_{\text{famille2}}$ .

$$\text{De même, on a : } \bigcup_{f > e} E'_f \cap \overbrace{BC_0} = \overbrace{BC_0} \cap \begin{cases} \bigcup_{ml \in ML} PEB(ml), \text{ soit encore} \\ f > e \\ DE(ml) = f + 1 \end{cases}$$

$$\bigcup_{f > e} E'_f \cap \overbrace{BC_0} = \overbrace{BC_0} \cap \begin{cases} \bigcup_{ml \in ML} PEB(ml). \\ DE(ml) > e + 1 \end{cases}$$

$$\text{On pose alors : } \hat{d}^e(BC_0 \cap \begin{cases} \bigcup_{ml \in ML} PEB(ml) \\ DE(ml) > e + 1 \end{cases}) = 0.$$

Puisque le mode est disjoint en entrée  $BC_0$  se décompose en une union disjointe :

$$\left( BC_0 \cap \begin{cases} \bigcup_{ml \in ML} PEB(ml) \\ e + 1 \geq DE(ml) \geq 1 \end{cases} \right) \cup \left( BC_0 \cap \begin{cases} \bigcup_{ml \in ML} PEB(ml) \\ DE(ml) > e + 1 \end{cases} \right).$$

On a défini ainsi  $\hat{d}^e(BC_0)$  pour tout  $e$  vérifiant  $j \geq e \geq 0$ , et on pose de manière arbitraire  $\hat{d}^e(C_0 - BC_0) = 0$ . On définit aussi  $\hat{d}^{j+1}(BC_0) = \hat{c}^{j+1}(BC_0)$ .

Le lemme 1 appliqué à l'ordre  $j + 1$  donne alors :  $\exists \hat{c}^{j+1}(E_0), \hat{c}^j(E_0), \dots, \hat{c}^0(E_0)$  telles

que si on impose aux entrées primaires du circuit pour tout  $e$  vérifiant  $j+1 \geq e \geq 0$  :  
 $\hat{x}^e(E_0) = \hat{c}^e(E_0)$ , on a alors les égalités suivantes :  $\hat{x}^e(C_0) = \hat{d}^e(C_0)$ .

Soit en particulier  $\hat{x}^{j+1}(C_0) = \hat{c}^{j+1}(C_0)$  (iii)

De même par construction, pour tout  $e'$  vérifiant  $j \geq e' \geq 0$  on a les égalités suivantes :

$$\left( \begin{array}{l} \hat{x}^{e'}(\bigcup_{f>e'} E'_f \cap \overline{BC_0}) = 0 \\ \hat{x}^{e'}(\bigcup_{e' \geq f \geq 0} E'_f \cap \overline{BC_0}) = \hat{c}^{e'}(\bigcup_{e' \geq f \geq 0} E'_f \cap \overline{BC_0}) \end{array} \right) \quad \text{(iv)}$$

On cherche maintenant à vérifier (ii) pour les variables indexées par des points  $n$  n'appartenant pas à  $\overline{BC_0}$ .

Pour tout  $i \geq 0$  :  $E'_i = E_{i+1} \cup \left( \overline{BC_0} \cap \bigcup_{\substack{ml \in ML \\ DE(ml) = i+1}} PEB(ml) \right)$ , et

$E_{i+1} \cap \overline{BC_0} = \emptyset$ , donc pour tout  $e$  vérifiant  $j+1 \geq e \geq 1$ , on a :

$$\bigcup_{e \geq f \geq 0} E_f = E_0 \cup \bigcup_{e \geq f \geq 1} E_f = E_0 \cup \bigcup_{e \geq f \geq 1} E'_{f-1} \cap \overline{BC_0}, \text{ soit encore par changement d'indices : } \bigcup_{e \geq f \geq 0} E_f = E_0 \cup \bigcup_{e-1 \geq f \geq 0} E'_f \cap \overline{BC_0}$$

$$\text{De même, on a : } \bigcup_{f>e} E_f = \bigcup_{f>e} E'_{f-1} \cap \overline{BC_0} = \bigcup_{f>e-1} E'_f \cap \overline{BC_0}$$

$$\text{Soit pour tout } e \text{ vérifiant } j+1 \geq e \geq 1, \hat{c}^e(\bigcup_{e \geq f \geq 1} E_f) = \hat{c}^{e-1}(\bigcup_{e \geq f \geq 1} E'_f). \quad \text{(v)}$$

On suppose qu'on impose aux entrées primaires pour tout  $e$  vérifiant  $j+1 > e \geq 0$  :

$$\left( \begin{array}{l} \hat{x}^e(\bigcup_{f>e} E_f) = 0 \\ \hat{x}^e(\bigcup_{e \geq f \geq 0} E_f) = \hat{c}^e(\bigcup_{e \geq f \geq 0} E_f) \end{array} \right), \text{ alors (v) donne par construction que pour tout } e' \text{ tel}$$

que  $j \geq e' \geq 0$ , (on prend  $e' = e - 1$ ), les égalités suivantes sont vérifiées :



$$\left\{ \begin{array}{l} \hat{x}^{e'} \left( \bigcup_{f > e'} E'_f - \overline{BC_0} \right) = \hat{x}^{e'+1} \left( \bigcup_{f > e'} E'_f - \overline{BC_0} \right) \\ \hat{x}^{e'} \left( \bigcup_{e' \geq f \geq 0} E'_f - \overline{BC_0} \right) = \hat{x}^{e'+1} \left( \bigcup_{e' \geq f \geq 0} E'_f - \overline{BC_0} \right) \end{array} \right. , \text{ soit encore :}$$

$$\left\{ \begin{array}{l} \hat{x}^{e'} \left( \bigcup_{f > e'} E'_f - \overline{BC_0} \right) = 0 \\ \hat{x}^{e'} \left( \bigcup_{e' \geq f \geq 0} E'_f - \overline{BC_0} \right) = \hat{c}^{e'} \left( \bigcup_{e' \geq f \geq 0} E'_f - \overline{BC_0} \right) \end{array} \right. \quad (\text{vi})$$

(iv) et (vi) étant vérifiées, (i) se trouve vérifiée, donc (ii) l'est aussi. (ii) et (iii) étant vérifiées, l'égalité finale recherchée est obtenue :  $\hat{x}^{j+1}(C_{j+1}) = \hat{c}^{j+1}(C_{j+1})$ .

P(j+1) est donc démontrée et donc par récurrence P(j) est démontrée pour tout  $j \geq 0$ . La contrôlabilité faible est donc elle aussi démontrée.

#### IV.2.3 Démonstration du théorème d'observabilité faible

Puisque la fonction DS est majorée, tout mode global vérifie de manière triviale pour tout majorant  $m$  de la fonction DS :  $O_m = \bigcup_{ml \in ML} PE(ml)$ , et

$$P_m = \bigcup_{ml \in ML} PER(ml) - \overline{PTB(ml)}, \text{ soit aussi pour un mode ayant la propriété de}$$

chaînage avant :  $\underbrace{O_m \cap PEB}_{\bigcup_{ml \in ML} PTB(ml)}$ . Pour démontrer l'implication

d'observabilité faible, il suffit de démontrer en partant des hypothèses du théorème que la proposition Q(j) suivante est vraie pour tout  $j \geq 0$  :

$\forall \hat{c}_1(O_j), \forall \hat{c}_2(O_j)$ , 2 familles distinctes de valeurs booléennes (elles diffèrent pour au moins un indice), si  $\hat{x}_1^0(O_j) = \hat{c}_1(O_j)$  et  $\hat{x}_2^0(O_j) = \hat{c}_2(O_j)$  et  $\hat{x}_1^0(P_j) = \hat{x}_2^0(P_j)$ , alors

$$\exists k, 0 \leq k \leq j \text{ tel que } \hat{x}_1^k \left( \bigcup_{j-k \geq f \geq 0} S_f \right) \text{ distinct de } \hat{x}_2^k \left( \bigcup_{j-k \geq f \geq 0} S_f \right).$$

$\hat{x}_1$  et  $\hat{x}_2$  sont les variables booléennes généralisées représentant les états des nœuds du circuit pour deux conditions initiales différentes.

La démonstration s'effectue par récurrence sur  $j$ , après avoir démontré le lemme 2 sous les mêmes hypothèses que le théorème.

**Lemme 2 :**  $\forall k \geq 0, \forall \hat{d}_1^0(FO_k)$ , famille de valeurs booléennes, et  $\forall \hat{d}_2^0(FO_k)$  famille de

valeurs booléennes différente de la précédente, si  $\hat{x}_1^0(R_k) = \hat{x}_2^0(R_k)$  et pour  $i \in \{1, 2\}$ ,  $\hat{x}_i^0(FO_k) = \hat{d}_i^0(FO_k)$ , alors  $\hat{x}_1^0\left(\bigcup_{\substack{ml \in ML \\ DS(ml) = k}} PT(ml)\right)$  distinct de

$$\hat{x}_2^0\left(\bigcup_{\substack{ml \in ML \\ DS(ml) = k}} PT(ml)\right).$$

Démonstration du lemme 2 :

$\bigcup_{\substack{ml \in ML \\ DS(ml) = k}} PT(ml)$  est une union disjointe puisque le mode est disjoint en sortie. Donc

On considère les modes locaux pour lesquels  $DS(ml) = k$ . L'injectivité locale pour chacun de ces modes (mode global injectif) se traduit par une injectivité pour l'ensemble de leurs fonctions d'activation associées vue comme une seule fonction :

A  $\hat{x}\left(\bigcup_{\substack{ml \in ML \\ DS(ml) = k}} PER(ml)\right)$  donné,  $\hat{x}\left(\bigcup_{\substack{ml \in ML \\ DS(ml) = k}} PT(ml)\right)$  dépend injectivement de

$\hat{x}\left(\bigcup_{\substack{ml \in ML \\ DE(ml) = k}} PE(ml) - PER(ml)\right)$ . Soit encore que,  $\hat{x}\left(\bigcup_{\substack{ml \in ML \\ DS(ml) = k}} PT(ml)\right)$  dépend

injectivement de  $\hat{x}^0(FO_k - R_k)$  à  $\hat{x}^0(R_k)$  donné. Donc le lemme 2 est vérifié.

Démonstration de Q(0) :

Q(0) s'écrit :  $\forall \hat{c}_1^0(O_0)$ , famille de valeurs booléennes, et  $\forall \hat{c}_2^0(O_0)$  famille de valeurs booléennes différente, si  $\hat{x}_1^0(P_0) = \hat{x}_2^0(P_0)$  et  $\hat{x}_1^0(O_0) = \hat{c}_1^0(O_0)$  et  $\hat{x}_2^0(O_0) = \hat{c}_2^0(O_0)$ , alors  $\hat{x}_1^0(S_0)$  distinct de  $\hat{x}_2^0(S_0)$ .

Par définition de DS, on a :  $O_0 = FO_0$ , de même puisque le mode global est aussi sans bascule isolée en sortie  $S_0 = \bigcup_{\substack{ml \in ML \\ DS(ml) = 0}} PTP(ml) = \bigcup_{\substack{ml \in ML \\ DS(ml) = 0}} PT(ml)$  et

$R_0 = P_0$ . Q(0) est donc le lemme 2 appliqué à l'ordre 0.

On suppose que la proposition Q(j) est vraie, on va alors montrer que Q(j+1) est vraie. On va en fait démontrer Q(j+1) en raisonnant par l'absurde : on suppose que Q(j+1)

n'est pas vérifiée, c'est-à-dire que :

$\exists \hat{c}_1(O_{j+1}), \exists \hat{c}_2(O_{j+1})$ , deux familles distinctes de valeurs booléennes telles que,  $\exists \hat{c}^0(P_{j+1})$  telle que  $\hat{x}_1^0(O_{j+1}) = \hat{c}_1(O_{j+1})$ , et  $\hat{x}_2^0(O_{j+1}) = \hat{c}_2(O_{j+1})$ , et  $\hat{x}_1^0(P_{j+1}) = \hat{x}_2^0(P_{j+1}) = \hat{c}^0(P_{j+1})$ , et  $\forall k, 0 \leq k \leq j+1$  on a :

$$\hat{x}_1^k(\bigcup_{j+1-k \geq f \geq 0} S_f) = \hat{x}_2^k(\bigcup_{j+1-k \geq f \geq 0} S_f).$$

De manière immédiate, on a les relations suivantes :  $O_j \subset O_{j+1}$ , et  $P_j \subset P_{j+1}$ . Par conséquent l'application de  $Q(j)$  donne que :

$$\hat{x}_1^0(O_j) = \hat{x}_2^0(O_j) \text{ et est ainsi déterminable a priori.} \quad (\text{vii})$$

Par définition de DS et puisque l'on a la propriété de propagation avant, on a :

$$\bigcup_{\substack{ml \in ML \\ DS(ml) = j+1}} \text{PER}(ml) \cap \overline{\text{PTB}(ml)} \subset \bigcup_{\substack{ml \in ML \\ DS(ml) \leq j}} \text{PE}(ml) = O_j. \quad (\text{viii})$$

$$\text{Par définition de } P_{j+1}, \text{ on a : } \bigcup_{\substack{ml \in ML \\ DS(ml) = j+1}} \text{PER}(ml) - \overline{\text{PTB}(ml)} \subset R_j. \quad (\text{ix})$$

$$R_{j+1} = \bigcup_{\substack{ml \in ML \\ DS(ml) = j+1}} \text{PER}(ml), \text{ soit encore } R_{j+1} \text{ vaut :}$$

$$\bigcup_{\substack{ml \in ML \\ DS(ml) = j+1}} \text{PER}(ml) \cap \overline{\text{PTB}(ml)} \cup \bigcup_{\substack{ml \in ML \\ DS(ml) = j+1}} \text{PER}(ml) - \overline{\text{PTB}(ml)}.$$

D'où en utilisant (viii) et (ix),  $R_{j+1} \subset P_{j+1} \cup O_j$  et  $\hat{x}_1^0(R_{j+1}) = \hat{x}_2^0(R_{j+1})$  est déterminable a priori. L'application de  $Q(j)$  donne aussi directement que  $\hat{c}_1(O_{j+1} - O_j)$  est distinct de  $\hat{c}_2(O_{j+1} - O_j)$ , soit encore que  $\hat{c}_1(\text{FO}_{j+1})$  est distinct de  $\hat{c}_2(\text{FO}_{j+1})$ . On applique alors le lemme 2 en posant  $\hat{d}_1(\text{FO}_{j+1}) = \hat{c}_1(\text{FO}_{j+1})$  et  $\hat{d}_2(\text{FO}_{j+1}) = \hat{c}_2(\text{FO}_{j+1})$ . Cela donne :  $\hat{x}_1^0(\bigcup_{\substack{ml \in ML \\ DS(ml) = j+1}} \text{PT}(ml))$  est distinct de

$$\hat{x}_2^0 \left( \bigcup_{ml \in ML} PT(ml) \right)_{(DS(ml) = j+1)}$$

Or  $\bigcup_{ml \in ML} PT(ml)_{(DS(ml) = j+1)} = S_{j+1} \cup \bigcup_{ml \in ML} PTB(ml)_{(DS(ml) = j+1)}$  et  $\hat{x}_1^0 \left( \bigcup_{j+1 \geq f \geq 0} S_f \right)$  est

distinct de  $\hat{x}_2^0 \left( \bigcup_{j+1 \geq f \geq 0} S_f \right)$  donc :

$$\hat{x}_2^0 \left( \bigcup_{ml \in ML} PTB(ml) \right)_{(DS(ml) = j+1)} \text{ est distinct de } \hat{x}_2^0 \left( \bigcup_{ml \in ML} PTB(ml) \right)_{(DS(ml) = j+1)}. \quad (x)$$

Soient pour la suite de la démonstration  $\hat{x}_2^n = \hat{x}_2^{n+1}$  et  $\hat{x}_1^n = \hat{x}_1^{n+1}$  de nouvelles variables booléennes généralisées après changement de l'origine des temps. Par définition on a :  $\underline{P}_j = \bigcup_{\substack{ml \in ML \\ DS(ml) \leq j}} PSR(ml) - PTB(ml) \subset \bigcup_{\substack{ml \in ML \\ DS(ml) \leq j}} PS(ml)$  donc les

fonctions d'activation des modes locaux donnent que pour  $i \in \{1, 2\}$ ,  $\hat{x}'_i{}^1(P_j) = \hat{x}'_i{}^0(P_j)$  (traversée des bascules) est déterminée en fonction de

$$\hat{x}'_i{}^0 \left( \bigcup_{ml \in ML} PE(ml) \right)_{(DS(ml) \leq j)} = \hat{x}'_i{}^0(O_j)$$
 et est donc déterminable a priori d'après (vii).

Pour tout  $k$ ,  $0 \leq k \leq j$ ,  $\hat{x}'_i{}^k \left( \bigcup_{j-k \geq f \geq 0} S_f \right) = \hat{x}'_i{}^{k+1} \left( \bigcup_{j-k \geq f \geq 0} S_f \right)$  donc on a :

$$\hat{x}'_1{}^k \left( \bigcup_{j-k \geq f \geq 0} S_f \right) = \hat{x}'_2{}^k \left( \bigcup_{j-k \geq f \geq 0} S_f \right). \text{ Par conséquent, l'application de } Q(j) \text{ aux nou-}$$

velles variables d'état implique que  $\hat{x}'_1{}^0(O_j) = \hat{x}'_2{}^0(O_j)$ , soit en particulier puisque le mode est sans bascule isolée en sortie :

$$\hat{x}'_1{}^1 \left( \bigcup_{ml \in ML} PTB(ml) \right)_{(DS(ml) = j+1)} = \hat{x}'_2{}^0 \left( \bigcup_{ml \in ML} PTB(ml) \right)_{(DS(ml) = j+1)}.$$

Soit encore en prenant l'égalité à l'instant précédent aux entrées des bascules :

$$\hat{x}'_1{}^1 \left( \bigcup_{ml \in ML} PTB(ml) \right)_{(DS(ml) = j+1)} = \hat{x}'_2{}^0 \left( \bigcup_{ml \in ML} PTB(ml) \right)_{(DS(ml) = j+1)}. \text{ Ce qui est en contradic-}$$

tion avec (x).

$Q(j + 1)$  est donc démontrée par l'absurde et donc par récurrence  $Q(j)$  est démontrée pour tout  $j \geq 0$ . L'observabilité faible est donc elle aussi démontrée.

#### **IV.2.4 Démonstration du théorème de contrôlabilité forte**

On suppose que le mode global vérifie les hypothèses du théorème de contrôlabilité forte, c'est-à-dire qu'il vérifie les hypothèses du théorème de contrôlabilité faible, et en plus le mode global est non rebouclant. Si on reprend la démonstration du théorème de contrôlabilité faible avec l'hypothèse  $\forall m1 \in ML, PER(m1) = \emptyset$ , alors la propriété  $P(j)$  est désormais considérée comme indépendante de l'état initial de  $\widehat{SB}$ . Le lemme 1 donne une dépendance des points de  $C_0$  qui n'est plus associée à aucun paramètre donné au départ. La propriété  $P(j)$  est alors démontrée par récurrence sur  $j$  sans devoir se préoccuper de l'état initial car le nouveau mode global considéré conserve la propriété d'être non rebouclant.

#### **IV.2.5 Démonstration du théorème d'observabilité forte**

On suppose que le mode global vérifie les hypothèses du théorème d'observabilité forte, c'est-à-dire qu'il vérifie les hypothèses du théorème d'observabilité faible, et en plus toute bascule de rebouclage est une bascule de transmission. Le théorème d'observabilité forte est donc un cas particulier du théorème d'observabilité faible.

#### **IV.2.6 Démonstration du théorème d'accessibilité faible**

Ce théorème résume la mise en commun des hypothèses et des résultats des théorèmes de contrôlabilité faible et d'observabilité faible.

#### **IV.2.7 Démonstration du théorème d'accessibilité forte**

Les hypothèses du théorème d'accessibilité forte contiennent celles du théorème d'observabilité forte. Donc, si pour un mode global activé vérifiant ces hypothèses on impose pendant un nombre suffisant de coups d'horloge des valeurs nulles aux entrées primaires du mode global, alors son état initial est déterminable après observation des sorties primaires. De plus à partir du choix arbitraire des valeurs imposées aux entrées primaires du mode global (on leur attribue en effet des valeurs nulles), on détermine l'état final des sorties des bascules du mode global. Cela permet d'appliquer le théorème de contrôlabilité forte à partir de cette nouvelle origine des temps et donc de démontrer la contrôlabilité forte. Le théorème d'accessibilité forte est ainsi vérifié. Pour éviter d'effectuer deux passes pour contrôler les bascules du mode global, on ren-

force les hypothèses en rajoutant que le mode est non rebouclant. La démonstration est alors immédiate puisqu'on a les hypothèses du théorème de contrôlabilité forte. On remarquera que cette nouvelle hypothèse implique que toute bascule de rebouclage est une bascule de transmission puisqu'il n'y a plus alors de bascules de rebouclage.

#### IV.2.8 Démonstration du théorème de contrôlabilité continue

En partant des hypothèses du théorème de contrôlabilité continue, la démonstration est faite par récurrence sur la latence des ensembles de points de transmission temporellement indépendants. La proposition de récurrence à l'ordre  $m$ ,  $R(m)$  est :  $\forall k \geq 0, \forall T$  ensemble de points de transmission temporellement indépendants tel que  $L(T) = m$ ,  $\forall \hat{d}^{m+k}(T), \hat{d}^{m+k-1}(T), \dots, \hat{d}^1(T)$ , familles de valeurs booléennes,  $\exists \hat{d}^{m+k}(D), \hat{d}^{m+k-1}(D), \dots, \hat{d}^0(D)$  dépendantes de l'état initial du mode global et où  $D = ED(T)$  telles que si on impose sur les entrées primaires pour tout  $i$  tel que  $m+k \geq i \geq 0$  :  $\hat{x}^i(D) = \hat{d}^i(D)$ , alors pour tout  $j$  tel que  $m+k \geq j \geq 1$  on a les égalités suivantes :  $\hat{x}^j(T) = \hat{d}^j(T)$ .

Avoir  $R(m)$  vérifiée pour  $D = ED(T)$  est équivalent à avoir  $R(m)$  vérifiée pour  $D = EP$  car lorsque le mode global est activé, par définition de  $ED$ , l'état des points de  $T$  ne peut dépendre de l'état des points de  $EP - ED(T)$  même après un nombre quelconque de coups d'horloge.

$R(0)$  est vérifiée d'après le lemme 1 (page 90) car  $(L(T) = 0) \Rightarrow (T \subset C_0)$ .

On suppose que la proposition  $R(m)$  est vraie, on va alors montrer que  $R(m+1)$  est vraie.

Comme pour la démonstration du théorème de contrôlabilité faible, on considère un nouveau mode global où on a supprimé les modes locaux du mode global initial dont la distance aux entrées ( $DE$ ) valait 0, ainsi que leurs bascules de sortie. Les sorties de leurs bascules de transmission sont désormais assimilées à des entrées primaires pour ce nouveau mode.  $R(m)$  est aussi vérifiée pour ce nouveau mode. On utilisera la notation ' (prime) pour dénoter tout ce qui concerne ce nouveau mode global. Les relations entre les nouveaux et anciens ensembles définis pour un mode global découlent directement de ces égalités :  $\forall m1 \in ML', DE'(m1) = DE(m1) - 1$ .

On prend des origines des temps différentes pour ces deux modes en posant

$$\hat{x}^{,n} = \hat{x}^{n+1}.$$

$\forall k \geq 0, \forall T$  ensemble de points de transmission temporellement indépendants tel que  $L(T) = m + 1, \forall \hat{d}^{m+k+1}(T), \hat{d}^{m+k}(T), \dots, \hat{d}^1(T)$ , familles de valeurs booléennes.

On considère  $T' = T - C_0$ , il est un ensemble de points de transmission temporellement indépendants du nouveau mode global, avec :  $L(T') = m$ . Soient pour tout  $j$  tel que  $m + k \geq j \geq m$  :  $\hat{d}^j(T') = \hat{d}^{j+1}(T)$ . L'application de R(1) pour ce nouveau mode global donne alors :  $\exists \hat{d}^{m+k}(D'), \hat{d}^{m+k-1}(D'), \dots, \hat{d}^0(D')$  dépendantes de l'état initial du nouveau mode global et où  $D' = ED(T')$  telles que si on impose sur les entrées primaires pour tout  $i$  tel que  $m + k \geq i \geq 0$  :  $\hat{x}^i(D') = \hat{d}^i(D')$  (xi)

alors pour tout  $j$  tel que  $m + k \geq j \geq m$  on a les égalités suivantes :  $\hat{x}^j(T') = \hat{d}^j(T')$ , soit encore par changement d'indice : pour tout  $j$  tel que  $m + 1 + k \geq j \geq m + 1$  on a les égalités suivantes :  $\hat{x}^j(T') = \hat{d}^j(T')$ . (xii)

L'état initial du nouveau mode global correspond à celui de l'instant 1 de l'ancien mode global :  $\hat{x}^0(\widetilde{SB}') = \hat{x}^1(\widetilde{SB}')$ . D'après la démonstration du théorème de contrôlabilité faible, si on impose arbitrairement  $\hat{x}^0(\bigcup_{i>0} E_i) = 0$ , alors  $\hat{x}^0(\widetilde{SB}')$  est prédéterminable indépendamment des valeurs que l'on va imposer à  $\hat{x}^0(E_0)$  et ne dépend que de  $\hat{x}^0(\widetilde{SB})$ , l'état initial de l'ancien mode global.

Soit  $TC'_0 = \overbrace{D' \cap EB}$ , alors puisque le mode global possède la propriété de chaînage arrière, on a  $TC'_0 \subset C_0$ .  $D'$  se décompose en une union disjointe :

$$D' = \overbrace{TC'_0} \cup (D' \cap \bigcup_{i>0} E_i).$$

Soit  $TC_0 = T \cap C_0$ , alors puisque  $T$  est un ensemble de points de transmissions temporellement indépendants, on obtient  $TC_0 \cap TC'_0 = \emptyset$ . On a aussi :

$$ED(TC_0 \cup TC'_0) = D \cap E_0 \text{ et } D' \cap \bigcup_{i>0} E_i = D \cap \bigcup_{i>0} E_i, \text{ donc } D \text{ se décompose en}$$

$$\text{une union disjointe } D = ED(TC_0 \cup TC'_0) \cup (D' \cap \bigcup_{i>0} E_i). \quad \text{(xiii)}$$

Pour tout  $j$  tel que  $m + k \geq j \geq 0$ , on pose :  $\hat{d}^j(TC'_0) = \hat{d}^j(\overline{TC'_0})$ , la correspondance entre les éléments de ces deux familles ne peut pas être faite en utilisant l'égalité des indices dans deux familles, car ils sont tous distincts, mais elle est faite en utilisant la relation  $\text{indice}_{\text{famille1}} = \underbrace{\text{indice}_{\text{famille2}}}$ . On pose de manière arbitraire

$\hat{d}^{m+k+1}(TC'_0) = 0$ . L'application de  $R(0)$  pour l'indice  $m+k+1$  donne alors :

$\exists \hat{d}^{m+k+1}(D \cap E_0), \hat{d}^{m+k}(D \cap E_0), \dots, \hat{d}^0(D \cap E_0)$  telles que si on impose sur les

entrées primaires pour tout  $i$  tel que  $m+k+1 \geq i \geq 0$  :  $\hat{x}^i(D \cap E_0) = \hat{d}^i(D \cap E_0)$ ,

alors pour tout  $j$  tel que  $m+k+1 \geq j \geq 0$  on a les égalités suivantes :

$\hat{x}^j(TC_0 \cup TC'_0) = \hat{d}^j(TC_0 \cup TC'_0)$ , en particulier  $\hat{x}^j(TC_0) = \hat{d}^j(TC_0)$ . (xiv)

Par construction, et par définition de  $TC'_0$ , on a aussi pour tout  $j$  tel que  $m+k \geq j \geq 0$  :

$\hat{x}^j(D' \cap EB) = \hat{d}^j(D' \cap EB)$ . (xv)

Soient  $\hat{d}^0(D' \cap \bigcup_{i>0} E_i) = 0$  (pour pouvoir appliquer  $R(m)$  au nouveau mode global),

et pour tout  $j$  tel que  $m+1+k \geq j \geq 1$  :  $\hat{d}^j(D' \cap \bigcup_{i>0} E_i) = \hat{d}^{j-1}(D' \cap \bigcup_{i>0} E_i)$ . (xvi)

Si pour tout  $j$  tel que  $m+1+k \geq j \geq 0$ ,  $\hat{x}^j(D) = \hat{d}^j(D)$ , alors (xiv) et (xv) sont vérifiées et d'après (xiii) et (xvi) pour tout  $j$  tel que  $1+m+k \geq j \geq 1$  :

$\hat{x}^j(D' \cap \bigcup_{i>0} E_i) = \hat{d}^{j-1}(D' \cap \bigcup_{i>0} E_i)$ , soit encore par changement d'origine des

temps, pour tout  $j$  tel que  $m+k \geq j \geq 0$  :  $\hat{x}^j(D' \cap \bigcup_{i>0} E_i) = \hat{d}^j(D' \cap \bigcup_{i>0} E_i)$ . (xvii)

(xv) et (xvii) donnent (xi) donc (xii). Sachant que  $T = TC_0 \cup T'$ , (xiv) et (xii) impli-

quent que pour tout  $j$  tel que  $m+1+k \geq j \geq m+1$  :  $\hat{x}^j(T) = \hat{d}^j(T)$ .  $R(m+1)$  est donc démontrée. Par récurrence,  $R(m)$  est vraie pour tout  $m$ , c'est-à-dire que tout ensemble de points de transmission temporellement indépendants du mode global est contrôlable en continu.



## **V. Application au test structurel**

### **V.1 Schéma d'utilisation des théorèmes sur les propriétés de scan**

L'équivalent pour un mode global du passage en mode scan dans une technique de scan est l'activation de ce mode global. Les étapes de décalage en entrée et en sortie, permettent aux modes globaux ayant des propriétés d'accessibilité faible ou forte de propager des vecteurs de test combinatoires locaux aux bascules de transmission à partir de ses entrées primaires, et d'observer ensuite les résultats de leur application à ses sorties primaires. Cependant, les vecteurs de test combinatoires locaux sont propagés d'une façon différente par rapport au scan. En effet, on ne retrouve plus forcément à chaque coup d'horloge les valeurs à propager en d'autres points du circuit, mais elles peuvent subir des transformations bijectives quelconques tout en se propageant. La conservation de l'information se fait de manière globale pour des groupes de bascules et non plus obligatoirement d'une bascule à une autre comme dans le cas de la méthode de scan. Cela rend complexe la mise en œuvre du contrôle et de l'observation de l'état du mode global. De plus, lorsqu'on a seulement la propriété d'accessibilité faible, on est confronté à des problèmes d'accessibilité pour les bascules de rebouclage qui ne sont pas de transmission, et il faut donc essayer de les atténuer ou même de les supprimer quand cela est possible.

### **V.2 Mise en œuvre des contrôles et de l'observation pour le mode global**

#### **V.2.1 Algorithme de mise en œuvre du contrôle instantané**

Soit un mode global vérifiant les hypothèses du théorème de contrôlabilité faible. En corollaire de la démonstration du théorème, si on connaît l'état initial des sorties des bascules du mode global, tout état des points de transmission du mode global peut être obtenu en au plus  $m$  coups d'horloge, où  $m = \max_{ml \in ML} (\{DE(ml)\})$ . C'est-à-dire les sorties des bascules de transmission sont elles contrôlées au coup d'horloge suivant (traversée des bascules par les données). La méthode utilisée dans la démonstration est à la fois constructive et récursive, et donne implicitement l'ordre dans lequel il suffit d'appliquer les fonctions d'activation des modes locaux et aussi l'ordre dans lequel il suffit d'inverser ces fonctions après avoir calculé l'état des points d'entrée qui servent de paramètres. On présente ici un algorithme de mise en œuvre itératif donnant toutes les opérations élémentaires à effectuer (dans V.2.2, on montre sur un exemple, l'application de cet algorithme). Les opérations élémentaires sont :

- La fixation arbitraire à 0 : employée quand il y a un choix arbitraire de valeurs à faire, elle affecte l'ensemble de points considérés.

- L'application des fonctions d'activation : employée quand on veut évaluer les valeurs des points de sortie des modes locaux en fonction des valeurs d'entrée lorsque ces modes sont activés, elle affecte un ensemble de points de sortie.
- La traversée en avant des bascules : employée pour avoir à un instant courant les valeurs de sorties de bascules en fonction des valeurs de leurs entrées à l'instant précédent, elle affecte un ensemble de points d'entrée de type bascule.
- L'inversion des fonctions : employée quand on veut évaluer les valeurs de points d'entrée des modes locaux en fonction des valeurs de transmission lorsque ces modes sont activés et que les valeurs des points d'entrée utilisés comme paramètres sont connus, elle affecte un ensemble de points d'entrée.
- La traversée en arrière des bascules : employée pour avoir à un instant courant les valeurs d'entrées de bascules en fonction des valeurs de leurs sorties à l'instant suivant, elle affecte un ensemble de points de sortie de type bascule.
- L'affectation à la valeur ciblée : employée pour initialiser des valeurs à un instant courant en fonction de valeurs données et ciblées par le concepteur, elle affecte l'ensemble de points considérés.

Pour alléger l'écriture, on emploie les abréviations suivantes : FA0 pour fixation arbitraire à 0, FN pour application des fonctions d'activation, BAV pour traversée en avant des bascules, INV pour inversion des fonctions d'activation, BAR pour traversée en arrière des bascules, et AFF pour affectation à la valeur ciblée. Le format utilisé pour la description des étapes élémentaires est le suivant :

instant : opération : points concernés

Si  $m > 0$ , la première étape est appelée étape de propagation en avant de l'état initial. On simule partiellement le mode global activé de l'instant 0 à  $m - 1$ . A l'instant 0, les valeurs des sorties des bascules du mode global sont celles de l'état initial.

Pour  $i$  variant de 0 à  $m - 1$  faire {

$$i : \text{FA0} : \bigcup_{j > i} E_j \cup \left( \bigcup_{\substack{ml \in ML \\ DE(ml) > i}} \text{PEB}(ml) \cap \overbrace{\bigcup_{\substack{ml \in ML \\ DE(ml) < i}} \text{PTB}(ml)} \right)$$

$$i : \text{FN} : \bigcup_{\substack{ml \in ML \\ DE(ml) > i}} \text{PSB}(ml)$$

$$i + 1 : \text{BAV} : \overbrace{\bigcup_{\substack{ml \in ML \\ DE(ml) > i}} \text{PSB}(ml)}$$

}

La deuxième étape est appelée étape de propagation arrière des valeurs ciblées. On part des résultats précédents de simulation, et on détermine les valeurs de simulation restantes. On traite les points associés aux modes locaux de distance aux entrées égale à  $i$  dans la phase  $i$ . Dans une phase  $i$ , par des suites d'inversions des fonctions d'activation, on détermine les valeurs aux instants où elles ne sont pas connues en partant des instants les plus petits jusqu'à l'instant  $m$  où on affecte les valeurs ciblées.

*/\* phase m \*/*

$$m : \text{AFF} : \bigcup_{\substack{ml \in ML \\ DE(ml) = m}} \text{PT}(ml) = \bigcup_{\substack{ml \in ML \\ DE(ml) = m}} \text{PTP}(ml)$$

$$m : \text{INV} : \bigcup_{\substack{ml \in ML \\ DE(ml) = m}} \text{PE}(ml) - \text{PER}(ml)$$

*/\* autres phases \*/*

Pour  $i$  variant de  $m-1$  à  $0$  faire {

*/\* phase i \*/*

Pour  $j$  variant de  $i$  à  $m-1$  faire {

$$j : \text{BAR} : \bigcup_{\substack{ml \in ML \\ DE(ml) = i}} \text{PTB}(ml)$$

$$j : \text{FA0} : \bigcup_{\substack{ml \in ML \\ DE(ml) = i}} \text{PTP}(ml)$$

$$j : \text{INV} : \bigcup_{\substack{ml \in ML \\ DE(ml) = i}} \text{PE}(ml) - \text{PER}(ml)$$

$$j : \text{FN} : \bigcup_{\substack{ml \in ML \\ DE(ml) = i}} \text{PSB}(ml) - \text{PTB}(ml)$$

$$j+1 : \text{BAV} : \bigcup_{\substack{ml \in ML \\ DE(ml) = i}} \overbrace{PSB(ml) - PTB(ml)}$$

}

$$m : \text{AFF} : \bigcup_{\substack{ml \in ML \\ DE(ml) = i}} PT(ml)$$

$$m : \text{INV} : \bigcup_{\substack{ml \in ML \\ DE(ml) = i}} PE(ml) - PER(ml)$$

}

### V.2.2 Exemple d'application

Voici un exemple de mode global comprenant six modes locaux : A, B, C, D, E, et F. Leurs fonctions d'activation sont respectivement :  $F_A$ ,  $F_B$ ,  $F_C$ ,  $F_D$ ,  $F_E$ , et  $F_F$ . Les ensembles de leurs points d'entrée primaire sont respectivement :  $E_A$ ,  $E_B$ ,  $E_C$ ,  $E_D$ ,  $E_E$ , et  $E_F$ . Les ensembles de leurs points de sortie primaire sont respectivement :  $S_A$ ,  $S_B$ ,  $S_C$ ,  $S_D$ ,  $S_E$ , et  $S_F$ , on suppose qu'ils sont aussi des points de transmission. Soit I l'un de ces modes locaux,  $R_I$  est l'ensemble de ses bascules de rebouclage qui ne sont pas des bascules de transmission. Si J est un autre de ces modes locaux alors  $RT_{IJ}$  est l'ensemble des bascules de rebouclage de I qui sont aussi des bascules de transmission de I et des bascules d'entrée de J. Et  $T_{IJ}$  est l'ensemble des bascules de transmission de I qui ne sont pas des bascules de rebouclage de I et qui sont des bascules d'entrée de J. On notera de la même façon les ensembles de bascules et les ensembles de leurs sorties, pour noter les ensembles de leurs entrées on soulignera les noms des ensembles. On suppose le mode global surjectif et activable. Le schéma suivant du mode global décrivant les interconnexions entre les modes locaux, montre que le mode est aussi non rebouclant, disjoint et qu'il a les propriétés de propagation avant et arrière. Les valeurs de la distance aux entrées pour A, B, C, D, E et F sont respectivement : 0, 1, 2, 3, 3 et 4.

Etape de propagation avant de l'état initial.

0 : FA0 :  $E_B, E_C, E_D, E_E, E_F$

0 : FN :  $\underline{R_B}, \underline{R_C}, \underline{R_D}, \underline{R_E}, \underline{R_F}, \underline{RT_{BC}}, \underline{T_{BC}}, \underline{RT_{CD}}, \underline{T_{CD}}, \underline{RT_{CE}}, \underline{T_{CE}}, \underline{RT_{EF}}, \underline{T_{EF}}$

1 : BAV :  $R_B, R_C, R_D, R_E, R_F, RT_{BC}, T_{BC}, RT_{CD}, T_{CD}, RT_{CE}, T_{CE}, RT_{EF}, T_{EF}$

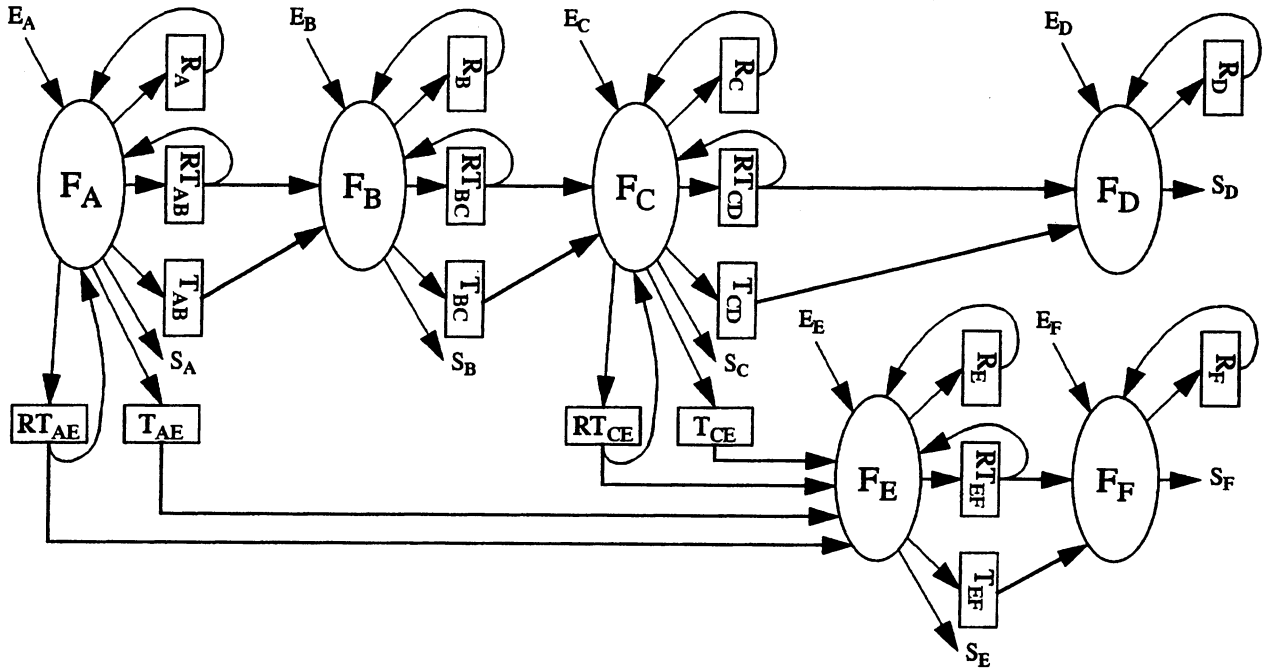


Figure 9 : représentation graphique de l'exemple de mode global

- 1 : FA0 :  $E_C, E_D, E_E, E_F, RT_{AE}, T_{AE}$   
 1 : FN :  $R_C, R_D, R_E, R_F, RT_{CD}, T_{CD}, RT_{CE}, T_{CE}, RT_{EF}, T_{EF}$   
 2 : BAV :  $R_C, R_D, R_E, R_F, RT_{CD}, T_{CD}, RT_{CE}, T_{CE}, RT_{EF}, T_{EF}$   
 2 : FA0 :  $E_D, E_E, E_F, RT_{AE}, T_{AE}$   
 2 : FN :  $R_D, R_E, R_F, RT_{EF}, T_{EF}$   
 3 : BAV :  $R_D, R_E, R_F, RT_{EF}, T_{EF}$   
 3 : FA0 :  $E_F$   
 3 : FN :  $R_F$   
 4 : BAV :  $R_F$

Etape de propagation arrière des valeurs ciblées.

Phase 4 :

- 4 : AFF :  $S_F$   
 4 : INV :  $E_F, RT_{EF}, T_{EF}$

Phase 3 :

- 3 : BAR :  $RT_{EF}, T_{EF}$   
 3 : FA0 :  $S_E, S_D$

- 3 : INV :  $RT_{CD}, T_{CD}, RT_{CE}, T_{CE}, RT_{AE}, T_{AE}, E_E, E_D$   
 3 : FN :  $\underline{R}_D, \underline{R}_E$   
 4 : BAV :  $R_D, R_E$   
 4 : AFF :  $S_D, S_E, \underline{RT}_{EF}, \underline{T}_{EF}$   
 4 : INV :  $RT_{CD}, T_{CD}, RT_{CE}, T_{CE}, RT_{AE}, T_{AE}, E_E, E_D$

## Phase 2 :

- 2 : BAR :  $\underline{RT}_{CD}, \underline{T}_{CD}, \underline{RT}_{CE}, \underline{T}_{CE}$   
 2 : FA0 :  $S_C$   
 2 : INV :  $RT_{BC}, T_{BC}, E_C$   
 2 : FN :  $\underline{R}_C$   
 3 : BAV :  $R_C$   
 ... : ... : ...  
 4 : AFF :  $S_C, \underline{RT}_{CD}, \underline{T}_{CD}, \underline{RT}_{CE}, \underline{T}_{CE}$   
 4 : INV :  $RT_{BC}, T_{BC}, E_C$

## Phase 1 :

- 1 : BAR :  $\underline{RT}_{BC}, \underline{T}_{BC}$   
 1 : FA0 :  $S_B$   
 1 : INV :  $RT_{AB}, T_{AB}, E_B$   
 1 : FN :  $\underline{R}_B$   
 2 : BAV :  $R_B$   
 ... : ... : ...  
 4 : AFF :  $S_B, \underline{RT}_{BC}, \underline{T}_{BC}$   
 4 : INV :  $RT_{AB}, T_{AB}, E_B$

## Phase 0 :

- 0 : BAR :  $\underline{RT}_{AB}, \underline{T}_{AB}, \underline{RT}_{AE}, \underline{T}_{AE}$   
 0 : FA0 :  $S_A$   
 0 : INV :  $E_A$

0	:	FN	:	$\underline{R}_A$
1	:	BAV	:	$R_A$
...	:	...	:	...
4	:	AFF	:	$S_A, \underline{RT}_{AB}, \underline{T}_{AB}, \underline{RT}_{AE}, \underline{T}_{AE}$
4	:	INV	:	$E_A$

### V.2.3 Algorithme de mise en œuvre du contrôle continu

Soit un mode global vérifiant les hypothèses du théorème de contrôlabilité continue. En corollaire de la démonstration du théorème, si on connaît l'état initial des sorties des bascules du mode global, pour tout ensemble T de points de transmission temporellement indépendants du mode global, toute séquence de valeurs en ces points peut être obtenue après au plus m coups d'horloge, où m est la latence de T. Par conséquent on contrôle de fait les sorties des bascules associées aux points de transmission de T de type bascule avec une latence m + 1 (traversée des bascules par les données). La méthode utilisée dans la démonstration est à la fois constructive et récursive, et donne implicitement l'ordre dans lequel il suffit d'appliquer les fonctions d'activation des modes locaux et aussi l'ordre dans lequel il suffit d'inverser ces fonctions après avoir calculé l'état des points d'entrée qui servent de paramètres. On présente ici un algorithme de mise en œuvre itératif donnant toutes les opérations élémentaires à effectuer (dans V.2.4, on montre un exemple d'application de cet algorithme). Les opérations élémentaires utilisées sont les mêmes que celles utilisées pour le contrôle instantané (V.2.1).

Si  $m > 0$ , la première étape est l'étape de propagation avant de l'état initial, identique à celle utilisée pour le contrôle instantané. La définition de m ne peut que réduire le nombre d'opérations utilisées par rapport au contrôle instantané. On peut encore réduire ce nombre d'opérations en restreignant les opérations aux modes locaux qui sont dans  $\text{supp}(T)$ , c'est-à-dire on remplace ML dans les expressions des ensembles de nœuds par  $\text{supp}(T)$ .

La deuxième étape est l'étape de propagation arrière des séquences de valeurs ciblées. On part des résultats précédents de simulation, et on détermine les valeurs de simulation restantes (pour les instants 0 à m + k - 1 où k est la longueur de la séquence ciblée, on supposera  $k > 1$  sinon on est dans un cas particulier du contrôle instantané). On traite les points associés aux modes locaux de distance aux entrées égale à i dans la phase i. Dans une phase i, par des suites d'inversions des fonctions d'activation, on

détermine les valeurs aux instants où elles ne sont pas connues en partant des instants les plus petits jusqu'aux instants supérieurs à  $m$  où on affecte les valeurs ciblées.

/\* phase  $m$  \*/

Pour  $j$  variant de  $m$  à  $m + k - 1$  faire {

$$j : \text{AFF} : \begin{array}{l} \bigcup_{\substack{ml \in \text{supp}(T) \\ DE(ml) = m}} PT(ml) \cap T = \bigcup_{\substack{ml \in ML \\ DE(ml) = m}} PT(ml) \cap T \end{array}$$

$$j : \text{FA0} : \begin{array}{l} \bigcup_{\substack{ml \in \text{supp}(T) \\ DE(ml) = m}} PT(ml) - T \end{array}$$

$$j : \text{INV} : \begin{array}{l} \bigcup_{\substack{ml \in \text{supp}(T) \\ DE(ml) = m}} PE(ml) - PER(ml) \end{array}$$

$$j : \text{FN}^6 : \begin{array}{l} \bigcup_{\substack{ml \in \text{supp}(T) \\ DE(ml) = m}} PSB(ml) - PTB(ml) \end{array}$$

$$j + 1 : \text{BAV}^7 : \begin{array}{l} \bigcup_{\substack{ml \in \text{supp}(T) \\ DE(ml) = m}} \overbrace{PSB(ml) - PTB(ml)} \end{array}$$

}

/\* autres phases \*/

Pour  $i$  variant de  $m-1$  à  $0$  faire {

/\* phase  $i$  \*/

Pour  $j$  variant de  $i$  à  $m-1$  faire {

$$j : \text{BAR} : \begin{array}{l} \bigcup_{\substack{ml \in \text{supp}(T) \\ DE(ml) = i}} PTB(ml) \cap \bigcup_{\substack{ml \in \text{supp}(T) \\ DE(ml) > i}} \overbrace{PEB(ml) - PER(ml)} \end{array}$$

$$j : \text{FA0} : \begin{array}{l} \bigcup_{\substack{ml \in \text{supp}(T) \\ DE(ml) = i}} PT(ml) - \bigcup_{\substack{ml \in \text{supp}(T) \\ DE(ml) > i}} \overbrace{PEB(ml) - PER(ml)} \end{array}$$

6. Opération inutile et qui peut être supprimée si  $j=k+m-1$

7. Cf note 6 page 111



$$j : \text{INV} : \bigcup_{\substack{ml \in \text{supp}(T) \\ DE(ml) = i}} \text{PEB}(ml) - \text{PER}(ml)$$

$$j : \text{FN} : \bigcup_{\substack{ml \in \text{ML} \\ DE(ml) = i}} \text{PSB}(ml) - \text{PTB}(ml)$$

$$j+1 : \text{BAV} : \bigcup_{\substack{ml \in \text{ML} \\ DE(ml) = i}} \overbrace{\text{PSB}(ml) - \text{PTB}(ml)}$$

}

Pour j variant de m à m + k - 1 faire {

$$j : \text{AFF} : \bigcup_{\substack{ml \in \text{supp}(T) \\ DE(ml) = i}} \text{PT}(ml) \cap T$$

$$j : \text{BAR} : \bigcup_{\substack{ml \in \text{supp}(T) \\ DE(ml) = i}} \text{PTB}(ml) \cap \bigcup_{\substack{ml \in \text{supp}(T) \\ DE(ml) > i}} \overbrace{\text{PEB}(ml) - \text{PER}(ml)}$$

$$j : \text{FA0} : \bigcup_{\substack{ml \in \text{supp}(T) \\ DE(ml) = i}} \text{PT}(ml) - T - \bigcup_{\substack{ml \in \text{supp}(T) \\ DE(ml) > i}} \overbrace{\text{PEB}(ml) - \text{PER}(ml)}$$

$$j : \text{INV} : \bigcup_{\substack{ml \in \text{supp}(T) \\ DE(ml) = i}} \text{PEB}(ml) - \text{PER}(ml)$$

$$j : \text{FN}^8 : \bigcup_{\substack{ml \in \text{ML} \\ DE(ml) = i}} \text{PSB}(ml) - \text{PTB}(ml)$$

$$j+1 : \text{BAV}^9 : \bigcup_{\substack{ml \in \text{ML} \\ DE(ml) = i}} \overbrace{\text{PSB}(ml) - \text{PTB}(ml)}$$

}

}

<sup>8</sup>. Cf note 6 page 111

<sup>9</sup>. Cf note 6 page 111

### V.2.4 Exemple d'application

On considère le même mode global que celui de l'exemple en V.2.2. L'ensemble des points appartenant à l'union de  $\underline{RT}_{CD}$ ,  $\underline{T}_{CD}$ ,  $\underline{RT}_{EF}$ ,  $\underline{T}_{EF}$  et  $S_C$  est pris comme ensemble de points de transmission temporellement indépendants. La latence vaut alors 3. On considère ici une séquence des valeurs ciblées de longueur 3.

L'étape de propagation avant de l'état initial est identique à celle de l'exemple en V.2.2, on peut cependant la restreindre aux instants 0 à 3.

Etape de propagation arrière des séquences de valeurs ciblées.

Phase 3 :

- 3 : AFF :  $\underline{RT}_{EF}$ ,  $\underline{T}_{EF}$
- 3 : FA0 :  $S_E$
- 3 : INV :  $RT_{CE}$ ,  $T_{CE}$ ,  $RT_{AE}$ ,  $T_{AE}$ ,  $E_E$
- 3 : FN :  $\underline{R}_D$ ,  $\underline{R}_E$
- 4 : BAV :  $R_D$ ,  $R_E$
- 4 : AFF :  $\underline{RT}_{EF}$ ,  $\underline{T}_{EF}$
- 4 : FA0 :  $S_E$
- 4 : INV :  $RT_{CE}$ ,  $T_{CE}$ ,  $RT_{AE}$ ,  $T_{AE}$ ,  $E_E$
- 4 : FN :  $\underline{R}_D$ ,  $\underline{R}_E$
- 5 : BAV :  $R_D$ ,  $R_E$
- 5 : AFF :  $\underline{RT}_{EF}$ ,  $\underline{T}_{EF}$
- 5 : FA0 :  $S_E$
- 5 : INV :  $RT_{CE}$ ,  $T_{CE}$ ,  $RT_{AE}$ ,  $T_{AE}$ ,  $E_E$

Phase 2 :

- 2 : BAR :  $\underline{RT}_{CE}$ ,  $\underline{T}_{CE}$
- 2 : FA0 :  $\underline{RT}_{CD}$ ,  $\underline{T}_{CD}$ ,  $S_C$
- 2 : INV :  $RT_{BC}$ ,  $T_{BC}$ ,  $E_C$
- 2 : FN :  $\underline{R}_C$
- 3 : BAV :  $R_C$

3 : AFF :  $\underline{RT}_{CD}, \underline{T}_{CD}$   
 3 : BAR :  $\underline{RT}_{CE}, \underline{T}_{CE}$   
 3 : FA0 :  $S_C$   
 3 : INV :  $RT_{BC}, T_{BC}, E_C$   
 3 : FN :  $\underline{R}_C$   
 4 : BAV :  $R_C$   
 ... : ... : ...  
 5 : AFF :  $\underline{RT}_{CD}, \underline{T}_{CD}$   
 5 : BAR :  $\underline{RT}_{CE}, \underline{T}_{CE}$   
 5 : FA0 :  $S_C$   
 5 : INV :  $RT_{BC}, T_{BC}, E_C$

## Phase 1 :

1 : BAR :  $\underline{RT}_{BC}, \underline{T}_{BC}$   
 1 : FA0 :  $S_B$   
 1 : INV :  $RT_{AB}, T_{AB}, E_B$   
 1 : FN :  $\underline{R}_B$   
 2 : BAV :  $R_B$   
 ... : ... : ...  
 5 : BAR :  $\underline{RT}_{BC}, \underline{T}_{BC}$   
 5 : FA0 :  $S_B$   
 5 : INV :  $RT_{AB}, T_{AB}, E_B$

## Phase 0 :

0 : BAR :  $\underline{RT}_{AB}, \underline{T}_{AB}, \underline{RT}_{AE}, \underline{T}_{AE}$   
 0 : FA0 :  $S_A$   
 0 : INV :  $E_A$   
 0 : FN :  $\underline{R}_A$   
 1 : BAV :  $R_A$

... : ... : ...  
 5 : BAR :  $\underline{RT}_{AB}, \underline{T}_{AB}, \underline{RT}_{AE}, \underline{T}_{AE}$   
 5 : FA0 :  $S_A$   
 5 : INV :  $E_A$

### V.2.5 Algorithme de mise en œuvre de l'observation

Les deux algorithmes précédents pour la mise en œuvre du contrôle instantané et continu fournissent un moyen de déterminer les valeurs à affecter aux entrées pour obtenir les valeurs ciblées au niveau des points de transmission d'un mode global. En ce qui concerne l'observation des résultats en utilisant un mode global activé, ceux-ci peuvent être obtenus par simulation simple. Pour un mode global vérifiant les hypothèses des théorèmes d'observabilité faible ou forte, toute faute d'un circuit qui a été propagée aux seuls points de transmission (et non pas aux points de sortie qui ne sont pas de transmission), sera détectée avec un retard d'au plus  $m$  coups d'horloge où  $m$  est la distance aux sorties (ou entrées ce qui est équivalent) maximale des modes locaux. On suppose que la simulation démarre avec un état initial des points d'entrée du mode global connu, elle doit durer  $m$  coups d'horloge pour que toute variation sur cet état initial soit détectée sur les sorties primaires du mode global. Tout en conservant ce résultat, on cherche à réduire le nombre de nœuds touchés par les opérations élémentaires d'application des fonctions d'activation et de traversée en avant des bascules. A partir de l'opérateur DE (distance aux entrées) défini pour les modes locaux et de la structure du mode global, on définit un nouvel opérateur DT (distance de transmission) qui permettra de sélectionner plus finement les nœuds qu'il est nécessaire de simuler.

Si  $t$  est l'instant où l'on veut observer toutes les entrées du mode global, un mode local  $m_l$  doit alors être observé à cet instant  $t$  et à tous les instants où il sert à propager les valeurs de ses prédécesseurs d'ordre quelconque. Soit  $t + dprop(m_l)$  l'instant final de l'observation, alors pour un mode sans prédécesseur on a :  $dprop(m_l) = 0$ , sinon on a :

$$dprop(m_l) = 1 + \max_{m_l' \in pred(m_l)} (dprop(m_l'))$$

Donc d'après la définition de DE, on a :  $dprop(m_l) = DE(m_l)$ .

Pour pouvoir calculer les valeurs des points de sortie d'un mode local à un instant donné, il faut avoir déjà calculé ou fixé les valeurs à cet instant pour les points d'entrée du mode local et donc à l'instant précédent pour ses points de sortie rebouclants et pour les points de transmission de type bascule associés à ses modes locaux prédécesseurs. L'instant final  $t + DT(m_l)$  jusqu'auquel un mode local doit transmettre des données

calculées ou fixées à l'un de ses modes successeurs, est calculé de proche en proche de la façon suivante : pour un mode local  $ml$  qui n'a pas de successeur  $DT(ml) = DE(ml)$ , sinon  $DT(ml) = \max_{ml' \in \text{succ}(ml)} (DT(ml')) - 1$ .  $DT$  est appelée la distance de transmission. D'après la définition de  $DE$ , pour tout mode local  $ml$  on a :  $DT(ml) \geq DE(ml)$ . Si  $ml'$  est un mode local successeur du mode local  $ml$  associé à la bascule de transmission  $b$ , alors on définit aussi pour l'entrée  $n$  de  $b$ ,  $DT(n)$  comme  $DT(ml') - 1$ . Si  $n$  est un point de transmission primaire d'un mode local  $ml$  alors  $DT(n)$  est défini comme  $DE(ml)$ .

On remarquera que  $DT(ml) = \max_{n \in PT(ml)} (DT(n))$  et surtout que la valeur d'un point  $n$  de transmission doit être calculée si on est à un instant inférieur ou égal à  $t + DT(n)$ .

L'algorithme pour la mise en œuvre de l'observation est alors le suivant :

Pour  $i$  variant de 0 à  $m$  faire {

$$i : FN : \bigcup_{\substack{ml \in ML \\ DE(ml) \geq i}} PTP(ml) \cup \bigcup_{\substack{ml \in ML \\ DT(ml) > i}} \underbrace{PEB(ml) - PER(ml)} \cup PSR(ml)$$

$$i + 1 : BAV^{10} : \bigcup_{\substack{ml \in ML \\ DT(ml) > i}} (PEB(ml) - PER(ml)) \cup \overbrace{PSR(ml)}$$

$$i + 1 : FA0^{11} : \bigcup_{\substack{ml \in ML \\ DT(ml) > i}} PEP(ml)$$

}

### V.2.6 Exemple d'application

On considère le même exemple que celui utilisé pour la mise en œuvre du contrôle instantané au V.2.2 page 107. Les valeurs de la distance de transmission des modes locaux A, B, C, D, E et F sont respectivement 2, 1, 2, 3, 3 et 4. La simulation pour l'observation se résume alors en :

$$0 : FN : \underline{R_A}, \underline{R_B}, \underline{R_C}, \underline{R_D}, \underline{R_E}, \underline{R_F}, \underline{RT_{AB}}, \underline{T_{AB}}, \underline{RT_{AE}}, \underline{T_{AE}}, \underline{RT_{BC}}, \underline{T_{BC}}, \underline{RT_{CD}}, \underline{T_{CD}}, \underline{RT_{CE}}, \underline{T_{CE}}, \underline{RT_{EF}}, \underline{T_{EF}}, S_A, S_B, S_C, S_D, S_E, S_F$$

<sup>10</sup>. L'ensemble des points concernés est vide si  $i=m$

<sup>11</sup>. Cf note 10 page 116

- 1 : BAV :  $R_A, R_B, R_C, R_D, R_E, R_F, RT_{AB}, T_{AB}, RT_{AE}, T_{AE}, RT_{BC}, T_{BC}, RT_{CD}, T_{CD}, RT_{CE}, T_{CE}, RT_{EF}, T_{EF}$
- 1 : FA0 :  $E_A, E_B, E_C, E_D, E_E, E_F$
- 1 : FN :  $\underline{R}_A, \underline{R}_C, \underline{R}_D, \underline{R}_E, \underline{R}_F, \underline{RT}_{AB}, \underline{RT}_{AE}, \underline{T}_{AE}, \underline{RT}_{BC}, \underline{T}_{BC}, \underline{RT}_{CD}, \underline{T}_{CD}, \underline{RT}_{CE}, \underline{T}_{CE}, \underline{RT}_{EF}, \underline{T}_{EF}, S_B, S_C, S_D, S_E, S_F$
- 2 : BAV :  $R_A, R_C, R_D, R_E, R_F, RT_{AB}, RT_{AE}, T_{AE}, RT_{BC}, T_{BC}, RT_{CD}, T_{CD}, RT_{CE}, T_{CE}, RT_{EF}, T_{EF}$
- 2 : FA0 :  $E_A, E_C, E_D, E_E, E_F$
- 2 : FN :  $\underline{R}_D, \underline{R}_E, \underline{R}_F, \underline{RT}_{AE}, \underline{T}_{AE}, \underline{RT}_{CD}, \underline{T}_{CD}, \underline{RT}_{CE}, \underline{T}_{CE}, \underline{RT}_{EF}, \underline{T}_{EF}, S_C, S_D, S_E, S_F$
- 3 : BAV :  $R_D, R_E, R_F, RT_{AE}, T_{AE}, RT_{CD}, T_{CD}, RT_{CE}, T_{CE}, RT_{EF}, T_{EF}$
- 3 : FA0 :  $E_D, E_E, E_F$
- 3 : FN :  $\underline{R}_F, \underline{RT}_{EF}, \underline{T}_{EF}, S_D, S_E, S_F$
- 4 : BAV :  $R_F, RT_{EF}, T_{EF}$
- 4 : FA0 :  $E_F$
- 4 : FN :  $S_F$

### V.2.7 Améliorations des mises en œuvre

En fonction de la structure du mode global par rapport à ses bascules de rebouclage, il est possible de simplifier l'étape de propagation avant de l'état initial en diminuant le nombre de points concernés pour chacune des opérations élémentaires. En effet, cette étape sert à calculer pour tout mode local  $m_l$ , les valeurs de ses points d'entrée rebouclants à l'instant  $DE(m_l)$ . Si un mode local est sans point de sortie rebouclant et s'il en est de même pour tous ses modes successeurs d'ordre quelconque alors il n'est pas nécessaire de calculer les valeurs de ses points de sortie, ni les valeurs de ses points d'entrée qui ne sont pas des sorties de bascules de rebouclage. Inversement si un mode local a au moins un point de sortie rebouclant, alors il faudra calculer de la même façon, les valeurs de ses points de sortie rebouclants. S'il a des modes successeurs d'ordre quelconque ayant au moins un point de sortie rebouclant, alors il faudra calculer de la même façon, les valeurs de ses points de sortie rebouclants et de ses points de transmission qui sont des entrées des bascules de transmission associées aux successeurs d'ordre quelconque où il y a des points de sortie rebouclants. Pour tout mode local  $m_l$ , on définit  $SR(m_l)$  la valeur booléenne appelée "successeur rebouclant", elle vaut 1 si  $m_l$

a des points de sortie rebouclants ou si l'un de ses successeurs d'ordre quelconque en a, et vaut 0 sinon.  $SR(ml)$  peut être calculée de proche en proche : si  $ml$  est sans successeur, on pose  $SR(ml) = 1$  quand  $PSR(ml) \neq \emptyset$  et  $SR(ml) = 0$  quand  $PSR(ml) = \emptyset$ , sinon on a  $SR(ml) = \max_{ml' \in succ(ml)} (SR(ml'))$ .

Soient  $MLR = \bigcup_{\substack{ml \in ML \\ SR(ml) = 1}} ml$ , et

$PTBR(ml) = \bigcup_{\substack{ml' \in MLR \\ ml' \in succ(ml)}} \underbrace{PEB(ml') - PER(ml')} \cap PTB(ml)$ , l'étape de propa-

gation avant de l'état initial devient alors :

Pour  $i$  variant de 0 à  $m - 1$  faire {

$i : FA0 : \bigcup_{\substack{ml \in MLR \\ DE(ml) > i}} PEP(ml) \cup \left( \bigcup_{\substack{ml \in MLR \\ DE(ml) > i}} PEB(ml) \cap \overbrace{\bigcup_{\substack{ml \in MLR \\ DE(ml) < i}} PTB(ml)} \right)$

$i : FN : \bigcup_{\substack{ml \in MLR \\ DE(ml) > i}} PSR(ml) \cup PTBR(ml)$

$i + 1 : BAV : \bigcup_{\substack{ml \in MLR \\ DE(ml) > i}} \overbrace{PSR(ml) \cup PTBR(ml)}$

}

Dans l'exemple explicité au V.2.2 page 107, il n'y a pas de changement car a priori on n'a rien exprimé sur le caractère vide ou non des ensembles contenant des bascules de rebouclage. Les raisonnements précédents donnent aussi implicitement qu'il suffit de connaître l'état initial seulement pour les points suivants :

$\bigcup_{ml \in MLR} \overbrace{PSR(ml) \cup PTBR(ml)}$ . Si on s'intéresse au contrôle continu pour un

ensemble  $T$  de points de transmission temporellement indépendants, alors il suffit de remplacer  $MLR$  par  $MLR \cap \text{supp}(T)$  dans la définition de  $PTBR(ml)$  et dans les formules précédentes de la version améliorée de l'étape de propagation de l'initial page 118. Cela s'applique aussi à l'ensemble des points dont il suffit de connaître l'état initial. Dans le cas particulier d'un mode global non rebouclant, il n'y a plus aucune opération à effectuer dans cette première étape, on retrouve ainsi le résultat de la con-

trôlabilité forte.

Les processus de contrôle et d'observation qui utilisent un mode global se résument en l'application d'opérations élémentaires dans un ordre donné. Ces opérations sont du type symbolique en ce sens qu'elles s'appliquent à des valeurs booléennes quelconques. On a donc déterminé une fois pour toute la séquence des opérations à effectuer pour appliquer chaque vecteur de test effectif combinatoire local en sortie des bascules de transmission, et pour observer le résultat de son application. Il est donc intéressant de vouloir simplifier ces séquences symboliques de contrôle et d'observation, car tout gain d'opérations au niveau symbolique est répercuté au niveau effectif avec un facteur multiplicatif égal au nombre de vecteurs effectifs différents utilisés. Dans le cas de l'observation, l'intérêt est moindre si on utilise un simulateur de fautes pour vérifier la couverture finale, car la simulation est forcément refaite au niveau effectif et seule l'extraction des points d'entrée primaire du mode global et des instants auxquels il faut les fixer à 0 ou à une valeur définie est nécessaire pour simuler l'observation. Pour le contrôle, l'intérêt est essentiel car on a besoin de connaître la séquence de vecteurs de test à appliquer aux entrées primaires du mode global en fonction du vecteur effectif combinatoire local et de l'état initial. Des réductions symboliques dépendantes des fonctions d'application sont réalisables. En effet, si on considère une classe de fonctions d'application de forme simple pour lesquelles leurs composées ou leurs inverses ont aussi une forme simple, alors lorsqu'on combine celles-ci avec les opérations BAV, BAR et FA0, elles fournissent des expressions simples des valeurs en un point et à un instant donné  $t$  en fonction de valeurs en d'autres points en des instants distants de  $t$  de plus d'une unité. Ce type d'expression permet de gagner du temps grâce aux simplifications qui diminuent en particulier le nombre de variables intermédiaires et donc ce qui est équivalent le nombre d'opération du type BAV et BAR. Par exemple, soit la classe des fonctions qui opèrent sur les valeurs de un ou plusieurs points et qui les associent sans changement ou par simple conjugaison à d'autres points. Les fonctions inverses restent de la même forme ainsi que leurs composées (on pourrait utiliser des structures algébriques sur des ensembles de fonctions associées à des ensembles de points pour décrire ce genre de propriétés). Lorsqu'on les combine avec les opérations BAV, BAR, et FA0, on obtient des fonctions associées à des ensembles de points indexés par des instants, et qui si on ne tient plus compte des indices temporels sont soit des FA0, soit des fixations à 1, soit une fonction de la classe précédemment définie. En particulier si par exemple le mode global possède localement des groupements de modes locaux qui sont des registres à décalage, on retrouve localement après simplification les formules qui régissent les chaînes de scan, à savoir que les valeurs après  $n$



cours d'horloge en un point donné du registre sont celles qui se trouvaient à l'instant de départ au niveau de son prédécesseur d'ordre  $n$  (on suppose qu'il est encore dans le registre).

### V.2.8 Evaluation de la complexité

Soit un mode global donné, on peut le représenter de multiples façons. Une des représentations les plus naturelles est la constitution d'une liste (vue comme un ensemble) de modes locaux, et l'association pour chaque mode local des listes de points (points d'entrée, de sortie, de transmission, ...), ainsi qu'une fonction booléenne associée à ces points. Soit  $P$  l'ensemble des points employés pour décrire ce mode global et  $ML$  l'ensemble des modes locaux du mode global. On considère toutes les structures de données représentant ce mode global qui se déduisent de celle-ci en un temps  $O(|P| + |ML|)$ . Des algorithmes de marquage permettent de vérifier en  $O(|P| + |ML|)$  les propriétés de disjonction, de chaînage avant et arrière. Si on considère un graphe à arcs multiples, la construction du graphe des modes locaux se fait aussi en  $O(|P| + |ML|)$ , sinon le remplacement des arcs multiples en des arcs simples se fait en  $O(A + |ML|)$  où  $A$  est le nombre d'arcs dans ce graphe. Si on a déjà montré la disjonction en sortie, alors  $A \leq |P|$  et donc le graphe final des modes locaux s'obtient en  $O(|P| + |ML|)$ . La vérification que le graphe est acyclique, et lorsqu'il est acyclique, les calculs de DE, DS, DT, et SR se font en  $O(A + |ML|)$  où  $A$  est le nombre d'arcs dans ce graphe après réduction des arcs multiples. C'est-à-dire, si on a déjà montré la disjonction en sortie, ils se font aussi en  $O(|P| + |ML|)$ .

Le graphe des modes locaux permet de réaliser des regroupements de points de façon à traiter globalement tous les points qui sont considérés comme ayant les mêmes propriétés. Si le mode global est disjoint et s'il a les propriétés de propagation avant et arrière, alors les groupements peuvent être réalisés en  $O(|P| + |ML|)$ . Les différentes formes des regroupements sont les suivantes : PEP( $m_l$ ), PSP( $m_l$ ),

$$PSR(m_l) - PTB(m_l), \overline{PSR(m_l) - PTB(m_l)},$$

$$(PTB(m_l) \cap \overline{PEB(m_l')}) - PSR(m_l), \overline{PTB(m_l) - PSR(m_l)} \cap PEB(m_l'),$$

$PTB(m_l) \cap PEB(m_l') \cap PSR(m_l)$ , et  $PER(m_l) \cap PEB(m_l')$  avec  $m_l$  et  $m_l'$  deux modes locaux distincts. En prenant une notation compatible avec celle de l'exemple de mise en œuvre, ces ensembles s'écrivent respectivement :  $E_{m_l}$ ,  $S_{m_l}$ ,  $R_{m_l}$ ,  $R_{m_l}$ ,  $T_{m_l, m_l'}$ ,  $T_{m_l, m_l'}$ ,  $RT_{m_l, m_l'}$ , et  $RT_{m_l, m_l'}$ .  $E_{m_l}$  et  $S_{m_l}$  décrivent pour un mode local

les liens qu'il a par rapport aux entrées et sorties du circuit. Les  $\underline{R}_{ml}$  et  $R_{ml}$  sont les entrées et les sorties des bascules de rebouclage qui ne sont pas de transmission (on parlera aussi de bascules de rebouclage seul, et de points d'entrée et de sortie seulement rebouclants), ces ensembles dépendent à la fois de la structure du mode local (existence de bascules de rebouclage) et du choix fait pour  $PT(ml)$  permettant d'avoir les propriétés de bijectivité des fonctions d'activation. Les quatre autres formes font intervenir un mode successeur (d'où l'emploi de la lettre T pour dire que ces points sont des entrées ou des sorties de bascules de transmission) et sont donc liées à la structure du mode global. La lettre R indique que les points sont des entrées ou sorties de bascules de rebouclage. Si la ou les lettres sont soulignées alors on considère les entrées sinon on considère les sorties.

L'ensemble des points d'entrée et de sortie du mode global à savoir  $\bigcup_{ml \in ML} PS(ml) \cup PE(ml)$  est l'union disjointe (sur tous les indices) des ensembles précédents appelés par la suite ensembles réduits de points :

$$\left\{ \begin{array}{l} \bigcup_{\substack{ml \in ML \\ ml' \in ML \\ ml \neq ml'}} E_{ml} \cup S_{ml} \cup \underline{R}_{ml} \cup R_{ml} \cup \underline{T}_{ml, ml'} \cup T_{ml, ml'} \cup \underline{RT}_{ml, ml'} \cup RT_{ml, ml'} \end{array} \right.$$

Pour chaque ensemble réduit de points, en fonction de sa forme et des valeurs de DE, DT et SR de son ou ses indices et de son appartenance ou non à  $\text{supp}(T)$  (dans le cas du contrôle continu), on détermine quelles formules élémentaires des algorithmes de contrôle et d'observation s'appliquent et à quels instants. Il suffit donc de parcourir tous les modes locaux une fois et parcourir ainsi toutes les indexations de formes non vides, pour déterminer tous les ensembles de points auxquels s'appliquent les formules élémentaires. La génération des listes des ensembles réduits de points non vides utilisés pour les différents instants et formules élémentaires des algorithmes de mise en œuvre du contrôle et de l'observation du mode global, est donc de complexité au plus  $O(|ML| \times t)$  où t est le nombre d'instants considérés pour l'algorithme. Par souci de gain de place et d'efficacité, on peut ne pas les générer physiquement, et utiliser à la place un processus qui gère la recherche des ensembles réduits de points pour chacune des lignes des algorithmes. Pour accéder de manière beaucoup plus rapide aux points concernés dans les opérations élémentaires utilisées pour le contrôle ou l'observation, il est intéressant de générer des structures ordonnées des ensembles réduits en fonction de  $DE(ml)$ ,  $DT(ml)$  et  $SR(ml)$ . En effet à chaque ligne des algorithmes, on peut considérer que les ensembles réduits de points concernés, sont ceux qui se trouvent dans

une union d'intervalles associés à des relations d'ordre partiel données par DE, DT et SR. Les bornes de ces intervalles vont alors varier en fonction de l'instant et de la phase considérés.

Si on veut diminuer le nombre d'opérations symboliques pour la mise en œuvre en utilisant une classe de fonctions simples, non seulement il faudra une représentation générique de ces fonctions, de leurs fonctions inverses et de leurs composées, mais aussi des algorithmes de calcul pour la composition et l'inversion. Enfin, il faudra adapter l'ensemble des processus élémentaires qui gèrent la recherche des ensembles réduits de points concernés par les opérations élémentaires. En effet, certains ensembles réduits de points ne seront plus traités au niveau des opérations élémentaires mais au niveau d'opérations plus génériques associant la classe de fonctions précédente et des opérations de changement d'indices temporels qui généralisent les opérations de BAV et BAR.

### **V.3 Traitement des bascules de rebouclage seul et limitations potentielles**

La détection de certaines fautes peut nécessiter soit de fixer certaines sorties de bascules qui ne sont pas a priori complètement contrôlables (par exemple les points d'entrée seulement rebouclants), soit d'observer certaines entrées de bascules qui ne sont pas a priori complètement observables (par exemple les points de sortie seulement rebouclants). On s'intéresse donc au contrôle simultané d'un ensemble de points de sortie qui ne sont pas tous des points de transmission, à l'observation d'un ensemble de points d'entrée qui ne sont pas tous des sorties de bascules de transmission, et aux phénomènes de masquage qui y sont associés.

#### **V.3.1 Contrôle des points seulement rebouclants**

Soient un mode global vérifiant les hypothèses du théorème de contrôlabilité continue, et  $T$  un ensemble de points de sortie seulement rebouclants. On suppose d'une part que les sorties des bascules correspondantes sont utilisées en association éventuelle avec des entrées primaires pour appliquer un vecteur de test sensibilisant une faute, et que d'autre part cette faute ne peut pas être sensibilisée en fixant seulement des valeurs sur un ensemble d'entrées primaires et de sorties de bascules de transmission. On suppose que  $\text{supp}(T)$  est un ensemble de modes locaux temporellement indépendants. Alors  $\text{predt}(T)$  est un ensemble de points de transmission temporellement indépendants et il peut être contrôlé de manière continue d'après le théorème de contrôlabilité continue. Puisque le mode global est disjoint en entrée,  $\text{predt}(T) \cup \text{PEP}(\text{supp}(T))$  est contrôlable de manière continue. Donc  $\bigcup_{m_l \in \text{supp}(T)} \text{PE}(m_l) - \text{PER}(m_l)$  est contrôlable de

manière continue. Si on considère chaque mode local de  $\text{supp}(T)$  avec ses bascules de rebouclage comme une machine d'états finis, l'objectif est double. Il est de justifier un état où on a les valeurs recherchées sur les points de sortie qui sont dans  $T$ . Et il faut de plus que les instants de justification sur les différents modes locaux de  $\text{supp}(T)$  puissent correspondre.

La justification d'un état pour une machine d'états finis en partant d'un état connu est un problème général qui se pose lors du test de machines d'états finis. Cela est souvent fait en énumérant les états par des algorithmes faisant des recherches en largeur et/ou en profondeur ou à partir de graphes partiels de transitions d'états [MA87]. Cela est d'autant plus complexe que le graphe est de grande taille. Soit par exemple une machine d'états finis sur  $n$  bits, la plus longue transition minimale qui permet de passer d'un état à un autre est de longueur majorée par  $2^n - 1$  (le nombre d'états moins un), et cette valeur est atteinte par exemple pour un compteur sur  $n$  bits. C'est pour cette raison qu'un nombre maximal d'itérations est en général donné pour limiter de manière raisonnable les temps de calcul pour la recherche de la séquence de justification. Certaines séquences de justification peuvent aussi ne pas exister, en voici un exemple qui

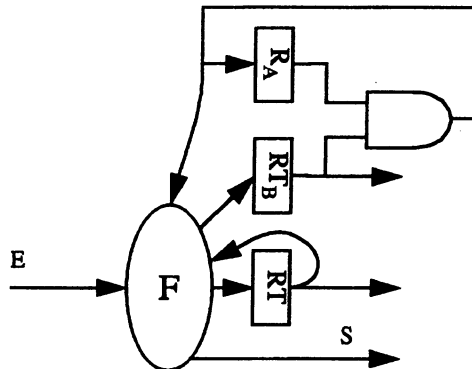


Figure 10 : exemple de problème de justification d'état

montre que partant de l'état de remise à zéro, certaines fautes de type collage ne pourront pas être excitées (et donc détectées) en utilisant seulement l'activation du mode local. Dans cet exemple si on suppose que la sortie de  $R_A$  vaut 0 à l'état de remise à zéro, alors un collage à zéro en entrée ou en sortie de  $R_A$  ne peut être excité à partir de l'état de remise à zéro par des séquences de vecteurs de test appliquées sur les entrées  $E$  tout en activant le mode local. En effet, tout état où l'entrée ou la sortie de  $R_A$  vaut 1 n'est pas atteignable : en fonctionnant en mode activé si la sortie de  $R_A$  vaut 0 à l'instant initial alors elle reste à 0 d'après l'expression de l'entrée de  $R_A$  en fonction de sa sortie et de celle de  $RT_B$ . Dans ce cas, on est obligé de ne plus activer le mode local pour chercher à justifier l'état ciblé. Il se peut aussi que le mode local possède différen-

tes activations équivalentes en terme de propagation (même points de transmission et même points d'entrée non rebouclants) qui fournissent d'autres expressions pour les fonctions d'activation tout en conservant la bijectivité du mode local, alors ces activations sont interchangeables et cette nouvelle activation peut ne plus avoir de problème de contrôle. La justification lorsqu'elle est possible, elle ne l'est pas forcément pour tous les états initiaux. En général, l'état initial à considérer est celui obtenu par les étapes d'initialisation partant de l'état de remise à zéro. Cet état peut donc dépendre des fixations arbitraires que l'on fait sur les entrées primaires.

Certaines machines d'états finis ayant des représentations de haut niveau de type mathématique simple permettent de justifier facilement un état quelconque en général en un nombre limité de coups d'horloge. Soit par exemple un mode local activé rebouclant qui fonctionne comme un registre d'accumulation sur  $n$  bits d'un nombre en

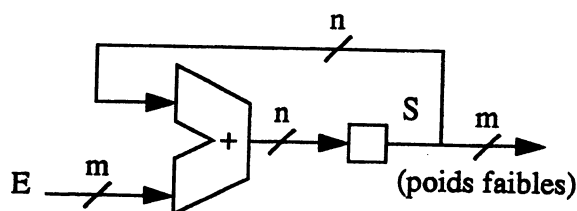


Figure 11 : exemple de contrôle pour un registre d'accumulation

entrée sur  $m$  bits avec  $m < n$ , les bascules de transmission sont alors les  $m$  bascules des poids faibles. Les  $m$  bits de poids faible des bascules de rebouclage dépendent bijectivement des  $m$  bits en entrée lorsque l'état du registre est fixé. En notant les instants à l'aide des indices supérieurs, on a de manière triviale,  $S^k = S^0 + \sum_{0 \leq i < k} E^i$  (modulo  $2^n$ ). Par conséquent tout état cible peut être atteint en au plus  $\lceil (2^n - 1) / (2^m - 1) \rceil$  coups d'horloge ( $\lceil x \rceil$  est le plafond de  $x$ , c'est le plus petit entier supérieur ou égal à  $x$ ). Dans le pire des cas, ce nombre de coups d'horloge est nécessaire, c'est le temps qu'il faut pour passer de la valeur 0 en  $S$  à l'instant 0, à la valeur  $2^n - 1$ . Puisque l'application de 0 en entrée ne modifie pas l'état de sortie, dès qu'on a atteint la valeur cible, on peut la conserver en sortie aussi longtemps qu'on le souhaite. L'état cible est atteint de manière périodique à partir d'un certain instant, avec dans le cas présent une période de 1. On démontre facilement que si pour chaque machine d'états finis associée à un mode local  $m_l$  de  $\text{supp}(T)$ , l'état à justifier peut l'être à partir d'un certain instant de manière périodique de période  $T_{m_l}$  pour l'état initial donné par l'étape d'initialisation. Alors si pour tout paire  $(m_l, m_l')$  de modes locaux distincts de  $\text{supp}(T)$ ,  $T_{m_l}$  et

$T_{m_l}$  sont premiers entre eux alors tous les états ciblés peuvent être justifiés simultanément. Si on note  $J_{m_l}$  le premier instant de justification périodique, alors  $J_{m_l} \leq 2^{BR(m_l)}$  et l'instant de justification pour  $T$  est majoré par :

$$L(T) + \text{ppcm}\left(\bigcup_{m_l \in \text{supp}(T)} T_{m_l}\right) + \max\left(\bigcup_{m_l \in \text{supp}(T)} T_{m_l}\right).$$

Dans le cas où de telles relations n'existent pas entre les périodes de justification, il faut vérifier les compatibilités ou non entre les instants de justifications périodiques possibles pour les modes locaux dont les périodes ne sont pas premières deux à deux. S'il n'y a pas de compatibilité, alors il faudra utiliser des instants non périodiques, (c'est aussi le cas où des justifications périodiques n'existent pas) et faire des recherches de coïncidences d'instant de justification.

### **V.3.2 Observation des points seulement rebouclants**

Lorsqu'une faute a été sensibilisée et propagée sur une bascule de sortie du mode global, il faut alors propager cette faute sur les sorties primaires du mode global. On discute d'abord le cas où les points de sortie sur lesquels la faute est propagée appartiennent à l'ensemble des points de sortie d'un même mode local  $m_l$ . Dans ce cas, l'objectif est de propager la faute sur les points de transmission de  $m_l$  en activant le mode local, afin de pouvoir ensuite l'observer sur les sorties primaires du mode global en l'activant. Une fois que la faute est propagée sur un point de transmission, le nombre de coups d'horloge maximal nécessaires pour détecter la faute est donné par la distance aux sorties de  $m_l$ . En effet, pour un mode local  $m_l$ ,  $1 + DS(m_l)$  représente le nombre maximum d'observations nécessaires pour que les valeurs de points d'entrée du mode local à un instant donné  $t$  soient déterminées de manière unique en fonction des valeurs observées aux sorties primaires de cet instant  $t$  jusqu'à l'instant  $t + DS(m_l)$  inclus. Les sorties primaires concernées sont celles du mode local à l'instant  $t$  et celles de ses modes successeurs d'ordre quelconque.

En supposant que les fonctions d'activation du mode local sont encore valides malgré la faute, le problème se résume à rechercher une séquence de contamination d'états appliquée en entrée d'une machine d'états finis, c'est-à-dire une séquence qui propage la faute sur certains bits d'états (dits contaminés) ou sur certaines sorties primaires. C'est un problème général qui se pose lors du test de machines d'états finis. Dans le cas présent, les points ciblés pour la contamination sont les points de sortie primaire et les points de transmission (la contamination d'un seul de ces points suffit). Deux causes peuvent entraîner la non contamination des points ciblés, la première est la non exis-

tence de séquence de contamination et la deuxième est la non applicabilité d'une telle séquence à cause de la latence pour contrôler de manière continue le mode local. On reprend l'exemple donné par la figure 10 page 123. Soit une faute qui a contaminé  $R_A$  en le fixant à 0 au lieu de 1 pour un circuit sans faute, et qui n'a pas contaminé les autres points de sortie du mode local (S et les entrées de  $R_B$  et de TR). Si  $R_B$  est à 0, alors la contamination de cette faute aux points de S ou en entrée de TR ne peut être obtenue car au coup d'horloge suivant les sorties de TR et de  $R_B$  ont les valeurs du circuit sans faute et  $R_A$  vaut 0 comme pour le circuit sans faute. Cet exemple illustre donc la première cause, il peut aussi illustrer la deuxième. En effet, on suppose cette fois-ci que  $R_B$  est à 1 et que les autres points de sortie du mode local autres que l'entrée de  $R_A$  (S et les entrées de  $R_B$  et de TR) ne sont pas contaminés. Si les valeurs en E dépendent au moins en partie des séquences de test utilisées pour contaminer  $R_A$ , et qu'elles imposent 0 en entrée de  $R_B$  alors au coup d'horloge suivant les sorties de TR et  $R_B$  ont les valeurs du circuit sans faute, la sortie de  $R_B$  vaut 0 et donc S et les entrées de TR et  $R_B$  ont aussi les valeurs du circuit sans faute. Par conséquent, on se retrouve dans le cas précédent et on ne peut avoir de contamination des points de transmission ou des points de sortie primaire.

Lorsque la faute a contaminé plusieurs points de sorties associés à plusieurs modes locaux, il est nécessaire de contaminer au moins un point de transmission ou un point de sortie primaire pour l'un des modes locaux concernés (dits contaminés), mais cela n'est pas suffisant. En effet, si l'ensemble des modes locaux concernés n'est pas temporellement indépendant, il peut y avoir des phénomènes entre un mode local contaminé et l'un de ses successeurs d'ordre quelconque aussi contaminé. Cela signifie que la propagation de valeurs de points de transmission contaminés d'un mode local  $m_1$  peut être stoppée au passage par un mode local  $m_1'$  (un successeur d'ordre quelconque de  $m_1$ ) dont certaines entrées rebouclantes sont contaminées. Cela arrive lorsque leurs combinaisons se compensent pour donner les valeurs du circuit sans faute sur les points de transmission et les sorties primaires de  $m_1'$ . Donc dans les cas où l'ensemble des modes locaux contaminés n'est pas temporellement indépendant, il vaut mieux commencer par essayer de contaminer les points de transmission des modes locaux qui ne possèdent pas de prédécesseurs, ni de successeurs d'ordre quelconque non contaminés. En effet, l'ensemble de ces modes locaux concernés est alors temporellement indépendant. Ils sont contrôlables de manière continue et les modes locaux utilisés pour leur observation ne sont pas contaminés. Si on échoue de cette manière, les modes locaux contaminés restants sont temporellement dépendants. Parmi ceux pour lesquels on a

déterminé des séquences de contamination des points de transmission, on extrait un ensemble maximal de modes locaux temporellement indépendants, on les contrôle de manière continue pour contaminer les points de transmission. Si les séquences obtenues ne permettent pas de propager la faute aux sorties primaires (à cause de compensations lors de la propagation), il faut alors soit changer d'ensemble maximal, soit tenir compte des délais de propagation entre un mode local contaminé et un autre mode local contaminé. Ceci afin de tirer parti du fait qu'on a souvent plusieurs séquences qui contaminent des points de transmission différents ou qui contaminent à des instants différents.

Dans l'exemple du registre d'accumulation de la figure 11 page 124 on se trouve dans un cas extrême où si les points de sortie seulement rebouclants sont contaminés par une faute alors l'ensemble de tous les points de sortie seulement rebouclants restera contaminé tant que le mode reste activé. Par contre cette contamination ne se propagera pas à des points de transmission. En effet, la formulation mathématique donnée montre que la valeur en un point de sortie ne dépend pas des valeurs des points d'entrée associés à des poids plus forts, or ici les points de sortie seulement rebouclants ont des poids plus forts que les points de transmission. Pour remédier à ce problème, il suffirait de rendre observable le point de poids le plus fort. On verra au VI.2.2 page 135 comment on peut le faire avec un impact minimum sur le chemin critique et en utilisant peu de logique combinatoire. La solution proposée a aussi l'avantage que lorsque la sortie de la bascule de rebouclage de poids fort n'est pas encore contaminée, on peut retarder les instants de contamination des points de transmission en fixant l'entrée à 0. En fonction des entrées mises après contamination de la sortie de la bascule de poids faible, on pourra aussi faire différents choix pour l'ensemble de points de transmission que l'on veut contaminer.

### **V.3.3 Limitations dues à l'utilisation d'un modèle fonctionnel**

Un simulateur de fautes doit être utilisé pour vérifier la détection d'une faute ciblée car les modèles fonctionnels considérés lorsque les modes locaux sont activés pourraient être faux en présence de cette faute. Dans ce cas, on peut continuer à utiliser les mêmes principes de justification et de contamination d'états, mais en simulant le résultat obtenu pour un modèle de circuit avec et sans cette faute. On se trouve face à deux cas : soit la séquence trouvée a quand même permis d'avoir la justification ou la contamination voulue, soit durant la séquence de justification ou de contamination, la faute a été propagée sur des points de transmission (ou des sorties primaires) et il suffit alors de réaliser l'observation des points de transmission. Sinon il faut rechercher d'autres séquences ou bien abandonner.



Les deux critères suivants peuvent être pris pour aider la génération des vecteurs de test associée à chaque faute. Il vaut mieux chercher d'abord à prendre des vecteurs de test locaux qui ont le minimum de points de justification parmi les points qui ne sont ni des sorties de bascules de transmission, ni des entrées primaires. De même, il vaut mieux chercher à prendre des vecteurs de test locaux qui ont le minimum de points de sensibilisation parmi les points qui ne sont ni des sorties de bascules de transmission, ni des entrées primaires.

## **VI. Méthodes d'obtention et de sélection de modes globaux adaptés**

### **VI.1 Introduction**

Dans la méthode proposée, l'objectif est d'extraire des modes globaux (avec les propriétés requises) à partir d'un circuit donné et/ou d'effectuer des modifications sur le circuit (en satisfaisant les contraintes du concepteur) permettant d'obtenir de tels modes globaux. Les contraintes du concepteur sont prises dans l'ordre de priorité suivant : circuit fonctionnant à la vitesse requise par les spécifications, vecteurs de test du circuit donnant un taux de couverture de fautes maximal (pour une durée d'application qui doit rester réalisable et exploitable), puis sur un même plan les trois dernières contraintes (ou critères) de surface, de simplicité de mise en œuvre et de longueur des séquences de test. Ces trois derniers critères ont des importances relatives différentes selon les spécificités du circuit.

Au VI.2, différentes méthodes d'obtention de modes locaux seront exhibées, et on analysera vis à vis des contraintes du concepteur les modes ainsi obtenus. Finalement, au VI.3, on montrera comment par rapport à ces contraintes on construit une ou plusieurs solutions pour le test grâce à des modes globaux vérifiant les hypothèses des théorèmes d'accessibilité. On décrira aussi l'utilisation des modes globaux de manière hiérarchique. Dans les cas où on n'arrive pas à trouver de telles solutions par l'emploi brut des modes globaux, des principes sont décrits pour faire quelques extensions et pour utiliser conjointement les méthodes de chemins équilibrés ou les méthodes de test symbolique en exploitant au mieux les diverses propriétés d'accessibilité liées aux modes globaux.

### **VI.2 Obtention des modes locaux**

#### **VI.2.1 Utilisation de blocs fonctionnels du circuit**

Soit  $S_n$  l'ensemble des systèmes de  $n$  fonctions booléennes à  $n$  variables indépendantes.  $S_n$  est de cardinal  $(2^n)^{2^n}$ , c'est-à-dire le nombre d'applications d'un ensemble à  $2^n$  éléments vers un ensemble à  $2^n$  éléments. Parmi celles-ci,  $(2^n)!$  sont bijectives ce qui représente en utilisant la formule de Stirling un rapport de l'ordre de  $\sqrt{2 \times \pi \times 2^n} \times e^{-2^n}$ . Pour  $n = 8$  le rapport vaut  $26,5 \cdot 10^{-111}$ , c'est à la fois une proportion très faible et cependant un nombre énorme de systèmes possibles :  $857 \cdot 10^{504}$  bijections. En fait, dans les circuits réels, on utilise souvent des opérateurs mathématiques qui ont des propriétés de bijectivité ou qui ont des propriétés "très proches". Les

opérateurs tels que l'additionneur, le soustracteur, le multiplicateur, le diviseur, le multiplexeur, l'opérateur de négation, ou les opérateurs de permutations fournissent des exemples ayant ces propriétés et couramment employés dans les chemins de données. Dans l'obtention d'un mode global satisfaisant aux contraintes du circuit, l'une des tâches consistera à répertorier les modes locaux utilisables et utiles pour satisfaire ces contraintes ou pour minimiser les coûts du circuit. Dans ce paragraphe, on décrit comment on peut exploiter la fonctionnalité de ces opérateurs, et au VI.2.2 page 135 on discute des améliorations possibles pour augmenter l'observabilité et la contrôlabilité tout en tenant compte des contraintes en vitesse et de la diminution du coût en surface. On montre aussi comment on peut utiliser ces opérateurs dans le cas de machines d'états finis que l'on considère comme des séquenceurs.

Dans un premier temps, on considère les opérateurs isolés, c'est-à-dire connectés seulement à des registres ou à des entrées et sorties primaires.

### 1 - Opérateurs unaires

Les opérateurs unaires comme les opérateurs de négation et de permutation sont parmi les plus faciles à traiter, car on peut les considérer directement comme des modes locaux. La figure 12 page 130 montre l'exemple très simple du complément à 1. Lors-

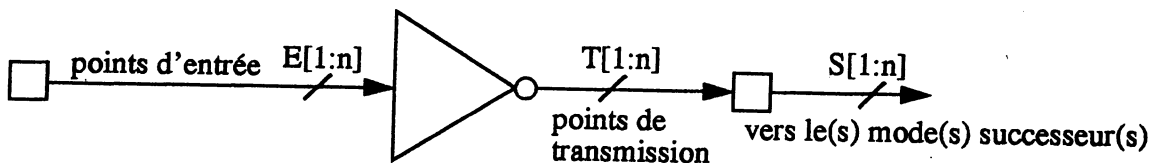


Figure 12 : complément à 1

qu'on sélectionne des modes locaux pour former un mode global, on augmente les possibilités de sélection et de chaînage des modes locaux (voir VI.3.1 page 139 et VI.3.4 page 143), si on considère des modes locaux éclatés en sous-modes locaux bijectifs. Ces sous-modes locaux bijectifs sont des modes locaux bijectifs obtenus en considérant des sous-ensembles des ensembles des points de transmission, d'entrée, de sortie et des points d'activation (il n'est pas nécessaire d'avoir affaire à des opérateurs unaires). Il est évident que ces opérateurs peuvent être activés en même temps, c'est pourquoi pour l'obtention des modes globaux, on considérera qu'ils sont compatibles entre eux pour la disjonction même s'ils ne sont pas forcément disjoints en sortie (ils pourraient par exemple avoir en commun des points de sortie rebouclants qui ne sont pas de transmission).

### 2 - Multiplexeur

On s'intéresse maintenant aux situations plus riches des opérateurs binaires ou n-aires.

Pour réutiliser la fonctionnalité d'un multiplexeur à l'intérieur d'un mode local bijectif, les deux mots d'entrée peuvent servir a priori d'ensemble de points d'entrée. Ce mode local bijectif va par la suite être utilisé pour transmettre des données à l'intérieur d'un mode global. Donc, dans le cas où le multiplexeur reboucle sur lui-même au travers d'un registre, on n'a pas le choix sur le mot d'entrée à lui associer (il faut au moins un point d'entrée non rebouclant) (voir figure 13). Dans le cas où il n'y a pas de rebouclage

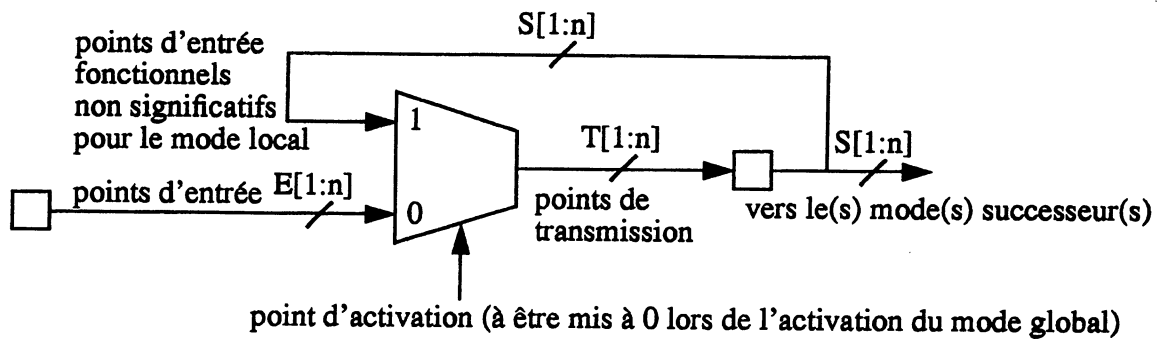


Figure 13 : multiplexeur avec rebouclage

par un registre, il y a deux possibilités d'utilisation du multiplexeur, la sélection de l'une ou l'autre va se faire lorsqu'il faudra créer le mode global final. Dans tous les cas, il y a un seul point d'activation qui doit être contrôlé à 0 (ou à 1 suivant le mot d'entrée choisi) durant l'activation du mode global. Si on suppose qu'on a un signal qui vaut 1 pour signifier l'activation du mode global, alors il suffit de combiner ce signal avec le signal d'activation par l'intermédiaire d'une porte "non ou" (ou "ou"). Cela n'est pas nécessaire si le point d'activation est une entrée primaire. On remarquera que dans le cas du multiplexeur avec rebouclage, ce signal d'activation peut aussi facilement être utilisé pour réaliser des modes de maintien, qui peuvent servir en entrée de chemins équilibrés (BALLAST).

### 3 - Additionneur ou soustracteur

Pour un additionneur (ou un soustracteur) simple, à savoir sans rebouclage<sup>12</sup> par un registre (voir figure 14), comme pour le multiplexeur, on peut choisir quel mot utiliser

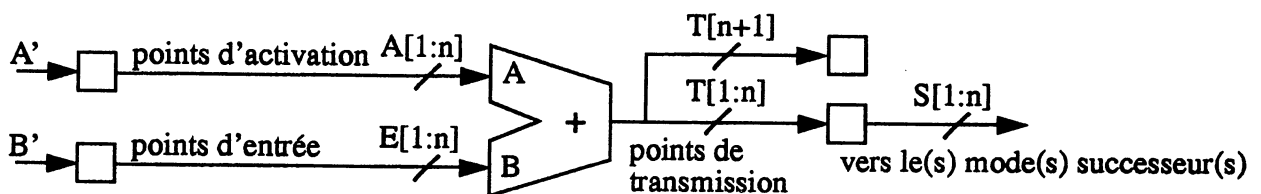


Figure 14 : additionneur simple

<sup>12</sup> Le cas d'un additionneur (ou soustracteur) avec un rebouclage a déjà été donné en exemple sous le nom d'accumulateur (voir la figure 11 page 124).

comme ensemble de points d'entrée (A ou B). Mais de plus toutes les valeurs du mot restant (A dans le cas de la figure) fournissent des modes locaux bijectifs distincts. Le mot A sert donc de paramètre aux modes locaux, et toutes les valeurs de A sont admissibles. On parlera pour A de points d'activation à  $\varphi$  (par analogie à la notation employée dans les méthodes de synthèse). A un point d'activation, on associe sa ou ses bascules d'activation, qui sont les bascules dont les sorties déterminent la valeur du point d'activation. L'ensemble des bascules d'activation pour un mode local est noté BA. On suppose que  $BA \cap BE = \emptyset$  même si des extensions peuvent être faites quand l'intersection est non vide. Dans cet exemple l'ensemble des entrées des bascules de BA est l'ensemble A'. Dans la méthode de test proposée, pour un point d'activation à  $\varphi$  donné, deux possibilités sont offertes. La première est de forcer ce point à une valeur figée lors de l'activation du mode global (comme dans le cas du point d'activation du multiplexeur). La deuxième consiste à faire en sorte que l'ensemble des entrées des bascules d'activation de ce point et l'ensemble des entrées des bascules d'entrée soient deux ensembles de points de transmission temporellement indépendants, c'est-à-dire où tout point du premier ensemble est temporellement indépendant de tout point du deuxième ensemble. Quand l'union de ces points est un ensemble de points temporellement indépendants, on est dans un premier cas particulier simple. Quand les bascules d'activation et les bascules d'entrée appartiennent à deux chaînes de scan distinctes, on est dans un deuxième cas particulier simple. L'indépendance temporelle est essentielle car elle permet de conserver les théorèmes sur l'accessibilité. Cette double utilisation des points de transmission allonge la durée nécessaire pour l'observation. La durée nécessaire pour déterminer complètement la valeur des points de transmission d'un mode local  $m_l$  n'est plus donnée par DS, mais en est déduite de la façon suivante :

**Etape 1 :** Poser  $E = \{m_l\}$

**Etape 2 :** Ajouter à E tous les successeurs d'ordre quelconque des éléments de E

**Etape 3 :** Ajouter à E tous les modes locaux associés par les points d'activation aux éléments de E.

**Etape 4 :** Si E a plus d'éléments qu'avant la dernière étape 2, alors aller à l'étape 2, sinon aller à l'étape 5.

**Etape 5 :** La durée nécessaire est donnée par le maximum de DS sur E.

Cette double utilisation des points de transmission a aussi quelques effets sur la mise en œuvre du contrôle. Avant d'inverser les équations du mode local pour un instant donné, il faut calculer d'abord les valeurs des points d'activation à cet instant. Cela est

toujours possible à cause de l'indépendance temporelle.

#### 4 - Multiplicateur

On considère maintenant l'opérateur de multiplication sur des nombres positifs non signés de taille  $n$  et  $m$  bits. Le résultat est sur  $n + m - 1$  bits. On fixe l'une des opérands comme paramètre à une valeur non nulle  $k$ , alors l'opérateur noté  $M_k$  agit de manière injective. On suppose que  $k$  est une opérande sur  $m$  bits, le résultat varie alors de  $0$  à  $k \times (2^n - 1)$ . On décompose  $k$  en base 2 :  $k = k_0 \cdot 2^0 + \dots + k_{m-1} \cdot 2^{m-1}$ . Soit  $p$  le plus petit indice  $i$  pour lequel  $k_i$  est non nul. Soit  $g$  le plus grand indice  $i$  pour lequel  $k_i$  est non nul. Soit  $c_0 \cdot 2^0 + \dots + c_{m-1} \cdot 2^{m-1}$  la décomposition en base 2 d'une valeur obtenue par  $M_k$ . Si on considère la valeur  $k \times 2^i$  où  $0 \leq i < n$  alors  $c_{p+i} = 1$  et  $c_{g+i} = 1$ . Pour la valeur  $k \times 0$ , on a :  $c_{p+i} = 0$  et  $c_{g+i} = 0$ . Il faut que l'ensemble des valeurs obtenues par  $M_k$  utilise au plus  $n$  bits significatifs pour pouvoir extraire  $n$  points de transmission pour lesquels il y a bijectivité. Donc il faut que  $p = g$ , c'est-à-dire il faut que  $k$  soit une puissance de 2. Réciproquement, la figure 15 montre le mode local bijectif qui est associé au multiplicateur quand  $k$  est la  $i$ ème

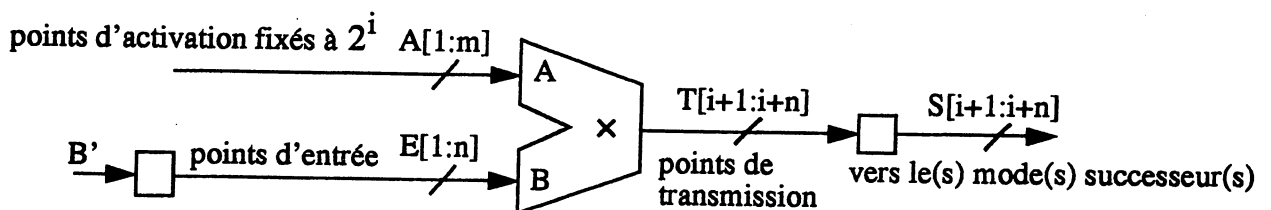


Figure 15 : multiplicateur simple

puissance de 2. Par rapport à l'addition, les points d'activation doivent forcément être contrôlés à l'une des  $m$  valeurs possibles pour  $k$ .

#### 5 - Opérateurs galoisiens

On considère les opérateurs qui se décomposent en des structures à base de "ou exclusifs". Ces opérateurs peuvent être qualifiés de galoisiens en ce sens qu'ils sont modélisables de manière biunivoque par des applications affines sur des espaces affines de dimension finie associés au corps de Galois<sup>13</sup>  $(\mathbb{Z}/2\mathbb{Z}, \oplus, \cdot)$  noté aussi  $GL(2)$ . L'opérateur  $\oplus$  est l'opération "ou exclusif", et l'opérateur " $\cdot$ " est l'opération "et". On pourra utiliser des représentations matricielles et appliquer des théories d'algèbre

<sup>13</sup>. Plus généralement, un corps de Galois est un corps ayant un nombre fini d'éléments.

linéaire pour traiter ces opérateurs et en particulier pour rechercher la bijectivité et pour les inverser. Les principes sont ceux de la diagonalisation de matrices et du pivot de Gauss [FAUR89] qui sont simplifiés pour l'application au corps de Galois. On trouvera dans l'annexe K, page 231 : "Inversion des opérateurs galoisiens", les méthodes pour la recherche et l'inversion de ces structures bijectives. Le système suivant des formules booléennes d'une partie combinatoire d'un circuit donne un exemple d'opérateur galoisien.

$$\begin{cases} f = a \oplus b \\ g = a \oplus c \oplus 1 \\ h = b \oplus c \oplus d \end{cases} \text{ s'écrit aussi matriciellement : } \begin{bmatrix} f \\ g \\ h \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} \oplus \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}.$$

Toutes les matrices 3x3 extraites de  $\begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix}$  sont inversibles. Donc pour chaque varia-

ble a, b, c, ou d prise comme paramètre, on peut résoudre ce système à trois équations et trois inconnues. Par exemple en prenant b comme paramètre, la résolution donne :

$$\begin{bmatrix} a \\ c \\ d \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} b \\ f \\ g \\ h \end{bmatrix} \oplus \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}.$$

Ces opérateurs galoisiens sont fréquemment utilisés pour les opérations du type codage, décodage, brouillage ou de génération de séquences pseudo-aléatoires, et on les retrouve souvent dans les circuits de télécommunication comme on le verra sur l'exemple d'application du chapitre 3 page 149 intitulé "Application à des circuits de terminaison de ligne pour l'ATM". Ils sont aussi fréquemment utilisés dans les méthodes de test aléatoire ou fondées sur des analyses de signature. Les utilisations les plus répandues concernent des sous-classes de ces opérateurs qui sont modélisés par des polynômes à coefficients dans  $GL(2)$ . La littérature est très abondante sur ce sujet. Dans [LEVE90], les différentes structures de compaction par division polynomiale sont analysées et classées.

## 6 - Composition d'opérateurs

Dans un deuxième temps, on peut aussi considérer la composition d'un nombre quelconque de ces opérateurs. Cela peut se faire par des interconnexions entre ces opérateurs ou aussi en combinant deux opérateurs en un seul. Par exemple une unité arithmétique et logique possède un ou plusieurs signaux de contrôle pour effectuer des

opérations distinctes. Si pour différentes configurations les comportements sont bijectifs, on pourra déterminer les signaux d'activation à  $\varphi$  pour diminuer le coût potentiel du mode local. Il faut alors étudier les propriétés de bijectivité qui sont déduites par composition ; les principes d'exploitation de ces propriétés restent les mêmes. Le concepteur peut aussi identifier d'autres fonctions mathématiques implantées qui ont des propriétés similaires. Pour aider le concepteur à trouver des parties combinatoires du circuit qui ont des propriétés de bijectivité, différents types d'outils peuvent être imaginés. Le premier ferait des recherches par des calculs booléens formels ou non. Cela serait certainement faisable dans le cas de parties combinatoires très petites étant donné la complexité des calculs [MUEH82] ; on pourrait aussi intégrer cet outil à un outil de synthèse. Le deuxième outil ferait des recherches de solutions potentielles à valider. Il partirait de calculs de probabilités conditionnelles d'avoir en sortie de blocs combinatoires des valeurs à 1 (ou à 0 ce qui est équivalent). On trouvera la description de ces techniques de calcul de probabilité basées sur des représentations de fonctions booléennes à l'aide de diagrammes de décision binaire dans [CHEN89]. On trouvera des techniques d'évaluations empiriques de ces probabilités dans [THEA89]. Elles correspondent à des résultats statistiques de simulations de vecteurs aléatoires.

### **VI.2.2 Constructions de modes locaux optimaux**

La façon la plus simple de construire un mode local est celle classique du scan. On considère une entrée de bascule (ou une sortie primaire) où on insère un multiplexeur. Le choix de l'entrée primaire ou de la sortie de bascule à connecter sur l'entrée restante du multiplexeur sera explicité au VI.3.4 page 143. Le coût sur le temps de propagation est celui d'un multiplexeur. Quand un multiplexeur déjà existant est utilisé, le coût en surface et en temps de propagation est alors plus faible car une porte logique simple du type "non-et" suffit sur le signal de contrôle du multiplexeur.

#### **1 - Registre d'accumulation**

Dans l'exemple du registre d'accumulation (le raisonnement est le même pour l'additionneur simple), on s'est demandé comment augmenter l'observabilité du point de sortie de poids le plus fort. Dans ce genre de logique, les chemins critiques se trouvent sur les poids les plus forts, on propose donc de modifier l'équation du point de transmission de poids le plus faible tout en conservant la bijectivité du mode local. On place (voir la figure 16) un "ou exclusif" en entrée de la bascule de transmission de poids le plus faible. Les deux entrées du "ou exclusif" deviennent respectivement l'ancienne entrée de la bascule et la sortie de la bascule de poids le plus fort. Pour que cette fonctionnalité ne soit valide que lorsque le mode local est activé, il suffit d'ajouter de la logique for-



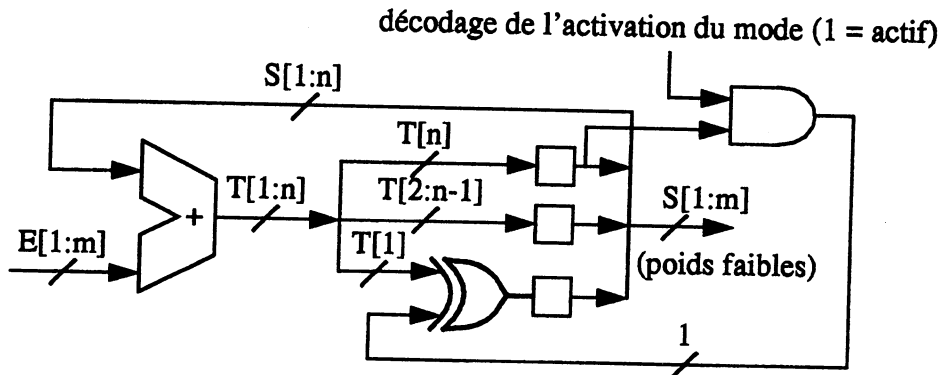


Figure 16 : registre d'accumulation rendu accessible

çant à 0 l'entrée du "ou exclusif" où arrive le bit de poids fort. Le coût en surface est donné par l'ajout d'une porte "ou exclusif" et d'une porte "et". Ce coût est à comparer à l'ajout de  $n$  multiplexeurs dans le cas d'une méthode de scan classique. Le coût final en vitesse a de fortes chances d'être nul car on n'a pas touché aux chemins les plus critiques de ce bloc fonctionnel. Il est aussi à remarquer que la durée maximale de la justification d'un état quelconque partant d'un état connu, vaut  $2^{n-m}$ . C'est aussi le nombre maximum d'observations consécutives nécessaires pour vérifier la valeur initiale de l'état.

## 2 - Multiplicateur

Dans le cas du multiplicateur présenté au VI.2.2 page 135, pour diminuer le coût en chemin critique, on choisit la plus grande valeur possible pour  $k$ , à savoir  $2^m - 1$ . Alors les points de transmission contiennent les  $n$  points de poids fort et a priori aussi les extrémités des chemins les plus critiques. Relativement aux points précédents, les  $m - 1$  points de poids faible peuvent être mis en scan, l'ajout d'un multiplexeur ne va pas limiter la vitesse de fonctionnement du circuit. En contrepartie, forcer la valeur  $k$  pour l'activation a un coût en temps de propagation de l'ordre d'une porte logique du type "et" ou "ou". Ce coût est inférieur à celui obtenu quand on utilise un multiplexeur. Si ce coût est trop important, on verra quels moyens on peut employer à un niveau global du circuit pour supprimer ce coût restant. Le coût en surface est d'environ  $m - 1$  multiplexeurs plus  $m - 1$  portes logiques du type "et", il est à comparer à  $m + n - 1$  multiplexeurs dans le cas du scan. Le gain en connexions ajoutées est de  $3 \cdot n - 2 \cdot (m - 1)$ . Dans tous les cas où ces genres d'opérateurs sont utilisés et principalement dans les cas où les entrées des bascules ou les sorties primaires appartiennent a priori à des chemins critiques, il vaut mieux décrire ces modes locaux activés à un haut niveau (VHDL RTL par exemple) afin que l'outil de synthèse puisse optimiser tout de suite en vitesse ou en surface en tenant compte de ce qui devra être ajouté pour

le test.

### 3 - Séquenceur

Un séquenceur peut être considéré comme une machine d'états finis dérivée d'un compteur. C'est-à-dire selon les conditions aux entrées et la valeur du compteur, on l'incrémente, l'arrête ou parfois on fait des sauts à certaines valeurs. En fait, cette description s'applique à une machine d'états finis quelconque, elle correspond à un codage particulier. Ce codage est bien adapté quand le graphe de transition des états comporte peu de branches et est de grande taille. L'objectif recherché est de transformer cette machine en une autre machine d'états finis qui possède un mode fonctionnement activable en tant que mode local. Ce mode local doit aussi être utilisable à l'intérieur d'un mode global pour transmettre les vecteurs de test. En outre, on veut minimiser l'impact de son implantation sur le chemin critique, et rendre tous les états de cette machine facilement contrôlables et observables via ce mode local.

Dans un premier temps, on modifie cette machine de façon à pouvoir activer un mode de fonctionnement correspondant à l'incrémentation du compteur. Cette modification a priori agit peu sur la complexité des équations d'états (ou de transition) de la machine. On suppose donc, qu'après synthèse, l'impact de cette modification a été très faible sur la surface et le chemin critique. Ce mode sera activé par un signal d'activation ajouté en entrée de la machine d'états finis. On appellera ce mode, mode autonome. Dans le cas d'un compteur binaire (comptant dans l'ordre), la figure 17 montre un exemple de

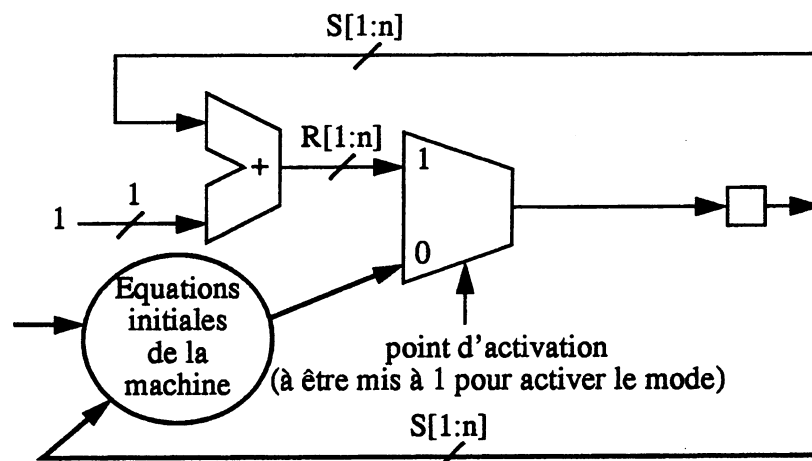
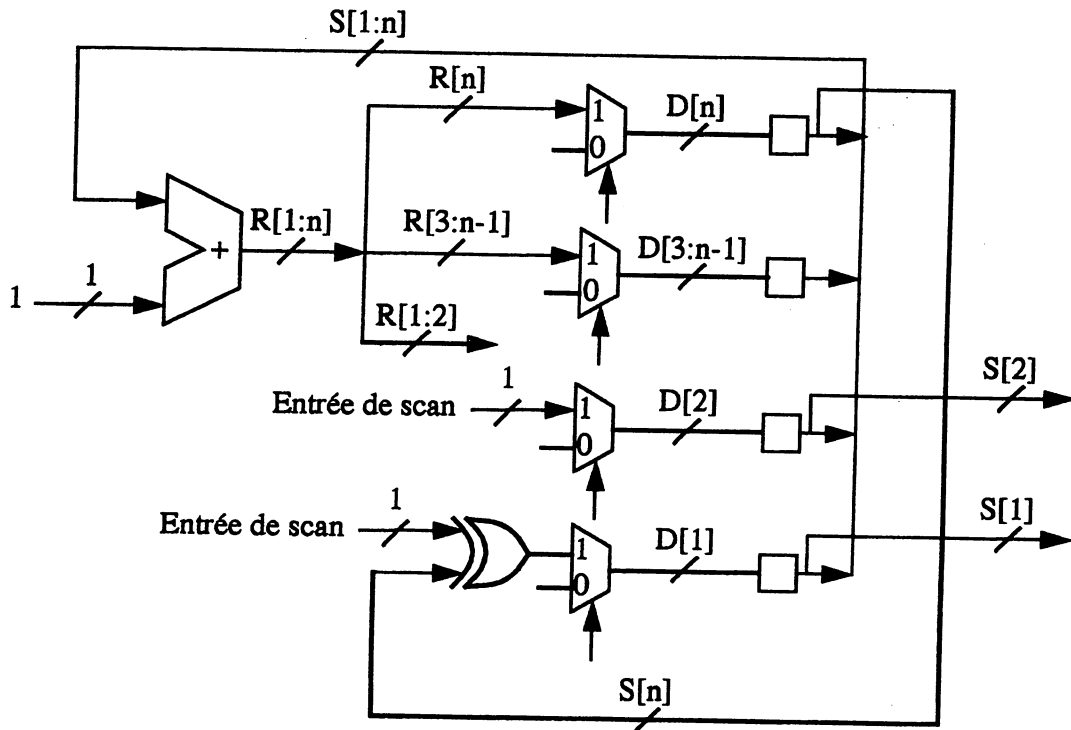


Figure 17 : séquenceur avec compteur binaire en mode autonome

mode autonome. La machine d'états finis correspondante est alors synthétisée. Un outil d'analyse temporelle permet ensuite de déterminer les entrées des bascules du séquenceurs qui sont en extrémité de chemins critiques.

Dans un deuxième temps, on modifie le mode autonome en mettant en scan les bascules

non critiques. Ces bascules deviennent des bascules de transmission, elles passent en mode scan lorsque le mode autonome est activé. S'il reste des bascules du séquenceurs qui ne sont pas mises en scan (car elles sont à l'extrémité de chemins trop critiques), alors on cherche à augmenter la contrôlabilité et l'observabilité de ces bascules. Comme dans le cas du registre d'accumulation (1 du VI.2.2 page 135), partant d'un état connu, en procédant par incrémentations successives, on atteint un état quelconque en un nombre déterminé de coups d'horloge. En revanche, l'observation de la valeur d'une bascule se fait directement par elle-même si elle est de transmission, sinon l'observation se fait par l'intermédiaire des bascules de poids supérieur. Par conséquent, lorsque la bascule de poids le plus fort n'est pas de transmission, un moyen d'observation indirecte de cette bascule (et implicitement celles de poids inférieur) consiste à ajouter un "ou exclusif" en sortie de cette bascule pour reboucler sur une autre bascule de poids plus faible (voir figure 18). On emploiera par exemple, comme bascule de



**Figure 18** : exemple de mode autonome de séquenceur après rebouclage par un "ou exclusif"

rebouclage, la bascule de transmission de poids le plus faible. Le mode local obtenu est rebouclant. En principe, on ne détériore pas le chemin critique car on agit sur la donnée qui venait directement de la sortie de la bascule précédente dans la chaîne de scan. Pour faciliter le contrôle et l'observation de l'état des bascules et rester dans le cadre de la définition des modes locaux rebouclants, il faut que les bascules de transmission des séquenceurs soient au départ mises sur des chaînes de scan distinctes.

Une autre façon de procéder consiste à ne pas ajouter de "ou exclusif", dans ce cas le

mode local créé est non rebouclant, il est de type scan (on peut aussi le décomposer en plusieurs modes locaux). Ces modes locaux ainsi créés facilitent alors la mise en œuvre de l'accessibilité des points de transmission du mode global final. Les valeurs des bascules d'état qui n'ont pas été utilisées pour la transmission, n'interviennent pas comme paramètres pour les fonctions d'activation. Cependant, dans la pratique, les problèmes de test liés à ces bascules non mises en scan, sont semblables à ceux d'un mode local avec des bascules de rebouclage seul. La différence essentielle pour la résolution de ces problèmes réside dans la contamination des bascules de transmission, qui s'effectue non plus au dernier coup d'horloge en étant toujours en mode activé, mais en passant dans le mode de fonctionnement normal. De manière générale, comme dans le cas du registre d'accumulation, les latences pour le contrôle ou l'observation d'un état quelconque vont dépendre exponentiellement de la différence entre le nombre de bascules d'état et le nombre de bascules de transmission. On peut souvent améliorer les performances en termes de surface, de vitesse et de longueur des séquences de test en utilisant un compteur qui ne compte plus dans l'ordre classique du type registre à décalage à rebouclage linéaire, ou LFSR (Linear Feedback Shift Register). En terme de surface, il n'utilise que peu de portes (contrairement à un incrémenteur classique). Ces portes sont des "ou exclusifs", on a donc affaire à des opérateurs galoisiens. De plus, en choisissant une structure à "ou exclusifs" internes (voir [LEVE90]), les équations locales à chaque entrée de bascule du registre, font intervenir une ou deux sorties de registres. Cela permet en général de gagner en terme de chemin critique en évitant des propagations de retenues. La latence pour l'observation et le contrôle des bascules d'état d'un compteur de type LFSR sera beaucoup plus faible que dans le cas précédent, elle vaudra le nombre de bascules d'état si une seule bascule est mise en scan.

## **VI.3 Obtention des modes globaux**

### **VI.3.1 Formulation du problème**

#### **1 - Principes**

La première démarche que l'on peut adopter est de partir d'une description de type RTL et de sélectionner une liste des entrées de bascules et des sorties primaires qui sont potentiellement sur des chemins critiques qui risquent de ne pas satisfaire les contraintes de vitesse. Cette liste est approximative, déterminée à partir de la complexité des expressions booléennes et demeure à l'appréciation du concepteur. On ne cherche pas à obtenir un résultat précis, on cherche surtout à exclure de la liste les points dont on sait pertinemment qu'ils ne feront pas partie des chemins critiques finaux. En effet, on aura alors sur ces derniers points aucune contrainte temporelle pour y insérer de la logi-

que combinatoire comme un multiplexeur ou un "ou exclusif". Si d'autres informations temporelles sont facilement accessibles comme les caractéristiques temporelles d'un bloc caractérisé ou des résultats d'essais de synthèse, elles peuvent être utilisées pour affiner la sélection des entrées des bascules afin de faciliter la sélection des modes locaux. En cas d'échec de ces techniques, il faudra affiner cette liste initiale par des résultats de synthèse. Cette liste sera appelée la liste des points critiques, elle pourra comporter un ordre implicite sur le caractère critique d'un point. Elle pourra être vide dès le départ si le circuit n'est pas critique en vitesse. On aura alors forcément une solution optimale en terme de taux de couverture, il suffit d'implanter un scan complet. Et on s'intéressera dans ce cas aux trois derniers critères d'optimisation.

On suppose désormais qu'on dispose d'une liste de points critiques. Dans un premier temps, en utilisant les méthodes décrites au VI.2 page 129, on recherche pour chaque point critique des modes locaux vérifiant à la fois que ce point fait partie des points de sortie des modes locaux, et que ces modes locaux ont un coût minimum en vitesse pour la liste des points critiques. Quand on ne trouve pas de modes locaux pour un point critique, dans le cas où on décide de ne pas insérer de multiplexeur, alors il faut utiliser soit des méthodes du type équilibrage de chemins (pour conserver un taux de couverture de 100%), soit des ATPGs séquentiels (ou des approches symboliques par des chemins de test). Dans le cas contraire, il faudra compter sur les performances de l'outil de synthèse utilisé pour l'optimisation en vitesse. Pour faciliter la tâche de l'outil de synthèse, il vaut mieux avoir au départ le minimum de points critiques.

Dans un deuxième temps, on considère les points critiques restants et leurs modes locaux associés (appelés les modes locaux critiques). On cherche alors à sélectionner une partie de ces modes locaux qui d'une part couvre un maximum de points critiques, et qui d'autre part forment un mode global acyclique ayant les propriétés de disjonction. La propriété de bijectivité du mode global ainsi formé découle des choix des modes locaux qui sont tous choisis ou rendus bijectifs. Les propriétés de chaînage avant et arrière seront obtenues par la suite en "interconnectant" les modes locaux obtenus par l'intermédiaire de chaînes de scan. Cette opération appelée chaînage des modes locaux est décrite au VI.3.4 page 143.

## 2 - Obtention du graphe d'incompatibilité des modes locaux pour la disjonction

On détermine des relations de non compatibilité pour la disjonction entre les modes locaux critiques. Ces relations de non compatibilité proviennent directement des définitions des disjonctions en entrée et en sortie. Pour savoir si deux modes locaux vont être compatibles, il suffit de réaliser deux intersections, celles de leurs ensembles de

points de sortie dans le cas de la disjonction en sortie, et celles de leurs ensembles de points d'entrée qui ne sont pas rebouclants ( $PE(m_l) - PER(m_l)$ ) dans le cas de la disjonction en entrée. Ces non compatibilités peuvent être recherchées par analyse des appartenances multiples à ces ensembles des entrées et sorties de bascules ainsi que des entrées et sorties primaires. Le résultat peut être représenté à l'aide d'un graphe non orienté où les sommets sont les modes locaux critiques, et où les arcs sont définis par la relation "n'est pas compatible pour la disjonction avec"). Ce graphe est appelé graphe d'incompatibilité. Le problème de recherche de compatibilité pour la disjonction se formalise alors en "trouver un sous-graphe généré par une partie des sommets du graphe, tel que toutes les composantes connexes ne comportent qu'un seul sommet<sup>14</sup>, et tel que l'ensemble des points de sortie couvre un sous ensemble maximal de l'ensemble des points critiques".

### 3 - Obtention du graphe maximal des modes locaux

Le graphe maximal des modes locaux est un graphe orienté dont les sommets sont les modes locaux critiques et dont les arcs sont définis de la façon suivante : un arc relie le mode  $m_l$  au mode  $m_{l'}$  si et seulement si ( $m_l$  et  $m_{l'}$  sont distincts et  $BT(m_l) \cap (BE(m_{l'}) - BER(m_{l'})) \neq \emptyset$ ). Comme pour le graphe précédent, on peut procéder par analyse des appartenances doubles de chacune des bascules du circuit à la fois aux bascules de transmission d'un mode local et aux bascules d'entrée non rebouclantes d'un autre mode local. Le problème de recherche de compatibilité pour la cyclicité se formalise alors en "trouver un sous-graphe généré par une partie des sommets du graphe, tel qu'il soit acyclique, et tel que l'ensemble des points de sortie couvrent un sous ensemble maximal de l'ensemble des points critiques".

Le problème global de sélection des modes locaux consiste d'abord à résoudre ces deux problèmes simultanément c'est-à-dire avec le même ensemble de sommets. Une heuristique non optimale est proposée avec possibilité de donner plus ou moins de priorité au caractère critique des points ou au nombre de ces points couverts ou encore au coût relatif en surface. Elles débutent toutes par des étapes de réductions du problème. Si la taille des graphes est faible, on peut faire des recherches de solutions optimales.

#### VI.3.2 Réductions du problème

On ne change pas le problème en supprimant, à l'intérieur du graphe maximal des modes locaux, tous les arcs entre des modes locaux qui sont incompatibles. Cela peut être fait au moment de la construction de ces deux graphes. De même les cycles qui uti-

<sup>14</sup>. Comme ce graphe est déjà sans cycles de longueur 1, cela revient à obtenir un graphe sans arêtes.

lisent des modes locaux incompatibles n'ont pas besoin d'être pris en compte.

Le problème de recherche de compatibilité pour la cyclicité devient alors "trouver un sous-graphe généré par une partie des sommets du graphe, tel que tout cycle comporte au moins deux sommets incompatibles, et tel que l'ensemble des points de sortie couvre un sous ensemble maximal de l'ensemble des points critiques".

De manière classique, on peut considérer seulement les composantes fortement connexes (CFCs) du graphe maximal des modes locaux pour faire les recherches de cycles.

Tous les sommets appartenant à des CFCs à un seul élément et pour lesquels il n'y a pas de relations d'incompatibilité pour la disjonction sont sélectionnés d'office, car ils n'interviendront pas sur les choix ultérieurs d'autres sommets. Seuls les sommets restants sont pris ensuite en considération.

### **VI.3.3 Heuristique proposée**

L'heuristique est de type glouton, basée sur un classement des modes critiques. Différents classements des modes critiques sont possibles. En effet, on peut considérer un premier ordre partiel des modes critiques donné par le nombre de points critiques associés à un mode critique. Un deuxième ordre partiel de type lexicographique est obtenu après un tri préalable des points critiques selon leur caractère critique. La comparaison de deux modes critiques se fait alors en comparant le point le plus critique de chacun d'entre eux, puis si ceux-ci sont égaux ou non comparables (par exemple s'ils ont des caractères critiques semblables) prendre les deuxièmes points critiques ; on poursuit pour les points suivants jusqu'à ce que tous les points soient épuisés ou jusqu'à ce qu'on soit arrivé à des points comparables et distincts. Dans le cas où on épuise la liste points critiques associée à l'un des modes critiques, alors on choisira en priorité celui qui a la liste la plus longue. De même, pour le premier ordre partiel, si deux ordres sont associés au même nombre de points critiques, on choisira en priorité celui qui est placé avant l'autre pour le deuxième ordre partiel. Le premier ordre favorise le nombre de points critiques associés à un mode local, alors que le deuxième favorise le caractère critique des points critiques associés aux modes critiques. Le coût relatif estimé en surface peut aussi servir à classer des modes critiques précédemment classés sur le même plan. Le premier ordre partiel correspond davantage à l'objectif même de la formulation précédente du problème de sélection des bascules. Cependant dans le cas où il y a des priorités pour le traitement des points critiques, le deuxième ordre peut être plus adapté. Pour l'heuristique elle-même, on suppose qu'on dispose donc d'une liste  $L$  triée des modes critiques. L'heuristique donne comme résultat un ensemble  $S$  de modes critiques.

Initialiser S à  $S = \emptyset$ ;

Tant que (L est non vide) faire

{

sélectionner le premier élément, m1 de L ;

si (m1 est compatible pour la disjonction avec tous les éléments de S)

/\* utilisation du graphe des non compatibilités\*/

alors

{

si (m1 appartient à la même CFC du graphe maximal que l'un des éléments déjà dans S)

alors

{

si (le sous-graphe du graphe maximal généré par m1 et les éléments de S dans la même CFC que m1 est acyclique)

alors ajouter m1 à S ;

}

sinon ajouter m1 à S ;

}

supprimer m1 de L ;

}

rendre (S);

### **VI.3.4 Chaînage des modes locaux**

Le chaînage des modes locaux s'effectue en ajoutant au mode global trouvé de nouveaux modes locaux transparents (portion de chaînes de scan). Il a pour objectif d'assurer les propriétés de chaînage avant et arrière pour le mode global final tout en conservant les propriétés de disjonction et d'acyclicité. Pour diminuer le coût en vitesse et en surface, on peut aussi chercher à ce que pour chacun des modes initiaux les entrées des bascules d'activation associées à des points d'activation à  $\varphi$  soit temporellement indépendantes des entrées des bascules d'entrée du même mode. Ce problème sera appelé problème d'indépendance des points d'activation à  $\varphi$ . Un deuxième problème



plus contraignant peut se poser, c'est celui du nombre d'entrées et de sorties primaires du circuit utilisables comme entrées et sorties primaires du mode global.

Soit  $ML$ , l'ensemble des modes locaux du mode global initial.  $\sum_{ml \in ML} |PEP(ml)|$ ,  $\sum_{ml \in ML} |PTP(ml)|$  et  $\max_{ml \in ML} (|PT(ml)|)$ <sup>15</sup> minorent le nombre d'entrées (ou de sorties) primaires utilisées pour le mode global final. Si l'un de ces nombres est supérieur à ce qui est disponible, alors on supprimera certains modes locaux, et/ou on ajoutera des pattes (en plus de celle qui permet d'activer le mode global), et/ou on emploiera plusieurs modes globaux ayant des entrées ou des sorties primaires communes afin de tester un circuit partitionné (les bascules des limites des partitions appartenant à plusieurs modes globaux à la fois). On agira de la même façon quand le nombre exact d'entrées (ou de sorties primaires) nécessaires sera connu.

A chaque mode local  $ml$  d'un mode global disjoint, on définit sa capacité sortante comme le nombre de ses points de transmission, et sa capacité entrante comme le nombre de ses points d'entrée de type bascule non rebouclants. Elles sont notées respectivement :  $cs(ml)$  et  $ce(ml)$ . A chaque arc  $a$  du graphe des modes locaux, on définit aussi sa capacité comme le nombre de bascules de transmission auxquelles il est associé. Elle est notée :  $c(a)$ . On prend la notation  $(ml, ml')$  pour désigner l'arc d'origine  $ml$  et d'extrémité  $ml'$ . On montre facilement le théorème suivant :

Un mode global disjoint a les propriétés de chaînage avant et arrière si et seulement si tout mode local  $ml$  du mode global vérifie :

$$\sum_{ml' \in \text{succ}(ml)} c((ml, ml')) = cs(ml) \text{ et } \sum_{ml' \in \text{pred}(ml)} c((ml', ml)) = ce(ml).$$

On considère un mode global  $ML'$  disjoint, alors tout mode local  $ml$  du mode global vérifie :

$$\sum_{ml' \in \text{succ}(ml)} c((ml, ml')) \leq cs(ml) \text{ et } \sum_{ml' \in \text{pred}(ml)} c((ml', ml)) \leq ce(ml). \quad (\text{xviii})$$

Si afin d'obtenir les propriétés de chaînage avant et arrière, des portions de chaînes de scan sont ajoutées à la fois pour relier aux entrées primaires les bascules isolées en entrée restantes, et pour relier aux sorties primaires les bascules isolées en sortie restantes, alors le nombre d'entrées (ou de sorties) primaires utilisées vaut :

<sup>15</sup>. C'est une conséquence du caractère acyclique du graphe des modes locaux.

$$\sum_{ml \in ML'} \left( |PEP(ml)| + ce(ml) - \sum_{ml' \in \text{pred}(ml)} c((ml', ml)) \right).$$

Le problème de chaînage se résume alors à ajouter des arcs au graphe des modes locaux de capacités définies de façon à vérifier les inégalités (xviii), à conserver un graphe acyclique, à vérifier les relations d'indépendance temporelle, et à minimiser :

$$\sum_{ml \in ML} \left( cs(ml) - \sum_{ml' \in \text{succ}(ml)} c((ml, ml')) \right) \text{ ou encore }^{16}$$

$$\sum_{ml \in ML} \left( ce(ml) - \sum_{ml' \in \text{pred}(ml)} c((ml', ml)) \right).$$

A chaque arc  $(ml, ml')$  ajouté va correspondre un mode de scan interconnectant  $c((ml, ml'))$  bascules de transmission de  $ml$  à  $c((ml, ml'))$  bascules d'entrée de  $ml'$ . Les portions de chaînes de scan ajoutées pour obtenir le mode global final n'interviennent ni sur la cyclicité du graphe, ni sur les relations d'indépendance entre des points de transmission dont les modes locaux associés étaient déjà présents dans le graphe initial. Cependant, si des bascules d'activation associées à des points d'activation à  $\varphi$ , sont à cette occasion mises en scan, alors on considère à nouveau les problèmes d'indépendance temporelle liés à ces points.

### VI.3.5 Construction et conception hiérarchique

Pour le scan complet la conception hiérarchique est simple car il suffit de mettre bout à bout les chaînes de scan de chaque bloc dans chaque niveau de hiérarchie. La ou les chaînes de scan obtenues et le signal de passage en mode scan traversent alors les différents blocs hiérarchiques.

A chaque bloc d'un niveau de hiérarchie est associé un ou plusieurs modes globaux. Il y a peu de chances que l'interconnexion des blocs fixée par le concepteur fournisse aussi un nouveau mode global acyclique ayant des propriétés de chaînage avant et arrière. C'est pourquoi les modes locaux non critiques (qui peuvent être choisis comme des modes de type scan), sont à traiter comme dans le cas des multiplexeurs utilisés pour le scan complet. Les signaux qui commandent les activations, sont aussi traités comme pour le scan complet. Par contre les entrées qui servent à l'activation et qui sont à  $\varphi$  peuvent être exploitées comme telles si on procède comme dans le cas du chaînage avant et arrière en tenant compte des relations d'indépendance temporelle entre les

<sup>16</sup> Il y a équivalence du choix des deux expressions à minimiser car lors d'un ajout d'un arc, le gain pour la première expression est aussi le gain pour la deuxième expression.

blocs d'une même hiérarchie. Au lieu de considérer des critères d'indépendance temporelle entre blocs dans une même hiérarchie, on peut être plus précis en s'intéressant aux sorties et aux entrées primaires du mode global associées à un bloc (c'est-à-dire à des ports d'entrée et de sortie du bloc). En effet, certains points de transmission en sortie du bloc peuvent ne pas dépendre de toutes les entrées primaires du mode global. Comme pour le chaînage avant et arrière (et implicitement aussi dans le cas du scan complet), il ne faut pas créer par regroupement et interconnexion des modes globaux cycliques. Les principes d'utilisation hiérarchique sont équivalents à ceux du scan complet, le résultat final donnera aussi des taux de couverture de 100%. Cependant des contraintes dues à des modes locaux critiques en "périphérie" des blocs hiérarchiques peuvent amener à en supprimer certains pour les remplacer par des structures de scan plus souples. De plus, dans le cas d'un mode global quelconque, la mise en œuvre de la génération hiérarchique des vecteurs de test devient plus complexe. Les vecteurs de test locaux restent les mêmes, mais il faut recalculer les vecteurs correspondant qu'il faut appliquer en entrée du circuit. L'opération n'est pas aussi simple que pour le scan complet où il suffit d'allonger les durées des étapes de décalage en entrée et en sortie. Ici, il est nécessaire de pouvoir resimuler et inverser les fonctions pour la génération des vecteurs de test effectifs globaux.

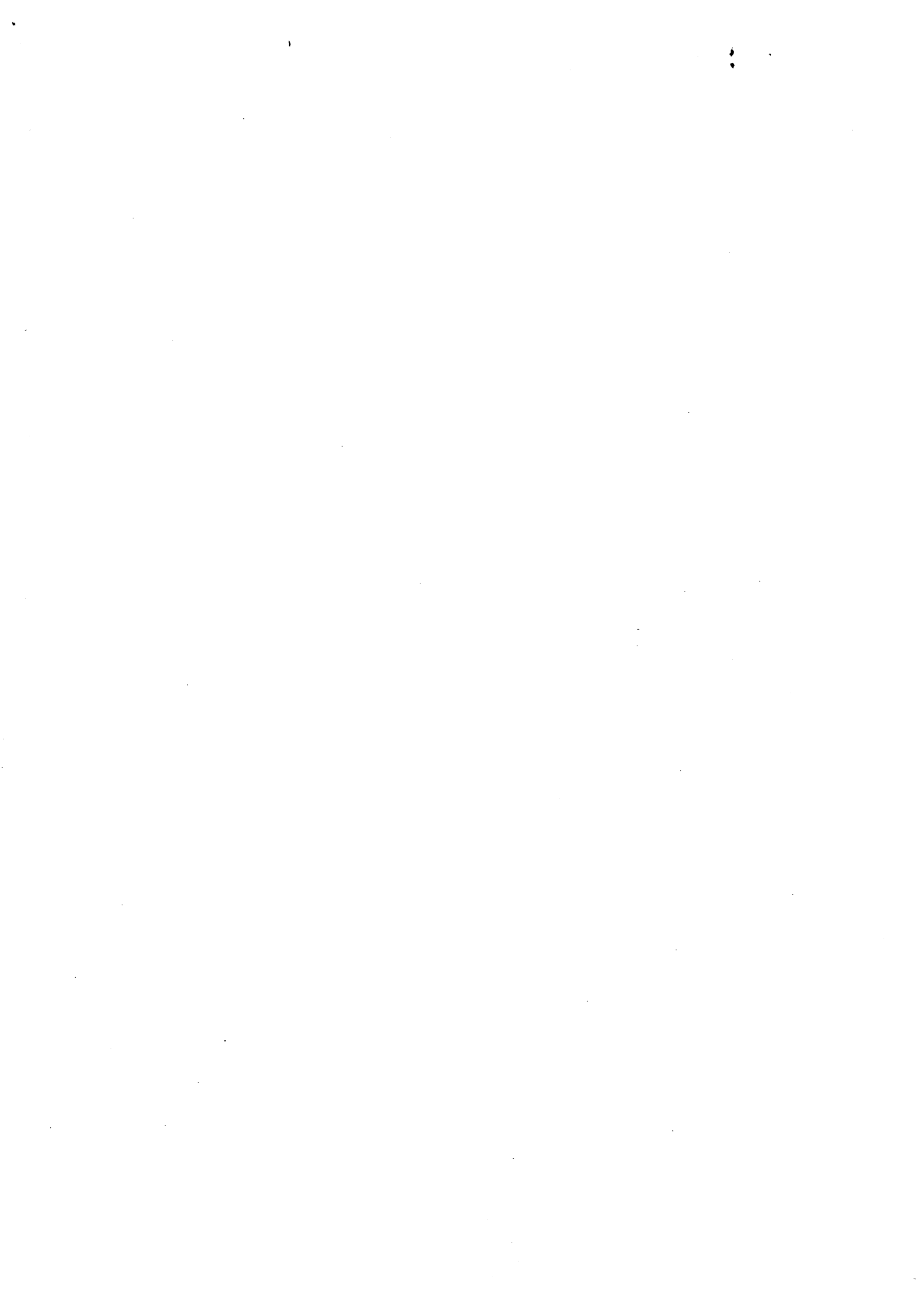
### **VI.3.6 Extensions envisageables**

Des améliorations de la méthode proposée sont envisageables en utilisant des chemins équilibrés et/ou des méthodes de test symbolique ou des ATPGs.

En effet, lorsqu'on a un point critique appartenant à un nuage (voir III.2 page 55), on peut rechercher à inclure ce nuage dans un chemin équilibré du graphe topologique afin qu'aucune logique ne vienne allonger le chemin critique considéré. Cependant, pour l'application des vecteurs de test, il faut alors pouvoir réaliser des modes de maintien et/ou il faut pouvoir ordonner les registres mis en scan. Ces contraintes peuvent être modélisées au niveau du graphe des modes locaux en terme de successeurs (ou de prédécesseurs) et d'indépendance temporelle qui permet le contrôle continu. L'utilisation de chemins équilibrés (s'ils existent) permet de conserver des taux de couverture de 100% pour les fautes séquentiellement testables. Dans des circuits fortement pipelinés, elle donnera aussi un gain appréciable en surface.

De même, lorsque certaines bascules ne sont ni employées comme des bascules de transmission ni dans des chemins équilibrés du graphe topologique alors les taux de couverture maximaux ne sont plus garantis, et on recherche des méthodes de test symbolique ou des ATPGs qui tirent profit des modes globaux déjà implantés pour tester

les nuages. Dans les deux cas, cela est réalisable en ajoutant des capacités d'utilisation des modes globaux. En effet généralement, seuls des chemins de transparence qui sont modélisables par des chaînes multiples de scan combinées avec des opérateurs simples comme l'inverseur, sont exploités de façon efficace.



## **Chapitre 3 : Application à des circuits de terminaison de ligne pour l'ATM**

Le circuit d'application décrit est un prototype hybride qui a ensuite évolué vers une version monolithique. Il sert de circuit de terminaison de ligne dans le réseau numérique large bande pour l'implantation de la couche physique de l'ATM. Il supporte des débits de 622 Mbit/s à la fois en émission et en réception. Après parallélisation sur huit bits et avant sérialisation, la fréquence de fonctionnement interne imposée reste de 78 MHz en pire cas industriel. Pour pallier les difficultés de l'impact des structures de scan classiques sur la vitesse, des nouvelles structures de test appelées modes globaux ont été implantées. Elles exploitent de nombreuses fonctionnalités du circuit précédemment décrites. La description et le processus d'obtention de modes globaux adaptés ont été réalisés manuellement au niveau VHDL-RTL. Etant donné l'indisponibilité de nombreux outils adaptés à la mise en œuvre du test structurel à partir de modes globaux quelconques, une solution temporaire de remplacement a été employée pour générer les vecteurs de test. Elle a consisté à dériver les vecteurs de test finaux en partant de vecteurs de validation fonctionnelle et en activant de manière aléatoire les modes globaux implantés. Le taux de couverture final ainsi obtenu est de 94.2% pour des fautes de type collage à un ou à zéro.



## **I. Introduction**

Le monde des télécommunications ne cesse d'évoluer. Les demandes de nouveaux services ne cessent de croître. Elles concernent aussi bien les transmissions de données, les services de nature audio-visuelle que le multimedia. Le réseau numérique à intégration de service (RNIS) présente certaines limitations pour répondre à toutes ces demandes, mais il est bien adapté à la téléphonie pour lequel il a été conçu. La limitation principale est le débit continu de 64 kbit/s (l'utilisation de liaisons multiples est possible mais engendre d'autres problèmes). La commande du réseau et la gestion des ressources ne pourraient faire face à la multiplicité des services et encore moins à des services à forte contrainte temps réel. Ces limitations sont à l'origine du RNIS à large bande [DICE88] à mode de transfert asynchrone, ATM (Asynchronous Transfer Mode). Il permet de résoudre de nombreux problèmes tels que l'adaptation aux débits, le multiplexage statistique (optimisation des ressources de transmission et de commutation), la qualité de l'information transmise, les contraintes de temps réel, l'indépendance entre le réseau et le service transporté, la souplesse d'exploitation.

Une validation de ces techniques et de la faisabilité de certains services pour le RNIS à large bande est en cours. Elle utilise le démonstrateur développé dans le cadre du projet européen RACE<sup>12</sup> 1022 "Technology for ATD"<sup>13</sup> [TGE90] [SCHO92]. Le succès du réseau ATM va dépendre de la disponibilité d'un ensemble de circuits qui implantent les premières couches du réseau ATM [PRYC91]. La caractéristique principale de ces circuits est leur fréquence élevée de fonctionnement. Ces fréquences correspondent à des débits recommandés par le CCITT<sup>14</sup> de 622,08 Mbit/s et de 155,52 Mbit/s.

Un composant clé de cet ensemble de circuits est le circuit de terminaison de ligne. Ce prototype a été développé par SGS-THOMSON sous la forme d'abord d'un module hybride [DIAZ93]. Il est appelé LTH (Line Terminator Hybrid) et contient deux circuits (voir figure 19) :

- Un circuit CMOS appelé LTP (Line Terminator Parallel) qui réalise le traitement parallèle des données sur des mots de 8 bits.

- Un circuit BiCMOS appelé LTS (Line Terminator Serial) qui réalise les conversions CMOS/ECL, ECL/CMOS, série/parallèle, et parallèle/série.

Ce module supporte des débits sur les lignes série à la fois de 622,08 Mbit/s et de

<sup>12</sup>. RACE : Research for Advanced Communications in Europe

<sup>13</sup>. ATD : Asynchronous Time Division switching

<sup>14</sup>. CCITT : Comité Consultatif International Télégraphique et Téléphonique, appelé maintenant UIT-T : Union Internationale des Télécommunications - Secteur de la Normalisation des Télécommunications



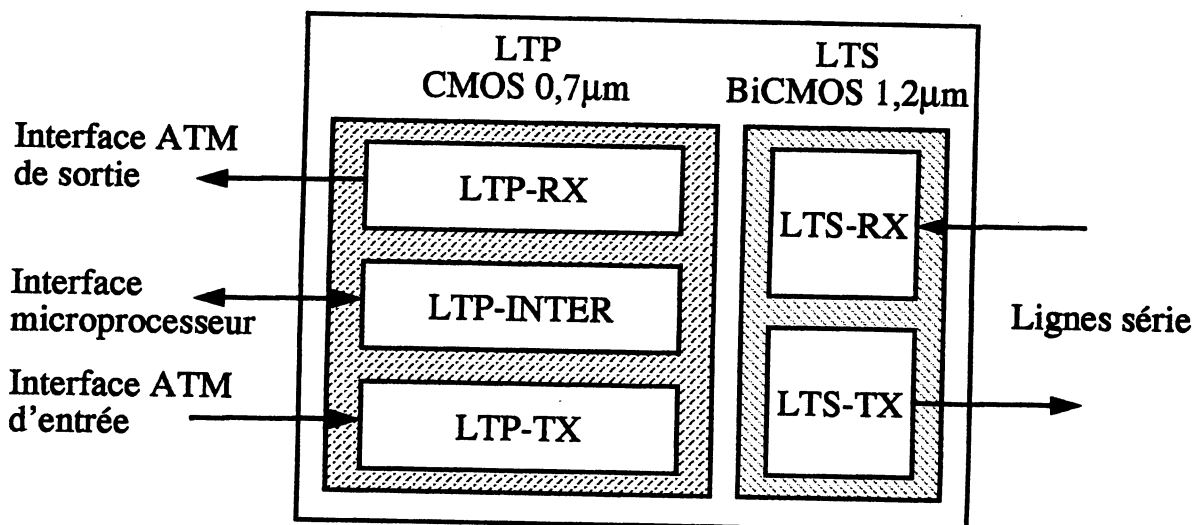


Figure 19 : architecture générale du LTH

155,52 Mbit/s. Les parties émission (TX), et réception (RX) sont indépendantes et peuvent fonctionner simultanément (aux mêmes fréquences ou à des fréquences différentes). Elles implantent les principales fonctions de la sous-couche de convergence de transmission [BULO92a] de la couche physique du réseau ATM [CCIT91a] [CCIT91b]. Il n'y a pas de transport direct sur les lignes série d'un signal d'horloge. Les données de la ligne série sont envoyées en phase avec une horloge interne dérivée de l'horloge provenant de l'interface ATM d'entrée par une multiplication par 8 de la fréquence grâce à un PLL<sup>15</sup>. Les données parallèles sur 8 bits en sortie du LTP-TX sont sérialisées (multiplexées) au passage par le LTS-TX. En réception, le processus est plus complexe, car on effectue un recouvrement d'horloge à partir des données reçues. Les données ainsi reçues et échantillonnées sont parallélisées (démultiplexées) avant d'être transmises au LTP-RX. Pour un débit en émission de 622,08 Mbit/s (155,02 Mbit/s), le LTS-TX fonctionne à 622,08 MHz (155,02 MHz) et le LTP-TX à 77,76 MHz (19,44 MHz). De même pour un débit en réception<sup>16</sup> de 622,08 Mbit/s (155,02 Mbit/s), le LTS-RX fonctionne à 622,08 MHz (155,02 MHz) et le LTP-RX à 77,76 MHz (19,44 MHz).

Les interfaces ATM de sortie (d'entrée) sont des interfaces parallèles où les données de la ligne série se trouvent sous forme de mots de 8 bits après (avant) traitement par le LTP. En plus de ces 8 bits et du signal d'horloge, l'interface ATM normalisée [ROIC91] pour le projet RACE 1022, possède 2 bits supplémentaires l'un pour indi-

<sup>15</sup>. PLL : Phase Locked Loop

<sup>16</sup>. Le débit en réception est donné par le débit en émission du LTH qui se trouve à l'autre bout de la ligne. La ligne physique peut être un câble coaxial ou une fibre optique.

quer le tramage par un signal de début de cellule et l'autre pour indiquer la présence d'une cellule morte (explications au II). Ces interfaces permettent de connecter le LTH à d'autres circuits du réseau ATM, et en particulier aux circuits qui servent à la réalisation de la couche ATM.

L'interface microprocesseur permet quant à elle le dialogue simple entre le LTH et un microprocesseur du type MOTOROLA 68000. Le microprocesseur configure ainsi le module (différents modes de fonctionnement sont possibles) et est averti principalement du nombre d'erreurs détectées sur les données reçues et des défaillances au niveau des lignes série. Le bloc LTP-INTER contient des registres de contrôle et de statut, et implante le protocole de l'interface normalisée pour le projet RACE 1022 [LOO90]. Ce bloc utilise une horloge de fréquence d'environ 40 MHz qui est générée par le LTS. Dans une nouvelle version du circuit de terminaison de ligne pour l'ATM [DIAZ94], une nouvelle interface microprocesseur a été implantée. Elle permet de sélectionner sur des broches externes le type d'interface utilisé (MOTOROLA 68000, INTEL 8086, ST9, ...). Cette version est une puce monolithique appelé LTM (Line Terminator Monochip) et réalisé dans la technologie BiCMOS4 (0,7 $\mu$ m) de SGS-THOMSON. La partie LTS a été redessinée, et des améliorations ont aussi été apportées pour tester de manière indépendante les blocs correspondants au LTS et au LTP. Cette version permet d'améliorer les performances des PLLs et de diminuer aussi les coûts de fabrication. On trouvera dans [DIAZ95] les résultats concernant cette dernière version et dans [BULO92b] ceux concernant le LTH.

En ce qui concerne le test du LTH, le bloc LTP-INTER ne pose pas de problèmes particuliers en terme de vitesse de fonctionnement. Par contre, les blocs LTP-TX et LTP-RX ainsi que le LTS ont demandé des attentions toutes particulières pour conjurer à la fois les optimisations en vitesse et les contraintes de test. Le test du LTP-TX et celui du LTP-RX ont largement fait appel aux méthodes présentées au chapitre précédent. A la base des fonctions employées, on retrouvera la plupart de celles décrites au II. Au III, on décrira la mise en œuvre et les résultats obtenus. Finalement au IV, un aperçu sera donné sur les principales méthodes employées pour tester le LTS seul ou à l'intérieur du LTH (la difficulté vient essentiellement de la présence de PLLs et non pas de la complexité des fonctions implantées) et pour utiliser un équipement de test limité à 200 MHz.

## II. Spécifications du LTP

Seuls les aspects les plus importants du circuit sont décrits ici. Une description beaucoup plus complète du LTP et du LTS se trouve dans [DIAZ91].

### II.1 Structure des données utilisées

Il est important de connaître le type de données que traite le LTP. Le LTP traite des paquets de bits de longueur constante appelés cellules. A l'intérieur du LTP, les bits sont regroupés en octets. La longueur des cellules est normalisée et vaut 53 octets. Cependant le LTP possède aussi les modes de fonctionnement en émission et/ou réception avec des cellules à 60 octets. Les cinq premiers octets d'une cellule (voir figure 20)

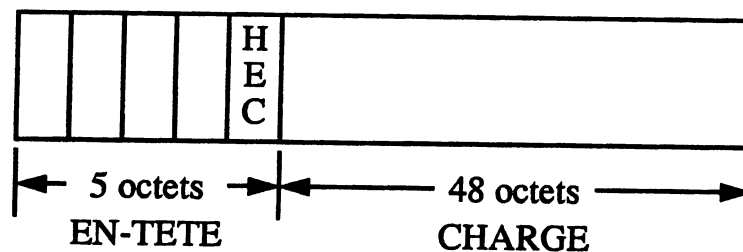


Figure 20 : format d'une cellule

forment ce qui est appelé l'en-tête de la cellule, et les 48 (ou 55) autres forment ce qui est appelé la charge. La charge est un champ d'information qui doit être transporté de manière transparente à travers le réseau. L'en-tête sert en particulier au routage de la cellule dans le réseau. Les principales fonctions du LTP sont de reconnaître le cadrage cellule au niveau de la réception, et de préparer les données au niveau de l'émission afin d'assurer le cadrage correct en réception. Le mécanisme pour réaliser le cadrage cellule est basé sur l'algorithme HEC (Header Error Check que l'on peut traduire par *contrôle d'erreur sur l'en-tête*). Cela signifie que si un syndrome HEC correct est reconnu pour plusieurs cellules consécutives, on suppose alors que l'hypothèse de frontière de cellule faite dans le flot des bits entrants, était correcte. Pour améliorer la sécurité, l'efficacité du mécanisme de cadrage cellule et le recouvrement d'horloge, on brouille chaque cellule émise, et on la débrouille après réception. Après le cadrage cellule, le LTP détecte et corrige des erreurs sur l'en-tête. L'en-tête est un code de redondance<sup>17</sup>. Son calcul est similaire à celui d'un code de redondance cyclique, CRC (Cyclic Redundancy Code) [PETE76].

Le LTP traite deux types de cellules : les cellules de données et les cellules mortes. Les cellules mortes (voir figure 21) ont un en-tête fixé. Le contenu de la charge est "6A"

<sup>17</sup>. L'en-tête est lié à un code polynomial non cyclique.

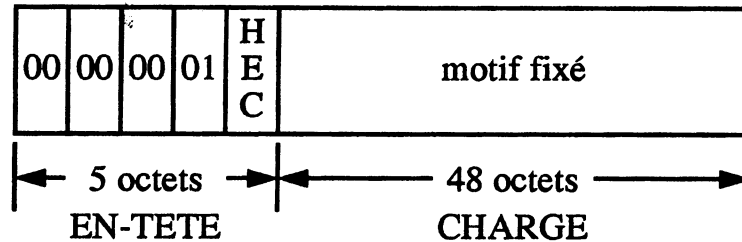


Figure 21 : cellule morte

répété 48 fois. Les cellules permettent de maintenir un flot continu de cellules dans le flot ATM. C'est-à-dire quand aucune cellule (ou aucune cellule significative) n'est à transporter, on insère des cellules mortes. Ce flot continu de cellules évite des pertes de synchronisation du cadrage cellule et du débrouillage dues à l'absence de cellules significatives.

## II.2 Fonctions d'émission

Le bloc LTP-TX implante les fonctions suivantes :

**Détection d'erreur d'en-tête.** Sur la cellule venant de l'interface ATM d'entrée, le LTP détecte si le champ HEC est correct. Après avoir ajouté le motif "01010101"<sup>18</sup> au champ HEC, le polynôme associé aux bits de l'en-tête doit être divisible par  $X^8 + X^2 + X + 1$ . S'il ne l'est pas, une erreur est reportée dans le registre de statut.

**Détection de la perte de la synchronisation cellule à l'interface ATM.** Cette fonction est réalisée sur le signal de début de cellule. Le LTP détecte si ce signal arrive périodiquement tous les 53 octets (ou 60 octets si la cellule en compte 60). S'il n'arrive pas, alors le flot de cellules n'est pas continu et on reporte le problème dans le registre de statut.

**Insertion d'une cellule morte.** Le LTP génère en interne un cellule morte lorsque le signal d'indication de cellule morte est levé sur l'interface ATM ou lorsque une erreur est détectée sur le champ HEC.

**Brouillage.** Un mécanisme de brouillage à échantillons distribués, DSS (Distributed Sample Scrambling), est implanté dans le LTP. Cela permet de brouiller le flot de cellules ATM par addition modulo 2 d'une séquence pseudo-aléatoire. Le polynôme utilisé pour générer cette séquence est  $X^{31} + X^{28} + 1$ . Le brouillage peut aussi être désactivé via le registre de contrôle.

**Génération du champ HEC.** Après brouillage des quatre premiers octets de l'en-tête,

<sup>18</sup>. Ce motif permet de détecter des légers glissements de bit dans la récupération des données en réception.

un champ HEC correct est généré. On ajoute alors 2 bits extraits de la séquence de brouillage. Ces 2 bits fournissent assez d'information pour retrouver en réception la séquence exacte de brouillage utilisée. Le champ HEC véhicule aussi de l'information pour le cadrage cellule et pour la détection et la correction d'erreurs sur l'en-tête.

### **II.3 Fonctions de réception**

Le bloc LTP-TX implante les fonctions suivantes :

**Cadrage cellule.** Le LTP réalise un cadrage cellule (incluant un cadrage octet) tout en travaillant avec des octets. Il se conforme aux recommandations CCITT pour rechercher et valider un cadrage par des vérifications des champs HEC.

**Débrouillage.** Les données sont débrouillées après que le processus de synchronisation du débrouilleur est réalisé. Ce processus commence en même temps que celui du cadrage de cellule. 16 cellules consécutives bien cadrées et sans erreur sur l'en-tête fournissent la synchronisation du débrouilleur (les  $16 \times 2 = 32$  bits d'information reçus permettent de déterminer l'état du LFSR à 31 bascules utilisé pour le brouillage). L'algorithme utilisé est conforme aux recommandations CCITT et reconnaît aussi si les données ont été brouillées ou non.

**Contrôle et correction d'erreurs d'en-tête.** Quand le cadrage cellule et la synchronisation du débrouilleur sont obtenus, le champ HEC est utilisé pour détecter et corriger les erreurs de transmission. Le code correcteur permet de détecter des erreurs sur plusieurs bits (erreurs multiples), mais il ne peut corriger que les erreurs sur un seul bit (erreurs simples). Le nombre d'erreurs apparues est compté. En accord avec les recommandations du CCITT, une erreur simple ne sera pas corrigée si la cellule précédente avait aussi un en-tête erroné. Quand l'en-tête est erroné et n'est pas corrigé, la cellule est supprimée et est remplacée par une cellule morte. En fonction de la répartition et du type d'erreurs apparues, le LTP peut considérer que le cadrage cellule et/ou la synchronisation du débrouilleur sont perdus et qu'il faut alors recommencer les algorithmes de cadrage et de synchronisation.

**Détection des cellules mortes.** Quand le cadrage cellule et la synchronisation du débrouilleur sont obtenus, le LTP détecte les cellules mortes entrantes.

**Signaux d'indication de début de cellule et de cellule morte.** Lorsque les cellules sortent du LTP-RX par l'interface ATM de sortie, deux signaux sont générés pour indiquer le début d'une cellule et la présence d'une cellule morte.

**Calcul du champ HEC après débrouillage.** Après le débrouillage et la correction (ou l'absence de correction) de l'en-tête, le champ HEC n'est plus correct, donc il est recal-

culé.

#### **II.4 Fonctions d'interfaçage avec un microprocesseur**

Les principaux registres présents pour l'interfaçage avec un microprocesseur sont les suivants, le microprocesseur va pouvoir les lire et les écrire :

**Le registre de contrôle.** Le microprocesseur configure le LTH via le registre de contrôle. Les principaux paramètres concernés sont : la longueur de la cellule (53 ou 60 octets), le brouillage DSS (actif ou non), le contrôle du champ HEC au niveau de l'interface ATM d'entrée, les débits en émission et en réception (622,08 Mbit/s ou 155 Mbit/s).

**Le registre de statut.** Il mémorise chaque événement apparaissant dans le LTH. Les principaux événements générés par le LTH sont les suivants :

- Dans le LTS : perte de réception du signal série, perte du recouvrement d'horloge, perte de la génération de l'horloge série (utilisée pour l'émission).
- Dans le LTP-TX : perte du signal de début de cellule sur l'interface ATM d'entrée, erreur sur l'en-tête de la cellule entrant.
- Dans le LTP-RX : perte du cadrage cellule, débordement du "compteur d'erreurs simples", débordement du "compteur d'erreurs multiples".

**Le registre d'interruption<sup>19</sup>.** Les interruptions sont générées par les levées des bits du registre de statut.

**Les registres de masques.** Deux registres de masques sont présents. Le premier valide le registre d'interruption, et le second valide le registre de statut. Le masquage est fait bit à bit.

**Les registres compteurs d'erreurs.** Le microprocesseur peut contrôler la qualité de la transmission ATM et calculer le taux d'erreurs sur la ligne série ou BER (Bit Error Rate) en lisant deux registres internes : le "compteur d'erreurs simples" (sur 10 bits), et le "compteur d'erreurs multiples" (sur 6 bits).

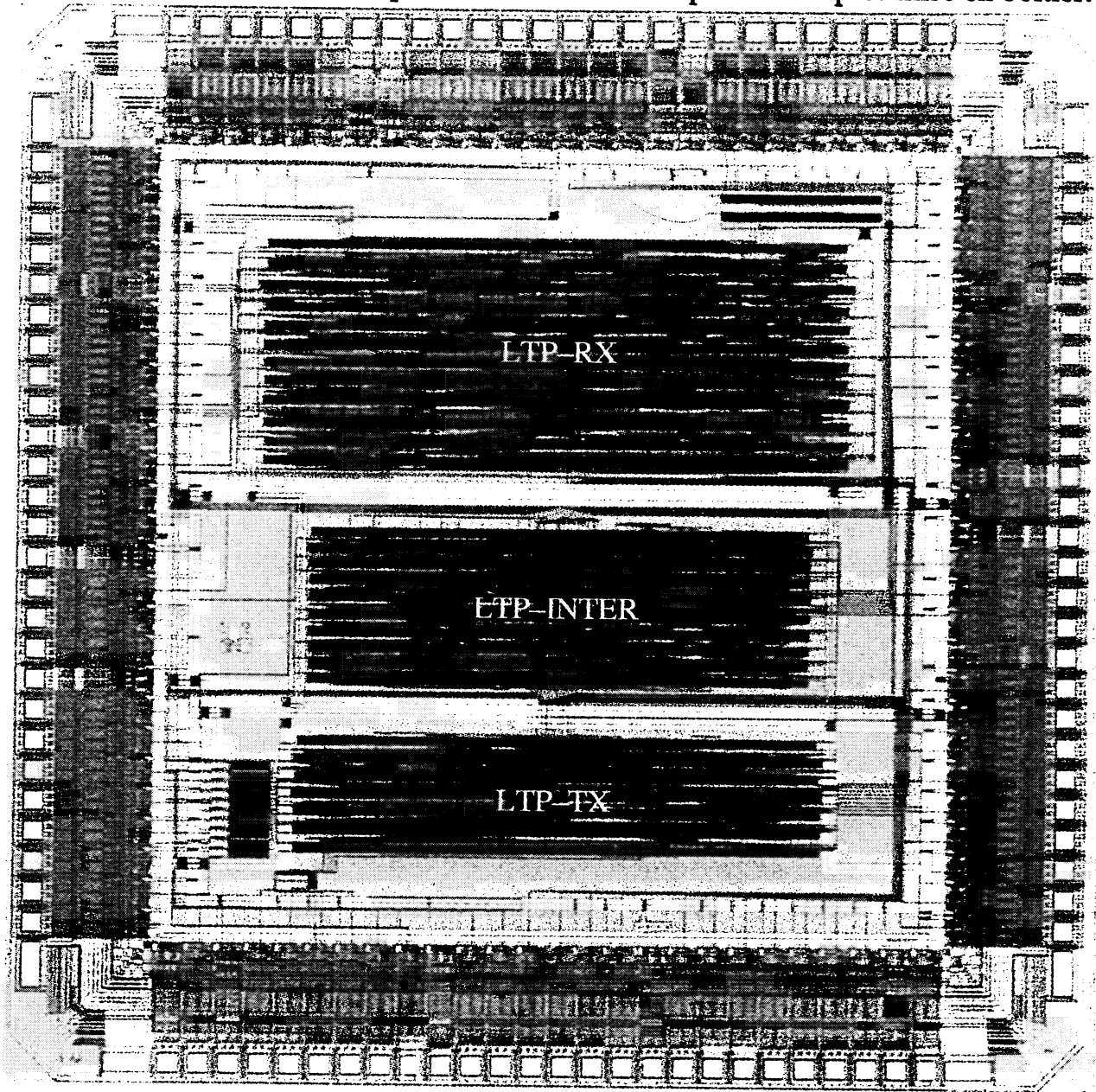
#### **II.5 Résumé des caractéristiques**

Les autres principales caractéristiques du circuit LTP sont les suivantes :

- Technologie CMOS 0,7  $\mu\text{m}$ , deux niveaux de métal de SGS-THOMSON (HCMOS4T).

<sup>19</sup>. Un protocole de vérification de l'échange des données avec le microprocesseur a aussi été implanté. En cas d'erreur lors d'un échange, un bit d'interruption spécial peut être levé.

- Intégration d'environ 8500 portes à partir d'une bibliothèque de cellules caractérisées.
- Taille de  $27,58 \text{ mm}^2$  (imposée par le nombre de plots).
- Fréquence maximum de fonctionnement obtenue sous pointes<sup>20</sup> de 188 MHz, fréquence de fonctionnement supérieure à 110 MHz en pire cas<sup>21</sup> après mise en boîtier.



**Figure 22 :** microphotographie du LTP

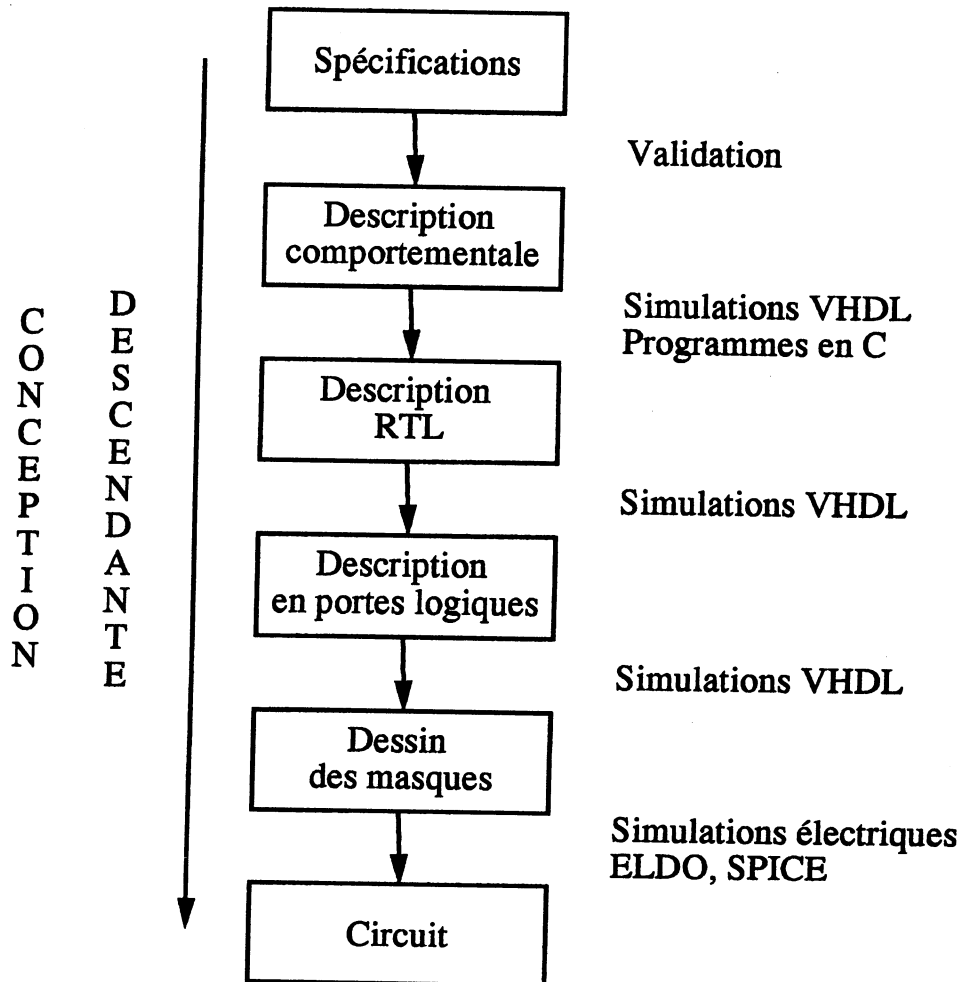
<sup>20.</sup> Avec une température ambiante de  $25^{\circ}\text{C}$  et une alimentation à 5,5V

<sup>21.</sup> Avec une température de fond de boîtier de  $130^{\circ}\text{C}$  et une alimentation à 4,5V

### III. Une conception orientée test et vitesse

#### III.1 Une méthode de conception descendante

L'approche de conception utilisée est une approche classique descendante [DIAZ86] essentiellement basée sur le langage VHDL (voir figure 23) et des outils de synthèse.



**Figure 23** : schéma simplifié de la méthode de conception du LTP

L'optimisation en vitesse s'est faite à la fois au niveau de l'architecture et au niveau de l'outil de synthèse SYNOPSIS. Des vérifications de contraintes temporelles ont été réalisées à l'aide de simulations rétro-annotées et à l'aide de l'outil d'analyse temporelle TATOO [BENK90] [BENK91]. Afin d'optimiser en vitesse de manière plus efficace et notamment afin d'éviter de nombreuses resynthèses orientées vitesse, les descriptions des structures de test ajoutées ont été faites dès le départ au niveau VHDL-RTL.



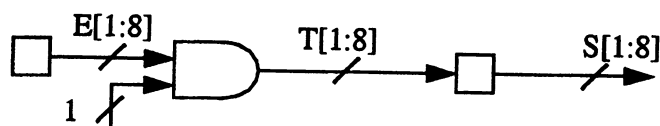
## III.2 Obtention des modes locaux

### III.2.1 Bascules des pipelines

Dans chacune des parties LTP-TX et le LTP-RX du LTP, un pipeline a été implémenté. Entre chaque registre (à huit bascules) des pipelines, des opérateurs élémentaires sont employés.

L'opérateur employé le plus simple est celui de l'identité, c'est-à-dire un opérateur qui transmet le mot sans le modifier. Il est alors vu soit comme un mode local transparent à huit points d'entrée, soit comme huit sous-modes locaux transparents à une entrée. Ces modes locaux n'ont pas de signaux d'activation.

L'opérateur, présenté figure 24, sert à remettre à zéro le mot qui se trouve dans le pipe-



signal de remise à zéro actif à 0

Figure 24 : opérateur de remise à zéro

line pour des initialisations synchrones. Il peut aussi servir comme mode local transparent. A la différence du précédent, il a un signal d'activation. Celui-ci, n'est pas à  $\varphi$ , et il doit être forcé à 1 lors de l'activation. Cela est très facilement réalisable en insérant une seule porte "ou" et en connectant son entrée restante à un signal d'activation global actif à 1. L'opérateur multiplexeur est aussi employé dans le LTP-TX et le LTP-RX pour mettre à jour par exemple le champ HEC calculé par ailleurs. Deux modes locaux à huit entrées sont alors possibles.

Un opérateur un peu plus complexe additionne modulo 2 bit à bit une valeur au mot du pipeline. Il peut s'implanter à l'aide d'une porte "ou exclusif" entre deux bascules consécutives du pipeline (voir figure 25). La deuxième entrée du "ou exclusif" est le signal

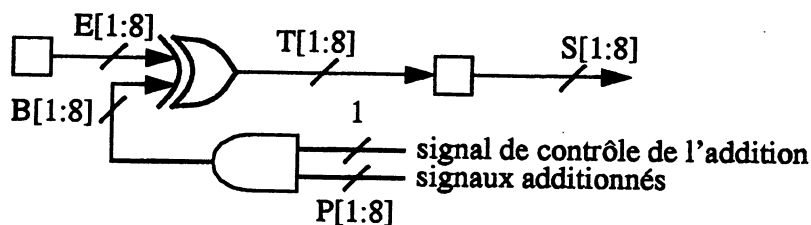


Figure 25 : opérateur d'addition modulo 2 contrôlé

additionné. On peut le considérer comme un signal d'activation à  $\varphi$ , mais alors il faudra en tenir compte au niveau des indépendances temporelles. Sinon, il suffit de le contrôler à une valeur fixée 0 ou 1. Le contrôle à 0 est facile en forçant à 0 le signal de contrôle

de l'addition. Cela n'induit pas a priori de coûts particuliers sur le chemin critique car ce sont les signaux additionnés qui risquent d'être critiques. Le coût en surface est lui aussi faible car ces signaux à contrôler à 0 sont propres à un ou plusieurs registres. Dans le cas du LTP-TX, on l'utilise pour ajouter la séquence de brouillage. Dans le cas du LTP-RX, on l'utilise pour ajouter la séquence de débrouillage ou pour ajouter un bit de correction d'erreur. Ce bit de correction d'erreur décode l'état d'un mot appelé syndrome qui caractérise l'indice du bit où il y a erreur. Plus précisément, à chaque bit de correction correspond une unique valeur du syndrome qui indique la présence d'une erreur simple. La correction d'erreur se fait alors en prenant le complément du bit correspondant dans le mot code, ce qui revient à lui ajouter 1. Le mot code comporte 40 bits et le syndrome en comporte 8. Le syndrome est aussi utilisé pour identifier la présence d'une erreur simple ou multiple. Tous ces décodages du syndrome induisent de nombreuses équations utilisant les mêmes variables et donc des difficultés pour minimiser les sortances des cellules précaractérisées de l'implantation. Par conséquent, c'est surtout ce facteur qui va augmenter le temps de propagation à travers ce bloc combinatoire de décodage (modélisable par un demi-PLA). Avec la même structure, on peut considérer un deuxième mode local, c'est celui obtenu en forçant le signal d'addition à 1, et en considérant les entrées additionnées comme les entrées du mode local et les entrées du pipeline comme des points d'activation.

### **III.2.2 Bascules de calcul de syndrome et de champ HEC**

Le calcul de syndrome ou de champ HEC se fait avec des opérateurs galoisiens. Le calcul est aussi fait sur des mots de 8 bits. Les bascules de transmission sont aussi des bascules de rebouclage. Dans [ALBE90] une méthode générale de parallélisation de codes à redondance cyclique, CRCs (Cyclic Redundancy Codes), à l'aide de fonctions de transfert associées à des filtres digitaux est présentée. On peut aussi évaluer manuellement ou non les équations finales par calcul formel des nouvelles équations après chaque itération (le nombre d'itérations à considérer correspond au degré de parallélisme employé) [LEER77] [PERE83] [RAMA88]. Pour l'activation du mode local, il suffira de contrôler le signal de remise à zéro. Les méthodes algébriques employées pour calculer et inverser les équations au niveau de ces opérateurs ont été implantées en LISP (LE\_LISP) sur station de travail SUN. Elles ont principalement servi à prendre en compte les évolutions des spécifications temporaires données par le CCITT.

### **III.2.3 Bascules des générateurs de séquences pseudo-aléatoires**

Les générateurs de séquences pseudo-aléatoires du LTP-TX et du LTP-RX utilisés pour le brouillage et le débrouillage sont semblables. Cependant, pour le débrouillage,

certain bits d'états du LFSR sont aussi modifiables en fonction des valeurs des bits échantillons récupérés pour réaliser la synchronisation du débrouilleur. Le polynôme utilisé pour générer cette séquence,  $X^{31} + X^{28} + 1$  est primitif. Après parallélisation des LFSRs, certains chemins pouvaient devenir critiques. On a donc décidé de ne mettre que huit bascules en scan pour chacun des deux LFSRs. On savait que cette structure était parfaitement contrôlable et observable (voir les méthodes de contrôle et d'observation pour un séquenceur utilisant un LFSR au 3 du VI.2.2 page 137).

### **III.2.4 Bascules des séquenceurs**

La présence de flots continus de cellules aussi bien en émission qu'en réception a favorisé l'emploi de séquenceurs. Les nombres d'états utiles de ces séquenceurs sont donnés par la longueur des cellules, c'est-à-dire 53 ou 60 selon le mode de fonctionnement programmé. Les modifications de séquenceurs proposées au 3 page 137 ont été implantées pour les séquenceurs du LTP-TX et du LTP-RX. On a vérifié expérimentalement en effectuant plusieurs synthèses de machines d'états finis avec des codages différents que l'on gagnait effectivement quelques nanosecondes sur les chemins critiques tout en fournissant à la fois contrôlabilité et observabilité de ces séquenceurs.

### **III.3 Obtention et activation des modes globaux**

Des problèmes d'indépendance temporelle liés à des opérateurs galoisiens ont amené à définir plusieurs modes globaux. Ces modes globaux utilisent les modes locaux précédemment décrits. On leur a donné des noms qui symbolisent les transferts des données par rapport aux chemins de données dans le LTP-TX et le LTP-RX. Comme le LTP-TX et le LTP-RX utilisent des horloges différentes, on distinguera les modes globaux associés au LTP-TX de ceux associés au LTP-RX même s'ils ont une certaine analogie dans leur utilisation.

Il y a d'abord les modes transparents : transparent émission (TRTX) et transparent réception (TRRX). Dans ces modes, le circuit utilise au maximum les pipelinages du circuit et les données sont véhiculées de manière transparente ; on ne fait que traverser des modes locaux transparents (scan). Pour TRTX, les 8 bits de données de l'entrée ATM sont les 8 entrées de scan de 8 chaînes de scan, et les sorties du LTP-TX à l'interface avec le LTS-TX sont les sorties des chaînes de scan. Pour TRRX, les 8 bits de données à l'interface avec le LTS-RX sont les 8 entrées de scan de 8 chaînes de scan, et les 8 bits de données à l'interface ATM de sortie sont les sorties des chaînes de scan. Ces chaînes de scan sont appelées les chaînes de transparence.

Les autres modes globaux utilisés en conjonction avec les modes transparents ne pos-

sèdent pas les propriétés de chaînage avant et arrière. Les modes de chargement : **LOADTX** et **LOADRX**, permettent de transférer des données des chaînes de transparence vers d'autres bascules du circuit. Les modes de lecture : **READTX** et **READRX** permettent de transférer des données de bascules du circuit vers les chaînes de transparence. Les combinaisons exactes d'application (ordre, durée) des modes globaux sont déterminées en fonction des nuages à tester, et des sorties de nuages observées (on a employé des méthodes un peu plus générales que celles présentées au chapitre précédent).

Les modes associés au **LTP-TX** et au **LTP-RX** sont activables de manière indépendante à partir d'entrées distinctes du circuit. Deux entrées de test **TTX2** et **TTX1** sont attribuées aux modes de fonctionnement du **LTP-TX** (**TRTX**, **LOADTX**, **READTX**, **NORMALTX**), deux autres entrées de test **TRX2** et **TRX1** sont attribuées aux modes de fonctionnement du **LTP-RX** (**TRRX**, **LOADRX**, **READRX**, **NORMALRX**). Les modes **NORMALTX** et **NORMALRX** sont simplement les modes de fonctionnement du **LTP-TX** et du **LTP-RX** lorsqu'on ne les teste pas. Dans chacun des deux cas du **LTP-TX** et du **LTP-RX**, deux signaux sont nécessaires et suffisants pour coder les 4 modes de fonctionnement. Les décodages fournissent un signal pour chaque mode de test (**TR**, **LOAD**, **READ**), qui lorsqu'il vaut 1 indique que le mode est activé. Pour diminuer l'impact potentiel du décodage des modes de test sur les chemins critiques, on a ajouté une bascule derrière chacun des signaux de décodage (on n'a pas créé des chemins critiques qui ne seraient effectifs que lorsque le test est mis en œuvre). Cela fait en tout 6 bascules ajoutées.

### **III.4 Génération des vecteurs de test et résultats**

L'idéal pour le test de ce circuit aurait été de disposer à la fois d'un outil d'aide au choix des modes locaux et globaux, et surtout d'un outil pour la génération des vecteurs de test finaux. Si un choix manuel des modes peut se concevoir, la génération manuelle des vecteurs de test finaux n'est pas raisonnablement envisageable. De même le développement d'outils qui permettent l'exploitation de modes globaux quelconques n'était pas non plus réalisable par une personne seule en un temps "limité". Il aurait fallu développer dans un même environnement des outils de simulation classique, d'inversion de certaines classes de fonctions, de simulation partielle arrière (sens du temps renversé), d'ATPG combinatoire et de simulation de fautes. On disposait alors de l'outil de génération de vecteurs de test pour des parties combinatoires, intégré à l'outil de synthèse de **SYNOPSIS** (Test Compiler). On pouvait donc générer des vecteurs de test pour chaque nuage combinatoire. Il restait alors en théorie à amener et observer ces résultats

par les modes globaux. Cela n'a pas été fait car la mise en œuvre restait encore trop complexe (problèmes de références croisées entre la synthèse et la description VHDL-RTL, de simulation classique et de simulation arrière). Une autre solution tentée puis abandonnée consistait à utiliser un ATPG séquentiel industriel (INTELLIGEN). Les problèmes venaient de formats de listes de nœuds (netlists), de formats de vecteurs de test, de description des cellules de la bibliothèque utilisée et de difficultés à aider l'ATPG en décrivant les modes locaux et/ou globaux. On s'est alors tourné vers une solution combinant un choix aléatoire des modes globaux précédemment définis et un simulateur de fautes : l'accélérateur XP100 de ZYCAD [XP100].

Le circuit étant un prototype, on a sacrifié en partie le taux de couverture des vecteurs de test utilisés au profit des performances en vitesse du circuit qui étaient primordiales pour ce projet. Cependant, une utilisation de tout ou partie de ce circuit dans une phase d'industrialisation nécessitera des vecteurs de test supplémentaires. Ceux-ci pourront être obtenus sans modification des structures de test en utilisant par exemple des ATPGs plus adaptés ou encore en approfondissant la méthode de test aléatoire adaptée qui a été effectivement employée. Cette méthode a permis d'obtenir un taux de couverture de l'ordre de 94,2%.

Le principe utilisé pour cette méthode de test aléatoire a consisté à partir de vecteurs utilisés pour la validation fonctionnelle du circuit et à modifier de manière aléatoire les signaux de contrôle des modes de test tout en conservant les autres signaux d'entrée.

Le modèle de fautes employé dans les simulations de fautes est un modèle de collage à un et à zéro des nœuds du circuit. Dans un premier temps le simulateur regroupe les fautes équivalentes (fautes situées sur des chaînes d'inverseurs par exemple). Dans un deuxième temps, il recherche aussi certaines fautes qui peuvent être facilement déclarées non détectables. La non détectabilité est donnée par des mesures de testabilité. Par exemple une sortie conjuguée non connectée d'une bascule, est non observable, donc le collage à un et à zéro sur ce nœud est non détectable. On exclut ces fautes dans le calcul du taux de couverture. Il pourrait cependant encore rester des fautes non détectables à cause de phénomènes de redondance logique. Initialement, il restait 6245 fautes "réelles" qui ont été simulées. Le tableau suivant donne les longueurs des vecteurs de test utilisés ainsi que les taux de couverture obtenus. A la suite de ces stimuli, on a augmenté le taux de couverture en employant des vecteurs de test fonctionnels faisant certaines vérifications de manière beaucoup plus systématique. Puisque les séquences employées étaient environ 5 fois plus longues, afin de diminuer la durée des simulations de fautes, on s'est intéressé seulement aux fautes non couvertes par (6), soit 479

fautes. Le taux final cumulé obtenu est de 94,2%. Il se répartit approximativement

stimuli employés	cycles d'horloge	taux de couverture	commentaires
ga2 (1)	10111	77,7%	différents stimuli utilisés pour la validation du LTP
tr (2)	6405	56,6%	
nog (3)	3557	44,5%	
gar (4)	5029	54%	
(1+2)	16516	81,4%	cumul des vecteurs précédents
(1+2+3)	2073	83,5%	
(1+2+3+4) = 5	25102	85,7%	
(5+1')	35213	90,6%	k' signifie stimuli k avec des modes de test aléatoires
(5+1'+2')	41618	91,3%	
(5+1'+2'+3')	45175	92,1%	
(5+1'+2'+3'+4') = 6	50204	92,3%	

comme suit pour les trois blocs composant le LTP : TX (99,1%), RX (95,4%) et INTER (89,0%).

#### IV. Autres emplois des modes globaux

Le test du LTH commence par un test sur plaquettes des circuits LTP et LTS. Après un premier tri, seuls les circuits jugés bons sont montés en un module hybride<sup>22</sup>. Le coût du montage en hybride étant particulièrement élevé, il était important de réaliser les tests des tranches de silicium sous pointes à haute vitesse. Ce test était particulièrement critique pour le circuit LTS qui fonctionne à 622 MHz et qui comporte deux PLLs. Des méthodes de test adaptées ont été implantées, elles tiennent aussi compte de l'environnement de test disponible [BULO93]. Ce dernier est constitué d'une station de validation IMS ATS 200 qui fonctionne jusqu'à 200 MHz. Une fois le module hybride monté, on pouvait encore tester le LTS comme s'il était seul en utilisant les modes transparents TX et RX du LTP comme le montre la figure 26. De même pour tester le LTP à l'inté-

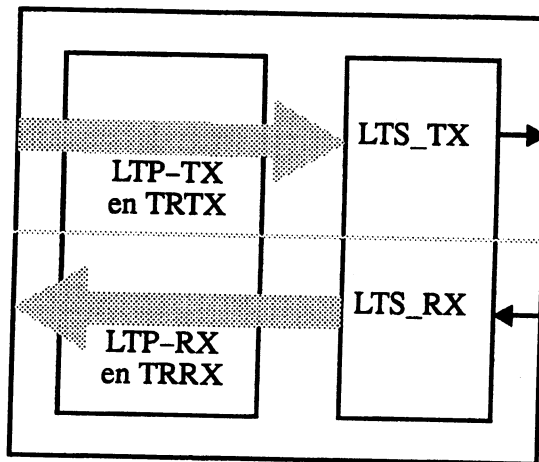
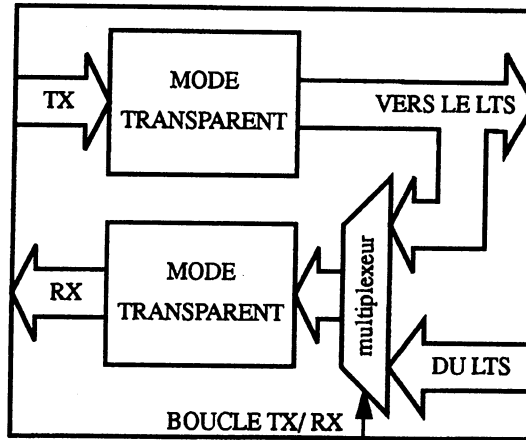


Figure 26 : test du LTS à l'intérieur du LTH

rieur du LTH comme s'il était seul, et ainsi s'affranchir facilement de certains problèmes de synchronisation d'horloge, on a ajouté un mode fonctionnement supplémentaire. Ce mode appelé boucle TX/RX est activé par une patte supplémentaire pour le test. Il interconnecte, lorsqu'il est activé, les sorties des chaînes de transparence du LTP-TX aux entrées des chaînes de transparence du LTP-RX (voir figure 27). L'horloge utilisée est aussi multiplexée. Ainsi, pour tester le LTP-TX, on active cette boucle pour observer les résultats en sortie des chaînes de transparence du LTP-TX via les chaînes de transparence du LTP-RX. Autrement dit, on maintient tout au long du test du LTP-TX le mode TRRX. Réciproquement, pour le test du LTP-RX, cette boucle est aussi activée, on maintient tout au long du test du LTP-RX le mode TRTX. Cela permet d'amener en entrée des chaînes de transparence du LTP-RX les vecteurs de test via le LTP-TX. Dans la version monolithique (LTMONO), des améliorations ont été

<sup>22</sup>. Certains LTPs et LTSs ont aussi été montés en boîtier.



**Figure 27 :** test du LTP à l'intérieur du LTH

apportées pour mieux tester les signaux se trouvant à l'interface entre le LTP (essentiellement LTP-INTER) et le LTS.





## **Conclusion**



La contribution de cette thèse au test de circuits complexes porte essentiellement sur une large récapitulation des méthodes de scan partiel employées et sur la proposition d'une généralisation de la méthode de scan qui permet de mieux tirer parti des fonctionnalités du circuit lors du test.

Dans le premier chapitre, on a comparé les différents principes algorithmiques employés pour la sélection des bascules d'un scan partiel. En général la performance de l'algorithme restait liée à l'ATPG séquentiel utilisé en final. De cette manière, un taux de couverture maximal ne pouvait être complètement assuré. De plus la diminution de l'impact du scan sur la vitesse du circuit n'était pas aussi bien traitée et considérée que la diminution de l'impact sur la surface. Or paradoxalement, la vitesse dans la conception de nombreux circuits est un facteur plus restrictif que la surface employée.

Le deuxième chapitre traite d'une généralisation de la méthode de scan. Elle apporte des solutions nouvelles à la sélection des bascules mises en scan généralisé. Par rapport au scan classique, on peut éviter d'utiliser des multiplexeurs de scan en utilisant la fonctionnalité du circuit. Ainsi on diminue le coût en surface, mais surtout on réduit ou supprime l'impact sur les chemins critiques. Lorsqu'on applique cette méthode de scan généralisé comme du scan complet à coût plus faible en surface et en vitesse, on peut alors la comparer à une méthode de scan partiel fournissant un taux de couverture maximal et ne nécessitant pas l'utilisation d'un ATPG séquentiel pour la génération des vecteurs de test finaux. Si au contraire, on s'intéresse principalement aux coûts en vitesse ou en surface alors cette méthode présente l'avantage d'apporter des solutions adaptées au contrôle et à l'observation des points critiques en vitesse mais aussi à la réduction de la longueur des séquences de test pour des séquenceurs.

Cette technique de scan généralisé présente l'avantage de formaliser et de démontrer des idées et des propriétés qui sont parfois perçues de manière intuitive. Les concepts de transmission d'information inhérents aux chaînes de scan sont intégrés via la définition de propriétés spécifiques de manière originale aux modes locaux et globaux introduits. Cela a permis de généraliser les notions de scan non seulement en considérant des propriétés liées à des fonctions bijectives, mais encore en considérant des propriétés liées à des bascules de rebouclage et/ou de transmission. De plus les notions d'indépendance temporelle introduites apportent des améliorations potentielles de la méthode que ce soit pour diminuer l'impact sur le chemin critique pour la mise en scan des bascules associées à des points d'activation à  $\varphi$ , ou que ce soit pour traiter les problèmes d'observation et de contrôle dans le cas de la présence de bascules de rebouclage seul.

Les résultats de l'application au LTP au chapitre 3 ont montré l'utilité de cette méthode dans le cas des circuits critiques en vitesse.

Cependant, la partie de la méthode concernant la mise en œuvre du contrôle et de l'observation des bascules de transmission n'a pas été appliquée telle qu'on l'a proposée. En effet, la mise en œuvre aurait nécessité des développements de logiciels qui auraient mobilisé des moyens importants. On pourrait d'ailleurs s'inspirer des propositions faites au chapitre 2 (paragraphe V) pour concevoir ces logiciels. D'autres améliorations sont envisageables. Tout d'abord, il y a celles liées à la conception d'un circuit :

- Actuellement, le concepteur apporte ses connaissances du circuit pour décrire les modes locaux et globaux, mais on pourrait envisager des outils d'aide à la génération de ces modes. Ces outils interviendraient au niveau RTL (que le concepteur appréhende plus facilement) ou encore à des niveaux de description inférieurs.
- On pourrait développer un outil complet qui intégrerait à la fois la génération des vecteurs de test grâce à un ATPG combinatoire, et la mise en œuvre du contrôle et de l'observation.
- On pourrait étudier comment aider les ATPGs séquentiels existants à utiliser les différentes propriétés associables aux modes locaux et globaux.
- La proposition suivante porte sur l'approfondissement des études sur l'utilisation hiérarchique de cette méthode afin de pouvoir exploiter pleinement les capacités de test d'éléments de bibliothèques de blocs fonctionnels testables grâce au scan généralisé.
- Les structures de test du scan généralisé pourraient être utilisées dans le cadre du débogage des circuits après leur fabrication.

D'autres extensions plus théoriques peuvent aussi être considérées :

- On pourrait faire des extensions autorisant l'emploi de certains cycles à l'intérieur du graphe des modes locaux.
- De même, la méthode pourrait être étendue à l'emploi de plusieurs modes globaux.
- Une autre extension possible serait de poursuivre des recherches sur la resynthèse de machines d'état finis de façon à réduire la longueur des séquences de test pour les contrôler et les observer.
- On pourrait rechercher des rapprochements avec les méthodes de test symbolique et/ou hiérarchisé.
- Enfin, pourquoi ne pas développer des ATPGs séquentiels qui puissent prendre en compte les propriétés associées aux modes locaux et globaux.

## **Références bibliographiques**



- [ABAD85] M. S. Abadir, M. A. Breuer  
“A knowledge-based System for Designing Testable VLSI Chips”, IEEE Design and Test, pp. 56-68, 1985
- [ABRA91] M. Abramovici, J. J. Kulikowski, R. K. Roy  
“The Best Flip-flops to Scan”, International Test Conference, pp. 166-173, 1991
- [ABRA92] M. Abramovici, P. S. Parikh  
“Warning: 100% Fault Coverage May Be Misleading!!”, International Test Conference, pp. 662-668, 1992
- [ABRA93] M. Abramovici  
“DOs and DON'Ts in Computing Fault Coverage”, International Test Conference, p. 594, 1993
- [AGRA82] V. D. Agrawal, M. Ray Mercer  
“Testability Measures -- What Do They Tell Us?”, International Test Conference, pp. 391-396, 1982
- [AGRA84] V. D. Agrawal, S. K. Jain, D. M. Singer  
“Automation in Design for Testability”, Custom Integrated Circuits Conference, Rochester NY, May 21-23, 1984
- [AGRA87] V. D. Agrawal, K.-T. Cheng, D. D. Johnson, T. Lin  
“A Complete Solution to the Partial Scan Problem”, International Test Conference, pp. 44-51, 1987
- [AGRA88] V. D. Agrawal, K.-T. Cheng, P. Agrawal  
“CONTEST: a Concurrent Test Generator for Sequential Circuits”, Design Automation Conference, pp. 84-89, 1988
- [ALBE90] G. Albertengo, R. Sisto



- “Parallel CRC Generation”, IEEE MICRO, pp. 63-71, October, 1990
- [ASHA94] P. Ashar, S. Malik  
 “Implicit Computation of Minimum-Cost Feedback-Vertex Sets for Partial Scan and Other Applications”, Design Automation Conference, pp. 77-80, 1994
- [BEEN89] F. P. M. Beenker, R. Dekker, R. Stans  
 “A testability Strategy for Silicon Compilers“, International Test Conference, pp. 660-669, 1989
- [BENK90] J. Benkosky, E. Vanden Meersch, L. J. M. Claesen, H. De Man  
 “Timing Verification Using Statically Sensitizable Paths”, IEEE Transactions on CAD, Vol. 9, No. 10, pp. 1073-1084, October, 1990
- [BENK91] J. Benkosky, R. Stewart  
 “TATOO: An Industrial Timing Analyzer with False Path Elimination and Test Pattern Generation”, European Conference on Design Automation, pp. 256-260, February, 1991
- [BENN93] R. G. Bennetts, F. P. M. Beenker  
 “Partial Scan: what Problem does it Solve?“, European Test Conference, pp. 99-106, 1993
- [BHAW91] S. Bhawmik, C. J. Lin, K. T. Cheng, V. D. Agrawal  
 “PASCANT: a Partial Scan and Test Generation System”, Custom Integrated Circuits Conference, paper 17.3, 1991
- [BLAT93] J. Blatny, Z. Kotásek, J. Hlavicka  
 “I-Path Analysis”, European Test Conference, pp. 255-252, 1993
- [BOUR71] W. Bouricius et al.  
 “Algorithms for Detection of Faults in Logic Circuits”, IEEE Transactions

on Computers, Vol. C-20, No. 11, pp 1258-1264, November, 1971

[BREU76] M. Breuer, A. Friedman

“Diagnosis and Reliable Design of Digital Systems”, Computer Science Press, Rockville, MD., 1976

[BRGL89] F. Brglez, D. Bryan, K. Kozminski

“Combinational Profiles of Sequential Benchmark Circuits”, International Symposium on Circuits and Systems, pp. 1929-1934, 1989

[BULO92a] J. Bulone, M. Diaz Nava

“A CMOS ASIC to Implement the TC Sublayer in the Physical Layer of the ATM Network”, Euro ASIC proceedings, June, 1992

[BULO92b] J. Bulone, M. Diaz Nava, M. Saugé

“Report on the Design of the R1022 Line Terminator”, RACE Project R1022 Technology for ATD, deliverable 22/SWP4/423/DS/P/420/B1, 1992

[BULO93] J. Bulone, A. Chion, M. Diaz Nava

“Test of a 622 Mhz Hybrid Module For ATM Network Using a 200 MHz Validation Station”, Custom Integrated Circuits Conference, paper 26.1, 1993

[CCIT91a] CCITT/COMXVIII/R 34-E

“Draft Recommendation I.432 B-ISDN User Network Interface - Physical Layer Specification”, Melbourne, 1991

[CCIT91b] CCITT SGXVIII

“Temporary Document 39 (XVIII/8)”, Geneva, Switzerland, June, 1991

[CHAK94a] S. T. Chakradhar, A. Balakrishnan, V. D. Agrawal

“An Exact Algorithm For Selecting Partial Scan Flip-flops”, Design Au-

tomation Conference, pp. 81-86, 1994

[CHAK94b] S. T. Chackradhar, S. Dey

“Resynthesis and Retiming for Optimum Partial Scan”, Design Automation Conference, pp. 87-93, 1994

[CHEN89] C. H. Chen, P. R. Menon

“An Approach to Functional Level Testability Analysis”, International Test Conference, pp. 373-380, 1989

[CHENG89] K.-T. Cheng, V. D. Agrawal

“An Economical Scan Design for Sequential Logic Test Generation”, International Fault Tolerant Computing Symposium, pp. 28-35, 1989

[CHENG90] Kwang-Ting Cheng, Vishwani D. Agrawal

“A Partial Scan Method for Sequential Circuits Feedback”, IEEE Transactions on Computers, Vol. 39, No. 4, pp. 544-548, April, 1990

[CHIC90] V. Chickermane, J. H. Patel

“An Optimization Based Approach to the Partial Scan Design Problem”, International Test Conference, pp. 377-386, 1990

[CLUS84] E. J. McCluskey

“A Survey of Design for Testability Scan Techniques”, VLSI Systems Designs, Vol. 12, pp. 38-61, December 1984

[CLUS93] E. J. McCluskey

“Quality and Single-Stuck Faults”, International Test Conference, p. 597, 1993

[DIAZ86] M. Diaz Nava

“Proposition d’une méthodologie de conception de circuits intégrés de communication Réalisation d’un communicateur pour le réseau local

- FIP", Thèse, INPG (Grenoble), Juillet 1986
- [DIAZ91] M. Diaz Nava, M. Saugé  
 "Specifications of the Line Terminator component", Version 6, SGS-THOMSON MICROELECTRONICS, November 20, 1991
- [DIAZ93] M. Diaz Nava, J. Bulone, D. Belot, L. Dugoujon  
 "A 622Mb/s Line Terminator for ATM Networks", IEEE ISSCC Digest of Technical Papers, 1993, and also partially republished in Nikkei Electronics and Electronic Design.
- [DIAZ94] M. Diaz Nava, D. Belot  
 "Specifications of the ATM Line Terminator monochip", Version 1, SGS-THOMSON MICROELECTRONICS, May, 1994
- [DIAZ95] M. Diaz Nava, D. Belot, L. Dugoujon, J. Bulone  
 "A 622/155 Mbit/s ATM line terminator monochip", to be published in the proceedings of the European Design and Test Conference, March, 1995
- [DICE88] G. Dicenet  
 "Le RNIS Techniques et atouts", éditions Masson, collection technique et scientifique des télécommunications, 1988
- [EICH77] E. B. Eichelberger, T. W. Williams  
 "A Logic Design Structure for LSI Testability", Design Automation Conference, pp. 462-468, 1977
- [FAUR89] P. Faurre  
 "Analyse, optimisation et filtrage numériques", Cours de l'Ecole Polytechnique, département de mathématiques appliquées, Tome I, p. 27, 1988-1989
- [FREE88] S. Freeman

“Test Generation for Data-Path Logic: the F-Path Method“, IEEE Journal of Solid-State Circuits, Vol. 23, No. 2, pp. 421-427, April 1988

[GOEL81] P. Goel

“An implicit Enumeration Algorithm to Generate Tests for Combinational Logic Circuits“, IEEE Transactions on Computers, Vol. C-30, No. 3 , pp 215-222, March 1981

[GOLD79] L. H. Goldstein

“Controllability/Observability Analysis of Digital Circuits“, IEEE Transactions on Circuits and Systems, Vol. CAS-26, No. 5, pp. 685-693, September 1979.

[GOLD80] L. H. Goldstein, E. L. Thigpen

“SCOAP: Sandia Controllability/observability Analysis Program“, Design Automation Conference, pp. 190-196, June 1980

[GUPS88] Rajesh Gupta, Rajiv Gupta, M. A. Breuer

“A Methodology for Partial Scan Design Using Balanced Sequential Structures“, Technical Report CRI-88-59, University of Southern California, Department of Electrical Engineering-Systems, December 1988

[GUPS89] Rajesh Gupta, Rajiv Gupta, M. A. Breuer

“BALLAST: a Methodology for Partial Scan Design“, International Fault Tolerant Computing Symposium, pp. 118-125, 1989

[GUPS90] Rajesh Gupta, Rajiv Gupta, M. A. Breuer

“The BALLAST Methodology for Structured Partial Scan Design“, IEEE Transactions on Computers, pp. 538-544, April, 1990

[HENN64] F. Hennie

“Fault Detecting Experiments for Sequential Circuits“, 5th Annual Sym-

posium on Switching Circuit Theory and Logical Design, Princeton, pp. 95-110, November, 1964

[HUDL89] Raghu V. Hudli, Sharad C. Seth

“Testability Analysis of Synchronous Sequential Circuits Based on Structural Data”, International Test Conference, pp. 364-372, 1989

[JAIN84] S. K. Jain, V. D. Agrawal

“STAFAN: an Alternative to Fault Simulation“, Design Automation Conference, pp. 18-24, 1984

[KARA91a] M. Karam, R. Leveugle, G. Saucier

“Hierarchical Test Generation Based on Delayed Propagation”, International Test Conference, pp. 739-747, 1991

[KARA91b] M. Karam

“Génération de test de circuits intégrés fondée sur des modèles fonctionnels”, Thèse, INPG (Grenoble), Octobre 1991

[KIM90] K. S. Kim, C. R. Kime

“Partial Scan by Use of Empirical Testability”, International Conference on CAD, pp. 314-317, 1990

[KIM93] K. S. Kim, C. R. Kime

“Partial Scan Using Reverse Direction Empirical Testability”, International Test Conference, pp. 498-506, 1993

[LEED90] D. H. Lee, S. M. Reddy

“On determining Scan Flip-flops in Partial-scan Designs”, International Conference on CAD, pp. 322-325, 1990

[LEER77] R. Lee

“Cyclic Code Redundancy”, Digital Design, Vol. 11, No 7, pp. 77-85,

July, 1977

- [LEVE90] R. Leveugle  
 “Analyse de signature et test en ligne intégré sur silicium”, Thèse, INPG (Grenoble), Janvier 1990
- [LEVI89] M.E. Levitt, J.A. Abraham  
 “The Economics of Scan Design”, International Test Conference, pp. 869-874, 1989
- [LOO90] G.-J. van Loo  
 “Micro-processor Interface for the ATM components”, Version 3.0, RNL\_311&321\_41\_CD\_CC, May 29, 1990
- [MA87] H.-K. T. Ma, S. Devadas, A. R. Newton, A. Sangiovanni-Vincentelli  
 “Test Generation for Sequential Finite State Machines”, International Conference on CAD, pp. 288-291, 1987
- [MA88] H.-K. T. Ma, S. Devadas, A. R. Newton, A. Sangiovanni-Vincentelli  
 “An Incomplete Scan Design Approach to Test Generation For Sequential Machines”, International Test Conference, pp. 730-734, 1988
- [MAXW93] P. C. Maxwell  
 “Let’s Grade ALL the Faults”, International Test Conference, p. 595, 1993
- [MICZ83] A. Miczo  
 “The Sequential ATPG: a Theoretical Limit”, International Test Conference, pp. 143-147, 1983
- [MICZ86] A. Miczo  
 “Digital Logic Testing and Simulation”, Harper & Row, New York, 1986
- [MIN92] H. B. Min, W. A. Rogers  
 “A Test Methodology for Finite State Machines Using Partial Scan De-

sign", *Journal of Electronic Testing: Theory and Application*, pp. 127-137, May, 1992

[MUEH82] E. I. Muehldorf, T. W. Williams

"Analysis of the Switching Behavior of Combinatorial Logic Networks", *International Test Conference*, pp. 379-390, 1982

[MURR91] B. T. Murray, John P. Hayes

"Test Propagation Through Modules and Circuits", *International Test Conference*, pp. 748-757, 1991

[NARA92a] S. Narayanan, C. Njinda, M. Breuer

"Optimal Sequencing of Scan Registers", *International Test Conference*, pp. 293-302, 1992

[NARA92b] S. Narayanan, C. A. Njinda, R. Gupta, M. A. Breuer

"SIESTA: a Multifacet Scan Design System", *European Conference on Design Automation*, pp. 246-251, 1992

[PARI93] P. Parikh, M. Abramovici

"A Cost-Based Approach to Partial Scan", *Design Automation Conference*, pp. 255-259, 1993

[PARK92] S. Park, S. B. Akers

"A Graph Theoretic Approach to Partial Scan Design by K-cycle Elimination", *International Test Conference*, pp. 303-311, 1992

[PERE83] A. Perez

"Byte-wise CRC Calculations", *IEEE MICRO*, Vol. 3, No 3, pp. 40-50, June, 1983

[PETE76] W. W. Peterson, E.-J. Weldon

"Error Correcting Codes", MIT Press, 1991



- [PRAD91] D. K. Pradhan, S. Nori, J. Swaminathan  
“A Methodology for Partial Scan Design”, European Test Conference, pp. 263-271, 1991
- [PRYC91] M. de Prycker  
“Asynchronous Transfer Mode solution for broadband ISDN”, Ellis Horwood series in Computer Communications and Networking, 1991
- [RAMA88] T. V. Ramabadran, S. S. Gaitonde  
“A Tutorial on CRC Computations”, IEEE MICRO, Vol. 8, No 4, pp. 62-75, August 1988
- [ROIC91] E. Roick, R. Buschke  
“Standard ATM interface” Version 5.0 frozen, VAL\_413\_020\_FS\_CC, June 6, 1991
- [SAMA89] A. Samad, M. Bell  
“Automating ASIC Design-for-testability - the VLSI Test Assistant”, International Test Conference, pp. 819-828, 1989
- [SCHO92] D. De Schoenmacker, P. Verbeeck  
“RACE 1022 ATM Technology Testbed (RATT) :a glimpse of a broadband future”, Electronics and Communicatin Engineering Journal, August, 1992
- [SEIS91] B. H. Seiss, P. M. Trouborst, M. H. Schulz  
“Test Point Insertion for Scan-based BIST”, European Test Conference, pp. 253-262, 1991
- [SODE93] J. M. Soden, C. F. Hawkins  
“Quality Testing Requires Quality Thinking”, International Test Conference, p. 596, 1993

- [STEE93a] J. Steensma, F. Catthoor, H. De Man  
“Modelling Delay in Symbolic Test for Data Paths with Partial Scan”, European Test Conference, pp. 543-544, 1993
- [STEE93b] J. Steensma, F. Catthoor, H. De Man  
“Partial Scan at the Register-Transfer Level”, International Test Conference, pp. 488-497, 1993
- [TGE90] TGE- RACE 1022 project  
“The R1022 ATD Technology Testbed. A description”, September, 1990
- [THEA89] K. Thearling, J. Abraham  
“An Easily Computed Functional Level Testability Measure”, International Test Conference, pp. 381-390, 1989
- [TRIS80] E. Trischler  
“Incomplete Scan Path With an Automatic Test Generation Methodology”, International Test Conference, pp 153-162, 1980
- [TURI93] J. Turino  
“DFT: Profit or Loss -- a Position Paper”, International Test Conference, p. 269, 1993
- [VARM93] Prab Varma  
“Scan DFT: Why More Can Cost Less”, International Test Conference, p. 267, 1993
- [WUND89] H.-J. Wunderlich  
“The Design of Random-testable Sequential Circuits”, International Fault Tolerant Computing Symposium, pp. 110-117, 1989
- [XP100] ZYCAD  
“XPLUS Reference Manuel” Rev. 4.5a, February 27, 1991



# **Annexe A : Éléments de la théorie des graphes**



### • Graphe, arcs et sommets

Un *graphe orienté* est un quadruplet  $G = (X, U, o, e)$  où  $X$  et  $U$  sont deux ensembles (dis-joints) et  $o$  et  $e$  deux applications de  $U$  dans  $X$

$X$  est l'ensemble des *sommets*

$U$  est l'ensemble des *arcs*

$o(u)$  est le sommet *origine* de l'arc  $u$

$e(u)$  est son sommet *extrémité*

On peut prendre la notation allégée :  $G = (X, U)$ .

Les graphes utilisés dans ce travail sont toujours des graphes orientés.

Des arcs ayant même origine et même extrémité sont dits *parallèles*. On dit aussi qu'ils forment un *arc multiple* ; leur nombre est la *multiplicité* de cet arc multiple.

Un graphe  $G$  est un *multigraphe* s'il contient des arcs multiples. C'est un *p-graphe* si les multiplicités de ses arcs sont inférieures ou égales à  $p$ .

Le *demi degré extérieur* d'un sommet  $x$  est le nombre d'arcs qui ont  $x$  pour origine.

Le *demi degré intérieur* d'un sommet  $x$  est le nombre d'arcs qui ont  $x$  pour extrémité.

Le *degré* d'un sommet  $x$  est le nombre d'arcs qui ont  $x$  pour origine ou pour extrémité (c'est-à-dire *incidents* à  $x$ ).

Un sommet est dit *pendant* si son degré vaut 1 ; *isolé* si son degré vaut 0.

On parlera de *source* pour tout sommet dont le demi degré intérieur vaut 1.

On parlera de *puits* pour tout sommet dont le demi degré extérieur vaut 1.

### • Graphes dérivés

Soit  $G = (X, U, o, e)$ , un graphe orienté. On appelle *sous-graphe (partiel)* de  $G$  tout quadruplet  $G' = (Y, V, o', e')$  où  $Y \subset X$ ,  $V \subset U$  et où  $o'$  et  $e'$  sont les restrictions à  $V$  de  $o$  et  $e$  respectivement, avec la condition :

Pour tout  $u \in V$ , on a  $o'(u) (= o(u)) \in Y$  et  $e'(u) (= e(u)) \in Y$

On dira aussi que le sous-graphe partiel  $G' = (Y, V, o', e')$  est le *sous-graphe de  $G$  engendré par  $Y$*  si  $V$  est l'ensemble des arcs de  $G$  qui ont leurs deux extrémités dans  $Y$ .

Le *graphe adjoint* (ou *graphe aux arêtes*)  $G^*$  du graphe orienté  $G$  est le graphe ayant pour sommets les arcs de  $G$ , et pour arcs les couples  $(u, v)$  où  $u$  et  $v$  sont deux arcs de  $G$  tels que  $e(u) = o(v)$ .

### • Chemins et cycles

On appelle *chemin* toute séquence "alternée"

$$\gamma = (x_1, e_1, x_2, e_2, \dots, x_i, e_i, x_{i+1}, \dots, e_p, x_{p+1})$$

telle que

Pour tout  $i$ ,  $x_i$  est un sommet de  $G$ .

Pour tout  $i$ ,  $e_i$  est une arête de  $G$ .

Pour tout  $i$ ,  $e_i$  est d'origine  $x_i$  et d'extrémité  $x_{i+1}$ .

$p$  est sa *longueur* (en nombre d'arêtes).

Le chemin est *élémentaire* si tous les  $x_i$  sont distincts.

Le chemin est *simple* si tous les  $e_i$  sont distincts.

Le chemin est dit *cyclique* si  $x_1 = x_{p+1}$ .

On parle de *circuit* pour tout chemin cyclique et simple. En fait on emploiera ici le terme *cycle* pour éviter de faire des confusions entre le circuit dans le sens de la théorie des graphes, et le circuit, l'objet physique.

La *profondeur séquentielle* d'un graphe est la plus grande longueur d'un chemin élémentaire du graphe.

On parlera de *graphe acyclique* pour un graphe sans cycles

### • Connexité

Définition de la relation  $\bar{C}$  :

Soient  $x$  et  $y$  deux sommets d'un graphe  $G=(X,U)$

$(x \bar{C} y)$  si et seulement s'il existe à la fois un chemin de  $x$  à  $y$  et un chemin de  $y$  à  $x$

De manière triviale,  $\bar{C}$  est une relation d'équivalence dans  $X$ .

On appelle *classe fortement connexe* toute classe d'équivalence de  $\bar{C}$ , et *composante fortement connexe* le sous-graphe engendré par une classe fortement connexe.

$G$  est dit *fortement connexe* s'il n'a qu'une seule composante fortement connexe (c'est-à-dire si pour tout couple  $(x,y)$  de sommets,  $G$  contient un chemin de  $x$  à  $y$ ).

On appelle *graphe réduit* de  $G$ , le graphe  $G/\bar{C}$  dont les sommets représentent les composantes fortement connexes de  $G$ , chaque arc de  $G$  reliant deux composantes distinctes étant reporté dans  $G/\bar{C}$  entre les deux sommets correspondants.

Ce graphe réduit permet de visualiser les relations entre composantes fortement conne-

es, c'est un graphe acyclique.

### • Numérotation

Une application  $f$  de  $X$  dans  $N$  est une *numérotation* de  $G$  si pour tout arc  $(x,y)$  de  $G$  on a :  $f(x) < f(y)$ . (A)

On parlera aussi de *numérotation croissante*, et dans le cas d'inégalité  $f(y) < f(x)$  on parlera de *numérotation décroissante*.

On a la proposition suivante :

$G$  admet une numérotation  $\Leftrightarrow G$  est acyclique.

Démonstration :

$\Rightarrow$  : Si  $c$  est un cycle, il est clair que toute application de  $X$  dans  $N$  enfreint la condition (A) sur au moins un arc de  $c$ .

$\Leftarrow$  : Réciproquement, si  $G$  est acyclique, soit  $X_0$  l'ensemble des sommets de  $G$  (ou sources) qui ne sont l'extrémité d'aucun arc ( $X_0$  ne peut être vide)

on posera  $f(x) = 0$  pour tout  $x \in X_0$ .

Considérons maintenant le sous-graphe  $G_1$  engendré par  $X - X_0$ , alors  $G_1$  est lui aussi acyclique. Certains de ses sommets n'ont pas d'antécédent.

Soit  $X_1$ , l'ensemble de ces sommets : on pose  $f(x) = 1$  pour tout  $x \in X_1$ , etc ...

Au bout d'un nombre fini d'itérations, on aura épuisé l'ensemble  $X$  ; il est clair que l'application  $f$  ainsi définie vérifie la condition (A).

Il est évident qu'une numérotation croissante fournit une numérotation décroissante et réciproquement.





**Annexe B : Fonctions de coûts tirées du  
dessin des masques**



Dans [CHIC90], à partir du dessin des masques, le coût de conversion d'une bascule en une bascule de scan (bascule qui intègre un multiplexeur pour une méthode de scan) est évalué comme la somme de trois coûts :

- Le coût de congestion d'un canal de routage, i.e. l'augmentation en hauteur d'une rangée de cellules précaractérisées. Il est noté  $w_c$ .
- Le coût en largeur, i.e. le dépassement en largeur de la rangée de cellules précaractérisées la plus large. Il est noté  $w_s$ .
- Le coût en longueur des fils, utilisé pour estimer le coût en vitesse. Il est noté  $w_d$ .

Ces coûts sont calculés à partir d'un placement et d'un routage donnés. La densité locale du canal en une bascule  $j$  est définie comme le nombre de fils que traverse une ligne de coupe à travers la région du canal située en  $j$ . La capacité du canal est le nombre maximum de fils qui peuvent être routés. Les points de congestion se trouvent où la densité locale du canal vaut la capacité du canal. La conversion d'une bascule en une bascule de scan augmente le nombre de fils à amener à cause de la donnée de scan et du signal de sélection du mode. Par conséquent on augmente aussi la congestion. Le

coût  $w_c$  peut être défini ainsi :  $w_c(j) = \frac{1}{\text{capacité\_du\_canal}(j) - \text{densité\_locale\_du\_canal}(j)}$ .

Une bascule de scan est plus large qu'une bascule normale. Le remplacement d'une bascule normale en une bascule de scan allonge la largeur de la rangée dans laquelle elle se trouvait. Le coût  $w_s$  peut être défini ainsi :

$w_s(j) = 0$  si  $\omega + e < \text{largeur\_maximum}$  et sinon  $w_s(j) = \frac{\omega + e - \text{largeur\_maximum}}{r}$ , où :  $\omega$  est la largeur de la rangée où se trouve  $j$ ,  $e$  est l'élargissement de la rangée si toutes les bascules sont mises en scan,  $r$  est le nombre de bascules dans la rangée, et  $\text{largeur\_maximum}$  est la largeur de la rangée initialement la plus large.

Enfin, pour estimer le coût de mise en scan d'une bascule qui se trouverait dans un chemin critique, on classe les fils en fonction de leur longueur, on détermine les 10% des fils les plus longs, et on attribue à la bascule  $j$  un coût constant  $w_d(j) = k$  si elle est connectée à l'un de ces fils, sinon on attribue un coût nul  $w_d(j) = 0$ .

Le coût total, est la somme des coûts :  $w(j) = w_c(j) + w_s(j) + w_d(j)$ .



## **Annexe C : Calcul des mesures de SCOAP**



Des mesures de contrôlabilité et d'observabilité appelées SCOAP (Sandia Controllability and Observability Program) destinées aux circuits séquentiels ou combinatoires ont été décrites par Goldstein dans [GOLD79]. SCOAP définit six fonctions, divisées en deux classes : combinatoires et séquentielles. Dans chaque classe, trois entiers positifs sont affectés à chaque nœud. Ils représentent la 1-contrôlabilité, la 0-contrôlabilité, et l'observabilité de ce nœud. Pour un nœud  $N$  du circuit, elles sont notées respectivement, d'une part :  $SC0(N)$ ,  $SC1(N)$ , et  $SO(N)$  dans le cas des mesures séquentielles et d'autre part :  $CC0(N)$ ,  $CC1(N)$ , et  $CO(N)$  dans le cas des mesures combinatoires.

$SC0(N)$  et  $SC1(N)$  évaluent respectivement le nombre minimum de coups d'horloges qu'il faut donner pour justifier la valeur 0 et 1 en  $N$ . Et  $SO(N)$  évalue le nombre minimum de coups d'horloge qu'il faut donner pour propager la valeur qui est en  $N$ .

$CC0(N)$  et  $CC1(N)$  évaluent respectivement le nombre minimum de portes logiques qu'il faut traverser pour justifier la valeur 0 et 1 en  $N$ . Et  $CO(N)$  évalue le nombre minimum de portes logiques qu'il faut utiliser pour propager la valeur qui est en  $N$ .

Pour calculer ces valeurs, on commence par séparer les éléments de la bibliothèque de cellules précaractérisées en éléments combinatoires et éléments séquentiels. Les combinatoires sont ceux dont les valeurs des sorties ne dépendent que des valeurs des entrées au même instant. Les séquentiels sont ceux dont les valeurs des sorties dépendent des valeurs des entrées à des instants précédents. Les éléments séquentiels ont donc un comportement mémorisant. On définit un nœud séquentiel comme la sortie d'un élément séquentiel et un nœud combinatoire comme la sortie d'un élément combinatoire. Pour chaque cellule, la contrôlabilité séquentielle à la valeur 0 ou 1 d'un de ses nœuds de sortie est calculée en examinant toutes les affectations des entrées possibles qui justifient cette valeur en sortie. Pour chaque affectation solution, la somme des contrôlabilités séquentielles des entrées correspondantes est calculée. La valeur minimale obtenue plus la profondeur séquentielle de la cellule donne alors sa contrôlabilité séquentielle. Sachant que par définition, la profondeur séquentielle d'une cellule séquentielle vaut 1, et celle d'une cellule combinatoire vaut 0.

Pour chaque cellule, l'observabilité séquentielle d'une de ses entrées est calculée en examinant toutes les affectations des autres entrées possibles qui propagent cette valeur ou son complément en l'une des sorties. Pour chaque affectation solution, on somme les contrôlabilités séquentielles des autres entrées et l'observabilité séquentielle la plus faible des sorties associées. La valeur minimale obtenue plus la profondeur séquentielle de la cellule donne alors l'observabilité séquentielle.

Pour les mesures combinatoires les équations sont les mêmes à part que lorsque la cel-



lule considérée est combinatoire on ajoute la valeur 1, et que lorsque la cellule est séquentielle au lieu d'ajouter 1, on n'ajoute rien.

L'algorithme de calcul utilisant ces équations locales est alors:

- Pour les entrées primaires I, on pose :  $SC0(I) = SC1(I) = 0$  et  $CC0(I) = CC1(I) = 1$ .
- Assigner aux autres nœuds N :  $SC0(N) = SC1(N) = CC0(N) = CC1(N) = \infty$ .
- Propager les valeurs affectées aux entrées des cellules aux sorties des cellules à partir des équations associées à chaque cellule.
- Répéter jusqu'à convergence.
- Pour les sorties primaires U, on pose :  $SO(U) = CO(U) = 0$ .
- Assigner aux autres nœuds N :  $SO(N) = CO(N) = \infty$ .
- Propager les valeurs affectées aux sorties des cellules aux entrées des cellules à partir des équations associées à chaque cellule et des contrôlabilités calculées auparavant.
- Répéter jusqu'à convergence.

Ces valeurs séquentiels mesurent la difficulté à générer des vecteurs de test en estimant le nombre de coups d'horloge nécessaires pour contrôler ou observer un nœud du circuit ; elles ne dépendent pas d'un ATPG particulier. Ces mesures ne tiennent pas compte des corrélations possibles entre les signaux du circuit, néanmoins elles restent intéressantes pour comparer la testabilité de deux circuits.

**Annexe D : Principes de l'algorithme "0/1  
knapsack"**



Voici la structure grossière de l'algorithme "0/1 knapsack" dont les paramètres sont explicités ci-après.

Le paramètre `Coût_maximum` est la limite supérieure du coût accepté pour la mise en scan du dessin, `n` est le nombre de bascules du circuit. Le paramètre  $w_k$  est le coût ou poids associé à la  $k$ ème bascule et  $p_k$  est le gain en testabilité quand seulement la  $k$ ème bascule est mise en scan. Au départ on supposera que l'on a ordonné les indices par ordre décroissant de densité, i.e.,  $p_1/w_1 \geq p_2/w_2 \geq \dots p_n/w_n$ . Les termes `coût_final` et `gain_maximum` s'expliquent d'eux-mêmes. La meilleure solution à l'instant courant est conservée dans le tableau `X[1] ... X[n]`. Si `X[k]` vaut 0, alors la bascule  $k$  n'est pas mise en scan, et si elle vaut 1, alors elle est mise en scan. Le tableau `Y` conserve les résultats intermédiaires. La fonction `calcul_du_gain` calcule l'amélioration du critère de testabilité en mettant de manière effective l'ensemble des bascules (passé comme paramètre d'entrée) dans la chaîne de scan et en appliquant les calculs de testabilité. La fonction `calcul_majorant` fournit un majorant de l'amélioration pour l'ensemble de bascules donné en argument.

Initialiser `coût_courant`, `gain_courant`, `gain_maximum`;

Boucler

```
{
  Tant_que ( $k \leq n$  et  $\text{coût\_courant} + w_k \leq \text{Coût\_maximum}$ ) Faire
```

```
  {
    Ajouter la bascule  $k$  à la chaîne de scan ;
     $Y[k] = 1$  ;  $\text{coût\_courant} = \text{coût\_courant} + w_k$  ;
    Incrémenter  $k$ ;
  }
```

Si ( $k > n$ ) Alors

```
{
  /* nous avons obtenu le meilleur choix jusque là */
  /* mettre à jour le gain, le coût et la sélection, et continuer à avancer*/
   $\text{gain\_courant} = \text{calcul\_du\_gain}(Y)$  ;
   $\text{gain\_maximum} = \text{gain\_courant}$  ;
```

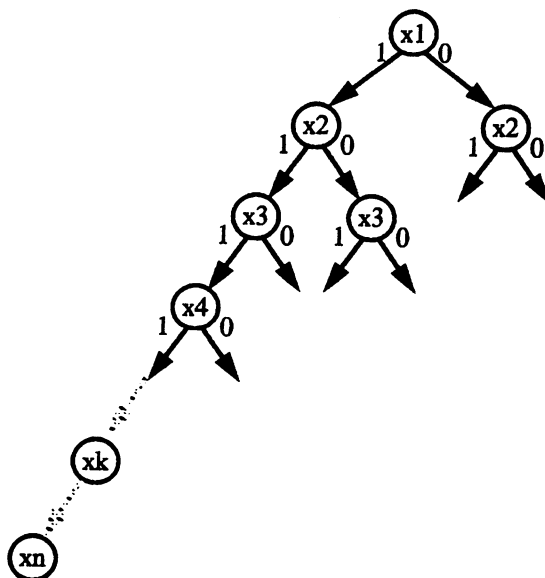
```

    coût_final = coût_courant ;
    k = n ;
    Transférer le tableau Y dans le tableau X;
}
Sinon
{
    /* la bascule k ne remplit pas les contraintes de coût */
    /* revenir sur la décision */
    Y[k] = 0;
}
Tant_que (calcul_majorant(coût_courant, k, Y, Coût_maximum) ≤ gain_maxi-
mum) Faire
{
    Sauvegarder l'arbre de recherche pour trouver la dernière bascule incluse ;
    Si (la racine de l'arbre de recherche est atteinte, i.e., k = 0) Alors
    {
        /* l'algorithme s'arrête ; imprimer la solution et quitter */
        Sortir(coût_final, gain_maximum, tableau X) ;
        quitter ;
    }
Sinon
    {
        /* défaire la décision d'inclusion de la bascule k */
        Y[k] = 0 ;
        coût_courant = coût_courant - wk ;
    }
}
incrémenter k;

```

}  
**Pour\_toujours;**

L'algorithme commence par un remplissage du "knapsack" de type glouton, à savoir dans l'ordre des densités de valeurs. Cela correspond à une série de mouvement dans le sens des fils gauches dans l'arbre de recherche (voir la figure 1 qui suit). Supposons que la bascule  $r$  soit la dernière à avoir été incluse, l'algorithme alors à ce point calcule le coût\_courant et le gain\_courant de la solution obtenue. Ensuite, il renverse la décision d'inclure la bascule  $r$  et essaie de remplacer la capacité inutilisée par d'autres bascules. Puisque le calcul du gain véritable coûte cher, le remplissage n'est fait qu'après le calcul d'un majorant. Si le majorant n'est pas meilleur que la meilleure solution trouvée jusqu'à présent, le nouveau chemin n'est pas exploité. L'algorithme piste en arrière pour trouver un mouvement non essayé, en supprimant la bascule  $r-1$ . Il continue le pistage en arrière jusqu'à ce qu'il trouve une bascule dont l'exclusion peut conduire à une meilleure solution.



**Figure 1 :** arbre de recherche



**Annexe E : Une heuristique de suppression  
de cycles d'un graphe**





Soient  $G$  un graphe et une fonction de coût additive qui vaut  $w_k$  pour chacun des sommets  $k$  du graphe. On cherche à minimiser le nombre de cycles du graphe par suppression de sommets sous la contrainte que le coût associé aux sommets supprimés soit inférieur à une valeur donnée,  $\text{Coût\_maximum}$ . Ce problème NP-complet peut être résolu par énumération exhaustive de tous les cycles passant par un sommet. Le calcul d'une solution optimale est acceptable de manière pratique que si on a un nombre restreint de sommets, par exemple 32. Dès lors, pour les graphes de grande taille, on procède par simplifications non optimales du problème en des problèmes plus simples jusqu'à devoir traiter seulement des graphes comportant au plus 32 sommets. Pour réduire le problème, on considère les composantes fortement connexes ou CFCs du graphe. Les CFCs sont des sous-graphes engendrés par les classes d'équivalence de la relation  $R$ , définie par :

( $a R b$ ) si et seulement s'il existe à la fois un chemin de  $a$  vers  $b$  et un chemin de  $b$  vers  $a$ .

La décomposition du graphe en ses CFCs peut se réaliser en un temps linéaire du nombre de sommets du graphe. Tout cycle du graphe initial va se retrouver dans l'une des CFCs. Si  $a$  est un arc orienté du sommet  $i$  vers le sommet  $j$  d'une CFC, on peut utiliser comme approximation d'un majorant du nombre de cycles passant par cet arc, le produit du nombre d'arcs d'extrémité  $i$  par le nombre d'arcs d'origine  $j$ . Soit formellement :

$$\text{cycle\_par\_arc\_majorant}(a) = \text{demi\_degré\_int}(i) \times \text{demi\_degré\_ext}(j).$$

De la même façon une estimation d'un majorant du nombre de cycles passant par un sommet  $i$  est donnée par :

$$\text{cycle\_par\_sommet\_majorant}(i) = \sum_{a \text{ incident au nœud } i} \text{cycle\_par\_arc\_majorant}(a)$$

Soit  $S$  un ensemble des sommets choisis pour être exclus du graphe, et  $\text{coût}(S)$ , la somme des coûts de chacun des sommets de  $S$ , alors on peut procéder comme suit :

**Etape 1 :** décomposer le graphe séquentiel du circuit en ses CFCs

**Etape 2 :** essayer de simplifier les CFCs par des règles évidentes de réduction. Une technique très utile est de combiner un sommet qui n'est que l'origine d'un seul arc, avec son successeur, et un sommet qui n'est que l'extrémité d'un seul arc, avec son prédécesseur.

**Etape 3 :** aller à l'étape 6 si toutes les CFCs comportent moins de 32 sommets.

**Etape 4 :** considérer les CFCs à plus de 32 sommets. Pour chacun de leur sommet  $k$  calculer la  $\text{densité\_de\_valeur} = \text{cycle\_par\_sommet\_majorant}(k) / w_k$ .

Insérer le sommet avec la plus grande densité\_de\_valeur dans S. La CFC à laquelle ce sommet appartient est par conséquent décomposée en CFCs plus petites.

**Etape 5 :** répéter de l'étape 2 à l'étape 4 jusqu'à ce qu'il ne reste aucune CFC ayant plus de 32 sommets ou que coût(S) dépasse Coût\_maximum. Dans ce dernier cas, S contient la liste des bascules mises en scan, donc le processus de sélection est terminé.

**Etape 6 :** toutes les CFCs ont moins de 32 sommets, donc tous les cycles peuvent être énumérés.

Soit le gain d'un ensemble S de sommets, où gain(S) est le nombre de cycles qui passent par au moins l'un des éléments de S. Soit  $p_k$  = le nombre de cycles auxquels appartient le sommet k. Alors, lorsque tous les cycles peuvent être énumérés, on peut utiliser l'algorithme de l'annexe D, page 201: "Principes de l'algorithme "0/1 knapsack"" pour résoudre le problème posé. Une fonction de majoration est donnée par la relation évidente :  $\text{gain}(S \cup \{k\}) \leq \text{gain}(S) + p_k$

**Annexe F : Algorithme de génération du  
nuage d'influence**



L'algorithme de recherche du nuage d'influence a pour paramètres d'origine :

- un seuil qui va déterminer la taille du nuage
- un graphe  $G$  dont les sommets représentent des bascules et dont les arcs orientés représentent l'adjacence entre deux bascules (bascules reliées par de la logique combinatoire)
- une pondération des arcs de  $G$ , qui représente l'observabilité séquentielle relative (RSO) de deux bascules adjacentes.

Pour une bascule  $FF_i$  donnée, on recherche l'ensemble des bascules de RSO pour  $FF_i$  inférieure au seuil. La RSO entre deux bascules non adjacentes est donnée par le minimum de la somme des poids sur l'ensemble des chemins les reliant. Si aucun chemin ne les relie leur RSO est infinie.

L'algorithme détermine d'abord deux ensembles de sommets, l'ensemble  $S$  obtenu en parcourant les arcs dans le sens prédécesseur successeur, et l'ensemble  $P$  obtenu en parcourant les arcs dans le sens successeur prédécesseur. Le nuage d'influence est alors l'union de  $S$  et  $P$ .

Voici la description de la procédure de calcul de  $S$  :

calcul\_de\_S ( $FF_i$ )

{

Initialiser  $S$  à  $S = \{FF_i\}$  ;

Mettre tous les successeurs de  $FF_i$  (différents de  $FF_i$ ) dans l'ensemble frontière des successeurs :  $FS$  ;

Tant que (toutes les bascules de  $FS$  ne vérifient pas  $RSO > \text{seuil}$ ) faire

{

déterminer la bascule  $FF_j$  de RSO la plus faible dans  $FS$  ;

supprimer  $FF_j$  de  $FS$  ;

ajouter  $FF_j$  à  $S$  ;

pour l'ensemble des successeurs  $FF_k$  de  $FF_j$  qui ne sont pas dans  $S$  faire

{

si  $FF_k$  est déjà dans  $FS$

```

{
  /* on a une reconvergence */
  réévaluer RSO de FFk par :
  min(ancien RSO de FFk, (RSO de FFj + RSO entre FFk et FFj)) ;
}
sinon
{
  ajouter FFk à FS ;
  évaluer RSO de FFk par :
  RSO de FFj + RSO entre FFk et FFj ;
}
}
}

```

De la même façon, on calcule P en prenant les prédécesseurs au lieu des successeurs et FP la frontière des prédécesseurs au lieu de FS la frontière des successeurs. Le nuage d'influence résultat est  $S \cup P$ . L'algorithme tient compte des phénomènes de reconvergence et des cycles qui peuvent exister en réévaluant les valeurs de RSO.

## **Annexe G : Calcul des non détectabilités**





Lorsque des points d'observation et de contrôle sont supprimés dans un circuit supposé combinatoire, certaines fautes deviennent non détectables. On cherche alors à déterminer le nombre de fautes qui deviennent non détectables. Ce calcul s'effectue par propagation des non contrôlabilités et des non observabilités.

Soient  $\bar{0}$  et  $\bar{1}$ , qui représentent respectivement la notation pour les états d'une ligne qui n'est pas contrôlable à 0 et celle pour les états d'une ligne qui n'est pas contrôlable à 1. La non contrôlabilité se propage à travers les portes logiques en suivant des règles simples directement liées à leurs fonctions combinatoires. En voici quelques exemples : la sortie d'une porte "non et" est  $\bar{0}$  si au moins une de ses entrées est  $\bar{1}$  ; par ailleurs cette sortie est  $\bar{1}$  si toutes ses entrées sont  $\bar{0}$ . La sortie d'un inverseur est  $\bar{0}$  si son entrée est  $\bar{1}$ , et elle est  $\bar{1}$  si son entrée est  $\bar{0}$ . Ainsi, après la suppression de la contrôlabilité de certaines entrées primaires, on détermine les nœuds qui deviennent non contrôlables par propagation avant sur les portes des non contrôlabilités. Lorsque des nœuds sont non contrôlables cela entraîne parfois le fait que d'autres nœuds deviennent non observables. Par exemple si une porte "ou" a deux entrées dont l'une est non contrôlable à 0, alors l'autre n'est plus observable. De même un nœud qui est entrée de plusieurs portes, qui ont toutes des sorties non observables, devient non observable. On réalise donc ensuite des propagations arrière pour déterminer les nœuds qui deviennent non observables. Les fautes qui deviennent non détectables, sont celles qui ne peuvent être observées car le nœud correspondant est devenu non observable, ou encore celles qui ne peuvent être activées car le nœud correspondant est devenu non contrôlable (collage à 1 pour un nœud  $\bar{0}$  et collage à 0 pour un nœud  $\bar{1}$ ).



**Annexe H : Principes de l'ATPG**  
**STALLION**



L'ATPG séquentiel, STALLION, considère le circuit comme une seule machine à états finis, et utilise alors un graphe partiel de transition d'états qu'il extrait à partir d'une énumération d'états [MA87]. Ce graphe, STG (state transition graph), contient autant d'états valides que possible. Les états sont dérivés de la description en portes du circuit. Partant de l'état de remise à zéro R (supposé résistant aux fautes), des transitions sont générées en utilisant une extension de l'algorithme implicite d'énumération de PODEM [GOEL81]. La recherche des états suivants se poursuit alors en donnant une préférence soit à une recherche en profondeur (beaucoup de coups d'horloge successifs), soit à une recherche en largeur (beaucoup d'états successeurs pour un même état). Le générateur utilise aussi une extension de PODEM pour réaliser la propagation et l'excitation d'une faute : FEP (fault excitation-and-propagation). Il réalise aussi des justifications arrières d'états par un algorithme d'énumération de transitions : SJ (state justification).

La structure algorithmique du générateur du test est la suivante :

**Etape 1** : trouver un (nouvel) état S0 et une (nouvelle) séquence T1 qui excitera et propagera l'effet de la faute considérée aux sorties primaires en un nombre prescrit de coups d'horloge, en utilisant l'algorithme FEP. Si aucune solution n'existe, sortir sans vecteur de test pour cette faute.

**Etape 2** : Rechercher un (nouveau) chemin de propagation (potentiellement une séquence d'initialisation) T0 de l'état de remise à zéro R à S0 dans le STG partiel. S'il est trouvé, aller à l'étape 5.

**Etape 3** : Si le STG contient tous les états valides, aller à l'étape 1.

**Etape 4** : Trouver un chemin T0 de l'état de remise à zéro R à l'état initial S0, utilisant l'algorithme SJ. Si aucune solution n'existe, aller à l'étape 1.

**Etape 5** : Faire une simulation de fautes de la séquence potentielle d'initialisation T0. Si elle détecte la faute, définir la séquence de test T2 comme étant T0 et aller à l'étape 7. Dans le cas contraire, si elle est une séquence d'initialisation valide, alors continuer, sinon aller à l'étape 1.

**Etape 6** : Concaténer la séquence d'entrée T0 qui représente le chemin de l'état de remise à zéro R à l'état initial S0 avec T1 pour former T2 qui est la séquence de test pour la faute testée.

**Etape 7** : Sortir avec la séquence de test T2.



**Annexe I : Algorithmes de sélection des  
lignes d'état**





Le problème de sélection des lignes d'états dans le cadre de la méthode de sélection de bascules à mettre en scan à l'aide d'énumération d'états se formalise de la façon suivante. Soient  $N$  éléments (chacun correspondant à une faute), à chacun de ces éléments est associé un sous ensemble de groupes de lignes. On suppose que pour chaque élément aucun des groupes de lignes n'est un sur ensemble ou un sous ensemble d'un autre groupe de lignes. Le but est d'identifier un groupe de lignes pour chaque élément tel que le nombre de lignes distinctes sélectionnées dans les groupes de lignes soit minimal. Les groupes de lignes peuvent contenir un ou plusieurs éléments (on a des groupes à un élément par exemple dans le cas de fautes dont on n'est pas arrivé à propager les effets aux sorties primaires, mais seulement aux états suivants).

Voici d'abord un premier algorithme de type glouton (greedy) très rapide et utilisant différents points de départ.

```
{
pour (i=1 ; i N ; i=i+1)
    {
    Elément = e[i] ;
    pour (j=1 ; j Element.Nombre_de_Choix ; j=j+1)
        {
        Solution = greedy (Elément.Choix[j],Elément) ;
        }
    }
sélectionner la meilleure Solution c'est-à-dire celle de cardinal minimal ;
}
```

```
greedy (groupe_de_lignes,élément)
{
Lignes = Lignes  $\cup$  groupe_de_lignes ;
pour (i=1; i N;i=i+1)
    {
    si (e[i] élément)
```

```

{
  pour (j=1 ; j e[i].Nombre_de_Choix ; j=j+1)
    {
      card = |Lignes  $\cup$  e[i].Choix[j]| ;
    }
  sélectionner l'indice k qui donne card minimal ;
  Lignes = Lignes  $\cup$  e[i].Choix[k] ;
}
}
retourner (Lignes) ;
}

```

Voici un deuxième algorithme qui donne des solutions plus optimales.

```

{
  trouver l'ensemble Lignes de toutes les lignes nécessaires ;
  tant que (il y a des choix à faire)
    {
      prendre  $N_e$  éléments  $e_1, e_2, \dots, e_{N_e}$  et les choix pour ces éléments :
       $c_1, c_2, \dots, c_{N_e}$  qui minimisent  $|Lignes \cup e_i[c_i]|$  pour  $i = 1, \dots, N_e$  ;
      Ajouter à Lignes les choix faits pour les  $e_i$  ;
    }
}

```

Cet algorithme est en  $O(N^{N_e})$ . Au plus  $N_e$  est grand au plus la solution peut potentiellement être meilleure. Le cas  $N_e = N$  est la recherche exhaustive et demande beaucoup plus de temps CPU. Des faibles valeurs peuvent permettre de s'approcher de la solution optimale.

**Annexe J : Estimation du taux de  
couverture pour les approches en fréquence  
et en distance**



On suppose qu'on dispose d'un tableau d'options qui donne pour chaque faute différents vecteurs de test possibles et les bascules à utiliser dans chaque cas considéré. Le taux de couverture est alors estimé sans faire de simulation de fautes par la formule suivante :

$$\text{estimation} = \frac{(F_{\text{func}} + f_{\text{comb}} + \alpha F_{\text{ff}} + \alpha N_{\text{ff}} f_{\text{mux}})}{F_{\text{total}} + \alpha N_{\text{ff}} f_{\text{mux}}}, \text{ avec}$$

$F_{\text{func}}$  = fautes couvertes par des vecteurs fonctionnels

$F_{\text{ff}}$  = fautes des bascules non détectées par des vecteurs fonctionnels

$F_{\text{comb}}$  = fautes combinatoires non détectées par des vecteurs fonctionnels

$F_{\text{total}} = F_{\text{ff}} + F_{\text{comb}} + F_{\text{func}}$  = nombre total de fautes dans le circuit sans scan

$f_{\text{comb}}$  = couverture avec le scan des fautes combinatoires,  $f_{\text{comb}} \leq F_{\text{comb}}$

$\alpha$  = fraction des bascules dans le circuit

$f_{\text{mux}}$  = nombre de fautes dans un multiplexeur de scan FF

$N_{\text{ff}}$  = nombre total de FFs dans le circuit

La formule d'estimation est toujours pessimiste, et devient très pessimiste si  $F_{\text{func}}$  est faible (typiquement en dessous de 70%). En effet, les vecteurs de test utilisés dans cette approche vont introduire du test aléatoire.



# **Annexe K : Inversion des opérateurs galoisiens**





Les notations utilisées dans cette annexe sont les suivantes : l'opérateur  $\oplus$  est l'opération "ou exclusif", et l'opérateur " $\cdot$ " est l'opération "et". L'objectif est de résoudre le système :

$$Y = A \cdot X' \oplus B \text{ avec } Y = \begin{bmatrix} y_1 \\ | \\ | \\ y_n \end{bmatrix}, A = \begin{bmatrix} a_{ij} \\ 1 \leq i \leq n \\ 1 \leq j \leq m \end{bmatrix}, X' = \begin{bmatrix} x_1 \\ | \\ | \\ x_m \end{bmatrix} \text{ et } B = \begin{bmatrix} b_1 \\ | \\ | \\ b_n \end{bmatrix}.$$

Cela revient à chercher à inverser l'opérateur galoisien donné par l'expression  $A \cdot X' \oplus B$ .

On suppose que  $m \geq n$  (sinon on ne pourrait avoir la propriété de surjectivité) et que les  $x_j$  pour  $j > n$  sont considérés comme des paramètres.

On réécrit ce système d'équations de la façon suivante :

$$A^{(1)} \cdot X = B^{(1)}$$

$$\text{avec } A^{(1)} = \begin{bmatrix} a_{ij} \\ 1 \leq i \leq n \\ 1 \leq j \leq n \end{bmatrix}, X = \begin{bmatrix} x_1 \\ | \\ | \\ x_n \end{bmatrix} \text{ et } B^{(1)} = Y \oplus B \oplus \begin{bmatrix} a_{ij} \\ 1 \leq i \leq n \\ n < j \leq m \end{bmatrix} \cdot \begin{bmatrix} x_{n+1} \\ | \\ | \\ x_m \end{bmatrix}.$$

On supposera dans un premier temps que  $a_{11}^{(1)} = 1$  (ce qui est identique à  $a_{11}^{(1)} \neq 0$ ).

En soustrayant de la  $i$ -ème équation ( $i = 2, \dots, n$ ), la première lorsque  $a_{11}^{(1)} = 1$ , on obtient le système équivalent

$$x_1 \oplus a_{12}^{(1)} \cdot x_2 \oplus \dots \oplus a_{1n}^{(1)} \cdot x_n = b_1^{(1)}$$

$$a_{22}^{(2)} \cdot x_2 \oplus \dots \oplus a_{2n}^{(2)} \cdot x_n = b_n^{(2)}$$

⋮

$$a_{n2}^{(2)} \cdot x_2 \oplus \dots \oplus a_{nn}^{(2)} \cdot x_n = b_n^{(2)}$$

Cette étape revient à prémultiplier  $A^{(1)}$  et  $B^{(1)}$  par la matrice triangulaire inférieure

à diagonale unité

$$M_1 = \begin{bmatrix} 1 & & 0 \\ a_{21}^{(1)} & & \cdot \\ \cdot & 1 & \cdot \\ \cdot & \vdots & \cdot \\ a_{n1}^{(1)} & 0 & 1 \end{bmatrix}$$

On pose  $A^{(2)} = M_1 \cdot A_1$  et  $B^{(2)} = M_1 \cdot B_1$ .

On recommence le même processus avec  $A^{(2)}$  pour laquelle on supposera aussi  $a_{22}^{(2)} = 1$ , puis  $A^{(3)}$  etc ... En  $n - 1$  étapes, on obtient ainsi une matrice triangulaire supérieure  $A^{(n)} = U$ .

Ecriture de l'étape générale  $p$ . On prémultiplie  $A^{(p)}$  et  $B^{(p)}$  par

$$M_p = \begin{bmatrix} 1 & & & 0 \\ \cdot & 1 & & \cdot \\ \cdot & a_{p+1,p}^{(p)} & 1 & \cdot \\ \cdot & \vdots & \vdots & \cdot \\ 0 & a_{n,p}^{(p)} & 0 & 1 \end{bmatrix}$$

Cela revient à effectuer

$$a_{i,j}^{(p+1)} = \begin{cases} a_{i,j}^{(p)} \oplus a_{i,p}^{(p)} \cdot a_{p,i}^{(p)} & i, j = p+1, \dots, n \\ a_{i,j}^{(p)} & \text{autrement} \end{cases}$$

$$b_i^{(p+1)} = \begin{cases} b_i^{(p)} \oplus a_{i,p}^{(p)} \cdot b_p^{(p)} & i = p+1, \dots, n \\ b_i^{(p)} & \text{autrement} \end{cases}$$

On pose alors

$$A^{(p+1)} = M_p \cdot A^{(p)}, \quad B^{(p+1)} = M_p \cdot B^{(p)}$$

Chaque étape nécessite de l'ordre de  $(n - p)^2$  opérations d'où pour la réduction à un système triangulaire supérieur  $A^{(n)} \cdot X = B^{(n)}$  de l'ordre de  $\sum_p (n - p)^2 = O\left(\frac{n^3}{3}\right)$  opérations.

La résolution du système devient alors immédiate par les *formules de remontée* (“remontée” car on commence par calculer  $x_n$ ) :

$$x_i = b_i^{(n)} \oplus a_{i,i+1}^{(n)} \cdot x_{i+1} \oplus \dots \oplus a_{i,n}^{(n)} \cdot x_n \quad i = n, n-1, \dots, 1.$$

Ces formules nécessitent  $\sum_1 (n-i) = O\left(\frac{n^2}{2}\right)$  opérations.

Globalement la résolution se réalise en  $O(n^3)$  opérations.

On a supposé que la condition  $a_{p,p}^{(p)} = 1$  était vérifiée au cours de la factorisation de Gauss. Si à l'étape  $p$ ,  $a_{p,p}^{(p)} = 0$ , on peut par permutation des lignes et/ou des colonnes de  $A^{(p)}$  mettre en première position un élément à 1, sous la seule hypothèse initiale que la matrice  $A$  soit inversible. Ces éléments non nuls sont aussi appelés pivots. Un échec lors de la recherche des pivots démontre la non inversibilité de la matrice  $A$ , c'est-à-dire la non bijectivité de l'opérateur galoisien représenté.



## **Liste des figures**



## Chapitre 1 :

Figure 1 : modèle combinatoire de la bascule de scan ..... 50

## Chapitre 2 :

Figure 2 : combinaison de deux chaînes de scan par un “ou exclusif” ..... 69

Figure 3 : rebouclage dans un chaîne de scan par un “ou exclusif” ..... 70

Figure 4 : exemple de structure d’un mode global ..... 71

Figure 5 : exemple d’une combinaison de deux chaînes de scan par une bascule de rebouclage seul ..... 72

Figure 6 : représentation graphique d’un mode local ..... 77

Figure 7 : exemple d’un mode local rebouclant ..... 78

Figure 8 : exemple d’un graphe de modes locaux ..... 84

Figure 9 : représentation graphique de l’exemple de mode global ..... 108

Figure 10 : exemple de problème de justification d’état ..... 123

Figure 11 : exemple de contrôle pour un registre d’accumulation ..... 124

Figure 12 : complément à 1 ..... 130

Figure 13 : multiplexeur avec rebouclage ..... 131

Figure 14 : additionneur simple ..... 131

Figure 15 : multiplicateur simple ..... 133

Figure 16 : registre d’accumulation rendu accessible ..... 136

Figure 17 : séquenceur avec compteur binaire en mode autonome ..... 137

Figure 18 : exemple de mode autonome de séquenceur après rebouclage par un “ou exclusif” ..... 138

## Chapitre 3 :

Figure 19 : architecture générale du LTH ..... 152



<b>Figure 20 : <u>format d'une cellule</u></b> .....	154
<b>Figure 21 : <u>cellule morte</u></b> .....	155
<b>Figure 22 : <u>microphotographie du LTP</u></b> .....	158
<b>Figure 23 : <u>schéma simplifié de la méthode de conception du LTP</u></b> .....	159
<b>Figure 24 : <u>opérateur de remise à zéro</u></b> .....	160
<b>Figure 25 : <u>opérateur d'addition modulo 2 contrôlé</u></b> .....	160
<b>Figure 26 : <u>test du LTS à l'intérieur du LTH</u></b> .....	166
<b>Figure 27 : <u>test du LTP à l'intérieur du LTH</u></b> .....	167



**AUTORISATION DE SOUTENANCE**

Vu les dispositions de l'arrêté du 30 Mars 1992 relatifs aux Etudes Doctorales

Vu les Rapports de présentation de :

Monsieur Habib MEHREZ

Monsieur Yves BERTRAND

Monsieur Joseph BULONE

est autorisé à présenter une thèse en soutenance en vue de l'obtention du diplôme de  
**Docteur de l'Institut National Polytechnique de Grenoble, spécialité**  
"MICROELECTRONIQUE".

Fait à Grenoble, le  
23 NOV. 1994

Pour le Directeur de l'INPG  
Centre de Recherche  
le Directeur de l'École Doctorale  
J.L. LACOMBE





## **Résumé**

Cette thèse propose une extension des méthodes classiques de chemins de "scan". On utilise des opérateurs combinatoires plus généraux à la place des multiplexeurs à une seule sortie. Ils peuvent comporter des entrées et des sorties multiples. Ils peuvent boucler sur eux-mêmes par l'intermédiaire d'une ou plusieurs bascules. Lorsqu'ils vérifient certaines propriétés de bijectivité et qu'ils forment une structure propageant de l'information, alors cette structure est aussi utile que les chaînes du "scan" complet et s'utilise de manière semblable. Elle permet aussi une approche hiérarchique du test des circuits. On montre comment tirer profit de cette méthode plus générale pour réduire l'impact de la méthode de "scan" complet sur les performances de circuits complexes implantant des fonctions mathématiques courantes ou des séquenceurs. Des résultats sont donnés pour le cas réel d'un circuit CMOS, très rapide, spécifique pour le réseau numérique large bande et pour lequel les contraintes en vitesse étaient primordiales.

## **Mots clés**

Chemin de scan généralisé, scan partiel, scan complet, circuits VLSI, génération de test, simulation symbolique, modèle fonctionnel, test de chemins de données, test de séquenceurs, conception de circuits très rapides.