

UNIVERSITÉ DE MONTRÉAL

UNE ENVELOPPE POUR LA NORME P1500 INCORPORANT DES
STRUCTURES DE TEST INTÉGRÉ POUR LES SYSTÈMES SUR
PUCES

ABDELAZIZ LARAB

DÉPARTEMENT DE GÉNIE ÉLECTRIQUE
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION DU
DIPLOME DE MAÎTRISE EN SCIENCES APPLIQUÉES
(GÉNIE ÉLECTRIQUE)

JUILLET 2006



Library and
Archives Canada

Bibliothèque et
Archives Canada

Published Heritage
Branch

Direction du
Patrimoine de l'édition

395 Wellington Street
Ottawa ON K1A 0N4
Canada

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

ISBN: 978-0-494-19310-5

Our file Notre référence

ISBN: 978-0-494-19310-5

NOTICE:

The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

AVIS:

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protègent cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.


Canada

UNIVERSITÉ DE MONTRÉAL
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire est intitulé :

UNE ENVELOPPE POUR LA NORME P1500 INCORPORANT DES
STRUCTURES DE TEST INTÉGRÉ POUR LES SYSTÈMES SUR
PUCES

Présenté par : LARAB Abdelaziz

En vue de l'obtention du diplôme : Maîtrise ès sciences appliquées

A été dûment accepté par le jury d'examen constitué de :

M. SAVARIA Yvon, Ph. D., président

M. BOIS Guy, Ph. D., membre

M. KHOUAS Abdelhakim, Ph. D., membre et directeur de recherche

DÉDICACE

À la mémoire de mon père Mohamed Saïd et mon beau-père Saleh.

À ma mère et ma belle-mère.

À ma femme et mes enfants.

À toute ma famille et ma belle-famille.

REMERCIEMENTS

J'aimerais remercier d'abord, Monsieur Abdelhakim Khouas, qui m'a accueilli au sein de son équipe de recherche au département de génie électrique de l'école Polytechnique de Montréal et qui m'a fait bénéficier de ses compétences scientifiques, ses qualités humaines et sa constante disponibilité. Je tiens particulièrement à lui exprimer ici ma profonde et amicale reconnaissance pour m'avoir soutenu financièrement. J'ai été très touché par la confiance qu'il m'a témoignée tout au long de mon travail.

Je tiens à exprimer mes plus sincères remerciements à Monsieur Yvon Savaria, Directeur du Groupe de Recherche en Microélectronique (GRM) de l'école Polytechnique de Montréal, qui me fait l'honneur de juger ce travail et d'en être le président du jury. J'adresse également mes remerciements à Monsieur Guy Bois, Professeur au département génie informatique et membre du GRM, pour avoir accepté de juger ce travail et d'en être membre du jury.

Je tiens à exprimer ma reconnaissance à ma femme et à mes enfants, pour leur patience inconditionnelle face à mon humeur ponctuée des hauts et bas durant ces années d'études. Je remercie également mes parents et tous mes proches qui m'ont encouragé tout au long de mes études.

Finalement, mes remerciements vont à Ghyslaine Ethier-Carrier pour son aide et à l'ensemble du personnel de soutien administratif et informatique du GRM qui n'ont ménagé aucun effort pour rendre agréable notre passage au sein du groupe.

RÉSUMÉ

Les systèmes sur puces « Systems On Chip » (SOC) deviennent de plus en plus complexes et leurs tests difficiles à réaliser. Ces dernières années, plusieurs architectures de test ont été proposées pour ces systèmes afin de leur garantir une bonne qualité de test avec un coût raisonnable. Mais, ces architectures engendrent une augmentation de la surface des SOC ainsi qu'une importante dégradation des délais internes de propagation.

L'objectif principal de ce travail est de réduire la surface additionnelle engendrée par l'insertion des architectures de test dans les SOC. C'est pourquoi, nous avons proposé une nouvelle architecture de test pour les modules des SOC et pour les interconnexions entre ces modules. Cette nouvelle architecture est une interface matérielle dans laquelle, les modules du SOC sont encapsulés. En mode normal, l'interface devient transparente pour permettre le fonctionnement normal des modules. En mode test, l'interface est configurable en registre à décalage et en structure du test intégré pour réaliser le test des modules et des interconnexions à partir de l'extérieur et à l'interne du SOC. Le registre à décalage sert à transférer les données de test à appliquer aux entrées des modules et aux interconnexions à partir d'équipements de test externes aux SOC. Ce registre est également utilisé pour l'acheminement des réponses de test vers les testeurs externes pour analyse. La structure du test intégré sert à générer les données de test à appliquer aux entrées des modules et aux interconnexions ainsi que l'analyse des réponses de test à l'interne du SOC.

Les résultats de validation de l'architecture de test proposée ont été obtenus par simulation sur des circuits d'essai. Ces résultats ont montré que l'architecture de test proposée permet une réduction de la surface du semi-conducteur. Cette réduction peut dépasser les 9% avec les circuits qui présentent un rapport plots/surface initiale élevé.

ABSTRACT

Systems On Chip (SOC) are getting more complex and their tests become difficult to perform. During the last few years, several test architectures were suggested to these systems to guarantee higher levels of test quality and reduce the costs. The principal disadvantages of these architectures are the area overhead and the internal delay degradations.

The main objective of this work is to reduce the overhead area caused by the test architectures insertion. Therefore, we suggest a new architecture that can be used to test SOC modules and the interconnections between the modules. This architecture is a wrapper in which SOC modules can be encapsulated. In normal mode, this wrapper becomes transparent to allow a normal operation of the modules. In test mode, the wrapper can be configured as a shift register and as an integrated test structure. The shift register is used to transfer test data to be applied to the input signals of modules and to interconnects from external test equipments. This shift register is also used to transport test responses to external test equipments for analysis. The integrated test structure is used to generate and apply test data to the input signals of the modules and to the interconnects, and also to analyze the responses inside SOC. The validation results of the new test architecture are obtained by simulation using benchmarks circuits.

These results show that with the proposed test architecture the area overhead reduction can exceed 9% when the circuits have a higher ratio pins/initial area.

TABLE DES MATIÈRES

DÉDICACE.....	IV
REMERCIEMENTS.....	V
RÉSUMÉ.....	VI
ABSTRACT.....	VIII
TABLE DES MATIÈRES	X
LISTE DES FIGURES.....	XV
LISTE DES TABLEAUX.....	XVIII
LISTE DES SIGLES ET ABRÉVIATIONS	XIX
INTRODUCTION GÉNÉRALE	1
I.1 Motivation.....	1
I.2 Organisation du mémoire.....	5
CHAPITRE 1 CONCEPTS DE BASE.....	7
1.1 Introduction.....	7
1.2 Système sur puce et les modules IP	7
1.3 Structure de test du module IP	10
1.4 Génération de vecteurs de test pour les SOC	12
1.4.1 Génération déterministe	12

1.4.2	Génération pseudo-aléatoire.....	13
1.5	Conception en vue de test	14
1.6	Test intégré (BIST)	16
1.6.1	Architecture BIST	17
1.6.1.1	Générateur de vecteurs de test	17
1.6.1.2	Analyseur des réponses.....	19
1.7	Norme IEEE P1500.....	20
1.7.1	Enveloppe P1500	21
1.7.2	Langage de test « Core Test Language » (CTL)	23
1.8	Conclusion	23
CHAPITRE 2 REVUE DES TECHNIQUES DE TEST DES SOC		24
2.1	Introduction.....	24
2.2	Architectures de test des modules IP	24
2.2.1	Architectures du test externe pour les modules IP	25
2.2.2	Architectures de test intégré pour les modules IP.....	27
2.2.2.1	Conversion des vecteurs de test aléatoires en vecteurs déterministes ...	28
2.2.2.2	BIST associé aux chaînes de balayage de l'IP	29
2.2.2.3	Conversion des vecteurs de test déterministes en vecteurs aléatoires ...	30
2.3	Architectures de test des interconnexions entre les modules IP	31
2.4	Conclusion	34
CHAPITRE 3 ARCHITECTURE DE TEST PROPOSÉE		35

3.1	Introduction.....	35
3.2	Présentation de la nouvelle architecture de test	36
3.3	Architecture de la nouvelle enveloppe.....	37
3.4	Cellules de l'enveloppe.....	39
3.4.1	Cellule à deux états de l'enveloppe.....	41
3.4.2	Modes de fonctionnement de WBC	43
3.4.3	Cellule à trois états de l'enveloppe	45
3.4.4	Cellule bidirectionnelle de l'enveloppe	47
3.4.5	Autres variantes des cellules de l'enveloppe	48
3.5	Registres de l'enveloppe	49
3.5.1	Registre d'instruction WIR	49
3.5.2	Registre de données WBR	52
3.5.3	Registre de données WBY	54
3.6	Contrôleur de test.....	55
3.7	Instructions de test implémentées par la nouvelle enveloppe	57
3.7.1	Instructions de test sans fonctionnalité BIST.....	58
3.7.1.1	Instruction WBYPASS	59
3.7.1.2	Instruction WEXTTESTS	60
3.7.1.3	Instruction WCORETESTS	61
3.7.1.4	Instruction WPRELOADS.....	62
3.7.1.5	Instruction WCLAMP.....	64
3.7.2	Instructions de test avec fonctionnalité BIST	65

3.7.2.1	Instruction WEXBIST	65
3.7.2.2	Instruction WCOREBIST	67
3.7.2.3	Instruction de test WOPMISR	68
3.8	Conclusion	70
CHAPITRE 4 IMPLÉMENTATION ET RÉSULTATS		71
4.1	Introduction	71
4.2	Insertion de l'architecture de test	72
4.3	Simulation fonctionnelle de l'architecture de test proposée	73
4.4	Chronogrammes des instructions de test sans BIST	74
4.4.1	Instruction de test WBAYPASS	75
4.4.2	Instruction de test WEXTESTS	76
4.4.3	Instruction de test WCORETESTS	77
4.4.4	Instruction de test WPRELOADS	78
4.5	Chronogrammes des instructions de test avec BIST	79
4.5.1	Instruction de test WEXBIST	79
4.5.2	Instruction de test WCOREBIST	82
4.5.3	Instruction de test WOPMISR	83
4.6	Délais de propagation	85
4.7	Estimation du gain en surface	86
4.7.1	Présentation des circuits d'essai	87
4.7.2	Résultats du gain en surface	88
4.8	Réduction du nombre de vecteurs de test déterministes	90

4.9	Conclusion	92
CHAPITRE 5 CONCLUSION ET TRAVAUX FUTURS		93
5.1	Conclusion générale	93
5.2	Travaux futurs	95
RÉFÉRENCES.....		97
ANNEXE.....		103

LISTE DES FIGURES

Figure 1.1: Modules IP qui peuvent composer un SOC.	8
Figure 1.2: Flot de conception du système sur puce [44].	9
Figure 1.3: Structure de test pour un IP du SOC [41].	10
Figure 1.4: Circuit séquentiel (a) avant insertion chaîne de balayage, ...	15
Figure 1.5: Architecture du test intégré.	17
Figure 1.6: Architecture d'un LFSR générique d'ordre N et à XOR externe.	18
Figure 1.7: Analyseur de réponses de test.	19
Figure 1.8: Architecture d'un MISR générique à XOR internes d'ordre N.	20
Figure 1.9: Architecture de l'enveloppe IEEE P1500.	21
Figure 2.1: Architecture du bus de test TestRail [31].	25
Figure 2.2: Architecture de test Cas-Bus [7].	27
Figure 2.3: Exemple de la logique de génération de séquence de test [25].	30
Figure 2.4: Combinaison du BIST et du Boundary Scan pour le test des interconnexions [5].	31
Figure 3.1: Architecture de test proposée.	36
Figure 3.2: Architecture de l'enveloppe proposée.	38
Figure 3.3: Cellule à deux états WBC.	42
Figure 3.4: Cellule de sortie à 3 états (WBOC3)	45
Figure 3.5: Cellule de contrôle de la sortie de la cellule WBC3.	46
Figure 3.6: Cellule bidirectionnelle.	47

Figure 3.7: Schéma synoptique des cellules : (a) WBC&XOR ; (b) WBC3&XOR.....	49
Figure 3.8: Registre d'instruction WIR [19].....	51
Figure 3.9: Registre à décalage WBR.....	52
Figure 3.10: Architecture du registre de court-circuit [19].....	55
Figure 3.11: Diagramme de transition du contrôleur TAP [19].....	55
Figure 3.12: Instruction WBYPASS.....	59
Figure 3.13: Instruction WEXTTESTS.....	60
Figure 3.14: Instruction WCORETESTS.....	61
Figure 3.15: Instruction WPRELOADS.....	63
Figure 3.16: Instruction WCLAMP.....	64
Figure 3.17: Instruction WEXBIST.....	66
Figure 3.18: Instruction WCOREBIST.....	67
Figure 3.19: Instruction WOPMISR.....	68
Figure 4.1: Tampon 4 bits encapsulé dans l'enveloppe de test proposée.....	74
Figure 4.2: Chronogramme de l'instruction de test WBYPASS.....	75
Figure 4.3: Chronogramme de l'instruction de test WEXTTESTS.....	76
Figure 4.4: Chronogramme de l'instruction de test WCORETESTS.....	77
Figure 4.5: Chronogramme de l'instruction de test WPRELOADS.....	78
Figure 4.6: Chronogramme de l'instruction de test WEXBIST.....	80
Figure 4.7: Séquence complète générée par un LFSR d'ordre 4 et $P=1+x^3+x^4$	81
Figure 4.8: MISR à XOR externe initialisé à F.....	81
Figure 4.9: Chronogramme de l'instruction de test WCOREBIST.....	82

Figure 4.10: Génération des vecteurs de test et le compactage des réponses.	83
Figure 4.11: Chronogramme de l'instruction de test WOPMISR.	84
Figure 4.12: Illustration de l'instruction de test WOPMISR.	85
Figure 4.13: Cellule du registre WBR: a) enveloppe proposée; b) enveloppe P1500.	86
Figure 4.14: Gain en surface versus le rapport (plots/surface) du circuit.	90

LISTE DES TABLEAUX

Tableau 3.1: Différents modes de configuration de WBC.....	43
Tableau 3.2: Valeurs des signaux de contrôle pour les instructions de test.....	57
Tableau 4.1: Circuits utilisés pour la validation de l'architecture de test proposée.	88
Tableau 4.2: Résultats du gain en surface.....	89
Tableau 4.3: Réduction du nombre de vecteurs de test déterministes.	91

LISTE DES SIGLES ET ABRÉVIATIONS

ATE	Automated Test Equipment
ATPG	Automated Test Patterns Generator
BIST	Built-In Self-Test
BSC	Boundary Scan Cell
bc	BIST cell
CTL	Core Test Language
CAS	Core Access Switch
CST	Circuit Sous Test
CI	Circuit Intégré
cfi	Cell functional in
cfo	Cell functional out
cti	Cell test in
cto	Cell test out
DFT	Design For Testability
fdi	Feedback in
fdo	Feedback out
HDL	Hardware Description Language
IP	Intellectual Properties
LFSR	Linear Feedback Shift Register
MISR	Multi In Put Shift Register

OPMISR	On Product MISR
PRPG	Pseudo Random Patterns Generator
PR	Parallel Register
pdr_in	Parallel data register in
pdr_out	Parallel data register out
pir	Parallel instruction register
SIA	Semiconductor International Association
SOC	System On Chip
SOB	System On Board
SR	Serial Register
sdr_in	Serial data register in
sdr_out	Serial data register out
sir	Serial instruction register
TAM	Test Access Mecanism
TMS	Test Mode Select
TC	Taux de Couverture
VHDL	Very High Speed Integrated Circuit Hardware Descreption Language
WIR	Wrapper Instruction Register
WBR	Wrapper Boundary Register
WBY	Wrapper Bypass Register
WSI	Wrapper Scan In
WSO	Wrapper Scan Out

WBC	Wrapper Boundary Cell
WBC3	Three State Wrapper Boundary Cell
wrck	Wrapper clock
wc	Wrapper cell

INTRODUCTION GÉNÉRALE

I.1 Motivation

Les différentes applications de l'électronique dans divers domaines scientifiques et techniques exigent de nouvelles générations de circuits intégrés (CI). Ces circuits doivent posséder plusieurs fonctionnalités et doivent être : rapides, compacts, peu coûteux et fiables. Grâce à l'évolution de la microélectronique et des outils d'aide à la conception, il est désormais possible de répondre favorablement à ces exigences. Actuellement, l'industrie des semi-conducteurs est capable de fabriquer de nouveaux circuits qui sont fortement intégrés. Ces nouveaux circuits sont connus sous le nom de système sur puce « System on Chip » (SOC). À cause de la forte concurrence qui règne dans cette industrie, ce secteur fait face à des défis importants. D'une part, il faut développer rapidement des composants complexes et d'autre part, il faut les produire à moindre coût. Comme solution à ces défis, les industriels ont adopté un nouveau flot de production pour les SOC en se basant sur le principe de la réutilisation « reuse » des modules préconçus. Ces modules sont nommés « Intellectual Properties » (IP) ou « cores » [45].

Les IP permettent de concevoir des SOC très complexes en un temps court, mais ils rendent le test de ces systèmes difficiles et coûteux à accomplir. Ces difficultés sont

dues à: 1) l'inaccessibilité aux plots d'entrée/sortie primaires des IP à partir de ceux du SOC; 2) la diversité des sources de provenance des IP qui fait que chaque module peut avoir une stratégie de test différente de celle des autres IP. La figure I.1 illustre les résultats de l'étude réalisée par la « Semiconductor International Association » (SIA) [42] qui prédit que dans un futur proche, le coût de test des SOC risque de dépasser le coût de leur fabrication.

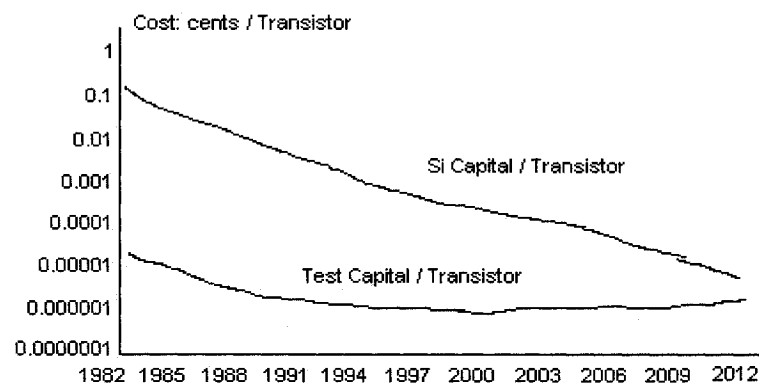


Figure I. 1: Coût de production par transistor versus coût de test par transistor [38].

À partir de cette étude, on peut donc conclure que, dans le futur, le principe de la réutilisation des IP ne pourra pas assurer la réduction du coût de production des SOC. Pour réduire le coût de production de ces systèmes, il faut également penser à minimiser le coût de leur test.

Les techniques de test développées pour les systèmes électroniques sur cartes « System On Board » (SOB) ne peuvent être appliquées aux SOC. En effet, les SOB sont conçus à partir de CI qui ont déjà été fabriqués et testés. De ce fait, le test de ces systèmes est

limité au test des interconnexions entre les CI. La réduction du coût de test des SOC n'est possible que par l'utilisation de nouvelles approches de test développées spécifiquement pour eux. Dans cette optique, plusieurs architectures de test ont été proposées, mais elles sont destinées soit aux IP soit aux interconnexions entre les IP. Il n'existe pas de technique qui permette de tester efficacement les IP et les interconnexions avec une même structure de test. Pour un test complet d'un SOC, il faut donc insérer des architectures pour le test des IP et d'autres pour le test des interconnexions. Ceci engendre une dégradation des délais internes et un accroissement important de la surface. Parmi les techniques de test développées pour améliorer la testabilité des SOC, on peut mentionner les deux plus importantes qui sont le test intégré « Built-In Self-Test » (BIST) et la norme IEEE P1500.

La technique BIST offre des structures de test peu coûteuses qui permettent la génération et l'application des données de test au cœur des IP et aux interconnexions, ainsi que l'analyse des réponses à l'interne du SOC. En général, le BIST n'assure pas une bonne qualité de test, ceci est dû à la nature aléatoire de ses vecteurs de test qui ne permettent pas de vérifier efficacement la présence ou l'absence des pannes difficiles.

La norme IEEE P1500 permet d'accéder aux signaux d'entrée/sortie des IP à partir de l'extérieur. Cette norme, qui utilise des vecteurs de test déterministes, permet de tester les IP et les interconnexions du SOC avec des équipements de test externes. Les vecteurs de test déterministes sont générés à l'externe du SOC par des générateurs

automatiques « Automated Test Patterns Generator » (ATPG). L'inconvénient majeur de la norme IEEE P1500 est l'utilisation prolongée des équipements de test externes « Automated Test Equipment » (ATE) qui sont généralement très dispendieux. Le temps d'utilisation d'ATE dépend de la longueur de la séquence de test et de la longueur du registre à décalage à travers lequel les valeurs de test sont appliquées aux IP et aux interconnexions.

À cause de l'augmentation rapide de la complexité des SOC et de leur vitesse, les ATE deviennent dépassés en termes d'espace mémoire et de vitesse. Afin d'assurer une bonne qualité de test pour les SOC et de réduire le temps d'application des vecteurs, les manufacturiers sont appelés à remplacer plus souvent leurs ATE. Ceci nécessite des investissements additionnels, qui font augmenter le coût de test des SOC.

L'objectif de notre travail est de proposer une architecture de test qui permet de vérifier les IP et les interconnexions des SOC avec une même structure de test. Ceci a pour but la réduction de la surface additionnelle et la minimisation des dégradations des délais internes qui sont engendrées par l'insertion des architectures de test dans les SOC. La nouvelle architecture doit être aussi configurable en structure BIST et en enveloppe P1500 pour permettre à l'intégrateur des IP de réaliser des bons compromis entre la qualité de test et le temps d'utilisation des ATE.

I.2 Organisation du mémoire

Ce mémoire est principalement structuré en quatre chapitres :

Le premier chapitre est consacré aux concepts de base du test des SOC. Nous y exposons un rappel sur les SOC et les différentes catégories d'IP qui rentrent dans leur conception. Nous abordons également les spécificités du test des SOC et finalement, les techniques de conception en vue de test conçues pour améliorer la testabilité des SOC.

Le deuxième chapitre est un survol des architectures existantes développées pour le test des SOC. Dans ce chapitre, nous présentons différentes architectures de test proposées pour les IP et les interconnexions des SOC. En premier lieu, nous exposons les techniques de test externe et celles qui sont intégrables qui ont été développées pour les IP. En dernier, nous présentons les techniques de test externe et les techniques BIST qui ont été conçues pour les interconnexions.

Le troisième chapitre est relatif à l'architecture de test proposée. Nous y présentons l'architecture et le fonctionnement des cellules d'entrée/sortie à deux états, à trois états, bidirectionnelles et des différents registres qui composent la structure de test proposée. Dans ce chapitre, nous parlons également du contrôleur de test qui gère le fonctionnement de la nouvelle architecture de test. À la fin de ce chapitre, nous illustrons et expliquons les différentes instructions de test que l'architecture proposée implémente, pour le test des IP et des interconnexions, à partir de l'extérieur et à l'interne du SOC.

Le quatrième chapitre présente les résultats de la validation de l'architecture de test proposée qui ont été obtenus par simulation sur des circuits d'essai. Ces résultats se rapportent au gain en surface et à la réduction du temps d'application des vecteurs de test. Ce chapitre est suivi par la conclusion générale et les perspectives pour des travaux futurs.

CHAPITRE 1

CONCEPTS DE BASE

1.1 Introduction

Dans ce chapitre, nous présentons les concepts de base concernant le test des SOC. Parmi ces concepts, nous abordons les spécificités du test du SOC et les types de vecteurs de test qui peuvent être appliqués aux SOC. Nous parlons également du BIST qui est la technique « Design For Testability » (DFT) la plus utilisée pour le test des SOC et de la norme IEEE P1500 développée pour standardiser le test des SOC. Pour mieux comprendre d'où vient la difficulté à tester un SOC, nous commençons par un rappel sur ces systèmes et sur les différentes catégories d'IP qui entrent dans leur conception.

1.2 Système sur puce et les modules IP

Le SOC est un circuit intégré très complexe, qui peut contenir plusieurs millions de transistors. Ce circuit peut remplacer tout un système électronique composé de plusieurs cartes. La figure 1.1 montre que le SOC est conçu par l'intégration de différents modules IP : numérique, analogique, mémoire, microprocesseur, module RF, etc. Ces modules IP sont classés en trois catégories [16].

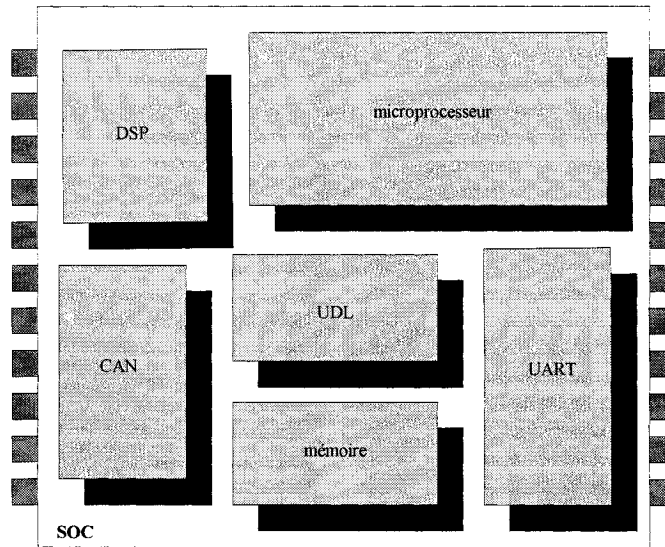


Figure 1.1: Modules IP qui peuvent composer un SOC.

La première catégorie est celle des IP matériels « Hard Core » qui sont des modules placés et routés suivant une bibliothèque technologique spécifique. Ces IP qui sont prêts à l'usage, sont livrés à l'utilisateur sous forme de boîtes noires avec des performances optimisées (puissance, délais, etc.). La portabilité de ces modules d'une technologie à une autre est faible mais la propriété intellectuelle associée à ces modules est bien protégée et leurs performances finales sont prévisibles.

La deuxième catégorie est celle des IP logiciels « Soft Core » qui sont livrés sous forme de code source synthétisable d'un niveau d'abstraction élevé. Ceci procure pour ce type d'IP une bonne portabilité d'une technologie à une autre. Par contre, la propriété intellectuelle associée à ces modules est non protégée et leurs performances finales dépendent de la technologie utilisée pour leur implémentation.

La troisième catégorie est celle des IP dits « Firm Core » qui se situe entre les IP câblés (Hard Core) et ceux qui sont entièrement logiciel (Soft Core). Ces IP sont décrits en « Hardware Description Language » (HDL) structurel en utilisant des composants élémentaires d'une bibliothèque générique. Les IP « Firm Core » sont livrés à l'intégrateur, placés et optimisés en surface et en vitesse mais non routés. Les performances de ces IP et leurs tailles sont prévisibles mais la protection de leur propriété intellectuelle n'est pas assurée.

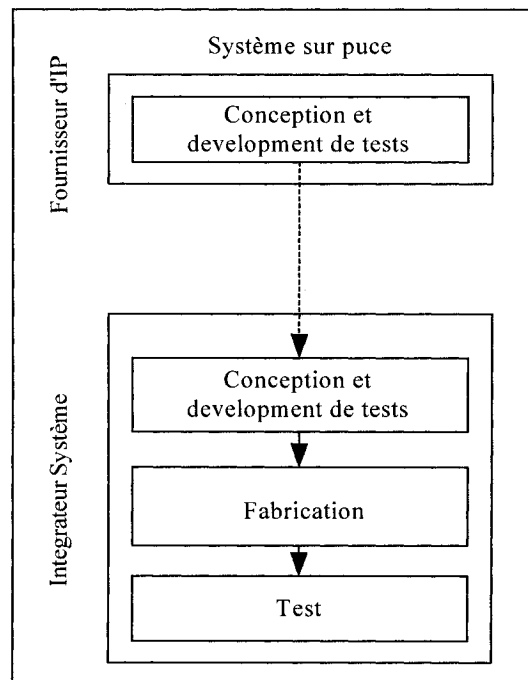


Figure 1.2: Flot de conception du système sur puce [45].

D'après le flot de conception des SOC présenté à la figure 1.2, nous distinguons deux types d'intervenants dans cette industrie. Il y a les fournisseurs qui conçoivent les IP et élaborent leurs stratégies de test, et les intégrateurs qui combinent les IP pour concevoir

des SOC. Les intégrateurs assurent la vérification de ces systèmes à la sortie du procédé de fabrication en réutilisant les vecteurs de test livrés avec les IP. Ces vecteurs de test, qui peuvent être déterministes ou aléatoires, tel que discuté à la section 1.4, permettent de détecter les défauts physiques qui peuvent affecter le bon fonctionnement du SOC. Les défauts physiques sont provoqués notamment par les imperfections des tranches du semi-conducteur « wafer » (dislocations, impuretés, etc.), l'instabilité du processus de fabrication, l'opérateur humain etc. [34]. L'application des vecteurs de test aux IP à la dernière étape de la fabrication du SOC n'est pas une opération facile à réaliser. Cette difficulté est due à l'inaccessibilité aux signaux d'entrée/sortie des IP à partir des plots primaires du SOC. Il faut donc que l'intégrateur prévoie des structures de test pour les IP dès le début de la conception du SOC.

1.3 Structure de test du module IP

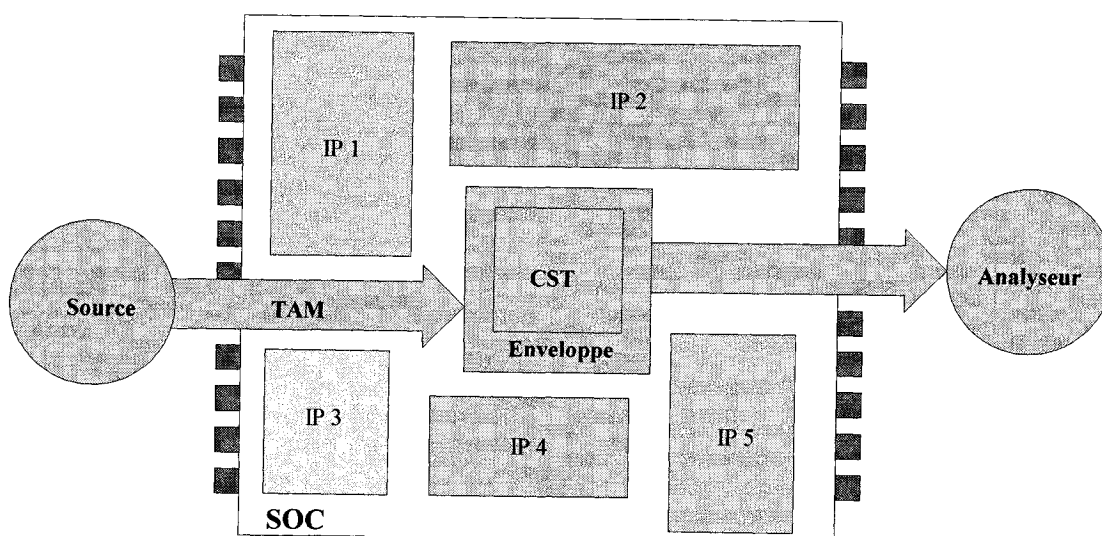


Figure 1.3: Structure de test pour un IP du SOC [41].

La figure 1.3 illustre le schéma de principe de la structure de test de l'IP que doit prévoir l'intégrateur dès le début de la conception du SOC. Cette structure est composée de trois éléments [45] : source/analyseur, mécanisme d'accès au test et enveloppe.

La source est l'élément à partir duquel, les vecteurs de test sont appliqués au circuit sous test (CST). L'analyseur est l'élément qui collecte les réponses du test du CST et les compare aux réponses du circuit correct. La source et l'analyseur peuvent être intégrés ou non dans le SOC.

Le « Test Access Mechanism » (TAM) est le chemin d'accès aux tests qui sert à transporter les vecteurs, de la source au CST et les réponses, du CST à l'analyseur. Les TAM peuvent être réalisés selon trois principes : 1) la possibilité de propager les données de test à travers les IP sans perdre d'information [11] [12], ce que nous appelons la transparence des IP; 2) utilisation des bus fonctionnels du système pour acheminer les données de test vers les IP [4]; 3) insertion des bus supplémentaires dédiés aux tests [44].

Le troisième et dernier élément de cette structure est l'enveloppe qui est une interface matérielle insérée entre l'IP et le reste du SOC. Cette interface permet d'accéder aux entrées et aux sorties de l'IP à partir de l'extérieur du SOC. En mode normal, l'enveloppe connecte l'IP aux autres modules du SOC, alors qu'en mode test, elle isole l'IP du reste du SOC et relie ses signaux d'entrée/sortie à un ATE externe à travers le TAM.

1.4 Génération de vecteurs de test pour les SOC

De nos jours, il existe des SOC avec plusieurs dizaines de plots d'entrée. Le test exhaustif de ces SOC nécessite de longues séquences de test dont le temps d'application peut atteindre plusieurs heures ou jours. Ceci n'est pas réaliste dans l'industrie des semi-conducteurs qui exige de faibles coûts de production. Pour détecter le maximum de pannes en un temps court, l'une des trois approches de test citées ci-après peut être adoptée: 1) l'approche pseudo-exhaustive [45] qui propose de décomposer un réseau de portes logiques en une série de petits réseaux et de tester chacun d'eux exhaustivement. Cette approche pseudo-exhaustive garantit une bonne qualité de test et réduit le nombre de vecteurs à appliquer. Mais, la décomposition d'une structure logique en sous circuits n'est pas toujours faisable ; 2) l'approche déterministe qui utilise des vecteurs de test spécifiques pour détecter des pannes spécifiques ; 3) l'approche pseudo aléatoire qui permet de détecter un bon nombre de pannes en utilisant uniquement des vecteurs de test générés aléatoirement à faible coût. Dans cette section, nous exposons seulement les deux dernières approches de test parmi celles présentées ci-dessus. En effet, l'approche déterministe et la pseudo aléatoire sont les plus efficaces et les plus utilisées dans l'industrie des semi-conducteurs.

1.4.1 Génération déterministe

Dans toute séquence de test exhaustif ou pseudo exhaustif, seul un nombre restreint de vecteurs sont efficaces pour la détection de pannes. Il est donc profitable de générer

uniquement les vecteurs de test utiles à la détection des pannes en se basant sur la génération déterministe. Ce type de génération, qui est orienté pannes, garantit un vecteur de test, s'il en existe un bien sûr [45]. Les vecteurs de test déterministes sont générés par des ATPG suivant des modèles de pannes. Parmi ces modèles de pannes, les plus utilisés sont le modèle de collage [33] et le modèle de court-circuit [41]. La génération des vecteurs de test déterministes est un problème NP complet. C'est pourquoi, les ATPG qui génèrent ces vecteurs, utilisent des Algorithmes tels que l'Algorithme D [37], PODEM [14] ou FAN [10]. Ces algorithmes fournissent des séquences de test efficaces mais qui ne sont pas forcément optimales. L'inconvénient majeur de la génération déterministe est le temps de génération qui augmente rapidement avec la complexité des IP. Malgré cet inconvénient, le test déterministe reste largement utilisé dans l'industrie de la microélectronique, parce qu'il assure une bonne qualité de test.

1.4.2 Génération pseudo aléatoire

Le test pseudo aléatoire consiste à utiliser des vecteurs de test générés aléatoirement. L'idée d'utiliser des vecteurs aléatoires est venue de l'application des vecteurs de test déterministes qui souvent détectent d'autres pannes en plus de celles pour lesquelles ils ont été développés [41]. Les vecteurs de test aléatoires peuvent être générés et appliqués à l'interne du circuit en utilisant un module de génération comme le registre linéaire à décalage rebouclé « Linear Feedback Shift Register » (LFSR) qui est présenté à la

section 1.6. Les vecteurs aléatoires peuvent également être générés par un programme informatique et appliqués à partir de l'extérieur du circuit en utilisant des ATE. Le test aléatoire permet d'atteindre une certaine qualité de test pour les SOC à moindre coût, mais il ne permet pas de détecter certaines pannes difficiles à détecter. La génération pseudo aléatoire est habituellement utilisée dans les techniques de test intégré et par les ATPG avant de passer au mode déterministe pour réduire le temps de génération des séquences de test.

1.5 Conception en vue de test

Les SOC sont des circuits intégrés complexes qui peuvent contenir plusieurs millions de portes logiques par puce, ce qui conduit à un rapport portes logiques/plots très élevé. Ceci rend la contrôlabilité et l'observabilité des signaux internes du SOC à partir des plots d'entrée/sortie très difficile à réaliser. Pour améliorer la testabilité des SOC, plusieurs techniques de conception en vue du test (DFT) ont vu le jour. La DFT consiste à insérer du matériel dédié au test dans un circuit à condition de ne pas changer sa fonctionnalité. Les techniques de DFT permettent de mieux contrôler et d'observer les nœuds internes du circuit à partir de l'extérieur et à l'interne du circuit. Parmi ces techniques, nous pouvons énumérer l'insertion des points de test et des chaînes de scan. Ces deux techniques sont les premières techniques DFT à être utilisées par les concepteurs. Nous pouvons également citer le BIST et la norme IEEE P1500 que nous détaillons dans les prochaines sections 1.6 et 1.7.

La technique d'insertion de points de test [40] propose d'ajouter du matériel à certains nœuds internes du circuit qui sont difficiles à tester. Cette technique nous permet d'imposer des valeurs externes aux nœuds internes et d'observer les réponses aux sorties du circuit. L'inconvénient de cette technique est le nombre élevé de plots supplémentaires à ajouter au circuit. Cette technique d'insertion des points de test ne peut pas être appliquée aux SOC limités en nombre de plots.

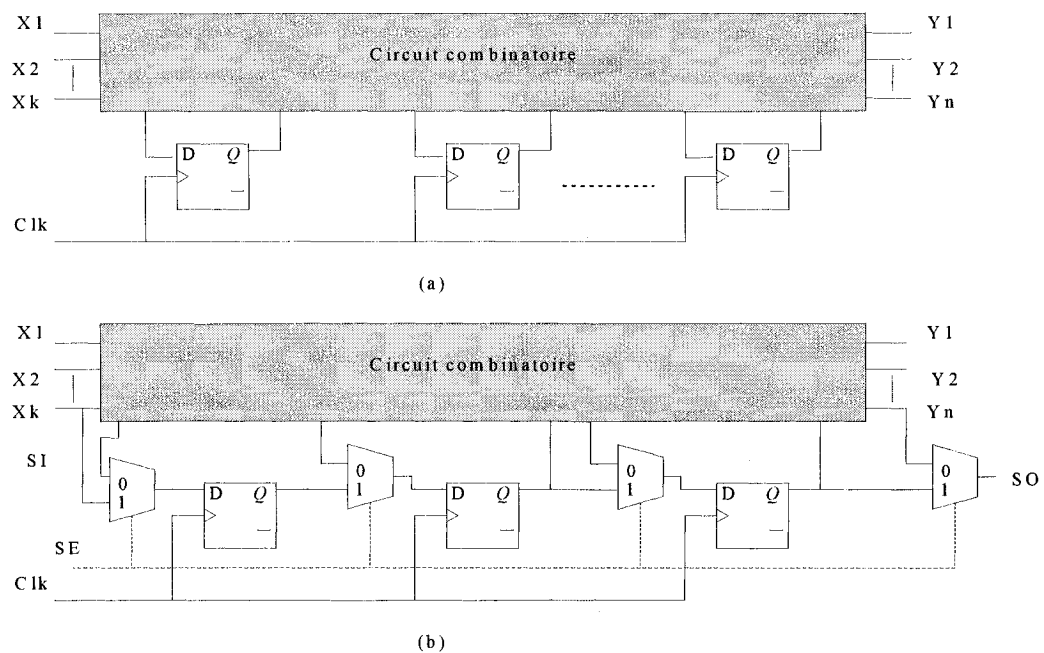


Figure 1.4: Circuit séquentiel (a) avant insertion chaîne de balayage, (b) après insertion chaîne de balayage.

La figure 1.4 montre le principe de la technique chaînes de balayage [34] qui consiste à remplacer tous les éléments mémoires d'un circuit séquentiel par des éléments mémoires contrôlés. Cette technique de test permet d'imposer des valeurs aux bascules

d'un IP séquentiel à partir de l'extérieur. En mode normal, les éléments mémoires agissent en tant que bascule normale. En mode test, les éléments mémoires sont connectés en série et forment un long registre à décalage. Ce registre sert à transporter en série les valeurs de test à appliquer aux entrées de la partie combinatoire et à récupérer les réponses de test de la partie combinatoire pour les transmettre vers un ATE externe pour analyse. Les inconvénients de la technique chaîne de balayage sont : 1) la surface additionnelle nécessaire pour l'implémentation des éléments mémoires contrôlés; 2) le temps de test qui dépend de la longueur de la chaîne de balayage et du nombre de vecteurs de test à appliquer.

1.6 Test intégré (BIST)

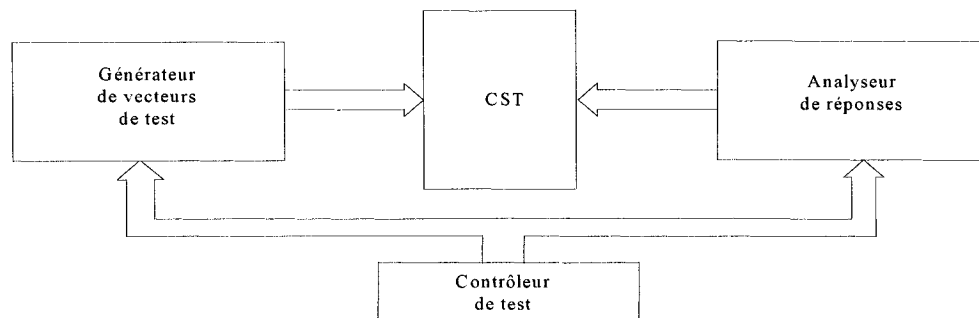
Le BIST [1] [2] [26] qui est une technique d'autotest, est l'une des méthodes DFT les plus utilisées. Cette technique permet de générer et d'appliquer des vecteurs de test ainsi que d'analyser des réponses de test à l'interne du SOC. L'usage de cette technique de test s'est généralisé suite aux problèmes de testabilité des CI rencontrés vers le début des années 90 [38]. Ces problèmes sont: 1) l'impossibilité de tester les CI à leur vitesse nominale, parce que les ATE externes sont généralement moins rapides que les circuits à tester; 2) l'augmentation du temps de génération des vecteurs de test et du temps de leur application; 3) l'augmentation du nombre de vecteurs de test à appliquer à partir d'ATE externes. Le BIST offre certains avantages comparativement aux techniques de test qui utilisent des ATE externes [22]. Premièrement, les vecteurs de test sont générés de façon pseudo aléatoirement à faible coût. Deuxièmement, l'application des vecteurs de test et

l'analyse des réponses de test se font à l'interne du SOC. Ceci donne la possibilité de tester les IP et les interconnexions à la vitesse nominale des SOC, ce qui réduit la durée d'application des vecteurs de test. Malgré ces avantages, l'utilisation du BIST engendre certains inconvénients comme l'augmentation de la surface et les dégradations des délais internes du circuit.

1.6.1 Architecture BIST

La figure 1.5 montre le schéma de principe d'une structure BIST pour le test du CST à l'interne. Cette structure BIST est composée d'un générateur de vecteurs de test, d'un analyseur de réponses et d'un contrôle de test qui sert à gérer le fonctionnement de la structure BIST.

Figure 1.5: Architecture du test intégré.



1.6.1.1 Générateur de vecteurs de test

Le générateur est la source à partir de laquelle proviennent les vecteurs de test qui sont appliqués au CST. Plusieurs techniques peuvent être utilisées pour générer à l'interne les vecteurs de test [9], parmi elles, nous citons: les ROM et les LFSR.

Une mémoire ROM permet de stocker des vecteurs de test déterministes générés par un ATPG externe. Ces vecteurs qui garantissent une bonne qualité de test sont emmagasinés pour être appliqués au CST. Mais la taille de la mémoire dans laquelle ces vecteurs sont emmagasinés peut être importante, ce qui peut engendrer une augmentation importante de la surface du circuit.

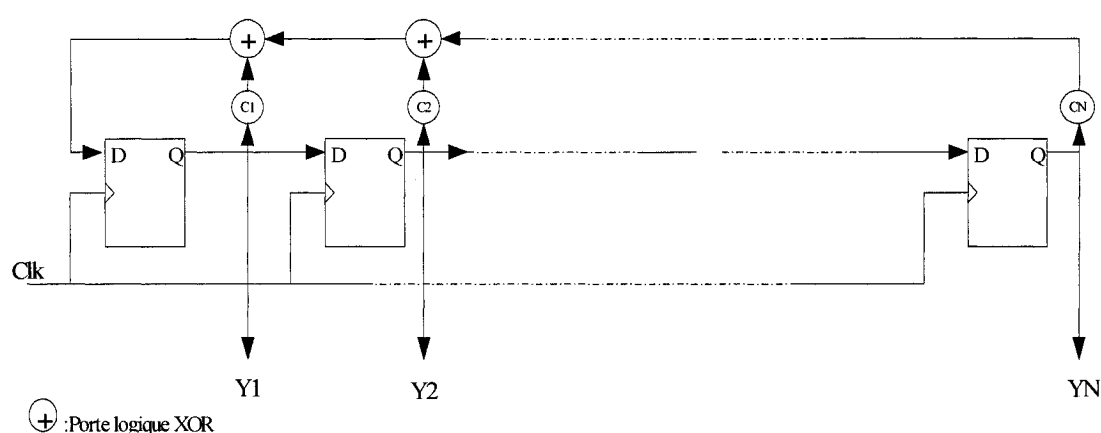


Figure 1.6: Architecture d'un LFSR générique d'ordre N et à XOR externe.

Le LFSR [40] est le moyen le plus utilisé pour la génération des vecteurs de test à l'intérieur d'un circuit. La figure 1.6 représente le schéma d'un LFSR générique d'ordre N (N : nombre de bascules) à XOR externe. Ce registre est qualifié de registre autonome, parce qu'il n'a aucune autre entrée à part celle de l'horloge. Chaque fois que le coefficient C_i ($1 \leq i \leq N$) est à 1, l'étage du LFSR qui lui correspond est rebouclé vers l'entrée de la première bascule du LFSR à travers une porte XOR. À chaque front montant de l'horloge, la bascule suivante prend la valeur de la bascule précédente, sauf la première bascule qui prend le résultat de l'opération XOR réalisée entre tous les rebouclages. Les sorties des différentes bascules constituent les bits du vecteur de test

généralisé. Le LFSR initialisé à une valeur autre que 00...0, génère $2^N - 1$ vecteurs de test avant de recommencer la même séquence.

1.6.1.2 Analyseur des réponses

L'analyseur sert à compacter les réponses de test du CST pour calculer la signature de test et à comparer cette signature à celle du circuit correct. La Figure 1.7 montre le schéma de principe d'un analyseur des réponses qui est composé d'un compacteur, d'un comparateur et d'une mémoire ROM. Dans certains cas, l'analyseur peut uniquement être formé d'un compacteur si la comparaison de la signature de test à celle du circuit correct est faite à l'extérieur du circuit.

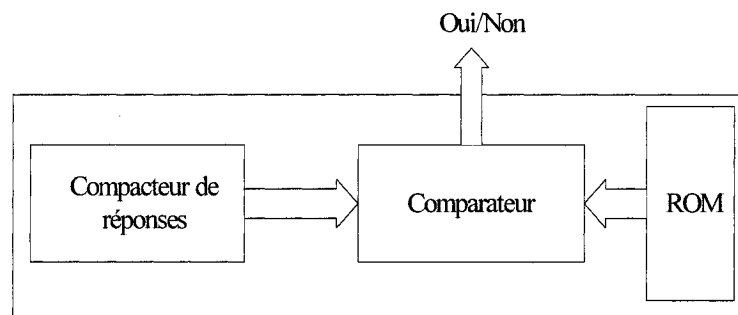


Figure 1.7: Analyseur de réponses de test.

Le compactage des réponses de test peut être réalisé par plusieurs techniques comme le comptage de transition de 0 à 1 et de 1 à 0 et le comptage des 1 [34]. Mais, le compactage des réponses de test avec le registre à décalage à plusieurs entrées « Multi Input Shift Register » (MISR) [9] reste la méthode la plus utilisée dans la technique BIST. La figure 1.8 représente le schéma d'un MISR générique d'ordre N à XOR

interne. La porte XOR du premier étage du MISR est contrôlée par le signal d'entrée et le signal de rebouclage. Les autres portes XOR du MISR sont contrôlées par le signal d'entrée, la sortie de la bascule précédente et le signal de rebouclage. À chaque front montant de l'horloge, le résultat de la porte XOR est emmagasiné dans la bascule suivante. Les valeurs finales des sorties des bascules du MISR constituent la signature du CST. L'inconvénient majeur du compactage des réponses avec le MISR est le masquage d'erreur. Le masquage est la possibilité qu'une erreur masque une autre erreur, ce qui permet au circuit fautif d'avoir la même signature que celle du circuit correct. Pour minimiser ce problème, nous pouvons utiliser des MISR d'ordre N élevé ou bien prendre des signatures intermédiaires avant la signature finale du test.

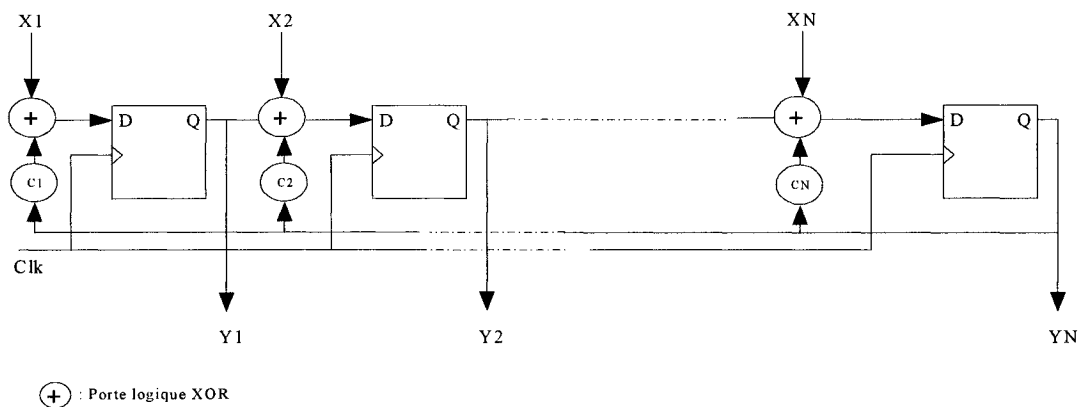


Figure 1.8: Architecture d'un MISR générique à XOR internes d'ordre N.

1.7 Norme IEEE P1500

Dans l'industrie des SOC, le test de ces systèmes est une opération partagée entre les fournisseurs des IP et les intégrateurs de ces IP. Les fournisseurs insèrent les structures de test dans les IP et génèrent les vecteurs de test pour ces modules. Les intégrateurs

élaborent des stratégies de test pour les SOC conçus en réutilisant les vecteurs de test fournis avec les IP. Pour réussir le test des IP qui sont généralement livrés à l'intégrateur sous forme de boîtes noires, il faut transmettre du concepteur à l'intégrateur toutes les informations concernant le test de ces modules. Dans cette optique, un groupe constitué d'universitaires et d'industriels, travaille depuis septembre 1995 pour élaborer la norme IEEE P1500 destinée au test des SOC [20] [32] [39]. Cette norme a été inspirée de la norme IEEE 1149.1 [19] qui a été développée pour le test des SOB. Ce groupe de travail a proposé une structure de test nommée enveloppe P1500 et un langage de test appelé « Core Test Language » (CTL).

1.7.1 Enveloppe P1500

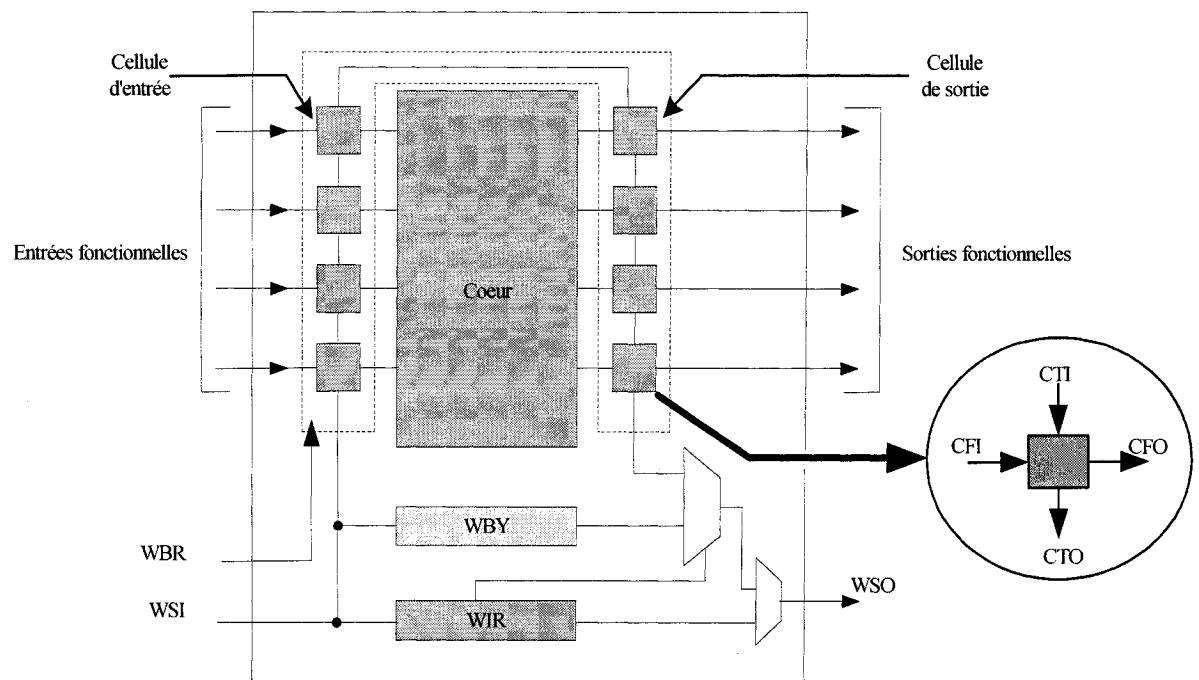


Figure 1.9: Architecture de l'enveloppe IEEE P1500.

L'enveloppe P1500 est l'ensemble des registres et des multiplexeurs insérés autour de l'IP. Cette interface de test offre l'accès aux plots d'entrée/sortie de l'IP à partir de quelques plots supplémentaires ajoutés au SOC. La figure 1.9 représente la structure de l'enveloppe P1500 qui est essentiellement composée de trois registres [20] [29]. Le premier est le registre d'instruction de l'enveloppe « Wrapper Instruction Register » (WIR). Dans ce registre, les codes opérations des instructions de test sont chargés en série et décodés pour configurer l'enveloppe P1500 dans le mode de fonctionnement approprié. Le deuxième est le registre à décalage de l'enveloppe « Wrapper Boundary Register » (WBR). Ce registre est formé par la concaténation en série de toutes les cellules de l'enveloppe P1500 associées aux signaux d'entrée/sortie du cœur de l'IP. En mode normal, WBR devient transparent pour ne pas altérer le fonctionnement normal des IP. En mode test, WBR agit comme un registre à décalage pour acheminer en série les valeurs de l'entrée sérielle de l'enveloppe « Wrapper Scan Input » (WSI) à appliquer aux signaux d'entrée de cœur de l'IP. Le registre WBR sert aussi à capturer les valeurs des signaux de sortie du cœur de l'IP. Par la suite, ces valeurs sont transmises en série vers un ATE via la sortie sérielle de l'enveloppe « Wrapper Scan Output » (WSO) pour analyse. Le troisième et dernier registre de l'enveloppe P1500 est le registre de court-circuit « Wrapper Bypass Register » (WBY). Ce registre est prévu pour court-circuiter WBR de l'IP non concerné par une procédure de test. WBY permet de raccourcir la longueur du chemin de balayage entre l'entrée WSI et la sortie WSO de l'enveloppe P1500. En plus de ces trois registres obligatoires, d'autres registres optionnels peuvent

être ajoutés à l'enveloppe P1500, tel que le registre d'identification utilisé par les fabricants pour identifier leurs produits.

1.7.2 Langage de test « Core Test Language » (CTL)

À cause de la diversité des sources de provenance des IP, le langage de test CTL [30] a été développé par le groupe P1500 pour uniformiser la manière dont les informations relatives aux tests des IP sont transférées du fournisseur à l'intégrateur. Parmi les informations que nous pouvons trouver dans un CTL, nous avons le type de DFT insérée dans l'IP, les différentes instructions de test et leurs codes opérations, les vecteurs de test à appliquer, les longueurs des chaînes de balayage et leurs nombres, les modèles de pannes utilisés, etc.

1.8 Conclusion

Le test est une opération fondamentale qui est réalisée à la sortie de tout procédé de fabrication du SOC afin de détecter les composants défectueux. Dans ce chapitre, nous avons exposé les raisons qui ont rendu le test du SOC très difficile à accomplir et les spécificités du test de ces systèmes comparativement aux SOB. Pour mieux comprendre le test des SOC, certaines notions de base ont été exposées telles que les défauts physiques, la génération des vecteurs de test et leur application. Nous avons également présenté la technique BIST et la norme IEEE P1500 qui ont été développées pour l'amélioration de la testabilité du SOC.

CHAPITRE 2

REVUE DES TECHNIQUES

DE TEST DES SOC

2.1 Introduction

Ces dernières années, plusieurs travaux de recherche ont été menés dans le milieu académique et industriel pour améliorer la testabilité des SOC. L'objectif de ces travaux est de développer de nouvelles architectures qui assurent une bonne qualité de test pour les SOC tout en diminuant le coût. Les différentes architectures de test proposées, peuvent être classées en deux groupes: 1) les architectures de test des IP; 2) les architectures de test des interconnexions entre les IP. Dans ce chapitre, nous exposons les avantages et les inconvénients de certaines de ces architectures afin de positionner notre travail par rapport à ce qui a été déjà réalisé dans le domaine du test des SOC.

2.2 Architectures de test des modules IP

Plusieurs architectures de test ont été proposées pour les IP du SOC. Parmi celles-ci, il y a celles qui utilisent des ATE externes pour tester les IP à partir de l'extérieur du SOC. Alors que d'autres architectures de test des IP ont été développées en se basant sur le test intégré présenté à la section 1.6. Les architectures de test intégré permettent de

contourner la problématique de l'accessibilité aux signaux d'entrée/sortie des IP à partir des plots des SOC.

2.2.1 Architectures du test externe pour les modules IP

Les architectures de test externe des IP sont des interfaces matérielles insérées autour de ces modules. En mode normal, ces interfaces deviennent transparentes pour ne pas gêner le fonctionnement normal des IP. En mode test, ces interfaces connectent les IP aux ATE externes à travers des TAM.

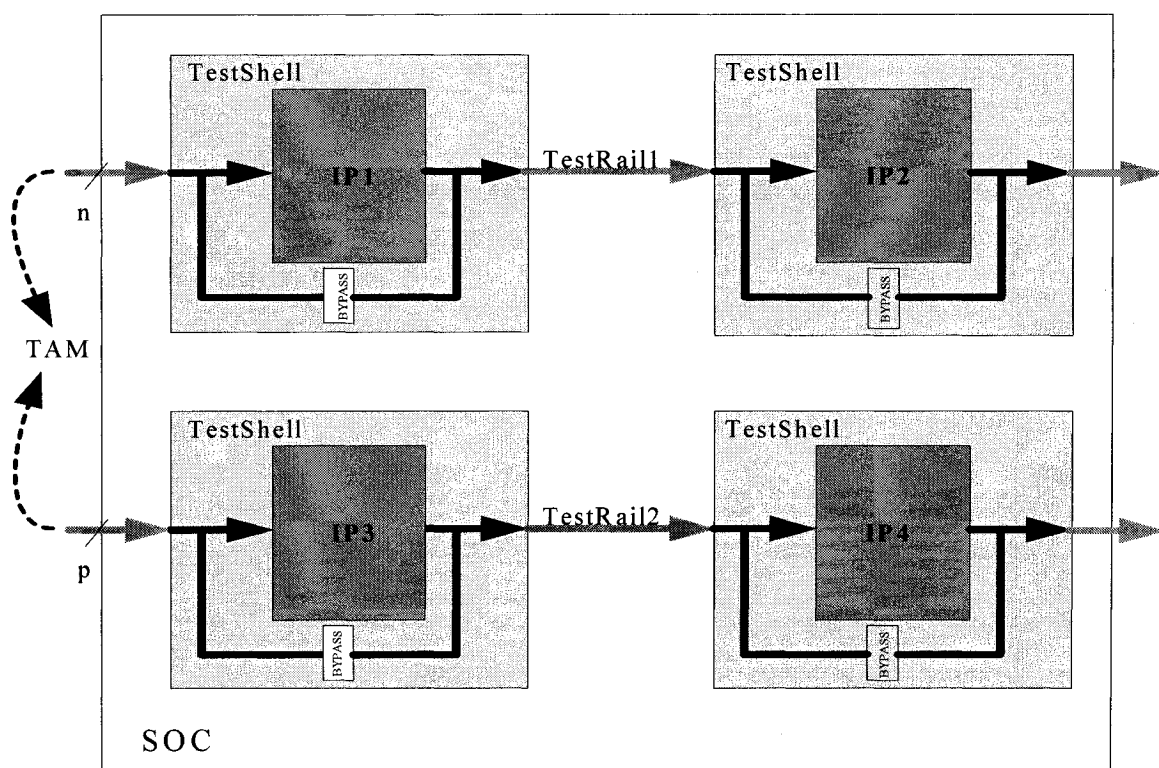


Figure 2.1: Architecture du bus de test TestRail [32].

Marinissen et Arendsen [32] ont proposé une architecture de test extensible appelée TestRail. Cette architecture simplifie l'accès aux signaux d'entrée/sortie des IP à partir de quelques plots supplémentaires ajoutés au SOC. La figure 2.1 présente le principe de cette architecture où les IP sont encapsulés dans des structures de test nommées TestShells. Le TestShell est l'interface qui connecte l'IP au TAM lorsque le SOC est en mode test et aux autres modules quand le SOC est en mode normal. Cette interface est composée de cellules associées aux signaux fonctionnels d'entrée/sortie de l'IP, de multiplexeurs et d'un registre de court-circuit. Dans cette solution de test que propose Marinissen et Arendsen, les IP peuvent être regroupés en petit nombre et leurs TestShells reliés en série. Les vecteurs de test sont appliqués aux signaux d'entrée des IP à partir d'ATE externe à travers les TAM et les Testshells. Les réponses de test sont transmises en série de l'IP à l'ATE pour analyse via les TestShells et les TAM. L'architecture TestRail permet de tester plusieurs IP en parallèle et de leur assurer une bonne qualité de test. Cependant l'usage du TestRail nécessite une étude préliminaire afin de déterminer l'impact de cette architecture sur la surface et les performances du SOC. Le coût relatif à cette architecture dépend de la complexité des IP à tester.

Benabdenbi et al. [7] ont proposé l'architecture de test CAS-BUS qui est présentée à la figure 2.2. Cette architecture offre un accès direct aux signaux d'entrée/sortie des IP du SOC à partir d'un nombre réduit de plots supplémentaires ajoutés au SOC. Les auteurs proposent d'associer à chaque IP un multiplexeur configurable appelé « Core Access Switch » (CAS). Selon le code opération de l'instruction chargée dans le registre de

configuration du multiplexeur, un CAS connecte au bus de test ou court-circuite les IP encapsulés dans des enveloppes. Cette architecture permet de tester plusieurs IP en parallèle, ce qui minimise le temps d'application des vecteurs de test. La bande passante du bus de test à insérer dans le SOC dépend du compromis à réaliser entre le temps d'application des vecteurs de test et la surface additionnelle. Cette contrainte peut limiter l'utilisation de l'architecture de test CAS-BUS aux IP qui ne nécessitent pas une grande bande passante.

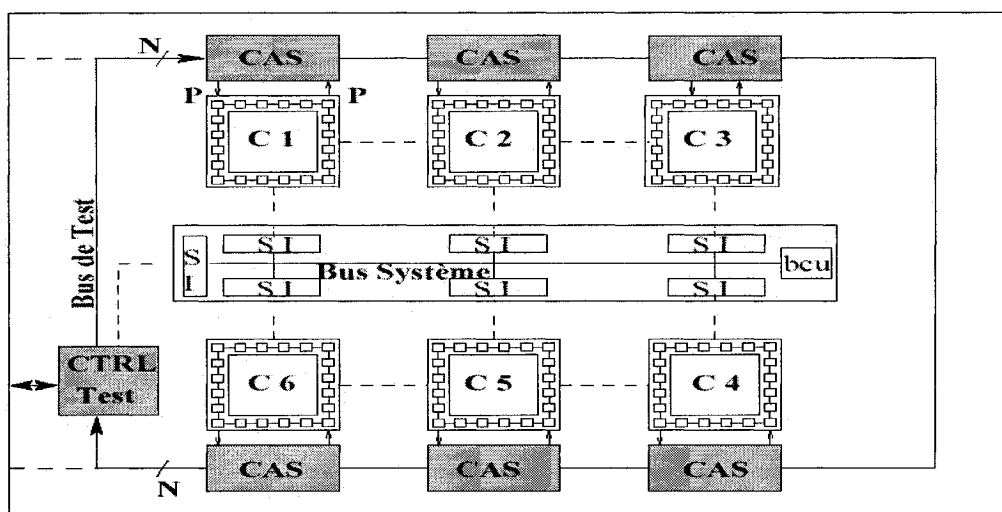


Figure 2.2: Architecture de test Cas-Bus [7].

2.2.2 Architectures de test intégré pour les modules IP

Pour réduire le coût de test des SOC, d'autres architectures de test basées sur la technique BIST ont été proposées pour les IP. Ces architectures permettent la génération et l'application des vecteurs de test ainsi que l'analyse des réponses de test à l'intérieur du SOC. Certaines de ces approches de test sont décrites ci-après.

2.2.2.1 Conversion des vecteurs de test aléatoires en vecteurs déterministes

Le BIST offre des structures de test peu coûteuses mais il n'assure pas une très bonne qualité de test à cause de la nature pseudo aléatoire de ses vecteurs qui ne détectent pas certaines pannes difficiles. Pour améliorer la qualité de test du BIST, certains chercheurs ont proposé de convertir les vecteurs aléatoires en vecteurs plus efficaces.

Akers [3] a proposé une architecture de test pour les IP du SOC qui est composée d'un compteur et d'un réseau de portes logiques XOR. Les vecteurs de test aléatoires générés par le compteur, sont modifiés par un réseau de portes logiques XOR en vecteurs déterministes. Selon l'auteur, cette architecture de test réduit la surface additionnelle par un facteur de 1,5 à 6 par rapport à celle qui utilise une mémoire pour stocker jusqu'à 65536 vecteurs déterministes. Les inconvénients de cette architecture de test sont l'augmentation de la surface qui est proportionnelle à la taille du réseau de portes XOR ainsi que les dégradations des délais internes.

Touba [43] a proposé une architecture de test qui consiste à insérer de la circuiterie combinatoire entre le LFSR et les entrées de l'IP. Le rôle de cette circuiterie est de transformer les vecteurs de test aléatoires qui ne détectent aucune panne en vecteurs efficaces. Selon Touba, cette architecture nécessite peu de portes logiques à insérer entre le LFSR et l'IP comparativement à d'autres architectures de test qui proposent la même approche. L'architecture proposée par Touba n'est pas extensible, ce qui nous conduit à analyser chacune des séquences de test originales des IP pour déterminer la circuiterie

combinatoire à insérer entre le LFSR et l'IP. Le temps d'analyse de ces séquences de test risque d'être long pour les IP complexes.

Kagaris et al. [23] ont proposé une nouvelle architecture de test qui génère des vecteurs déterministes à faible coût. L'idée est de générer une séquence de test déterministe pour chacun des IP du SOC et de l'analyser pour identifier les colonnes constantes et celles qui sont semblables ou complémentaires. Les signaux d'entrée de l'IP sous test qui correspondent aux colonnes constantes, sont reliés à V_{cc} ou à V_{ss} . Les signaux d'entrée de l'IP qui correspondent aux colonnes identiques ou complémentaires, sont connectés aux même sorties du compteur qui génère le reste des bits des vecteurs de test. Cette architecture réduit la surface additionnelle, mais le temps de génération et d'analyse des vecteurs de test déterministes de chaque IP peut être important.

2.2.2.2 BIST associé aux chaînes de balayage de l'IP

L'utilisation de vecteurs déterministes pour tester les IP séquentiels ne garantit pas forcément une bonne qualité de test. Ceci est dû à certains nœuds internes de l'IP qui sont difficiles à contrôler et à observer à partir des signaux d'entrée/sortie primaires de l'IP. Pour résoudre ce problème, Kiefer et Wunderlich [25] ont proposé l'architecture de test nommée BIST déterministe. Cette architecture associée à une multitude de chaînes de balayage, est présentée à la figure 2.3. L'architecture de test proposée par les deux auteurs, est composée d'un LFSR, d'un contrôleur de test, d'un module de modification

des vecteurs de test aléatoires en vecteurs déterministes nommé Modificateur de Séquence Logique (MSL) et d'un ensemble de portes XOR. Dans un premier temps, les vecteurs de test générés par LFSR sont convertis en vecteurs déterministes par un MSL. Ensuite, ces vecteurs déterministes sont décalés dans les chaînes de balayage de l'IP à travers les portes XOR. L'inconvénient que nous pouvons rencontrer avec cette architecture est le temps d'application des vecteurs de test. Ce temps d'application dépend de la longueur des chaînes de balayage et du nombre de vecteurs de test à appliquer.

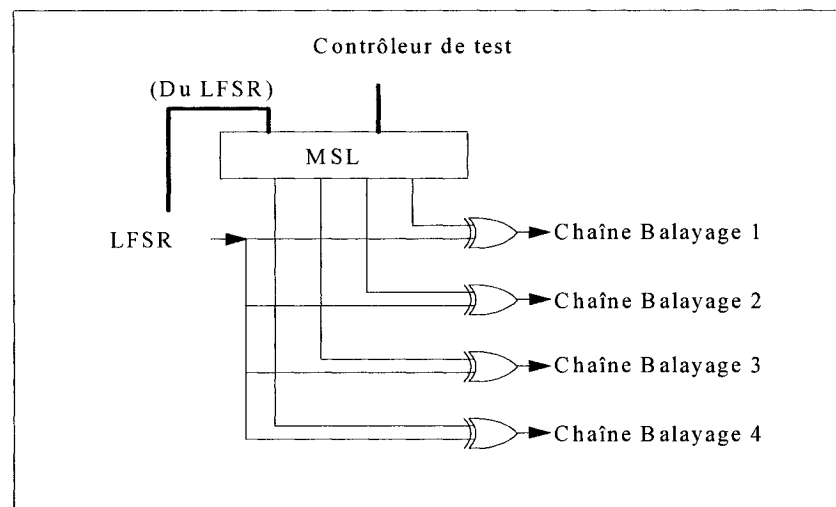


Figure 2.3: Exemple de la logique de génération de séquence de test [25].

2.2.2.3 Conversion des vecteurs de test déterministes en vecteurs aléatoires

Les vecteurs déterministes assurent une bonne qualité de test mais leur stockage dans le SOC nécessite une mémoire de taille importante, ce qui augmente la surface du SOC.

Comme solution à cette contrainte, Kunzmann [27] a proposé une architecture de test qui utilise des vecteurs déterministes pour détecter les pannes difficiles et des vecteurs aléatoires pour détecter les pannes faciles. Les vecteurs aléatoires sont générés en perturbant les vecteurs de test déterministes (inversion, rotation, etc.). Avec cette approche de test, l'auteur a réussi à réduire de 90% la longueur de la séquence de test déterministe qui devait être emmagasinée dans la mémoire insérée dans le SOC. Malgré la diminution considérable du nombre de vecteurs de test, la taille de la mémoire peut être importante pour les SOC complexes, ce qui demande une surface additionnelle importante.

2.3 Architectures de test des interconnexions entre les modules IP

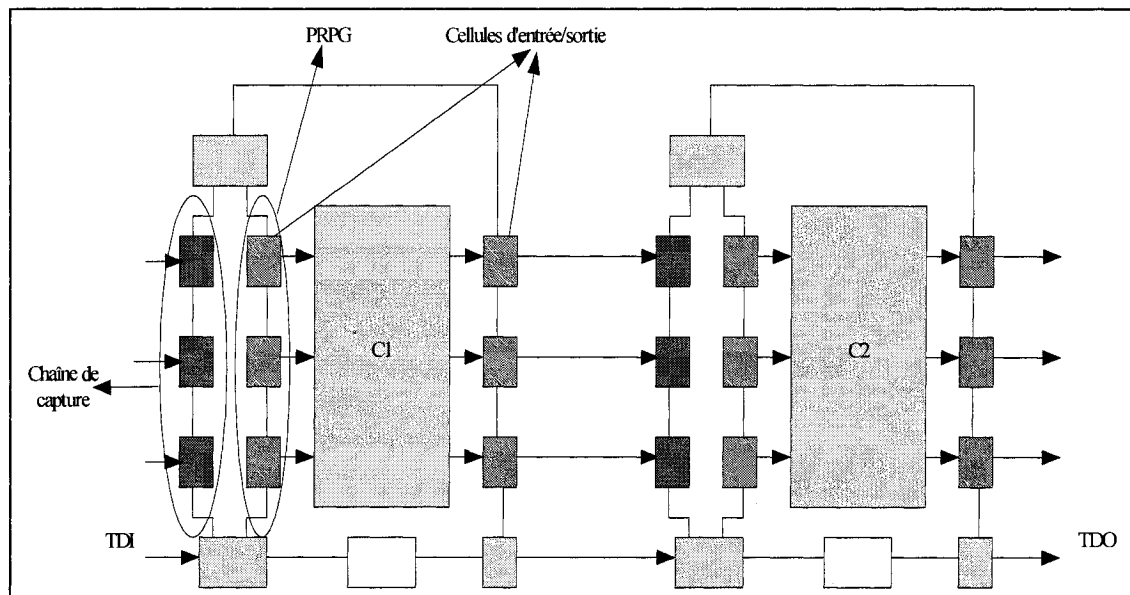


Figure 2.4: Combinaison du BIST et du Boundary Scan pour le test des interconnexions [5].

Les techniques de test développées pour le test des interconnexions entre les circuits intégrés (CI) des SOB, ont été réutilisées pour le test des interconnexions entre les IP du SOC. Parmi ces architectures, nous pouvons mentionner la norme IEEE 1149.1 [19] qui est connue sous le nom du Boundary Scan. Cette norme propose d'encapsuler chacun des CI du SOB dans une enveloppe dans laquelle, les signaux d'entrée/sortie du CI sont associés aux cellules. En mode test, les cellules du Boundary Scan forment un registre à décalage qui relie l'entrée sérieuse à la sortie sérieuse de l'enveloppe. À travers ce registre à décalage, les données de test à appliquer aux signaux d'entrée du CI à partir d'un ATE externe sont transférées en série. Le registre à décalage sert aussi à l'acheminement en série des réponses de test du CI vers un ATE externe pour analyse. La technique Boundary Scan ajoute peu de plots supplémentaires au SOB, mais sa bande passante est limitée à un bit. La limitation de la bande passante du Boundary Scan prolonge le temps d'application des vecteurs de test. Pour réduire le temps d'application des vecteurs de test, d'autres architectures de test basées sur le BIST ont été développées pour les interconnexions.

Ashok et Harold [5] ont proposé pour le test des interconnexions l'architecture représentée à la figure 2.4, qui combine le BIST avec le Boundary Scan. Ces deux auteurs proposent de doubler le nombre de cellules d'entrée de l'enveloppe Boundary Scan pour former deux chaînes à décalage connectées en parallèle. Les cellules associées aux signaux d'entrée du circuit fonctionnent comme un générateur de vecteurs de test aléatoires « Pseudo Random Pattern Generator » (PRPG). Par contre, les cellules

connectées en parallèles au PRPG forment un registre parallèle/série, appelé chaîne de capture. Ce registre parallèle/série capture les réponses des interconnexions sous test et les transmet en série vers un ATE externe. À titre d'exemple, pour le test des interconnexions entre les circuits C1 et C2 de la figure 2.4, les vecteurs sont générés par le PRPG formé par les cellules associées aux signaux d'entrée du circuit C1. Les vecteurs de test sont appliqués en parallèle aux interconnexions à travers C1. Les réponses de test des interconnexions sont récupérées par la chaîne de capture du circuit C2 et transférées en série vers un ATE externe pour analyse. Cette architecture de test exige des CI transparents pour permettre le transfert des données de test sans perte d'information.

Pendurkar [36] a présenté l'architecture du BIST distribué pour le test des interconnexions entre les CI d'un SOB. Cette solution propose d'ajouter du matériel au niveau des cellules du Boundary Scan associées aux signaux d'entrée/sortie des CI. En mode test des interconnexions, les cellules du Boundary Scan connectées aux signaux de sortie des CI, forment un LFSR qui génère les vecteurs de test. Par contre, les cellules du Boundary Scan reliées aux signaux de sortie des CI, agissent en MISR pour compacter les réponses de test. L'implantation de cette architecture de test nécessite une surface additionnelle importante et cause des dégradations des délais internes de propagation.

2.4 Conclusion

Le test des SOC peut être accompli par des approches de test externe ou par des approches de test intégré. Les approches du test externe requièrent des ATPG pour générer les vecteurs de test et des ATE pour leur application. Ces approches nécessitent aussi du matériel dédié au test à insérer dans le SOC pour rendre les signaux d'entrée/sortie des IP accessibles à partir de l'extérieur. Les architectures de test externe assurent une bonne qualité de test mais le coût de test est généralement élevé. Le coût dépend du surcoût en surface, du prix des ATE et du temps de génération des vecteurs de test. Les approches de test intégré offrent des structures peu coûteuses pour le test des IP et des interconnexions. Toutefois, l'insertion de ces architectures dans le SOC engendre une augmentation de la surface et la dégradation des délais internes de propagation. Malgré cela, les architectures de test intégré représentent une bonne solution pour les SOC, parce qu'elles réduisent le temps d'application des vecteurs de test et ne nécessitent pas d'outils externes pour générer les vecteurs de test.

CHAPITRE 3

ARCHITECTURE DE TEST PROPOSÉE

3.1 Introduction

Il est difficile d'avoir une seule architecture de test applicable à tous les SOC, parce que chacun de ces systèmes a ses propres contraintes de conception. Il est donc indispensable de diversifier les architectures afin de permettre aux intégrateurs de réaliser des compromis entre la qualité et le coût de test. Dans cette optique, nous avons développé une nouvelle architecture de test, que nous présentons dans ce chapitre. Cette architecture peut être utilisée pour vérifier les IP et les interconnexions du SOC avec ou sans la fonctionnalité BIST. Cette approche de test permet de réduire le nombre d'architectures à insérer dans le SOC, ce qui permet de minimiser la surface additionnelle et la dégradation des performances du SOC.

Dans ce chapitre, nous commençons notre présentation par un aperçu global sur l'architecture de test proposée. Nous exposons par la suite en détail la structure matérielle de la nouvelle architecture de test. À la fin du chapitre, nous décrivons les différentes instructions de test, que l'architecture proposée implémente pour le test des IP et des interconnexions du SOC avec ou sans fonctionnalité BIST.

3.2 Présentation de la nouvelle architecture de test

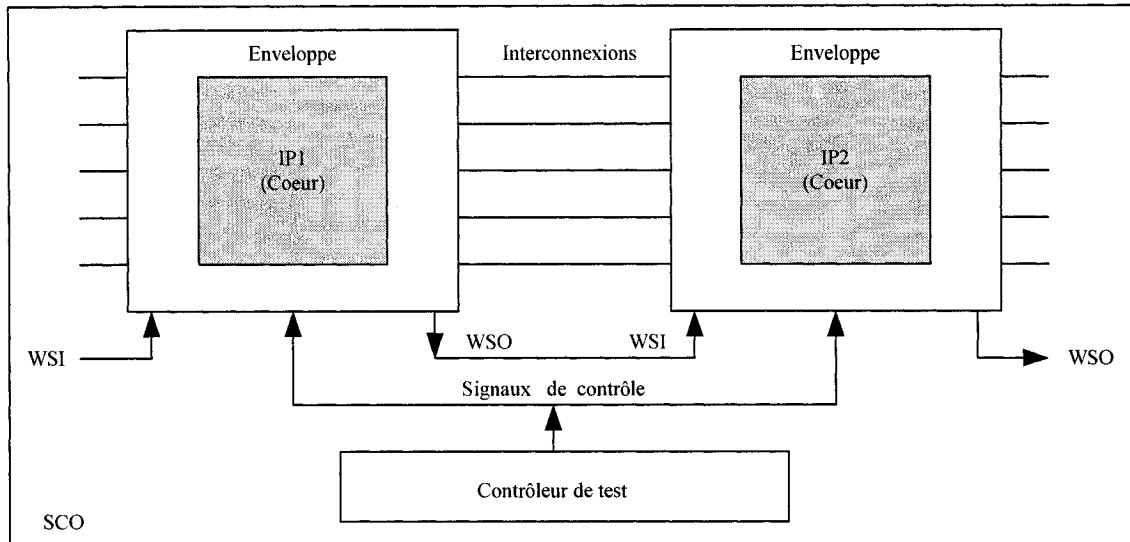


Figure 3.1: Architecture de test proposée.

La figure 3.1 illustre le schéma de principe de la nouvelle architecture proposée pour le test des IP et des interconnexions. Cette architecture est composée d'une enveloppe dans laquelle, les IP sont encapsulés et d'un contrôleur qui gère les opérations de test. En mode normal, l'enveloppe agit en tant qu'interface qui relie l'IP aux autres modules du SOC. En mode test, l'enveloppe agit en structure de test qui permet de vérifier les IP et les interconnexions avec et sans la fonctionnalité BIST.

À titre d'exemple, lors du test du cœur de l'IP1 et de celui de l'IP2 ainsi que des interconnexions entre ces deux IP sans la fonctionnalité BIST, l'enveloppe de l'IP1 et celui de l'IP2 agissent en tant qu'enveloppe P1500 tel que présentée à la section 1.8. Dans ce mode de test, les enveloppes sont connectées en série et forment un registre à

décalage qui relie le signal d'entrée de l'enveloppe « Wrapper Scan In » (WSI) de l'IP1 au signal de sortie de l'enveloppe « Wrapper Scan Out » (WSO) de l'IP2. Ce registre sert à transférer en série les valeurs de WSI à appliquer aux signaux d'entrée du cœur de l'IP1 et à ceux du cœur de l'IP2 ainsi qu'aux interconnexions entre les IP. Ce registre est également utilisé pour récupérer et transmettre en série les réponses des IP et des interconnexions vers un testeur externe via WSO pour analyse.

Pour tester le cœur de l'IP1 et celui de l'IP2 ainsi que les interconnexions entre ces deux IP avec la fonctionnalité BIST, les enveloppes des IP sont configurées en structures BIST dont le principe a été présenté précédemment à la section 1.7. Lors du test du cœur de l'IP, les cellules de l'enveloppe connectées aux entrées du cœur, agissent en LFSR pour générer et appliquer les vecteurs de test. Les réponses du cœur de l'IP sont compactées par le MISR formé par les cellules de l'enveloppe connectées aux sorties du cœur de l'IP. Par contre, lors du test des interconnexions entre l'IP1 et l'IP2, les vecteurs de test à appliquer aux interconnexions sont générés par les cellules de sortie de l'enveloppe de l'IP1 qui forment un LFSR. Les réponses de test des interconnexions sont compactées par les cellules d'entrée de l'enveloppe de l'IP2 qui agissent en MISR.

3.3 Architecture de la nouvelle enveloppe

La figure 3.2 représente l'architecture de l'enveloppe proposée, que nous définissons comme une enveloppe P1500 dans laquelle la fonctionnalité BIST est incorporée [28]. La différence entre l'enveloppe proposée et celle que propose la norme IEEE P1500

Le deuxième registre obligatoire est le registre à décalage de l'enveloppe « Wrapper Boundary Register » (WBR). Ce registre est inséré entre les signaux d'entrée/sortie du cœur de l'IP et les plots d'entrée/sortie de l'IP. En mode normal, les cellules de WBR relient en parallèle les signaux d'entrée/sortie du cœur de l'IP aux plots d'entrée/sortie de l'IP. En mode test sans BIST, le WBR est configuré en registre à décalage entre WSI et WSO. Ce registre offre un accès aux signaux d'entrée/sortie du cœur de l'IP à partir de l'extérieur du SOC. En mode test avec BIST, le WBR agit en structure BIST pour assurer le test des cœurs des IP et des interconnexions entre les IP à l'interne du SOC.

Le dernier registre obligatoire de l'enveloppe proposée est le registre de court-circuit « Wrapper Bypass Register » (WBY). Ce registre sert à court-circuiter WBR de l'IP qui ne participe pas dans une procédure de test. Ceci raccourcit la longueur du chemin de balayage, ce qui permet de réduire le temps d'application des vecteurs de test aux autres IP et interconnexions du SOC.

Avant de décrire l'architecture et le fonctionnement des registres énumérés ci-dessus, nous présentons en premier lieu l'architecture et le fonctionnement des différentes cellules de la nouvelle architecture de test.

3.4 Cellules de l'enveloppe

Dans cette section, nous présentons les cellules à deux états, à trois états et bidirectionnelles, que nous avons développé pour l'enveloppe proposée. Ces cellules ont

été conçues en modifiant l'architecture interne de la cellule « Boundary Scan Cell » (BSC) de la norme IEEE 1149.1 [19]. En mode normal, les cellules deviennent transparentes pour ne pas nuire au fonctionnement normal du cœur de l'IP. En mode test sans fonctionnalité BIST du cœur de l'IP et des interconnexions, les cellules assurent les opérations *Capture*, *Shift* et *Update* qui sont définies par la norme IEEE P1500 [20].

Capture : c'est l'opération qui permet de copier en parallèle les données externes dans les cellules de l'enveloppe associées aux entrées du cœur de l'IP. Cette opération permet également de copier les réponses du cœur de l'IP dans les cellules de l'enveloppe connectées aux signaux de sortie du cœur de l'IP.

Shift (décalage) : c'est l'opération de transfert en série des valeurs de WSI de l'enveloppe à appliquer aux signaux d'entrée du cœur de l'IP et aux interconnexions. Le shift est aussi l'opération qui permet de transmettre en série les réponses de test du cœur de l'IP et des interconnexions vers un analyseur externe à travers la sortie WSO de l'enveloppe.

Update (mise à jour) : c'est l'opération qui permet d'appliquer en parallèle les données de test qui ont été déjà chargées dans les cellules d'entrée de l'enveloppe aux entrées du cœur de l'IP. Cette opération permet aussi l'application en parallèle des données de test qui ont été chargées en série dans les cellules de sortie de l'enveloppe aux plots de sortie de l'IP lors du test des interconnexions.

En plus des opérations *Capture*, *Shift* et *Update*, que les cellules de l'enveloppe accomplissent conformément aux spécifications de la norme IEEE P1500, ces cellules sont conçues pour être configurables en structure BIST. Lors du test du cœur de l'IP en mode BIST, les cellules de l'enveloppe associées aux entrées du cœur de l'IP, agissent en tant qu'un LFSR et les cellules de l'enveloppe connectées aux sorties du cœur de l'IP, forment un MISR. Lors du test des interconnexions en mode BIST, les cellules associées aux entrées du cœur de l'IP sont configurées en MISR et celles reliées aux sorties du cœur de l'IP s'activent en tant qu'un LFSR.

3.4.1 Cellule à deux états de l'enveloppe

Pour les entrées et les sorties à deux états du cœur de l'IP, nous avons conçu la cellule « Wrapper Boundary Cell » (WBC) qui est représentée à la figure 3.3. Cette cellule est composée de trois multiplexeurs, de deux portes logiques XOR et AND et de deux bascules « Serial Register » (SR) et « Parallel Register » (PR). La cellule WBC a deux signaux fonctionnels, l'un est l'entrée fonctionnelle de la cellule « cell fonctionnel in » (*cfi*) et l'autre est la sortie fonctionnelle de la cellule « cell fonctionnel out » (*cfo*). Ces signaux relient les entrées et les sorties du cœur de l'IP aux plots d'entrée/sortie de l'IP à travers le multiplexeur M1. La cellule WBC a aussi deux signaux de test qui sont : l'entrée de test de la cellule « cell test in » (*cti*) et la sortie de test de la cellule « cell test out » (*cto*). Ces signaux de test connectent en série les bascules SR des différentes cellules de l'enveloppe pour former un registre à décalage.

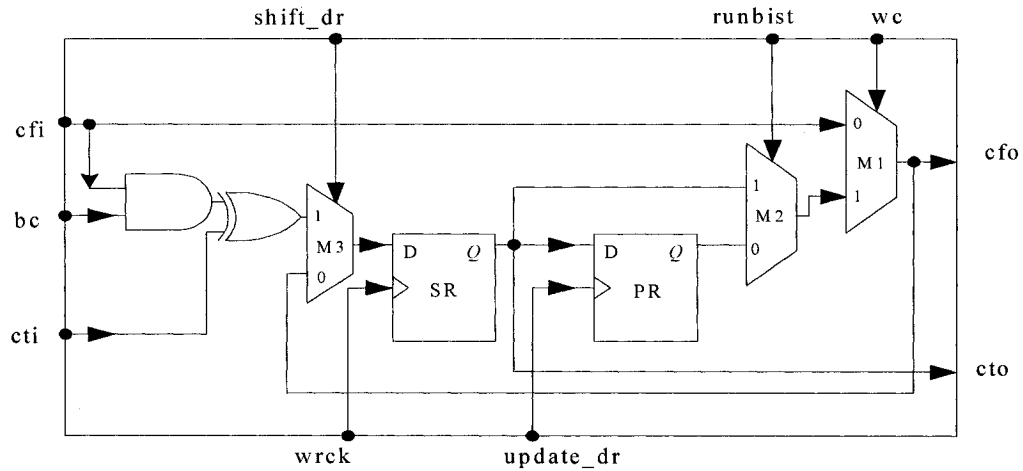


Figure 3.3: Cellule à deux états WBC.

La cellule WBC est contrôlée par un ensemble de signaux de contrôle. Les signaux de contrôle « wrapper cell » (*wc*), « BIST cell » (*bc*) et *runbist* sont générés par le décodeur du registre WIR qui est présenté à la section 3.5. Le signal *wc* permet au multiplexeur M1 de sélectionner la valeur à transmettre à la sortie *cfo* de WBC. Cette valeur peut être celle de l'entrée *cfi* ou celle de la sortie du multiplexeur M2. Le signal *runbist* permet de choisir la valeur à appliquer au signal de sortie *cfo* à travers le multiplexeur M2 parmi celles des bascules SR et PR. Le signal *bc* détermine laquelle des valeurs *cti* et *cti* XOR *cfi* qui peut être chargée dans la bascule SR sur un front montant de l'horloge de test « wrapper clock » *wrck*. Les autres signaux de contrôle *shift_dr* et *update_dr* sont générés par le contrôleur de test que nous présentons à la section 3.6. Le signal *shift_dr* contrôle le multiplexeur M3 pour sélectionner soit la valeur de la sortie *cfo* de WBC soit celle de la sortie de la porte XOR à charger dans la bascule SR sur un front montant de *wrck*. Le signal *update_dr* permet de transférer le contenu de la bascule SR dans la bascule PR de la cellule.

3.4.2 Modes de fonctionnement de WBC

Les valeurs des signaux de contrôle énumérés à la section précédente déterminent le mode de fonctionnement dans lequel la cellule WBC peut être configurée. Le tableau 3.1 présente ces modes de fonctionnement et les valeurs des signaux de contrôle qui leur correspondent. La première colonne du tableau présente les différents modes de configuration de WBC. Les colonnes 2, 3 et 4 indiquent les valeurs des signaux de contrôle *wc*, *bc* et *runbist* générés par WIR pour chacun des modes de fonctionnement de la cellule. Les colonnes 5, 6 et 7 donnent les valeurs des signaux de contrôle qui sont générés aux différents états du contrôleur de test pour chaque mode de fonctionnement.

Tableau 3.1: Différents modes de configuration de WBC.

Modes	Registre WIR			Contrôleur de test		
	<i>wc</i>	<i>bc</i>	<i>runbist</i>	<i>shift_dr</i>	<i>update_dr</i>	états
Normal	0	X	X	X	X	Test_reset
Test sans BIST	1	0	0	0	0	Capture_DR
	1	0	0	1	0	Shift_DR
	1	0	0	0	1	Update_DR
Test avec BIST	1	0	1	0	0	Capture_DR
(LFSR)	1	0	1	1	0	Shift_DR
	1	0	1	0	1	Update_DR
Test avec BIST	1	1	0	0	0	Capture_DR
(MISR)	1	1	0	1	0	Shift_DR
	1	1	0	0	1	Update_DR

En mode normal, *wc* est à 0, les bascules SR et PR de WBC sont court-circuitées et le signal d'entrée *cfi* est connecté au signal de sortie *cfo* à travers le multiplexeur M1. Dans ce mode de fonctionnement les signaux d'entrée/sortie du cœur de l'IP sont directement connectés aux plots d'entrée/sortie de l'IP.

En mode test sans fonctionnalité BIST, WBC permet d'acheminer en série les valeurs de test externes à appliquer aux entrées du cœur de l'IP et aux plots de sortie de l'IP. Dans ce mode de test, le signal *shit_dr* est à 1 et *bc* est à 0. Ces signaux permettent de charger la valeur de l'entrée *cti* dans la bascule SR de la cellule sur un front montant de l'horloge *wrck* à travers le multiplexeur M3. Lorsque le signal *runbist* est à 0 et *update_dr* est à 1, la valeur de la bascule SR est transférée dans la bascule PR à travers le multiplexeur M2. Lorsque *wc* est à 1, la valeur de la bascule PR est appliquée à la sortie *cfo* à travers le multiplexeur M1.

En mode test avec fonctionnalité BIST, WBC est utilisée pour former un LFSR afin de générer les vecteurs de test et de les appliquer aux entrées du cœur de l'IP et aux interconnexions ainsi qu'un MISR pour calculer la signature finale du test. Dans ce mode de test, *wc* et *shift_dr* sont à 1 mais *bc* et *runbist* peuvent être à 0 ou à 1 dépendamment de la fonction que réalise WBC. Dans le cas où *bc* est à 0 et *runbist* est à 1, la cellule WBC de l'enveloppe assure la fonction LFSR. Dans cette configuration, la valeur de l'entrée *cti* est chargée dans SR sur un front montant de l'horloge *wrck* à travers le multiplexeur M3. La valeur de SR est appliquée à la sortie *cfo* de WBC via les multiplexeurs M2 et M1. Lorsque *bc* est à 1 et *runbist* est à 0, WBC assure la fonction MISR pour compacter les réponses de test afin de calculer la signature finale du test. Dans cette configuration, la valeur de *cti* XOR *cfi* est chargée dans SR sur un front montant de l'horloge *wrck* à travers le multiplexeur M3. À la fin du compactage, la

valeur finale de SR est transférée dans PR quand *update_dr* est à 1 pour la protéger de toute contamination.

3.4.3 Cellule à trois états de l'enveloppe

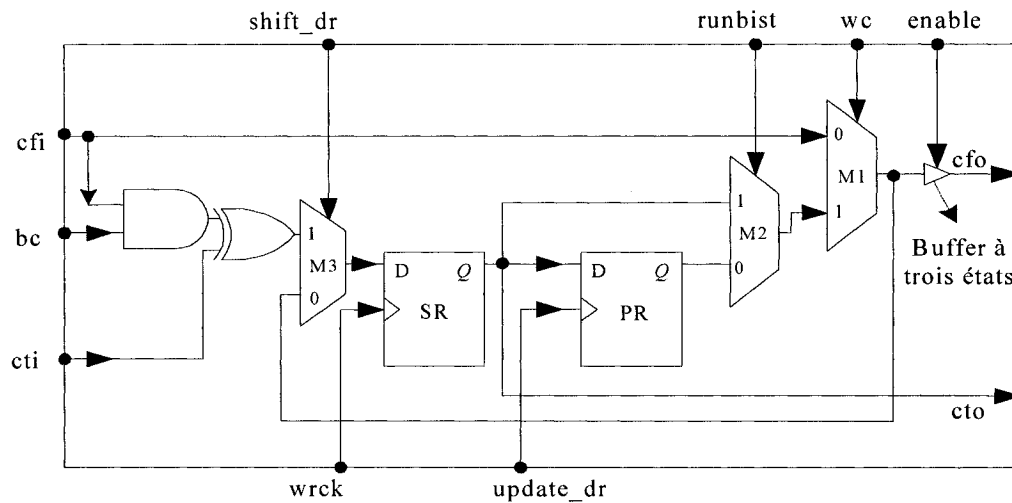


Figure 3.4: Cellule de sortie à 3 états (WBOC3).

Pour les signaux de sortie à trois états du cœur de l'IP, nous avons conçu la cellule « Three State Wrapper Boundary Cell » (WBC3) qui est représentée ci-dessus à la figure 3.4. Cette cellule est constituée d'une cellule de sortie à deux états WBC reliée en série avec un tampon à trois états. Le fonctionnement de WBC3 est identique à celui de WBC sauf pour la sortie *cfo* dont la valeur dépend du tampon à trois états commandé par le signal *enable*. En mode normal de l'IP, la valeur du signal *enable* est imposée par le cœur de l'IP. En mode test avec ou sans la fonctionnalité BIST du cœur de l'IP et des interconnexions, la valeur du signal *enable* est imposée par l'utilisateur. Dans les deux

modes de fonctionnement, la valeur à appliquer à l'entrée *enable* de WBC3 transite par la cellule de contrôle dont l'architecture est présentée à la figure 3.5.

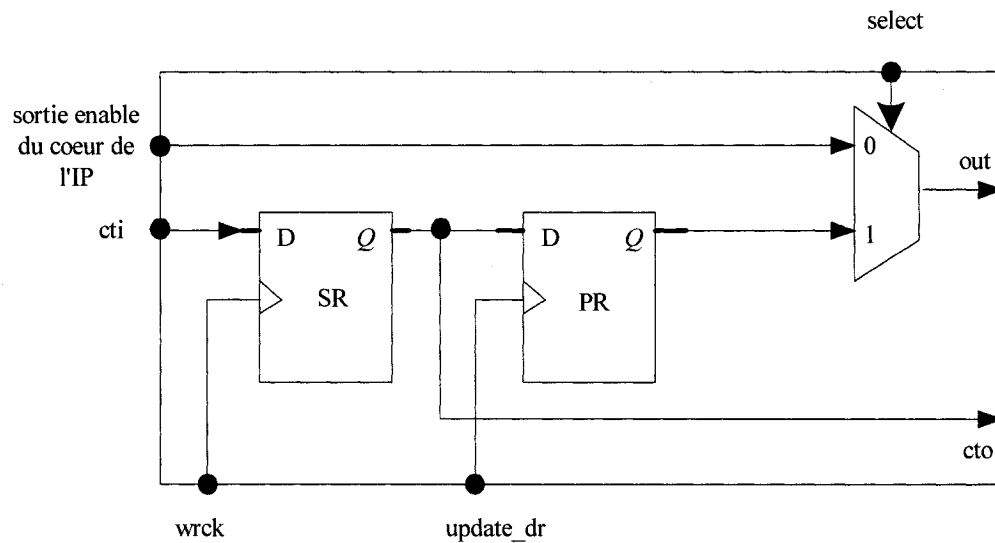


Figure 3.5: Cellule de contrôle de la sortie de la cellule WBC3.

La cellule de contrôle est composée de deux bascules SR et PR et d'un multiplexeur. La bascule SR permet de charger en série la valeur à imposer au signal *enable* à partir d'une source externe. La bascule PR est l'étage parallèle à travers lequel, la valeur de SR est transmise au signal de sortie *out* de la cellule de contrôle lorsque le signal *update_dr* est actif. Le multiplexeur de la cellule de contrôle est commandé par le signal *select* généré par le registre WIR de l'enveloppe. Ce multiplexeur permet de sélectionner soit la valeur de la sortie enable du cœur de l'IP soit celle de la bascule SR chargée à partir d'une source externe.

3.4.4 Cellule bidirectionnelle de l'enveloppe

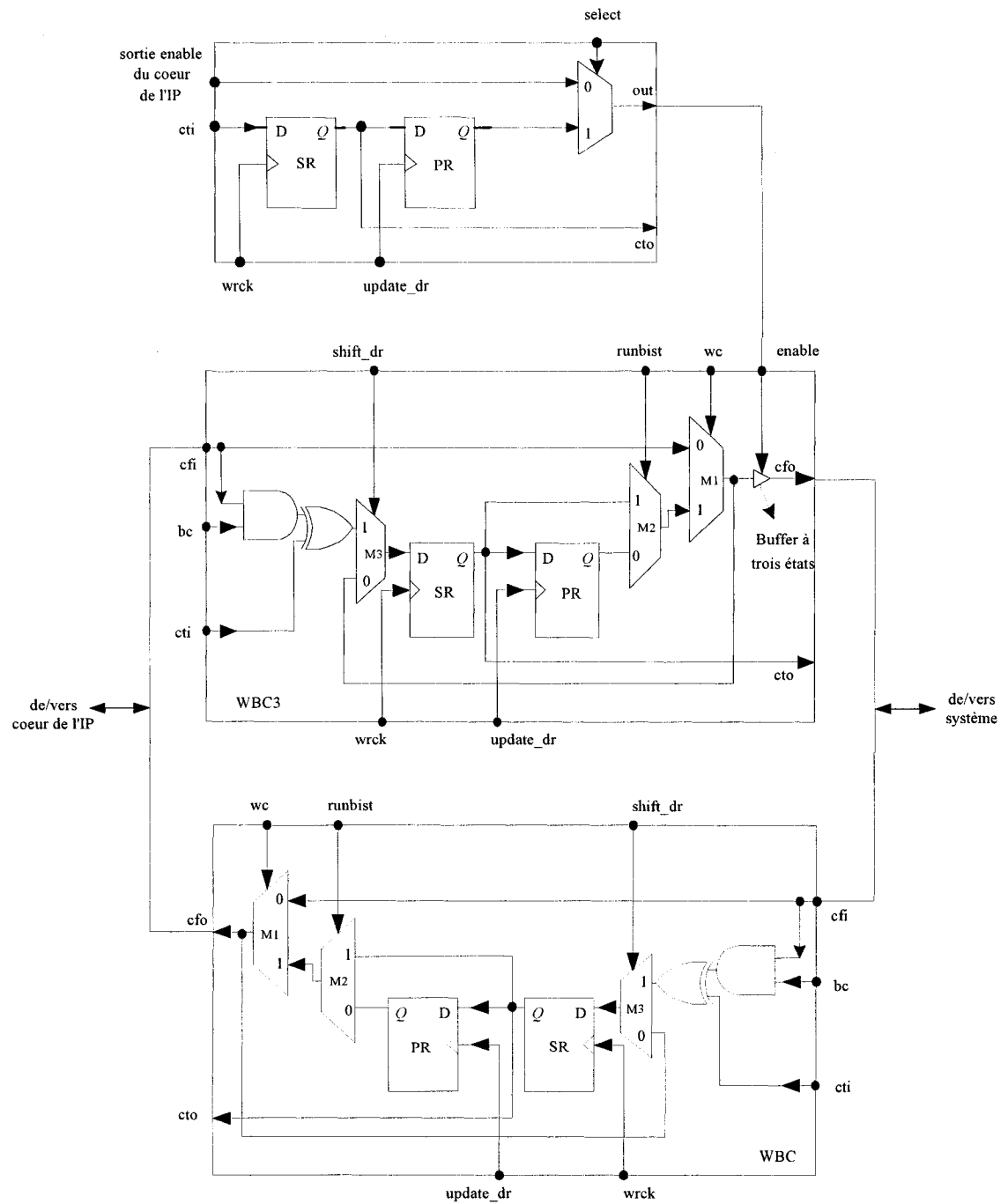


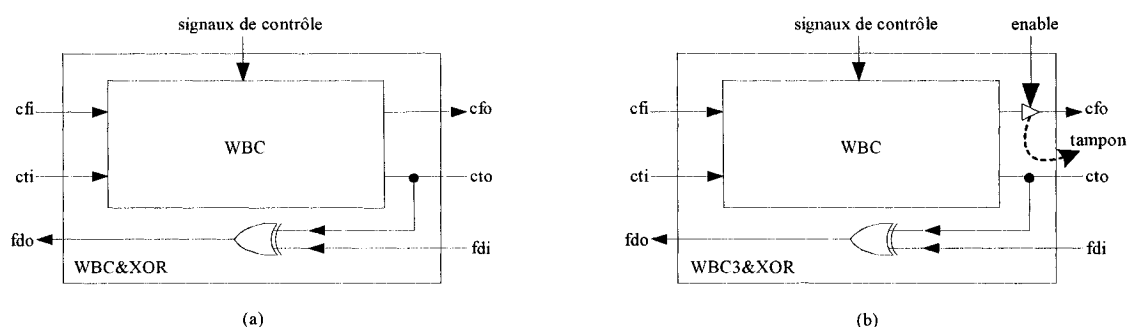
Figure 3.6: Cellule bidirectionnelle.

La figure 3.6 montre l'architecture de la cellule bidirectionnelle de l'enveloppe, que nous associons aux signaux bidirectionnels du cœur de l'IP. Cette cellule est composée d'une cellule à deux états WBC et d'une cellule de sortie à trois états WBC3 connectées en parallèle et d'une cellule de contrôle. Lorsque le signal *enable* est à 0, la cellule bidirectionnelle agit en tant que cellule d'entrée à deux états WBC à travers laquelle, les données externes sont appliquées aux entrées du cœur de l'IP. Lorsque le signal *enable* est à 1, la cellule bidirectionnelle agit en tant que cellule de sortie à trois états WBC3 via laquelle, les données du cœur de l'IP sont transmises vers les plots de sortie de l'IP.

3.4.5 Autres variantes des cellules de l'enveloppe

Pour former des LFSR et des MISR de différents polynômes caractéristiques avec les cellules de l'enveloppe lorsque celui-ci est configuré en structure BIST, il faut insérer des portes XOR à certains de ses endroits. Pour faciliter cette insertion, nous avons développé d'autres variantes des cellules à deux états, à trois états et bidirectionnelles. Pour les entrées et les sorties à deux états du cœur de l'IP, nous avons conçu la cellule WBC&XOR. Pour les sorties à trois états du cœur de l'IP, nous avons développé la cellule WBC3&XOR. Les schémas de principe des cellules WBC&XOR et WBC3&XOR sont présentés à la figure 3.7. WBC&XOR est composée de WBC et d'une porte logique XOR et WBC3&XOR est formée de WBC3 et d'une porte logique XOR. Dans chacune des deux cellules, l'une des deux entrées de la porte XOR est reliée à la sortie *cto* et l'autre entrée est connectée à l'entrée « *feedback_in* » (*fdi*) de la

cellule. La sortie du XOR est raccordée à la sortie « feedback_out » (*fdo*) de la cellule. Pour les signaux bidirectionnelles du cœur de l'IP, d'autres variantes de cellules bidirectionnelles peuvent être formées en utilisant les cellules WBC, WBC3, WBC&XOR et WBC3&XOR. Ces variantes sont représentées à l'annexe A.



3.5 Registres de l'enveloppe

Dans cette section, nous présentons l'architecture et le fonctionnement des trois registres obligatoires de l'enveloppe proposée, que nous avons introduit à la section 3.3. Ces registres sont : WIR, WBR et WBY.

3.5.1 Registre d'instruction WIR

Le registre instruction WIR est un registre série/parallèle, qui génère les signaux de contrôles nécessaires au fonctionnement des registres et des multiplexeurs de l'enveloppe proposée. Ce registre nous permet de charger en série une nouvelle instruction de test sans désactiver l'instruction précédente qui est déjà active dans

l'enveloppe. Pour la réalisation de ce registre, nous avons réutilisé le registre d'instruction « Instruction Register » (IR) du Boundary Scan. En effet, le fonctionnement de l'enveloppe proposée est presque identique à celui du Boundary Scan. La figure 3.8 montre l'architecture du registre IR qui est composé d'un étage sériel, d'un étage parallèle et d'un décodeur. L'étage sériel est le registre série/parallèle dans lequel, le code opération de l'instruction de test à appliquer est chargé en série via l'entrée WSI de l'enveloppe. L'étage parallèle dont la longueur est égale à celle de l'étage sériel, est un registre parallèle/parallèle. Cet étage sert à appliquer et à maintenir le contenu de l'étage sériel aux entrées du décodeur. Le décodeur est le dernier étage d'IR, qui sert à générer les signaux de contrôle utiles pour le fonctionnement de l'enveloppe.

Chaque cellule du registre IR est formée de deux bascules D la « Serial Register » (SR) et la « Parallel Register » (PR). La bascule SR est précédée par le multiplexeur m qui est commandé par le signal de contrôle *shift_ir* généré par le contrôleur de test. Le multiplexeur m sélectionne la valeur à charger dans SR sur un front montant de l'horloge de test *wrck_ir* qui est généré par le contrôleur de test. Cette valeur peut être celle de la cellule précédente (WSI pour la première cellule) ou celle de l'entrée *capture_données*. Cette entrée *capture_données* est optionnelle, elle est utilisée pour charger en parallèle la valeur de test de l'étage sériel d'IR ou le code opération de l'instruction de test à appliquer. Lorsque le signal de contrôle *update_ir* généré par le

contrôleur de test est actif, la valeur de SR est transférée en parallèle dans PR pour être appliquée à l'entrée du décodeur.

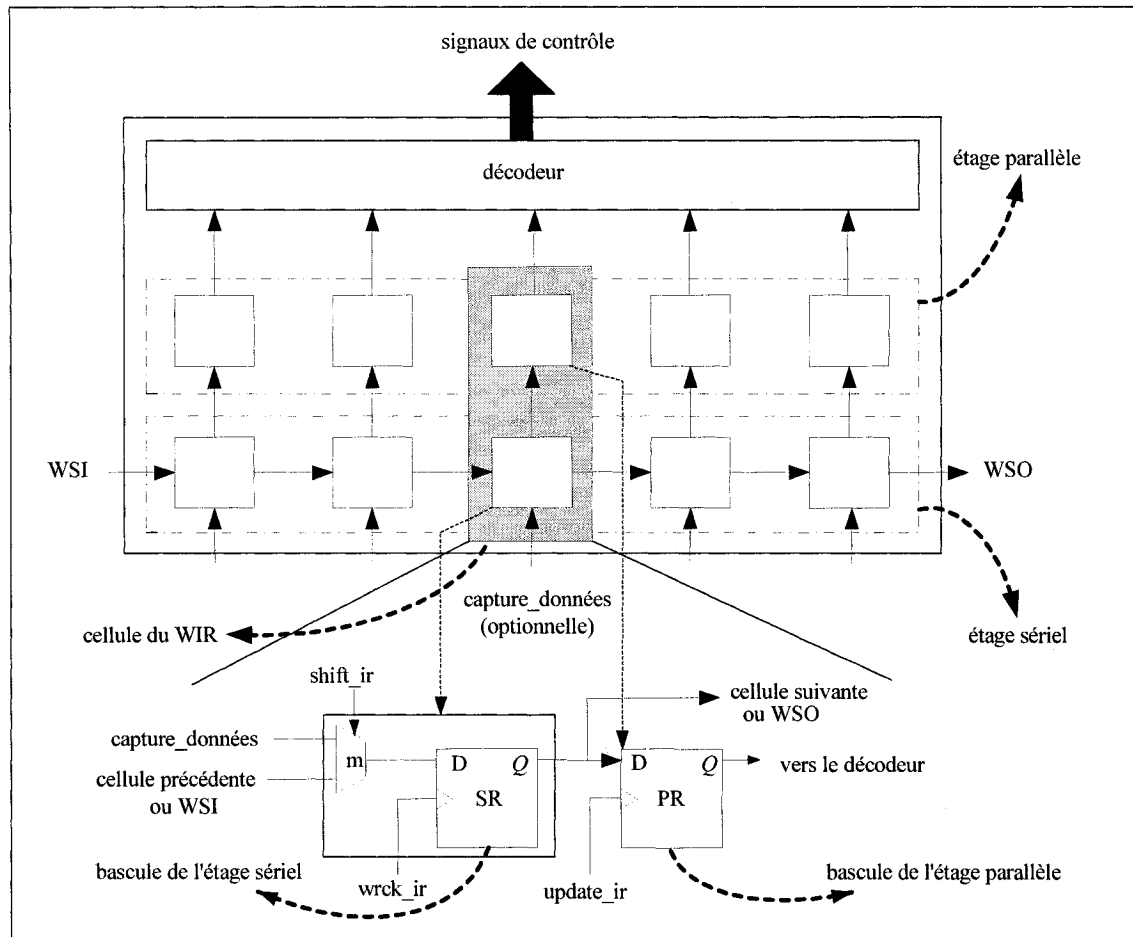


Figure 3.8: Registre d'instruction WIR [19].

Pour le registre WIR de l'enveloppe proposée, nous avons éliminé le multiplexeur *m* qui précède chacune des bascules SR parce que la norme IEEE P1500 ne spécifie aucune entrée parallèle pour le registre d'instruction. Lorsque le signal *shift_ir* est actif, le code opération de l'instruction de test est chargé dans l'étage sériel de WIR sur le front montant de l'horloge *wrck_ir* via l'entrée WSI.

3.5.2 Registre de données WBR

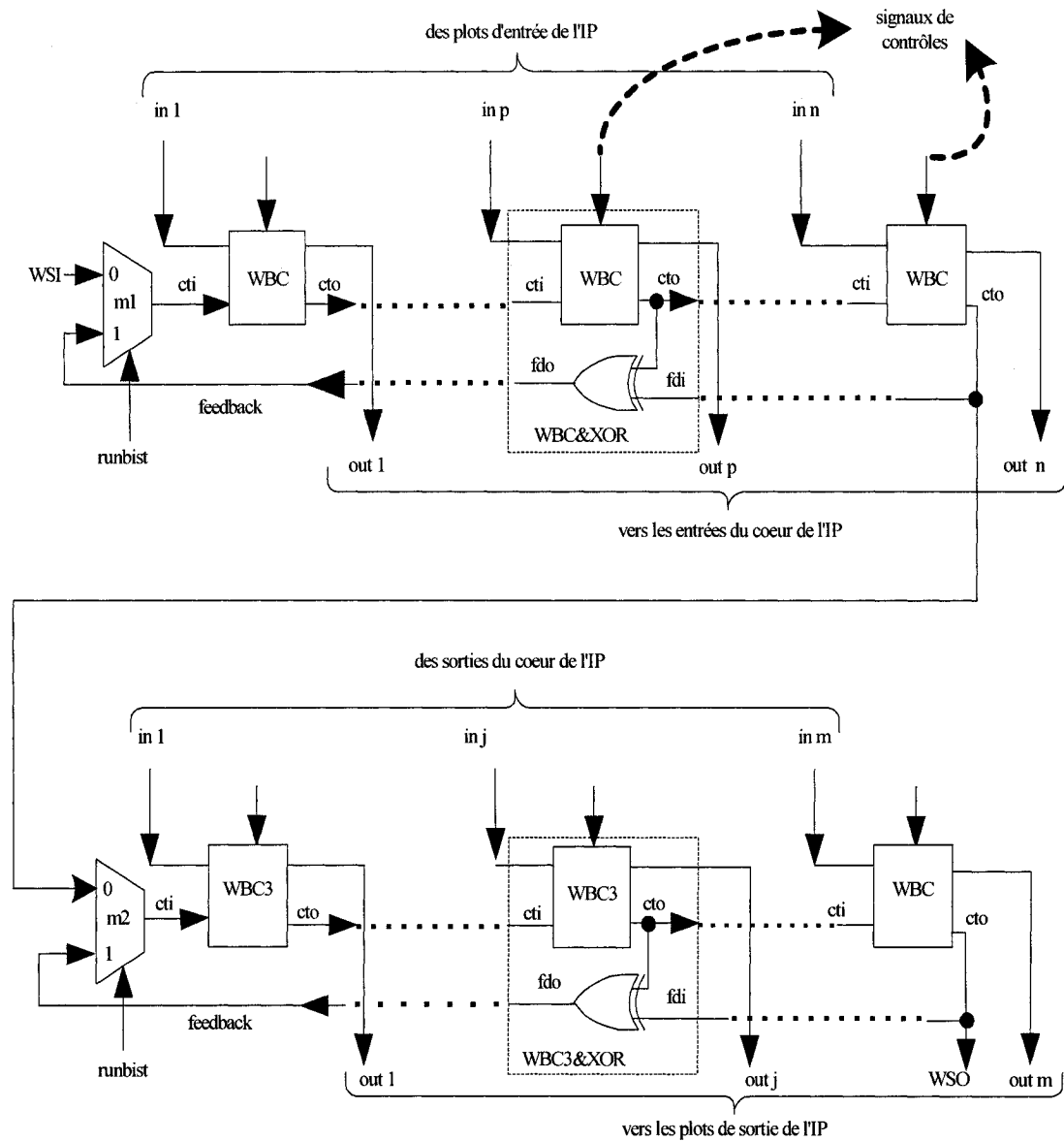


Figure 3.9: Registre à décalage WBR.

WBR est le registre à décalage inséré entre les entrées/sorties du cœur de l'IP et les plots de l'IP. La figure 3.9 montre que WBR est composé de deux multiplexeurs et d'un

ensemble de cellules sans porte XOR tels que WBC et WBC3 et avec porte XOR comme WBC&XOR et WBC3&XOR. WBR permet de contrôler et d'observer les entrées et les sorties du cœur de l'IP à partir de l'extérieur. Ce registre est également configurable en structure BIST pour réaliser le test des cœurs des IP et des interconnexions entre les IP à l'interne du SOC.

En mode normal du cœur de l'IP, le signal de contrôle *wc* des cellules de WBR est à 0. Ceci permet de connecter les entrées et les sorties du cœur de l'IP aux plots de l'IP à travers les signaux fonctionnels *cfi* et *cfo* des cellules de l'enveloppe. Dans ce mode, on dit que les cellules de WBR sont transparentes, ce qui permet d'appliquer directement les données externes au cœur de l'IP et les réponses du cœur de l'IP aux plots de l'IP.

En mode test sans fonctionnalité BIST, les signaux de contrôle des cellules de WBR *runbist* et *bc* sont à 0 et *wc* et *shift_dr* sont à 1. Ceci connecte les bascules SR des cellules de WBR en série à travers leurs signaux *cti* et *cto* pour former l'étage sériel de WBR qui relie l'entrée WSI à la sortie WSO de l'enveloppe. À l'état Shift_DR du contrôleur de test, les valeurs de WSI à appliquer aux entrées du cœur de l'IP et aux plots de sortie de l'IP sont chargées dans l'étage sériel de l'enveloppe pour protéger le cœur de l'IP. Lorsque le signal *update_dr* généré par le contrôleur de test à l'état Update_DR est actif, le contenu de l'étage sériel est appliqué en parallèle aux entrées du cœur de l'IP et aux plots de sortie de l'IP à travers l'étage parallèle de WBR formé par les bascules PR des cellules. Chaque fois qu'un nouveau vecteur de test est chargé dans

l'étage sériel, les réponses du test précédent du cœur de l'IP et des interconnexions sont acheminées en série vers un ATE externe via la sortie WSO de l'enveloppe.

En mode test BIST, les cellules du WBR associées aux entrées/sorties du cœur de l'IP, peuvent être configurées en LFSR et en MISR dépendamment du test à réaliser. Lors du test du cœur de l'IP, les bascules SR des cellules de WBR connectées aux entrées et aux sorties du cœur de l'IP sont respectivement configurées en LFSR et en MISR. À l'état Shift_DR du contrôleur, le signal de contrôle *shift_dr* appliqué aux cellules de WBR est à 1. Ceci permet au LFSR de générer un vecteur de test à chaque front montant de l'horloge de test *wrck*. Dans ce même état du contrôleur, les réponses du cœur de l'IP sont compactées par le MISR. Lors du test des interconnexions entre deux IP, les wrappers des deux IP sont configurés en structure BIST. Les bascules SR des cellules de WBR connectées aux sorties du cœur du premier IP, agissent en tant que LFSR. Les cellules de WBR associées aux entrées du cœur du deuxième IP, forment un MISR. À l'état Shift_DR du contrôleur de test où le signal de contrôle *shift_dr* est 1, le LFSR génère les vecteurs de test pour les interconnexions sur les fronts montants de l'horloge de test. Au même temps, les réponses des interconnexions sont compactées par le MISR.

3.5.3 Registre de données WBY

WBY est le registre qui permet de transférer la valeur de l'entrée WSI de l'enveloppe à sa sortie WSO en un seul cycle d'horloge. Ce registre est constitué d'une bascule D et

d'une porte logique AND, tel qu'il est illustré à la figure 3.10. Lorsque le signal de contrôle *shift_dr* généré à l'état Shift_DR du contrôleur de test est à 1, la valeur de WSI est chargée dans la bascule D au front montant de l'horloge *wrck*.

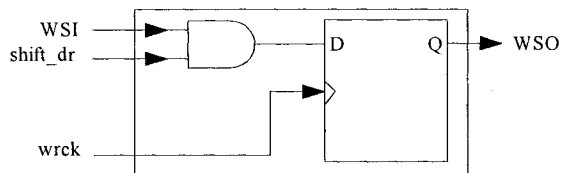


Figure 3.10: Architecture du registre de court-circuit [19].

3.6 Contrôleur de test

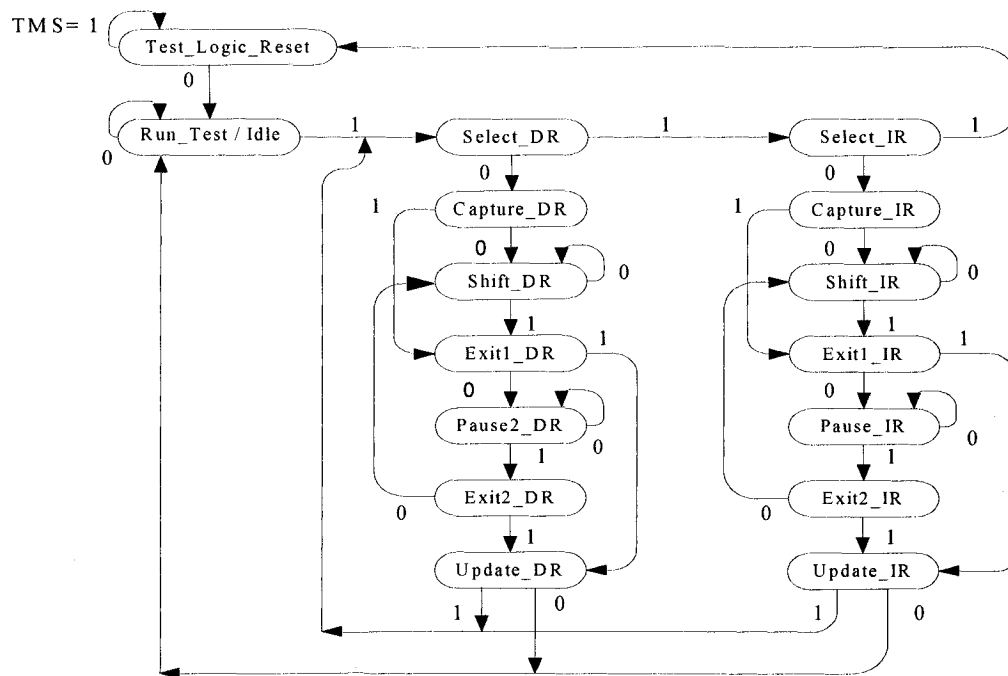


Figure 3.11: Diagramme de transition du contrôleur TAP [19].

Le contrôleur de test gère les différentes opérations effectuées au niveau de la nouvelle enveloppe tels que le chargement en série du code opération de l'instruction de test et

des données à appliquer au cœur de l'IP et aux interconnexions. Pour la nouvelle architecture de test proposée, nous avons réutilisé le contrôleur de test « Test Access Port controller » (TAP) du Boundary Scan que la norme IEEE P1500 utilise également. La figure 3.11 représente le diagramme de ce contrôleur qui est une machine séquentielle asynchrone (MSA). Dans cette MSA, la transition d'un état à un autre sur un front montant de l'horloge de test « Test Clock » (TCK) dépend de la valeur du signal de sélection « Test Mode Select » (TMS). En mode normal, le signal TMS est à 1 et le contrôleur demeure à l'état `Test_Logic_Reset`. Le diagramme de cette MSA est composé de deux branches.

Les états de la branche de droite du diagramme servent à la génération des signaux de contrôle appliqués au registre d'instruction WIR du warpper. À l'état `Shift_IR` de cette branche, le contrôleur génère le signal *shift_ir* et celui de l'horloge de test *wrck_ir*. Ces deux signaux permettent de charger le code opération de l'instruction de test à appliquer dans l'étage sériel de WIR. À l'état `Update_IR`, le contrôleur génère le signal *update_ir* qui est appliqué à l'étage parallèle du WIR. Ce signal permet de transférer le contenu de l'étage sériel dans l'étage parallèle de WIR pour être appliqué aux entrées du décodeur qui génère les signaux de contrôle nécessaires au fonctionnement de l'enveloppe.

Les états de la branche de gauche du diagramme servent à la génération des signaux de contrôle et du signal d'horloge de test qui sont nécessaires pour le test du cœur de l'IP et des interconnexions avec ou sans la fonctionnalité BIST. À l'état `Shift_DR` de cette

branche, le contrôleur de test produit le signal *shift_dr* et le signal d'horloge de test *wrck* qui sont appliqués aux cellules de WBR.

En mode test sans BIST, les signaux de contrôle *shift_dr* et *wrck* permettent de charger les données de test à appliquer aux entrées du cœur de l'IP et aux interconnexions dans l'étage sériel de l'enveloppe à partir d'un ATE externe. Ces signaux de contrôle servent aussi au transfert des réponses de test vers un ATE externe pour analyse. En mode test BIST, les signaux *shift_dr* et *wrck* contrôlent la génération des vecteurs de test à appliquer aux entrées du cœur de l'IP et aux interconnexions ainsi que le compactage des réponses à l'interne du SOC. À l'état Update_DR, le contrôleur génère le signal *update_dr* qui permet l'application du contenu de l'étage sériel de l'enveloppe aux entrées du cœur de l'IP et aux plots de sortie de l'IP.

3.7 Instructions de test implémentées par la nouvelle enveloppe

Tableau 3.2: Valeurs des signaux de contrôle pour les instructions de test.

Instruction	code	wci	wco	bci	bco	runbist	select	freeze
WBAYPASS	111	0	0	X	X	X	0	X
WEXTESTS	000	0	1	0	0	0	1	0
WCORETESTS	001	1	0	0	0	0	X	0
WPRELOADS	010	0	0	0	0	0	0	0
WCLAMP	011	1	1	0	0	0	1	1
WOPMISR	100	0	1	0	1	0	X	0
WCOREBIST	101	1	1	0	1	1	X	0
WEXBIST	110	1	1	1	0	1	1	0

Le tableau 3.2 présente l'ensemble des instructions que l'enveloppe proposée implémente pour le test des IP et des interconnexions avec ou sans la fonctionnalité BIST. Les cinq premières instructions de la première colonne du tableau sont celles du mode test sans BIST. Les trois dernières instructions de la première colonne sont celles du mode test avec BIST qui sont propres à notre architecture de test. La deuxième colonne du tableau présente les codes opération attribués aux instructions de test. Les autres colonnes du tableau présentent les valeurs des signaux de contrôle générés par le décodeur de WIR pour chacune des instructions de test. Les sorties *wci* et *bci* de WIR sont respectivement connectées aux signaux de contrôle *wc* et *bc* des cellules de l'enveloppe associées aux entrées du cœur de l'IP. Les sorties *wco* et *bco* de WIR sont respectivement connectées aux signaux de contrôle *wc* et *bc* des cellules de l'enveloppe reliées aux sorties du cœur de l'IP. Les sorties *runbist* et *freeze* du décodeur sont appliquées à toutes les cellules de l'enveloppe. La sortie *select* du décodeur est connectée à l'entrée *select* de la cellule de contrôle si l'enveloppe contient des cellules de sorties à trois états et bidirectionnelles. Nous détaillons ci-après les différentes instructions de test du tableau 3.2, que nous avons classé en deux catégories: les instructions du test sans fonctionnalité BIST et les instructions du test avec fonctionnalité BIST.

3.7.1 Instructions de test sans fonctionnalité BIST

Les instructions de test des IP et des interconnexions du SOC sans BIST sont : WBYPASS, WEXTTESTS, WCORETESTS, WCLAMP et WPRELOADS [31]. Ces

instructions permettent de tester les IP et les interconnexions conformément à la norme IEEE P1500.

3.7.1.1 Instruction WBYPASS

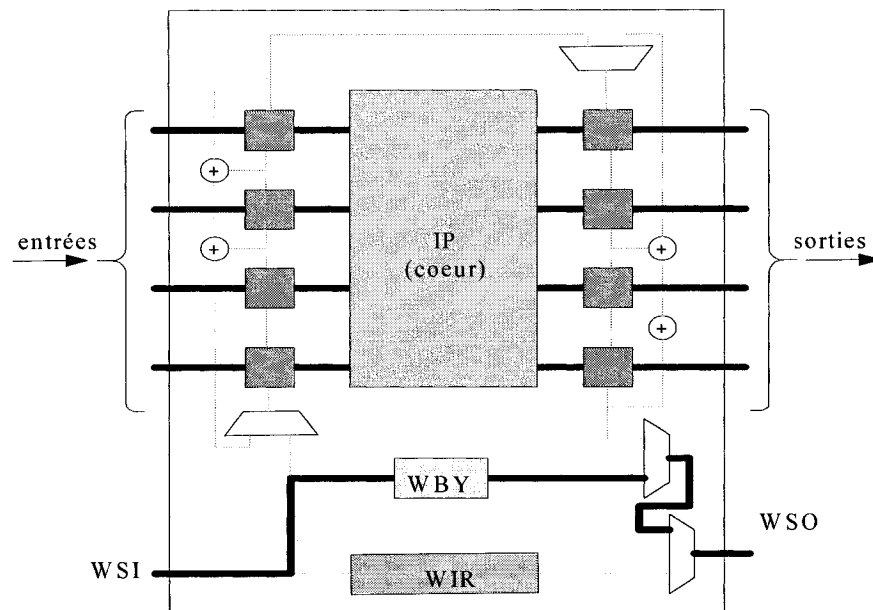


Figure 3.12: Instruction WBYPASS.

L'instruction WBYPASS est prévue pour configurer l'IP en fonctionnement normal et raccourcir la longueur du chemin d'accès au test (TAM). La figure 3.12 montre la configuration que l'enveloppe prend lorsque le code opération de WBYPASS est chargé et mis à jour dans le registre WIR. Dans cette configuration, les entrées/sorties du cœur de l'IP sont connectées directement aux plots d'entrée/sortie de l'IP via les cellules de l'enveloppe. Cette instruction permet aussi au registre WBY de court-circuiter WBR de

l'enveloppe, ce qui réduit la longueur du registre à décalage entre WSI et WSO. Ceci permet d'atteindre rapidement les autres IP et les interconnexions du SOC à tester.

3.7.1.2 Instruction WEXTESTS

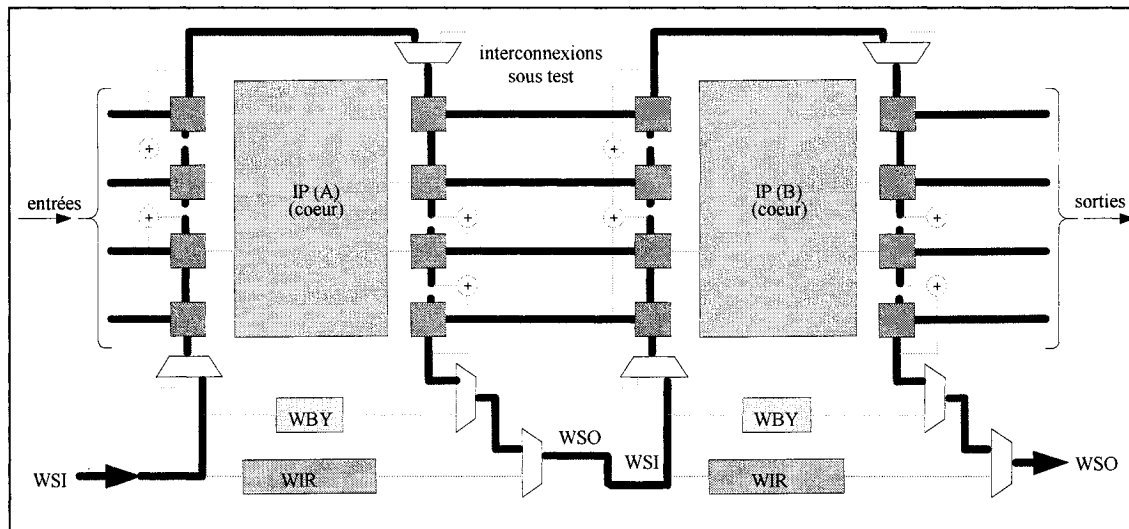


Figure 3.13: Instruction WEXTESTS.

L'instruction de test WEXTESTS permet de vérifier les interconnexions entre les IP d'un SOC avec des ATE externes. Lorsque le code opération de WEXTESTS est chargé et mis à jour dans WIR, les cellules de l'enveloppe forment un long registre à décalage entre WSI et WSO. Ce registre permet d'accéder aux plots d'entrée/sortie de l'IP à partir de l'extérieur. À titre d'exemple, le test des interconnexions entre l'IP(A) et l'IP(B) de la figure 3.13 est réalisé, tel qu'il est décrit ci-après. À l'état Shift_IR du contrôleur de test, le code opérations de WEXTESTS est chargé dans l'étage sériel de WIR de l'IP(A) et dans celui de l'IP(B). À l'état Update_IR du contrôleur de test, ce code opération est

mis à jour au niveau de chaque IP pour configurer leurs WBR en registre à décalage. À l'état Shift_DR du contrôleur de test, les valeurs de WSI à appliquer aux interconnexions sont chargées en série dans les bascules SR des cellules de l'enveloppe associées aux sorties du cœur de l'IP(A). À l'état Update_DR du contrôleur de test, les valeurs des bascules SR sont par la suite appliquées aux plots de sortie de l'IP(A) à travers les bascules PR. À l'état Capture_DR du contrôleur de test, les réponses des interconnexions sont copiées dans les bascules SR des cellules de l'enveloppe connectées aux entrées du cœur de l'IP(B). Ces réponses sont transférées en série vers un ATE externe lorsqu'un nouveau vecteur de test est chargé dans les cellules de l'enveloppe à l'état Shift_DR du contrôleur.

3.7.1.3 Instruction WCORETESTS

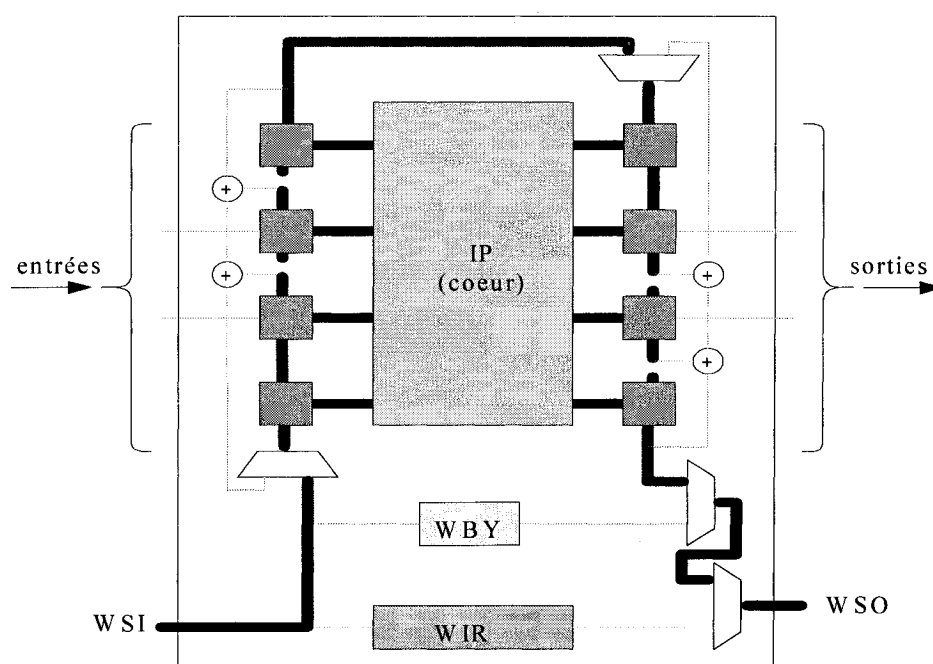


Figure 3.14: Instruction WCORETESTS.

L'instruction de test WCORETESTS permet de contrôler et d'observer les signaux d'entrée/sortie de cœur de l'IP à partir de l'extérieur du SOC. L'application de WCORETESTS pour le test du cœur de l'IP est réalisée en plusieurs étapes. À l'état Shift_IR du contrôleur de test, le code opération de cette instruction est chargé en série dans WIR. La figure 3.14 montre la configuration que l'enveloppe proposée prend lorsque le code opération de WCORETESTS est mis à jour à l'état Update_IR du contrôleur de test. Dans cette configuration WBR agit en registre à décalage qui relie l'entrée WSI à la sortie WSO de l'enveloppe. Avec cette instruction de test, à l'état Shift_DR du contrôleur de test, les valeurs de WSI à appliquer aux entrées du cœur de l'IP sont d'abord chargées en série dans les bascules SR des cellules de WBR afin de protéger le cœur de l'IP. À l'état Update_DR du contrôleur de test, les valeurs des bascules SR sont par la suite appliquées aux entrées du cœur à travers les bascules PR qui forment l'étage parallèle de WBR. À l'état Capture_DR du contrôleur de test, les réponses du cœur sont copiées en parallèle dans les bascules SR des cellules de WBR associées aux sorties du cœur de l'IP. Chaque fois qu'un nouveau vecteur de test est chargé dans l'étage sériel du WBR à l'état Shift_DR, la réponse du test précédent est transférée en série vers un ATE via WSO.

3.7.1.4 Instruction WPRELOADS

L'instruction de test WPRELOADS sert à effectuer des pré-chargements sériels des données de test dans les cellules de WBR de l'enveloppe sans que le fonctionnement

normal de l'IP ne soit altéré. Cette instruction précède toute instruction de test qui nécessite un pré-chargement de données. La figure 3.15 représente la configuration que l'enveloppe prend lorsque le code opération de WPRELOADS est chargé dans WIR à l'état Shift_IR et mis à jour à l'état Update_IR du contrôleur de test. Dans cette configuration, les signaux d'entrée/sortie du cœur de l'IP sont reliés directement aux plots d'entrée/sortie de l'IP à travers les cellules du WBR. Ainsi, les bascules SR des cellules de WBR forment un registre à décalage entre l'entrée WSI et la sortie WSO de l'enveloppe. Le pré-chargement des données de test dans les cellules de WBR est réalisé en deux étapes. D'abord, à l'état Shift_DR du contrôleur de test, les valeurs de WSI sont chargées en série dans l'étage sériel de WBR. Par la suite, à l'état Update_DR du contrôleur de test, ces valeurs sont transférées dans les bascules PR des cellules de WBR pour les appliquer au cœur de l'IP et aux plots de sortie de l'IP.

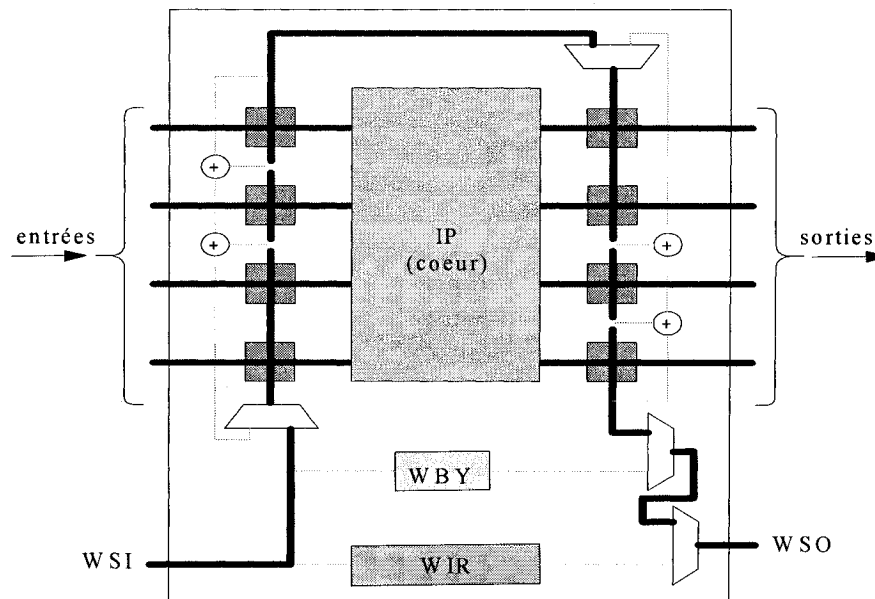


Figure 3.15: Instruction WPRELOADS.

3.7.1.5 Instruction WCLAMP

L'instruction de test WCLAMP a été prévue pour fixer l'état de sortie des cellules d'entrée/sortie du registre WBR à des valeurs pré-chargées auparavant par l'instruction WPRELOADS. Cette instruction permet de maintenir certains signaux à des valeurs fixes lors d'une procédure de test des IP et des interconnexions. Elle permet aussi de protéger les cœurs des IP qui ne rentrent pas dans une procédure de test. La figure 3.16 montre la configuration que l'enveloppe prend lorsque le code opération de cette instruction est chargé dans WIR à l'état Shift_IR du contrôleur de test et mis à jour à l'état Update_IR. Lorsque WCLAMP est active dans l'enveloppe, l'entrée WSI est reliée à la sortie WSO à travers le registre WBY afin de permettre le transfert en série des données de test à appliquer aux IP et aux interconnexions du SOC.

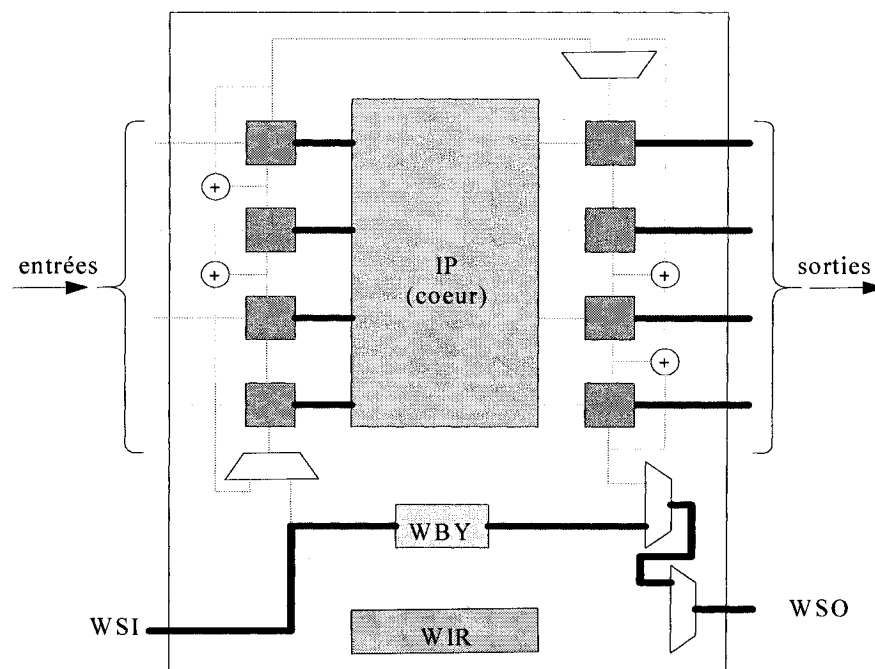


Figure 3.16: Instruction WCLAMP.

3.7.2 Instructions de test avec fonctionnalité BIST

En plus des instructions de la norme IEEE P1500 présentées dans la sous-section précédente, que l'enveloppe proposée implémente, nous avons prévu d'autres instructions de test. Ces instructions qui sont propres à notre architecture de test, configurent l'enveloppe proposée en structure BIST afin de tester les cœurs des IP et les interconnexions entre les IP à l'interne du SOC. Ce mode de test permet de minimiser le temps d'application des vecteurs de test. Ces instructions de test sont WEXBIST, WCOREBIST et WOPMISR, que nous présentons ci-dessous.

3.7.2.1 Instruction WEXBIST

L'instruction de test WEXBIST permet de tester les interconnexions entre les IP à l'interne du SOC. Cette instruction configure les cellules de WBR connectées aux sorties du cœur de l'IP en LFSR pour générer les vecteurs de test à appliquer aux interconnexions. WEXBIST configure aussi les cellules de WBR reliées aux entrées du cœur de l'IP en MISR pour compacter les réponses des interconnexions. Contrairement à l'instruction WEXTTESTS présentée auparavant, WEXBIST ne nécessite aucune génération préalable des vecteurs de test par des outils externes, ni d'ATE externe pour leur application. Cette instruction permet également de tester les interconnexions à la vitesse optimale du SOC. WEXBIST est toujours précédée par l'instruction WPRELOADS pour charger le vecteur initial du LFSR et du MISR. À titre d'exemple pour tester en mode BIST les interconnexions entre l'IP(A) et l'IP(B) de la figure 3.17,

nous appliquons d'abord l'instruction WPRELOADS pour les deux IP. Cette instruction configure l'enveloppe de l'IP(A) et celui de l'IP(B) en registre à décalage qui permet d'initialiser les bascules SR des cellules de l'enveloppe à partir de l'extérieur. Par la suite, nous chargeons le code opération de WEXBIST dans WIR de l'IP(A) et dans celui de l'IP(B) à l'état Shift_IR du contrôleur de test. Après la mise à jour de cette instruction à l'état Update_IR du contrôleur de test, l'enveloppe de l'IP(A) et celle de l'IP(B) agissent en structure BIST. Les vecteurs de test à appliquer aux interconnexions sont générés à l'état Shift_DR du contrôleur de test par le LFSR formé par les bascules SR des cellules associées aux sorties du cœur de l'IP(A). Au même moment, les réponses de test des interconnexions sont compactées par les bascules SR des cellules associées aux entrées du cœur de l'IP(B), qui sont configurées en MISR.

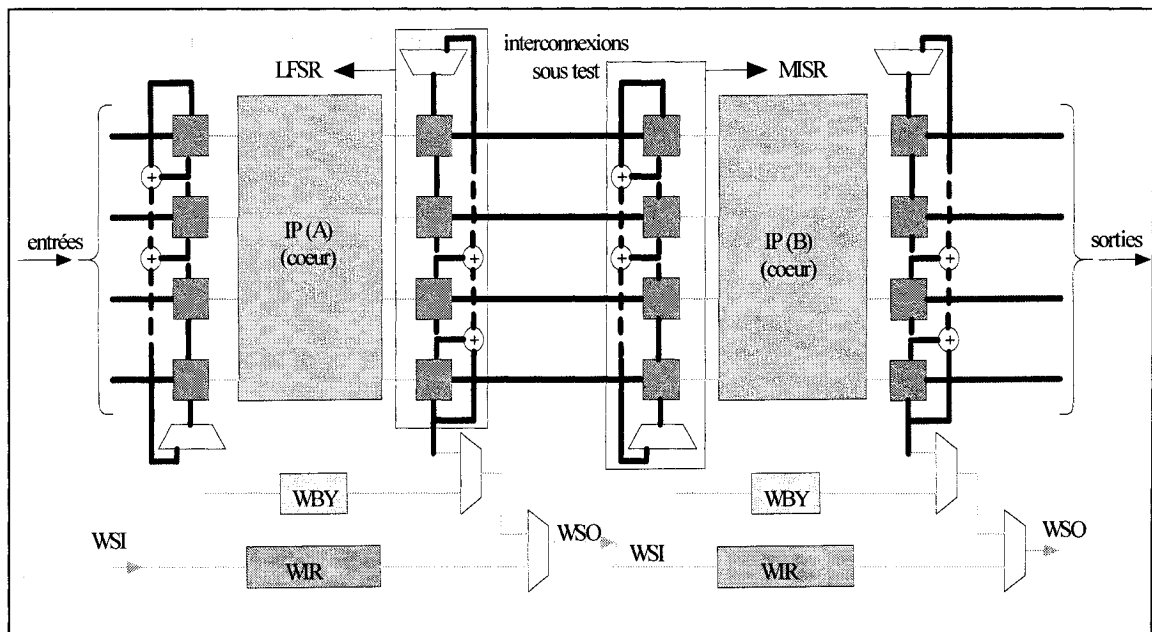


Figure 3.17: Instruction WEXBIST.

3.7.2.2 Instruction WCOREBIST

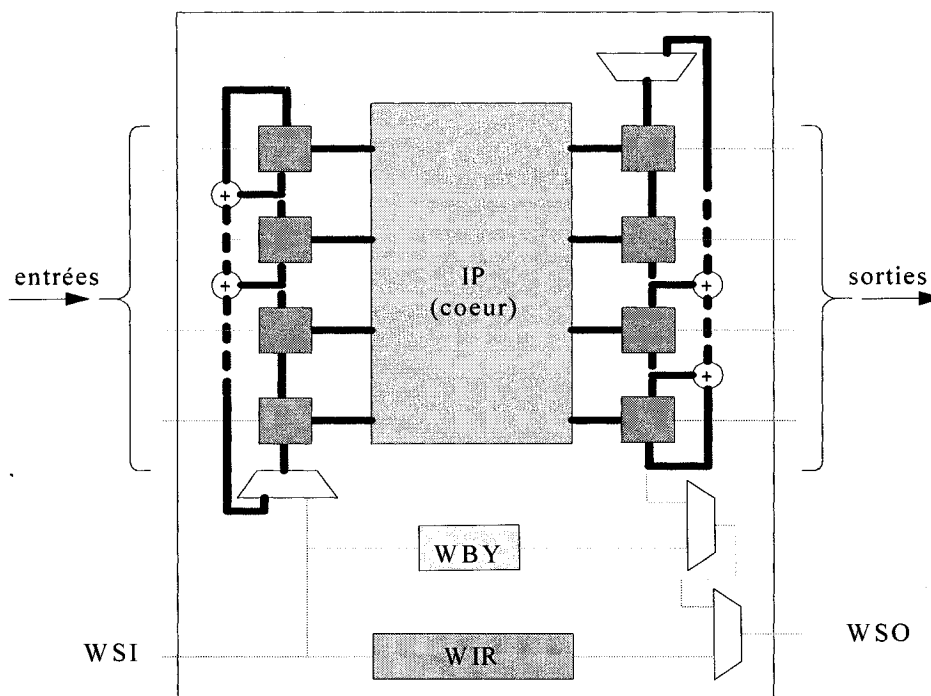


Figure 3.18: Instruction WCOREBIST.

WCOREBIST permet de générer les vecteurs de test à appliquer au cœur de l'IP et d'analyser leurs réponses à l'interne du SOC. Contrairement à l'instruction de test WCORETESTS présentée précédemment, l'instruction WCOREBIST n'utilise pas d'ATE externe pour appliquer les vecteurs de test au cœur de l'IP et pour analyser les réponses. L'instruction WCOREBIST est précédée par WPRELOADS afin de charger les valeurs initiales dans les bascules SR des cellules de l'enveloppe qui agissent en LFSR et en MISR. La figure 3.18 montre la configuration que l'enveloppe prend lorsque le code opération de WCOREBIST est chargé dans WIR à l'état Shift_IR du contrôleur

de test et mis à jour à l'état Update_IR. Dans cette configuration, les bascules SR des cellules de l'enveloppe connectées aux signaux d'entrée/sortie du cœur de l'IP forment respectivement un LFSR et un MISR. À l'état Shift_DR du contrôleur de test, le LFSR génère les vecteurs de test à appliquer aux entrées du cœur de l'IP. Dans ce même état, le MISR compacte les réponses de test du cœur de l'IP pour calculer la signature finale du test.

3.7.2.3 Instruction de test WOPMISR

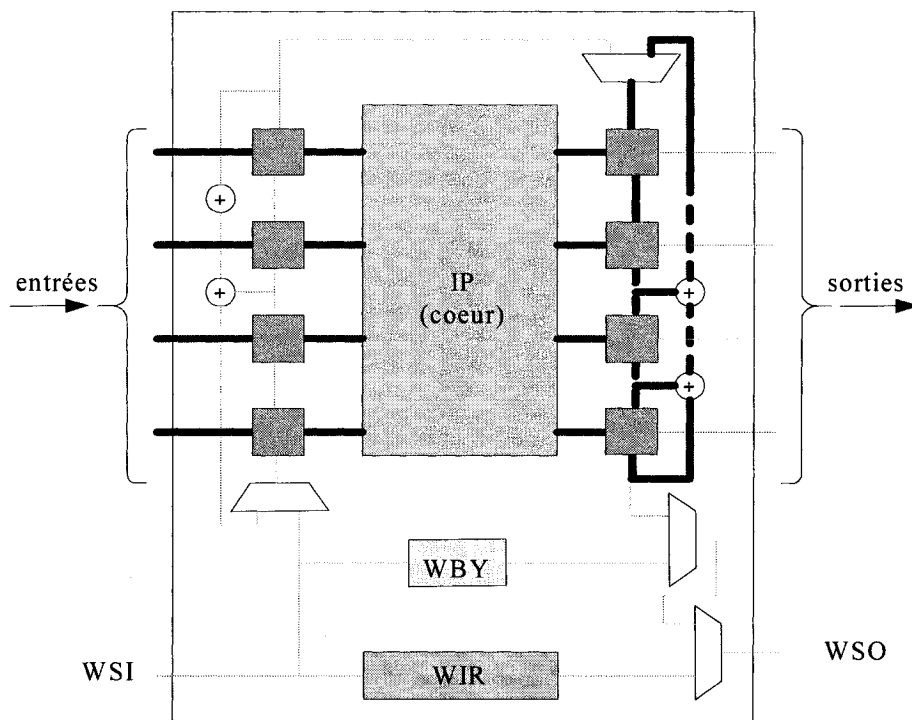


Figure 3.19: Instruction WOPMISR.

Le mode de test OPMISR « On-Product MISR » consiste à tester l'IP avec des vecteurs de test déterministes appliqués en parallèle à partir d'un ATE externe et à compacter les réponses de test à l'interne du SOC avec un MISR. Ce concept a été introduit par la compagnie IBM [6] afin d'assurer une bonne qualité de test pour les IP tout en utilisant des ATE qui ont moins d'espace mémoire. Pour réaliser cette approche de test, les concepteurs d'IBM intègrent des MISR au niveau des plots de sortie des IP qui sont déjà encapsulés dans une enveloppe. Ceci engendre l'augmentation de la surface additionnelle et la dégradation des délais internes de propagation.

Pour réaliser la même stratégie de test que celle d'IBM mais avec moins de surface additionnelle et moins de dégradation des délais internes, nous avons prévu l'instruction WOPMISR. Cette instruction permet à l'enveloppe proposée d'implémenter l'approche OPMISR en réutilisant les cellules connectées aux signaux de sortie du cœur de l'IP pour former le MISR. La figure 3.19 montre la configuration que l'enveloppe prend lorsque l'instruction de test WOPMISR est appliquée. L'instruction WOPMISR est toujours précédée par l'instruction WPRELOADS pour initialiser les bascules SR des cellules de l'enveloppe connectées aux sorties du cœur de l'IP à partir de l'extérieur. Pour implémenter l'approche OPMISR au niveau de l'enveloppe proposée, nous chargeons en série le code opération de WOPMISR dans WIR à l'état Shift_IR du contrôleur de test. Après la mise à jour de ce code à l'état Update_IR du contrôleur de test, WIR génère les signaux de contrôle nécessaires à la configuration de l'enveloppe. Ces signaux permettent aux entrées du cœur de l'IP d'être connectées en parallèle à un

mécanisme d'accès au test « Test Access Mechanism » (TAM) à travers les cellules de l'enveloppe. Ces signaux de contrôle configurent également les bascules SR des cellules de l'enveloppe associées aux sorties du cœur de l'IP en un MISR. Dans ce mode OPMISR, les vecteurs de test sont appliqués en parallèle aux entrées du cœur de l'IP à partir d'un ATE externe et les réponses sont compactées par le MISR.

3.8 Conclusion

La nouvelle architecture de test présentée dans ce chapitre, permet de tester les IP et les interconnexions du SOC avec une même structure de test. Ceci minimise le nombre d'architectures de test à insérer dans le SOC, ce qui permet de réduire l'importance de la surface additionnelle. Les possibilités de test que notre architecture procure pour les IP et les interconnexions du SOC permettent à l'intégrateur d'assurer une bonne qualité de test en un temps raisonnable. Ceci permet à l'intégrateur de réaliser des bons compromis entre la qualité et le coût de test du SOC.

CHAPITRE 4

IMPLÉMENTATION ET RÉSULTATS

4.1 Introduction

Dans ce chapitre, nous présentons les résultats de la validation de l'architecture de test proposée, que nous avons obtenu par simulation. Lors de cette validation, nous avons vérifié le fonctionnement de la nouvelle architecture de test selon les spécifications des différentes instructions de test présentées précédemment à la section 3.7. Nous avons aussi estimé les dégradations des délais internes engendrées par l'insertion de l'enveloppe proposée autour des IP. Pour vérifier l'impact de la nouvelle enveloppe sur la surface de l'IP, nous avons encapsulé quelques circuits d'essai dans l'architecture de test proposée, et estimé leurs surfaces. Ces surfaces ont été comparées à celles de mêmes circuits d'essai mais encapsulés dans l'architecture de test conventionnelle. Cette architecture conventionnelle est composée d'une enveloppe P1500 et d'une structure BIST. Le but de cette comparaison est d'évaluer le taux de réduction de la surface que la nouvelle enveloppe permet de réaliser comparativement à l'architecture conventionnelle. À la dernière étape de la validation, nous avons vérifié la qualité de test qu'assure la nouvelle architecture de test ainsi que la réduction de temps d'application des vecteurs de test externes. Avant d'exposer les différents résultats de la validation,

nous expliquons d'abord comment l'architecture de test proposée a été insérée autour des circuits d'essai.

4.2 Insertion de l'architecture de test

L'insertion de la nouvelle enveloppe autour des IP peut être réalisée de façon manuelle, mais cette opération peut être longue et fastidieuse pour les IP complexes. Ceci prolonge le temps d'insertion et augmente le risque d'erreurs. À titre d'exemple, la durée de l'insertion manuelle du Boundary Scan autour des CI complexes et sa vérification, peut atteindre six à huit semaines selon Mentor Graphics [15]. D'une manière générale, la structure du Boundary Scan est semblable à celle présentée au chapitre précédent. Pour réduire le temps d'insertion de l'architecture de test proposée, il faut automatiser cette opération. Dans cette éventualité, nous avons essayé d'utiliser les outils existants d'insertion automatique du Boundary Scan tels que le BSDArchitect de Mentor Graphics et BSDCompiler de Synopsys. Après vérification, ces outils ne peuvent pas insérer des structures BIST et intègrent automatiquement un contrôleur de test au niveau de chaque circuit, ce qui n'est pas prévu par la norme IEEE P1500. Nous avons donc opté pour l'insertion manuelle de la nouvelle enveloppe de test et de l'architecture de test conventionnelle autour des circuits d'essai. Pour chaque circuit, nous avons élaboré deux fichiers en VHDL « Very High Speed Integrated Circuit Hardware Description Language » structurel. Le premier fichier est celui de l'encapsulation du circuit dans l'enveloppe proposée. Le deuxième fichier est celui de l'encapsulation du circuit dans l'architecture de test conventionnelle.

4.3 Simulation fonctionnelle de l'architecture de test proposée

Le but de la simulation fonctionnelle est de vérifier si l'enveloppe proposée implémente correctement les instructions de test prévues pour le test des IP et des interconnexions du SOC. Pour les besoins de la simulation, nous avons choisi de synthétiser l'enveloppe proposée autour d'un tampon à 4 bits afin de faciliter la lecture des résultats et leur interprétation. Nous avons aussi élaboré un banc d'essai « testbench » en langage VHDL pour chacune des instructions du tableau 3.2 de la section 3.7. Dans les deux prochaines sections, nous présentons les chronogrammes de la simulation de ces bancs d'essais réalisée avec l'outil Modelsim de Mentor Graphics.

La figure 4.1 montre les différents étages de l'enveloppe dont les valeurs internes ont été vérifiées lors de la simulation. L'étage *sdr_in* « serial data register in » fait référence à la valeur du registre à décalage formé par les bascules SR des cellules de WBR reliées aux entrées du cœur de l'IP. L'étage *pdr_in* « parallel data register in » représente la valeur du registre parallèle formé par les bascules PR des cellules de WBR connectées aux entrées du cœur de l'IP. L'étage *sdr_out* « serial data register out » fait référence au registre à décalage formé par les bascules SR des cellules de WBR reliées aux sorties du cœur de l'IP. L'étage *pdr_out* « parallel data register out » représente le registre parallèle formé par les bascules PR des cellules de WBR connectées aux sorties du cœur de l'IP. L'étage *sir* « serial instruction register » et l'étage *pir* « parallel instruction register » font référence aux registres sériel et parallèle du WIR de l'enveloppe.

4.4.1 Instruction de test WBAYPASS

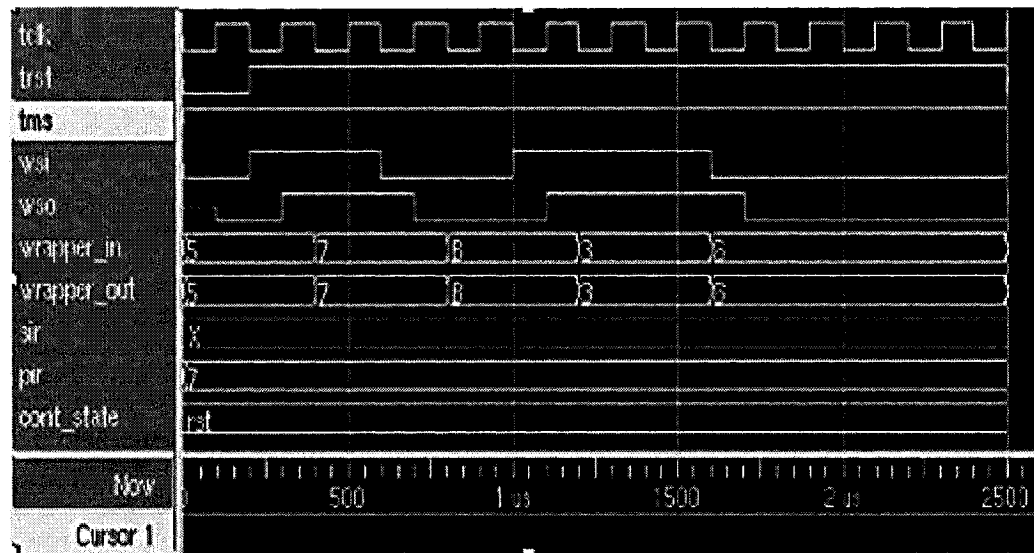


Figure 4.2: Chronogramme de l'instruction de test WBYPASS.

Le chronogramme de la figure 4.2 est celui de la simulation de l'instruction de test WBYPASS qui permet de configurer l'enveloppe proposée en mode fonctionnement normal. L'étage *pir* du WIR contient par défaut le code opération 7 en hexadécimal de l'instruction WBAYPASS du tableau 3.2. Ce chronogramme montre que lorsque WBAYPASS est active dans l'enveloppe et le signal de sélection « test mode select » (*tms*) du contrôleur de test) est à 1, le contrôleur de test reste toujours à l'état Test_Logic_RESET (*cont_state=rst*). Le chronogramme montre également qu'avec l'instruction WBYPASS, le tampon conserve son fonctionnement normal où les valeurs appliquées à ses entrées (*wrapper_in*) sont transférées en parallèle à ses sorties (*wrapper_out*). WBAYPASS permet aussi de transférer la valeur de WSI à WSO de

l'enveloppe sur un front montant de chaque cycle d'horloge de test « test clock » (*tlck*). Ceci prouve que le registre WBR de l'enveloppe est court-circuité par WBY conformément aux spécifications de WBAYPASS.

4.4.2 Instruction de test WEXTTESTS

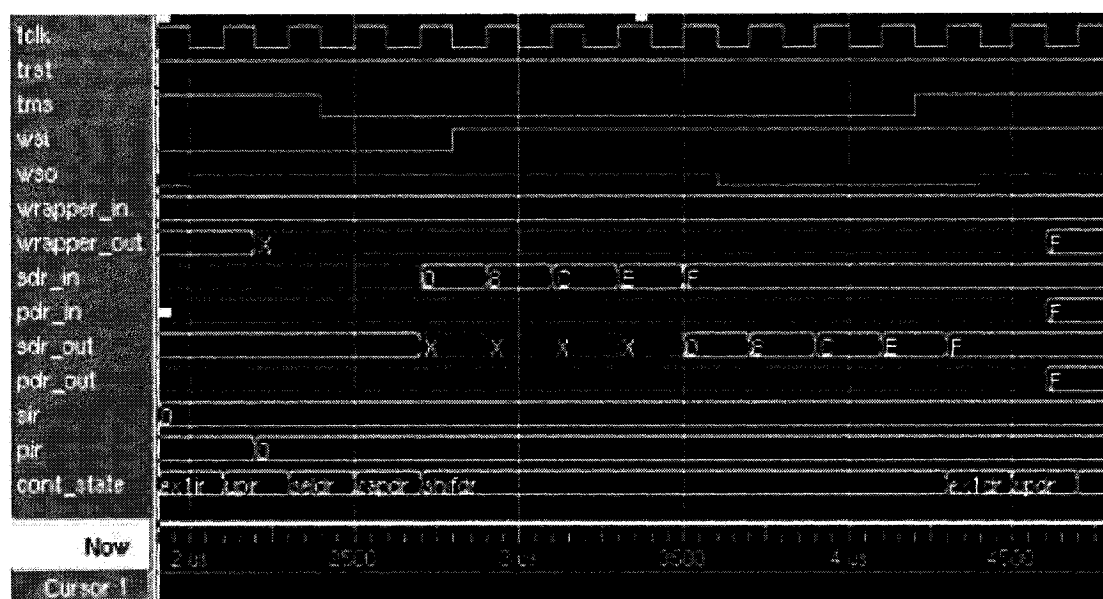


Figure 4.3: Chronogramme de l'instruction de test WEXTTESTS.

Le chronogramme de la figure 4.3 est celui de la simulation de l'instruction WEXTTESTS qui permet de tester les interconnexions entre les IP en utilisant des équipements de test externes. Le code opération 0 de WEXTTESTS du tableau 3.2, qui a été chargé auparavant dans l'étage *sir* de WIR à l'état Shift_IR du contrôleur de test. Ce code est mis à jour dans l'étage *pir* de WIR à l'état Update_IR du contrôleur de test (*cont_state=upir*). Cette mise à jour permet à l'étage *sdr_in* et *sdr_out* de WBR de

former un registre à décalage reliant WSI à WSO de l'enveloppe. À l'état Shift_DR du contrôleur de test (*cont_state=shifdr*), la valeur de test F à appliquer aux interconnexions est chargée dans *sdr_out* à travers l'étage *sdr_in* et l'entrée WSI. À l'état Update_DR du contrôleur de test (*cont_state=updr*), la valeur F est transmise en parallèle dans l'étage *pdr_out* pour être appliquée aux plots de sortie de l'enveloppe.

4.4.3 Instruction de test WCORETESTS

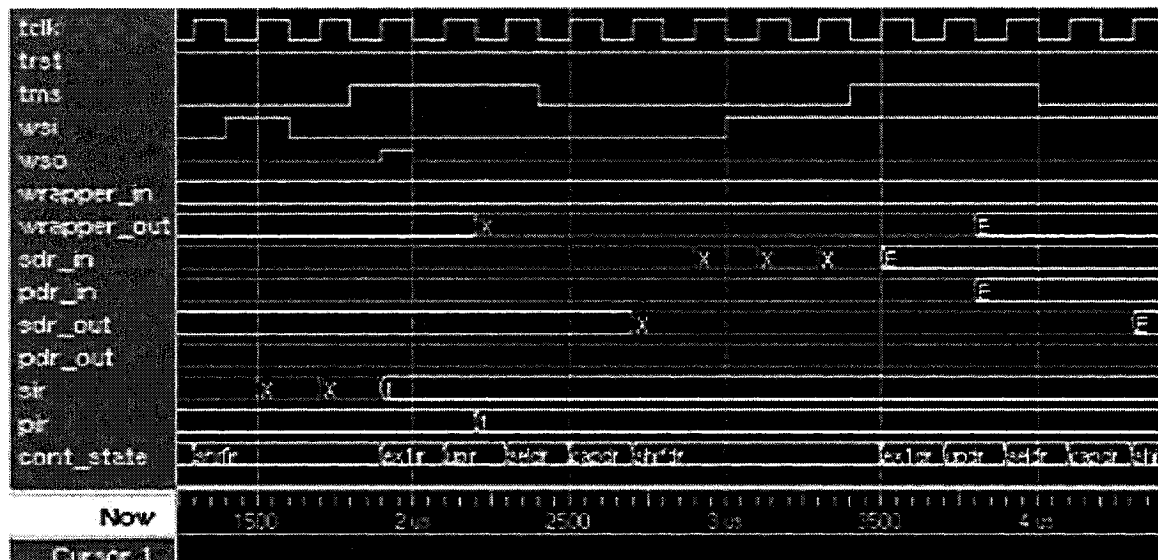


Figure 4.4: Chronogramme de l'instruction de test WCORETESTS.

Le chronogramme de la figure 4.4 est celui de la simulation de l'instruction WCOREBIST qui permet de tester la logique interne de l'IP à partir de l'extérieur du SOC. À l'état Shift_IR du contrôleur de test (*cont_state=shifir*), le code opération 1 de WCOREBIST du tableau 3.2 est d'abord chargé dans l'étage *sir* de WIR. À l'état Update_IR du contrôleur de test (*cont_state=upir*), ce code est transféré dans l'étage *pir*

de WIR pour être appliqué aux entrées du décodeur qui génère les signaux de contrôle. Ces signaux permettent à WBR de l'enveloppe d'agir en tant qu'un registre à décalage reliant WSI à WSO de l'enveloppe. À l'état Shift_DR du contrôleur de test (*cont_state=shifdr*), la valeur de test E à appliquer au tampon est chargée en série dans l'étage *sdr_in* du WBR. À l'état Update_DR du contrôleur de test (*cont_state=updr*), cette valeur de test est transférée en parallèle dans l'étage *pdr_in* pour être appliquée aux signaux d'entrée du tampon. À l'état Capture_DR du contrôleur de test (*cont_state=capdr*), la réponse du tampon est copiée dans l'étage *sdr_out* pour la transmettre vers un ATE externe pour analyse.

4.4.4 Instruction de test WPRELOADS

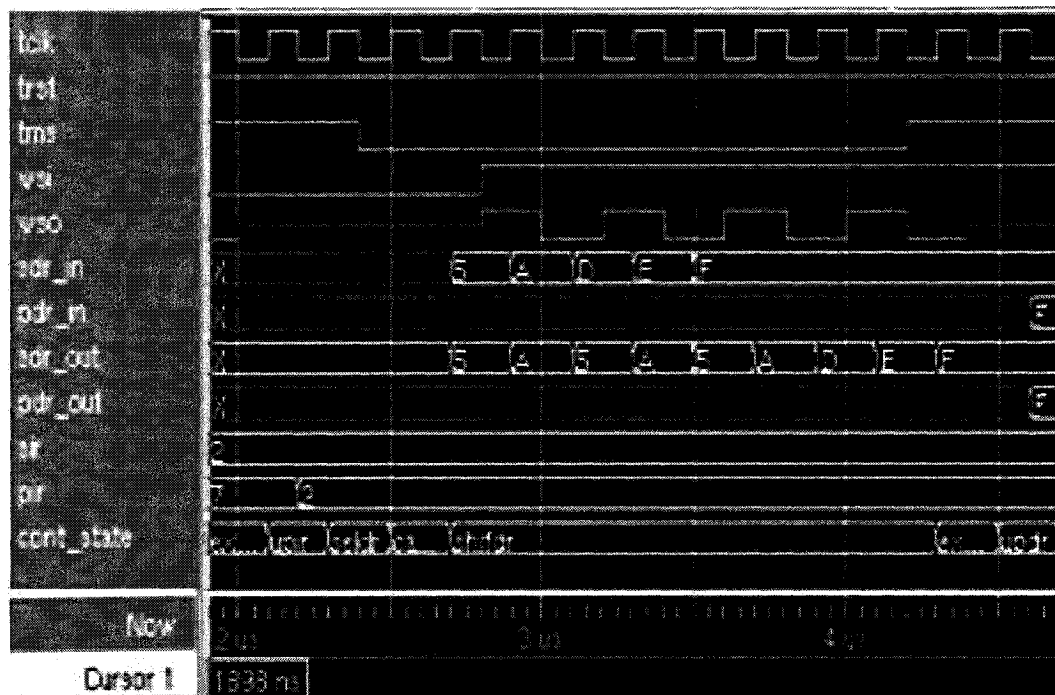


Figure 4.5: Chronogramme de l'instruction de test WPRELOADS.

Le chronogramme de la figure 4.5 est celui de l'instruction WPRELOADS qui précède toute instruction de test qui nécessite un pré-chargement de valeurs de test. Pour pré-charger la valeur hexadécimale F dans les étages *sdr_in* et *sdr_out* du registre WBR, nous appliquons l'instruction WPRELOADS. À l'état Shift_IR du contrôleur de test (*cont_state=shifir*), le code opération 2 de cette instruction présentée au tableau 3.2, est chargé dans l'étage *sir* de WIR. À l'état Update_IR du contrôleur de test (*cont_state=upir*), ce code est transféré dans l'étage *pir* de WIR pour configurer WBR en registre à décalage. À l'état Shift_DR du contrôleur (*cont_state=shifdr*) et à chaque cycle de l'horloge *tlck*, la valeur 1 de l'entrée WSI de l'enveloppe est décalée dans *sdr_in* et *sdr_out* de WBR. À l'état Update_DR du contrôleur de test (*cont_state=updr*), les valeurs chargées dans *sdr_in* et *sdr_out* sont transmises dans les étages parallèles *pdr_in* et *pdr_out* de WBR. Les valeurs de ces deux registres parallèles peuvent être appliquées ultérieurement à la logique interne du tampon et aux interconnexions à tester.

4.5 Chronogrammes des instructions de test avec BIST

Dans cette section, nous présentons les chronogrammes de simulation des instructions de test WEXBIST, WCOREBIST et WOPMISR qui peuvent être implémentées par la nouvelle architecture de test. Ces instructions permettent de test les cœurs d'IP et les interconnexions entre les IP à l'interne des SOC.

4.5.1 Instruction de test WEXBIST

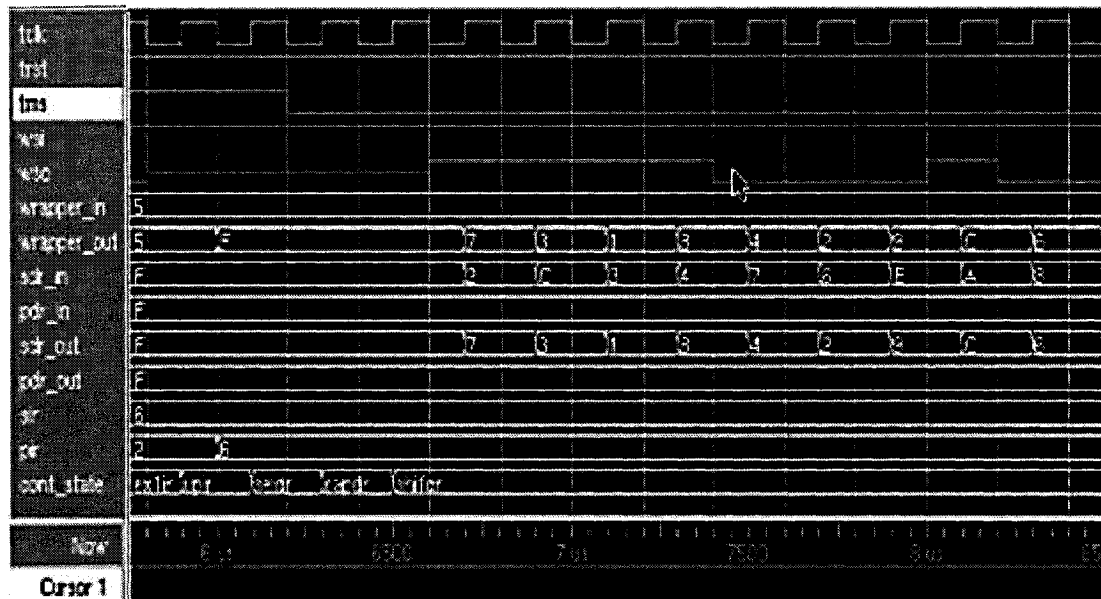


Figure 4.6: Chronogramme de l'instruction de test WEXBIST.

Le chronogramme de la figure 4.6 est celui de la simulation de l'instruction de test WEXBIST qui permet de tester les interconnexions entre les IP à l'interne du SOC. Avant d'appliquer cette instruction, nous avons initialisé les étages *sdr_in* et *sdr_out* à la valeur F avec l'instruction WPRELOADS. Sur le chronogramme, à l'état Shift_IR du contrôleur de test, le code opération 6 de WEXBIST du tableau 3.2 est chargé en série dans l'étage *sir* de WIR. À l'état Update_IR du contrôleur de test (*cont_state=upir*), ce code est mis à jour dans l'étage *pir* du WIR. Cette mise à jour de l'instruction, configure l'étage *sdr_in* en MISR et l'étage *sdr_out* en LFSR. À l'état Shift_DR (*cont_state=shifdr*), le LFSR génère un vecteur de test à chaque cycle de l'horloge de test *tclk*, tel qu'il est illustré à la figure 4.7. Au même temps, ces vecteurs sont appliqués en parallèle aux plots de sortie de l'enveloppe pour le test des interconnexions (*wrapper_out*). Avec un polynôme caractéristique adéquat, le LFSR formé par *sdr_out*

peut générer les vecteurs de test selon l'un de ces algorithmes : $(N+1)$ [35], $\log_2(n)$ [24], $\log_2(n+2)$ [13], $2\log_2(n)$ [8] et marche de 1 et 0 [21] développés pour le test des interconnexions. À l'état Shift_DR du contrôleur de test, l'étage *sdr_in* de l'enveloppe agit également en MISR pour compacter les réponses des interconnexions comme est illustré à la figure 4.8.

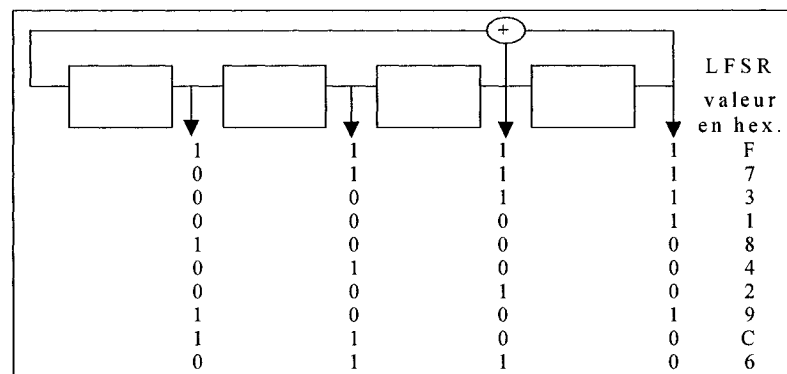


Figure 4.7: Séquence complète générée par un LFSR d'ordre 4 et $P=1+x^3+x^4$.

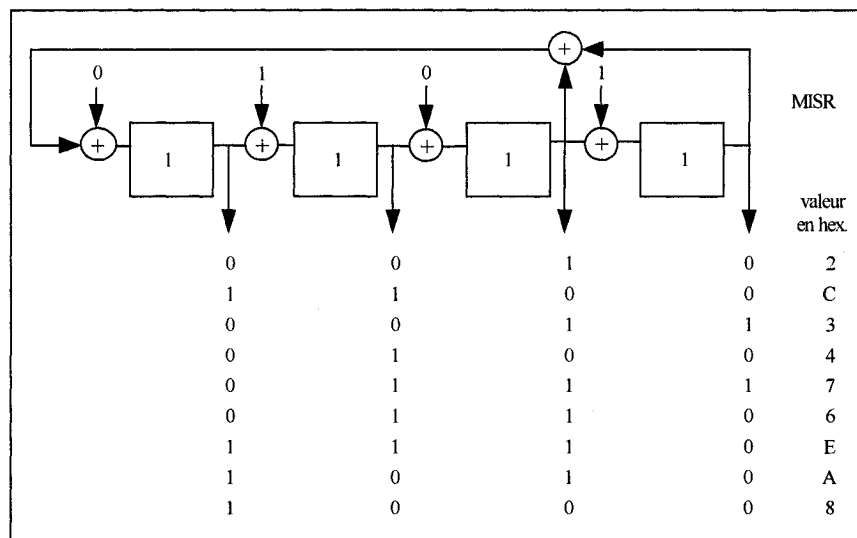


Figure 4.8: MISR à XOR externe initialisé à F.

4.5.2 Instruction de test WCOREBIST

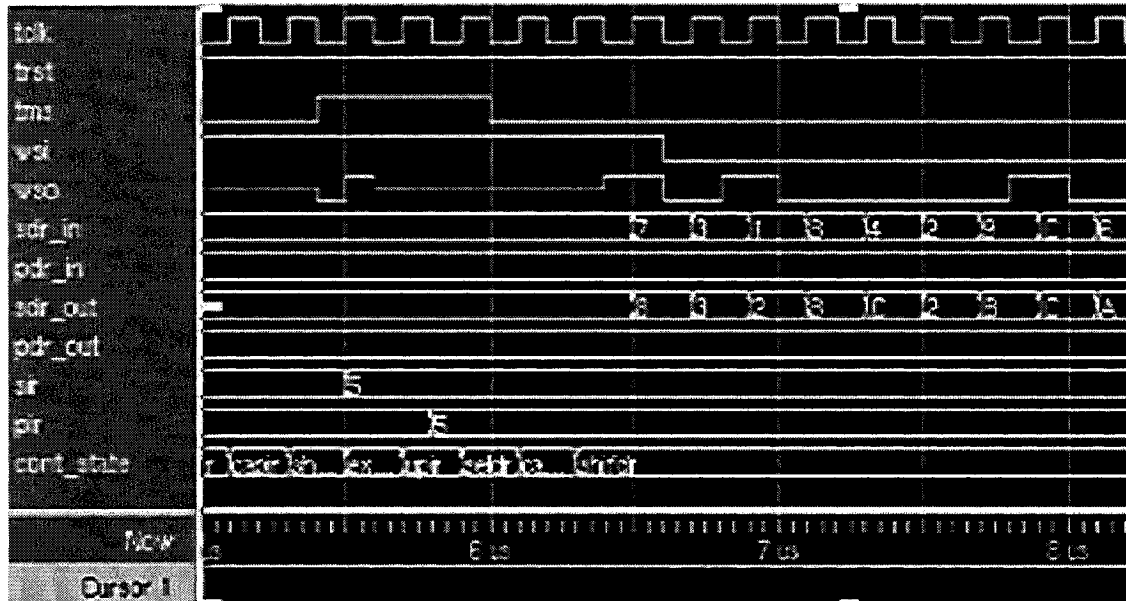


Figure 4.9: Chronogramme de l'instruction de test WCOREBIST.

Le chronogramme de la figure 4.9 est obtenu lors de la simulation de l'instruction de test WCOREBIST qui est prévue pour le test de la logique interne de l'IP à l'interne du SOC. Lors de cette simulation, avant d'appliquer l'instruction WCOREBIST, les étages *sdr_in* et *sdr_out* de l'enveloppe sont initialisés à F avec l'instruction WPRELOADS. Le chronogramme montre qu'à l'état Shift_IR du contrôleur de test (*cont_state=shifir*), le code opération 5 de WCOREBIST du tableau 3.2 est d'abord chargé dans l'étage *sir* de WIR. À l'état Update_IR du contrôleur (*cont_state=upir*), le contenu de *sir* est par la suite transféré à l'étage *pir* de WIR pour être appliqué aux entrées du décodeur de WIR. Les signaux de contrôle générés par le décodeur, configurent l'étage *sdr_in* en LFSR et l'étage *sdr_out* en MISR. À l'état Shift_DR du contrôleur de test (*cont_state=shifdr*), le

LFSR formé par l'étage *sdr_in* génère les vecteurs de test à appliqués aux entrées du tampon. Au même état du contrôleur de test, les réponses du tampon sont compactées par le MISR formé par l'étage *sdr_out*. Le principe de cette instruction de test WCOREBIST est illustré à la figure 4.10.

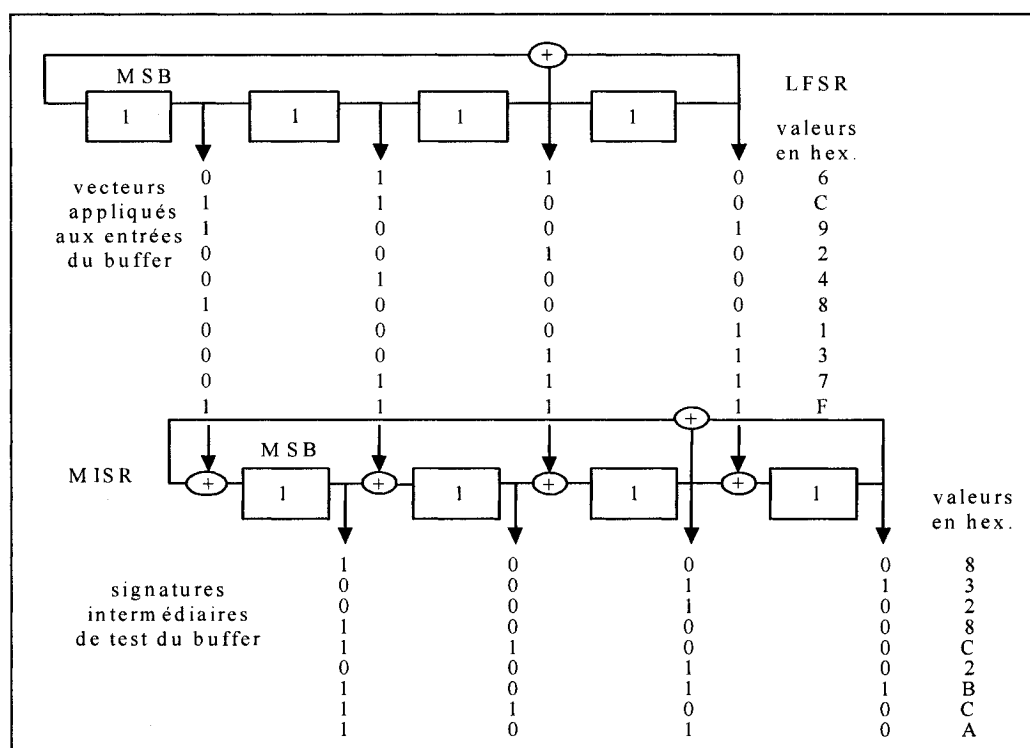


Figure 4.10: Génération des vecteurs de test et le compactage des réponses.

4.5.3 Instruction de test WOPMISR

Le chronogramme de la figure 4.11 est celui de la simulation de l'instruction de test WOPMISR, que nous avons prévu pour permettre à l'enveloppe proposée d'implémenter le mode test OPMISR employé la première fois par la compagnie IBM.

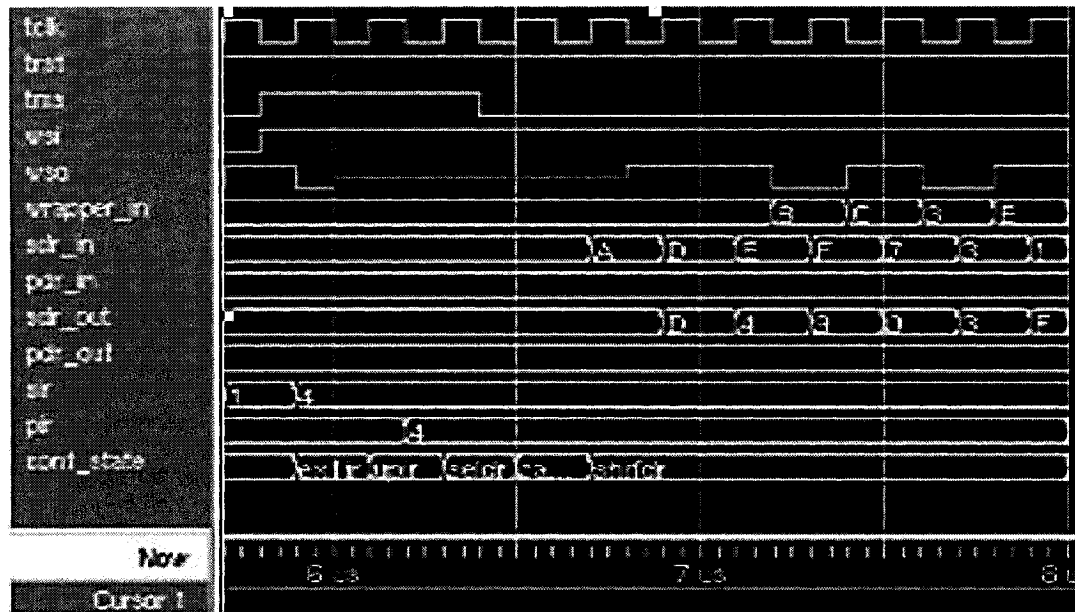


Figure 4.11: Chronogramme de l'instruction de test WOPMISR.

Avant d'appliquer l'instruction de test WOPMISR, nous avons initialisé l'étage *sdr_out* de l'enveloppe à F avec l'instruction WPRELOADS. Par la suite, à l'état Shift_IR du contrôleur de test (*cont_state=shifir*), le code opération 4 de WOPMISR du tableau 3.2 est d'abord chargé dans l'étage *sir* de WIR. À l'état Update_IR du contrôleur de test (*cont_state=upir*), ce code est mis à jour dans l'étage *pir* de WIR. Cette instruction permet de configurer l'étage *sdr_out* de l'enveloppe en MISR et de court-circuiter les cellules d'entrée de l'enveloppe. À partir du chronogramme, nous remarquons que les réponses du tampon aux vecteurs de test appliqués en parallèle à partir de l'extérieur via *wrapper_in* sont compactées par le MISR formé par l'étage *sdr_out*. La figure 4.12 montre le principe de WOPMISR où les données de test sont appliquées en parallèle au tampon à partir d'un testeur externe et les réponses sont compactées par un MISR à l'interne du SOC.

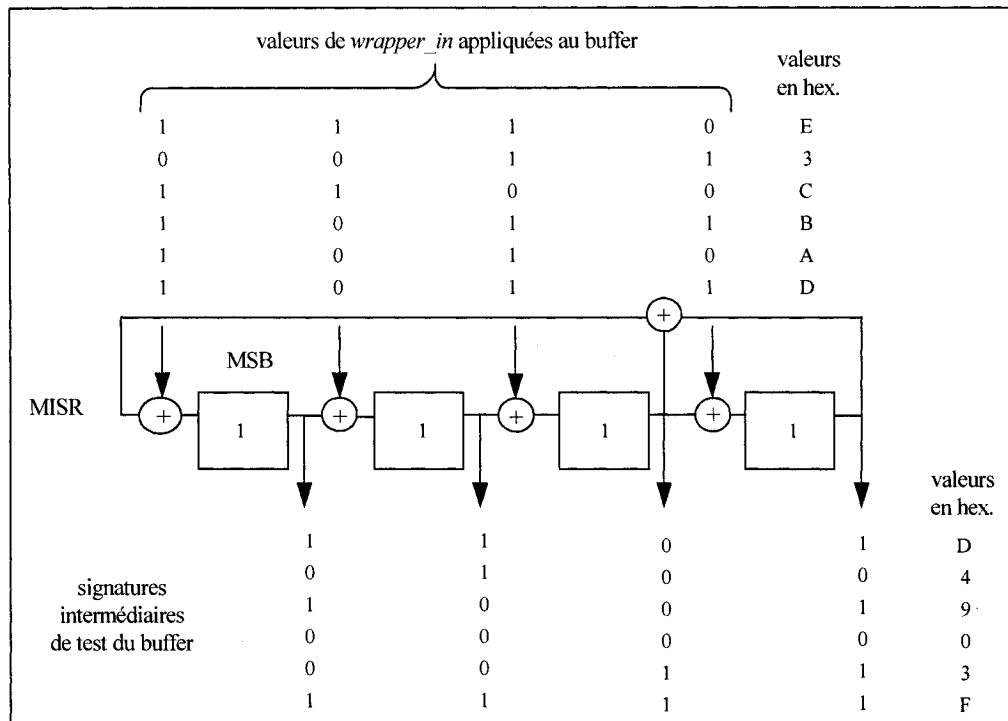


Figure 4.12: Illustration de l'instruction de test WOPMISR.

4.6 Délais de propagation

La figure 4.13 montre que la structure interne de la cellule de l'enveloppe proposée est différente de celle de l'enveloppe P1500. En mode normal, dans les deux cellules la valeur du signal d'entrée *cfi* est transmise au signal de sortie *cfo* à travers le multiplexeur *m*. Nous pouvons dire qu'en mode normal, les cellules de l'enveloppe proposée n'engendrent aucune dégradation des délais internes de propagation comparativement aux cellules de l'enveloppe P1500. En mode test, la valeur du signal d'entrée *cti* prend plus de temps pour se rendre à la sortie *cfo* dans la cellule de l'enveloppe proposée que dans celle de l'enveloppe P1500.

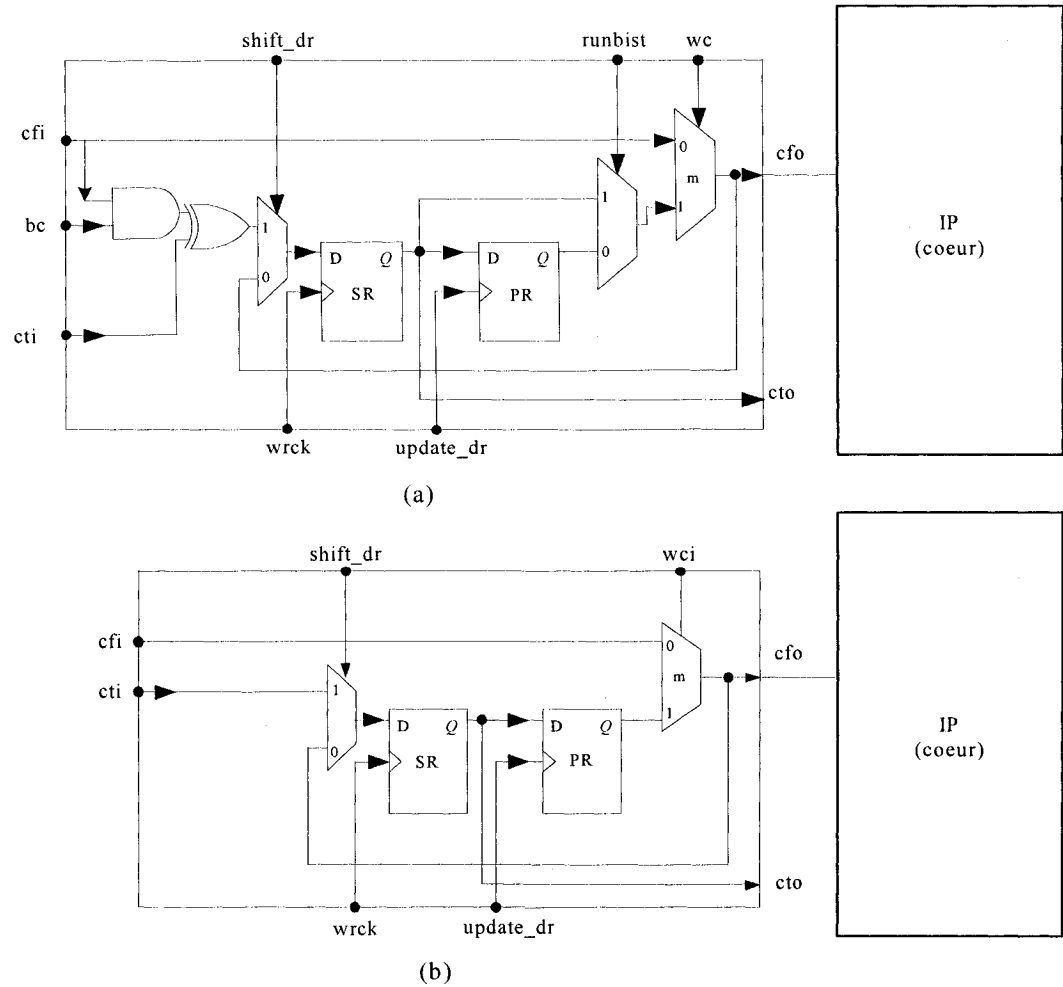


Figure 4.13: Cellule du registre WBR: a) enveloppe proposée; b) enveloppe P1500.

4.7 Estimation du gain en surface

Dans cette section, nous présentons les résultats du gain en surface obtenus avec l'architecture de test proposée comparativement à l'architecture de test conventionnelle. L'architecture de test conventionnelle est composée d'une enveloppe P1500 et d'une structure BIST. L'enveloppe P1500 permet de tester les IP d'un SOC à partir de

l'extérieur selon la norme IEEE P1500. La structure BIST permet la génération et l'application des vecteurs de test, ainsi que l'analyse des réponses à l'interne. Ceci permet de minimiser le temps d'utilisation des ATE externes. Avant d'exposer les résultats du gain en surface obtenus, nous présentons d'abord les circuits d'essai combinatoires et séquentiels que nous avons utilisé.

4.7.1 Présentation des circuits d'essai

Le tableau 4.1 montre les circuits d'essai, que nous avons choisi parmi les circuits combinatoires ISCAS85 [17] et les circuits séquentiels ISCAS89 [18] pour être utilisés pour la validation de l'architecture de test proposée. La première colonne du tableau présente les noms des circuits combinatoires et séquentiels. La deuxième et la troisième colonne indiquent le nombre de plots d'entrée/sortie fonctionnels de chaque circuit. La quatrième colonne donne la surface de chaque circuit d'essai en mm^2 obtenue avec la technologie $0,18\mu\text{m}$. Le plus petit circuit combinatoire du tableau est le C17 qui a 5 plots d'entrée, 2 plots de sortie et une surface de $0,00043 \text{ mm}^2$. Le plus grand circuit combinatoire utilisé est le C5315 qui a 178 plots d'entrée, 123 plots de sortie et une surface de $0,2454 \text{ mm}^2$. Parmi les circuits séquentiels présentés au tableau 4.1, le plus petit est le S510 qui a 18 plots d'entrée, 6 plots de sortie et une surface de $0,0263 \text{ mm}^2$. Par contre, S4863 est le plus grand circuit séquentiel que nous avons utilisé lors de la validation de l'architecture de test proposée. Ce circuit a 48 plots d'entrée, 15 plots de sortie et une surface de $0,340 \text{ mm}^2$.

Tableau 4.1: Circuits utilisés pour la validation de l'architecture de test proposée.

Circuits	Plots		Surface (mm ²)
	d'entrée	de sortie	
C17	5	2	0,00043
C432	36	7	0,01700
C499	41	32	0,02940
C1908	33	25	0,07280
C2670	233	31	0,12000
C5315	178	123	0,24540
S510	18	6	0,02630
S991	64	16	0,06720
S1423	16	4	0,09950
S1512	28	20	0,10900
S3271	25	13	0,21900
S4863	48	15	0,34000

4.7.2 Résultats du gain en surface

Le tableau 4.2 donne les surfaces des circuits d'essai du tableau 4.1 avant et après leur encapsulation dans la nouvelle architecture de test et dans l'architecture de test conventionnelle. Ces surfaces sont obtenues après placement routage des circuits encapsulés en technologie 0,18 μ m avec l'outil Encounter de Cadence.

La colonne 2 du tableau présente le nombre total de plots fonctionnels de chaque circuit essai utilisé. La colonne 3 donne la surface de chaque circuit d'essai avant son encapsulation. La colonne 4 donne le rapport entre le nombre de plots d'entrée/sortie et la surface de chaque circuit avant encapsulation. Les colonnes 5 et 6 montrent respectivement les surfaces des circuits après encapsulation dans l'enveloppe de test

proposée et dans l'architecture de test conventionnelle. La colonne 7 présente le gain en surface qui est le quotient de la division de la différence de surface entre les circuits de la colonne 5 et 4 sur la surface des circuits de la colonne 5. Le plus grand gain en surface obtenu est estimé à 9.87%, il a été réalisé avec le circuit combinatoire C17. Le plus faible gain en surface obtenu est de 0.84%, il a été réalisé avec le circuit séquentiel S4863.

Tableau 4.2: Résultats du gain en surface.

IP	Plots	Surface IP	IP/plot	Surface (mm2)		Gain (%)
				IP+Wrap. Prop.	IP+P1500+BIST	
C17	7	0,0004	17500,00	0,0037	0,0041	9,87
C432	43	0,017	2529,41	0,0285	0,0303	5,94
C1908	58	0,0728	796,70	0,1119	0,1159	3,45
C499	73	0,0294	2482,99	0,0477	0,0505	5,54
C2670	264	0,12	2200,00	0,2010	0,2110	4,74
C5315	301	0,2454	1226,57	0,2632	0,2752	4,36
S1423	20	0,0995	201,01	0,1000	0,1020	1,96
S510	24	0,0263	912,55	0,0310	0,0340	8,82
S3271	38	0,219	173,52	0,2227	0,2251	1,07
S1512	48	0,109	440,37	0,1178	0,1204	2,16
S4863	63	0,34	185,29	0,3428	0,3457	0,84
S991	80	0,0672	1190,48	0,0851	0,0891	4,49

En analysant les résultats du gain en surface figurant à la colonne 7 du tableau 4.2, nous avons constaté que le gain en surface est proportionnel au rapport entre le nombre de plots d'entrée/sortie et la surface du circuit. Le graphe de la figure 4.14 illustre clairement la variation du gain en surface versus le rapport plots/surface des circuits. Il

est donc plus avantageux d'appliquer l'architecture de test proposée sur des circuits qui présentent un rapport plots/surface élevé.

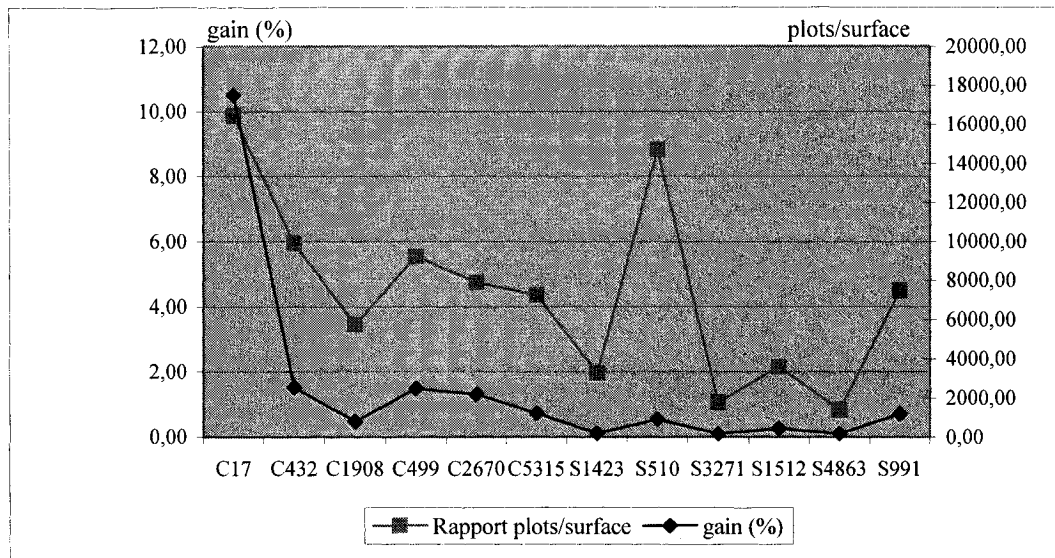


Figure 4.14: Gain en surface versus le rapport (plots/surface) du circuit.

4.8 Réduction du nombre de vecteurs de test déterministes

Les résultats du tableau 4.3 montrent que la nouvelle architecture qui combine le test aléatoire et le test déterministe, assure une qualité de test semblable à celle d'une architecture complètement déterministe. Dans la colonne 1 du tableau figure les noms des circuits d'essai d'ISCAS85 utilisés. La colonne 2 donne le nombre de pannes possibles dans chaque circuit. La colonne 3 donne le nombre de vecteurs de test déterministes générés par l'ATPG Tetramax de Synopsys suivant le modèle de faute de collage et de court-circuit. Ces vecteurs déterministes assurent un taux de couverture

(TC) de 100% pour les circuits de la colonne 1. Le taux de couverture représente le rapport entre le nombre de pannes détectées et le nombre total de pannes dans le circuit. La colonne 4 présente le nombre de vecteurs de test aléatoires générés par l'ATPG Tetramax de Synopsys et appliqués en premier lieu aux circuits de la colonne 1. Le nombre de vecteurs aléatoires appliqués à chacun des circuits est déterminé par les paramètres de l'ATPG Tetramax et par la difficulté à tester ces circuits. La colonne 5 donne le nombre de vecteurs déterministes complémentaires à appliquer aux circuits pour atteindre un TC de 100%. La colonne 7 du tableau présente le taux de réduction du nombre de vecteurs de test déterministes à appliquer aux circuits. Dans le cas des circuits C432 et C17, le TC de 100% peut être réalisé uniquement avec des vecteurs de test aléatoires. Pour le reste des circuits, la réduction du nombre de vecteurs de test déterministes varie entre 33.4% et 39.5%.

Tableau 4.3: Réduction du nombre de vecteurs de test déterministes.

IP	Nombre de pannes	Test det.	Test mixte		TC (%)	Réduction du nombre de vecteurs déterministes (%)
			BIST	Det.		
C432	86	26	82	0	100	100
C17	14	7	12	0	100	100
C5315	602	38	400	23	100	39.5
C2670	520	45	200	30	100	33.4
C1908	116	16	40	10	100	37.5
C499	146	20	65	13	100	35

4.9 Conclusion

La simulation fonctionnelle a montré que la nouvelle architecture agit conformément aux spécifications des instructions de test prévues pour les IP et les interconnexions des SOC. Le placement routage des circuits d'essai encapsulés dans l'architecture de test proposée et dans l'architecture conventionnelle a été réalisé pour estimer et comparer les surfaces de ces circuits encapsulés. Cette comparaison a montré que l'architecture de test proposée engendre moins de surface additionnelle comparativement à l'architecture de test conventionnelle. La combinaison des deux modes de test P1500 et BIST dans une même structure permet d'atteindre une qualité de test comparable à celle du test déterministe mais avec moins de vecteurs de test déterministes. Ceci offre la possibilité aux intégrateurs d'utiliser des ATE à faibles espaces mémoires et de réduire le temps de leur utilisation. Ces avantages peuvent contribuer à la réduction du coût de test des SOC.

CHAPITRE 5

CONCLUSION ET TRAVAUX FUTURS

5.1 Conclusion générale

Les IP permettent de concevoir des SOC complexes, mais ils ne facilitent pas le test de ces systèmes avec les techniques de test conventionnelles. Cette difficulté est due à la provenance hétérogène des IP et à l'inaccessibilité à leurs plots d'entrée/sortie à partir de ceux des SOC. Pour assurer une bonne qualité de test pour les SOC et réduire leurs coûts de test, plusieurs architectures de test ont été proposées. Malheureusement, ces architectures ne permettent pas de tester les IP et les interconnexions avec une même structure de test. Pour tester un SOC, il faut donc prévoir des architectures de test pour les IP et d'autres pour les interconnexions. Ceci engendre une augmentation de la surface du semi-conducteur ainsi qu'une importante dégradation des délais internes.

Dans ce mémoire, nous avons exposé en détail la structure et le fonctionnement de la nouvelle architecture de test que nous avons proposé pour les SOC. Cette architecture est une enveloppe que nous insérons entre les signaux d'entrée/sortie du cœur de l'IP et les plots de l'IP. Lorsque le test des cœurs des IP et des interconnexions est effectué avec des ATE externes, la nouvelle enveloppe agit en structure de test conforme aux spécifications de la norme IEEE P1500. Lorsque le test des cœurs des IP et des

interconnexions est réalisé à l'interne du SOC, l'enveloppe implémente une structure BIST conformément aux spécifications des instructions de test que nous avons prévu pour ce mode. Pour réaliser cette nouvelle enveloppe où chaque signal d'entrée/sortie du cœur de l'IP est associé à une cellule, nous avons conçu des cellules à deux états, à trois états et bidirectionnelles avec ou sans porte XOR. Le chaînage en série de ces cellules forme le registre de données WBR de l'enveloppe. Pour réaliser les registres d'instruction et de court-circuit de l'enveloppe proposée, nous avons réutilisé ceux du Boundary Scan. En ce qui concerne le contrôleur de la nouvelle architecture de test qui gère l'ensemble des opérations, nous avons également réutilisé celui du Boundary Scan qui est utilisé par la norme IEEE P1500.

La validation de l'architecture de test proposée, a été réalisée par simulation. En premier lieu, nous avons effectué des tests fonctionnels avec l'outil de simulation ModelSim de Mentor Graphics. Ces tests fonctionnels ont été réalisés pour vérifier si l'enveloppe implémente correctement les instructions de test prévues pour les IP et les interconnexions des SOC. Pour accomplir ces tests fonctionnels, nous avons synthétisé la nouvelle enveloppe autour un tampon à 4 bits avec l'outil Design Analyzer de Synopsys. Nous avons aussi élaboré en VHDL un banc de test pour chaque instruction de test. Après le test fonctionnel, nous avons procédé à l'analyse temporelle statique des cellules de l'enveloppe proposée avec l'outil Design Analyzer. Cette analyse temporelle a montré que les dégradations des délais internes engendrées par la nouvelle enveloppe sont comparables à celles introduites par l'enveloppe P1500. À la dernière étape de la

validation, nous avons comparé les surfaces de certains circuits d'essai d'ISCAS85 et d'ISCAS89 encapsulé dans l'enveloppe proposée à celles des mêmes circuits mais encapsulés dans la structure de test conventionnelle. La structure de test conventionnelle est composée d'une enveloppe P1500 et d'une structure BIST. Pour estimer les surfaces des circuits encapsulés, nous avons placé et routé ces circuits avec l'outil Encontour de Cadence. La comparaison entre les surfaces de ces circuits a montré que l'enveloppe proposée engendre moins de surface additionnelle que l'architecture de test conventionnelle. Cette réduction de la surface peut dépasser les 9% avec les circuits qui présentent un rapport plots/surface important.

En conclusion, nous estimons que l'architecture de test proposée offre à l'intégrateur la possibilité de réaliser un bon compromis entre la qualité du test, le coût de test et les dégradations des délais internes.

5.2 Travaux futurs

L'encapsulation manuelle des IP complexes dans l'architecture de test proposée prend beaucoup de temps et les erreurs sont inévitables. Toutefois, avec une insertion automatisée, il est possible de réaliser cette opération en un temps court et sans erreurs. Il est donc essentiel de développer un outil d'insertion automatique pour l'architecture de test proposé afin de faciliter son utilisation. Cet outil doit générer en VHDL structurel le fichier de l'IP encapsulé en partant du fichier source de l'IP et du fichier « Core Test

Language » (CTL) de l'IP. L'outil d'insertion automatique doit lire le CTL pour identifier le type des signaux d'entrée/sortie de l'IP et le polynôme caractéristique de la structure BIST que l'enveloppe doit implémenter. Par la suite, l'outil doit assigner à chaque signal d'entrée/sortie la cellule qui lui correspond parmi celles proposées dans ce mémoire. En dernier, l'outil doit ajouter le registre d'instruction et le registre de court-circuit pour compléter la structure de l'enveloppe.

RÉFÉRENCES

- [1] AGRAWAL V.D., KIME C.R., SALUJA K.K., “*A Tutorial on Built-In Self-Test, Part 1: Theory*”, (1993), IEEE Design and Test of Computers, Los Alamitos, CA, USA: IEEE Computer Society Press, Vol. 10, P. 73-82.
- [2] AGRAWAL V.D., KIME C.R., SALUJA K.K., “*A Tutorial on Built-In Self-Test, Part 2: Applications*”, (1993), IEEE Design and Test of Computers, Los Alamitos, CA, USA: IEEE Computer Society Press, Vol. 10, P. 69-77.
- [3] AKERS S., JANSZ W., “*Test Set Embedding in a Built-In Self-Test Environment*”, (1989), IEEE International Test Conference, Washington, DC, USA: IEEE Computer Society, P. 257-263.
- [4] ARM Web Site. AMBA Specifications. <http://www.arm.com/armtech/>.
- [5] ASHOK A. SETTY, HAROLD L., “*BIST and Interconnect Testing with Boundary Scan*”, (1991), IEEE Southeast Conference, Vol1, P. 12-15.
- [6] BANHART C., BRUNKHORST V., DISTER F., FARNSWORTH O., KELLER B., KOENEMANN B., “*OPMISR: The Foundation for Compressed ATPG Vectors*”, (2001), IEEE International Test Conference, P. 748-757.
- [7] BENABDENBI M., MAROUFI W., MARZOUKI M., “*CAS-BUS: A Scalable and Reconfigurable Test Access Mechanism for Systems on a Chip*”, (2000), IEEE Design Automation and Test in Europe, Conference, Paris, France, P. 141-145.

- [8] CHENG W.T., LEWANDOWSKI J.L., WU E., "*Optimal Diagnostic Methods for Wiring In Interconnects*", (1992), IEEE Transactions on Computer Aided Design, Vol. 11, No 9, P. 1161-1166.
- [9] FRANCIS C., "*Digital Circuit Testing A Guide to DFT and Other Techniques*", (1991), Édition: Academic Press Inc.
- [10] FUJIWARA H., SHIMONO T., "*On the Acceleration of Test Generation Algorithms*", (1983), IEEE Transactions on Computers, Vol. C-32, N° 12, P. 1137-1144.
- [11] GHOSH I., JHA N.K., DEY S., "*A Low Overhead Design for Testability and Generation Technique for Core-Based Systems*", (1997), IEEE International Test Conference: IEEE Computer Society Press, P. 50-79.
- [12] GHOSH I., JHA N.K., DEY S., "*A Fast and Low Cost Testing Technique for Core-Based System-on-Chip*", (1998), IEEE Design Automation conference, San Francisco: Association for Computing Machinery Inc, P. 542-547.
- [13] GOEL P., MCMAHON M.T., "*Electronic Chip-In-Place Test*", (1982), IEEE International Test Conference, P. 83-90.
- [14] GOEL P., "*An Implicit Enumeration Algorithm to Generate Tests for Combinational Logic Circuits*", (1981), IEEE Transactions on Computers, Vo. C-30, N° 3, P. 215-222.
- [15] <http://www.mentor.com/products/dft/boundaryscan/index.cfm>
- [16] <http://www-asim.lip6.fr/~mounir/>, université Paris VI.27 septembre 2002.

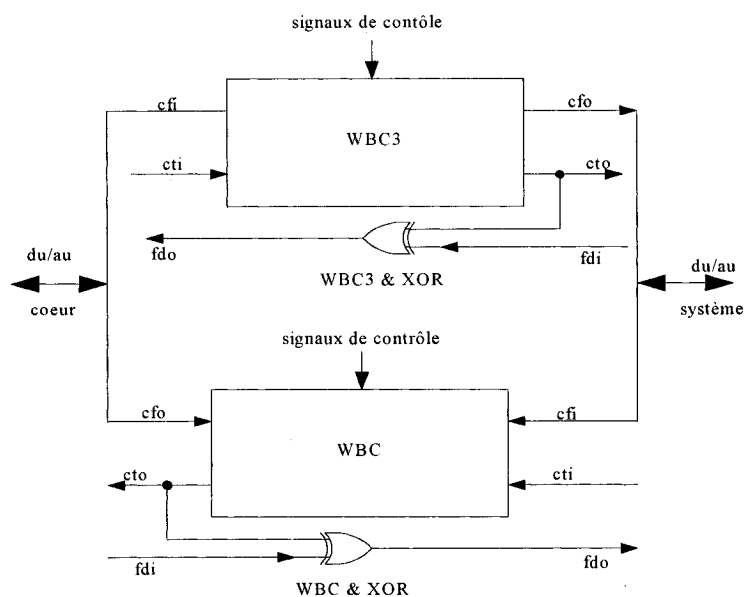
- [17] http://www.pldworld.com/_hdl/1/rassp.aticorp.org/vhdl/models/misc/ISCAS/ISCAS85/index.htm
- [18] http://www.pldworld.com/_hdl/1/rassp.aticorp.org/vhdl/models/misc/ISCAS/ISCAS89/index.htm
- [19] IEEE 1149.1 Web Site, <http://group.ieee.org/groups/group1149.1>.
- [20] IEEE P1500 Web Site, <http://group.ieee.org/groups/group1500>
- [21] JARWALA N., YAU C.W., “*A New Framework for Analyzing and Diagnosis Algorithms for Wiring Interconnects*”, (1989), IEEE International Test Conference, P. 63-70.
- [22] JHA N., GUPTA S., “*Testing of Digital Systems*”, (2003), Édition: Cambridge University Press, New York.
- [23] KAGARIS D., TRAGOUDAS S., MAJUMDAR A., “*Deterministic Test Pattern Production by a Counter*”, In Proceedings ED&TC, P. 37-41, 1996.
- [24] KAUTZ W.K., “*Testing of Faults in Wiring Interconnects*”, (1974), IEEE Transactions on Computers, Vol. C-23, P. 358-363.
- [25] KIEFER G.; WUNDERLICH H. J., “*Deterministic BIST with Multiple Scan Chains*”, (1998), International Test Conference, P. 1057-1064.
- [26] KONEMANN B., MUCHA J., ZWIEHOFF G., (1980), “*Built-In Test for Complex Digital Integrated Circuits*”, IEEE Journal of Solid-State Circuits, Vol. SC-15, No. 3, P. 315-319.

- [27] KUNZMANN A., “*Optimized Self-test for Deterministic Test Patterns Using Random Test Patterns*”, (1993), IFIP Workshop on Logic and Architecture Synthesis, P. 433-438.
- [28] LARAB A., KHOUAS A., “*NOUVEAU WRAPPER P1500 INCORPORANT UNE STRUCTURE BIST POUR LE TEST DES IP ET DES INTERCONNEXIONS D’UN SOC*”, (2005), IEEE Canadian Conference on Electrical and Computer Engineering, Saskatoon, P. 1902-1905.
- [29] MARCUS H., “*IEEE P1500 The Standard for Embedded Core Test*”,
http://hem.hj.se/~mabe/Master_2002/SysVerif_Test/IEEE_P1500.pdf
- [30] MARINISSEN E.J., ZORIAN Y., KAPUR R., TAYLOR T., WHETSEL L.,
 “*Towards a Standard for embedded core Test: An Example*”, (1999), IEEE International Test Conference, Atlantic City, P. 1-12.
- [31] MARINISSEN E.J., KAPUR R., LOUSBERG M., McLAURIN T.,
 MRICCHETTI M., ZORIAN Y., “*On IEEE P1500’s Standard for Embedded Core Test*”, (2002), Journal of Electronic Testing: Theory and Applications, Netherlands: Kluwer Academic Publishers. P. 365-383.
- [32] MARINISSEN E.J., ARENDSSEN R., “*A Structured and Scalable Mechanism for Test Access to Embedded Reusable Cores*”, (1998), Proceedings IEEE International Test Conference, Washington, DC, USA: IEEE Computer Society. P. 284-293.

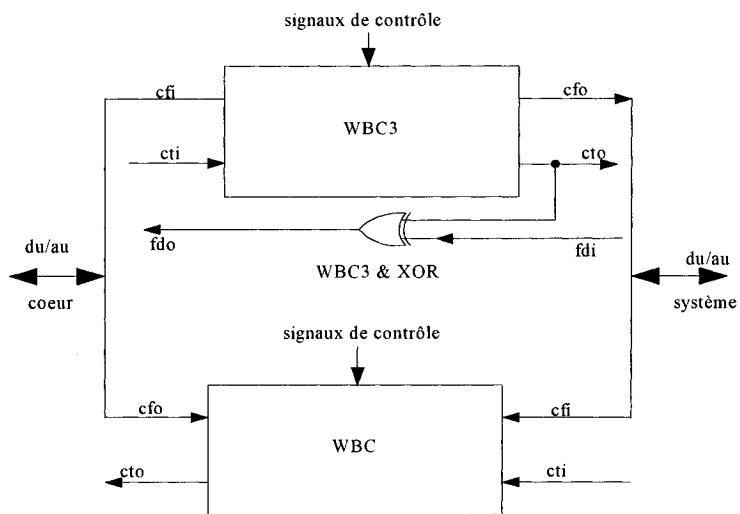
- [33] MICHAEL L., BUSHNELL and VISHAWANI D., "*Essentials of Electronic Testing for Digital, Memory & Mixed-Signal VLSI Circuits*", (2000), Édition: KLUWER ACADEMIC PUBLISHERS.
- [34] MOURAD S., ZORIAN Y., "*Principles of Testing Electronic Systems*" (2000), Édition: John Wiley & Sons.
- [35] PARK S., "*A New Complete Diagnosis Patterns for Wiring Interconnects*", (1996), Proceedings IEEE Design Automation Conference, Las Vegas, NV, USA.. P. 203-208.
- [36] RAJESH P., ABHIJIT C., ZORIAN Y., "*A Distributed BIST Technique for Diagnosis Of MCM Interconnections*", (1998), In Proceedings IEEE International Test Conference. P. 214-221.
- [37] ROTH J. P., "*Diagnosis of Automata Failures: A Calculus and a Method*", (1966), IBM Journal of Research and Development, P. 278-291.
- [38] RAJSKI J., TYSER J., "*Arithmetic Built-In Self-Test for Embedded Systems*", (1998), Édition: Upper Saddle River, New Jersey.
- [39] SAMAN A., DWAYNE B., "*Preliminary Outline of IEEE P1500 Scalable Architecture for testing Embedded Cores*", (1999), VLSI Test Symposium. P. 483-488.
- [40] STANLEY L. Hurst, "*VLSI Testing Digital and Mixed Analogue/Digital Techniques*", (1998), IEE Circuits, Devices and systems series 9, Édition: The Institution of Electrical Engineers.

- [41] SAVARIA Y., “*Conception et vérification des circuits VLSP*”, (1993), Édition: École Polytechnique de Montréal.
- [42] THE National Technology Roadmap for semiconductors (ITRS), (1999), <http://www.sia-online.org>.
- [43] TOUBA N. A., McCLUSKEY E. J., “*Transformed Pseudo-Random Patterns for BIST*”, (1995), VLSI Test Symposium. P. 410-416.
- [44] VENKATA I., SRINIVAS R., “*Direct Access Test Scheme-Design of Block and Core Cells for Embedded ASICs*”, (1990), IEEE International Test Conference. P. 488-492.
- [45] ZORIAN Y., MARINISSEN E. J., SUJIT D., “*Testing Embedded-Core-Based System Chips*”, (1998), International Test Conference, Washington: IEEE Computer Society, P. 130-143.

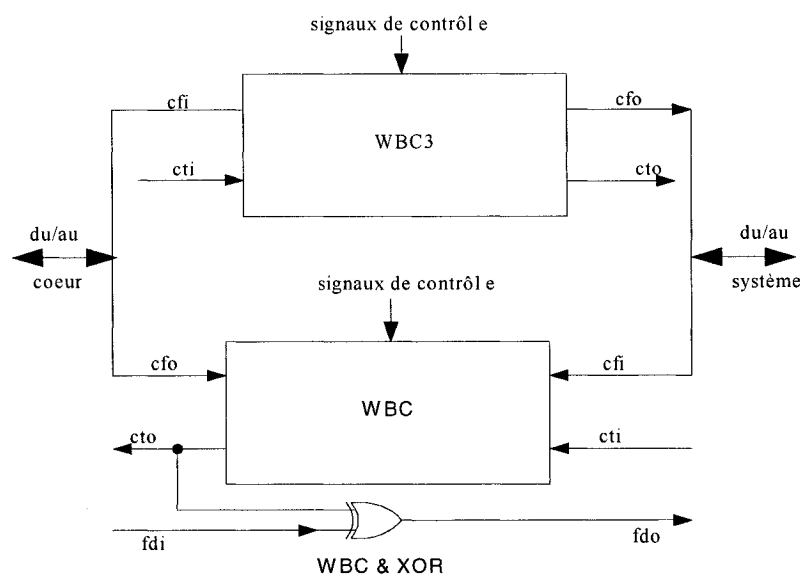
ANNEXE



a) Cellule bidirectionnelle formée par les cellules WBIC&XOR et WBOC3&XOR.



b) Cellule bidirectionnelle formée par les cellules WBIC et WBOC3&XOR.



c) Cellule bidirectionnelle formée par les cellules WBC&XOR