

## TABLE DES MATIÈRES

	Page
INTRODUCTION .....	15
CHAPITRE 1    REVUE DE LITTÉRATURE .....	17
1.1    Signature basée sur le type de cellules présumées défectueuses .....	19
1.2    Signature utilisant la disposition logique des bascules à sorties erronées .....	21
1.3    Signature utilisant les coordonnées physiques des bascules à sorties erronées .....	22
1.4    Signatures comparatives génériques générées à partir de la séparation des données de test .....	24
1.5    Autres travaux .....	25
1.6    Signatures proposées dans ce mémoire .....	26
1.7    Conclusions .....	26
CHAPITRE 2    CHOIX DU PROJET– FAMILLE TECHNOLOGIQUE CMOSP18, CIRCUITS DE RÉFÉRENCE b02 & b12 .....	27
2.1    Introduction .....	27
2.2    Technologie cible .....	27
2.3    Méthodologie cible .....	28
2.4    Circuits de référence b02 et b12 .....	29
2.5    Conclusions .....	31
CHAPITRE 3    ÉTAPES PRÉPARATOIRES - CONCEPTION D’UN CIRCUIT ASIC, TRAITEMENTS EFFECTUÉS PAR FASTSCAN ET AWK .....	32
3.1    Introduction .....	32
3.2    Diagramme de flot classique pour la conception d’un circuit ASIC .....	32
3.3    Diagramme de flot modifié de conception d’un circuit ASIC .....	33
3.4    Conclusions .....	39
CHAPITRE 4    PREMIÈRE MÉTHODE – SIGNATURE D’UN CIRCUIT BASÉE SUR LES NŒUDS COUVERTS PAR LES BASCULES D’OBSERVATION .....	40
4.1    Introduction .....	40
4.2    Définition de la première signature .....	40
4.3    Traitement spécifique pour l’obtention de la signature .....	44
4.4    Conclusions .....	47
CHAPITRE 5    DEUXIÈME MÉTHODE – LA SIGNATURE DU CIRCUIT PONDÉRÉE PAR LES CAPACITÉS PARASITES DES NŒUDS .....	48
5.1    Introduction .....	48
5.2    Définition de la deuxième signature .....	48
5.3    Traitement spécifique pour l’obtention de la signature .....	51
5.4    Conclusions .....	51

CHAPITRE 6	TROISIÈME MÉTHODE - SIGNATURE DU CIRCUIT PONDÉRÉE PAR LA RÉPARTITION DES PANNES SUR LES COUCHES MÉTALLIQUES .....	53
6.1	Introduction.....	53
6.2	Définition de la troisième signature.....	53
6.3	Traitement spécifique pour l'obtention de la signature .....	54
6.4	Conclusions.....	56
CHAPITRE 7	QUATRIÈME MÉTHODE-SIGNATURES DU CIRCUIT PONDÉRÉE PAR LES DÉPASSEMENTS DES RÈGLES DE DRC.....	57
7.1	Introduction.....	57
7.2	Définition du quatrième type de signatures (commun aux 3 variantes) .....	58
7.3	Traitement spécifique pour l'obtention des 3 variantes du quatrième type de signatures.....	64
7.3.1	Signature par la commande d'analyse des pannes (analyse faults) – méthode 4a .....	64
7.3.1.1	Usage pratique de la méthode 4a (première variante) .....	68
7.3.1.2	Sommaire pour la méthode 4a (première variante).....	69
7.3.2	Signature par la commande « ajout de listes (add lists) » - méthode 4b (deuxième variante).....	69
7.3.2.1	Particularités de l'utilisation du Fastscan dans la signature proposée .....	70
7.3.2.2	Sommaire pour la méthode 4b (deuxième variante) .....	71
7.3.3	Traitement spécifique pour l'obtention de la signature pour le test rapide, arrêt à la première défectuosité détectée - méthode 4c (troisième variante). .....	73
7.3.3.1	Sommaire pour la méthode 4c (troisième variante).....	76
7.4	Comparaisons entre toutes les signatures .....	76
CONCLUSIONS.....		79
RECOMMANDATIONS .....		81
ANNEXE I	CODE VHDL DU CIRCUIT B12 .....	82
ANNEXE II	CADENCE - DÉTAILS DE CONCEPTION DU DESSIN DES MASQUES POUR CIRCUIT B12 .....	95
ANNEXE III	FASTSCAN - ANALYSE INTERNE DU CIRCUIT B12 .....	98
ANNEXE IV	B12_CHIP.DEF (FRAGMENT).....	103
ANNEXE V	DESCRIPTION DÉTAILLÉE DU TRAITEMENT DE LA MÉTHODE 1, LA SIGNATURE À PONDÉRATION CONSTANTE (EQUIPROBABLE).....	108

ANNEXE VI	EXPLICATION DÉTAILLÉE SUR LA NÉCESSITÉ D'INTRODUCTION DE NMNC ET LA COUVERTURE D'UN NŒUD DE CADENCE PAR PLUSIEURS BASCULES D'ANALYSE INTERNE .....	117
ANNEXE VII	DESCRIPTION DÉTAILLÉE DES TRAITEMENTS SPÉCIFIQUES DE LA MÉTHODE 2 .....	119
ANNEXE VIII	DESCRIPTION DÉTAILLÉE DES TRAITEMENTS SPÉCIFIQUES DE LA MÉTHODE 3 .....	122
ANNEXE IX	EXEMPLE DE CALCUL DE LA SURFACE RÉELLE D'UN POLYGONE DRC .....	126
ANNEXE X	DESCRIPTION DÉTAILLÉE DES TRAITEMENTS SPÉCIFIQUES DE LA MÉTHODE 4A.....	128
ANNEXE XI	LES RÈGLES DE DESIGN POUR LA FAMILLE CMOSP18 QUI N'ONT PAS DONNÉ DE VIOLATIONS .....	142
ANNEXE XII	FICHIER DRC OBTENU DE CADENCE PAR LE DÉPASSEMENT DE R4 (LA SÉPARATION ENTRE CONNEXIONS SUR LA COUCHE M5) .....	145
ANNEXE XIII	DESCRIPTION DÉTAILLÉE DES TRAITEMENTS SPÉCIFIQUES DE LA MÉTHODE 4B.....	151
ANNEXE XIV	FICHIERS UTILISÉS DANS LA MÉTHODE 4B .....	158
ANNEXE XV	DESCRIPTION DÉTAILLÉE DES TRAITEMENTS SPÉCIFIQUES DE LA MÉTHODE 4C.....	163
ANNEXE XVI	GRAPHE RÉCAPITULATIF DES SIGNATURES (VALEURS NUMÉRIQUES) .....	169
BIBLIOGRAPHIE.....		170

## LISTE DES TABLEAUX

		Page
Tableau 2.1	Règles pour le dessin des masques des cellules de la bibliothèque de $0.18\ \mu\text{m}$ ; $\lambda = 0,09\mu\text{m}$ .....	28
Tableau 2.2	Caractéristiques des circuits utilisés b02 et b12 .....	30
Tableau 3.1	Rapport statistique d'analyse des pannes du Fastscan pour le circuit b12 .....	38
Tableau 4.1	Circuit fictif et l'analyse de Fastscan pour ses bascules .....	42
Tableau 4.2	Coefficients de testabilité sur les nœuds du circuit de notre cas d'étude .....	43
Tableau 4.3	La sensibilité des nœuds de la zone d'intérêt pour notre cas d'étude .....	43
Tableau 7.1	Règles de conception du dessin des masques, considérées pour le calcul de la pondération des nœuds .....	60
Tableau 7.2	Règles considérées et résultats obtenus pour le circuit b12 .....	63

## LISTE DES FIGURES

	Page
Figure 1-1	Exemples des pannes systématiques (a) et aléatoires (b).....18
Figure 1-2	Exemple de procédure de création de signatures.....19
Figure 1-3	Les types de cellules en fonction du nombre des apparitions en défectuosité. ....20
Figure 1-4	Identification des zones sensibles du dessin des masques suite à une analyse de la surface critique au niveau de la puce (les points chauds sont noircis). ....20
Figure 1-5	Signature anticipée pour les premières 15 cellules prédisposées aux pannes créée à partir de l'analyse de la figure antérieure. ....21
Figure 1-6	Un mappage logique par des bascules. ....22
Figure 1-7	Un mappage logique par des chaines des bascules. ....22
Figure 1-8	Un exemple de signature composite .....23
Figure 1-9	Utilisation des coordonnées x-y dans la signature. ....23
Figure 1-10	Signature correspondant à la Figure 1-9. ....24
Figure 1-11	La signature composite obtenue après le mappage physique des bascules.....24
Figure 3-1	Diagramme de flot pour la conception d'un circuit numérique ASIC. ....34
Figure 3-2	Diagramme de flot modifié pour la conception d'un circuit numérique ASIC. ....35
Figure 3-3	Traitement pour obtenir les nœuds de Mentor et le nœud de Cadence duquel il appartient.....37
Figure 4-1	Les bascules et les nœuds du circuit fictif .....42
Figure 4-2	Traitement pour l'obtention de la signature à pondération constante (Méthode 1).....46

Figure 4-3	Signature à pondération constante du circuit b12 pour 15 bascules d'intérêt.....	47
Figure 5-1	Traitement pour l'obtention de la signature pondérée par les capacités parasites des nœuds (Méthode 2).....	50
Figure 5-2	Les signatures à pondération variable (en fonction de la capacité parasite) et à pondération constante.....	52
Figure 6-1	Traitement pour l'obtention de la signature pondérée par les couches métalliques des nœuds (Méthode 3).....	55
Figure 6-2	Les signatures à pondération variable (en fonction du nombre des couches métalliques) et à pondération constante.....	56
Figure 7-1	Approximation de la surface du polygone DRC (en jaune) par un rectangle.....	62
Figure 7-2	Traitement pour l'obtention de la signature pondérée par les polygones DRC, première variante (Méthode 4a).....	65
Figure 7-3	Assignation des surfaces efficaces aux nœuds à partir de dessin des masques du circuit.....	66
Figure 7-4	Signatures à pondération variable produites par "analyse des pannes" pour un DRC plus conservateur, première variante (Méthode 4a).....	67
Figure 7-5	Usage de la méthode 4a pour l'obtention des signatures et des équations.....	68
Figure 7-6	Traitement pour l'obtention de la signature pondérée par les polygones DRC (Méthode 4b).....	72
Figure 7-7	Signatures à pondération variable produites par "ajout de listes" pour un DRC plus conservateur, deuxième variante (Méthode 4b).....	73
Figure 7-8	Signatures à pondération variable produites "à l'apparition de la première panne" pour un DRC plus conservateur, troisième variante (Méthode 4c).....	74
Figure 7-9	Traitement pour l'obtention de la signature pondérée par les polygones DRC (Méthode 4c).....	75
Figure 7-10	Graphe récapitulatif des signatures.....	78

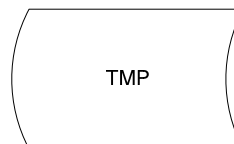
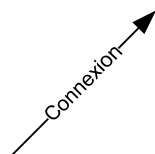
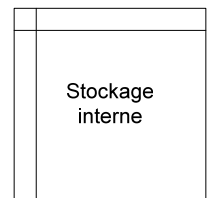
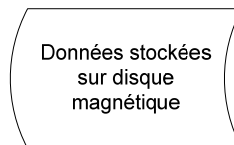
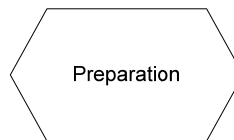
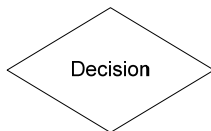
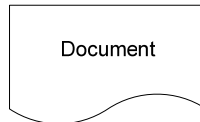
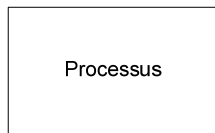
## LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

ASIC	Application Specific Integrated Circuit	Circuit intégré pour application spécifique
ATPG	Automatic Test Pattern Generation	Génération automatique de patrons de test
Awk	Aho, Weinberger, Kernighan text processing programming language	Aho, Weinberger, Kernighan Langage de traitement des lignes
CAA	Critical Analysis Area	Analyse d'aire critique
CFA	Critical Feature Analysis	Analyse de caractéristique critique
CMOSP18	TSMC's 0.18-micron CMOS Technology	Technologie CMOS de 0,18µm propriété du TSMC
COT	Client owned Technology	Technologie propriété du client
DEF	Design Exchange Format	Format d'échange de design
DFM	Design for Manufacturability	Conception facilitant la fabrication
DFT	Design for Testability	Conception facilitant la testabilité
DFY	Design for Yield	Conception favorisant le rendement
DRC	Design Rule Checking	Vérification de règles de conception
DSM	Deep-Submicron technology	Technologie sous micrométrie
ECO	Engineering Change Order	Commande de modification d'ingénierie
FF	Flip-flop	Basculé bistable
FSM	Finite State Machine	Machine à états finis
GDS	Graphic Data System	Système des données graphiques
HDL	Hardware description language	Langage de description matérielle
IC	Integrated Circuit	Circuit intégré
ISCAS	IEEE International Symposium on Circuits and Systems	Symposium International IEEE sur les circuits et systèmes
ITC'99	International Test Conference Atlantic City 1999	Conférence internationale sur le Test Atlantic City 1999
LCC	Lithography compliance check	Vérification de conformité lithographique
LEF	Library Exchange Format	Format d'échange de bibliothèque

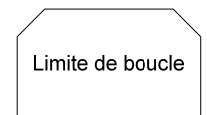
LINUX	Computing operating system	Système d'exploitation pour ordinateur PC
LVS	Layout versus Schematic	Vérification de la conformité du dessin des masques par rapport au schéma
MOSIS	Metal Oxide Semiconductor Implementation Service	
NMNC	Nœud Mentor nœud Cadence	Nombre des nœuds de Mentor qui sont connectés à un nœud de Cadence
OASIS	Open Artwork Systems Interchange Standard	Standard ouvert pour l'échange binaire des dessins des masques
RDRs	Restrictive Design Rules	Règles de conception restrictives
RTL	Register Transfer Level	Niveau transfert de registre
SOC	System-on-Chip	Système sur puce
SOFE	Stop on first error	Arrêt à la première erreur
SPICE	Simulation program with Integrated Emphasis	Programme de simulation de circuits
TFM	Test for Manufacturing	Test de la fabrication
TSMC	Taiwan Semiconductor Manufacturing Company	
VHDL	VHSIC (Very High Speed Integrated Circuits) hardware description language	Langage de description matériel (pour circuits intégrés à très grande vitesse)
VLSI	Very Large Scale Integration	Intégration à très grande échelle



## LISTE DE SYMBOLES ET UNITÉS DE MESURE (UTILISÉES DANS LES FIGURES)



Fichier temporaire



## INTRODUCTION

Maintenant que des milliards de portes logiques peuvent être entassées sur une seule puce VLSI, des milliers voire des millions de vecteurs de test doivent être appliqués aux entrées de la puce afin, dans un premier temps, de détecter les puces défectueuses, et dans un deuxième temps, de diagnostiquer les causes des pertes de rendement. Ces étapes sont particulièrement importantes dans les premiers mois d'utilisation d'un nouveau procédé de fabrication, pour atteindre le plus rapidement possible des niveaux de rendement acceptables. C'est principalement pour cette raison que les manufacturiers de puces suivent de près l'évolution des mécanismes de défauts menant aux pertes de rendement. Auparavant, ces mécanismes étaient plutôt dominés par les défauts dits aléatoires produites par les particules présentes durant la fabrication des circuits intégrés. Cependant, avec le développement de la technologie des semi-conducteurs à l'échelle nanométrique (en dessous de la barre de 130 nm), les défauts complexes dues aux propriétés de ces technologies ou encore spécifiques au design deviennent prédominantes par rapport aux défauts aléatoires (Tirée de Adhir Upadhyay, 2008, p.1), ce qui provoque des pannes qui sont systématiques. Ces pannes, extrêmement difficiles à localiser dans la complexité colossale des circuits modernes, doivent être identifiées afin de minimiser l'impact sur les coûts totaux et sur l'échéancier de la disponibilité des puces sur le marché.

Le présent mémoire explore et compare diverses approches statistiques de diagnostic visant à faciliter l'identification des pannes systématiques. Ces approches sont basées sur le concept de signature de défaut caractéristique, et font appel à des informations diverses pour construire ces signatures, en cumulant les erreurs qui apparaissent à la sortie des bascules pendant le test de plusieurs puces aux données extraites des règles de conception effectivement appliquée au dessin des masques.

La contribution de ce mémoire est de 1) proposer une méthodologie automatisée de construction de signatures différentes avec les bascules du circuit et les nœuds couverts par

ces bascules dans la zone d'intérêt de la puce sous test, et 2) comparer ces diverses signatures pour déterminer l'impact de l'information utilisée.

L'organisation de ce mémoire est la suivante. Le premier chapitre présente la revue de la littérature. Le deuxième définit les choix du projet en termes de technologie et circuits cibles. Le troisième s'intéresse au flot de conception et aux étapes préliminaires pour préparer les données nécessaires à l'élaboration des signatures. Le quatrième chapitre traite de l'implémentation de la première méthode de signature proposée, la signature de référence où la contribution des nœuds est équiprobable. Les chapitres 5 et 6 présentent l'implémentation des signatures dans lesquelles la contribution des nœuds varie respectivement en fonction de leurs capacités parasites et des couches métalliques impliquées. Le chapitre 7, qui introduit les signatures dont la pondération est fonction de la surface des polygones produits par l'outil de Cadence (utilisé pour la conception de dessin des masques) suite à un DRC ("Design Rule Checking") pour représenter le non-respect des règles de dessin. Ce chapitre offre des variantes des signatures de même type de pondération à partir des commandes différentes en Fastscan (un outil de génération automatisée de vecteurs de Mentor Graphics). Enfin, les conclusions, les améliorations à apporter, les recommandations pour les travaux futurs sont présentées à la fin du mémoire.

## **CHAPITRE 1**

### **REVUE DE LITTÉRATURE**

Le processus d'identification des causes qui provoquent la baisse du rendement des circuits implique généralement trois étapes (Tirée de Geir Eide, 2010, p. 1):

- 1) La séparation des circuits qui contiennent des pannes systématiques de ceux présentant des pannes aléatoires.
- 2) L'identification de modes semblables de défectuosité à travers un grand nombre de dispositifs défectueux.
- 3) La localisation et l'identification des pannes par une analyse de défaillance physique (« failure analysis »).

L'objectif premier de ce processus est de limiter le nombre de cas de défaillances physiques à analyser, cette troisième et dernière étape étant très longue et coûteuse. La figure 1-1 montre des exemples de deux types de pannes, systématiques et aléatoires, qui ont fait l'objet d'une analyse de défaillance physique. La première étape du processus consiste à faire ressortir certaines corrélations ou symptômes globaux communs entre les circuits défectueux. Dans la totalité des travaux répertoriés jusqu'à présent, cet exercice débute par l'énoncé d'une ou plusieurs hypothèses quant à la caractéristique probable menant à cette corrélation, comme par exemple l'emplacement physique des puces défectueuses dans le réticule ou la gaufre. La seconde étape du processus apparaît comme un raffinement où l'on essaie d'identifier de manière plus précise la (ou les) sources des défectuosités aux caractéristiques communes.

Les deux premières étapes du processus font appel aux données brutes recueillies lors du test en production des puces. Une première différenciation à faire entre les méthodes existantes menant à l'identification des causes principales des pertes de rendement est l'utilisation ou non d'un outil de diagnostic alimenté par les valeurs logiques (erronées ou non) (Tirée de Geir Eide, 2010, p. 1). Une seconde différenciation entre ces méthodes est l'utilisation d'informations supplémentaires, tel le dessin des masques ou les règles de dessin (Tirée de C.

Schuermeyer et autres, 2005, p.3). Un concept commun à plusieurs de ces méthodes est celui de la signature. Une signature peut être vue comme une manière de regrouper et visualiser toute l'information disponible provenant de sources diverses dans un format compacté. L'extraction de données allouées aux signatures peut être fait de celles impliquées dans la conception (règles de design du schéma, du dessin des masques) dans la fabrication (vecteurs de test, comportement aux vérifications) ou économiques (prix du test et prix du circuit, productivité). Elle peut prendre différentes formes. Elle est utilisée de deux manières différentes : 1) comme outil de visualisation guidant les efforts de diagnostic, auquel cas il n'y a pas de point de comparaison, et 2) servir comme un point de comparaison avec une réponse anticipée (pré calculée).

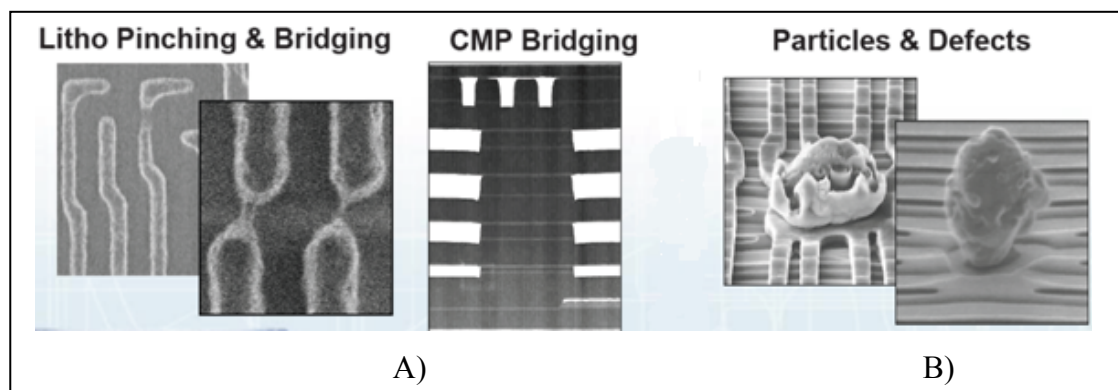


Figure 1-1 Exemples des pannes systématiques (a) et aléatoires (b).  
Tirée de Jeff Wilson, (2009, p. 3)

La figure 1.2 illustre la procédure de création de signatures. Les entrées du moteur d'extraction sont les résultats non traités des tests en production et des informations supplémentaires provenant du design.

On parlera d'une signature composite (Tirée de C. Schuermeyer et autres, 2005, p.3) lorsque celle-ci est générée par la compilation des résultats obtenus pour une collection des puces (dés) testées. Le rôle d'une signature composite est de montrer le comportement moyen de tous les dés testés, au lieu de celui spécifique d'un seul dé. La compréhension du comportement de toute la population rend possible l'identification des groupes des dés déviants, qui ne réagissent pas comme prévu. Comme indiqué précédemment, les signatures

peuvent être générées directement à partir des données brutes du test, ou encore à partir de données filtrées par un outil de diagnostic.

Dans ce qui suit, nous décrivons les principaux types de signatures présents dans la littérature, à savoir :

- 1) La signature basée sur le type de cellules présumées défectueuses.
- 2) La signature utilisant la disposition logique des bascules à sorties erronées.
- 3) La signature utilisant les coordonnées physiques des bascules à sorties erronées.
- 4) Les signatures comparatives génériques générées à partir de la séparation des données de test.

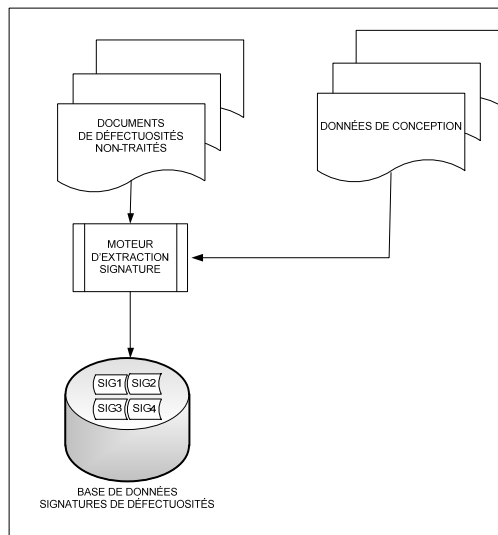


Figure 1-2 Exemple de procédure de création de signatures.  
(Tirée de C. Schuermeyer et autres, 2005, p.3)

### 1.1 Signature basée sur le type de cellules présumées défectueuses

Il s'agit d'une signature composite créée à partir des résultats fournis par l'outil de diagnostic, qui prend la forme d'un histogramme où sur l'axe Oy on représente les types de cellules standards et sur l'axe Ox se retrouve le nombre des dés rapportés échoués pour chaque type de cellule identifié comme site potentiel d'une défectuosité (voir la figure 1.3).

Dans l'exemple de la figure 1.3, la cellule H1M1D est celle qui est désignée le plus souvent comme candidate défectueuse. Cependant cette cellule est celle qui est utilisée le plus souvent dans le design, ce qui explique en partie sa position dans l'histogramme. Cet exemple illustre le fait que ces résultats peuvent nécessiter une normalisation afin d'être interprétés correctement. Une telle normalisation est proposée par R. Turakhia et autres, 2009, p. 167, basée sur l'analyse de la surface dite critique des cellules. Cette analyse consiste à identifier les parties sensibles du dessin des masques en fonction de la distribution présumée de la taille des défauts (aléatoires). Un exemple de résultat de cette analyse est présenté à la figure 1.4.

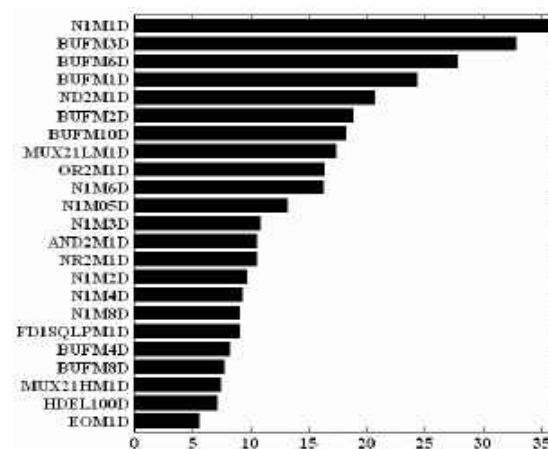


Figure 1-3 Les types de cellules en fonction du nombre des apparitions en défectuosité.  
Tirée de C. Schuermyer et autres (2005, p.3)

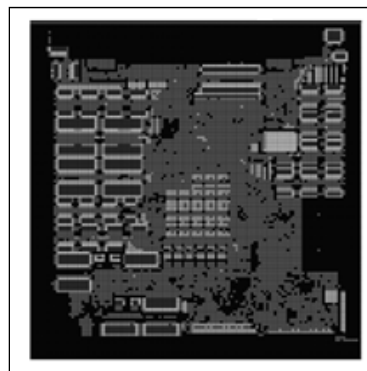


Figure 1-4 Identification des zones sensibles du dessin des masques suite à une analyse de la surface critique au niveau de la puce (les points chauds sont noircis).  
Tirée de R. Turakhia et autres (2009, p. 167-172)

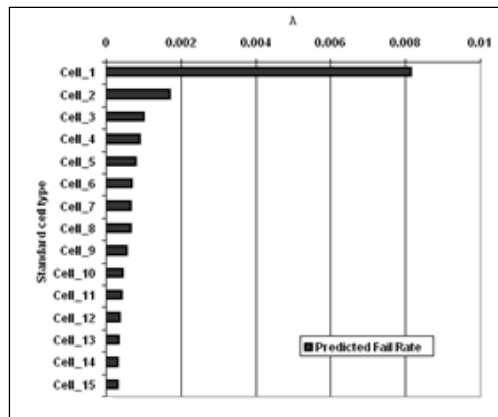


Figure 1-5 Signature anticipée pour les premières 15 cellules prédisposées aux pannes créée à partir de l'analyse de la figure antérieure.

Tirée de R. Turakhia et autres (2009, Pages 167-172)

La normalisation permet de créer une signature anticipée, montrée à la figure 1.5. La création de cette signature anticipée exige la disponibilité de la distribution de la taille des défauts (rarement disponibles). De plus, elle s'appuie sur le caractère aléatoire (non systématique) des défauts.

## 1.2 Signature utilisant la disposition logique des bascules à sorties erronées

Il s'agit d'une signature composite ne nécessitant pas l'utilisation d'un outil de diagnostic. Cette signature, qui compile simplement le nombre de fois que le contenu d'une bascule est erroné, est adéquate pour un volume des composants réduites, situation rencontrée quand on obtient les premières composantes après la conception.

La signature peut être générée selon différents niveaux de granularité variable, selon le nombre de dés testés. Lorsque le nombre de dés est suffisamment élevé, le niveau de granularité le plus fin peut être utilisé, ce qui correspond au niveau de chaque bascule compilée individuellement. Ce niveau est illustré à la figure 1.6. Si la quantité des données est limitée (le cas d'une production réduite), les bascules défectueuses peuvent être regroupées pour la signature. La figure 1.7 montre un exemple où les bascules sont regroupées selon la chaîne à laquelle elles appartiennent. Il est à noter que même s'il y a des



multiples défauts sur la même chaîne, la signature contient juste la décision passé/échoué (à droite sur la figure).

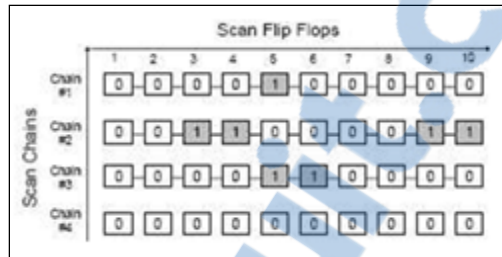


Figure 1-6 Un mappage logique par des bascules.  
Tirée de C. Schuermyer et autres, (2005, p.4)

Un exemple de signature composite pour le regroupement par chaîne, sous la forme du diagramme de Pareto, est présenté à la figure 1.8. Dans cet exemple, la chaîne la plus souvent défectueuse est la chaîne 16. Comme pour le cas des signatures basées sur le type de cellules présumées défectueuses, une normalisation s'avère nécessaire pour bien interpréter ces résultats.

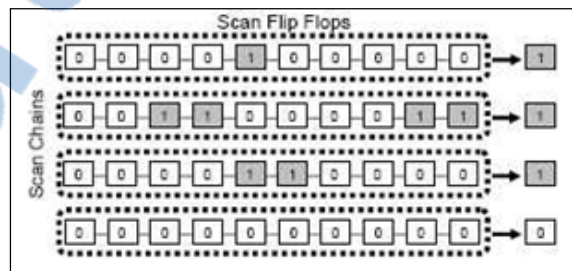


Figure 1-7 Un mappage logique par des chaînes des bascules.  
Tirée de C. Schuermyer et autres, (2005, p.4)

### 1.3 Signature utilisant les coordonnées physiques des bascules à sorties erronées

Il s'agit d'une signature composite ne nécessitant pas l'utilisation d'un outil de diagnostic. Cette signature utilise les coordonnées x-y des bascules à sorties erronées dans le design.

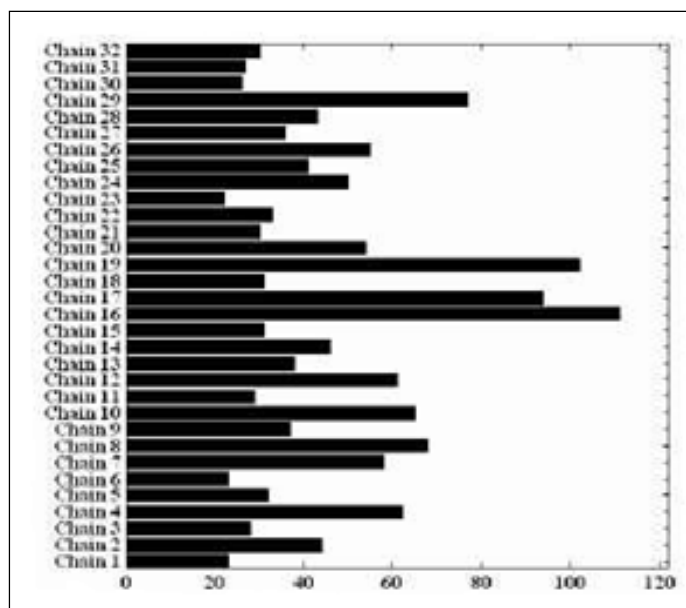


Figure 1-8 Un exemple de signature composite  
Tirée de C. Schuermyer et autres, (2000, p.5)

Les bascules sont habituellement regroupées en utilisant un carrelage de tuiles de mêmes dimensions (voir la figure 1.9). Les carrées en gris sont des emplacements défectueux potentiels et les carrées noircies montrent les bascules avec des sorties erronées. Comme le même nombre de bascules peut varier d'une tuile à l'autre, ce type de signature peut requérir une normalisation. La figure 1.10 donne la signature individuelle (un seul dé) correspondante à la figure 1.11.

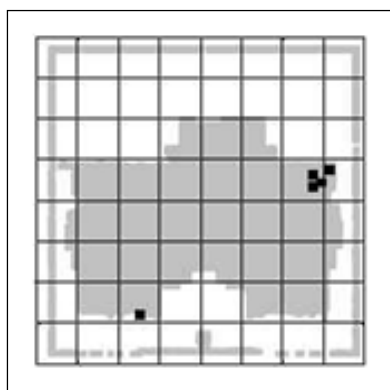


Figure 1-9 Utilisation des coordonnées x-y dans la signature.  
Tirée de C. Schuermyer et autres, (2005, p.6)

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	1
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0
0	0	0	0	0	0	0	0

Figure 1-10 Signature correspondant à la Figure 1-9.  
Tirée de C. Schuermyer et autres, (2005, p.6)

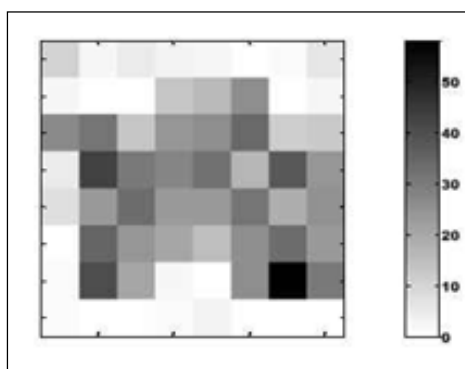


Figure 1-11 La signature composite obtenue après le mappage physique des bascules.  
Tirée de C. Schuermyer et autres, (2005, p6)

À la figure 1-11, on voit un exemple de signature composite obtenue. Ce type de signature est surtout utile pour faire ressortir les corrélations de coordonnées au sein d'un lot de puces défectueuses.

#### 1.4 Signatures comparatives génériques générées à partir de la séparation des données de test

C. Schuermyer et autres, (2005) ont développé une technique intéressante pour la création de réponses anticipées. Ils séparent l'ensemble des données en 2 groupes, l'un pour générer la réponse anticipée et l'autre pour créer la signature en tant que telle. L'hypothèse de base sous-jacente à cette technique est qu'une séparation aléatoire des 2 groupes va se traduire par

une forte corrélation entre les 2 groupes, alors qu'une sélection faite sur un critère particulier va faire ressortir les puces marginales qui pourraient être affectées par une panne de type systématique, si cette panne systématique est liée à ce critère particulier. Le défi avec ce genre de technique est que son efficacité dépend principalement de l'intuition de celui qui l'utilise, quant aux critères de séparation à privilégier.

## **1.5        Autres travaux**

Certains travaux exécutés au cours des dernières années n'ont pas eu comme but la mise au point d'une signature mais ont analysé profondément le dessin des masques dont ceux (J. Mekkoth et autres, 2006), pour mettre en évidence les causes probables des défauts. Les pannes détectées lors du diagnostic initial ne considèrent pas l'information du dessin des masques. L'analyse se fait à partir des règles de design, sans vraiment mentionner comment et introduit un nouvel concept d'ATPG adaptatif et la création de vecteurs de test additionnels qui pointent vers des pannes spécifiques. M. Keim et autres, 2006, en se servant d'outils de DRC commerciaux, inspectent le dessin de masques et mettent en évidence les violations pour certaines règles de design. Ils utilisent les règles écrites d'une telle manière qu'elles identifient les caractéristiques d'intérêt et déterminent les paramètres physiques reliés à ces caractéristiques. Ils ont aussi souligné l'importance de vecteurs de test additionnels mais obtenus seulement après que le dessin de masques soit finalisé. La conclusion la plus importante est pour eux que le diagnostic ne pointe exactement sur l'emplacement de la panne ou de la source de défaut mais plutôt produit une liste d'endroits possibles ou suspects. Comme l'équipe de D. Appello l'a remarqué (D. Appello et autres, 2004, p210), il faut trouver une optimisation entre le temps de test et la précision et l'exactitude de l'outil de diagnostic. Cela implique une compréhension de limitations du mode test, un poids ou une hiérarchie des candidats en considérant les défauts les plus nombreuses.

## **1.6 Signatures proposées dans ce mémoire**

Notre projet cible principalement l'étape de l'identification des causes des pannes systématiques. Nous proposons une série de signatures composites pouvant servir de point de comparaison à la signature utilisant la disposition logique des bascules à sorties erronées (décrite à la section 1.2), sans avoir recours à un outil de diagnostic. Les signatures développées dans le cadre de ce mémoire constituent une estimation de la probabilité d'apparition de valeurs erronées dans chacune des bascules d'intérêt d'un circuit sous test. Les signatures proposées ciblent les couches de métallisation des puces, principalement en raison de notre impossibilité d'accéder au dessin des masques des cellules normalisée. Il s'agit ici d'un travail d'exploration visant à analyser l'apport des diverses informations ainsi que la manière d'utiliser les outils informatiques sur la forme des signatures, grâce entre autres à un système d'équations menant à l'estimation de coefficients de pondération pour chacune des causes potentielles. Les informations utilisées vont des capacités parasites aux règles du dessin des masques. Même si certains auteurs mentionnent au passage l'utilisation de ces règles de dessin, aucune signature basée n'a été répertoriée jusqu'ici. Une autre caractéristique qui différencie les signatures proposées dans ce mémoire est le fait que l'apport des signaux dits globaux est exclu des signatures, ce qui rend à notre avis les signatures plus efficaces. Les signatures proposées peuvent être vues comme étant complémentaires à celles existantes dans la littérature.

## **1.7 Conclusions**

Sans avoir épuisé la grande diversité des signatures existantes dans la littérature, nous avons présenté les signatures les plus populaires ainsi que celles se rapprochant le plus de ce que nous proposons. Comme souligné précédemment, notre approche est différente de celles proposées dans la littérature. Au chapitre suivant, nous discutons du choix du circuit et de la technologie de fabrication pour exemplifier les signatures proposées.

## **CHAPITRE 2**

### **CHOIX DU PROJET– FAMILLE TECHNOLOGIQUE CMOSP18, CIRCUITS DE RÉFÉRENCE b02 & b12**

#### **2.1 Introduction**

Dans le cadre de ce projet, nous devons choisir une technologie et des circuits cibles, ainsi qu'une méthodologie de conception afin de mettre en œuvre l'approche développée. Nous les décrivons dans ce chapitre.

#### **2.2 Technologie cible**

Parmi les technologies accessibles via la société CMC Microsystèmes, celle qui répondait le mieux à nos besoins était la technologie CMOSP18 (CMOS 0.18  $\mu\text{m}$ ) de la compagnie TSMC, qui incidemment est la plus grande fonderie au monde en ce moment. Le choix est basé principalement par l'accessibilité aux bibliothèques spécifiques et à la documentation de ce procédé de fabrication. En même temps cette technologie est suffisamment complexe pour appuyer nos modèles.

Selon la documentation du manufacturier, le procédé CMOS TSMC 0.18 $\mu\text{m}$  peut servir à la mise en œuvre de différents types de circuits (ex. mémoires embarquées, circuits numériques et mixtes / RF, circuits à tensions élevées, capteurs d'image CMOS) ciblant de nombreuses applications (ex. graphique 3D de haute performance, télévision numérique, logique programmable à haute capacité). Cette technologie peut atteindre des densités de 160000 portes par millimètre carré, ainsi que des fréquences d'horloge de 500MHz pour la logique et la mémoire, avec une dissipation de puissance de grille inférieure à 3 nW/MHz. La technologie supporte 6 couches de métaux et une couche de polysilicium, avec une tension d'alimentation nominale de 1,8 V.

### 2.3 Méthodologie cible

La méthodologie de conception choisie est celle des cellules normalisées. Le design basé sur les cellules normalisées est une approche de conception largement adoptée pour les circuits intégrés à application spécifique (ASIC) ou les systèmes sur puce (System-on-Chip, SOC). (Tirée de Weste N et autres, 2005) Cette approche utilise une bibliothèque de cellules pré-désignées, qui est une collection de modules de base. L'utilisation d'une bibliothèque de cellules normalisées permet l'automatisation de la création de dessin des masques, via les étapes de placement et routage. Cette automatisation réduit le temps de conception et induit moins d'erreurs que le dessin des masques manuel.

Bien que la technologie TSMC 0,18 $\mu\text{m}$  comprenne 6 couches de métal, seulement les couches Métal 1 et Métal 2 ont été utilisées pour les interconnexions à l'intérieur même des cellules. Cette contrainte permet aux outils de placement et routage (Place et Route, P & R) d'utiliser pleinement les autres couches métalliques. En plus des règles de conception, la topologie des cellules suit les règles indiquées dans le tableau 1, qui sont nécessaires pour les outils de placement et routage.

Tableau 2.1 Règles pour le dessin des masques des cellules  
de la bibliothèque de 0.18  $\mu\text{m}$ ;  $\lambda = 0,09\mu\text{m}$

Paramètre de design de la cellule	Valeurs
Hauteur de la cellule	108 $\lambda$
Largeur de la cellule	Multiple de 9 $\lambda$
Distance séparant deux lignes d'interconnexion	9 $\lambda$ pour métal 1 jusqu'au métal 5; 18 $\lambda$ pour métal 6
Largeur de métal	4 $\lambda$ pour toutes les couches
Offset de métal	0 pour toutes les couches
Pins d'alimentation/de la terre/rail	11 $\lambda$ pour VDD et VSS

Notes :

1. Lambda est un facteur d'échelle appliqué à la technologie souhaitée pour ses éléments (transistors, métal, poly, etc.) en répondant aux règles de conception spécifiques. Il permet que les règles puissent être conservées entièrement aux changements des dimensions d'une famille à l'autre.
2. Les valeurs utilisées réellement par Cadence ont été modifiées par CMC.

## 2.4 Circuits de référence b02 et b12

Les circuits b02 et b12, utilisés comme exemples dans le cadre de ce projet, sont des circuits de référence pour le test et se trouvent dans la famille (« benchmark ») des circuits ITC'99 (Scott Davidson, 1999). Les circuits des tests de performance ITC'99, développés par le Groupe CAD à l'École Polytechnique de Torino (199T), constituent un ensemble de circuits conçus avec des caractéristiques typiques pour la synthèse des circuits. Pour chaque banc d'essai sont disponibles la description VHDL RTL et la liste des portes et interconnexions provenant de la synthèse. Les principales caractéristiques des circuits sont:

- Complètement synthétisable au niveau VHDL de description RTL (style de Synopsys Design Compiler);
- aucune directive spécifique de compilateur;
- requiert seulement la logique standard IEEE et les paquets arithmétiques;
- entièrement synchrones;
- une seule phase de signal d'horloge connectée directement à la mémoire des éléments;
- signal de reset global toujours disponible;
- pas de mémoire interne (à l'exception des banques de registres);
- pas de bus trois-états ou de porte câblée.

La gamme des descriptions VHDL au niveau RTL commence à partir de petits circuits monolithiques (1 entité, 1 processus, 70 lignes de code) jusqu'aux circuits complexes, multi-entités et multiprocessus (exemple : 11 entités, 33 processus, 1424 lignes de code). Au niveau des portes logiques, la gamme des circuits commence à partir d'un circuit de petite taille s27 (2 entrées, 29 portes, 4 flip-flops, 150 pannes) jusqu'à un circuit complexe (37 entrées, 69917 portes, 3320 bascules, 429712 pannes). Étant conçus pour une évaluation



d'ATPG séquentielle pure, les circuits n'incluent aucune structure DFT (comme scan complet ou partiel) mais l'insertion de la chaîne de registres à balayage (scan chain) est facile à cause de la structure simple d'horloge. Au début, pour la mise au point de la procédure automatisée, spécialement pour la quatrième méthode, le circuit b02 (voir le tableau 2.2), plus simple, a été largement utilisé. Par la suite, la méthodologie, les équations et les signatures montrées dans ce mémoire ont été développés pour le circuit b12 qui est plus complexe.

Tableau 2.2 Caractéristiques des circuits utilisés b02 et b12

NOM	VHDL		NIVEAU PORTE						LISTE DE PANNES		FONCTION NALITÉ
	LIGNE	PROC	PORTE	ÉP	SP	FF	L0	L1	COMP	COLL	
b02	70	1	28	1	1	4	0	0	148	62	FSM qui reconnaît BCD nombres
b12	569	4	1076	5	6	121	0	0	6306	2856	Jeu pour 1 joueur (devine une séquence)

LINES: lignes VHDL

PROC: Nombre de processus

GATES: Nombre de portes

PI: Nombre d'entrées primaires

PO: Nombre de sorties primaires FF: Nombre de bascules

L0/1: Nombre de logique-zéro/logique-one

COMP: Nombre des pannes dans la liste complète

COLL: Nombre des pannes dans la liste des pannes équivalentes (collapsed faults)

La procédure peut être appliquée pour n'importe quel circuit numérique qui se prête à une vérification ATPG. L'exemplification est faite sur la variante de synthèse standard mais il existe aussi la variante optimisée du circuit. Le code VHDL du circuit b12 est présentée dans l'annexe I.

## **2.5 Conclusions**

Dans ce chapitre, nous avons présenté la technologie cible ainsi que les circuits de référence.  
Au chapitre suivant, nous nous intéressons au flot de conception.

## CHAPITRE 3

### ÉTAPES PRÉPARATOIRES - CONCEPTION D'UN CIRCUIT ASIC, TRAITEMENTS EFFECTUÉS PAR FASTSCAN ET AWK

#### 3.1 Introduction

Dans ce chapitre, nous présentons les étapes de conception d'un circuit ASIC, flots classique et modifié pour nos besoins, ainsi que les traitements pour obtenir une séquence correspondante de vérification ATPG.

#### 3.2 Diagramme de flot classique pour la conception d'un circuit ASIC

Le diagramme de la figure 3.1 présente le flot normal et général de conception d'un circuit numérique ASIC. Étant donné qu'il se trouve bien expliqué dans de nombreux ouvrages (dont le tutoriel du Collège Militaire Royal du Canada, 2006), nous n'insistons pas sur les détails. Le but de ce processus de conception est d'obtenir un dessin des masques sans erreur de règles de design. La première étape est la synthèse du circuit décrit en VHDL ou Verilog pour la famille technologique ciblée, suivie de l'introduction de (ou le remplacement) des bascules normales par des bascules modifiées afin de permettre le test en balayage (ou bascules de type scan). Ensuite la conception du dessin des masques est effectuée en deux temps : le placement et le routage. Pour éviter l'apparition d'erreurs, le schéma au niveau porte logique et le dessin des masques obtenus sont comparés à l'aide de l'utilitaire LVS (Layout versus Schematic) et finalement le dessin des masques est vérifié par l'outil DRC afin de détecter les possibles violations des règles de conception. Le circuit est considéré valide pour la fabrication seulement lorsqu'aucune erreur n'est signalée. Notons que deux étapes n'apparaissent pas à la Figure 3.1 pour éviter de surcharger la figure. Il s'agit de l'ajout d'un logo d'identification et de l'ajout de métal et poly afin d'atteindre les normes de densité minimale du fabricant. Ces étapes sont effectuées juste avant la vérification finale de DRC. Nous considérons que l'utilisation que nous ferons de l'outil DRC dans la création de certaines signatures sera avant ces 2 étapes, l'ajout de métal rendant l'analyse très difficile.

Nous présentons à l'annexe II quelques résultats issus de la création du dessin de masque pour le circuit b12.

### **3.3 Diagramme de flot modifié de conception d'un circuit ASIC**

Pour effectuer le calcul des différentes signatures, le flot a été modifié conformément à la figure 3.2, à savoir :

- 1) Pour éviter les non conformités créées suite à l'ajout des cellules d'entrée/sortie au circuit initial, la génération de la liste des pannes et des vecteurs de test a été faite sur la liste des connections obtenue après cet ajout et non sur le fichier original de ITC99. L'insertion des bascules de type scan par DFTAdvisor demeure par contre au même endroit dans le flot.
- 2) Dans le but d'introduire de manière volontaire des erreurs de DRC, le fichier original de Cadence avec les règles de DRC a été remplacé par un autre fichier avec des règles plus conservatrices, imposés par les requis du projet; en changeant les règles DRC originales par des règles plus conservatrices, nous obtenons dans le fichier cible un nombre de dépassements qui se traduisent par des polygones DRC. Le fichier cible contient la règle, la valeur limite dépassée, ainsi que le numéro de dépassement obtenu et les coordonnées x; y des coins de chaque polygone. Ces informations seront utilisées plus tard afin de créer des signatures basées sur ces polygones. Un exemple de ce type de fichier est donné à l'annexe XII, pour le dépassement de la quatrième règle.
- 3) Nous avons ajouté un fichier de commandes qui enregistre les résultats dans un fichier cible.

La première modification nous oblige à ajouter une étape de traitement visant à établir la correspondance entre les nœuds tels qu'ils apparaissent dans l'environnement de Mentor (appelés nœuds Mentor) et ceux qui apparaissent dans l'environnement de Cadence (appelés nœuds Cadence). En plus d'un changement de nom, il s'avère que plusieurs nœuds Mentor peuvent représenter un même nœud Cadence.

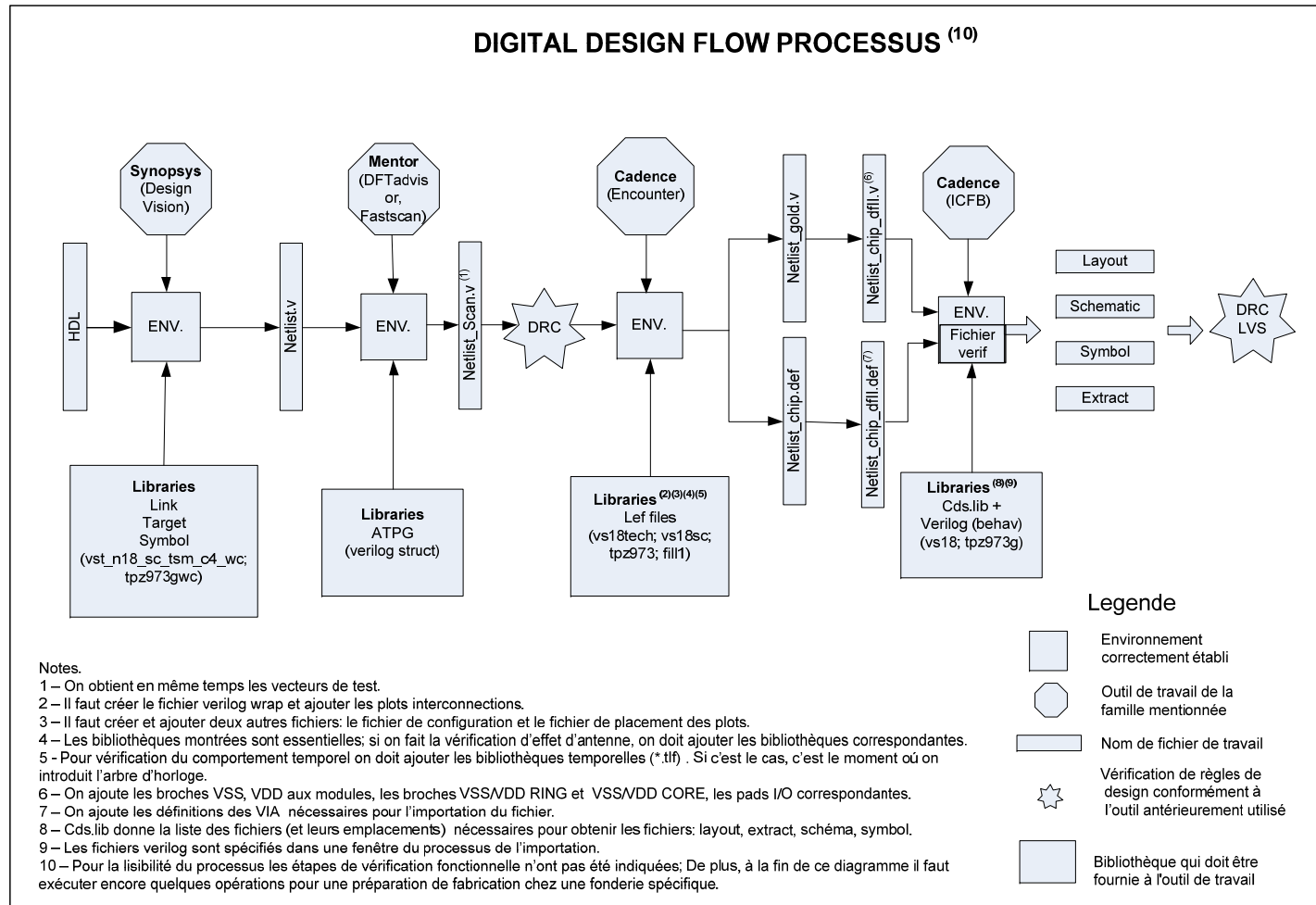


Figure 3-1 Diagramme de flot pour la conception d'un circuit numérique ASIC.

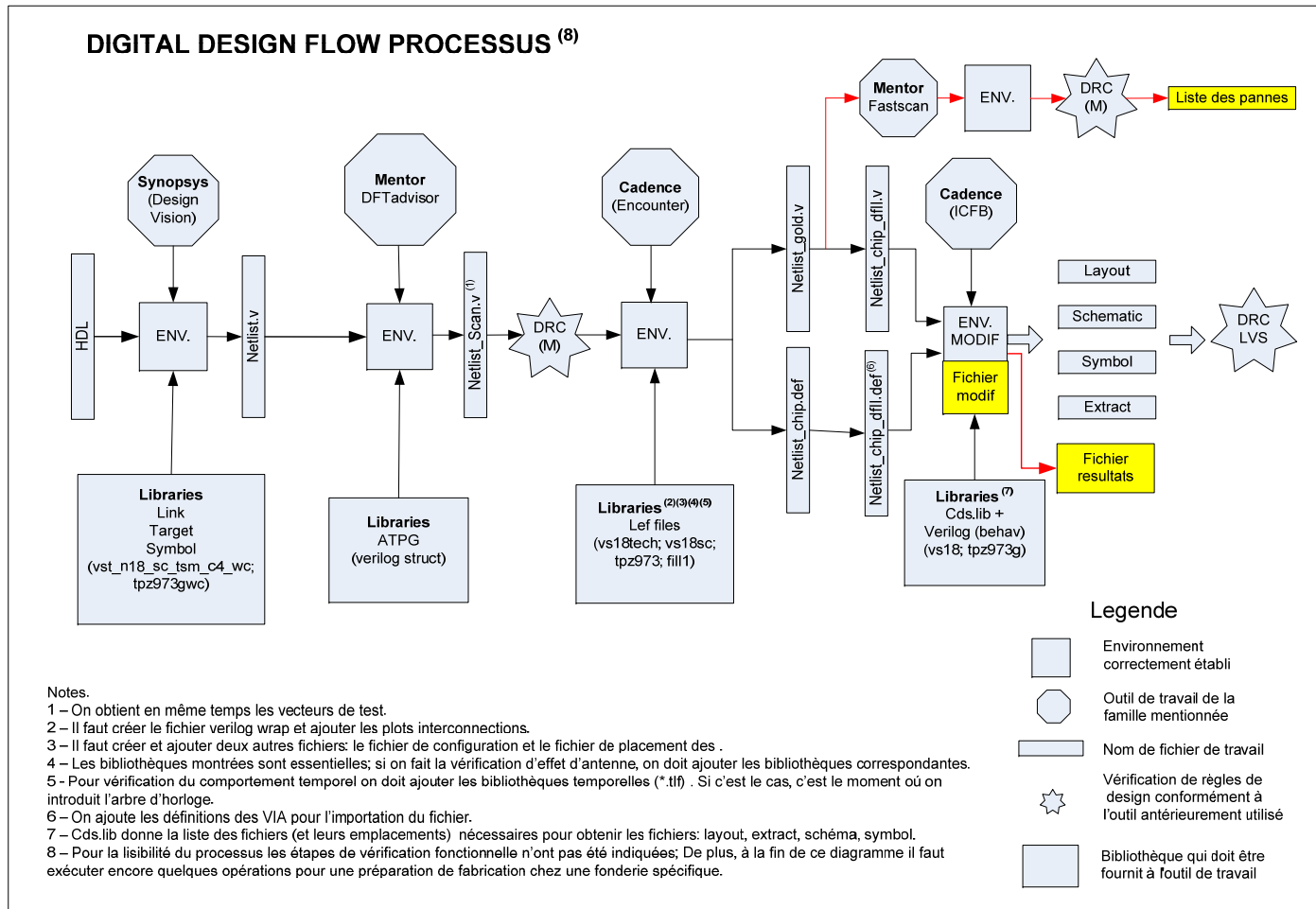


Figure 3-2 Diagramme de flot modifié pour la conception d'un circuit numérique ASIC.

De ce fait, nous avons besoin de calculer nombre de nœuds Mentor par nœud Cadence, aussi appelé coefficient NMNC. La figure 3.3 montre le processus à suivre pour obtenir ce coefficient. Le fichier de type DEF du circuit, obtenu conformément à la figure 3.2 est le point de départ pour ce traitement. Puisque le fichier DEF est très important pour notre étude, un fragment du fichier obtenu pour le circuit b12 (b12\_chip\_DEF) est présenté à l'annexe IV. Le traitement consiste d'abord à associer les nœuds Mentor (présents sous la forme des points d'interconnexions des cellules) avec les nœuds électriques (Cadence) correspondants. Il est à noter aussi que le NMNC peut se vérifier avec le "fan-out" résultant de l'extraction des capacités parasites du dessin des masques exécuté par Cadence. Ensuite nous séparons les nœuds en deux listes distinctes: celle contenant les nœuds qui sont sur les connections métalliques et celle contenant les nœuds qui ne se trouvent pas sur les connections métalliques. Dans le cadre de ce travail, nous considérons seulement les nœuds métalliques, les autres étant éliminés. Finalement, nous éliminons de la liste les signaux globaux. Nous définissons comme global tout signal qui, lorsque défectueux, affecte une bonne proportion des bascules. Nous avons fixé de manière préliminaire cette proportion à 40%. Les signaux d'horloge, de « reset », de « scan enable » sont des exemples typiques de signaux globaux. La raison motivant l'enlèvement de ces signaux globaux est issue de l'observation, dans nos premières signatures, que leur contribution aux signatures peut être vue comme un important bruit de fond qui se superpose sur la contribution des autres signaux. À titre d'exemple, le circuit b12 a quatre signaux globaux. La première modification nous permet donc de séparer les étapes d'insertion automatisée des bascules « scan » faite par l'outil DFTAdvisor et la génération automatisée des vecteurs de test faite par l'outil Fastscan. Pour le circuit b12, qui est formé de 121 bascules, le résultat de cette insertion sera le nouveau circuit b12\_scan en format Verilog, un fichier des commandes (de type dofile) et une procédure de test qui vont aider à lancer Fastscan. En mode Fault du Fastscan (signifiant simulation de pannes), nous avons obtenu la liste de toutes les pannes du circuit et la couverture des pannes par les vecteurs de test générés.

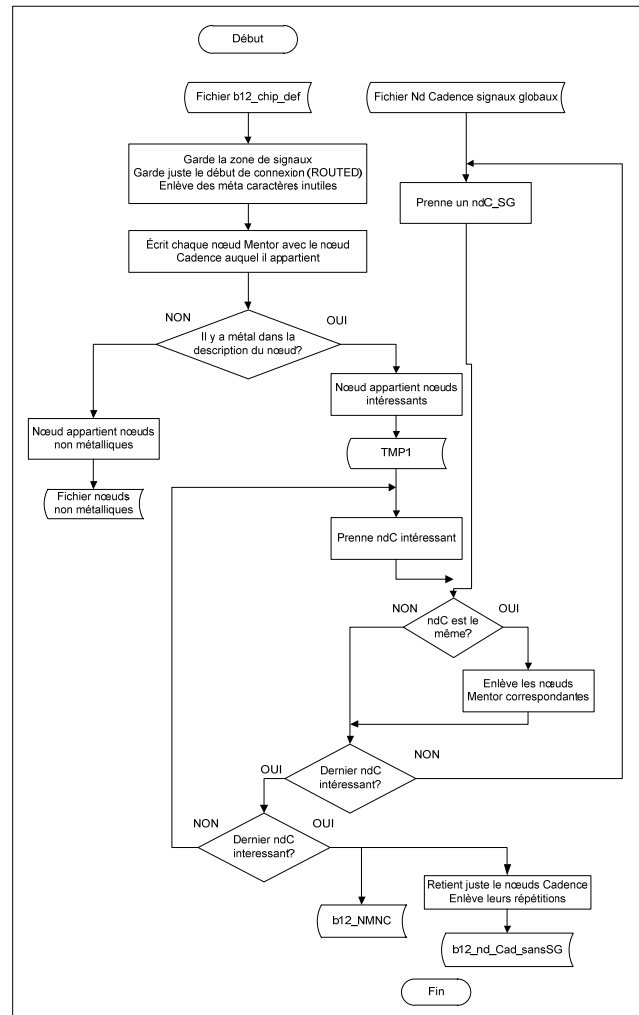


Figure 3-3 Traitement pour obtenir les nœuds de Mentor et le nœud de Cadence duquel il appartient.

Notons que dans le cadre de ce projet, nous utilisons le modèle de pannes « collé à (stuck-at) 0 et à 1 ». Le tableau 3 montre le rapport statistique d'analyse des pannes effectué par Fastscan sur le circuit b12. À l'annexe III, nous avons donné des exemples des fragments de la chaîne d'analyse et un fragment du fichier qui contient les vecteurs de test ATPG du circuit b12. En bout de ligne, les outils Mentor nous permettent d'obtenir la liste des pannes testables (4998) et non-testables (286) du circuit cible b12 (à voir aussi la note explicative du tableau 3.1). Il est important de noter qu'il y a des nœuds qui peuvent se trouver dans les deux fichiers. En effet, puisqu'il y a 2 pannes possibles par nœud (collé à 0 et collé à 1), il se



peut qu'une de ces 2 pannes soit non testable et que l'autre panne soit testable. La dernière étape préparatoire consiste à établir l'équivalence entre les noms des bascules Cadence et Mentor. Cette étape nécessite : 1) la création d'un fichier de commandes qui contient toutes les pannes testables du circuit avec lequel nous exécutons une analyse des pannes sous Fastscan en gardant la transcription de l'exécution, et 2) le traitement du fichier de transcription à l'aide d'un script en langage *awk*, qui consiste à enlever l'excès de texte et retenir juste les mots pertinents.

Tableau 3.1 Rapport statistique d'analyse des pannes du Fastscan pour le circuit b12

	Statistics	report
-----		
	#faults	#faults
fault class	(coll.)	(total)
FU (full)	3738	5284
-----	-----	-----
DS (det_simulation)	2666	4130
DI (det_implication)	842	868
RE (redundant)	119	175
AU (atpg_untestable)	111	111
-----	-----	-----
test_coverage	96.93%	97.83%
fault_coverage	93.85%	94.59%
atpg_effectiveness	100.00%	100.00%
-----		
#test_patterns		114
#simulated_patterns		128
CPU_time (secs)		0.8

Note explicative

Les valeurs antérieurement présentées sont retrouvées dans ce tableau.

4130 + 868=4998 pannes testables et

175 + 111=286 pannes non testables

### **3.4 Conclusions**

Dans ce chapitre, nous avons présenté les étapes préparatoires à la création des signatures. Dans les 4 prochains chapitres, nous allons présenter les différentes signatures développées dans le cadre de ce projet.

## CHAPITRE 4

### PREMIÈRE MÉTHODE – SIGNATURE D’UN CIRCUIT BASÉE SUR LES NŒUDS COUVERTS PAR LES BASCULES D’OBSERVATION

#### 4.1 Introduction

Dans ce chapitre, nous présentons la première des quatre types de signatures développées dans le cadre de ce projet. Il s’agit de la signature la plus simple et la plus rapide à calculer. Elle suppose que les pannes sont équiprobables.

#### 4.2 Définition de la première signature

Comme mentionné précédemment, les signatures développées dans le cadre de ce mémoire constituent une estimation de la probabilité d’apparition de valeurs erronées dans chacune des bascules d’intérêt d’un circuit sous test, lors d’un test de plusieurs exemplaires de ce même circuit. Notons que les signatures peuvent être appliquées sur un sous-ensemble de bascules du circuit. De manière plus spécifique, les diverses signatures sont définies sous la forme:

$$S_m = [X_1 X_2 \dots \dots X_i \dots X_m] \quad (4.1)$$

où  $S_m$  est la signature de la zone d’intérêt, un vecteur matrice ligne;

$X_i$  représente la probabilité que la bascule  $i$  contienne une valeur erronée

$m$  est le nombre de bascules d’intérêt considérées dans le calcul des signatures.

Comme la première signature est basée sur l’hypothèse que les pannes sont équiprobables, le calcul de cette signature consiste à compter le nombre de pannes détectables à chacune des bascules d’intérêt. À cette fin, nous introduisons un coefficient de testabilité,  $T_{ij}$ , lié à la

capacité de l'outil Fastscan de détecter les pannes possibles affectant les nœuds du circuit. Pour notre modèle, nous ne considérons que les pannes de type collé à (« stuck-at ») 1 et à 0. Notre coefficient de testabilité sera égal à 1 si les deux pannes sont détectables pour un nœud ciblé  $j$  via la bascule d'intérêt  $i$ , à 0,5 si juste une panne est détectable et à 0 si aucune panne ne peut être détectée par la bascule pour ce nœud. En considérant ces informations, le terme  $X_i$  devient :

$$X_i = \frac{\sum_{j=1}^n T_{ij}}{A_{int}} \quad (4.2)$$

où  $n$  est le nombre de nœuds testables de la zone d'intérêt ;

$T_{ij}$  est le coefficient de testabilité du nœud  $j$  via la bascule  $i$ ;

$A_{int}$  est désignée comme la surface d'intérêt. Elle contient tous les nœuds testables par l'ensemble des bascules d'intérêt. Elle peut être vue comme l'agglomération de petites cases dans lesquelles peuvent tomber les défauts menant à une valeur erronée dans une bascule d'intérêt. Elle est calculée comme suit :

$$A_{int} = \sum_{j=1}^n A_N(j) \quad (4.3)$$

où  $A_N(j)$  la sensibilité relative du nœud/case  $j$  à recevoir une défectuosité menant à une valeur erronée dans une bascule d'intérêt. Cette sensibilité relative est fonction du nombre de pannes ( $s@0$ ,  $s@1$ ) pouvant affecter le nœud  $j$  qui sont détectables via l'ensemble des bascules d'intérêt.  $A_N(j)$  sera respectivement égal à 1, 0.5 et 0 si les 2 pannes, une seule des deux pannes, et aucune des 2 pannes sont/est détectable via ces mêmes bascules.

Un exemple de calcul de la signature pour un circuit fictif suit, dans le cas quand la pondération des nœuds est constante et égale à 1. Prenons un circuit fictif qui contient 5 bascules d'analyse interne  $B_1$  jusqu'à  $B_5$  et 10 nœuds (Mentor) conforme à la figure 4.1 et au tableau 4.1. Supposons que nous voulons calculer la signature de la zone d'intérêt couverte par les bascules  $B_1$ ,  $B_2$  et  $B_3$ .

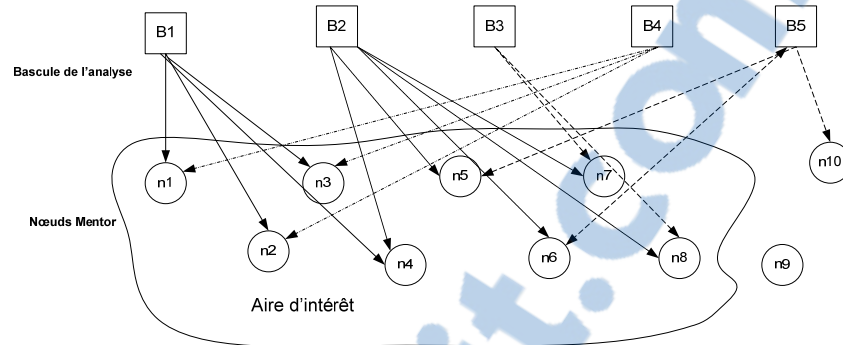


Figure 4-1 Les bascules et les nœuds du circuit fictif

Tableau 4.1 Circuit fictif et l'analyse de Fastscan pour ses bascules

Bascule	Nom des nœuds couverts	Pannes couvertes par les bascules	Observations
$B_1$	$n_1, n_2, n_3, n_4$	$n_1@0, n_1@1, n_2@0, n_2@1, n_3@0, n_3@1, n_4@0, n_4@1$	4 nœuds complètement couverts
$B_2$	$n_4, n_5, n_6, n_7, n_8$	$n_4@0, n_5@0, n_5@1, n_6@0, n_6@1, n_7@0, n_8@0, n_8@1$	3 nœuds complètement couverts et 2 nœuds partiellement couverts
$B_3$	$n_7, n_8$	$n_7@0, n_8@0, n_8@1$	1 nœud complètement couvert et 1 nœud partiellement couvert
$B_4$	$n_1, n_2, n_3$	$n_1@0, n_1@1, n_2@0, n_2@1, n_3@0, n_3@1$	3 nœuds complètement couverts
$B_5$	$n_5, n_6, n_{10}$	$n_5@0, n_5@1, n_6@0, n_6@1, n_{10}@0, n_{10}@1$	3 nœuds complètement couverts

Note. Le terme  $n_j@k$  indique que le nœud  $j$  est couvert pour la panne  $s@k$ , ( $k=0$  ou  $1$ )

Cette zone contient 8 nœuds, de  $n_1$  à  $n_8$ . Basé sur les données présentées plus bas (Tableau 4.2), nous calculons les coefficients de testabilité comme suit (Tableau 4.3). La sensibilité de chaque nœud apparaît au Tableau 4.3.

Tableau 4.2 Coefficients de testabilité sur les nœuds du circuit de notre cas d'étude

Bascule	Nœuds couverts	Coefficients de testabilité
$B_1$	$n_1, n_2, n_3, n_4$	$T_{11}=1, T_{12}=1, T_{13}=1, T_{14}=1, T_{15}=0, T_{16}=0, T_{17}=0, T_{18}=0$
$B_2$	$n_5, n_4, n_6, n_7, n_8$	$T_{21}=0, T_{22}=0, T_{23}=0, T_{24}=0,5, T_{25}=1, T_{26}=1, T_{27}=0,5, T_{28}=1$
$B_3$	$n_7, n_8$	$T_{31}=0, T_{32}=0, T_{33}=0, T_{34}=0, T_{35}=0, T_{36}=0, T_{37}=0,5, T_{38}=1$

Tableau 4.3 La sensibilité des nœuds de la zone d'intérêt pour notre cas d'étude

Nœud	$n_1$	$n_2$	$n_3$	$n_4$	$n_5$	$n_6$	$n_7$	$n_8$
$n_j @ 0$	oui	oui	oui	oui	oui	oui	oui	oui
$n_j @ 1$	oui	oui	oui	oui	oui	oui	non	oui
$A_N(j)$	1	1	1	1	1	1	0.5	1

Nous calculons la surface d'intérêt comme est montré dans l'équation 4.3 :

$$A_{int} = \sum_{j=1}^8 A_N(j) = 1 + 1 + 1 + 1 + 1 + 1 + 0,5 + 1 = 7,5$$

Pour  $B_1$  la signature est  $X_1 = \frac{\sum_{j=1}^8 T_{1j}}{A_{int}} = \frac{1+1+1+1}{7,5} = 0,53$

De la même manière :

$$X_2 = \frac{\sum_{j=1}^8 T_{2j}}{A_{int}} = \frac{4}{7,5} = 0,53 \quad X_3 = \frac{\sum_{j=1}^8 T_{3j}}{A_{int}} = \frac{1,5}{7,5} = 0,2$$

Donc la signature recherchée est:  $S_3 = [X_1 \ X_2 \ X_3] = [0,53 \ 0,53 \ 0,2]$

### 4.3 Traitement spécifique pour l'obtention de la signature

Pour produire la signature, nous commençons par l'analyse des pannes avec l'outil Fastscan pour chaque bascule et un fichier de commandes (dofile) avec la totalité de pannes du circuit. Chacune des commandes incluses dans ce fichier doit permettre à l'analyse de poursuivre en cas de détection d'une panne par simulation, implication ou inactive (non testable). En effet, par défaut l'analyse se termine à la première détection mais les commandes vont produire une analyse complète. Le résultat de l'exécution de ces commandes sera enregistré comme un fichier de transcription qui ensuite sera traité par plusieurs programmes. La figure 4.2 montre le traitement à partir de cette transcription pour obtenir la signature du circuit b12 (nous donnons plus des détails sur les traitements individuels à l'annexe V). Notons que la plupart des traitements sont effectués à l'aide de l'utilitaire awk. Pour atteindre notre but, il est premièrement nécessaire de trouver pour toutes les bascules leur couverture des nœuds à tester (Figure 4-2, bloc du traitement conforme à la figure-A V.1). Par un processus de sélection, nous retenons un certain nombre de bascules avec leur couverture. Le choix des bascules retenues pour la signature (illustré de manière symbolique par l'accolade de la figure 4.2) se fait en trois temps :

- 1) en créant une liste des bascules dans un ordre décroissant de la couverture des nœuds;
- 2) en tronquant cette liste pour ne garder que les bascules couvrant le plus de nœuds;
- 3) en choisissant de manière aléatoire les bascules d'intérêt parmi les bascules avec la plus grande couverture.

De manière arbitraire, la taille de la liste tronquée a été fixée à 2 fois le nombre de bascules d'intérêt pour avoir une représentativité sur la généralité de notre méthode. Ce nombre représente un compromis entre la représentativité du sous-ensemble versus le circuit complet et les ressources nécessaires pour la mise en œuvre des signatures. Par exemple, pour le circuit b12, nous avons fixé (de manière arbitraire) le nombre de bascules d'intérêt à 15. Voici la liste des bascules choisies qui seront utilisées dans les calculs de signatures: B1826, B1828, B1829, B1830, B1831, B1832, B1845, B1847, B1848, B1849, B1850, B1857, B1868, B1940, B1941.

Pour poursuivre le calcul de la signature, il nous faut tout d'abord calculer le coefficient de testabilité qui sera associé à chacun des nœuds détectables par les bascules d'intérêt. Ce calcul débute avec le fichier de pannes testables fournis par Fastscan, qui est traité (conformément à la figure-A V.3) afin de calculer le coefficient de testabilité des nœuds associés à ces pannes. Notons que nous avons enlevé au préalable les nœuds qui appartiennent aux signaux globaux détectables par la majorité des bascules, une opération qui permettra de donner plus de sensibilité à l'apport des signaux locaux dans la signature élaborée. Deux fichiers sont produits, le premier (*liste\_nd\_M\_sansG*) contenant les nœuds Mentor complètement testables et le second (*b12\_noeuds\_sansSG*) l'ensemble des nœuds Mentor associés aux pannes testables. Le calcul se poursuit avec le fichier de pannes non testables fournis par Fastscan, qui est traité (conformément à la figure-A V.2) afin de calculer le coefficient de testabilité des nœuds associés à ces pannes. Il est important de noter ici qu'un nœud partiellement testable peut être associé à la fois une panne testable et à une non-testable. C'est la raison pour laquelle un traitement supplémentaire est requis (conformément à la figure-A V.4) pour produire une liste unique des nœuds partiellement ou complètement testables avec leur coefficient de testabilité (*b12\_testability\_M\_noeuds\_sansSG*).

Notons que nous ajoutons également comme information dans cette liste le nombre de nœuds Mentor associés à chaque nœud Cadence, information tirée du fichier de la liste des connections (en format DEF), qui nous fournit la liste des nœuds de Cadence et leurs nœuds associés de Mentor. Cette information supplémentaire n'est pas utilisée pour ce premier type de signature, mais qui servira plus tard pour les autres types. Le fichier *b12\_testability\_M\_noeuds\_sansSG* est par la suite utilisé afin de pondérer les nœuds détectables par chaque bascule d'intérêt de leur coefficient de testabilité (traitement conforme aux figures-A V.5 et V.6), résultant pour chaque bascule d'intérêt d'un fichier dit ``vue'', avec les nœuds couverts et leur coefficient de testabilité. Notons qu'une étape de filtrage est effectuée pour ne conserver que les nœuds associés aux couches métalliques, pour demeurer compatibles avec les autres types de signatures.



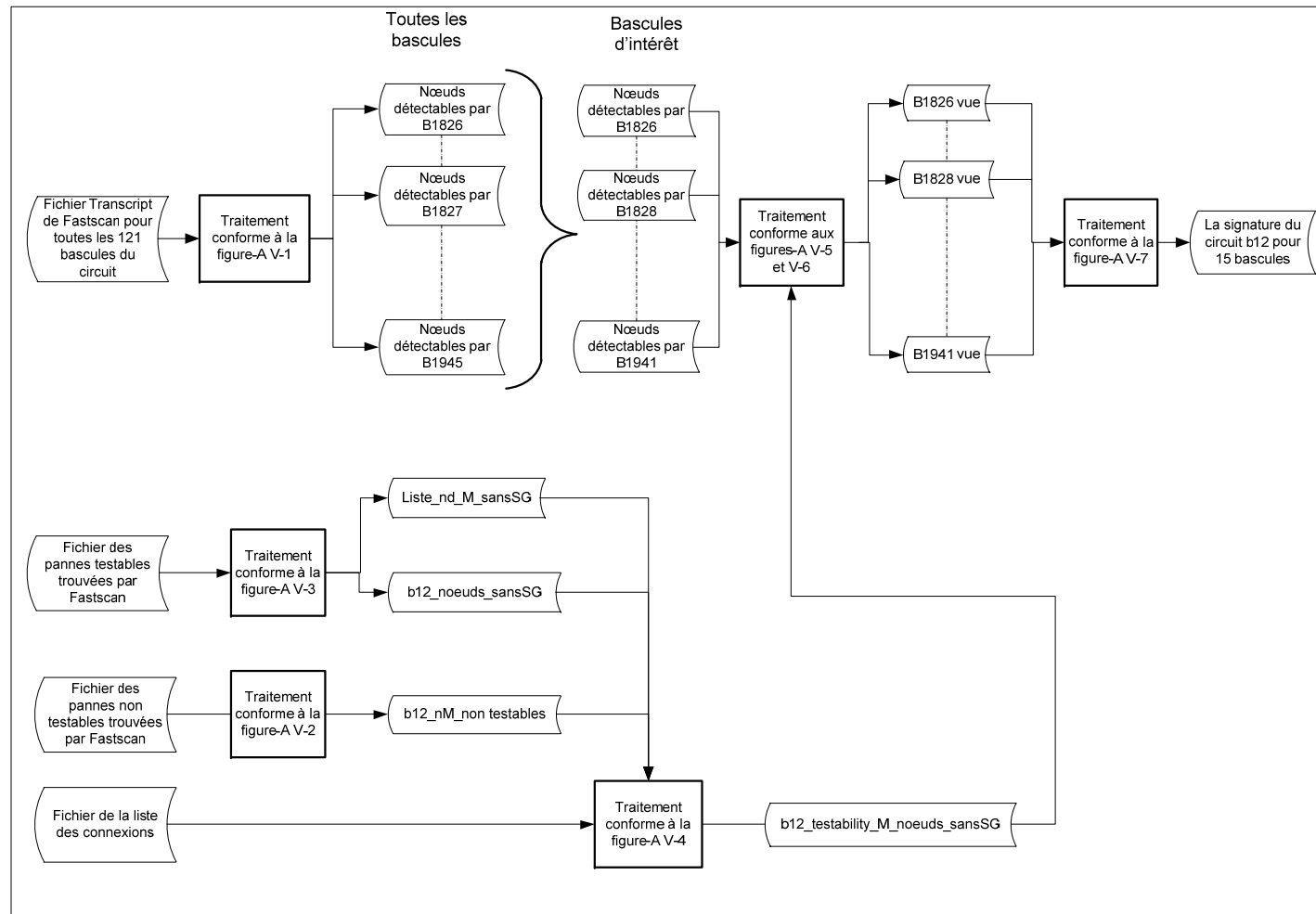


Figure 4-2 Traitement pour l'obtention de la signature à pondération constante (Méthode 1).

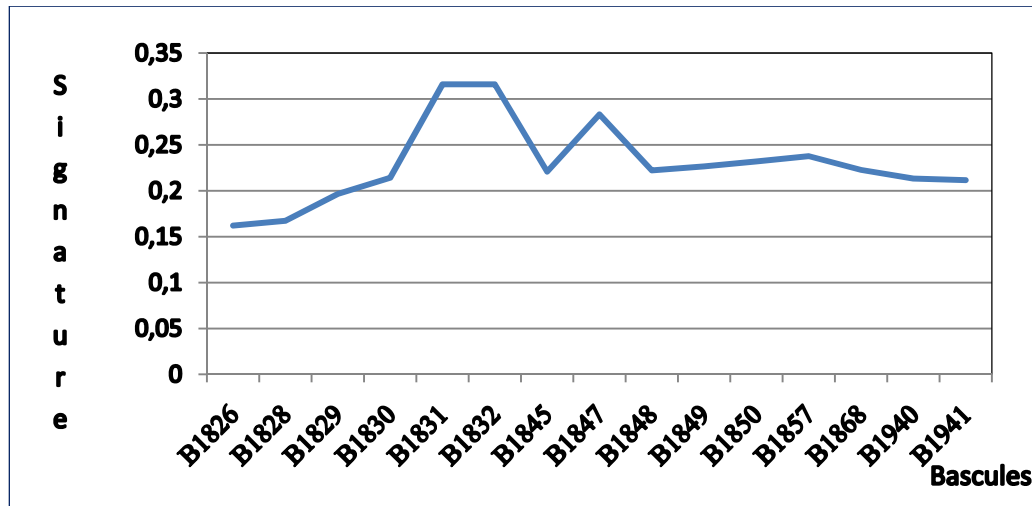


Figure 4-3 Signature à pondération constante du circuit b12 pour 15 bascules d'intérêt.

À partir de ces fichiers "vue", nous calculons (conformément au traitement de la figure-A V.7 et aux équations précédentes) la signature de chaque bascule impliquée dans la signature du circuit. Le résultat de cette procédure est montré à la figure 4.3.

#### 4.4 Conclusions

Nous avons présenté dans ce chapitre le premier type de signature, qui suppose que toutes les pannes sont équiprobables. Cette signature peut servir comme moyen de rapide de vérification des puces pour le triage et un déclencheur d'alarme dans les cas où les défauts sont plus nombreuses que prévu. Elle n'est pas en mesure de pointer précisément la source et la localisation exacte de pannes, mais à cause de son court traitement, il est préférable qu'elle soit utilisée en première étape.

Les résultats de ce premier type de signatures serviront comme référence pour les autres types développés dans ce mémoire.

## CHAPITRE 5

### DEUXIÈME MÉTHODE – LA SIGNATURE DU CIRCUIT PONDÉRÉE PAR LES CAPACITÉS PARASITES DES NŒUDS

#### 5.1 Introduction

Dans ce chapitre, nous présentons le deuxième des quatre types de signatures développées dans le cadre de ce projet. À partir de ce type, toutes les signatures vont ajouter une fonction de pondération des nœuds de Mentor dans leur contribution à la signature. Dans ce chapitre, la pondération est déterminée par la capacité parasite du nœud. La capacité parasite est reliée à la dimension du nœud technologique utilisé et à son répartition sur le dessin de masque. Nous supposons que la sensibilité d'un nœud est proportionnelle à cette capacité qui elle, est en première approximation proportionnelle à la surface du nœud. Cette capacité parasite est obtenue directement du dessin des masques du circuit produit par Cadence.

#### 5.2 Définition de la deuxième signature

Comme mentionné au chapitre 4, les signatures développées constituent une estimation de la probabilité d'apparition de valeurs erronées dans chacune des bascules d'intérêt d'un circuit sous test, lors d'un test de plusieurs exemplaires de ce même circuit. La différence par rapport à la signature antérieurement calculée est la pondération des nœuds Mentor assignée en fonction de leur capacité parasite, ce qui inclut le fait que plusieurs nœuds Mentor peuvent correspondre à un même nœud Cadence.

Dans la forme déjà connue:

$$S_m = [X_1 X_2 \dots \dots X_i \dots X_m] \quad (5.1)$$

Où  $S_m$  est la signature de la zone d'intérêt, un vecteur matrice ligne;

$X_i$  représente la probabilité que la bascule  $i$  contienne une valeur erronée;

$m$  est le nombre de bascules d'intérêt considérées dans le calcul des signatures;

le terme  $X_i$  est de la forme :

$$X_i = \frac{\sum_{j=1}^n T_{ij} * Cpp_j * \frac{1}{NMNC_j}}{A_{intpc}} \quad (5.2)$$

où

$n$  est encore le nombre des nœuds testables de la zone d'intérêt ;

$T_{ij}$  est encore le coefficient de testabilité du nœud  $j$  établi par rapport au nombre des pannes testables l'affectant et couvertes par la bascule  $i$ ;

$Cpp_j$  est la capacité parasite du nœud Cadence associé au nœud Mentor  $j$  basé sur le dessin des masques du circuit;

$NMNC_i$  est le nombre des nœuds Mentor connectés au nœud de Cadence qui est associé au nœud Mentor  $j$ ;

$A_{intpc}$  est la surface d'intérêt du circuit, pondérée par la capacité parasite

$$A_{intpc} = \sum_{j=1}^n A_N(j) * Cpp_j * \frac{1}{NMNC_j} \quad (5.3)$$

où

$A_N(j)$  est toujours la sensibilité relative (non pondérée par le capacité parasite) du nœud/case  $j$  à recevoir une défectuosité menant à une valeur erronée dans une bascule d'intérêt. Cette sensibilité relative est encore fonction du nombre de pannes ( $s@0$ ,  $s@1$ ) pouvant affecter le nœud  $j$  qui sont détectables via l'ensemble des bascules d'intérêt.  $A_N(j)$  est à nouveau respectivement égal à 1, 0.5 et 0 si les 2 pannes, une seule des deux pannes, et aucune des 2 pannes sont/est détectable via ces mêmes bascules.

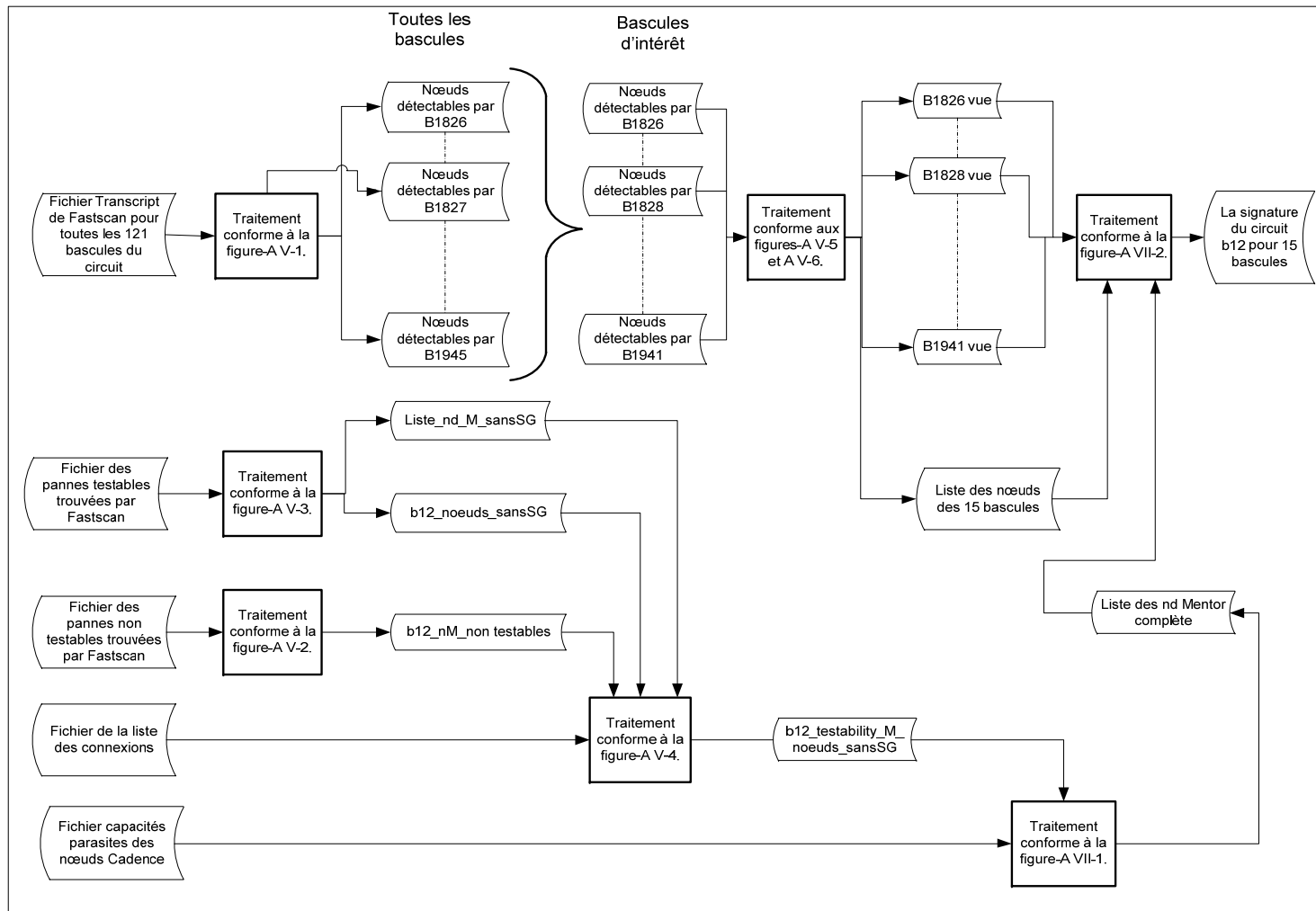


Figure 5-1 Traitement pour l'obtention de la signature pondérée par les capacités parasites des nœuds (Méthode 2).

### **5.3        Traitement spécifique pour l'obtention de la signature**

Puisque la procédure d'obtention de cette signature est similaire à celle de la première méthode, nous ne présentons dans ce chapitre que les différences par rapport à celle-là. La figure 5-1 montre le déroulement du traitement spécifique pour obtenir la signature proposée. On remarque l'apparition du fichier *Capacités parasites des nœuds Cadence* produit par l'outil de Cadence, First Encounter, qui fournit les valeurs des capacités parasites des nœuds. Celui-ci va ajouter ces capacités à notre base de données suite à un traitement présenté de manière plus détaillée à la figure-A VII-1 de l'annexe VII. Les autres traitements sont identiques à ceux présentés au chapitre précédent, sauf le traitement final qui produit les signatures. Ce traitement est présenté à la figure-A VII.2 de l'annexe VII. Le résultat de cette procédure est montré à la figure 5-2. On note que les deux premières signatures développées ont une forme similaire.

### **5.4        Conclusions**

Nous avons présenté dans ce chapitre le deuxième type de signature, qui ajoute un poids au nœud relié à sa capacité parasite. Cette information supplémentaire ne modifie pas l'allure générale de la signature.

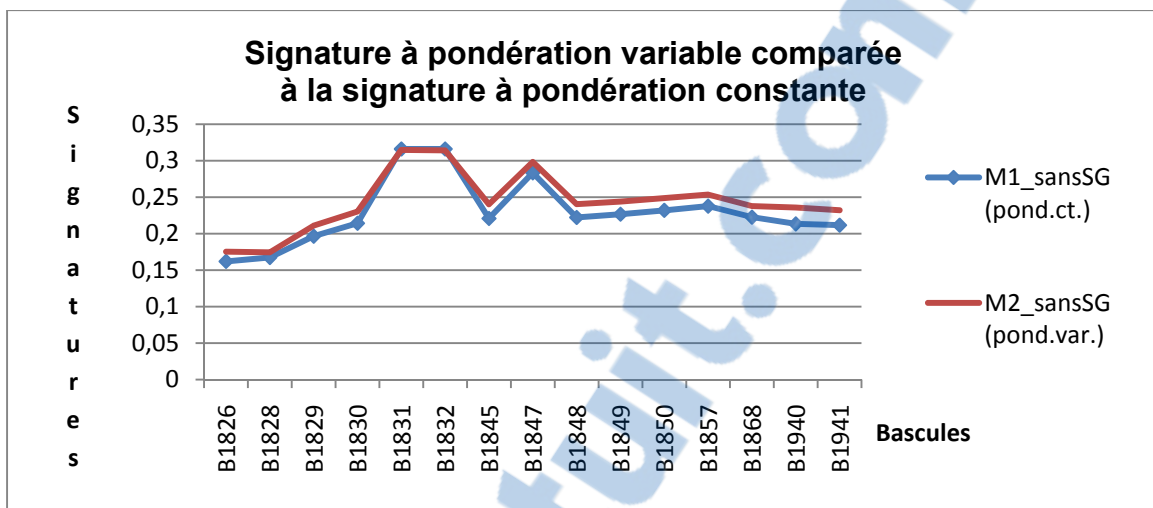


Figure 5-2 Les signatures à pondération variable (en fonction de la capacité parasite) et à pondération constante.

## CHAPITRE 6

### TROISIÈME MÉTHODE - SIGNATURE DU CIRCUIT PONDÉRÉE PAR LA RÉPARTITION DES PANNES SUR LES COUCHES MÉTALLIQUES

#### 6.1 Introduction

Dans ce chapitre, nous présentons le troisième des quatre types de signatures développées dans le cadre de ce projet. De la même façon que pour la méthode antérieure, cette signature utilise une pondération des nœuds de Mentor dans leur contribution à la nouvelle signature. Dans le cas du troisième type, la pondération est déterminée par les couches métalliques qui touchent les nœuds. Nous supposons que la sensibilité d'un nœud est proportionnelle au nombre de ces couches, qui est en première approximation proportionnelle à la surface du nœud. Puisque les couches métalliques ne sont pas fournies directement par aucun outil, nous devons ajouter un traitement supplémentaire pour les obtenir du dessin des masques.

#### 6.2 Définition de la troisième signature

La différence par rapport aux 2 signatures antérieurement calculées est la pondération des nœuds Mentor assignée en fonction du nombre des couches métalliques, ce qui inclut encore le fait que plusieurs nœuds Mentor peuvent correspondre à un même nœud Cadence.

La signature a toujours la forme déjà présentée:

$$S_m = [X_1 X_2 \dots \dots X_i \dots X_m] \quad (6.1)$$

Où  $S_m$  est la signature de la zone d'intérêt, un vecteur matrice ligne;

$X_i$  représente la probabilité que la bascule  $i$  contienne une valeur erronée

$m$  est le nombre de bascules d'intérêt considérées dans le calcul des signatures

le terme  $X_i$  est de la forme :



$$X_i = \frac{\sum_{j=1}^n T_{ij} * Fm_j * \frac{1}{NMNC_j}}{A_{intcm}} \quad (6.2)$$

où  $n$  est encore le nombre des nœuds testables de la zone d'intérêt ;

$T_{ij}$  est toujours le coefficient du nœud  $j$  établi par rapport au nombre de pannes testables l'affectant et couvertes par la bascule  $i$ ;

$Fm_j$  est le nombre de couches métalliques associée au nœud  $j$  et obtenu par le traitement à la figure VIII.1 appliqué au dessin des masques du circuit;

$NMNC_i$  est encore le nombre de nœuds Mentor connectés au nœud de Cadence qui est associé au nœud Mentor  $j$ ;

$A_{intcm}$  est la surface d'intérêt du circuit, pondérée par le nombre de couches

$$A_{intcm} = \sum_{j=1}^n A_N(j) * Fm_j * \frac{1}{NMNC_j} \quad (6.3)$$

où  $A_N(j)$  correspond toujours à la sensibilité relative du nœud/case  $j$  à recevoir une défectuosité menant à une valeur erronée dans une bascule d'intérêt.

### 6.3 Traitement spécifique pour l'obtention de la signature

Puisque la procédure d'obtention de cette signature est similaire à celle de la première méthode, nous ne présentons dans ce chapitre que les différences par rapport à celle-ci. La figure 6-1 montre le déroulement du traitement spécifique pour obtenir la signature proposée. Le point de départ pour l'obtention des couches métalliques sera le fichier DEF du dessin des masques produit par Cadence. Suite à un traitement présenté plus en détails à la figure-A VIII.1 de l'annexe VIII, on obtient le fichier *Liste nœuds couches*. Ensuite on ajoute ce paramètre au fichier *b12\_testability\_M\_noeuds\_sansSG* pour obtenir notre nouvelle base de données, *Liste de nd Mentor complète*, grâce à une procédure présentée à la figure-A VIII.2 de l'annexe VIII.

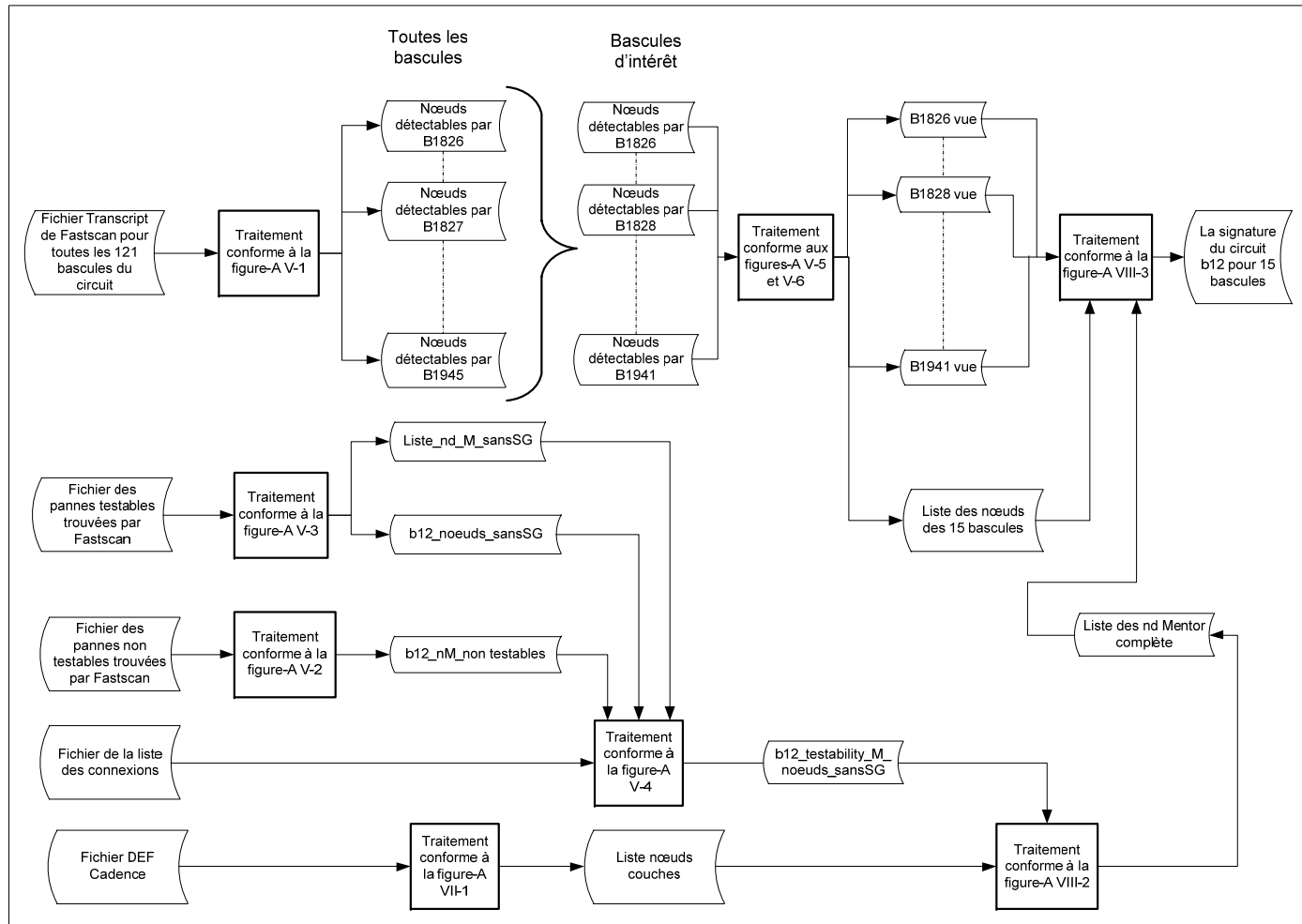


Figure 6-1 Traitement pour l'obtention de la signature pondérée par les couches métalliques des nœuds (Méthode 3).

Suite à cette procédure nous associons aux nœuds de Mentor le nombre de couches qui les touchent. Les autres traitements sont identiques comme les traitements présentés au chapitre 4, sauf le traitement final qui produit les signatures. Ce traitement est présenté à la figure-A VIII.3 de l'annexe VIII. Le résultat de cette procédure est montré à la figure 6.2. Encore ici, on peut observer que la nouvelle information ne modifie pas l'allure globale de la signature.

## 6.4 Conclusions

Nous avons présenté dans ce chapitre le troisième type de signature, qui ajoute un poids au nœud relié au nombre des couches métalliques qui le touchent. L'allure de la signature demeure la même.

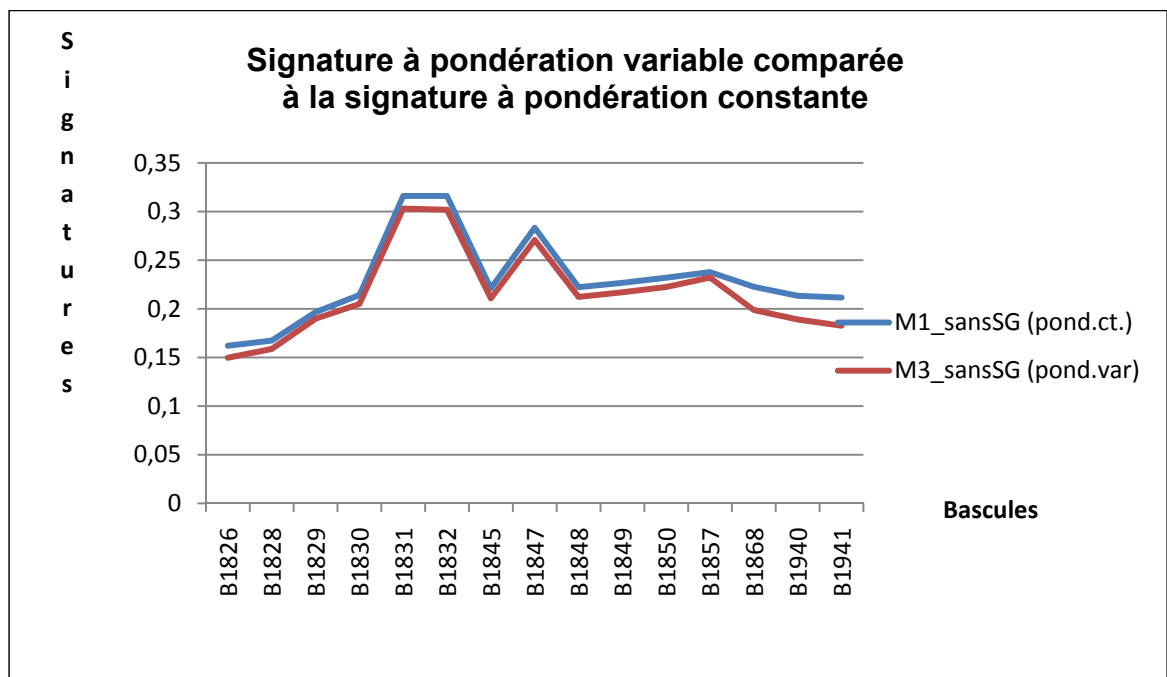


Figure 6-2 Les signatures à pondération variable (en fonction du nombre des couches métalliques) et à pondération constante

## CHAPITRE 7

### QUATRIÈME MÉTHODE-SIGNATURES DU CIRCUIT PONDÉRÉE PAR LES DÉPASSEMENTS DES RÈGLES DE DRC

#### 7.1 Introduction

Dans ce chapitre, nous présentons le dernier type de signatures développées selon trois variantes. De la même manière que les méthodes antérieures, ces trois dernières signatures utilisent une pondération des nœuds de Mentor dans leur contribution à la signature. Cette fois-ci, nous supposons que la sensibilité d'un nœud est une fonction liée aux règles du dessin des masques. Cette sensibilité dépend en effet de la marge entre les règles minimales à respecter et les dimensions et distances apparaissant aux endroits d'intérêt sur le dessin des masques. Pour ce faire, nous utilisons l'outil de vérification des règles de dessin (DRC, « Design Rule Checking ») de l'environnement Cadence: nous ajustons les règles afin qu'elles soient plus conservatrices (i.e. définies de telle sorte que les dimensions et les distances minimales doivent être plus grandes que celles des règles nominales). Ceci se traduit par des structures présentant peu ou pas de marge par rapport à ces règles conservatrices causant des violations qui sont représentées par des polygones surlignant leur localisation. La première des 3 variantes/signatures développées se sert de la même commande de Fastscan que les premiers types déjà présentés. La deuxième variante utilise une autre commande de Fastscan pour obtenir la couverture des bascules. La troisième variante utilise la même commande de Fastscan que la deuxième, mais cherche en plus la première apparition d'une panne dans la couverture des bascules qui sont dans la zone d'intérêt, les apparitions suivantes étant négligées. Dans chaque variante, le coefficient de testabilité est différent alors que la pondération demeure la même

## 7.2 Définition du quatrième type de signatures (commun aux 3 variantes)

L'hypothèse de départ sur laquelle est basé le quatrième type de signature (et ses 3 variantes) est que les pannes systématiques sont étroitement liées aux surfaces des polygones DRC. Pour des fins de modélisation, nous définissons 2 types de surface. Dans un premier temps, nous appelons surface réelle la surface des polygones tels qu'ils sont produits par l'outil DRC. Considérant que les polygones de différentes natures (i.e. liés à différentes règles de dessin) peuvent afficher une sensibilité différente aux défauts, nous pondérons ces surfaces réelles pour un facteur ( $k_p \leq 1$ ) ce qui nous permet d'introduire dans un deuxième temps la notion de surface efficace des polygones DRC. L'indice  $p$  du facteur de pondération est l'index des règles considérées (dans notre cas,  $p$  prend des valeurs de 1 à 15, car 15 règles de design sont considérées, voir aussi le tableau 7.1).

Rappelons que les signatures développées constituent une estimation de la probabilité d'apparition de valeurs erronées dans chacune des bascules d'intérêt d'un circuit sous test, lors d'un test en production de plusieurs exemplaires de ce même circuit. La différence par rapport aux signatures antérieurement calculées est la pondération des nœuds Mentor qui est maintenant assignée en fonction de leur surface efficace. D'un point de vue modélisation, nous considérons encore la surface de la puce (et le cas échéant la zone d'intérêt) comme un ensemble de cases dans lesquelles les défauts peuvent tomber. La zone d'intérêt, lorsqu'elle est un sous-ensemble de la surface de la puce, est définie de la même manière, i.e. à partir des bascules d'intérêt et en remontant en amont jusqu'aux nœuds de Cadence, en passant par les nœuds de Mentor. Ici, le nombre de cases associées à chaque nœud de Cadence d'intérêt pour une règle donnée dépend de la surface des polygones DRC liés à ce nœud pour cette règle et de la sensibilité de la règle à être le foyer de défauts. La probabilité qu'un nœud de Cadence soit défectueux devient égale au rapport entre d'une part la somme des surfaces efficaces liées aux polygones générés en fonction des règles s'appliquant pour ce nœud, et d'autre part la surface efficace totale de la zone d'intérêt.

La signature a toujours la forme déjà présentée:

$$S_m = [X_1 X_2 \dots \dots X_i \dots X_m] \quad (7.1)$$

où  $S_m$  est la signature de la zone d'intérêt, un vecteur matrice ligne;

$X_i$  représente la probabilité que la bascule  $i$  contienne une valeur erronée

$m$  est le nombre de bascules d'intérêt considérées dans le calcul des signatures.

Le terme  $X_i$  est défini :

$$X_i = \frac{\sum_{p=1}^r \left( k_p * \sum_{j=1}^n T_{ij} * Ar_{pij} * \frac{1}{NMNC_j} \right)}{Aint_{sp}} \quad (7.2)$$

où  $n$  est encore le nombre de nœuds testables de la zone d'intérêt ;

$T_{ij}$  est toujours le coefficient de testabilité du nœud Mentor  $j$  établi par rapport au nombre de pannes testables l'affectant et couvertes par la bascule  $i$ ;

$Ar_{pij}$  est la surface réelle des polygones produits autour de nœud Mentor  $j$  par violation de la règle  $p$  et couvert par la bascule  $i$

$NMNC_i$  est encore le nombre de nœuds Mentor connectés au nœud de Cadence qui est associé au nœud Mentor  $j$ ;

$Aint_{sp}$  est la surface efficace totale d'intérêt du circuit, pondérée par le nombre des polygones produits pour chaque règle. Ce terme est défini :

$$Aint_{sp} = \sum_{p=1}^r \left( k_p * \sum_{j=1}^n A_N(j) * Ar_{pij} * \frac{1}{NMNC_j} \right) \quad (7.3)$$

où  $A_N(j)$  correspond toujours à sensibilité relative du nœud/case  $j$  à recevoir une défectuosité menant à une valeur erronée dans une bascule d'intérêt

Pour notre cas d'étude,  $r = 15$  règles. Les règles choisies sont décrites au tableau 7.1. Les règles ont été arbitrairement ajustées afin de créer des dépassements/violations d'environ 3,5%. Nous avons également choisi  $i = 15$  bascules. Nous pouvons ainsi obtenir un système

d'équations où les racines sont déterminées et uniques. Ce système est décrit plus loin, à la section 7.3.3.1. Les 15 bascules, choisies parmi les 121 bascules scan appartenant au circuit b12, sont les mêmes que celles des chapitres précédents (i.e.  $B_{1826}$  à  $B_{1941}$ ).

Tableau 7.1 Règles de conception du dessin des masques, considérées pour le calcul de la pondération des nœuds

Index de la règle considérée	Appellation du fabricant	Définition (Mx définit la couche métallique x)	Règle de dessin de masque ( $\mu\text{m}$ )	Condition de test pour la signature ( $\mu\text{m}$ )
1	M1.S.1	Espacement minimal entre deux régions M1	$M1 = 0,22$	$M1 = 0,228$
2	M2.S.1	Espacement minimal entre deux régions M2	$M2 = 0,28$	$M2 = 0,29$
3	M3.S.1	Espacement minimal entre deux régions M3	$M3 = 0,28$	$M3 = 0,29$
Pas d'index puisque il n'y a pas des violations	M4.S.1	Espacement minimal entre deux régions M4	$M4 = 0,28$	$M4 = 0,29$
4	M5.S.1	Espacement minimal entre deux régions M5	$M5 = 0,28$	$M5 = 0,29$
Pas d'index puisque il n'y a pas des violations	M6.S.1	Espacement minimal entre deux régions M6	$M6 = 0,46$	$M6 = 0,476$
5	M1.W.1	Largeur minimale de la région M1	$M = 0,22 - 0,24$	$M1 = 0,248$
6	M2.W.1	Largeur minimale de la région M2	$M2 = 0,28$	$M2 = 0,29$
7	M3.W.1	Largeur minimale de la région M3	$M3 = 0,28$	$M3 = 0,29$
8	M4.W.1	Largeur minimale de la région M4	$M4 = 0,28$	$M4 = 0,29$

Index de la règle considérée	Appellation du fabricant	Définition (M <sub>x</sub> définit la couche métallique x)	Règle de dessin de masque (μm)	Condition de test pour la signature (μm)
9	M5.W.1	Largeur minimale de la région M5	M5 = 0,28	M5 = 0,29
10	M6.W.1	Largeur minimale de la région M6	M6 = 0,44	M6 = 0,456
11	VIA1	Via entre M1 et M2	-	Nous comptons les vias à chaque nœud
12	VIA2	Via entre M2 et M3	-	Nous comptons les vias à chaque nœud
13	VIA3	Via entre M3 et M4	-	Nous comptons les vias du chaque nœud
14	VIA4	Via entre M4 et M5	-	Nous comptons les vias à chaque nœud
15	VIA5	Via entre M5 et M6	-	Nous comptons les vias à chaque nœud

Certaines observations intéressantes se dégagent du tableau ci-dessus:

- 1) Les règles de métal  $M_x$  ( $x=2, 3, 4, 5$ ) requièrent les mêmes valeurs pour les 4 couches.
- 2) Pour la barre assignée (à une valeur de 103,5%) il y a des règles qui ne donnent pas de violations. Ces règles ne sont pas prises en considération (voir aussi l'annexe X pour les autres règles qui ne sont pas impliquées dans la pondération).
- 3) Pour la largeur de trace, la valeur de test n'est pas critique puisque toutes les traces sont construites à la valeur exacte de "routing".
- 4) Il y a pas une règle correspondante au nombre des vias dans le logiciel de processus des dessins des masques, mais nous avons inséré les cinq couches de vias produits par cette famille technologique.
- 5) Comme hypothèse simplificatrice, pour enlever le besoin d'une autre pondération, nous avons présupposé que le nombre de vias de chaque type, assigné à une connexion, est le même, c.à.d. que s'il y a doublage de vias, il est effectué partout de manière identique.

Le polygone DRC le plus souvent produit est un trapèze comme dans notre exemple qui suit, mais il se peut que le polygone soit un triangle ou un rectangle dépendamment de la position



et la dimension des connexions face à face. Pour simplifier le calcul de la surface réelle des polygones DRC pour les règles concernant la séparation entre les connexions, nous approximations ces polygones par des rectangles. Nous présentons, comme exemple, la treizième erreur du fichier obtenu suite à notre méthodologie pour la règle no.4, i.e. la séparation entre les connexions de la couche M5 (voir le tableau 7.1).

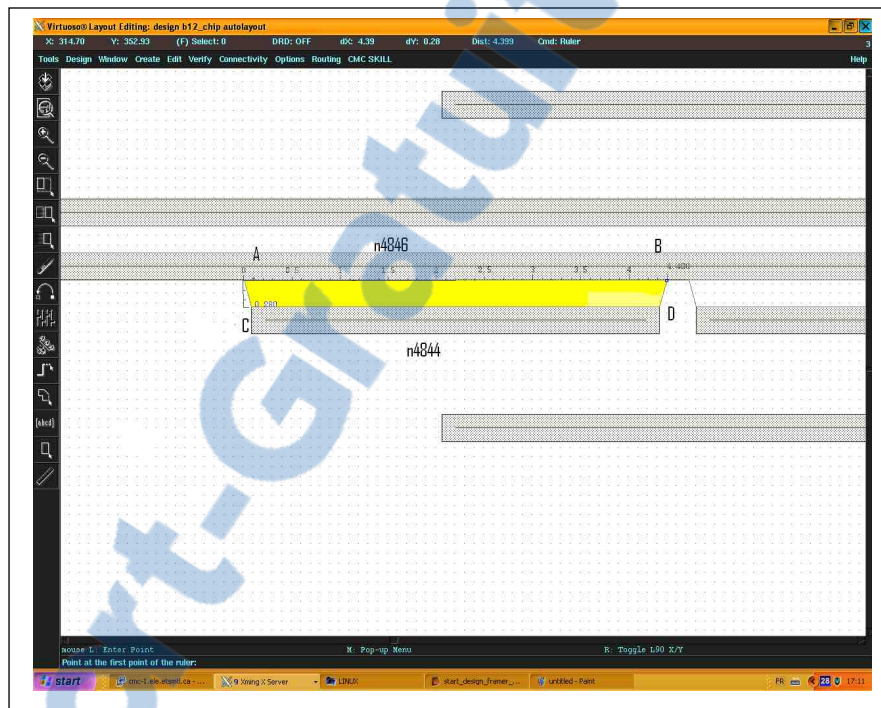


Figure 7-1 Approximation de la surface du polygone DRC (en jaune) par un rectangle.

Notons que nous présentons à l'annexe XII le fichier complet obtenu suite au DRC de Cadence d'où est tiré cet exemple. Aussi à l'annexe IX nous présentons un exemple du calcul de la surface réelle  $Ar_{pij}$  de ce polygone et la valeur de la surface du rectangle qui le remplace dans les calculs de la pondération. Dans le cas des règles où la largeur de la connexion est le paramètre recherché, le polygone DRC est effectivement un rectangle.

Il est à noter que pour toutes les couches sauf M1, un côté du rectangle, soit celui effectif ou celui remplacé est à dimension fixe et égale à la valeur initiale DRC. Rappelons que le DRC

a été effectué avec une valeur conservatrice. Cette observation nous aide à simplifier le calcul de l'équation 7.2 pour les termes de la même règle.

Dans le cas de M1, la largeur des connections varie entre 0,22 $\mu$ m et 0,24  $\mu$ m et la séparation varie entre 0,26  $\mu$ m et 0,27  $\mu$ m. Pour avoir un parcours similaire aux autres règles, nous avons adopté une valeur moyenne pour ce côté du rectangle sur  $M_1$ . Voici un sommaire des règles considérées pour notre étude :

Tableau 7.2 Règles considérées et résultats obtenus pour le circuit b12

#	Nom de la règle	Valeur de test	Nombre de polygones/vias DRC	Nombre de nœuds affectés	Nombre de nœuds distincts affectés
$R_1$	M1.S.1	0,228	1515	15 <sub>(Note 1)</sub>	6
$R_2$	M2.S.1	0,29	321	642	333
$R_3$	M3.S.1	0,29	1197	2394	491
$R_4$	M5.S.1	0,29	43	86	42
$R_5$	M1.W.1	0,248	16649	673 <sub>(Note 1)</sub>	328
$R_6$	M2.W.1	0,29	2915	2915	587
$R_7$	M3.W.1	0,29	1566	1566	536
$R_8$	M4.W.1	0,29	444	444	254
$R_9$	M5.W.1	0,29	84	84	58
$R_{10}$	M6.W.1	0,456	4	4	2
$R_{11}$	Via1	NA	2255	-	Note 2
$R_{12}$	Via2	NA	2277	-	Note 2
$R_{13}$	Via3	NA	819	-	Note 2
$R_{14}$	Via4	NA	158	-	Note 2
$R_{15}$	Via5	NA	3	-	Note 2

## Notes

- 1) La différence entre le nombre des polygones résultants du DRC et le nombre de nœuds sur la couche M1, par rapport aux autres couches, s'explique par la restriction qui nous est imposée de ne pas pouvoir visualiser l'intérieur (dessin des masques des cellules normalisées), pour des raisons de confidentialité. Lorsque nous utilisons les règles de dessin plus conservatrices, de nombreuses violations apparaissent, violations qui impliquent les couches non-visibles de l'intérieur des cellules. Comme nous ne pouvons pas accéder aux masques des cellules, nous ne pouvons pas identifier les nœuds internes impliqués. Les polygones DRC associés à ces violations avec les nœuds internes ont donc été retirés de l'analyse. On retrouve 1500 de ces polygones pour la règle 1 (séparation sur M1) et 15976 pour la règle 5 (largeur sur M1). Nous considérons comme travail futur l'extension de notre analyse pour les nœuds internes des cellules.
- 2) Le traitement pour trouver l'affectation des vias aux nœuds a été effectué pour l'ensemble des vias et non par chacune des règles/couches séparément.

### 7.3 Traitement spécifique pour l'obtention des 3 variantes du quatrième type de signatures

Dans cette section, nous présentons la procédure pour obtenir les signatures avec la pondération dépendante des règles de conception de dessin des masques. Comme mentionné précédemment, puisque nous utilisons Fastscan de 3 manières différentes, nous proposons trois variantes pour le quatrième type de signature.

#### 7.3.1 Signature par la commande d'analyse des pannes (analyse faults) – méthode 4a

Dans cette sous-section, nous gardons le même traitement des résultats de Fastscan que pour les méthodes précédentes. Puisque la procédure d'obtention de cette signature est similaire à celle de la première méthode, nous ne présentons ici que les différences par rapport à cette première méthode. La figure 7.2 montre le déroulement du traitement spécifique pour obtenir la signature proposée (ce traitement est présenté de manière détaillé à l'annexe X).

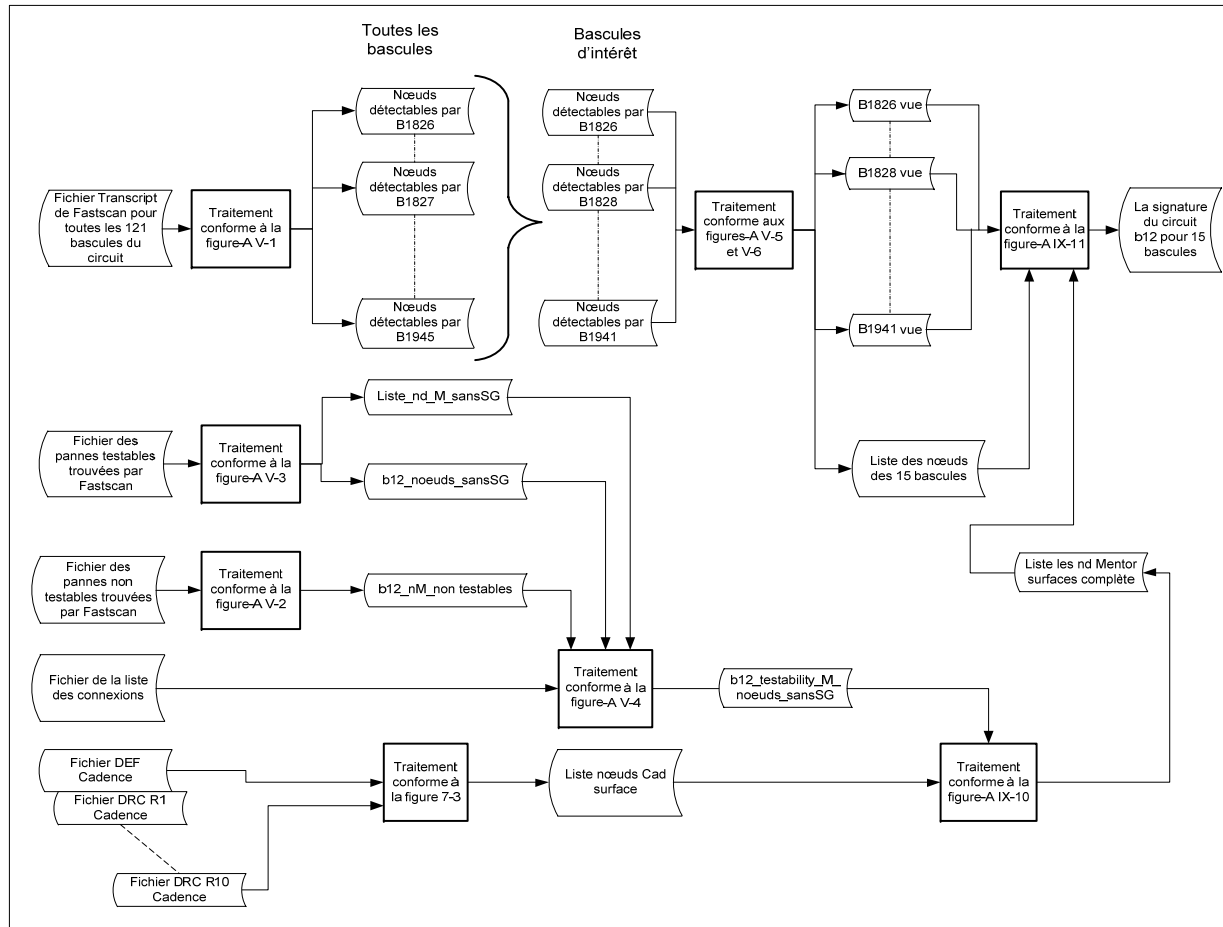


Figure 7-2 Traitement pour l'obtention de la signature pondérée par les polygones DRC, première variante (Méthode 4a).

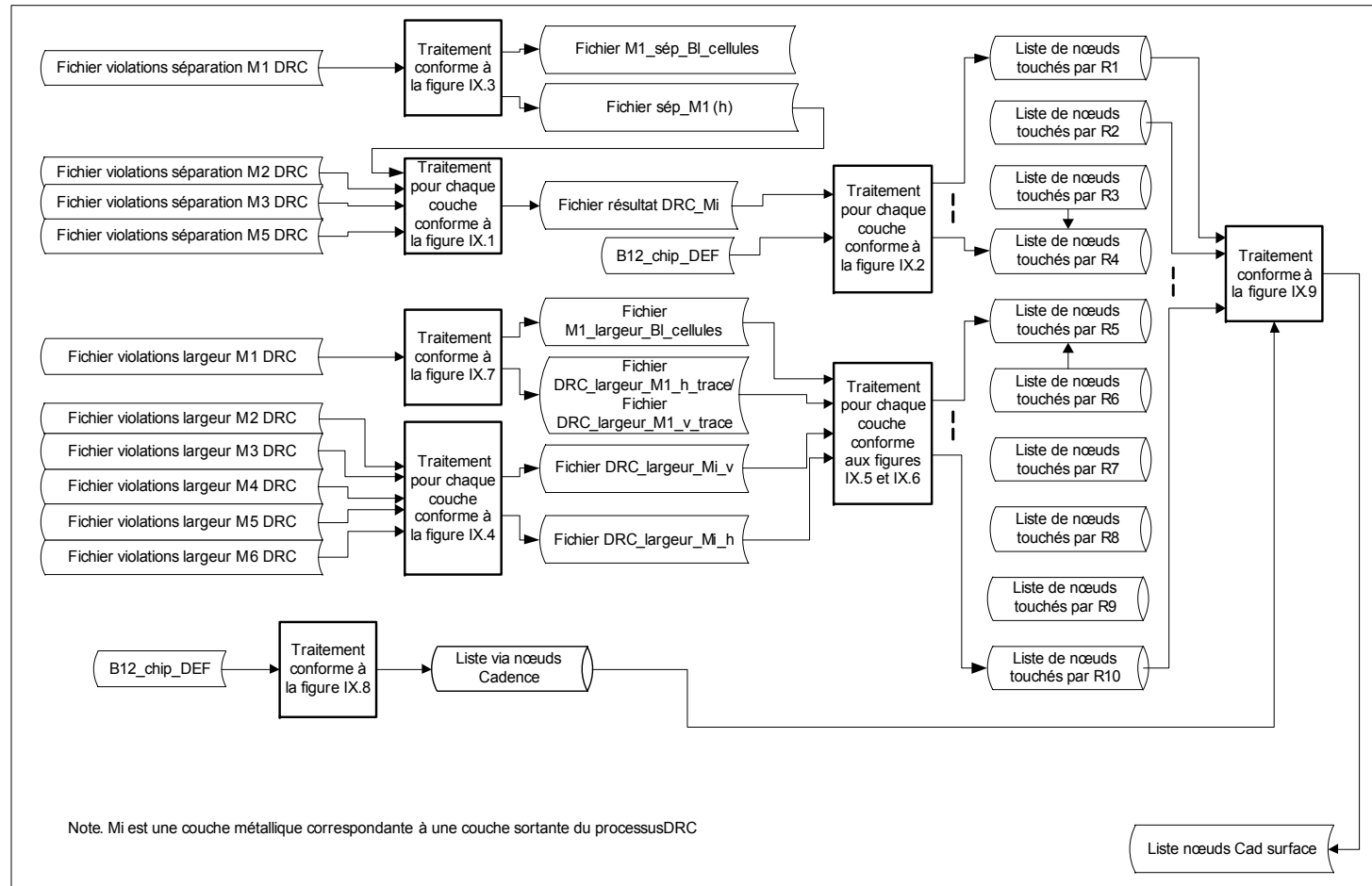


Figure 7-3 Assignment des surfaces efficaces aux nœuds à partir de dessin des masques du circuit

Le traitement d'assignation de la surface efficace (des polygones détectés) à chaque nœud Cadence est résumé à la figure 7.3 (et détaillé à l'annexe X). Le résultat de la procédure de la figure 7.3 est la liste complète des nœuds de Cadence, avec le filtrage des signaux globaux, et pour chacun des nœuds, tous les polygones associés à chaque règle de conception.

Suite au traitement présenté de manière plus détaillée à la figure-A X.10 de l'annexe X, nous créons une autre base de données adéquate pour la signature requise, le fichier *Liste des nd Mentor surfaces complète*.

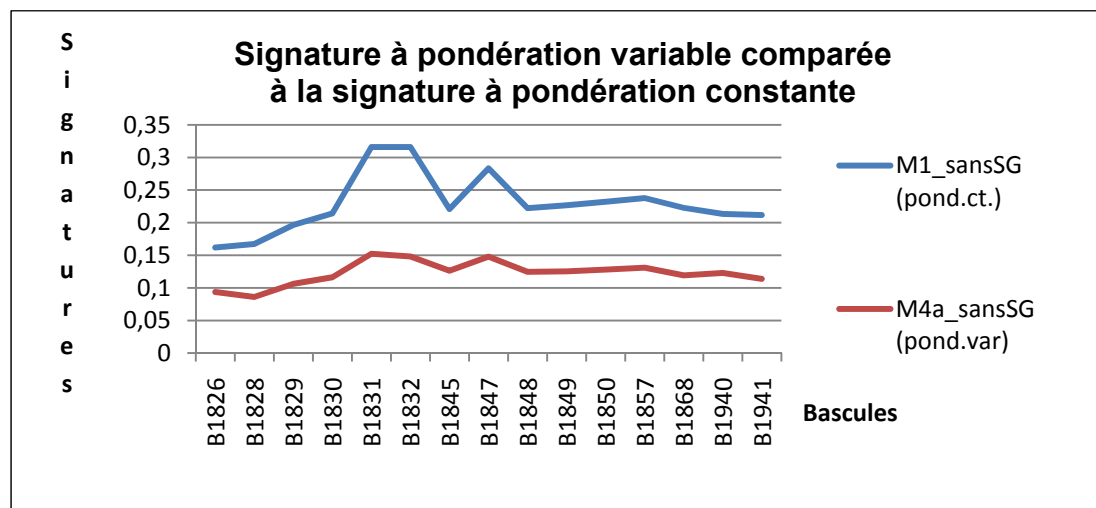


Figure 7-4 Signatures à pondération variable produites par "analyse des pannes" pour un DRC plus conservateur, première variante (Méthode 4a).

Ce fichier contient la liste des nœuds Mentor et les valeurs requises pour le calcul de la signature. Les autres traitements sont identiques à ceux présentés au chapitre 4, sauf le traitement final qui produit les signatures. Ce traitement est présenté à la figure-A X.11 de l'annexe X. Le résultat de cette procédure est montré à la figure 7.4. Les 15 bascules détectent 1135 nœuds distincts avec la commande "analyse des pannes" (analyze faults), nœuds qui représentent un rapport de  $1135/2244 = 51\%$  de la surface du circuit b12. Nous avons établi que 2244 est le nombre total des nœuds des signaux après l'exclusion des nœuds

appartenant aux signaux globaux (voir chapitre 2). Nous observons que la nouvelle information a modifié l'allure globale de la signature.

### 7.3.1.1 Usage pratique de la méthode 4a (première variante)

La figure 7.5a nous rappelle le flot pour l'obtention des signatures des bascules, considérant la matrice des 15 valeurs  $k_p$  connues. C'est une première utilisation de la méthode rencontrée dans le cas d'une technologie bien connue et déterminée où l'introduction des nouvelles puces est basée sur l'expérience antérieure.

À l'apparition d'une toute nouvelle puce, fabriquée dans une technologie non mure, donc où la matrice  $k$  est inconnue, il est possible d'établir un système d'équation liant la signature réelle (celle obtenue à partir des résultats des tests) et la signature calculée. Cela nous permet d'estimer le facteur de pondération  $k_p$  pour chaque règle  $p$  considérée à partir des équations de cette méthodologie. Une fois les bascules intéressantes connues, nous pouvons écrire les équations de la méthodologie pour ces bascules.

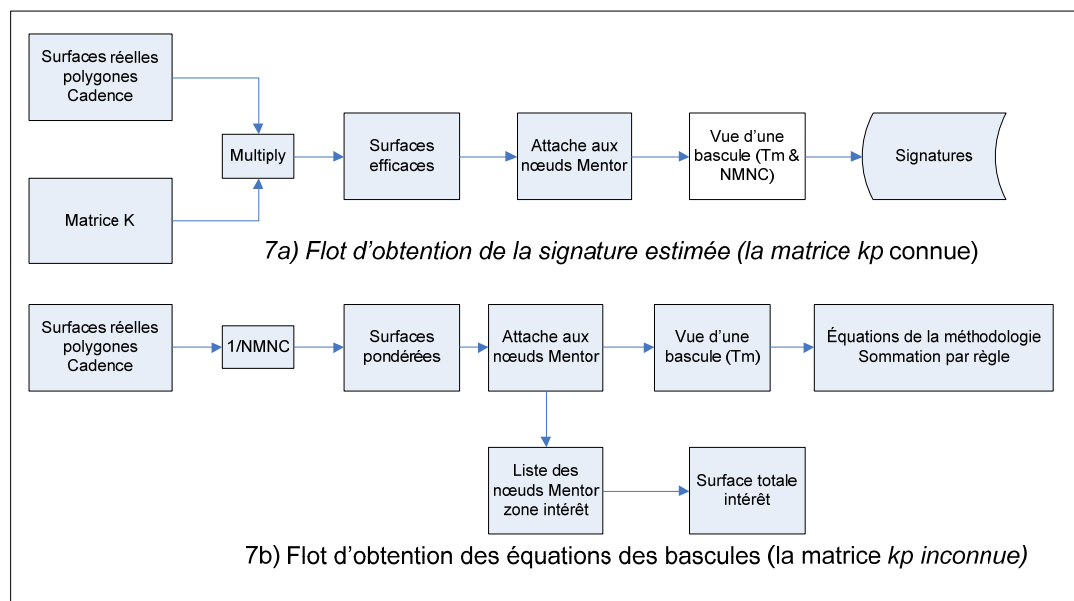


Figure 7-5 Usage de la méthode 4a pour l'obtention des signatures et des équations.

Puisqu'il s'agira d'un système de 15 équations à 15 inconnues, les 15 valeurs de  $k_p$  sont déterminées. Les équations nous permettent aussi de trouver les bascules les plus "sensibles" dans la vérification de la puce par la valeur maximale du produit du nombre des nœuds couverts et des surfaces efficaces.

### **7.3.1.2 Sommaire pour la méthode 4a (première variante)**

Nous avons présenté dans cette sous-section la première variante du quatrième type de signature, qui ajoute un poids au nœud relié à la surface des polygones DRC produits autour de lui par les dépassements des règles de conception du dessin des masques. Cette pondération met en évidence la dépendance de la signature à la conception de la puce et sa fabrication. Par la sensibilité des règles et la surface des polygones rattachés aux nœuds, la signature peut offrir des jalons plus précis pour la localisation des défauts et leur source, mais au prix d'une augmentation du temps du traitement par rapport aux signatures développées auparavant. Cette augmentation est fortement liée à la complexité du circuit et bien sûr à la technologie utilisée. Pour une signature pondérée pour un circuit très simple comme le b02, en technologie CMOSP18, c.à.d. 6 couches métalliques et un seul poly, le calcul de la pondération pour toutes les règles, a demandé au total 400 ms de temps CPU. Pour la signature du circuit b12, nous avons comptabilisé un temps CPU total de 32,7 secondes, ce qui exclut la vérification des résultats intermédiaires. À noter que ce temps n'inclut pas la génération du dessin des masques ou celles des vecteurs de test ATPG, qui font partie intégrante de l'approche traditionnelle de conception.

### **7.3.2 Signature par la commande « ajout de listes (add lists) » - méthode 4b (deuxième variante)**

Nous présentons ici la deuxième variante du quatrième type de signature, qui utilise la même pondération des nœuds de Mentor quant à leur contribution à la signature, basée sur les polygones DRC. Cependant, la manière d'utiliser l'outil Fastscan est différente. Puisque la procédure d'obtention de cette signature est similaire à celle de la méthode 4a, nous ne



présentons ici que les différences par rapport à cette méthode. La figure 7-6 montre le déroulement du traitement spécifique pour obtenir la signature proposée. Les résultats de Fastscan sont traités afin d'obtenir une forme adéquate à nos traitements ultérieurs, conformément aux diagrammes des figures-A XIII.1, XIII.2, XIII.3 de l'annexe XIII. Le fichier obtenu, *FS\_equiv*, est présenté dans l'annexe XIV. Il fournit (via le traitement décrit à la figure-A XIII.4 de l'annexe XIII) les nœuds couverts par les bascules qui nous intéressent, à savoir les mêmes 15 bascules qu'auparavant. Un autre traitement appliqué au fichier *FS\_equiv*, conforme à la figure-A XIII.5, calcule les nouveaux coefficients de testabilité obtenus suite à la commande ``ajout des listes`` (décrite plus loin) et crée une autre base de données adéquate pour la signature requise, le fichier *Liste nd M 15 bascules*. Ce traitement se sert du fichier *Liste des nd Mentor surfaces modifiée* qui reprend tous les renseignements du fichier produit au type précédente de signature, i.e. *Liste des nd Mentor surfaces complète*, et enlève les coefficients de testabilité calculés par l'analyse des pannes. Par un traitement décrit à la figure-A XIII.6 nous obtenons les signatures désirées. Le résultat de cette procédure est montré à la figure 7.7. Nous observons que la nouvelle commande de Fastscan a modifié l'allure globale de la signature, en amplifiant davantage le relief de celle-ci.

### 7.3.2.1 Particularités de l'utilisation du Fastscan dans la signature proposée

La commande principale pour cette deuxième variante de signature est "ajout des listes" (add listes). Cette commande offre une information plus précise quant à l'impact des pannes sur le contenu des bascules. En effet, cette commande, qui roule dans le mode de simulation PANNE (FAULT), permet d'afficher les valeurs logiques avec et sans panne.

Pour obtenir la signature envisagée, il faut passer par plusieurs étapes avec Fastscan. Premièrement nous mettons l'outil en mode PANNE (FAULT) pour obtenir les vecteurs de test. Notons que ces vecteurs de test ont déjà été obtenus dans le cadre de simulation faite dans le chapitre 3.1. Il est important d'appliquer les mêmes vecteurs dans le même ordre afin de pouvoir comparer les méthodes/signatures entres elles. L'étape suivante consiste à ajouter

la liste des nœuds à surveiller, à savoir la sortie des bascules d'intérêt. Cependant, comme l'outil ne peut pas rapporter directement la valeur de sortie des bascules, nous devons plutôt lui fournir les nœuds de sortie des portes logiques dont la sortie est branchée à l'entrée de ces bascules. Ces nœuds sont contenus dans le fichier *sortiesportes\_15\_bascules*. Finalement, la simulation de pannes est ensuite lancée et les résultats sont acheminés dans un fichier de capture sous la forme décrite à l'annexe XIV, fichier qui subira divers traitements, conformément aux figures-A XIV.1, XIV.2, XIV-3 de l'annexe XIV. Ces traitements vont permettre de rendre l'information pertinente accessible à un traitement de texte.

#### **7.3.2.2 Sommaire pour la méthode 4b (deuxième variante)**

Nous avons présenté dans cette sous-section la deuxième variante du quatrième type de signature, qui ajoute un poids au nœud relié à la surface des polygones DRC produisent autour de lui par les dépassements des règles de conception du dessin des masques.

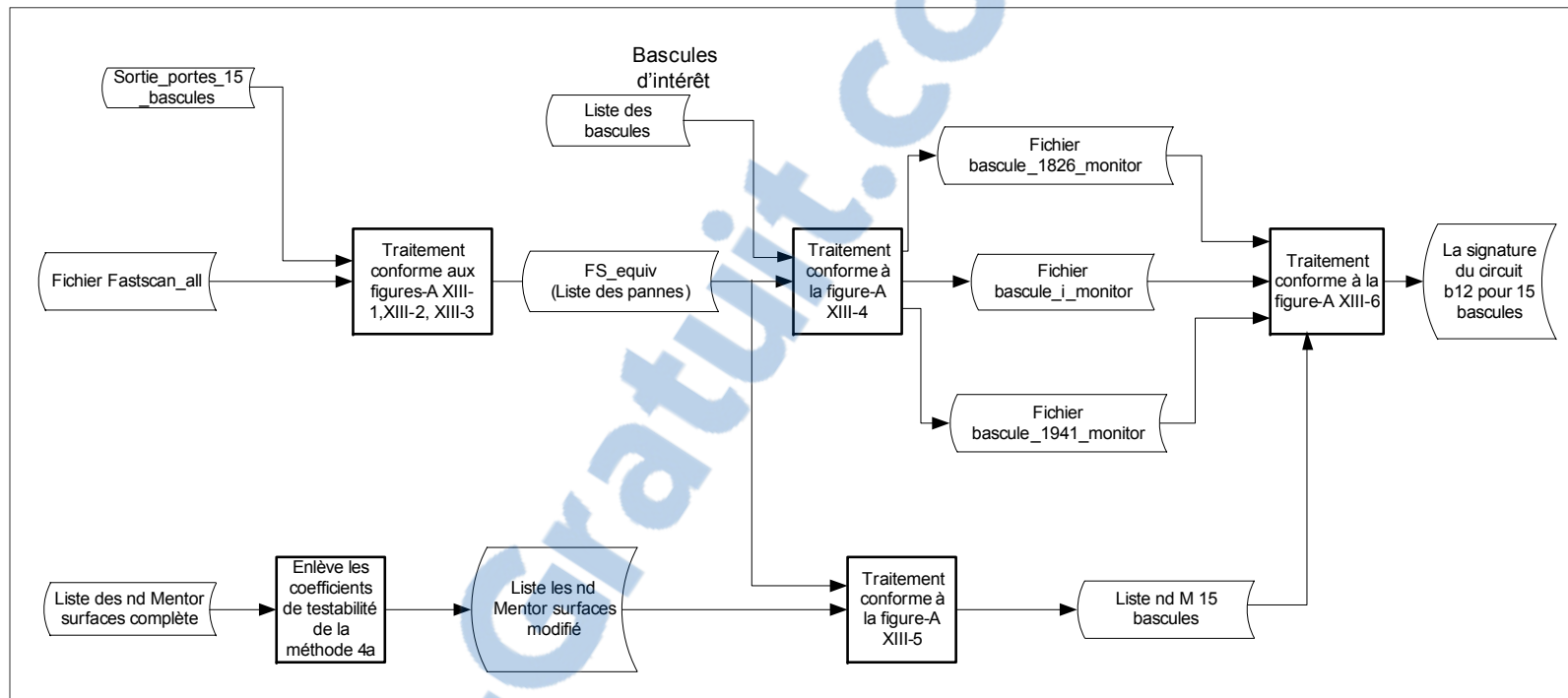


Figure 7-6 Traitement pour l'obtention de la signature pondérée par les polygones DRC (Méthode 4b)

Par rapport à la première variante, la différence majeure est l'utilisation de Fastscan. La variante proposée utilise Fastscan de même façon qu'aux chapitres 4,5 et 6, pendant que la deuxième variante est plus précise quant aux effets des pannes. L'effort demandé est semblable à celui mis en œuvre pour la première variante (méthode 4a). Le système d'équations développé pour la méthode 4a est aussi applicable pour la méthode 4b.

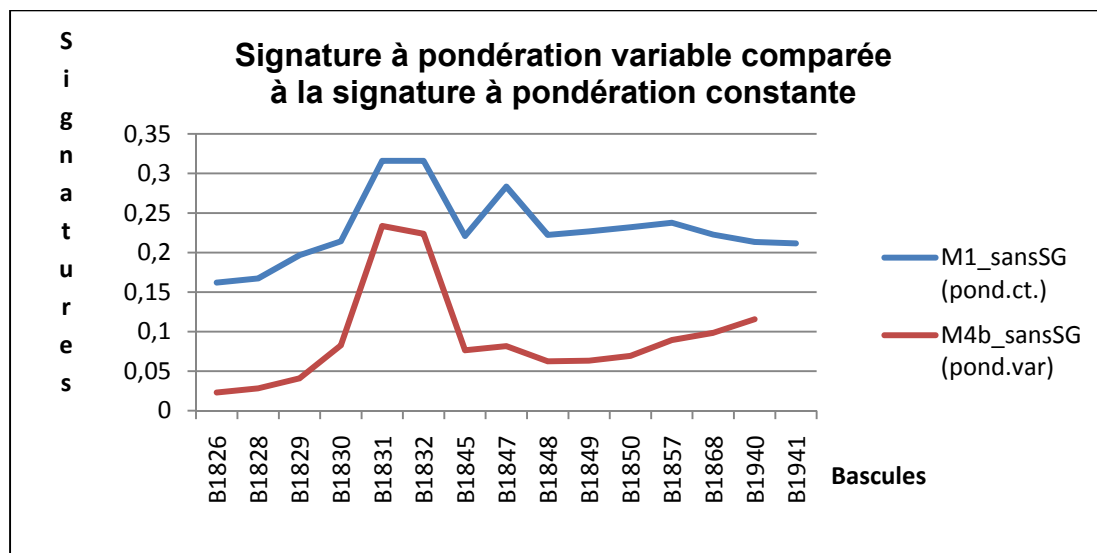


Figure 7-7 Signatures à pondération variable produites par "ajout de listes" pour un DRC plus conservateur, deuxième variante (Méthode 4b).

### 7.3.3 Traitement spécifique pour l'obtention de la signature pour le test rapide, arrêt à la première défectuosité détectée - méthode 4c (troisième variante).

Nous décrivons ici la dernière des trois variantes, qui est similaire à la variante précédente. La seule différence est qu'ici nous supposons que l'application des vecteurs est stoppée à l'apparition de la première panne, contrairement au type précédent où l'application des vecteurs était complète. L'application des vecteurs stoppée à l'apparition de la première panne est représentative d'un test en production, alors que celle qui est faite de manière complète est représentative d'un test à des fins de diagnostic. La dernière variante de signatures présentée ici utilise donc toujours la même pondération des nœuds de Mentor dans leur contribution à la signature, basée sur les polygones DRC ainsi que la même

commande («add list») de l'outil Fastscan. Ce qui diffère est le traitement menant à la création des nœuds couverts par chaque bascule, contenus dans le fichier "monitor". Puisque la procédure d'obtention de cette signature est similaire à celle de la méthode 4b, nous ne présentons dans cette section que les différences par rapport à cette méthode. La figure 7-8 montre le déroulement du traitement spécifique pour obtenir la signature proposée. Nous traitons les résultats du Fastscan pour connaître la couverture de chaque bascule lorsque le test s'arrête à la première panne, conformément aux diagrammes des figures-A XV.1, XV.2 de l'annexe XV. Le fichier obtenu pour chaque bascule, *Fichier\_premiere\_panne*, est traité, conformément à la figure-A XV.3 de l'annexe XV, pour que soient liés à chaque nœud les paramètres requis pour le calcul de la signature, paramètres qui sont inclus dans le fichier *bascule\_i\_monitor1*. Par un traitement décrit à la figure-A XV.4 nous obtenons la signature proposée de cette méthode. Le résultat de cette procédure est montré à la figure 7.9. L'allure de cette signature est somme toute assez similaire à la précédente. Une comparaison entre toutes les signatures est faite plus loin.

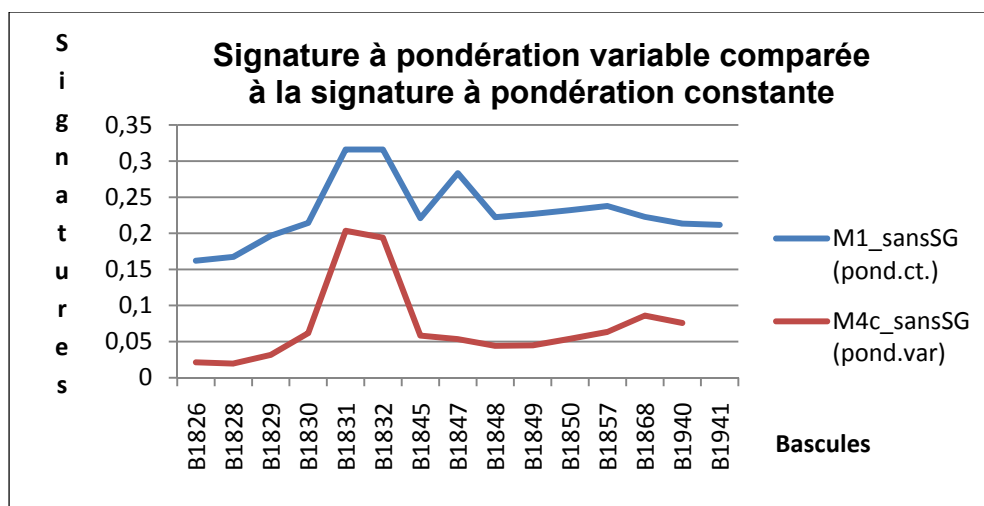


Figure 7-8 Signatures à pondération variable produites "à l'apparition de la première panne" pour un DRC plus conservateur, troisième variante (Méthode 4c).



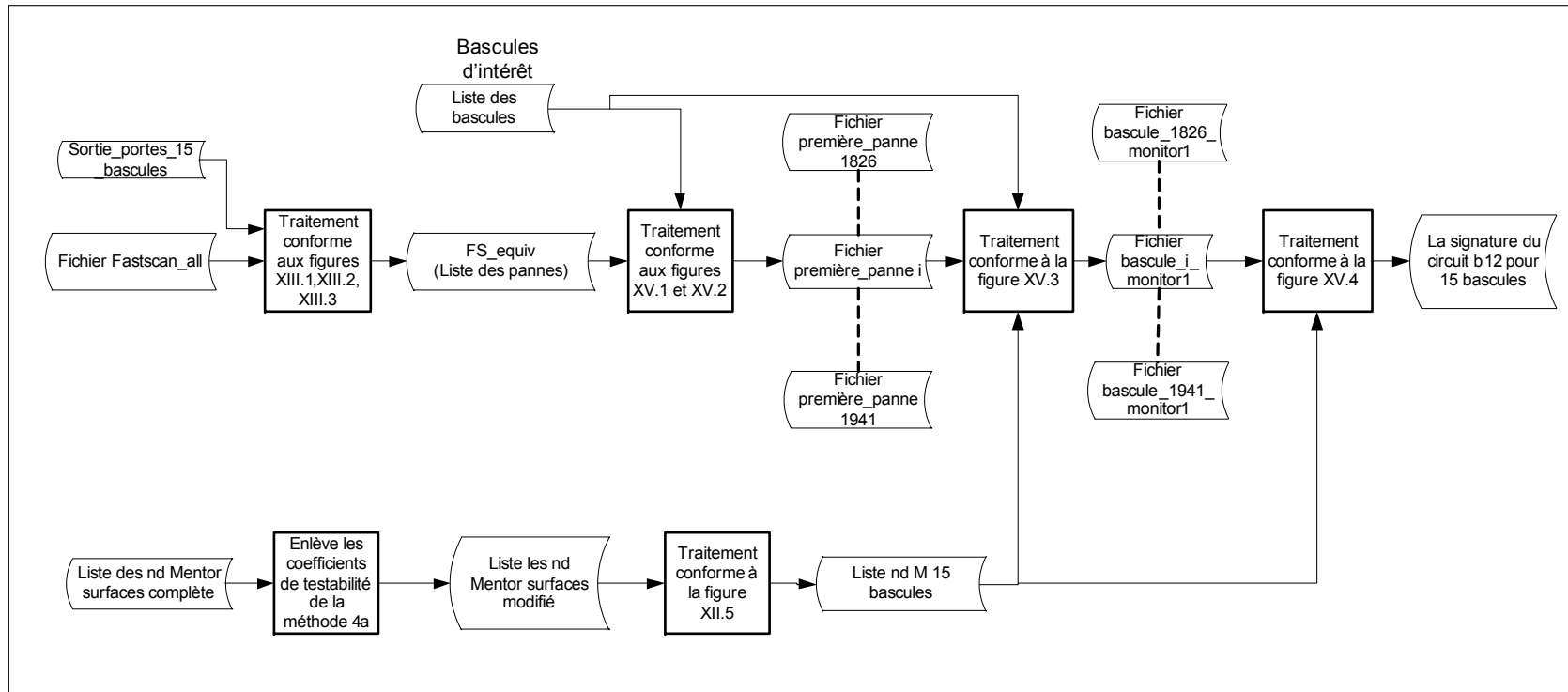


Figure 7-9 Traitement pour l'obtention de la signature pondérée par les polygones DRC (Méthode 4c)

### 7.3.3.1 Sommaire pour la méthode 4c (troisième variante)

Nous avons présenté la dernière des trois variantes du quatrième type de signature, qui ajoute un poids au nœud relié à la surface des polygones DRC produisant autour de lui par les dépassements des règles de conception du dessin des masques. Par rapport à la deuxième variante, la différence principale est le fait de supposer que le test arrête à la première panne détectée. L'effort demandé au traitement est 3 secondes de plus pour le temps CPU que la méthode 4b. Le système d'équations développé pour la méthode 4a est aussi applicable pour la méthode 4c.

## 7.4 Comparaisons entre toutes les signatures

Le graphique de la figure 7.10 présente les 7 signatures développées. Si nous jetons un regard sur les traitements effectués pour obtenir toutes les signatures, nous pouvons constater que la première signature obtenue (signature à pondération constante) demande le moins d'efforts par rapport aux autres. Comme cette méthode ne prend en considération que les bascules et les nœuds couverts par celles-ci, ces signatures en cas de défauts donnent des informations assez évasives sur la localisation et la source des pannes. La deuxième méthode assigne un poids aux nœuds relié à la capacité parasite du nœud. La capacité parasite est fortement liée à la fabrication et à la technologie utilisée pour la fabrication de la puce mais aussi ces signatures en cas de défauts donnent des informations évasives sur la localisation et la source des pannes. L'avantage de ces deux méthodes est la simplicité du traitement qui nécessite juste l'utilisation directe des outils Fastscan (Mentor Graphics) et Encounter (Cadence). La troisième méthode, où la pondération des nœuds est dépendante aux couches métalliques touchant le nœud, demande en plus un traitement spécifique outre l'utilisation des deux outils déjà mentionnés. Par la liaison faite aux opérations de fabrication pour exécuter les couches métalliques, cette méthodologie peut donner un peu plus de détails

sur la localisation des pannes et leur source. Les deuxième et troisième méthodes (à pondération variable) donnent des signatures étroitement rapprochées à la signature à la pondération constante (ou la signature de référence) comme on peut voir dans les graphes présentés à la figure 7.10. Nous constatons que le graphe de la deuxième méthode est en dessous et celui de la troisième méthode est au-dessus du graphe de référence mais leur allure est similaire. Cela s'explique par le fait que les nœuds choisis ont une capacité plus réduite que la capacité moyenne des nœuds du circuit mais ils touchent plus de couches métalliques que le nombre moyen de couches par nœud. Cela nous amène à la conclusion que notre zone d'intérêt est constituée des nœuds répandus sur un nombre relativement grand des couches et en même temps sur des connexions relatives courtes et espacés, ce qui s'explique en partie par le retrait des signaux globaux. Les deux conclusions sont convergentes. À noter que pour ces signatures la forme presque identique nous indique que la contribution de la couverture des bascules est plus importante que la pondération des nœuds.

Les graphes obtenus par la pondération des nœuds dépendante des règles de conception de dessin des masques se trouvent en-dessous des 3 premières courbes antérieurement présentées, c.à.d. la valeur de pondération assignée aux nœuds est plus réduite mais en même temps l'apport de la pondération est plus importante parce que la forme de graphe est différente. La différence d'allure entre la courbe de la méthode 4a et celles des 3 premières méthodes est due non seulement aux valeurs de pondération différentes mais à une différence de la loi de variation de cette pondération.

En sachant que la pondération des nœuds est identique pour tous les types de la méthode 4, la différence entre les trois courbes s'explique amplement par la différence de couverture de chaque bascule due à l'utilisation de Fastscan. Par le fait que les signatures de la méthode 4b et 4c sont très semblables, l'effort supplémentaire demandé pour le développement de la méthode 4c ne justifie son utilisation que dans des cas spéciaux pour des tests partiels.



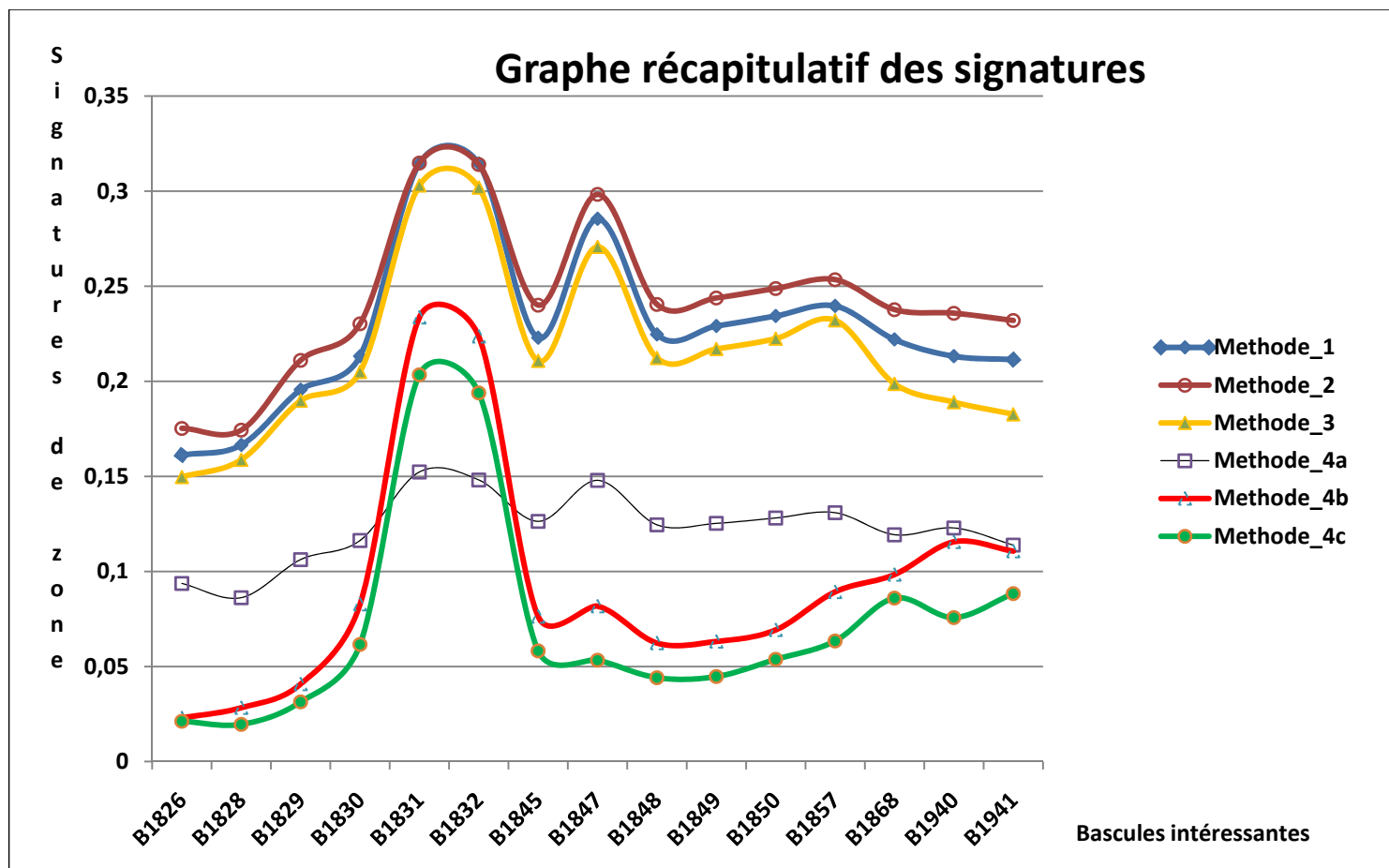


Figure 7-10 Graphe récapitulatif des signatures.  
 Note. Les valeurs numériques des courbes sont présentées dans l'annexe XVI.

## CONCLUSIONS

Ce mémoire présente une analyse de quatre méthodes permettant d'obtenir des signatures composites pouvant servir de point de référence. La première méthode, qui suppose que les pannes sont équiprobables, est une méthode simple et rapide utilisant les outils de génération automatisée de vecteurs de test (ATPG). En partant de la liste de connections Verilog du circuit, cette méthode utilise l'analyse logique au niveau de la puce, sans utiliser les ressources du dessin des masques. La seconde méthode est un peu plus élaborée en ajoutant aux données logiques des informations extraites du dessin des masques, à savoir les capacités parasites des nœuds. La troisième méthode pondère la sensibilité des nœuds en fonction des couches métalliques impliquées. Ces trois premières méthodes sont relativement simples. Cette simplicité est obtenue au prix d'une résolution plutôt faible de la méthode: chacune des trois méthodes peut être utilisée pour confirmer ou infirmer leur hypothèse de base (respectivement : pannes équiprobables, corrélation entre la valeur de la capacité parasite et la sensibilité d'un nœud, et sensibilité des nœuds en fonction des couches métalliques impliquées), en ne donnant toutefois que peu d'indices sur la nature et la localisation d'éventuelles pannes qui seraient plus probables voire systématiques. De plus, les résultats obtenus montrent que les trois premières donnent des signatures similaires, et que l'effort nécessaire pour ajouter l'information requise pour la seconde et la troisième est difficilement justifiable, conséquence probable du rejet des nœuds globaux.

La quatrième et dernière méthode, avec laquelle sont générées trois signatures différentes (en fonction de l'utilisation de l'outil ATPG), utilise la pondération reliée aux polygones générés par l'outil de vérification des règles du dessin des masques (DRC). C'est la méthode la plus élaborée mais qui fournit le plus d'informations sur la localisation et la source probables des pannes. En effet, en liant directement les résultats fournis par cette méthode à des règles DRC particulières, ceci donne des indices quant au type de structures en cause, si le rendement lié à l'une de ces règles est anormalement bas. La première variante de

signature issue de la quatrième méthode (méthode 4a) utilise l'outil ATPG de la même manière que les 3 premières méthodes, mais conduit à une signature à l'allure fort différente. Ceci est une indication claire de la différenciation amenée par la pondération basée sur les règles du dessin des masques. L'application d'une autre stratégie au niveau de l'outil ATPG, présente dans la seconde et troisième variantes, amène également un élément de différenciation intéressant. Dans les 2 cas de différenciation, nous considérons que le changement de l'allure globale des signatures justifie l'augmentation de la complexité causée par l'utilisation d'informations plus précises. Par contre, les résultats indiquent que l'ajout de la condition d'arrêt à la première panne (troisième variante, méthode 4c) n'est pas justifiable. Bien que le principal circuit de référence utilisé, le circuit b12, ne soit pas un circuit très complexe, nous considérons que les tendances cette étude de cas devraient également être observées avec des circuits plus complexes. Les diverses signatures développées dans le cadre de ce projet, en particulier celle de la méthode 4b, devraient s'avérer des instruments utiles pour la détection des pannes systématiques. La validité de chacune de ces méthodes reste à être vérifiée dans un environnement de fabrication de grand volume doté de moyens spécifiques de capture et d'analyse des données produites par le test en production.

## **RECOMMANDATIONS**

Pour les travaux futurs, nous proposons quelques pistes à suivre dans le processus d'obtention des signatures. Une première piste est celle d'une mise en œuvre du concept avec des puces plus complexes et fabriquées en gros volume. Cela implique l'application d'une technologie existante en ligne de production, d'un circuit en cours de fabrication et d'un nombre plus élevé de règles de dessin des masques à considérer. Une seconde piste est celle de l'utilisation de la signature de la méthode 4b en complémentarité avec une basée sur les cellules erronées. Une troisième piste est l'utilisation de diagnostic après l'application des signatures. Finalement, une optimisation des traitements présentés est aussi une piste de développement intéressante. Cette optimisation devrait se traduire par une augmentation de la vitesse d'exécution.

## ANNEXE I

### CODE VHDL DU CIRCUIT B12

entity b12 is

```
port(nl      : out bit_vector ( 3 downto 0);
     speaker  : out bit;
     nloss    : out bit;
     k        : in  bit_vector ( 3 downto 0);
     clock    : in  bit;
     reset    : in  bit;
     start    : in  bit);
```

end b12;

architecture BEHAV of b12 is

```
constant RED : integer := 0;
constant GREEN : integer := 1;
constant YELLOW : integer := 2;
constant BLUE : integer := 3;
constant LED_ON  : bit := '1';
constant LED_OFF : bit := '0';
constant PLAY_ON  : bit := '1';
constant PLAY_OFF : bit := '0';
constant KEY_ON   : bit := '1';
constant NUM_KEY  : natural := 4;
constant COD_COLOR : natural := 2;
constant COD_SOUND : natural := 3;
constant S_WIN    : natural := 2**COD_COLOR;
constant S_LOSS   : natural := S_WIN + 1;
constant SIZE_ADDRESS : natural := 5;
constant SIZE_MEM   : natural := 2**SIZE_ADDRESS;
constant COUNT_KEY  : natural := 33;
constant COUNT_SEQ  : natural := 33;
constant DEC_SEQ    : natural := 1;
constant COUNT_FIN  : natural := 8;
constant ERROR_TONE : natural := 1;
```

```

constant RED_TONE   : natural := 2;
constant GREEN_TONE : natural := 3;
constant YELLOW_TONE : natural := 4;
constant BLUE_TONE  : natural := 5;
constant WIN_TONE   : natural := 6;

signal wr          : bit;
signal address     : natural range SIZE_MEM - 1 downto 0;
signal data_in     : natural range 2**COD_COLOR - 1 downto 0;
signal data_out    : natural range 2**COD_COLOR - 1 downto 0;
signal num         : natural range 2**COD_COLOR - 1 downto 0;
signal sound       : natural range 2**COD_SOUND - 1 downto 0;
signal play        : bit;

begin
  process (clock,reset)
    variable s: bit ;
    variable counter : natural range 7 downto 0;
  begin
    if (reset='1') then
      s := '0';
      speaker <= '0';
      counter :=0;
    elsif clock'event and clock = '1' then
      if play = '1' then
        case sound is
          when 0 =>
            if counter > RED_TONE then
              s := not s;
              speaker <= s;
              counter:=0;
            else
              counter:= counter + 1;
            end if;
          when 1 =>
            if counter > GREEN_TONE then
              s := not s;

```

```

    speaker <= s;
    counter:=0;
else
    counter:= counter + 1;
end if;
when 2 =>
    if counter > YELLOW_TONE then
        s := not s;
        speaker <= s;
        counter:=0;
    else
        counter:= counter + 1;
    end if;
when 3 =>
    if counter > BLUE_TONE then
        s := not s;
        speaker <= s;
        counter:=0;
    else
        counter:= counter + 1;
    end if;
when S_WIN =>
    if counter > WIN_TONE then
        s := not s;
        speaker <= s;
        counter:=0;
    else
        counter:= counter + 1;
    end if;
when S_LOSS =>
    if counter > ERROR_TONE then
        s := not s;
        speaker <= s;
        counter:=0;
    else

```

```

        counter:= counter + 1;
    end if;
    when others =>
        counter := 0;
    end case;
else
    counter :=0;
    speaker <= '0';
end if;
end if;
end process;
process (clock,reset)
    variable count : natural range 2**COD_COLOR - 1 downto 0;
begin
    if (reset='1') then
        count := 0;
        num <= 0;
    elsif clock'event and clock = '1' then
        count := (count + 1) mod (2**COD_COLOR);
        -- count := count + 1;
        -- removed(!)fs030699
        num <= count;
    end if;
end process;
process(clock,reset)
    type RAM is array (natural range SIZE_MEM - 1 downto 0) of natural range 2**COD_COLOR - 1 downto
0;
    variable memory : RAM ;
    variable mar : natural range SIZE_MEM - 1 downto 0;
begin
    if reset = '1' then
        data_out <= 0;
        for mar in 0 to SIZE_MEM - 1 loop
            memory(mar) := 0;
        end loop;

```



```

elsif clock'event and clock='1' then
    data_out <= memory(address);
    if wr='1' then
        memory(address) := data_in;
    end if;
end if;
end process;
process (clock,reset)
    constant G0: integer :=0;
    constant G1: integer :=1;
    constant G2: integer :=2;
    constant G3: integer :=3;
    constant G4: integer :=4;
    constant G5: integer :=5;
    constant G6: integer :=6;
    constant G7: integer :=7;
    constant G8: integer :=8;
    constant G9: integer :=9;
    constant G10: integer :=10;
    constant G10a: integer :=11;
    constant G11: integer :=12;
    constant G12: integer :=13;
    constant Ea: integer :=14;
    constant E0: integer :=15;
    constant E1: integer :=16;
    constant K0: integer :=17;
    constant K1: integer :=18;
    constant K2: integer :=19;
    constant K3: integer :=20;
    constant K4: integer :=21;
    constant K5: integer :=22;
    constant K6: integer :=23;
    constant W0: integer :=24;
    constant W1: integer :=25;
    variable gamma : integer range 25 downto 0;

```

```

variable ind    : natural range 2**COD_COLOR - 1 downto 0;
variable scan   : natural range SIZE_MEM - 1 downto 0;
variable max    : natural range SIZE_MEM - 1 downto 0;
variable timebase : natural range 63 downto 0;
variable count  : natural range 63 downto 0;
begin
  if (reset='1') then
    nloss <= LED_OFF;
    nl <= (others => LED_OFF);
    play <= PLAY_OFF;
    wr <= '0';
    scan := 0 ;
    max := 0 ;
    ind := 0 ;
    timebase := 0 ;
    count := 0 ;
    sound <=0 ;
    address <=0 ;
    data_in <=0 ;
    gamma := G0;
  elsif clock'event and clock = '1' then
    if start = '1' then
      gamma := G1;
    end if;
    case gamma is
      when G0 => gamma := G0;
      when G1 =>
        nloss <= LED_OFF;
        nl <= (others => LED_OFF);
        play <= PLAY_OFF;
        wr <= '0';
        max := 0;
        timebase := COUNT_SEQ;
        gamma := G2;
      when G2 =>

```

```

scan := 0;
wr <= '1';
address <= max;
data_in <= num;
gamma := G3;
when G3 =>
  wr <= '0';
  address <= scan;
  gamma := G4;
when G4 =>
  gamma := G5;
when G5 =>
  nl(data_out) <= LED_ON;
  count := timebase;
  play <= PLAY_ON;
  sound <= data_out;
  gamma := G6;
when G6 =>
  if count = 0 then
    nl <= (others => LED_OFF);
    play <= PLAY_OFF;
    count := timebase;
    gamma := G7;
  else
    count := count - 1;
    gamma := G6;
  end if;
when G7 =>
  if count = 0 then
    if scan /= max then
      scan := scan + 1;
      gamma := G3;
    else
      scan := 0;
      gamma := G8;
    end if;
  end if;

```

```

    end if;
else
    count := count - 1;
    gamma := G7;
end if;
when G8 =>
    count := COUNT_KEY;
    address <= scan;
    gamma := G9;
when G9 =>
    gamma := G10;
when G10 =>
    if count = 0 then
        nloss <= LED_ON;
        max := 0;
        gamma := K0;
    else
        count := count - 1;
        if k(0)=KEY_ON then
            ind := 0;
            sound <= 0;
            play <= PLAY_ON;
            count := timebase;
            if (data_out = 0) then
                gamma := G10a;
            else
                nloss <= LED_ON;
                gamma := Ea;
            end if;
        elsif k(1)=KEY_ON then
            ind := 1;
            sound <= 1;
            play <= PLAY_ON;
            count := timebase;
            if (data_out = 1) then

```

```

    gamma := G10a;
else
    nloss <= LED_ON;
    gamma := Ea;
end if;
elsif k(2)=KEY_ON then
    ind := 2;
    sound <= 2;
    play <= PLAY_ON;
    count := timebase;
    if (data_out = 2) then
        gamma := G10a;
    else
        nloss <= LED_ON;
        gamma := Ea;
    end if;
elsif k(3)=KEY_ON then
    ind := 3;
    sound <= 3;
    play <= PLAY_ON;
    count := timebase;
    if (data_out = 3) then
        gamma := G10a;
    else
        nloss <= LED_ON;
        gamma := Ea;
    end if;
else
    gamma := G10;
end if;
end if;
when G10a =>
    nl(ind) <= LED_ON;
    gamma := G11;
when G11 =>

```

```

if count = 0 then
    nl <= (others => LED_OFF);
    play <= PLAY_OFF;
    count := timebase;      -- attiva contatore LED spento
    gamma := G12;          -- stato FSM
else
    count := count - 1;     -- decrementa contatore
    gamma := G11;          -- stato FSM
end if;
when G12 =>
    if count = 0 then      -- controlla se fine conteggio
        if scan /= max then -- controlla se sequenza non finita
            scan := scan + 1; -- incrementa indirizzo
            gamma := G8;     -- stato FSM
        elsif max /= (SIZE_MEM - 1) then -- controlla se memoria non e' esaurita
            max := max + 1;   -- incrementa registro massima sequenza
            timebase := timebase - DEC_SEQ; -- decremento prossima sequenza
            gamma := G2;     -- stato FSM
        else
            play <= PLAY_ON;  -- attiva il suono
            sound <= S_WIN;   -- comunica il codice del suono
            count := COUNT_FIN; -- attiva contatore fine suono
            gamma := W0;      -- stato FSM
        end if;
    else
        count := count - 1;   -- decrementa contatore
        gamma := G12;        -- stato FSM
    end if;
when Ea =>
    nl(ind) <= LED_ON;       -- attiva LED tasto
    gamma := E0;            -- stato FSM
when E0 =>
    if count = 0 then        -- controlla se fine conteggio
        nl <= (others => LED_OFF); -- spegne LED tasti
        play <= PLAY_OFF;    -- disattiva il suono
    end if;

```

```

    count := timebase;      -- attiva contatore LED spento
    gamma := E1;            -- stato FSM
else
    count := count - 1;     -- decrementa contatore
    gamma := E0;            -- stato FSM
end if;
when E1 =>
    if count = 0 then      -- controlla se fine conteggio
        max := 0;         -- azzera registro massima sequenza
        gamma := K0;      -- stato FSM
    else
        count := count - 1; -- decrementa contatore
        gamma := E1;      -- stato FSM
    end if;
when K0 =>
    address <= max;        -- indirizza ultimo integer range 3 down to 0e
    gamma := K1;          -- stato FSM
when K1 =>                -- serve per dare tempo per leggere la memoria
    gamma := K2;          -- stato FSM
when K2 =>
    nl(data_out) <= LED_ON; -- attiva LED tasto
    play <= PLAY_ON;       -- attiva suono
    sound <= data_out;     -- comunica il codice del suono
    count := timebase;     -- attiva contatore LED acceso
    gamma := K3;          -- stato FSM
when K3 =>
    if count = 0 then      -- controlla se fine conteggio
        nl <= (others => LED_OFF); -- spegne LED tasti
        play <= PLAY_OFF;  -- disattiva il suono
        count := timebase; -- attiva contatore LED spento
        gamma := K4;      -- stato FSM
    else
        count := count - 1; -- decrementa contatore
        gamma := K3;      -- stato FSM
    end if;

```

```

when K4 =>
  if count = 0 then      -- controlla se fine conteggio
    if max /= scan then  -- controlla se fine lista
      max := max + 1;    -- incrementa indirizzo
      gamma := K0;       -- stato FSM
    else
      nl(data_out) <= LED_ON; -- attiva LED tasto
      play <= PLAY_ON;    -- attiva suono
      sound <= S_LOSS;    -- codice suono perdita
      count := COUNT_FIN; -- attiva contatore LED acceso
      gamma := K5;        -- stato FSM
    end if;
  else
    count := count - 1;   -- decrementa contatore
    gamma := K4;          -- stato FSM
  end if;
when K5 =>
  if count = 0 then      -- controlla se fine conteggio
    nl <= (others => LED_OFF); -- spegne LED tasti
    play <= PLAY_OFF;    -- disattiva il suono
    count := COUNT_FIN;  -- attiva contatore LED spento
    gamma := K6;         -- stato FSM
  else
    count := count - 1;   -- decrementa contatore
    gamma := K5;          -- stato FSM
  end if;
when K6 =>
  if count = 0 then      -- controlla se fine conteggio
    nl(data_out) <= LED_ON; -- attiva LED tasto
    play <= PLAY_ON;    -- attiva suono
    sound <= S_LOSS;    -- codice suono perdita
    count := COUNT_FIN;  -- attiva contatore LED acceso
    gamma := K5;        -- stato FSM
  else
    count := count - 1;   -- decrementa contatore

```

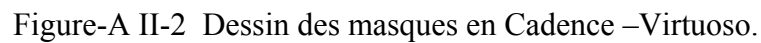
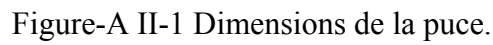


```

    gamma := K6;          -- stato FSM
end if;
when W0 =>
    if count = 0 then      -- controlla se fine conteggio
        nl <= (others => LED_ON); -- attiva tutti i LED
        play <= PLAY_OFF;  -- disattiva il suono
        count := COUNT_FIN; -- attiva contatore LED acceso
        gamma := W1;       -- stato FSM
    else
        count := count - 1; -- decrementa contatore
        gamma := W0;       -- stato FSM
    end if;
when W1 =>
    if count = 0 then      -- controlla se fine conteggio
        nl <= (others => LED_OFF); -- disattiva tutti i LED
        play <= PLAY_ON;   -- attiva il suono
        sound <= S_WIN;    -- comunica il codice del suono
        count := COUNT_FIN; -- attiva contatore LED spento
        gamma := W0;       -- stato FSM
    else
        count := count - 1; -- decrementa contatore
        gamma := W1;       -- stato FSM
    end if;
end case;
end if;
end process;
end BEHAV;

```

## CADENCE - DÉTAILS DE CONCEPTION DU DESSIN DES MASQUES POUR CIRCUIT B12



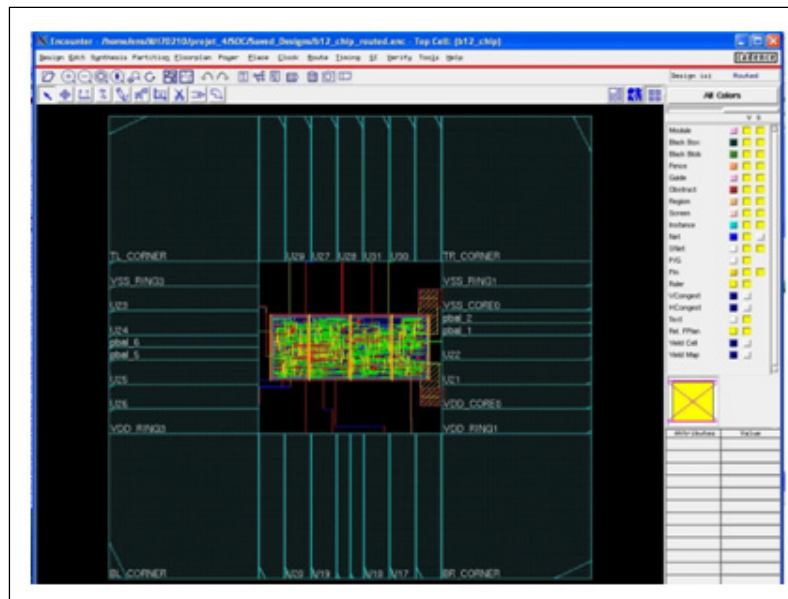


Figure-A II-3 Dessin des masques en Cadence –Encounter.

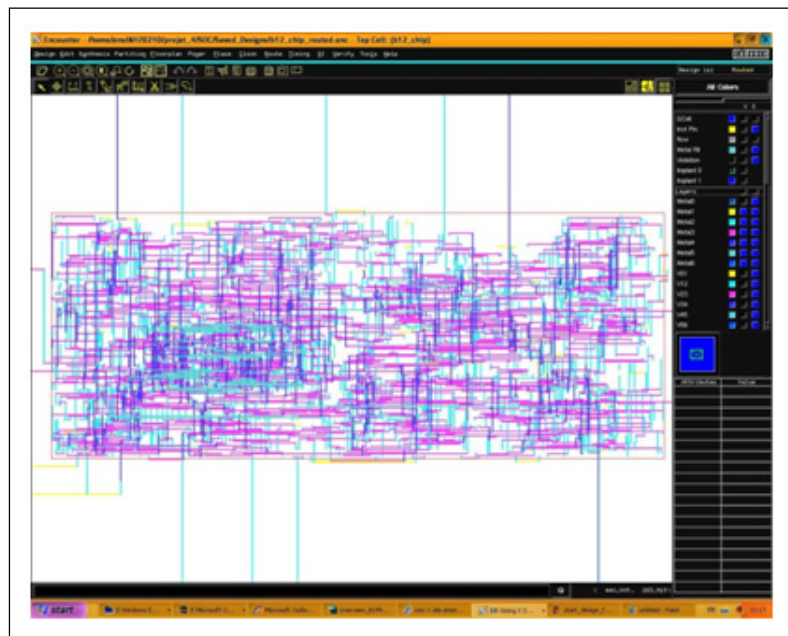
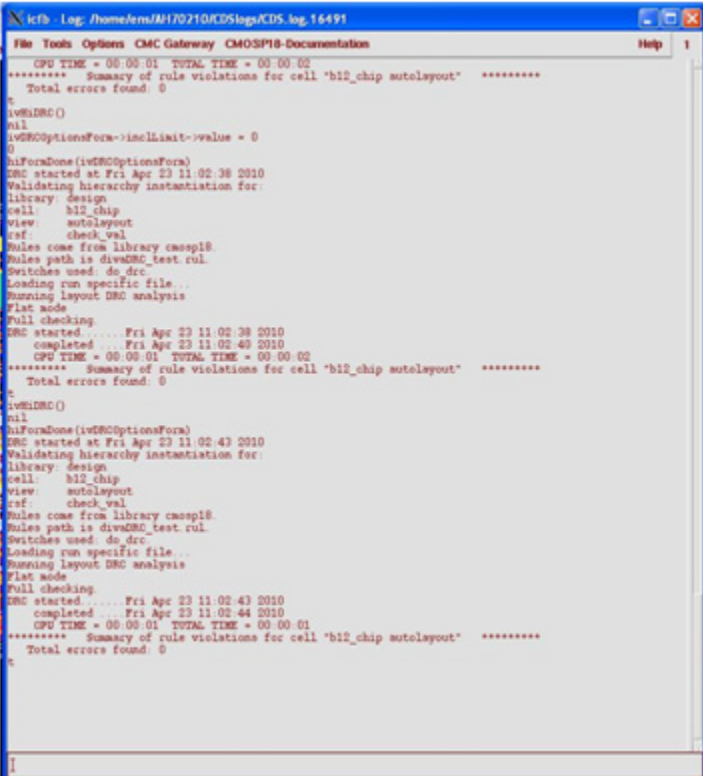


Figure-A II-4 Interconnexions métalliques sur le circuit b12 (sans aucun polygone de DRC).



```

kfb -log: /home/jens/AH70210/KDSbgs/KDS.log.16491
File Tools Options CMC Gateway CMO3P18-Documentation Help 1
CPU TIME = 00:00:01 TOTAL TIME = 00:00:02
***** Summary of rule violations for cell "b12_chip autolayout" *****
Total errors found: 0
t
t
ivvDRC()
nil
ivvDRCOptionsForm->inclLimit->value = 0
0
hiFormDone(ivvDRCOptionsForm)
DRC started at Fri Apr 23 11:02:38 2010
Validating hierarchy instantiation for:
library: design
cell: b12_chip
view: autolayout
rsf: check_val
Rules come from library cmo3p18
Rules path is divaDRC_test.rul.
Switches used: do_drc
Loading run specific file...
Running layout DRC analysis
Flat mode
Full checking.
DRC started ..... Fri Apr 23 11:02:38 2010
completed ..... Fri Apr 23 11:02:40 2010
CPU TIME = 00:00:01 TOTAL TIME = 00:00:02
***** Summary of rule violations for cell "b12_chip autolayout" *****
Total errors found: 0
t
t
ivvDRC()
nil
hiFormDone(ivvDRCOptionsForm)
DRC started at Fri Apr 23 11:02:43 2010
Validating hierarchy instantiation for:
library: design
cell: b12_chip
view: autolayout
rsf: check_val
Rules come from library cmo3p18
Rules path is divaDRC_test.rul.
Switches used: do_drc
Loading run specific file...
Running layout DRC analysis
Flat mode
Full checking.
DRC started ..... Fri Apr 23 11:02:43 2010
completed ..... Fri Apr 23 11:02:44 2010
CPU TIME = 00:00:01 TOTAL TIME = 00:00:01
***** Summary of rule violations for cell "b12_chip autolayout" *****
Total errors found: 0
t

```

Figure-A II-5 Rapport de Cadence – Virtuoso que la vérification des règles de design a été effectuée sans erreur.

## ANNEXE III

## FASTSCAN - ANALYSE INTERNE DU CIRCUIT B12

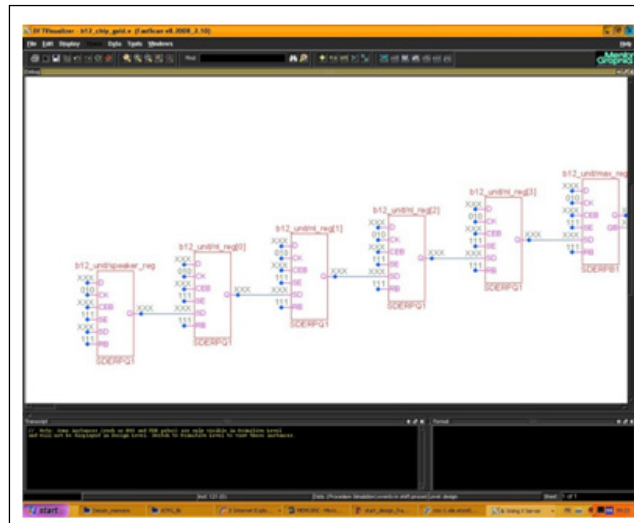


Figure-A III-1 Circuit b12 - Début de la chaîne d'analyse interne.

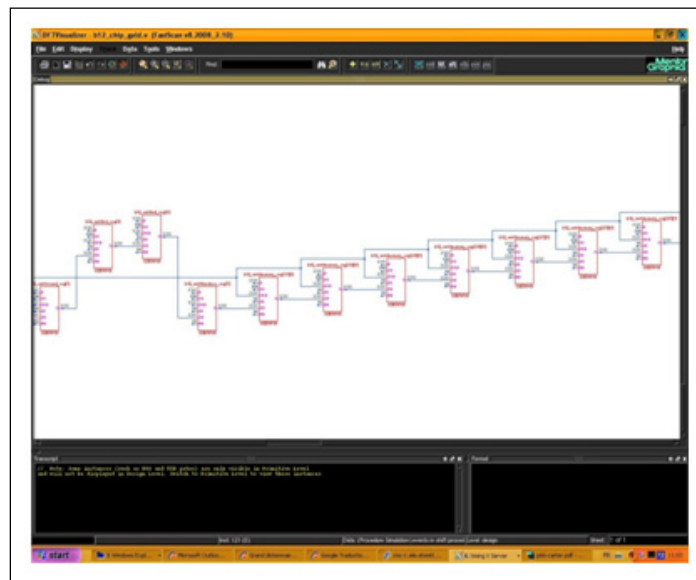


Figure-A III-2 Circuit b12 - Fragment de la chaine d'analyse interne.  
b12 - Fichier des vecteurs de test (fragment)

```

// FastScan v8.2008_2.10

// Design = /home/ens/AH70210/projet_4/dft_pads/b12_chip_gold.v
// Created = Fri Nov 13 11:57:59 2009
// Statistics:
//   Test Coverage = 97.83%
//   Total Faults = 5284
//   DS (det_simulation) = 4130
//   DI (det_implication) = 868
//   RE (redundant) = 175
//   AU (atpg_untestable) = 111
//   Total Patterns = 114
// Settings:
//   Simulation Mode = combinational, seq_depth = 0
//   Fault Type = stuck
//   Fault Mode = uncollapsed
//   Pos_Det Credit = 50%
//   Z external = X
//   Z internal = X
//   wired_net = WIRE
// Warnings:
// Clock Information:
//   /clock off-state = 0 type = shift-MASTER poflag = 0
//   /reset off-state = 0 type = reset-MASTER poflag = 0
ASCII_PATTERN_FILE_VERSION = 2;
SETUP =
  declare input bus "PI" = "/k[3]", "/k[2]", "/k[1]", "/k[0]", "/clock",
    "/reset", "/start", "/scan_in1", "/scan_en";
  declare output bus "PO" = "/nl[3]", "/nl[2]", "/nl[1]", "/nl[0]", "/speaker",
    "/nloss";
  clock "/clock" =
    off_state = 0;
    pulse_width = 1;
  end;
  clock "/reset" =
    off_state = 0;

```

```

    pulse_width = 1;
end;
scan_group "grp1" =
    scan_chain "chain1" =
        scan_in = "/scan_in1";
        scan_out = "/nloss";
        length = 121;
    end;
    procedure shift "grp1_load_shift" =
        force_sci "chain1" 0;
        force "/clock" 1 20;
        force "/clock" 0 30;
        period 40;
    end;
    procedure shift "grp1_unload_shift" =
        measure_sco "chain1" 10;
        force "/clock" 1 20;
        force "/clock" 0 30;
        period 40;
    end;
    procedure load "grp1_load" =
        force "/clock" 0 0;
        force "/reset" 0 0;
        force "/scan_en" 1 0;
        apply "grp1_load_shift" 121 40;
    end;
    procedure unload "grp1_unload" =
        force "/clock" 0 0;
        force "/reset" 0 0;
        force "/scan_en" 1 0;
        apply "grp1_unload_shift" 121 40;
    end;
end;
end;
CHAIN_TEST =

```

```

pattern = 0;
apply "grp1_load" 0 =
    chain "chain1" = "001100110011001100110011001100110011001100110011001100110011001\
        1100110011001100110011001100110011001100110011001100110011\
        001100110011001100110";
end;

apply "grp1_unload" 1 =
    chain "chain1" = "1100110011001100110011001100110011001100110011001100110011\
        00110011001100110011001100110011001100110011001100110011001\
        110011001100110011001";
end;

end;

SCAN_TEST =
    pattern = 0;
    apply "grp1_load" 0 =
        chain "chain1" = "00110111001000000110101000000100000110000000101000\
            01011000000000010011110000110110000101010110011001\
            110011001010000110100";
        end;

    force "PI" "110000010" 1;
    measure "PO" "101001" 2;
    pulse "/clock" 3;
    apply "grp1_unload" 4 =
        chain "chain1" = "01111000110111111001010111111011111001111111010111\
            10100111111111101110100001010100000010000011100110\
            101100111001111000011";
        end;

    pattern = 1;
    apply "grp1_load" 0 =
        chain "chain1" = "11110000001000010010101101100101000011010110100010\
            01111000101110011001101110001110000101000100111110\
            000000011101100101010";
        end;

    force "PI" "100000010" 1;
    measure "PO" "010100" 2;

```



```

pulse "/clock" 3;
apply "grp1_unload" 4 =
    chain "chain1" = "101111111011101101010010011010111100101001011101\
        10000111011101100101101110000000000111000000111110\
        001111110001111010001";
end;

-----

pattern = 113;

-----

force "PI" "011100100" 1;
measure "PO" "000001" 2;
pulse "/clock" 3;
apply "grp1_unload" 4 =
    chain "chain1" = "10001011000011101000010110100101110010110010111100\
        11001111000100101100001010010000000010000000111010\
        001110100000000011111";
end;
end;
SCAN_CELLS =
    scan_group "grp1" =
        scan_chain "chain1" =
            scan_cell = 0 MASTER TFFF "/b12_unit/nloss_reg" "I2" "SD" "Q";
            scan_cell = 1 MASTER TFFF "/b12_unit/count_reg[1]" "I2" "SD" "Q";
            scan_cell = 2 MASTER TFFF "/b12_unit/count_reg[0]" "I2" "SD" "Q";
            scan_cell = 3 MASTER TFFF "/b12_unit/s_reg" "I2" "SD" "Q";
            scan_cell = 4 MASTER TFFF "/b12_unit/memory_reg[0][0]" "I2" "SD" "Q";

            -----

            scan_cell = 118 MASTER FTFF "/b12_unit/nl_reg[1]" "I2" "SD" "Q";
            scan_cell = 119 MASTER FTFF "/b12_unit/nl_reg[0]" "I2" "SD" "Q";
            scan_cell = 120 MASTER FTFF "/b12_unit/speaker_reg" "I2" "SD" "Q";
        end;
    end; end;

```

## ANNEXE IV

### B12\_CHIP.DEF (FRAGMENT)

```
#
# created by First Encounter v06.10-s158_1 on Mon Oct 5 15:34:06
VERSION 5.6 ;
DIVIDERCHAR "/" ;
BUSBITCHARS "[]" ;
DESIGN b12_chip ;
UNITS DISTANCE MICRONS 1000 ;
PROPERTYDEFINITIONS
    COMPONENTPIN designRuleWidth REAL ;
    DESIGN FE_CORE_BOX_LL_X REAL 258.060 ;
    DESIGN FE_CORE_BOX_UR_X REAL 500.545 ;
    DESIGN FE_CORE_BOX_LL_Y REAL 325.360 ;
    DESIGN FE_CORE_BOX_UR_Y REAL 423.920 ;
END PROPERTYDEFINITIONS
DIEAREA ( 0 0 ) ( 758545 750000 ) ;
ROW CORE_ROW_0 core 258060 325360 FS DO 367 BY 1 STEP 660 0 ;
-----
ROW CORE_ROW_15 core 258060 417760 N DO 367 BY 1 STEP 660 0 ;

TRACKS Y 280 DO 1339 STEP 560 LAYER metal6 ;
-----
TRACKS Y 280 DO 1339 STEP 560 LAYER metal1 ;
GCELLGRID X -5 DO 2 STEP 9180 ;
GCELLGRID X 18325 DO 81 STEP 9150 ;
GCELLGRID X 758550 DO 1 STEP 8225 ;
GCELLGRID Y -25 DO 82 STEP 9150 ;
GCELLGRID Y 750005 DO 1 STEP 8880 ;

VIAS 26 ;
- VIAGEN56_1
```

```

+ VIARULE VIAGEN56
+ CUTSIZE 360 360
+ LAYERS metal5 via56 metal6
+ CUTSPACING 360 360
+ ENCLOSURE 260 260 260 260
+ ROWCOL 1 1
;
-----
END VIAS
COMPONENTS 617 ;
- VDD_RING0 PVDD2DGZ + SOURCE DIST + FIXED ( 236070 515000 ) S
;
- VSS_RING0 PVSS2DGZ + SOURCE DIST + FIXED ( 482475 515000 ) S
;
- VDD_RING1 PVDD2DGZ + SOURCE DIST + FIXED ( 523545 235000 ) W
;
- VDD_CORE0 PVDD1DGZ + SOURCE DIST + FIXED ( 523545 275000 ) W
;
-----
END COMPONENTS
PINS 15 ;
- nl[3] + NET nl[3] + DIRECTION OUTPUT + USE SIGNAL
;
- nl[2] + NET nl[2] + DIRECTION OUTPUT + USE SIGNAL
;
-----
END PINS
SPECIALNETS 2 ;
- VSS ( * VSS )
+ ROUTED metal1 1480 + SHAPE COREWIRE ( 255860 331520 ) ( 258060 * )
  NEW metal1 1480 + SHAPE COREWIRE ( 255860 343840 ) ( 258060 * )
  NEW metal1 1480 + SHAPE COREWIRE ( 255860 356160 ) ( 258060 * )
-----
  NEW metal5 0 + SHAPE IOWIRE ( 502260 455000 ) VIAGEN45_12
+ USE GROUND

```

```

;
- VDD ( * VDD )
+ ROUTED metal1 940 + SHAPE COREWIRE ( 253600 325630 ) ( 258060 * )
  NEW metal1 1480 + SHAPE COREWIRE ( 253600 337680 ) ( 258060 * )
  NEW metal1 1480 + SHAPE COREWIRE ( 253600 350000 ) ( 258060 * )
-----
+ USE POWER
;
END SPECIALNETS
NETS 620 ;
- nl[3]
  ( PIN nl[3] ) ( U17 PAD )
;
- nl[2]
  ( PIN nl[2] ) ( U18 PAD )
-----
- n32
  ( U17 I ) ( b12_unit\nl_reg\[3\] Q ) ( b12_unit\max_reg\[2\] SD )
  ( b12_unit\U2104 B1 )
+ ROUTED metal2 ( 354750 341600 ) ( * 346360 ) VIA2
  NEW metal3 ( 319110 348600 ) ( 354750 * ) VIA3
  NEW metal2 ( 319110 347760 ) ( * 348600 ) VIA2
  NEW metal1 ( 319110 346640 0 ) ( * 347760 ) VIA1
  NEW metal6 ( 475530 234700 0 ) ( * 259000 )
  NEW metal6 ( 474540 259000 ) ( 475530 * )
  NEW metal6 ( 474540 259000 ) ( * 346360 ) VIA5
  NEW metal5 ( 354750 346360 ) ( 474540 * )
  NEW metal4 ( 354750 346360 ) ( * 348600 )
  NEW metal4 ( 354750 348600 ) ( * 362040 )
  NEW metal4 ( 354750 362040 ) ( 355410 * )
  NEW metal4 ( 355410 362040 ) ( * 401240 ) VIA3
  NEW metal3 ( 354750 401240 ) ( 355410 * )
  NEW metal2 ( 354750 401240 ) ( * 402080 ) VIA1
  NEW metal2 ( 354750 341600 ) VIA1
  NEW metal5 ( 354750 346360 ) VIA4

```

```

NEW metal4 ( 354750 346360 ) VIA3
NEW metal3 ( 354750 401240 ) VIA2
;
- n33
( U18 I ) ( b12_unit\nl_reg\[3\] SD ) ( b12_unit\nl_reg\[2\] Q )
( b12_unit\U2102 B1 )
+ ROUTED metal3 ( 321090 335160 ) ( 336930 * ) VIA2
NEW metal2 ( 321090 334320 ) ( * 335160 ) VIA2
NEW metal2 ( 433290 234700 0 ) ( * 246680 ) VIA1
NEW metal1 ( 355410 246680 ) ( 433290 * )
NEW metal2 ( 355410 246680 ) ( * 326760 ) VIA1
NEW metal1 ( 355410 326760 ) ( * 327600 0 )
NEW metal2 ( 354750 329280 ) ( 355410 * ) VIA1
NEW metal2 ( 354750 329280 ) ( * 331800 ) VIA2
NEW metal3 ( 336930 331800 ) ( 354750 * )
NEW metal2 ( 336930 331800 ) ( * 335160 )
NEW metal2 ( 336930 335160 ) ( * 340480 ) VIA1
NEW metal2 ( 321090 334320 ) VIA1
NEW metal2 ( 355410 246680 ) VIA1
NEW metal3 ( 336930 331800 ) VIA2
;
- n34
( U19 I ) ( b12_unit\nl_reg\[2\] SD ) ( b12_unit\nl_reg\[1\] Q )
( b12_unit\U2100 B1 )
+ ROUTED metal1 ( 337590 332920 ) ( 340230 * )
NEW metal1 ( 340230 332920 ) ( * 333200 )
NEW metal1 ( 340230 333200 ) ( 340890 * 0 )
NEW metal3 ( 319110 326760 ) ( 337590 * ) VIA2
NEW metal2 ( 319110 326760 ) ( * 327880 ) VIA1
NEW metal1 ( 319110 327880 ) ( 319770 * )
NEW metal1 ( 319770 327880 ) ( * 328720 0 )
NEW metal2 ( 350790 234700 0 ) ( * 274680 ) VIA1
NEW metal1 ( 337590 274680 ) ( 350790 * )
NEW metal2 ( 337590 274680 ) ( * 326760 )
NEW metal2 ( 337590 326760 ) ( * 328160 ) VIA1

```

```

NEW metal2 ( 337590 328160 ) ( * 332920 ) VIA1
NEW metal3 ( 319110 326760 ) VIA2
NEW metal2 ( 337590 274680 ) VIA1
;
- n35
( U20 I ) ( b12_unit\nl_reg\[1\] SD ) ( b12_unit\nl_reg\[0\] Q )
( b12_unit\U2098 B1 )
+ ROUTED metal2 ( 332970 335720 ) ( * 336840 ) VIA2
NEW metal3 ( 332970 336840 ) ( 341550 * ) VIA2
NEW metal2 ( 341550 335440 ) ( * 336840 )
NEW metal2 ( 315150 335160 ) ( * 335720 ) VIA2
NEW metal1 ( 315150 335160 ) ( 315810 * )
NEW metal1 ( 315810 334320 0 ) ( * 335160 )
NEW metal2 ( 309870 234700 0 ) ( * 325640 ) VIA2
NEW metal4 ( 309870 325640 ) ( * 335720 ) VIA3
NEW metal3 ( 309870 335720 ) ( 315150 * )
NEW metal3 ( 315150 335720 ) ( 323070 * ) VIA2
NEW metal3 ( 323070 335720 ) ( 332970 * ) VIA2
NEW metal2 ( 323070 334880 ) ( * 335720 )
NEW metal2 ( 341550 335440 ) VIA1
NEW metal2 ( 315150 335160 ) VIA1
NEW metal4 ( 309870 325640 ) VIA3
NEW metal2 ( 323070 334880 ) VIA1
;
-----
END NETS
END DESIGN

```

## ANNEXE V

### DESCRIPTION DÉTAILLÉE DU TRAITEMENT DE LA MÉTHODE 1, LA SIGNATURE À PONDÉRATION CONSTANTE (EQUIPROBABLE)

Pour trouver la couverture de chaque bascule, le traitement de chaque transcription est exécuté conformément à l'organigramme de la figure-A V.1.

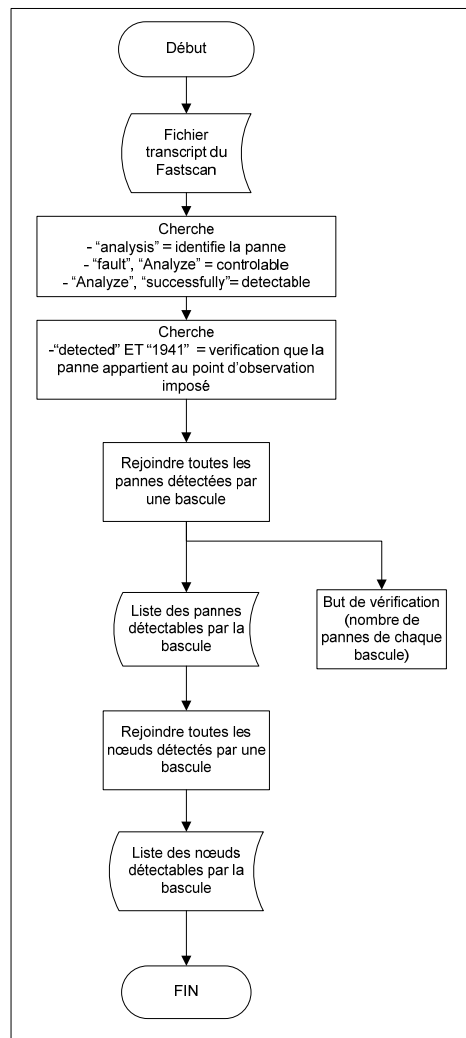


Figure-A V-1 Traitement pour la création de la couverture d'une bascule (exemplifié pour la bascule B1941).

Il est basé sur la recherche des mots pertinents qui déterminent le contrôle et l'observabilité de la panne. Dans cette manière, nous obtenons la liste de pannes et la liste des nœuds Mentor couverts par chaque bascule.

Une préparation importante pour toutes les méthodes de génération de signature a été la création d'un fichier complet des nœuds de Cadence, sans les nœuds des signaux globaux, et contenant tous les nœuds Mentor testables du circuit avec leur nœud d'appartenance de Cadence, le coefficient de testabilité et le facteur de pondération NMNC (nœud Mentor / nœud Cadence). La correspondance avec Cadence donne l'appartenance aux signaux globaux. À cette fin, nous avons créé dans un premier temps la liste complète des pannes non testables, `b12_nM_non testable`. L'étape suivante consiste à obtenir les coefficients de testabilité pour les pannes non testables conformément à l'organigramme de la figure-A V-2. Essentiellement, nous cherchons tous les nœuds impliqués dans la liste des pannes non testables fournie par Fastscan. Si un nœud s'y retrouve deux fois, ça veut dire qu'il est complètement non testable; nous attribuons la valeur 0 à son coefficient de testabilité. S'il se trouve juste une seule fois, nous lui attribuons la valeur 0,5.

Le traitement pour les nœuds testables qui ne contiennent pas des signaux globaux est montré dans la figure-A V-3. Premièrement, pour chaque nœud Mentor nous cherchons le nœud Cadence qui lui correspond dans la liste d'équivalence, et le cas échéant nous lui associons ce nœud de Cadence et le coefficient NMNC. S'il ne se trouve pas dans cette liste, le nœud Mentor peut être effacé puisqu'il n'est pas un signal local. Dans la phase suivante, chaque nœud est compté dans la liste de pannes testables. S'il s'y trouve deux fois, nous attribuons à son coefficient de testabilité la valeur 1, s'il s'y trouve juste une fois, nous lui attribuons la valeur 0,5. Il y a aussi une procédure de vérification pour les nœuds avec la testabilité de 0,5. Normalement ceux-ci doivent posséder une autre pondération de 0,5 dans la liste de pannes non testables pour avoir la certitude qu'ils ne sont pas manqués ou comptés deux fois. Cette procédure est faite dans le traitement de la figure-A V-4, Traitement des nœuds Mentor pour la testabilité et pour la correspondance aux nœuds de Cadence.



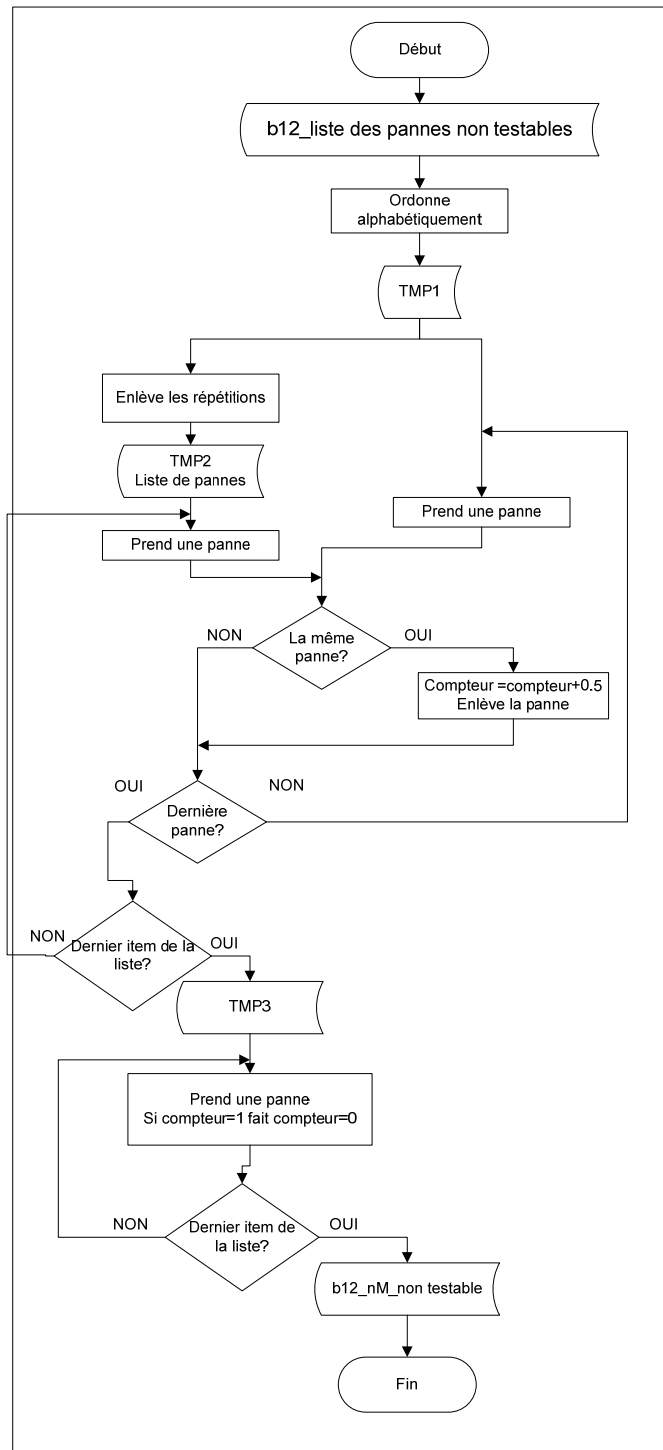


Figure-A V-2 Traitement pour obtenir les coefficients de testabilité des nœuds associés pour les pannes non testables.

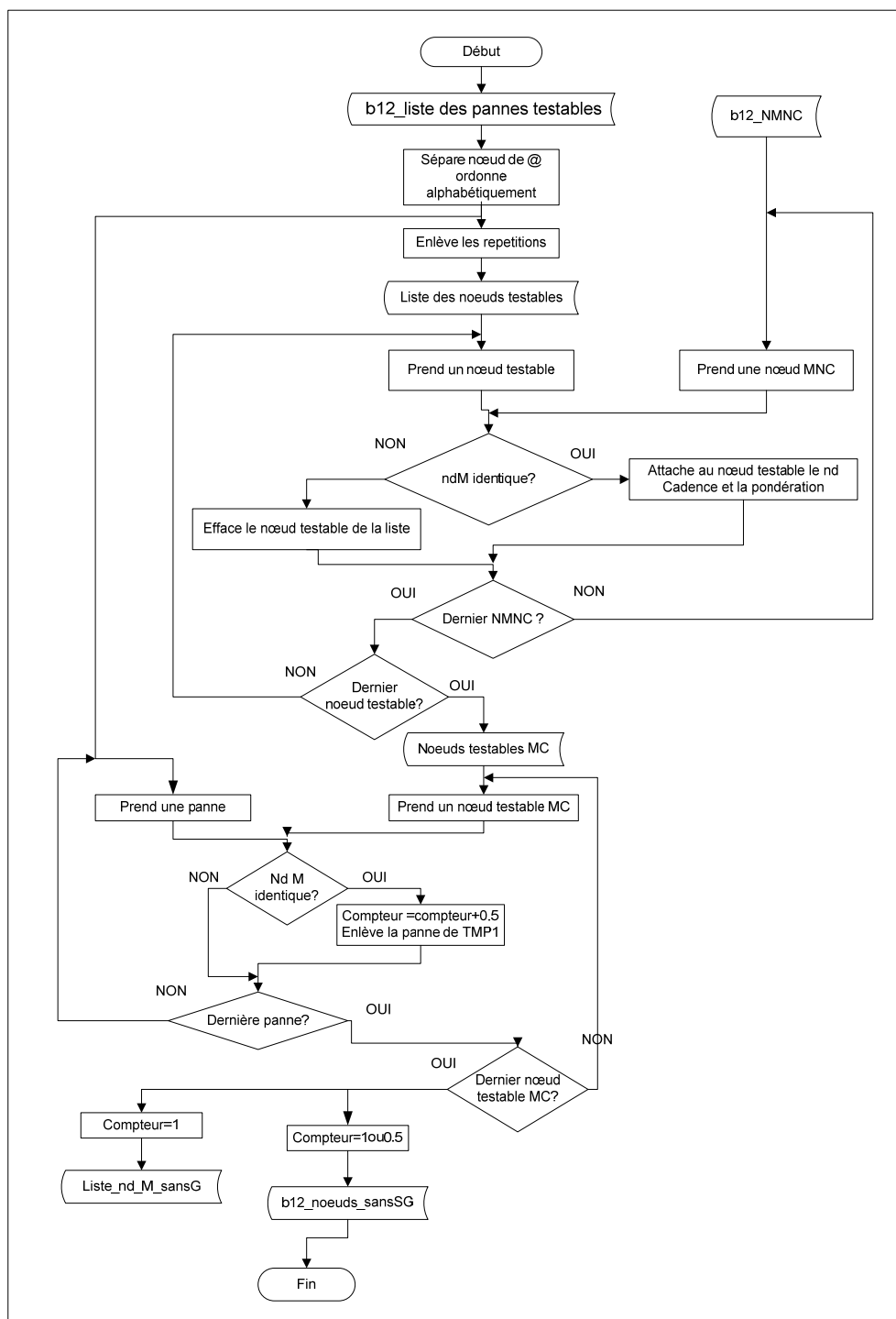


Figure-A V-3 Traitement pour les pannes testables sans signaux globaux.

L'annexe VI présente une explication détaillée de l'utilité du coefficient NMNC et sur la couverture d'un nœud de Cadence par plusieurs bascules de détection. Une fois obtenue la liste des pannes testables, nous construisons le vrai fichier de couverture de la bascule, dénommé "vue", qui va contenir les noms des pannes couvertes, leur nœud Cadence associé, le coefficient de testabilité et la correspondance avec les nœuds de Cadence. Cette construction est illustrée aux figures-A V-5 et V-6. Dans la figure-A V-5, Traitement pour construire la vue d'une bascule, nous commençons par le changement de nom des bascules et après, le traitement est similaire au traitement général présenté antérieurement. i.e. que nous éliminons les nœuds non métalliques, s'il y en a, ensuite nous recherchons dans le fichier le coefficient de testabilité et l'appartenance de nœud de Cadence. Au moment de l'identification d'un nœud, les paramètres de la base de données lui sont assignés. Pour ne répéter pas l'assignation, après qu'un nœud ait été identifié, le nœud du fichier initial est effacé.

Une fois obtenue la "vue" des bascules, pour obtenir la signature du circuit pour la zone d'intérêt ciblée, nous suivons la procédure décrite dans la figure-A V.7 Traitement pour construire la signature d'un circuit à pondération constante. Pour des raisons de compatibilité avec les autres méthodes, nous mentionnons que notre zone d'intérêt est ciblée pour 15 bascules (circuit b12). Donc la procédure consiste à effectuer les opérations mathématiques indiquées par les équations 4.2 et 4.3. Bien que les formules mathématiques prennent en considération le cas général des nœuds complètement testables dus aux bascules différentes, par la sensibilité relative  $A_{N_j}$  du nœud, nous n'avons pas rencontré cette situation, qui en effet simplifie la procédure. Le circuit n'est pas très complexe par sa structure et en même temps nous avons parcouru un quart de ses bascules. Ce problème pourra servir comme un point des futurs travaux. Une autre mention pour le fichier liste des nœuds distincts couverts par 15 bascules, obtenues dans la construction de la signature, même s'il n'est pas utilisé par la première méthode, il sera nécessaire dans toutes les autres méthodes.

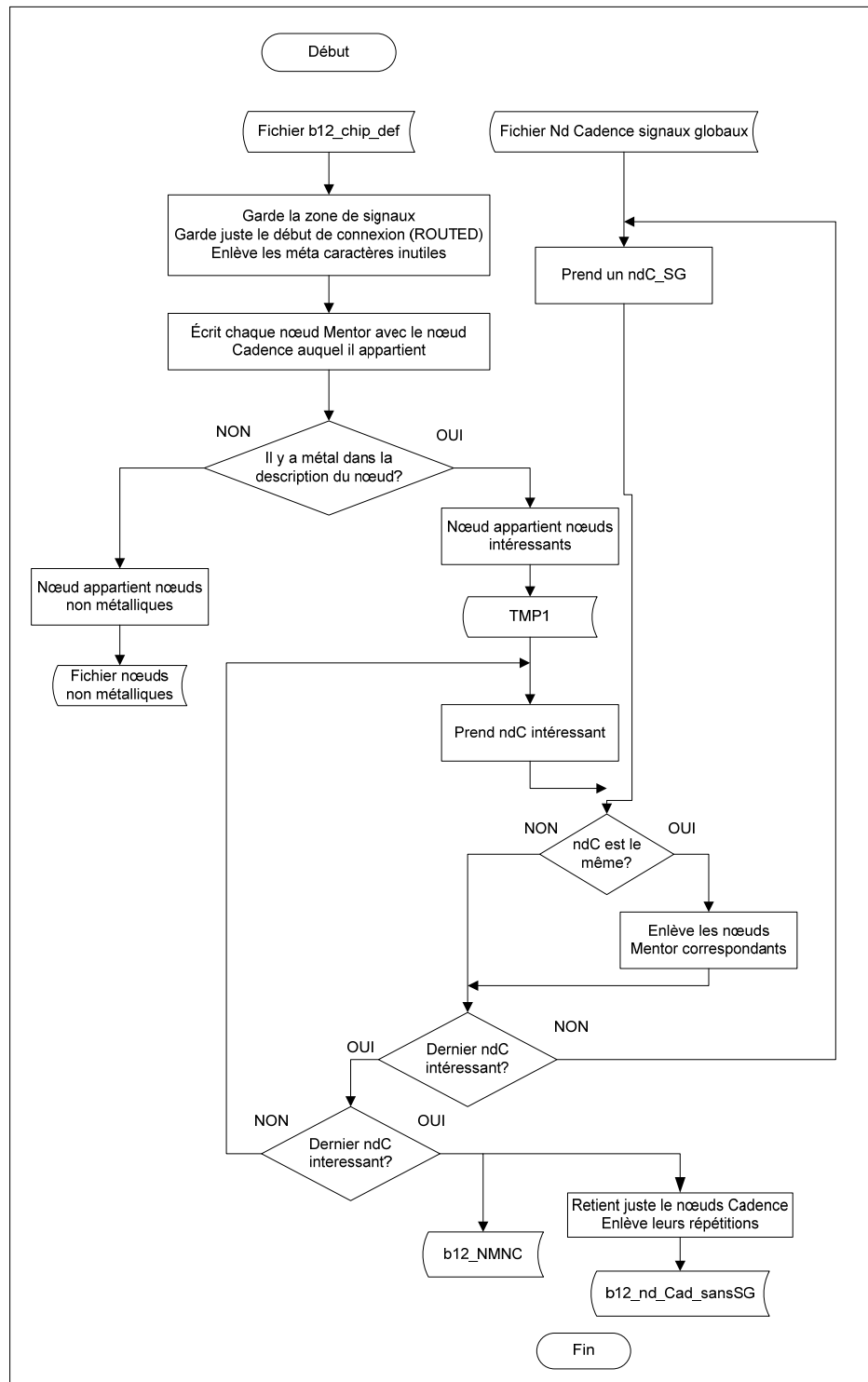


Figure-A V-4 Traitement des nœuds Mentor pour la testabilité et pour la correspondance aux nœuds de Cadence.

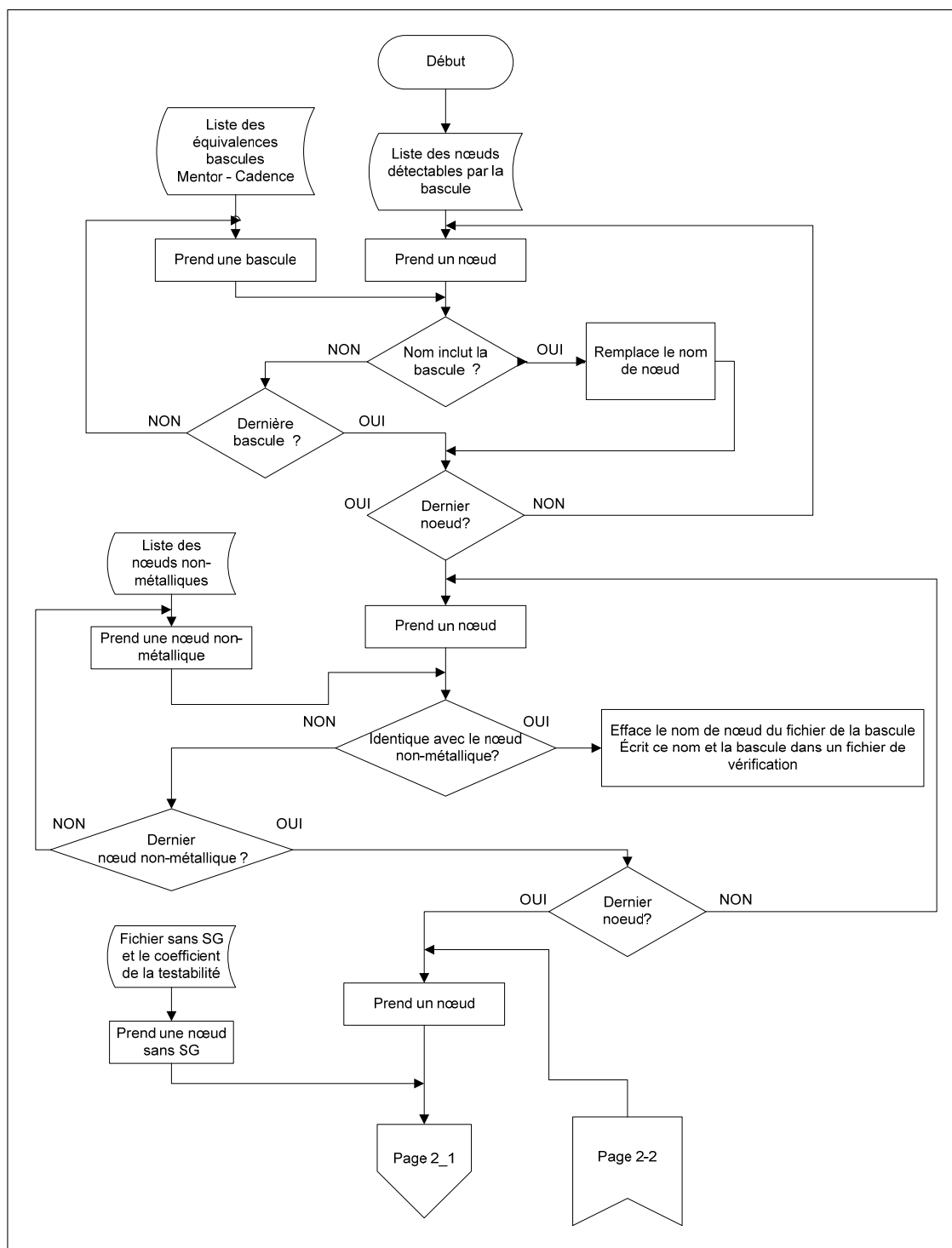


Figure-A V-5 Traitement pour construire la vue d'une bascule (partie 1).

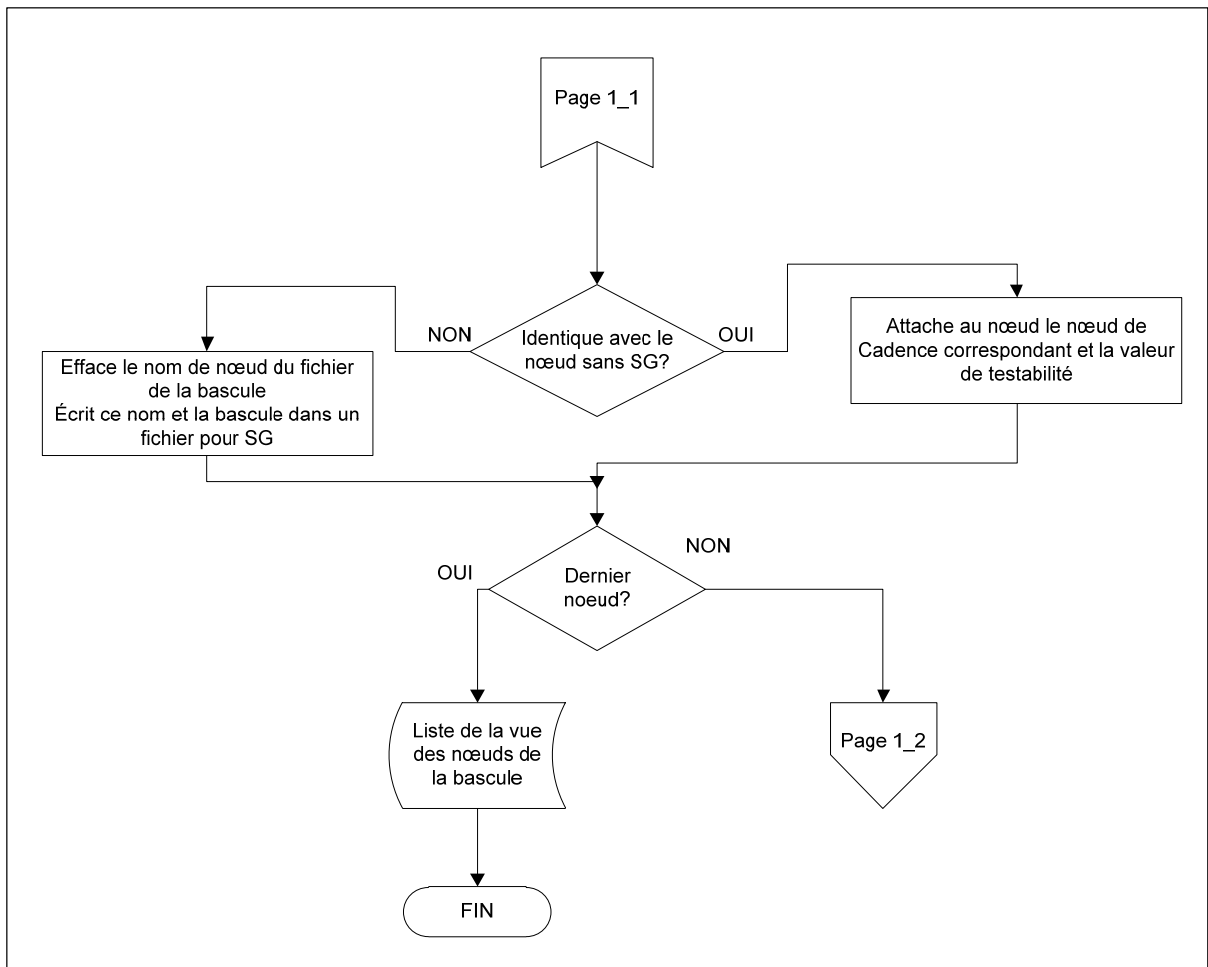


Figure-A V-6 Traitement pour construire la vue d'une bascule (partie 2).

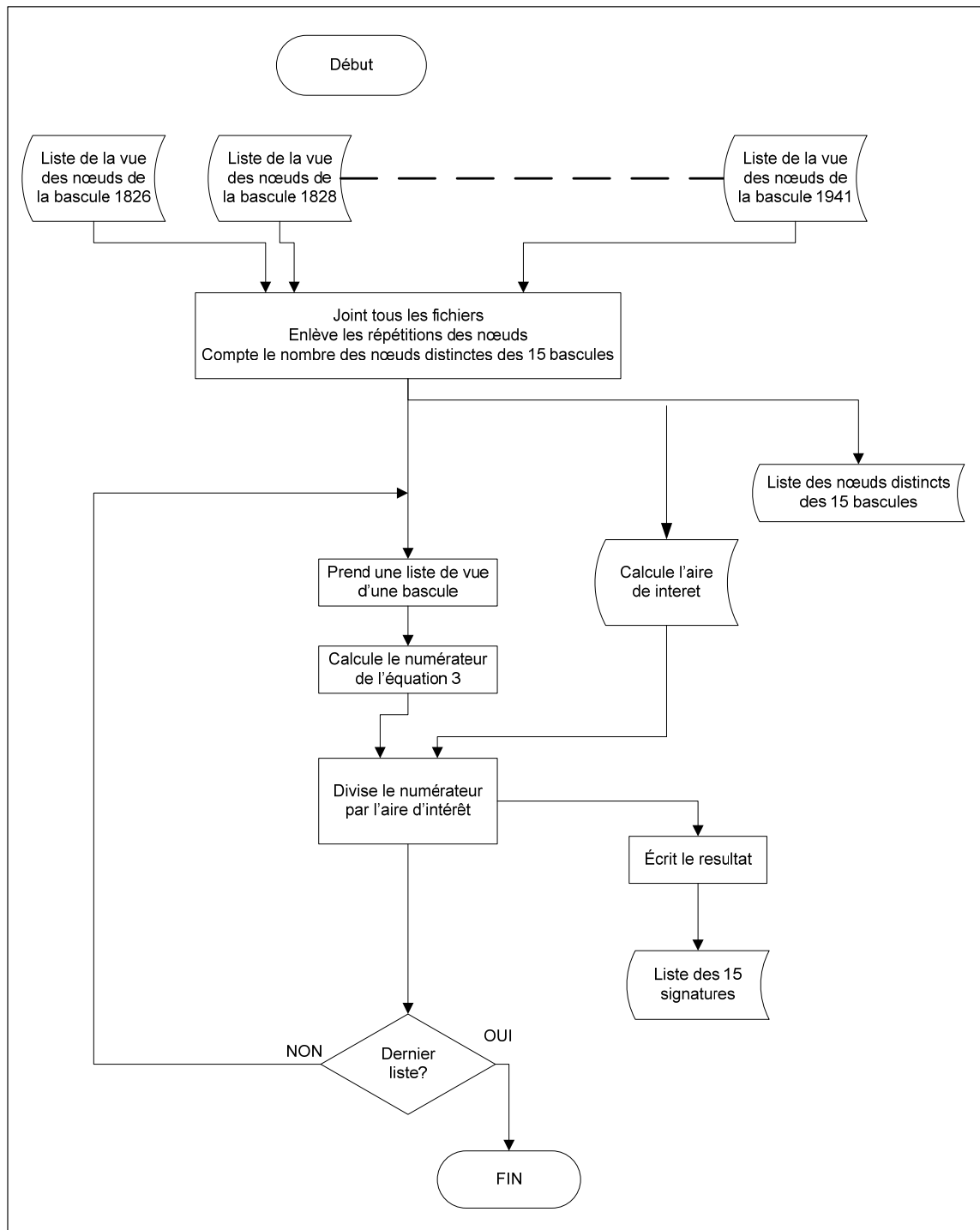


Figure-A V-7 Traitement pour construire la signature d'un circuit à pondération constante.

## ANNEXE VI

### EXPLICATION DÉTAILLÉE SUR LA NÉCESSITÉ D'INTRODUCTION DE NMNC ET LA COUVERTURE D'UN NŒUD DE CADENCE PAR PLUSIEURS BASCULES D'ANALYSE INTERNE

Si nous prenons le nœud Cadence *n99* (circuit b02 de la série ITC99), nous pouvons observer qu'il correspond à 4 nœuds Mentor, tel qu'illustré à la Figure-A suivante.

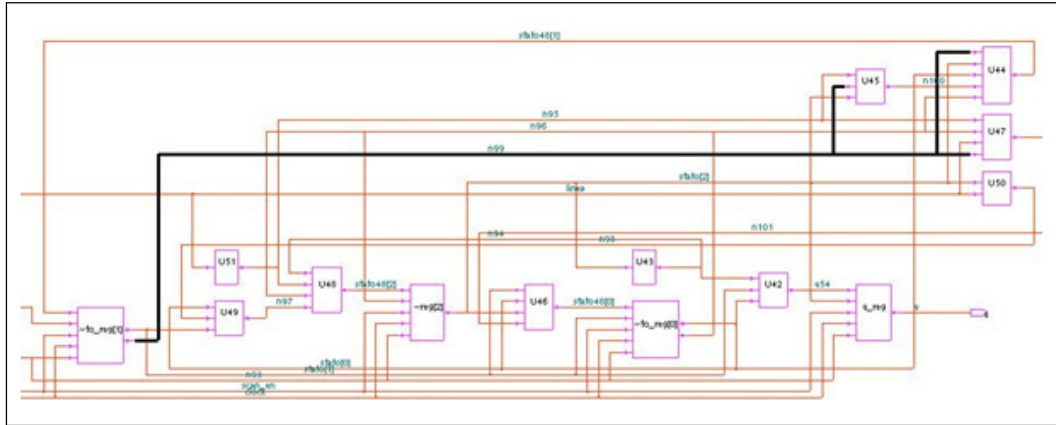


Figure-A VI-1 Nœud Cadence n99.

Après une analyse de la transcription de Fastscan, décrite au chapitre III, nous avons pour ce nœud Cadence les résultats du tableau suivant, où 1 est la valeur du coefficient de testabilité attribuée lorsque la bascule couvre le nœud Mentor correspondant pour le deux pannes et 0 dans le cas où ce nœud Mentor n'est pas détectable par cette bascule.

Tableau-A VI-1 Nœud Cadence associé aux nœuds Mentor

NŒUDS CADENCE	(B46) STATO_REG[2]	(B47) /STATO_REG[0]	(B48) /STATO_REG[1]	(B49) /U_REG	NŒUDS MENTOR
n99	0	0	1	0	/U45/A2
n99	0	0	1	0	/U44/A1
n99	0	1	0	0	/U47/B2
n99	0	1	1	0	/STATO_REG[1]/QB



Le nom de la bascule de la première ligne est celui donné par Mentor, auquel nous ajoutons B conformément à notre convention et en deuxième ligne le nom donné par Cadence. Par exemple, la bascule B48 peut détecter 3 des 4 nœuds Mentor correspondant au nœud Cadence n99, i.e. les nœuds Mentor U45/A2, U44/A1 et stato\_reg[1]/QB.

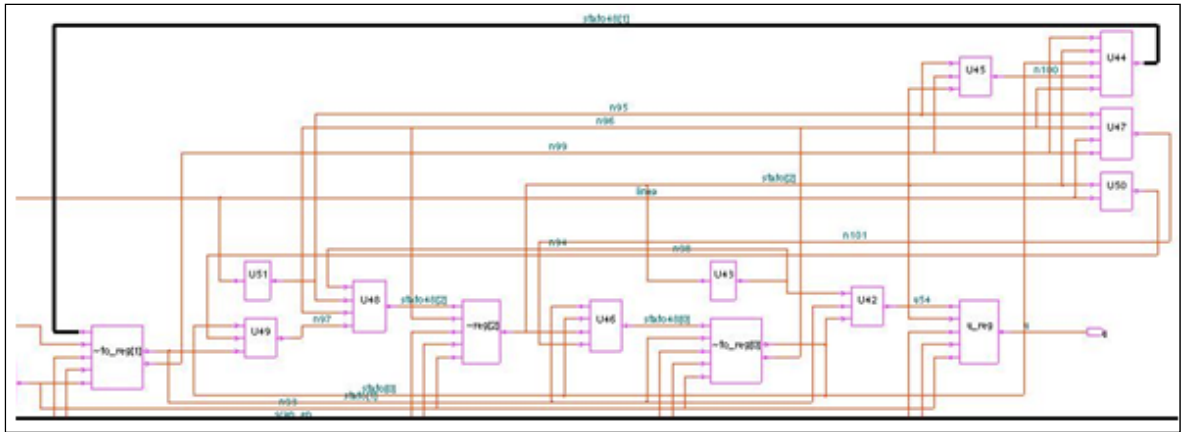


Figure-A VI-2 Connexion de la bascule B48.

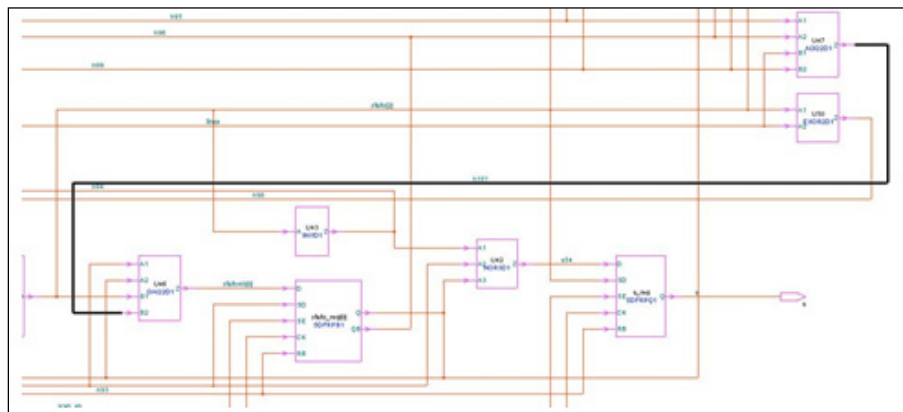


Figure-A VI-3 Connexion de la bascule B47.

Quelques observations importantes :

- un nœud de Cadence peut être couvert par plusieurs bascules en ATPG;
- les nœuds créés par le fan-out ne sont testables que via les bascules en aval.

## ANNEXE VII

### DESCRIPTION DÉTAILLÉE DES TRAITEMENTS SPÉCIFIQUES DE LA MÉTHODE 2

L'outil de Cadence, First Encounter, produit le fichier *Capacités parasites des nœuds Cadence* qui, sous la forme d'un tableau, présente tous les nœuds du circuit et leurs capacités parasites, obtenues par l'addition des capacités des connexions et celle des broches, le cas échéant. Le traitement effectué, conformément à la figure-A VII-1, assigne la valeur de la capacité parasite aux nœuds Mentor par l'intermédiaire des nœuds Cadence associés, le tout résultant en une base de données enrichie. Le mécanisme est basé sur deux boucles, une pour passer à travers tous les nœuds Cadence du fichier des capacités parasites en les comparant un à un avec les nœuds de Mentor du fichier les contenant (b12\_testability\_M\_noeuds\_sansSG) jusqu'au moment où il y a correspondance entre les nœuds de Cadence. À ce moment, nous attribuons au nœud Mentor la capacité parasite correspondante et nous avançons vers le suivant nœud du fichier répertoire des nœuds Mentor. Après la fin du traitement nous vérifions que tous les nœuds ont reçu leur capacité parasite. Pour calculer les signatures désirées, nous appliquons le traitement de la figure-A VII-2. La procédure consiste à effectuer les opérations mathématiques indiquées par les équations 5.2 et 5.3. Les deux traitements sont également effectués en awk.

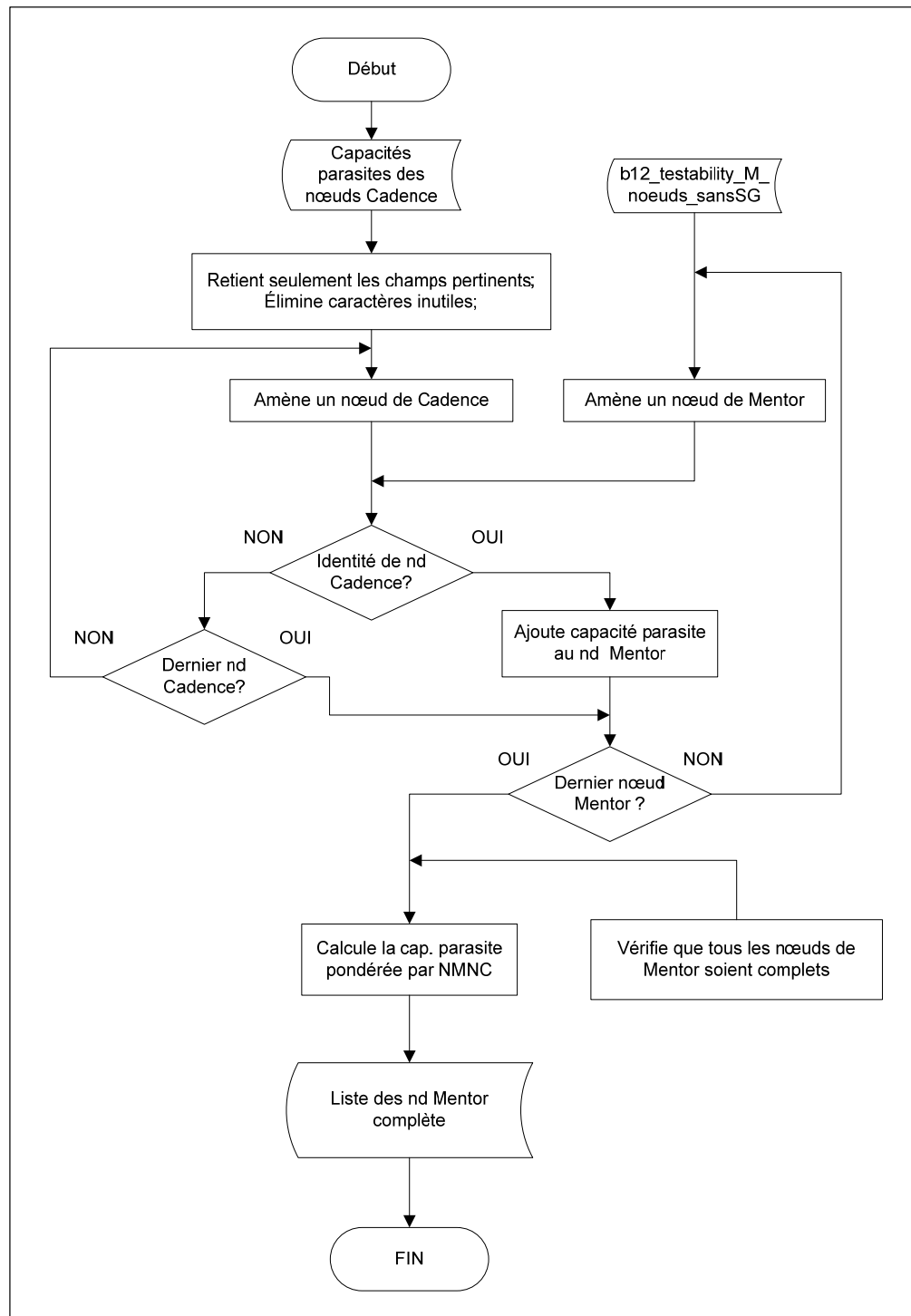


Figure-A VII-1 Assignment de la capacité parasite aux nœuds de Mentor.

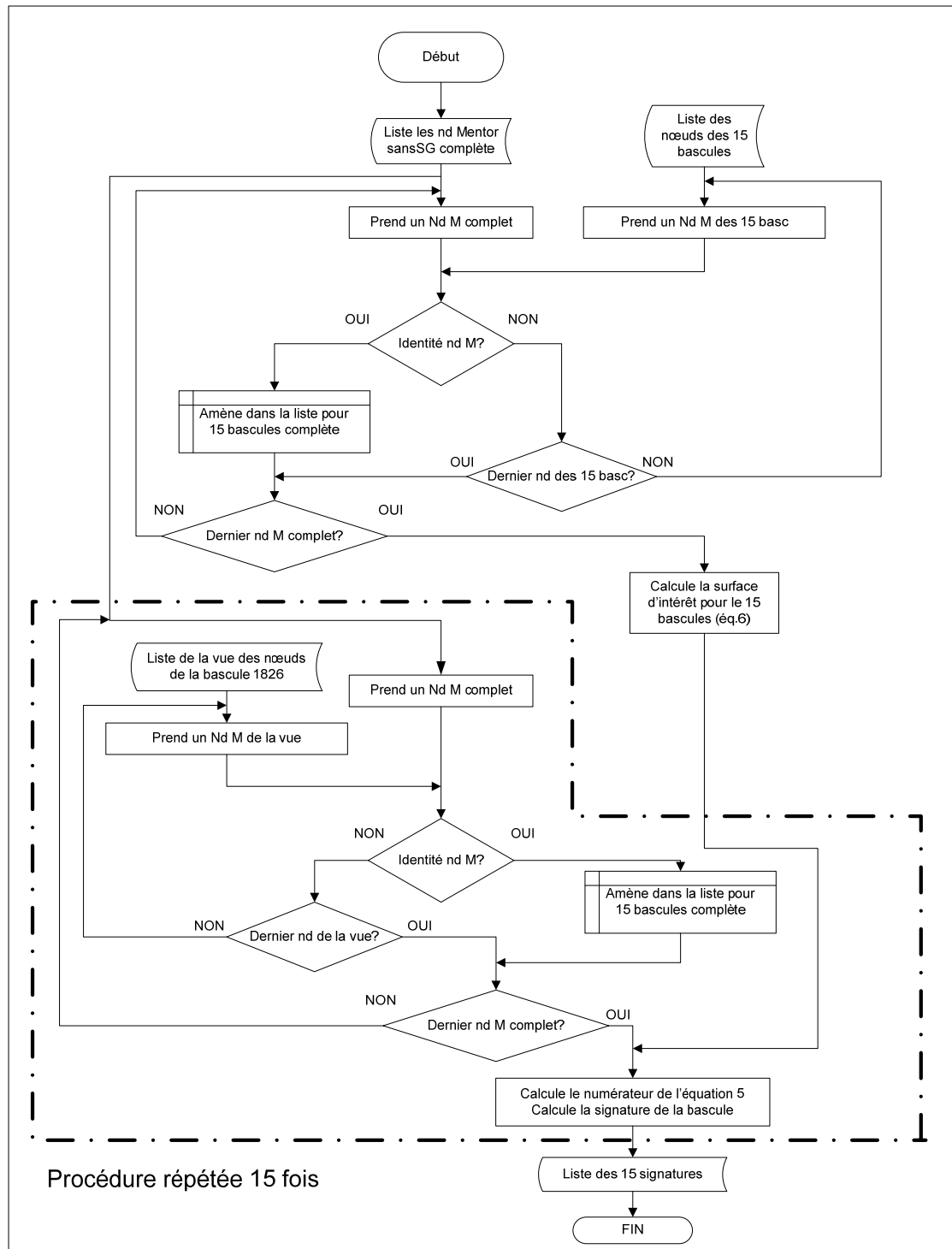


Figure-A VII-2 Traitement pour construire une signature pondérée par les capacités parasites des nœuds.

## ANNEXE VIII

### DESCRIPTION DÉTAILLÉE DES TRAITEMENTS SPÉCIFIQUES DE LA MÉTHODE 3

L'outil de Cadence, First Encounter, ne fournit pas la répartition des couches sur les nœuds. Le traitement effectué conformément à la Figure-A VIII.1, à partir du fichier DEF, fournit cette répartition. Il donne comme résultat la liste des nœuds Mentor chacun suivi par 6 coefficients, un par couche métallique. La valeur du coefficient pour une couche donnée est égale à 1 si le nœud utilise cette couche, zéro s'il ne l'utilise pas. C'est un traitement de texte qui découpe les informations pertinentes aux nœuds et leurs couches associées.

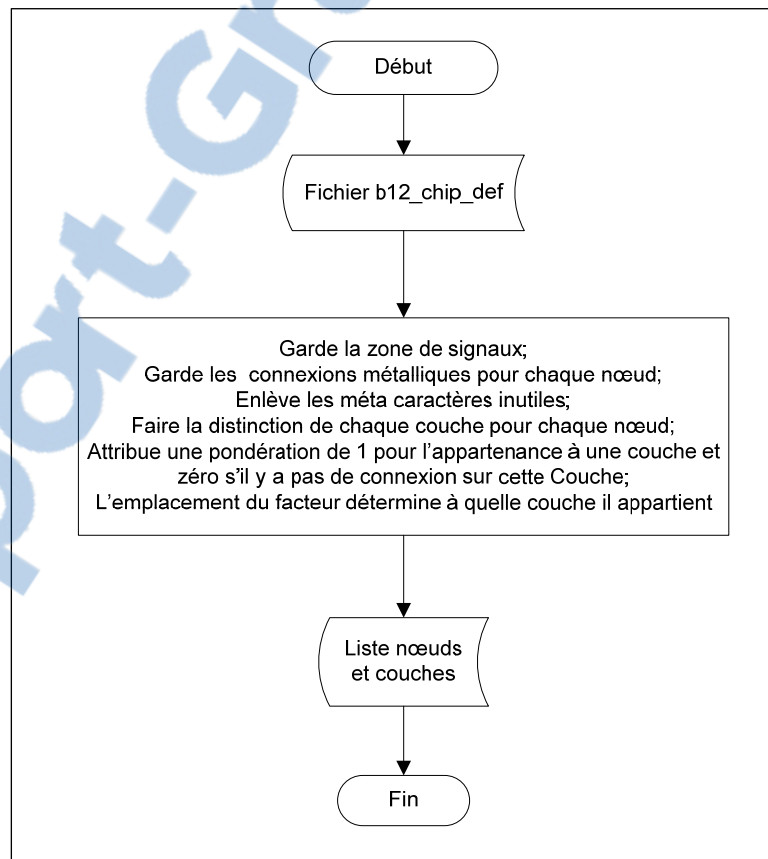


Figure-A VIII-1 Traitement pour trouver les couches métalliques associées aux nœuds Mentor.

Le traitement de la Figure-A VIII.2 assigne aux nœuds Mentor, par l'intermédiaire des nœuds Cadence associés, les couches métalliques trouvées antérieurement, résultant en une nouvelle base de données enrichie. Le mécanisme est basé sur deux boucles, une pour passer à travers tous les nœuds Cadence en les comparant un à un avec les nœuds de Mentor du fichier les contenant (b12\_testability\_ M\_noeuds\_sansSG) jusqu'au moment où il y a correspondance entre les nœuds de Cadence. À ce moment, nous attribuons au nœud les coefficients représentant les couches correspondantes et nous avançons vers le suivant nœud du fichier répertoire des nœuds Mentor. À la fin du traitement, nous vérifions que tous les nœuds ont reçu leurs coefficients des couches métalliques. Pour calculer les signatures désirées, nous appliquons le traitement de la Figure-A VIII.3. La procédure consiste à effectuer les opérations mathématiques indiquées par les équations 6.2 et 6.3. Les traitements sont également effectués en awk.

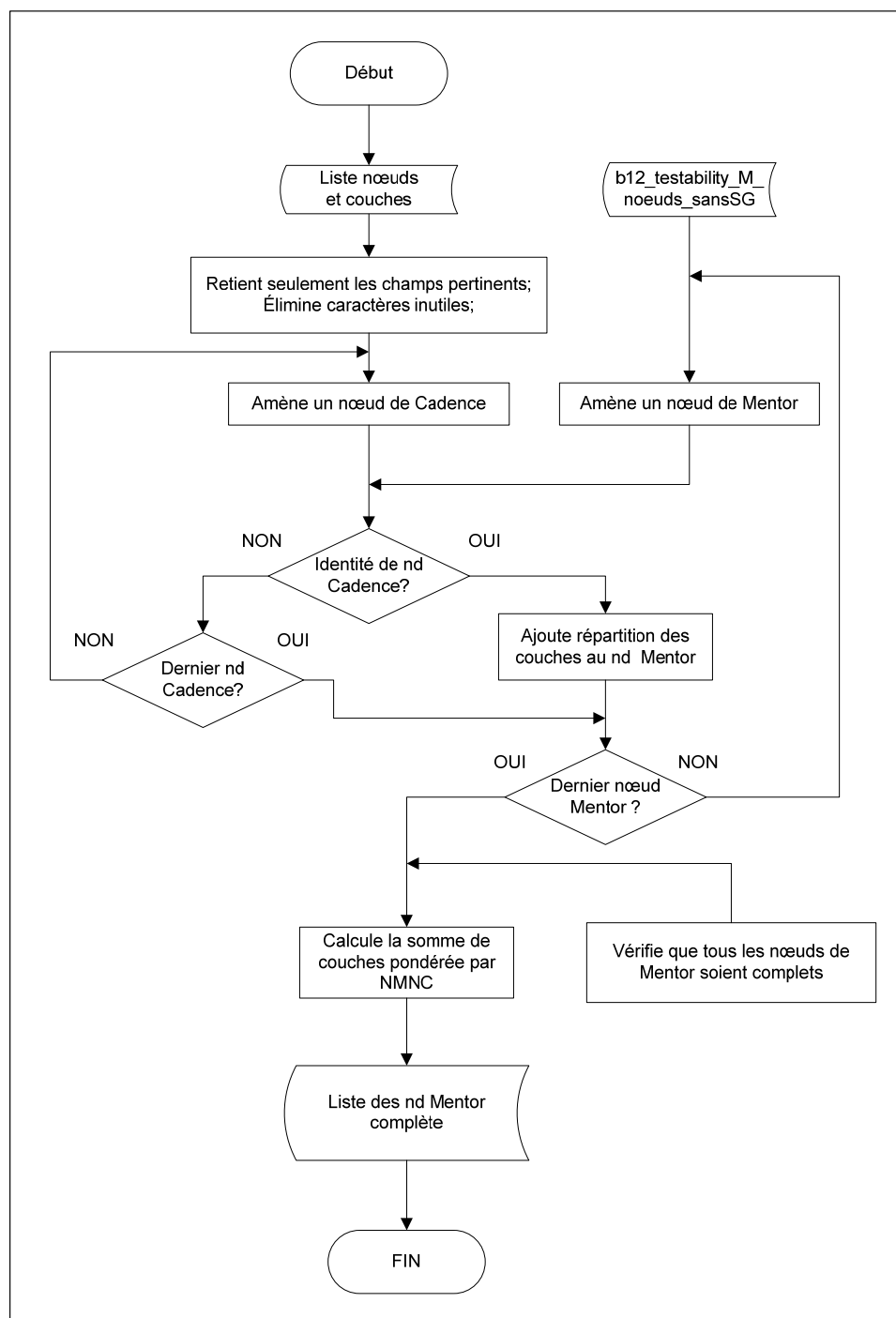


Figure-A VIII-2 Assignment des couches métalliques aux nœuds de Mentor.



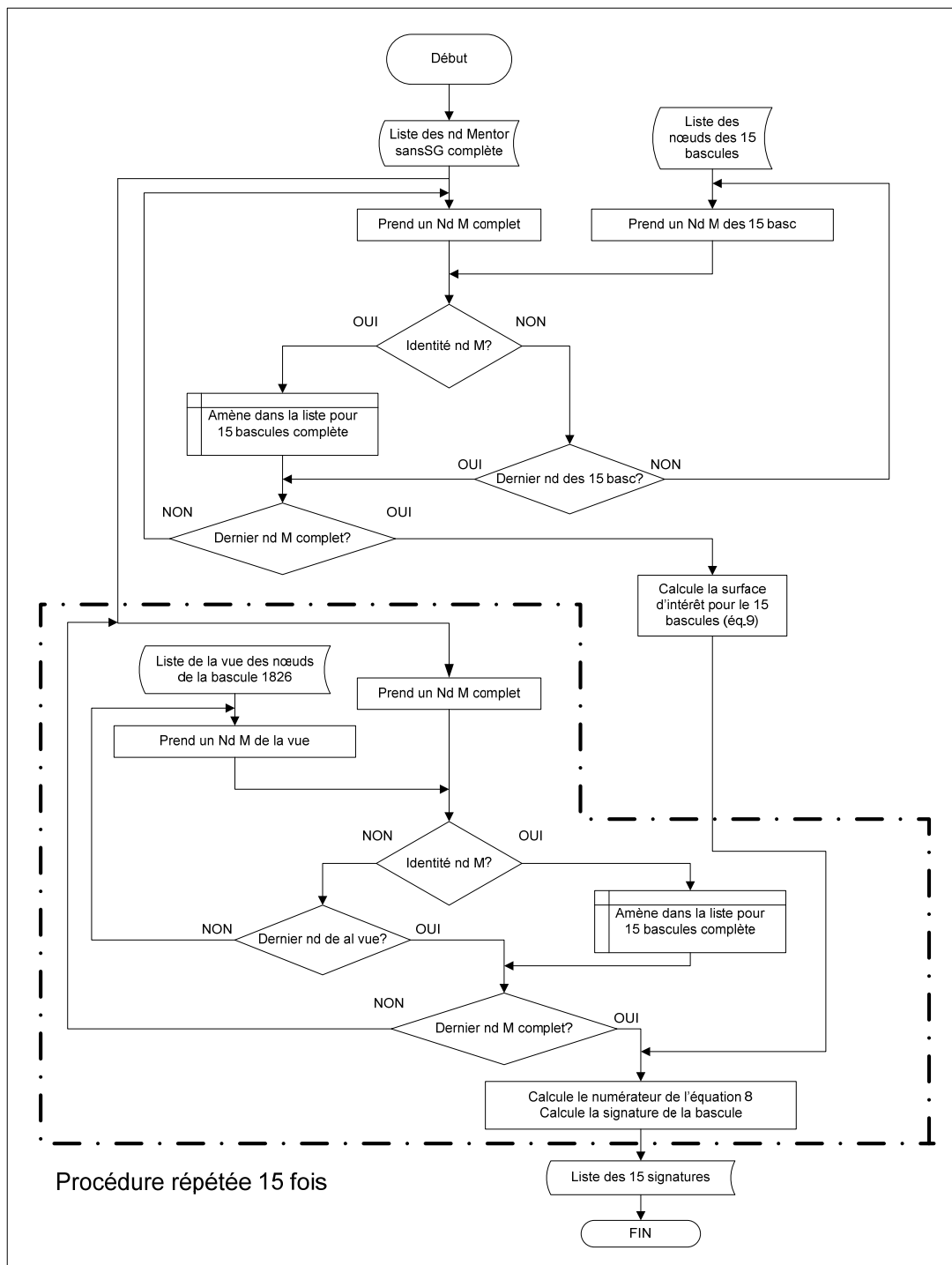


Figure-A VIII-3 Traitement pour construire une signature pondérée par les couches métalliques.



## ANNEXE IX

### EXEMPLE DE CALCUL DE LA SURFACE RÉELLE D'UN POLYGONE DRC

À la figure 7-1, le polygone DRC produit pour la règle 4 (M5.S1) est représenté par le trapèze en jaune. La surface exacte du trapèze est définie :

$$A_{\text{trapèze}} = \frac{(BASE + base) * h}{2}$$

où : *BASE* et *base* sont les bases (ou les côtés parallèles) du trapèze, *BASE* étant le plus long côté, et *h* la distance entre les bases

La surface du trapèze est approximée par le rectangle créé par le côté supérieur (nous prenons toujours le plus grand, *BASE*) et la distance de séparation entre les connections.

L'approximation devient :

$$A_{\text{trapèze}} \approx A_{\text{rectangle}} = BASE * h$$

Pour le polygone de dépassement, nous avons les points correspondant aux coordonnées du trapèze (extraites du fichier de l'annexe XII):

$$A (x=310,31; y=352,94)$$

$$C (x=310,39; y=352,66)$$

$$B (x=314,71; y=352,94)$$

$$D (x=314,63; y=352,66)$$

$$\text{et donc } BASE=AB=x_B-x_A=4,40; \text{ base } =CD=x_D-x_C=4,24; h=y_A-y_C=y_B-y_D=0,28$$

Pour la surface réelle  $Ar_{pij}$  introduit dans la pondération, où  $p=4$  (la séparation sur la couche 5 est R4),  $i$  correspond à la bascule qui couvre ce nœud,  $j$  est la valeur assignée aux nœuds n4844 et n4846 (voir l'annexe XII), nous l'approximons :

$$Ar_{4in4844/n4846} = A_{\text{trapèze}} \approx A_{\text{rectangle}} = (x_B - x_A) * (y_A - y_C) = 4,4 * 0,28 = 1,232$$

Pour estimer l'erreur d'approximation, voici le calcul de la surface exacte du trapèze:

$$A_{\text{trapèze}} = \frac{[(x_B - x_A) + (x_D - x_C)] * (y_A - y_C)}{2} = \frac{(4,40 + 4,24) * 0,28}{2} = 1,2096$$

La différence entre l'approximation et la valeur réelle étant de + 0,0224, l'erreur devient :

$$\frac{0,0224}{1,2096} = 0.018 \text{ (1,8\%).}$$

## **ANNEXE X**

### **DESCRIPTION DÉTAILLÉE DES TRAITEMENTS SPÉCIFIQUES DE LA MÉTHODE 4A**

La feuille de route de cette procédure consiste en plusieurs sous-traitements appliquée à chaque règle. Une première étape établit à partir du fichier résultant du dépassement de règles DRC, les nœuds affectés par le polygone correspondant. Une autre étape assigne à chaque polygone, sa pondération de la surface totale créée par la totalité des polygones qui ont apparus suite aux dépassements de cette règle. Cette pondération est transférée au(x) nœud(s) affecté par ce polygone, en lui attribuant, la pondération du nœud pour cette règle. Ensuite nous faisons une addition des pondérations obtenues par toutes les règles à chaque nœud. Finalement nous nous concentrons sur notre zone d'intérêt, c.à.d. les nœuds couverts par les 15 bascules, pour calculer la signature désirée.

La première étape, appliquée aux polygones DRC des règles de séparation est décrite dans la figure-A X-1. Le traitement s'applique au fichier obtenu de Cadence et consiste à superposer plusieurs modèles des polygones sur les polygones réels des DRC afin de trouver une correspondance au niveau de la forme. Une fois la correspondance établie, nous calculons les extrémités de la connexion. Le calcul est basé sur des relations géométriques. Dans le cas où le semi-polygone contient un côté égal à la barre de règle DRC (Figure-A X-2), l'extrémité de la connexion se trouve sur la médiane retirée d'une valeur égale à la valeur de la règle de conception (Figure-A X-2) et on y attache la fonction de recherche fp. Dans le cas où le côté du semi-polygone est plus grand, le segment d'intérêt se trouve par les projections des côtés parallèles sur la connexion (Figure-A X-3) et on y attache la fonction de recherche fl. Nous prenons toujours le plus petit segment donné par les coordonnées pour être sûr qu'il se trouve sur la connexion en sachant de quel côté du polygone il se trouve par les valeurs des coordonnées du polygone. À noter que tous les polygones ont comme largeur la barre DRC et les côtés parallèles sont celles qui nous intéressent. Pour chaque polygone DRC, la sortie du traitement fournit les coordonnées des deux points et le type de fonction de recherche. Nous

vérifions ensuite qu'aucun polygone n'est perdu. Après cette étape nous appliquons un traitement en commun du fichier résultat et du fichier représentant le circuit réel, le fichier DEF conforme à la figure-A X-4.

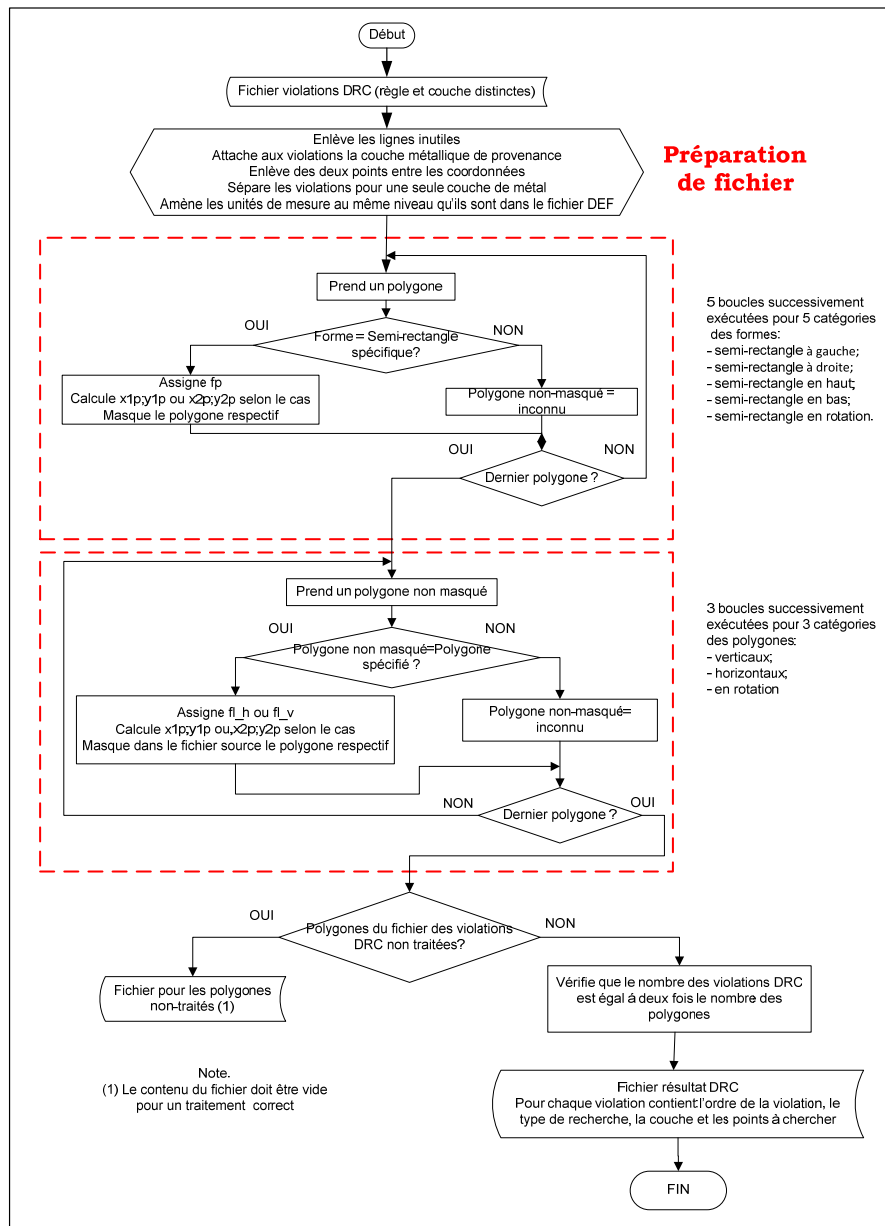


Figure-A X-1 Traitement pour le fichier DRC sortie du dépassement des règles de séparation entre connexions.

Si la fonction attachée est fp, nous cherchons une égalité entre le point et les extrémités des connexions existantes dans le fichier DEF. Dans le cas de la fonction fl, le segment déterminé doit se trouver à l'intérieur des connexions du fichier DEF. Certainement que la fonction fp est un cas particulier de fl, mais la vitesse de recherche des nœuds par fp réduit beaucoup le temps de roulage et justifie sa présence.

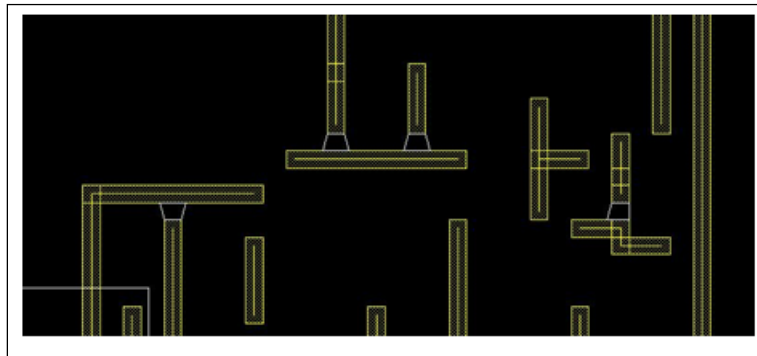


Figure-A X-2 Exemple de semi-polygones DRC pour la séparation sur M2(en blanc).

Les polygones de la figure-A X-2 sont formés au bout d'une connexion caractéristique à la fonction fp. La connexion de M2 en jaune, est l'aire qui entoure la connexion telle qu'elle se présente sur Cadence. Au centre de la connexion, la ligne telle qu'est enregistrée dans le fichier DEF

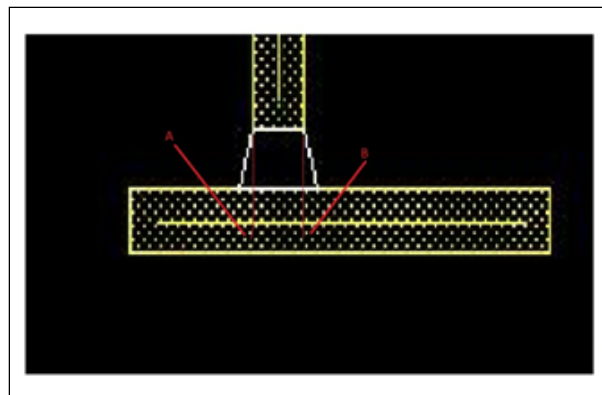


Figure-A X-3 Exemple de semi-polygones DRC pour la séparation sur M2(en blanc).

Les polygones de la figure-A X-2 sont formés au bout d'une connexion caractéristique à la fonction fl. Les projections montrent comment trouver le segment qui nous intéresse.

Puisque les résultats sont gardés dans plusieurs fichiers intermédiaires au moment de leur réunion, est nécessaire la vérification d'existence de ces fichiers pour annuler l'apparition des messages d'erreur qui peuvent arrêter le roulage du programme.

À noter la vérification faite pour montrer que le nombre des nœuds trouvés doit être le double des polygones DRC existantes. Le traitement décrit à la Figure-A X-5, calcule la surface de polygone obtenue suite à la violation de la règle. La couche métallique M1 en particulier subit un traitement supplémentaire. Sur cette couche, en dehors des connexions se trouvent les blocages des cellules qui constituent des zones d'interdiction pour les connexions donc nous avons éliminé leur contribution. En observant que la valeur d'un côté de ces polygones sur la couche M1 est égale à 340, nous avons réussi à enlever tous les polygones superposés aux zones de blocage comptant 1500 polygones. Les polygones restants, au nombre de 15, sont créés par les connexions entre des nœuds des signaux et les fils d'alimentations, chose vérifiée par observation directe sur le dessin des masques. En dehors de l'enlèvement des îlots de métal, le traitement sur la couche M1 est identique à celui décrit pour les autres couches.

Sachant que la largeur d'une connexion est fixe, le traitement conçu pour trouver les nœuds affectés par ce type de violation, se sert de l'idée que le polygone se trouve sur toute la longueur des connexions et que la médiane du polygone est la connexion elle-même. La forme est rectangulaire et la largeur de cette violation n'est pas variable comme pour la règle de séparation. La première étape, décrite dans la Figure-A X-6, calcule pour chaque polygone, à partir de ses coordonnées les dimensions de ses côtés maximales, sa surface et son orientation, verticale ou horizontale. Comme auparavant, l'étiquette d'identification reste le numéro de polygone dans le fichier DRC. Pour la facilité du débogage, chaque groupe des résultats de ces règles sont divisés en deux fichiers conformément à l'orientation des polygones, horizontale ou verticale. Il n'y a aucune différence de procédure entre les deux

types. L'affectation des nœuds aux polygones DRC, présentée dans les Figures-A X-7et X-8 s'applique aux fichiers résultants de la première étape et au fichier de dessin des masques du circuit.

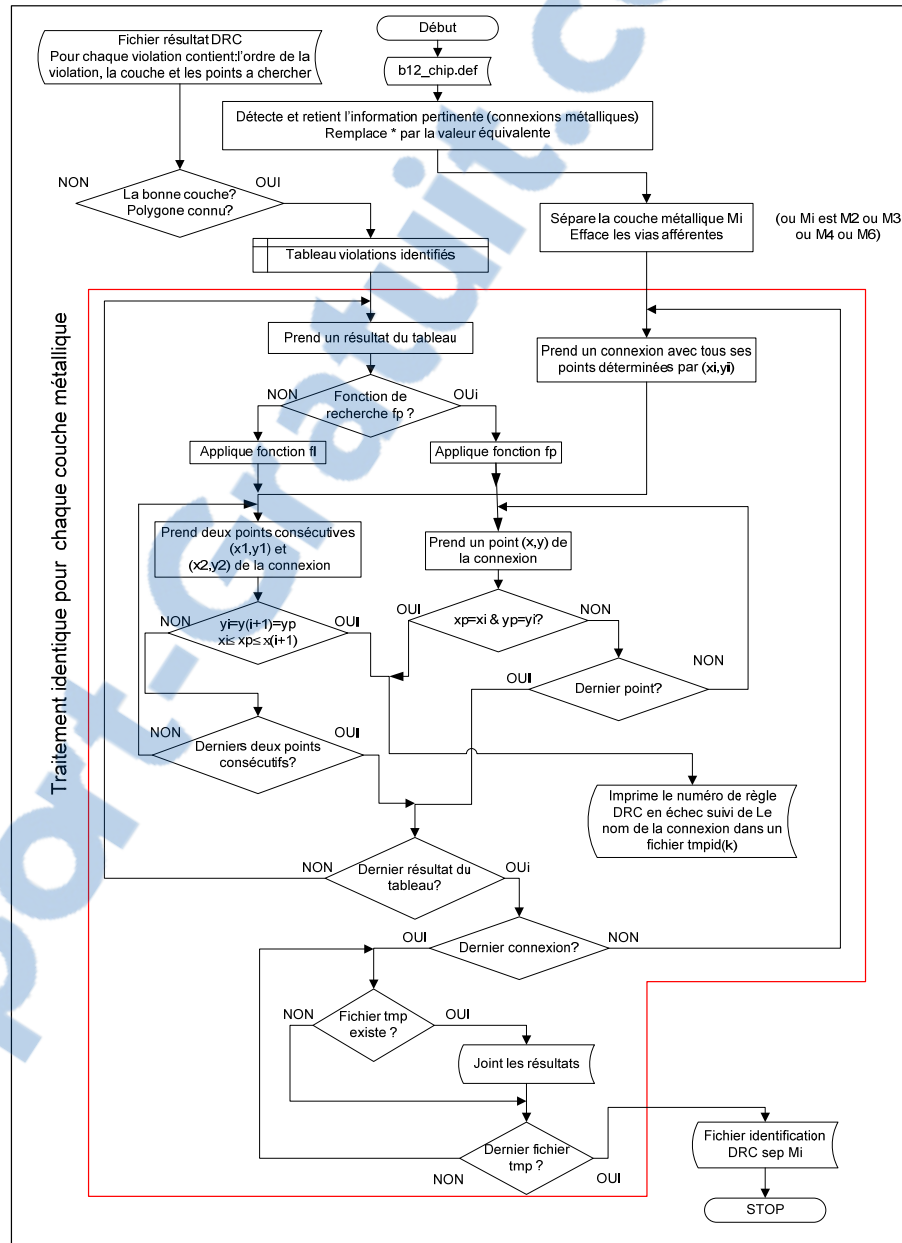


Figure-A X-4 Traitement pour le fichier DEF du circuit et le fichier de sortie du dépassement des règles de séparation entre connexions pour obtenir les nœuds affectés.

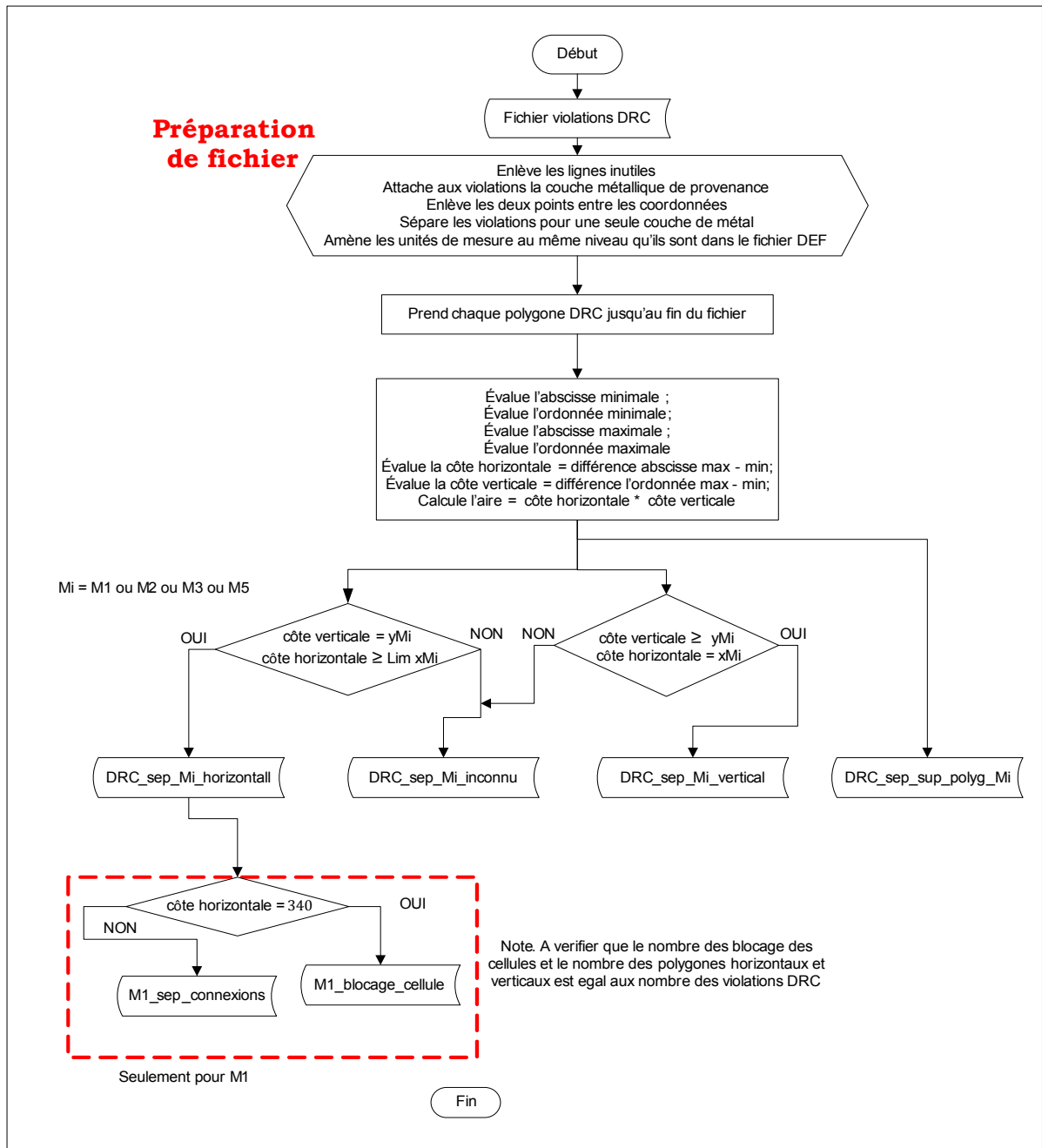


Figure-A X-5 Traitement pour le fichier DRC sortie du dépassement des règles de séparation entre connexions pour calculer la surface des polygones attachées aux nœuds.  
Note. Pour M1 le traitement inclut une dépistage des traces et des blocages des cellules.



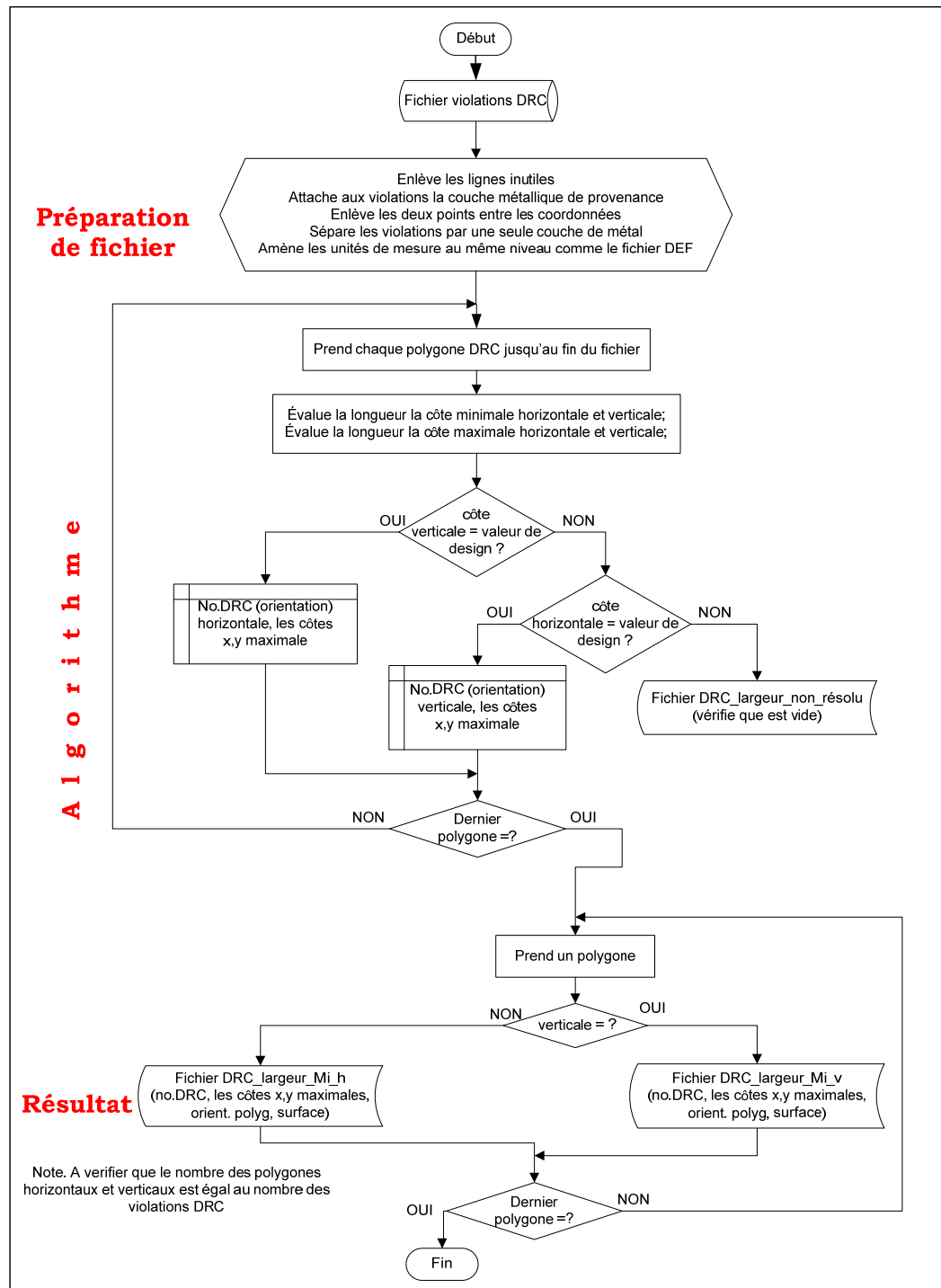


Figure-A X-6 Traitement pour obtenir les noeuds des polygones produites par DRC sur les règles de largeur des connexions.

À remarquer, pour le fichier DEF une opération nouvelle, la recherche des segments des connexions collinaires. Ces sont des segments verticaux ou horizontaux sur lesquelles se retrouvent des vias. À cause des vias, sur le dessin de masque ils sont représentés comme composées de plusieurs morceaux mais dans le mécanisme de Cadence pour les polygones DRC, ils comptent qu'une seule violation. La fonction d'attribution des polygones aux nœuds, voir les Figure-A X-7 et X.6 se fait par le roulage de deux boucles simultanément. Les deux boucles s'activent juste quand les orientations des polygones DRC et du dessin coïncident. Une boucle parcourt chaque segment du dessin des masques et au chaque segment une autre boucle roule avec toutes les médianes attachées aux polygones DRC. Dans le cas d'identité des coordonnées nous assignons au polygone le nœud du segment. Il se peut en final de trouver plusieurs résultats identiques, donc il faut faire une élimination des répétitions. Finalement chaque polygone DRC est associé à un nœud de Cadence. La couche M1 présente aussi des particularités causées par les blocages des cellules. Pour éliminer les zones de blocage, les fichiers des polygones verticaux et horizontaux sont passés par une étape intermédiaire avant d'arriver à la deuxième étape conformément au traitement de la Figure-A X-9. La procédure détecte et élimine tous les petits îlots, avec les deux dimensions en dessous de  $0,24\text{ }\mu\text{m}$  en laissant pour l'identification seulement les traces. Le raisonnement est simple, une connexion doit avoir une dimension beaucoup plus grand que sa largeur donc les petits îlots ne peuvent pas servir comme connexions.

Pour obtenir les dernières 5 règles de la procédure, résultants par les vias du circuit, nous poursuivons le traitement du fichier DEF conforme à la Figure-A X-10. Les vias représentent les trous technologiques pour interconnecter les couches métalliques et dans cette étude leur nombre est assimilé aux violations des règles de conception. La raison pour l'assimilation des vias aux règles de conception est que dans la fabrication des vias simples, demande beaucoup des opérations technologiques qui peuvent créer des pannes systématiques. Chaque type de via est compté sur le dessin des masques et attaché aux nœuds qui le joignent.

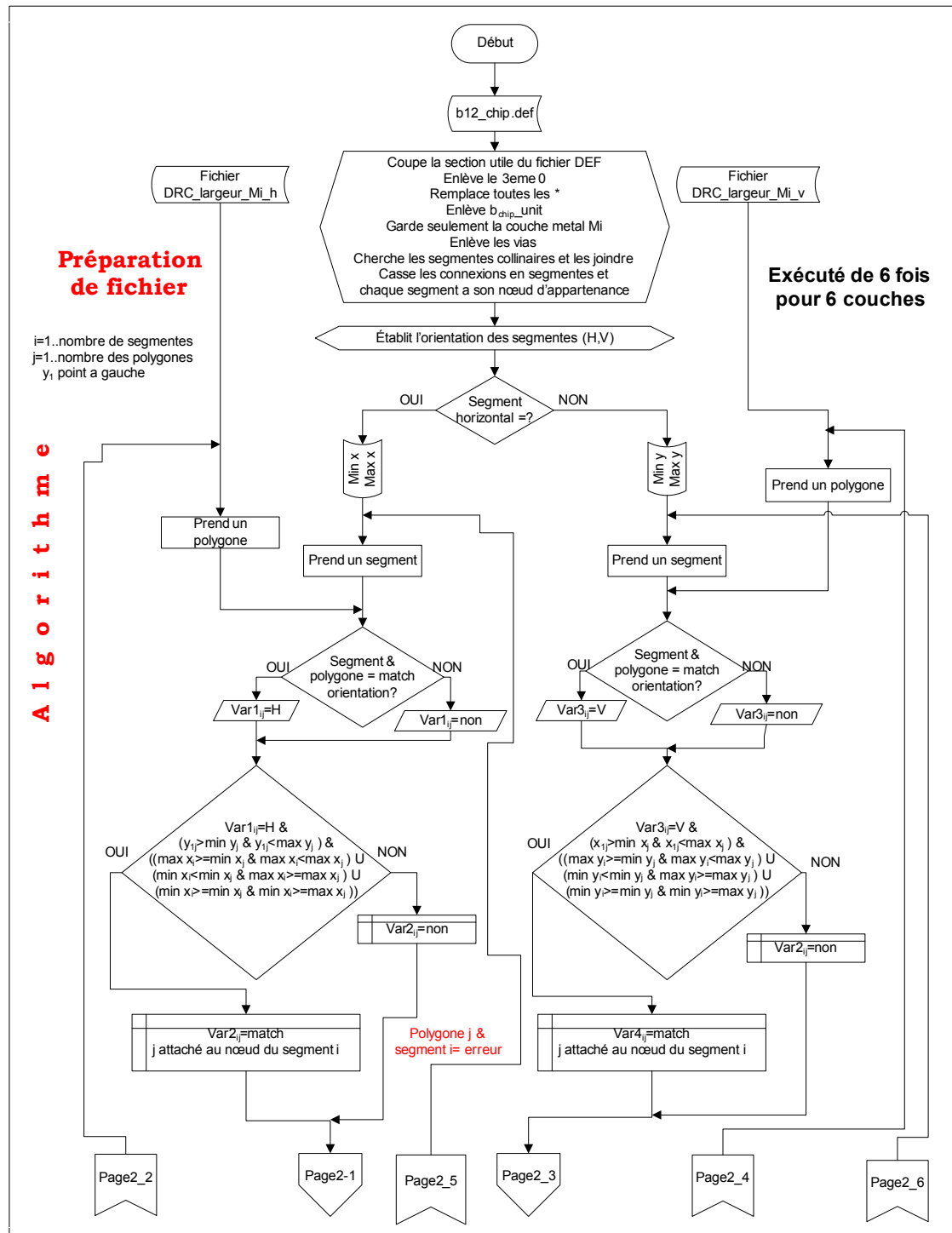


Figure-A X-7 Traitement pour obtenir les nœuds associés aux polygones produites par les règles de largeur des connexions (partie 1).

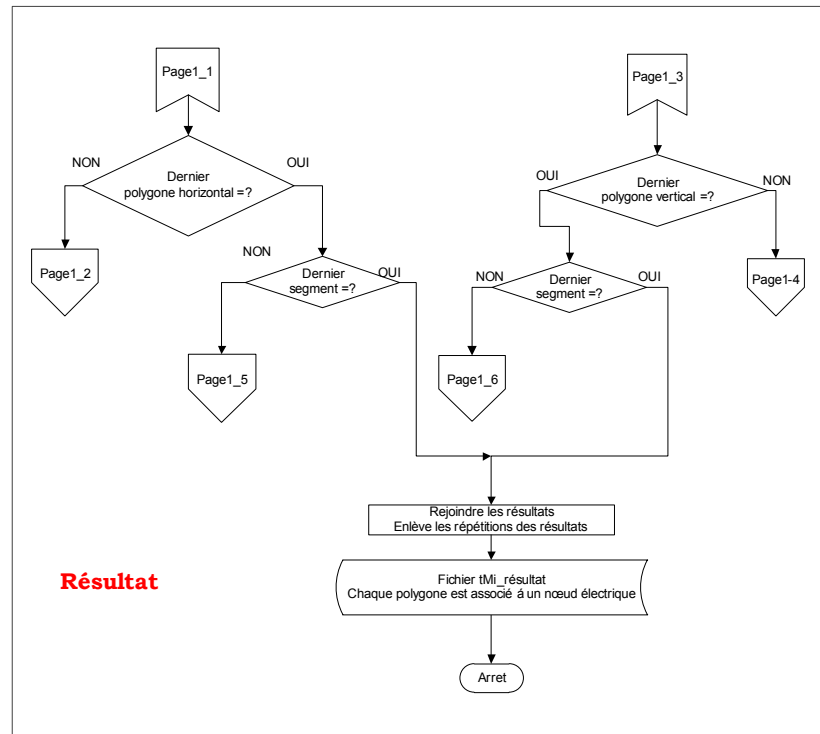


Figure-A X-8 Traitement pour obtenir les nœuds associés aux polygones produites par les règles de largeur des connexions (partie 2).

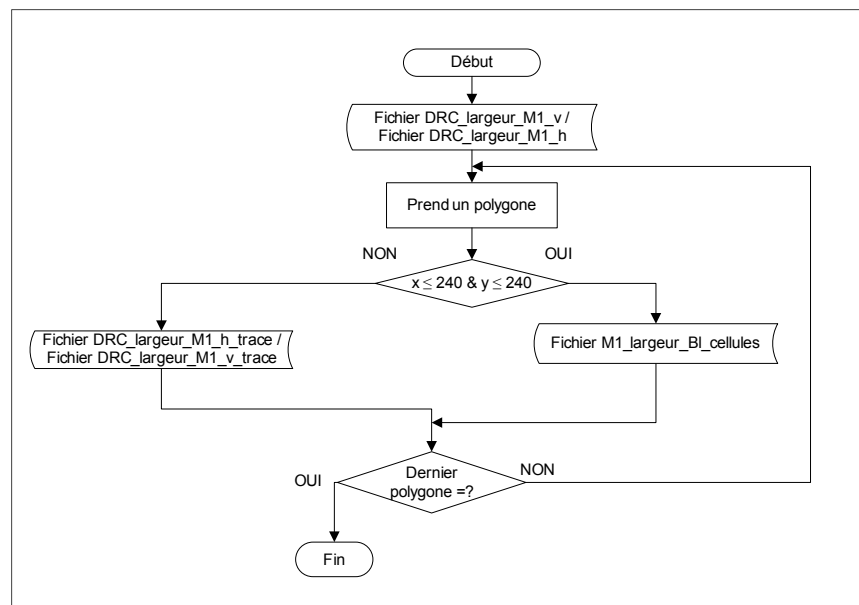


Figure-A X-9 Traitement pour obtenir les blocages des cellules associés aux polygones produits par les règles de largeur des connexions sur M1.

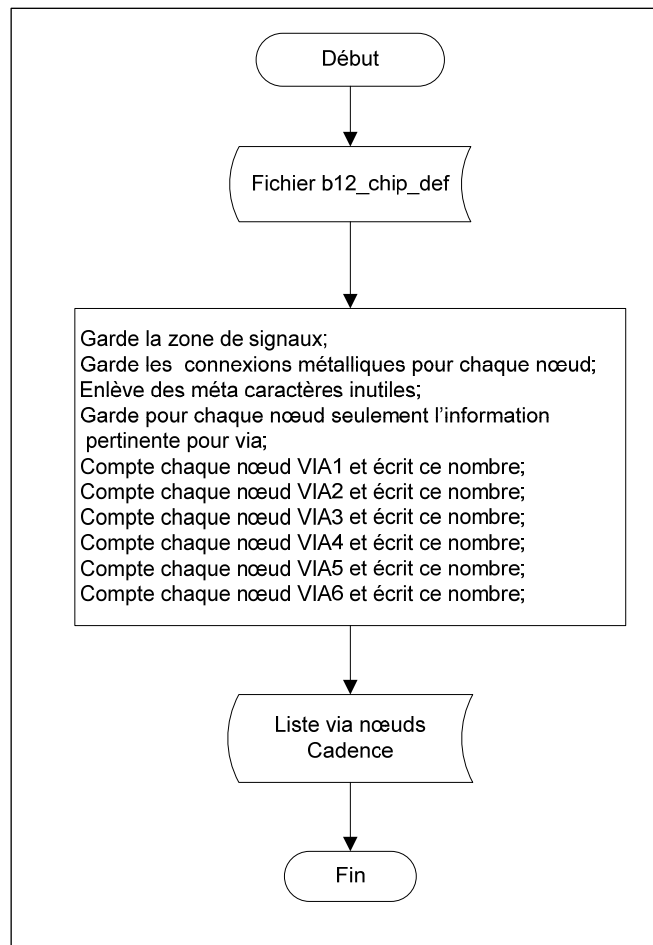


Figure-A X-10 Traitement pour obtenir les noeuds attachés aux vias.

La procédure montrée dans la Figure-A X-11 rejoint les données dans une base unique pour les nœuds de Cadence pour toutes les règles considérées. Nous utilisons une liste des nœuds obtenus au traitement exécuté dans la méthode 1. Cette liste contient tous les nœuds de Cadence du circuit après l'enlèvement des nœuds associés aux signaux globaux. La procédure consiste en trois boucles qui tournant en même temps. La première, à l'extérieur, parcourt toutes les fichiers d'informations associés aux règles de conception (15 au total). Une autre boucle passe à travers tous les nœuds de Cadence, sans signaux globaux. La troisième boucle s'exécute chaque fois qu'un nouveau nœud de Cadence est pris en considération et à ce moment toute la liste des nœuds touchés par une règle est parcourue. Une fois l'identité de ces deux nœuds est constatée, au nœud de Cadence lui est attachée la

surface du polygone trouvé par DRC. Si à la fin de cette dernière boucle, la liste ne donne aucune identification, au nœud de Cadence s'attache un zéro par cette règle. Une fois toutes les trois boucles exécutées nous calculons pour chaque nœud la surface totale des polygones attachés.

Dans la Figure-A X-11 nous présentons l'obtention de la base des données des nœuds Mentor. Nous parcourons la liste des nœuds Mentor du fichier b12\_testability\_M\_sansSG et au chaque nœud Mentor nous lui attachons au moment de l'identité des nœuds de Cadence d'appartenance, la surface des polygones DRC pondérée par NMNC.

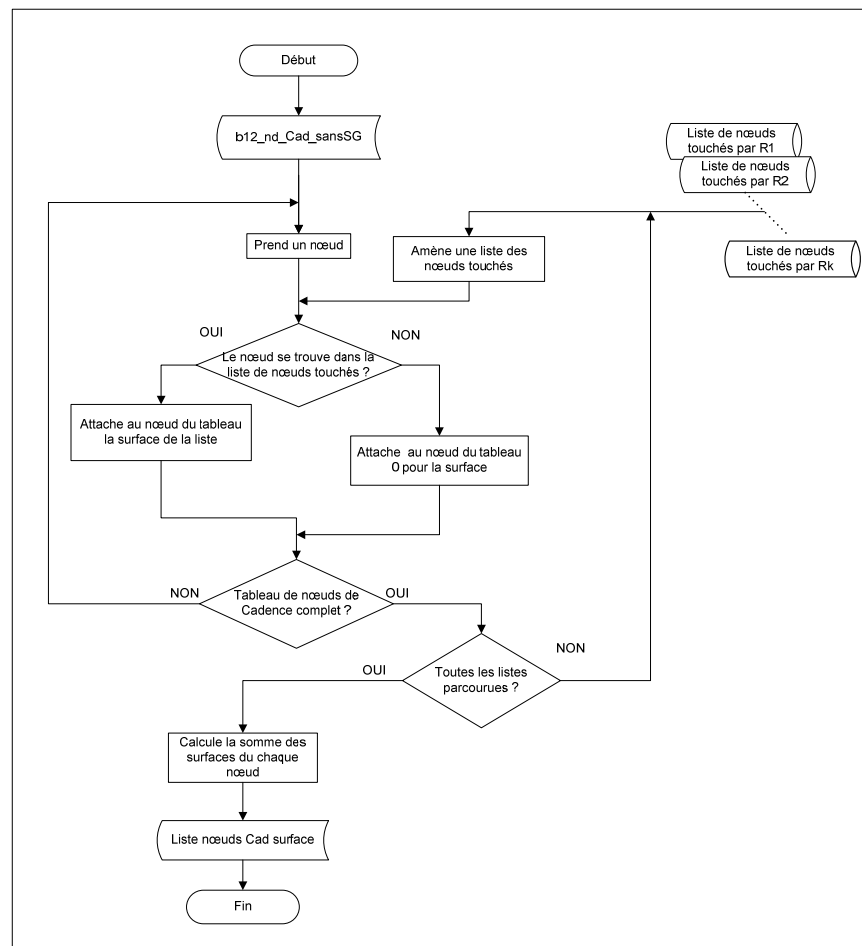


Figure-A X-11 Traitement pour assigner aux nœuds de Cadence les surfaces réelles des polygones.

En suivant la Figure-A X-13, nous obtenons la signature des 15 bascules par le mécanisme déjà connu. Nous calculons pour la totalité des nœuds de la zone d'intérêt, la surface totale pondérée par le coefficient de testabilité.

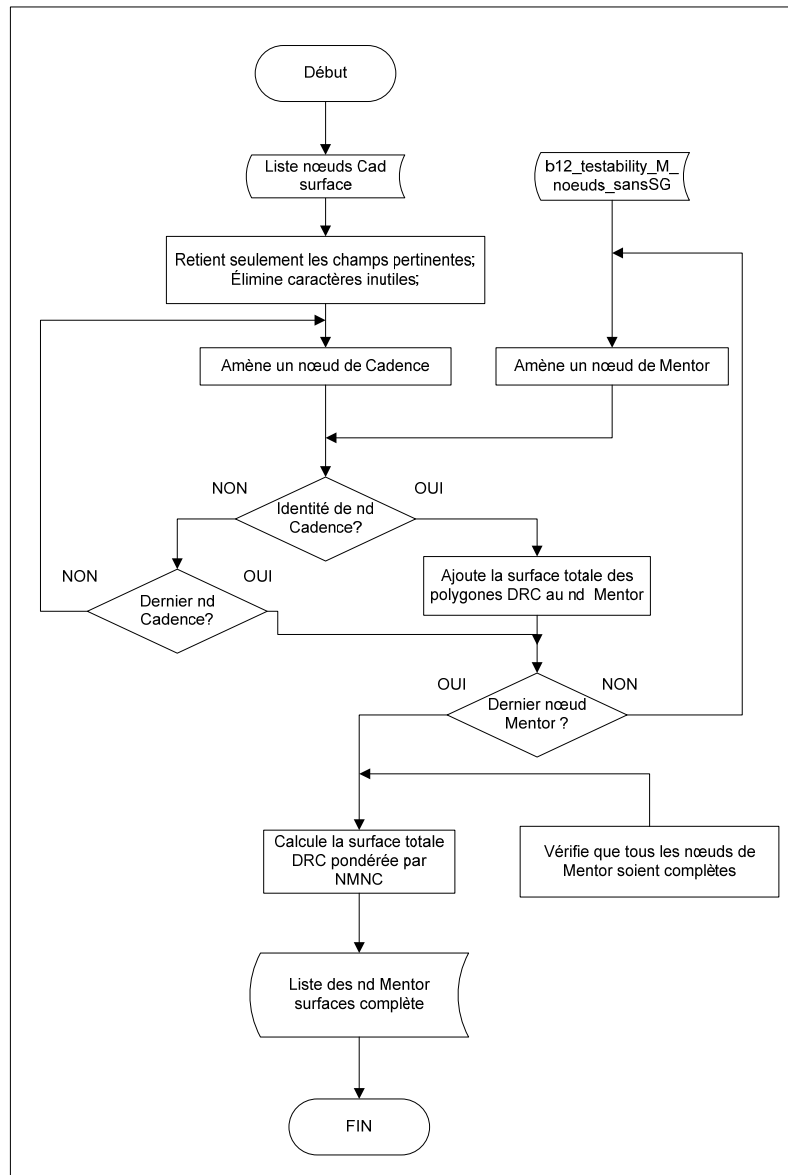


Figure-A X-12 Traitement pour assignées aux nœuds Mentor les surfaces des polygones DRC.

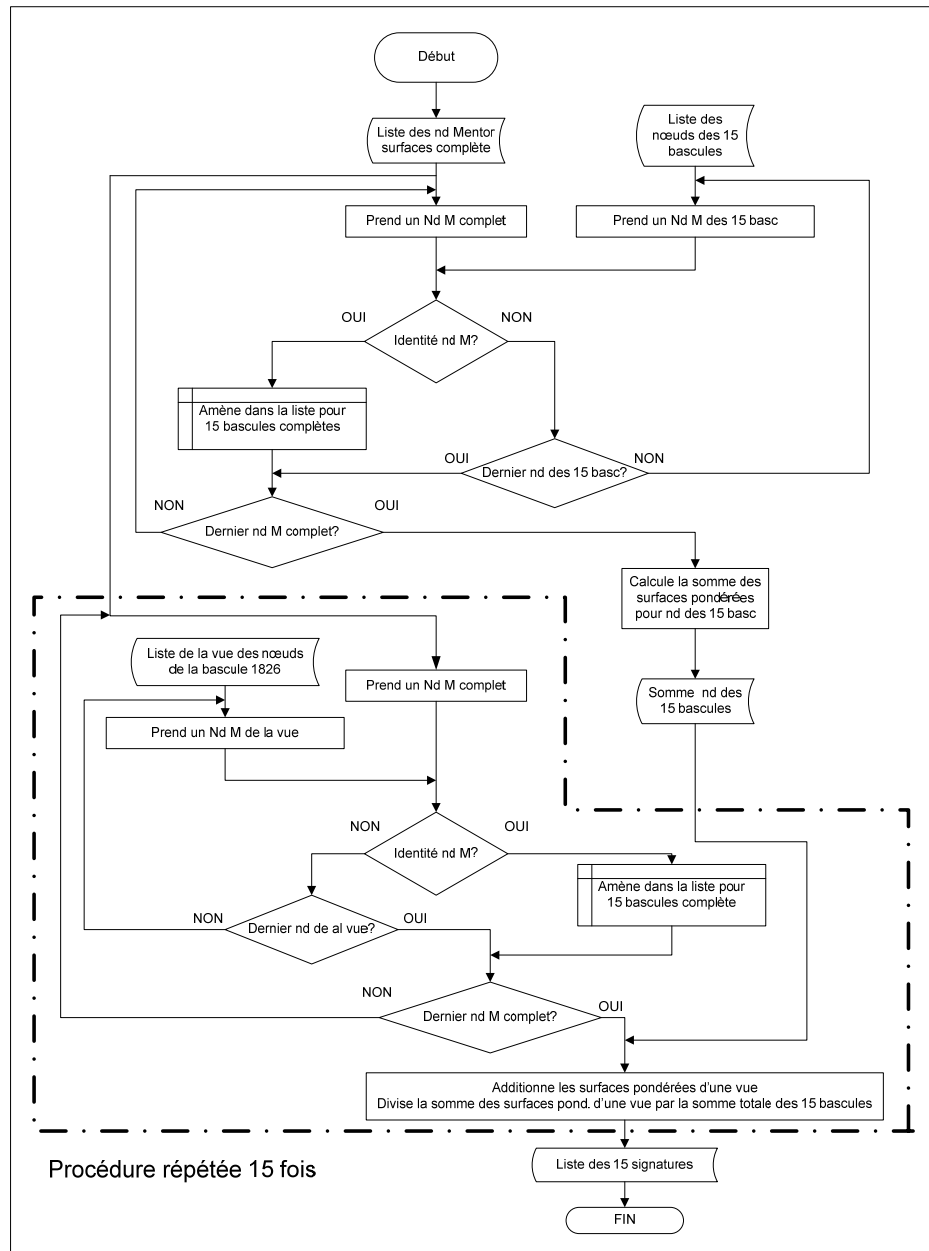


Figure-A X-13 Traitement pour construire une signature.

Finalement, pour chaque vue d'une bascule de la zone d'intérêt nous calculons la somme des surfaces qui appartiennent à cette vue. La signature d'une bascule est le rapporte entre la somme des aires de sa propre vue et la totalité des aires qui appartiennent aux nœuds de la zone d'intérêt.



## ANNEXE XI

### LES RÈGLES DE DESIGN POUR LA FAMILLE CMOSP18 QUI N'ONT PAS DONNÉ DE VIOLATIONS

Appel- lation du fabricant	Définition (Mx définit la couche métallique x)	Règle de dessin de masque (um)	Condition de test pour la signature
Mx.S.2	L'espacement minimal entre deux lignes de métal avec une ou les deux caractéristiques, la largeur et la longueur de la ligne > 10um; l'espace minimale doit être gardée entre une ligne de métal et un petit morceau de métal (<10um) qui est connecté a un métal large a l'intérieur d'une portée de 1.0um (voir D dans la Figure-A X - 1)	M1 $\geq 0,60$ M2 $\geq 0,60$ M3 $\geq 0,60$ M4 $\geq 0,60$ M5 $\geq 0,60$ M6 $\geq 0,60$	La barre = 1,0 n'a pas donné des violations pour aucune couche.
Mx.E.1	L'extension minimale de la région Mx au-delà de la région correspondante à cette couche (CO ou VIAx-1 ou VIA5) (voir C dans la Figure-A X - 2)	M1 $\geq 0,005$ M2 $\geq 0,010$ M3 $\geq 0,010$ M4 $\geq 0,010$ M5 $\geq 0,010$ M6 $\geq 0,090$	La barre = 0,1 n'a pas donné des violations pour aucune couche.
Mx.E.2	L'extension minimale de la région de bout de ligne Mx au-delà de la région correspondante à cette couche (CO ou VIAx-1). Pour les regions CO ou VIAx-1 situées sur un coin de 90 degrés, au moins un coté de l'extension de métal doit être traité comme une bout de ligne et l'autre côté doit se conformer au MxE.1 (voir C1 dans la Figure-A X - 2)	M1 $\geq 0,06$ M2 $\geq 0,06$ M3 $\geq 0,06$ M4 $\geq 0,06$ M5 $\geq 0,06$ M6    N.A.	La barre = 0,1 n'a pas donné des violations pour aucune couche.
Mx.A.1	La surface minimale de la région Mx	M1 $\geq 0,2025 \text{ um}^2$ M2 $\geq 0,2025 \text{ um}^2$ M3 $\geq 0,2025 \text{ um}^2$ M4 $\geq 0,2025 \text{ um}^2$ M5 $\geq 0,2025 \text{ um}^2$ M6 $\geq 0,562 \text{ um}^2$	M1 = $0,21 \text{ um}^2$ M2 = $0,21 \text{ um}^2$ M3 = $0,21 \text{ um}^2$ M4 = $0,21 \text{ um}^2$ M5 = $0,21 \text{ um}^2$ M6 = $0,583 \text{ um}^2$ Pas des violations

Appel- lation du fabricant	Définition (Mx définit la couche métallique x)	Règle de dessin de masque (um)	Condition de test pour la signature
Mx.R.1	Densité minimale de Mx sur toute la puce	M1 $\geq 30\%$ M2 $\geq 30\%$ M3 $\geq 30\%$ M4 $\geq 30\%$ M5 $\geq 30\%$ M6 $\geq 30\%$	Difficile à simuler
AMS.1	<p>L'encoche métallique doit être placée pour permettre le déblocage du stress d'une ligne large de métal. Le métal large est définie <math>&gt; 35\mu\text{m} * 35\mu\text{m}</math> (pour largeur autant que pour longueur) Les pads des connexions sont exclus. (voir Figure-A IX - 3);</p> <p>Le rainurage doit être <math>&gt; 1\mu\text{m}</math> en largeur, la dimension A (comme pour MxS.2) et <math>\geq 10\mu\text{m}</math> en longueur, la dimension B dans la Figure-A 3;</p> <p>L'espacement entre les encoches doit être <math>\geq 10\mu\text{m}</math> (les dimensions C, D);</p> <p>L'espacement encoche-marge de métal doit être <math>\geq 10\mu\text{m}</math> (les dimensions E, F);</p>	M1 = 35 M2 = 35 M3 = 35 M4 = 35 M5 = 35 M6 = 35	Erreurs juste pour la barre de 27,2 um

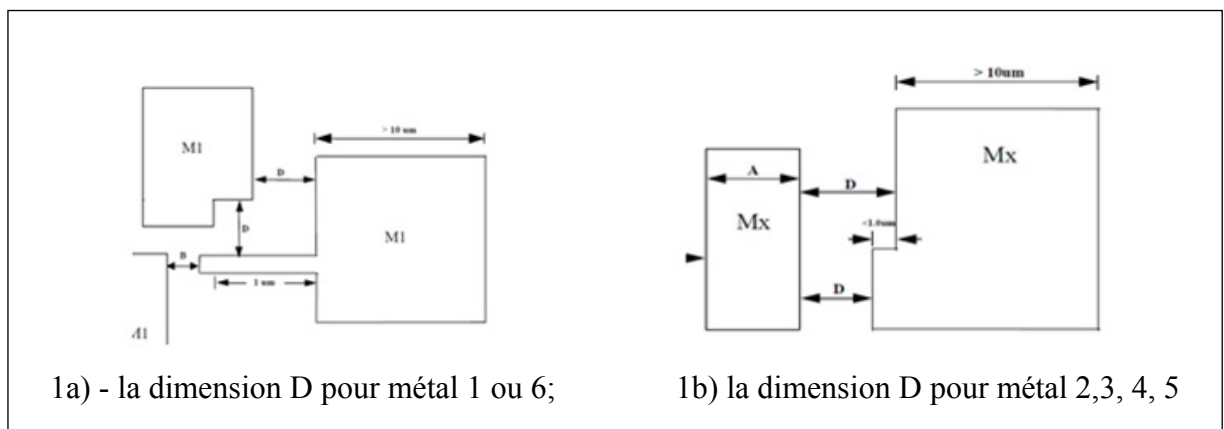
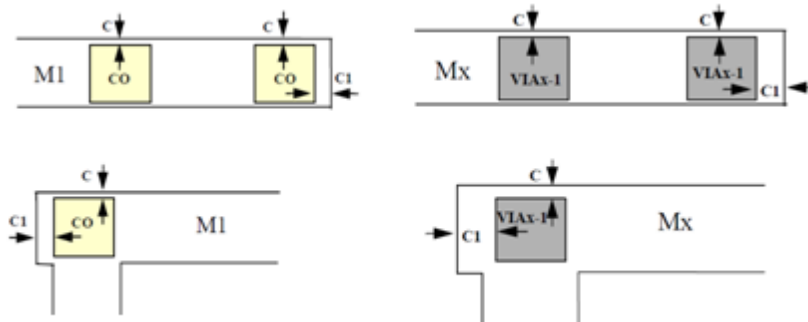
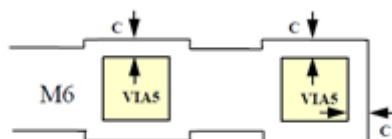


Figure-A XI-1 Définition de la règle MxS.2.



1a) les paramètres C (pour la règle MxE.1) et C1 (pour la règle MxE.1) sur métal 1.

1b) les paramètres C (pour la règle MxE.1) et C1 (pour la règle MxE.1) sur métal 2,3, 4, 5.



1c) les paramètres C, pour la règle MxE.1 sur métal 6.

Figure-A XI-2 Définition de la règle MxE.1 et la règle MxE.2.

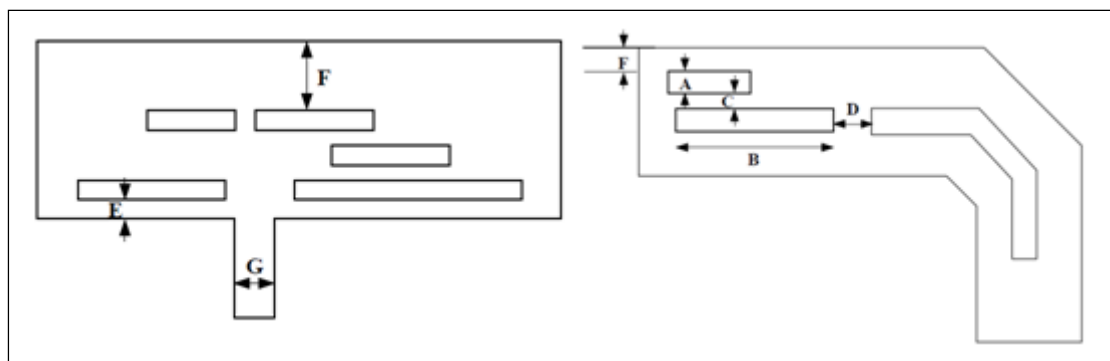


Figure-A XI-3 Guide pour placer des encoches.



## ANNEXE XII

### FICHER DRC OBTENU DE CADENCE PAR LE DÉPASSEMENT DE R4 (LA SÉPARATION ENTRE CONNEXIONS SUR LA COUCHE M5)

\*\*\*\*\*

#### DRC ERROR SUMMARY

Created: Mon Nov 2 13:35:18 2009

library name: design

cell name: b12\_chip

view name: autolayout

Last checked: Mon Nov 2 13:35:18 2009

\*\*\*\*\*

\*\*\*\*\*

#### Cell Error Summary (in error count order)

\*\*\*\*\*

Error Count	Cell Name	View Name	Libraray Name
-------------	-----------	-----------	---------------

1. 43 b12\_chip autolayout design

Total error count is 43

\*\*\*\*\*

#### DRC ERROR SUMMARY

Created: Mon Nov 2 13:35:18 2009

library name: design

cell name: b12\_chip

view name: autolayout

Last checked: Mon Nov 2 13:35:18 2009

Error Count: 43

\*\*\*\*\*

RULE LABEL: drc(metal5 (sep < 0.29))

ERROR COUNT: 43

- 1 281.94:342.3 282.01:342.02 349.69:342.02 349.61:342.3
- 2 297.12:358.26 302.75:358.26 302.83:358.54 297.19:358.54
- 3 299.1:358.82 316.69:358.82 316.61:359.1 299.17:359.1

4 302.39:359.38 316.61:359.38 316.69:359.66 302.47:359.66  
5 303.71:353.22 332.53:353.22 332.45:353.5 303.79:353.5  
6 304.38:357.14 306.71:357.14 306.79:357.42 304.45:357.42  
7 305.04:357.7 316.69:357.7 316.61:357.98 305.11:357.98  
8 305.04:358.54 305.11:358.26 316.61:358.26 316.69:358.54  
9 307.68:378.14 307.75:377.86 322.55:377.86 322.63:378.14  
10 308.33:359.94 319.91:359.94 319.99:360.22 308.41:360.22  
11 308.33:360.78 308.41:360.5 326.51:360.5 326.59:360.78  
12 309.65:376.18 331.87:376.18 331.79:376.46 309.73:376.46  
13 **310.31:352.94 310.39:352.66 314.63:352.66 314.71:352.94**  
14 310.31:376.74 331.79:376.74 331.87:377.02 310.39:377.02  
15 310.31:377.58 310.39:377.3 322.63:377.3 322.55:377.58  
16 312.3:366.66 327.17:366.66 327.25:366.94 312.37:366.94  
17 314.27:372.54 314.35:372.26 317.35:372.26 317.27:372.54  
18 314.94:352.94 315.01:352.66 323.87:352.66 323.95:352.94  
19 316.92:366.38 316.99:366.1 327.25:366.1 327.17:366.38  
20 318.89:369.74 318.97:369.46 322.55:369.46 322.63:369.74  
21 320.21:351.54 325.85:351.54 325.93:351.82 320.29:351.82  
22 320.21:352.38 320.29:352.1 323.95:352.1 323.87:352.38  
23 321.54:378.42 327.83:378.42 327.91:378.7 321.61:378.7  
24 321.54:379.26 321.61:378.98 336.49:378.98 336.41:379.26  
25 322.86:368.34 326.59:368.34 326.51:368.62 322.93:368.62  
26 327.48:354.9 357.53:354.9 357.61:355.18 327.55:355.18  
27 327.48:355.74 327.55:355.46 331.21:355.46 331.13:355.74  
28 327.48:366.1 340.37:366.1 340.45:366.38 327.55:366.38  
29 328.13:356.02 331.13:356.02 331.21:356.3 328.21:356.3  
30 328.13:366.66 341.03:366.66 341.11:366.94 328.21:366.94  
31 329.45:356.58 339.05:356.58 339.13:356.86 329.53:356.86  
32 330.77:372.26 344.99:372.26 345.07:372.54 330.85:372.54

33 332.1:355.46 347.71:355.46 347.63:355.74 332.17:355.74  
 34 332.1:356.3 332.17:356.02 339.13:356.02 339.05:356.3  
 35 332.75:357.14 337.15:357.14 337.07:357.42 332.83:357.42  
 36 334.74:377.02 334.81:376.74 339.79:376.74 339.71:377.02  
 37 335.39:378.7 335.47:378.42 345.65:378.42 345.73:378.7  
 38 337.38:354.62 337.45:354.34 339.05:354.34 339.13:354.62  
 39 337.38:377.3 339.71:377.3 339.79:377.58 337.45:377.58  
 40 337.38:378.14 337.45:377.86 345.73:377.86 345.65:378.14  
 41 386.21:388.78 386.29:388.5 387.23:388.5 387.31:388.78  
 42 388.19:384.86 388.27:384.58 389.29:384.58 389.21:384.86  
 43 402.71:394.38 402.79:394.1 410.33:394.1 410.41:394.38

#### Résultat du traitement du fichier antérieur pour obtenir les nœuds affectés de DRC

# <sup>1</sup>	NŒUD CADENCE	No. DRC
1	n37	42@
2	b12_unit\address\[0\]	43@
3	b12_unit\n4848	1@
4	b12_unit\data_in\[0\]	41@
5	b12_unit\n4886	4@
6	b12_unit\n4812	26@
7	b12_unit\n4885	14@
8	b12_unit\n4885	19@
9	b12_unit\n4885	28@
10	b12_unit\n5122	21@
11	b12_unit\wr2086	29@
12	b12_unit\nloss2059	2@
13	b12_unit\nloss2059	10@
14	b12_unit\n4912	30@
15	b12_unit\n4913	17@
16	b12_unit\n4913	31@
17	b12_unit\n4913	32@
18	b12_unit\n4865	34@
19	b12_unit\n4907	9@
20	b12_unit\n4907	20@

# <sup>1</sup>	NCEUD CADENCE	No. DRC
21	b12_unit\vn4844	5@
22	b12_unit\vn4868	38@
23	b12_unit\vn4846	12@
24	b12_unit\vn4846	13@
25	b12_unit\vn4821	7@
26	b12_unit\vn4857	27@
27	b12_unit\vn4857	33@
28	b12_unit\vn4910	18@
29	b12_unit\vn4829	8@
30	b12_unit\vn4893	24@
31	b12_unit\vn4990	37@
32	b12_unit\vn4888	3@
33	b12_unit\vn4861	22@
34	b12_unit\vn4861	25@
35	b12_unit\vn4869	16@
36	b12_unit\vn5014	40@
37	b12_unit\vn4823	36@
38	b12_unit\vn4864	6@
39	b12_unit\vn4864	11@
40	b12_unit\vn4864	23@
41	b12_unit\vn4878	35@
42	b12_unit\vn5011	15@
43	b12_unit\vn5011	39@
44	scan_en_top	42@
45	b12_unit\timebase\[5\]	11@
46	b12_unit\address\[4\]	41@
47	b12_unit\memory\[1\]\[0\]	43@
48	b12_unit\vn4886	3@
49	b12_unit\vn4812	38@
50	b12_unit\vn4885	12@
51	b12_unit\play	1@
52	b12_unit\wr2086	27@
53	b12_unit\vnloss2059	4@
54	b12_unit\vn4912	16@
55	b12_unit\vn4912	28@
56	b12_unit\vn4913	29@
57	b12_unit\vn4913	34@
58	b12_unit\vn4865	5@



# <sup>1</sup>	NœUD CADENCE	No. DRC
59	b12_unit\N4865	33@
60	b12_unit\N4907	15@
61	b12_unit\N4844	13@
62	b12_unit\N4844	18@
63	b12_unit\N4844	35@
64	b12_unit\N4821	6@
65	b12_unit\N4857	26@
66	b12_unit\N4910	22@
67	b12_unit\N4915	24@
68	b12_unit\N4829	7@
69	b12_unit\N4882	25@
70	b12_unit\N4893	23@
71	b12_unit\N4893	37@
72	b12_unit\N4990	40@
73	b12_unit\N4888	2@
74	b12_unit\N4888	8@
75	b12_unit\N4861	21@
76	b12_unit\N4916	30@
77	b12_unit\N4869	19@
78	b12_unit\N5014	39@
79	b12_unit\N4914	20@
80	b12_unit\N4864	9@
81	b12_unit\N4864	10@
82	b12_unit\N5095	17@
83	b12_unit\N4878	31@
84	b12_unit\N5011	14@
85	b12_unit\N5011	36@
86	b12_unit\N4989	32@

Note 1. Le nombre de nœuds trouvés est le double du nombre des polygones DRC obtenus (règle de séparation sur la couche métallique 5).

## **ANNEXE XIII**

### **DESCRIPTION DÉTAILLÉE DES TRAITEMENTS SPÉCIFIQUES DE LA MÉTHODE 4B**

Le traitement aux Figure-As XIII-1, XIII-2 et XIII-3 font la conversion entre le résultat de la simulation Fastscan, où l'information se présente sous une forme positionnelle et le format tableau (voir l'annexe XVI). En première phase, nous faisons l'identification entre la position du bit qui rapporte l'état logique erroné dans la séquence de la sortie et le nom de la bascule intéressante pour notre signature. C'est une opération inverse à celle effectuée par l'introduction du fichier antérieur, grâce à deux boucles simultanées. L'étape suivante du traitement consiste en plusieurs boucles qui comptent les bits de 1 et 0 sur les réponses de la simulation. En sachant que les bits indiqués sont seulement en cas de différence entre les états logiques corrects et erronés, nous comptons la première rencontre d'un tel bit jusqu'au fin de la séquence. Étant donné que nous ne connaissons pas le nombre des bascules impliquées dans la détection d'une panne, la recherche doit se répéter jusqu'à il ne reste plus aucun bit à compter sur la ligne. Ensuite une fois déterminée la position de tous les bits 1 et 0 sur la ligne, il faut associer cette position au nom de la bascule qui rapporte le bit. Finalement nous faisons une conversion de noms de Cadence vers Mentor pour être capable de faire la comparaison avec les autres méthodes. L'information de ce fichier, équivalent au fichier fourni par la simulation de Fastscan, contient toutes les pannes détectables par les 15 bascules intéressantes. Les 15 bascules détectent 5342 pannes, affectant 593 nœuds. Il y a parmi ces 5342 pannes, 125 qui proviennent de 31 nœuds internes et qui seront filtrés. Notons que les 593 nœuds sont en effet tous pris en considération par la commande "analyse des pannes". Cette observation permet d'utiliser dans l'opération d'attachement des surfaces des polygones le même fichier que celui de la méthode 4a.

La détection des pannes est faite dans la plupart des cas par une seule bascule (4107 pannes) mais aussi par plusieurs bascules simultanément (allant jusqu'à 7 bascules). Le fichier obtenu sert comme un nouveau point de départ pour calculer la testabilité des nœuds, comme il est décrit dans la Figure-A XIII-5.

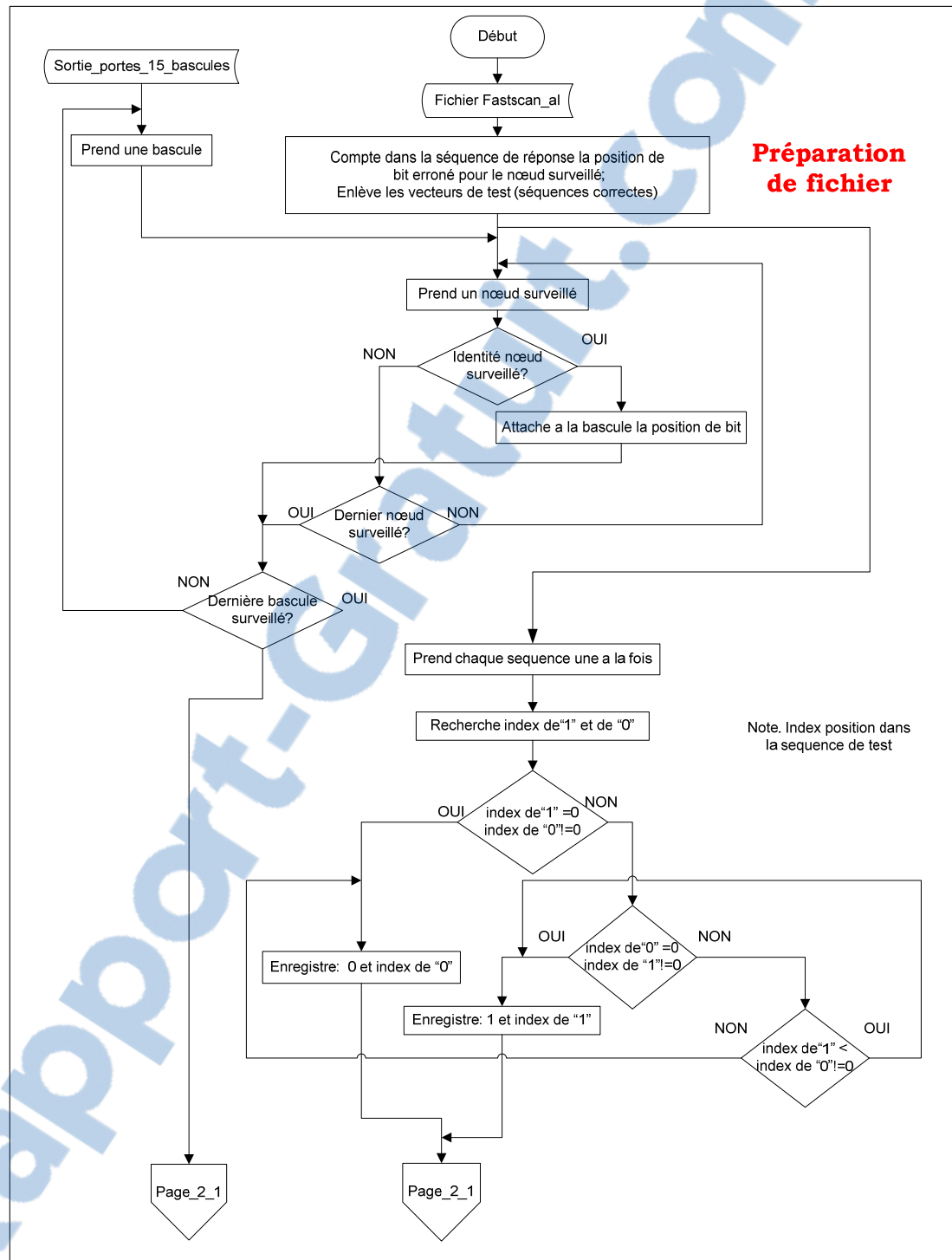


Figure-A XIII-1 Traitement de conversion du fichier résultat du Fastscan (partie 1).

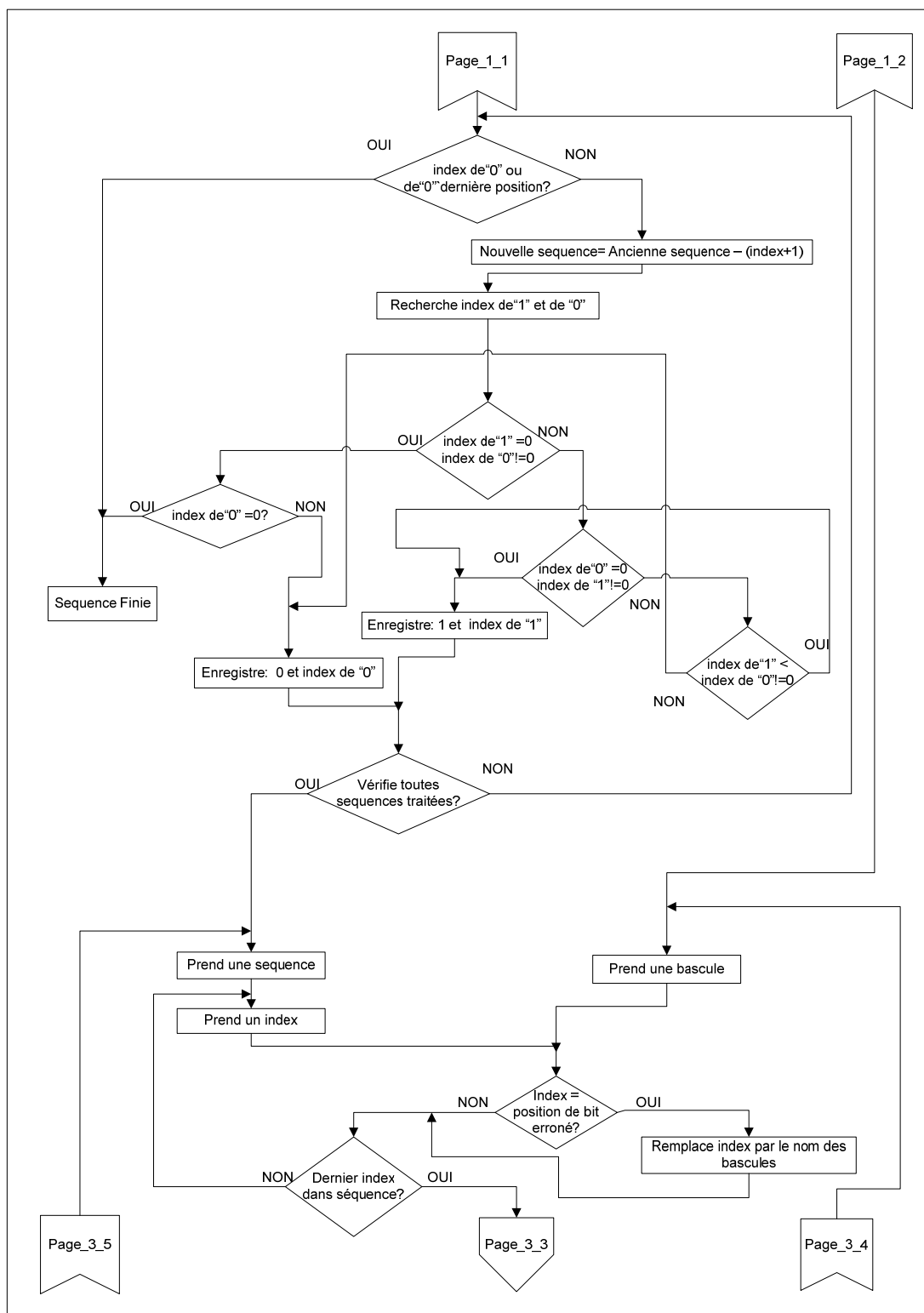


Figure-A XIII-2 Traitement de conversion du fichier résultat du Fastscan (partie 2).

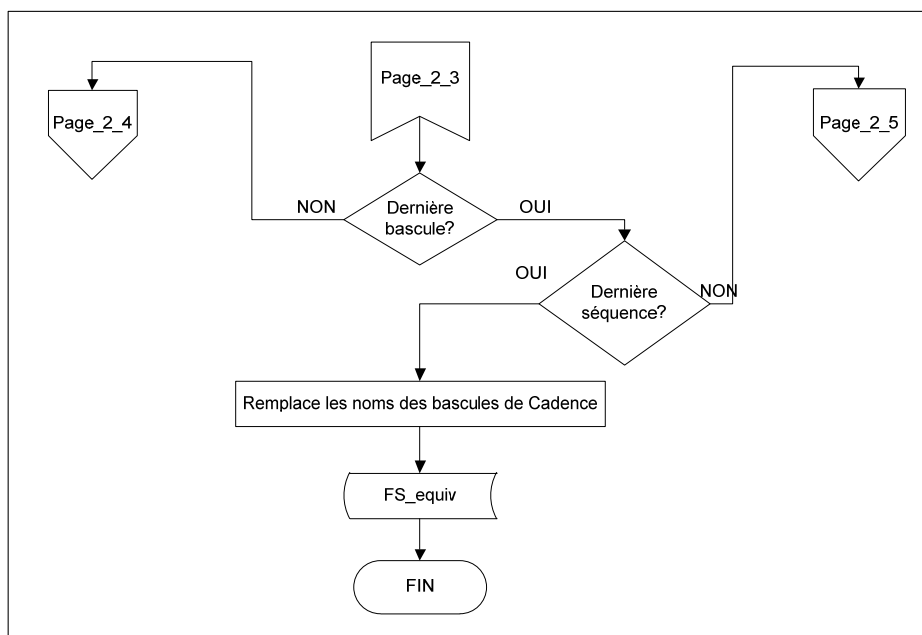


Figure-A XIII-3 Traitement de conversion du fichier résultat du Fastscan (partie 3).

Nous partons du fichier équivalent de celui qui a été fourni par Fastscan, dans lequel nous séparons les nœuds pour faire une liste complète des nœuds Mentor mis en évidence par cette simulation. Si les nœuds sont rencontrés deux fois dans la réponse de Fastscan, pour la panne collé à 1 et à 0 nous lui accordons une testabilité de 1 et si nous le rencontrons juste une fois, la testabilité sera 0,5. Ensuite, un autre système de deux boucles simultanées va attribuer le coefficient NMNC et la surface des polygones DRC qui leur sont associés. Au terme de ce traitement, nous obtenons un fichier complet avec tous les nœuds détectés par les 15 bascules. Dans le traitement de la Figure-A XIII-4, nous établissons un fichier équivalent au fichier "vue" des méthodes précédentes qui contient les nœuds détectables ou couverts par chaque bascule à la commande "ajout des listes". En première phase, nous partons du fichier de réponse du Fastscan, *FS\_equiv*, en faisant une interrogation sur chacune de ses séquences. Si la séquence contient un nom de bascule qui nous intéresse, nous allons l'amener dans un fichier temporaire. Chaque fichier temporaire est introduit dans une autre boucle qui le parcourt nœud par nœud. Simultanément une autre boucle parcourt le fichier construit dans l'étape précédente et qui contient tous les nœuds des 15 bascules.



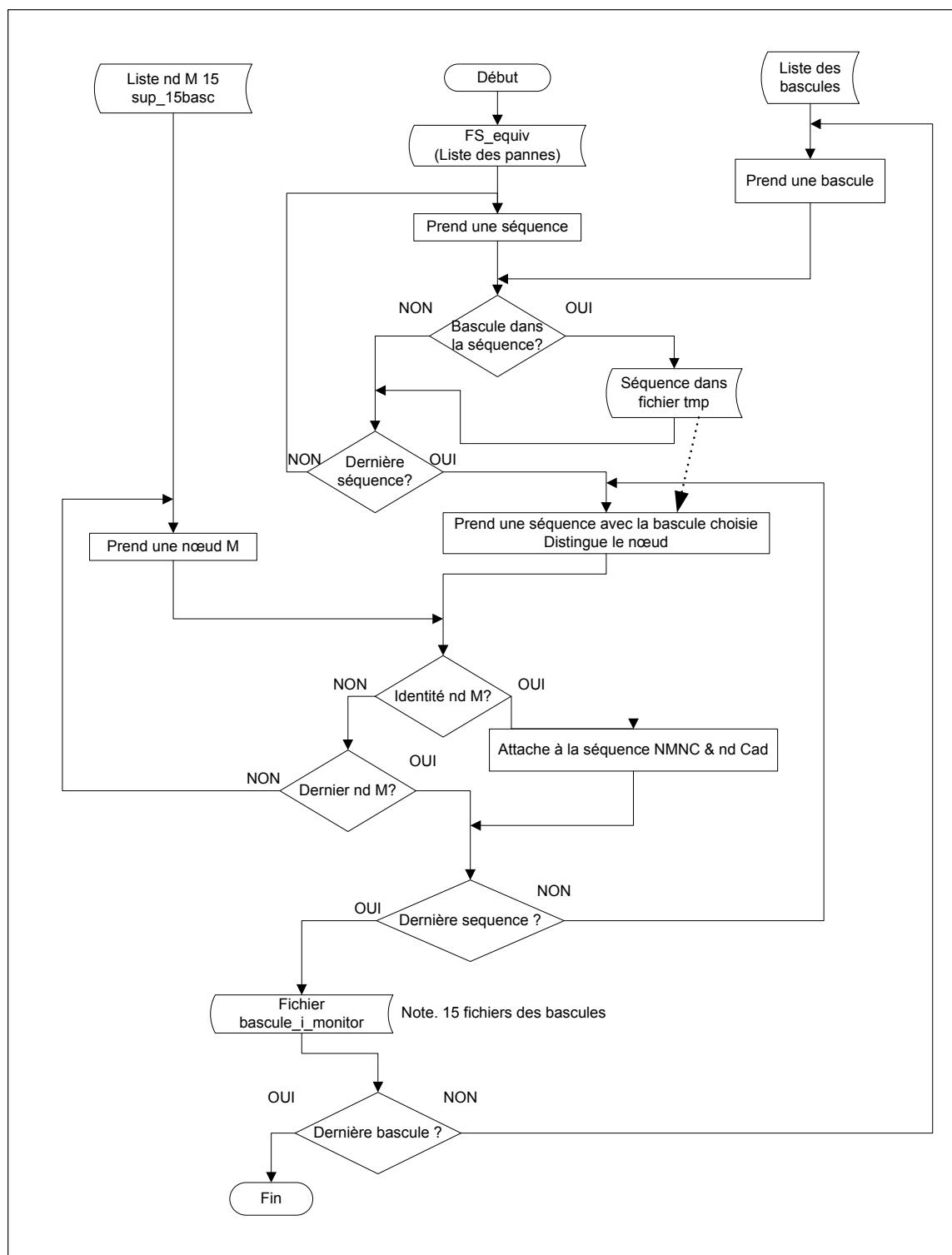


Figure-A XIII-4 Traitement pour établir les nœuds couverts de chaque bascule.

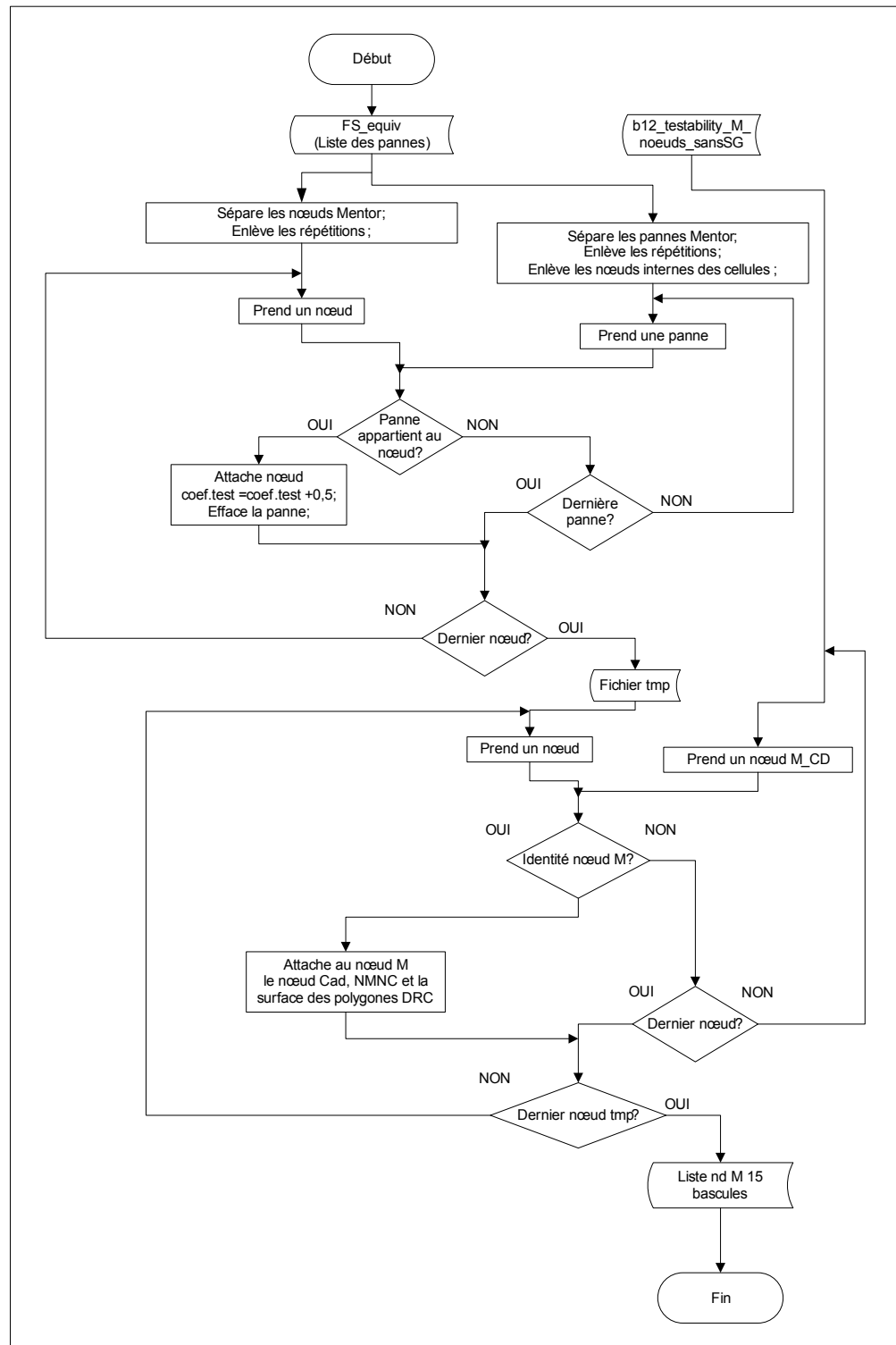


Figure-A XIII-5 Traitement pour calculer la testabilité, les surfaces des polygones et le coefficient NMNC des nœuds Mentor.

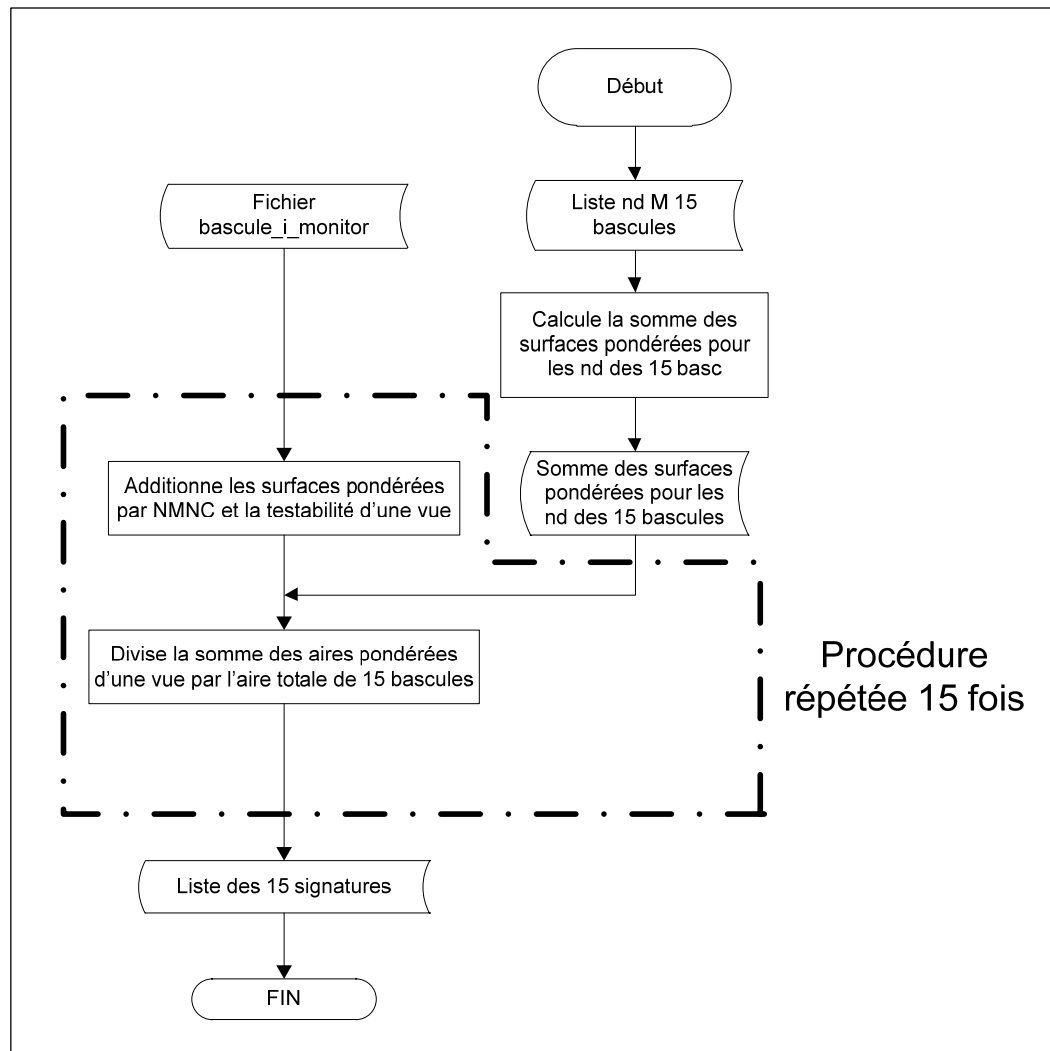


Figure-A XIII-6 Traitement pour obtenir la signature de la zone d'intérêt (méthode 4b).

Une fois trouvée une identité entre les noms, nous attribuons au nœud du fichier temporaire tous les caractéristiques manquantes apportées du fichier des nœuds appartenant aux 15 bascules. Dans cette manière nous obtenons 15 fichiers correspondants aux 15 bascules de la zone d'intérêt.

Nous bâtissons la signature proposée suite au traitement à la Figure-A XIII-6. La procédure effectue les calculs conformément aux équations 7.2 et 7.3.



## ANNEXE XIV

### FICHIERS UTILISÉS DANS LA MÉTHODE 4B

Fichier *sortie\_portes\_15\_bascules*

Nom de la bascule Mentor	Nom de la bascule Cadence	Sortie porte avant
B1832	data_out_reg[1]	U1555/Z
B1831	data_out_reg[0]	U1576/Z
B1847	count_reg2[3]	U2027/Z
B1857	count_reg2[5]	U2030/Z
B1868	play_reg	U1962/Z
B1850	count_reg2[4]	U1690/Z
B1849	count_reg2[2]	U1693/Z
B1940	nl_reg[0]	U2097/Z
B1848	count_reg2[1]	U1694/Z
B1830	sound_reg[0]	U1966/Z
B1826	max_reg[2]	U2006/Z
B1941	nl_reg[1]	U2099/Z
B1845	count_reg2[0]	U2033/Z
B1829	sound_reg[2]	U1787/Z
B1828	max_reg[4]	U2091/Z

Fichier résultat de Fastscan *Fastscan\_all* suite à une simulation en mode PANNE

(fragment)

```
// /b12_unit/U1555/Z
// |b12_unit/U1576/Z
// ||b12_unit/U2027/Z
// |||b12_unit/U2030/Z
// ||||b12_unit/U1962/Z
// |||||b12_unit/U1690/Z
```

```

//      |||||/b12_unit/U1693/Z
//      |||||/b12_unit/U2097/Z
//      |||||/b12_unit/U1694/Z
//      |||||/b12_unit/U1966/Z
//      |||||/b12_unit/U2006/Z
//      |||||/b12_unit/U2099/Z
//      |||||/b12_unit/U2033/Z
//      |||||/b12_unit/U1787/Z
//      |||||/b12_unit/U2091/Z
//      |||||
//      |||||
//-----
// 0 000111100111010
// 0 ----0----- { /b12_unit/U1955/Z@0 }
// 0 --1--0----- { /b12_unit/U2049/A1@0 }
// 0 --1--0----- { /b12_unit/U2049/Z@1 }
// 0 --1-----1-- { /b12_unit/U1831/Z@0 }
// 0 -----1----- { /b12_unit/U1799/Z@1 }
// 0 -----0--- { /b12_unit/U1796/A2@1 }
// 0 -----0--- { /b12_unit/U1796/Z@0 }
// 0 -----0- { /b12_unit/U1794/A1@0 }
// 0 -----0- { /b12_unit/U1794/Z@1 }
// 0 -----0--- { /b12_unit/U1811/A2@0 }
// 0 -----0--- { /b12_unit/U1811/Z@1 }
// 0 11----- { /b12_unit/U1864/Z@1 }
// 0 1----- { /b12_unit/U1869/Z@1 }
// 0 11----- { /b12_unit/U1835/Z@1 }
// 0 1----- { /b12_unit/U1861/Z@1 }
// 0 11----- { /b12_unit/U1837/Z@1 }

```

```
// 0 11----- { /b12_unit/U1836/Z@1 }
// 0 1----- { /b12_unit/U1832/Z@1 }
// 0 -----1 { /b12_unit/r271/U1_1_3/B@1 }
// 0 --1--0----- { /b12_unit/U1959/Z@0 }
// 0 11----- { /b12_unit/U1857/Z@1 }
// 0 -1----- { /b12_unit/U1859/Z@1 }
// 0 -1----- { /b12_unit/U1845/Z@1 }
// 0 11----- { /b12_unit/U1842/Z@1 }
// 0 1----- { /b12_unit/U1855/Z@1 }
// 0 -1----- { /b12_unit/U1858/Z@1 }
// 0 11----- { /b12_unit/U1846/Z@1 }
// 0 1----- { /b12_unit/U1844/Z@1 }
// 0 --1-----1-- { /b12_unit/U1812/A2@0 }
// 0 --1-----1-- { /b12_unit/U1812/Z@1 }
// 0 -----1----- { /b12_unit/U1819/Z@0 }
// 0 -----1----- { /b12_unit/U1899/Z@1 }
// 0 ---0-00----- { /b12_unit/U1873/A3@0 }
// 0 ---0-00----- { /b12_unit/U1873/Z@1 }
// 0 -----1-- { /b12_unit/U1999/Z@1 }
// 0 1----- { /b12_unit/U1983/A1@1 }
```

```
.....
// 98 011101101000100
// 98 -----1---1--- { /b12_unit/U1875/A1@0 }
// 99 010101001000100
// 99 -----1---1--- { /b12_unit/U1875/A1@0 }
// 100 110100100000100
// 100 -----1----- { /b12_unit/U1875/A1@0 }
// 101 001101000000111
// 102 001101001000100
```

```

// 102 -----1---1--- { /b12_unit/U1875/A1@0 }
// 103 000101101000100
// 103 -----1----- { /b12_unit/U1875/A1@0 }
// 104 001101011001000
// 105 111001011001100
// 105 -0----- { /b12_unit/U2124/B1@0 }
// 106 000000001100010
// 107 101100000000100
// 107 -----1--- { /b12_unit/U1875/A1@0 }
// 108 110100001000100
// 109 110101011101000
// 109 -0----- { /b12_unit/U2054/B1@0 }
// 109 0----- { /b12_unit/U2058/B1@0 }
// 110 100100101000100
// 110 -----1--- { /b12_unit/U1875/A1@0 }
// 111 000001001100111
// 112 110000000100010
// 113 001100101000100
// 113 -----1--- { /b12_unit/U1875/A1@0 }

```

**Fichier réponse du Fastscan *FS\_equiv* (fragment)**

Vecteur	Panne	Valeur erronée	Bascule d'observation	Valeur erronée	Bascule d'observation	Valeur erronée	Bascule d'observation
0	U1955/Z@0	0	B1850				
0	U2049/A1@0	1	B1847	0	B1850		
0	U2049/Z@1	1	B1847	0	B1850		
0	U1831/Z@0	1	B1847	1	B1845		
0	U1799/Z@1	1	B1940				
0	U1796/A2@1	0	B1941				
0	U1796/Z@0	0	B1941				
0	U1794/A1@0	0	B1829				

Vecteur	Panne	Valeur erronée	Bascule d'observation	Valeur erronée	Bascule d'observation	Valeur erronée	Bascule d'observation
0	U1811/A2@0	0	B1941				
0	U1811/Z@1	0	B1941				
0	U1864/Z@1	1	B1832	1	B1831		
0	U1869/Z@1	1	B1832				
0	U1835/Z@1	1	B1832	1	B1831		
0	U1861/Z@1	1	B1832				
0	U1837/Z@1	1	B1832	1	B1831		
0	U1836/Z@1	1	B1832	1	B1831		
0	U1832/Z@1	1	B1832				
0	r271U1_1_3/B@1	1	B1828				
0	U1959/Z@0	1	B1847	0	B1850		
0	U1857/Z@1	1	B1832	1	B1831		
0	U1859/Z@1	1	B1831				
0	U1845/Z@1	1	B1831				
0	U1842/Z@1	1	B1832	1	B1831		
0	U1855/Z@1	1	B1832				
0	U1858/Z@1	1	B1831				
0	U1846/Z@1	1	B1832	1	B1831		
0	U1844/Z@1	1	B1832				
0	U1812/A2@0	1	B1847	1	B1845		
0	U1812/Z@1	1	B1847	1	B1845		
0	U1819/Z@0	1	B1940				
0	U1899/Z@1	1	B1940				
0	U1873/A3@0	0	B1857	0	B1850	0	B1849
0	U1873/Z@1	0	B1857	0	B1850	0	B1849
0	U1999/Z@1	1	B1845				
0	U1983/A1@1	1	B1832				
0	U1983/Z@0	1	B1832				
0	U1913/A1@1	1	B1832	1	B1831		
0	U1913/Z@0	1	B1832	1	B1831		
0	U1856/Z@1	1	B1832	1	B1831		
0	U1854/Z@1	1	B1832	1	B1831		
0	U1843/Z@1	1	B1832	1	B1831		
0	U1841/Z@1	1	B1832	1	B1831		

## ANNEXE XV

### DESCRIPTION DÉTAILLÉE DES TRAITEMENTS SPÉCIFIQUES DE LA MÉTHODE 4C

Le traitement décrit aux figures-A XV-1 et XV-2 a pour but de déterminer le premier vecteur qui détecte une panne dans la séquence de test. Nous rappelons aussi que les vecteurs viennent toujours en même ordre, défini par le fichier des vecteurs. La procédure commence par un petit ménage du fichier équivalent de la réponse de Fastscan, *FS\_equiv* dans lequel nous enlevons les pannes internes et les bits des états logiques erronés pour garder juste les pannes et les vecteurs de test attachés à la bascule ou aux bascules de détection. A noter que la commande "analyze faults" ne prend pas en considération les nœuds internes. Nous savons qu'il y a 31 nœuds internes qui causent 125 pannes. Puis nous divisons ce fichier en sept fichiers chacun contenant toutes les pannes qui sont détectées par le même nombre des bascules. Puisque la combinaison maximale des bascules qui détecte une panne est de sept, nous obtenons 7 fichiers pour les sept possibilités des combinaisons allant de 1 à 7. Dans chaque fichier nous allons grouper chaque panne avec tous les vecteurs qui la détectent. Notre but est de chercher dans le cadre de chaque fichier le vecteur qui arrive en premier comme position dans la liste des vecteurs qui détectent la même panne. Une fois déterminé ce vecteur, les vecteurs qui ciblent la même panne seront effacés. L'avantage de faire une première recherche pour chacun de ce type de fichier est que l'endroit où se trouve le nombre de vecteurs tombe dans une position fixe et qu'il est facile à l'utiliser. Une fois déterminé le vecteur minimal par fichier, nous allons résoudre la compétition entre différents fichiers. Pour cela nous rejoignons toutes les séquences dans le même fichier et nous introduisons après chaque séquence amenée dans ce fichier un terminateur de séquence. En partant de la liste des pannes distinctes nous allons grouper de nouveau chaque panne avec toutes ses séquences qui le détectent. En ce moment toutes les séquences qui contiennent un seul terminateur peuvent être dirigées dans un fichier des résultats partiels puisque elles contiennent déjà le vecteur minimal et il n'y a pas d'autre compétition introduite par les autres fichiers. Maintenant nous cherchons le vecteur minimal (le vecteur est le seul champ

qui inclut juste des chiffres) de chaque panne. Une fois déterminé le vecteur minimal et ensuite sa position dans la séquence de chaque panne, nous cherchons les terminateurs de séquence qui encadrent le vecteur minimal. Le but est de garder comme combinaison des bascules de détection tout ce qu'existe de la position de vecteur minimale jusqu'au premier terminateur à droite et à gauche respectivement. Toute autre combinaison sera effacée. Le résultat, *liste des premières pannes à chaque bascule*, inclut la panne, le vecteur et la combinaison des bascules qui détectent la panne. Comme vérification ces listes doivent inclure toutes les pannes possibles une seule fois.

Le traitement décrit à la Figure-A XV-3 cherche la couverture de chaque bascule sur la base de des 15 fichiers "première panne" trouvés dans la procédure antérieure. C'est un ensemble de trois boucles simultanées, qui passent à travers la liste des bascules, le fichier des "*premières pannes*" et le fichier qui constitue la base de données des nœuds Mentor et leurs coefficients de testabilité, les nœuds de Cadence associés, les coefficients NMNC et les surfaces des polygones associés au DRC.

Le traitement pour obtenir les signatures, présenté à la Figure-A XV-4, effectue les calculs requis par les équations 7.2 et 7.3.





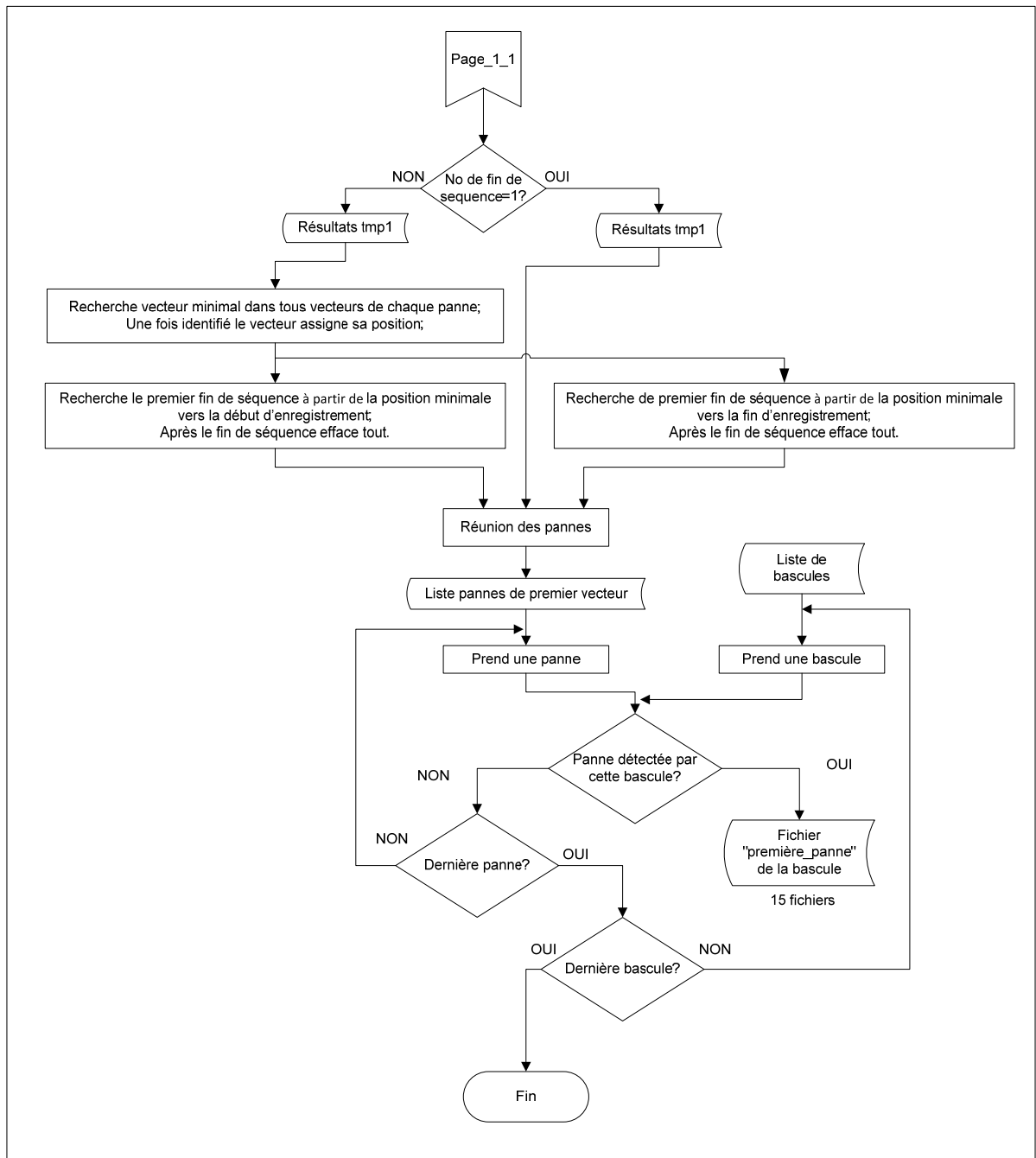


Figure-A XV-2 Traitement du fichier de réponse de Fastscan pour trouver les premiers vecteurs à détecter une panne (partie 2).

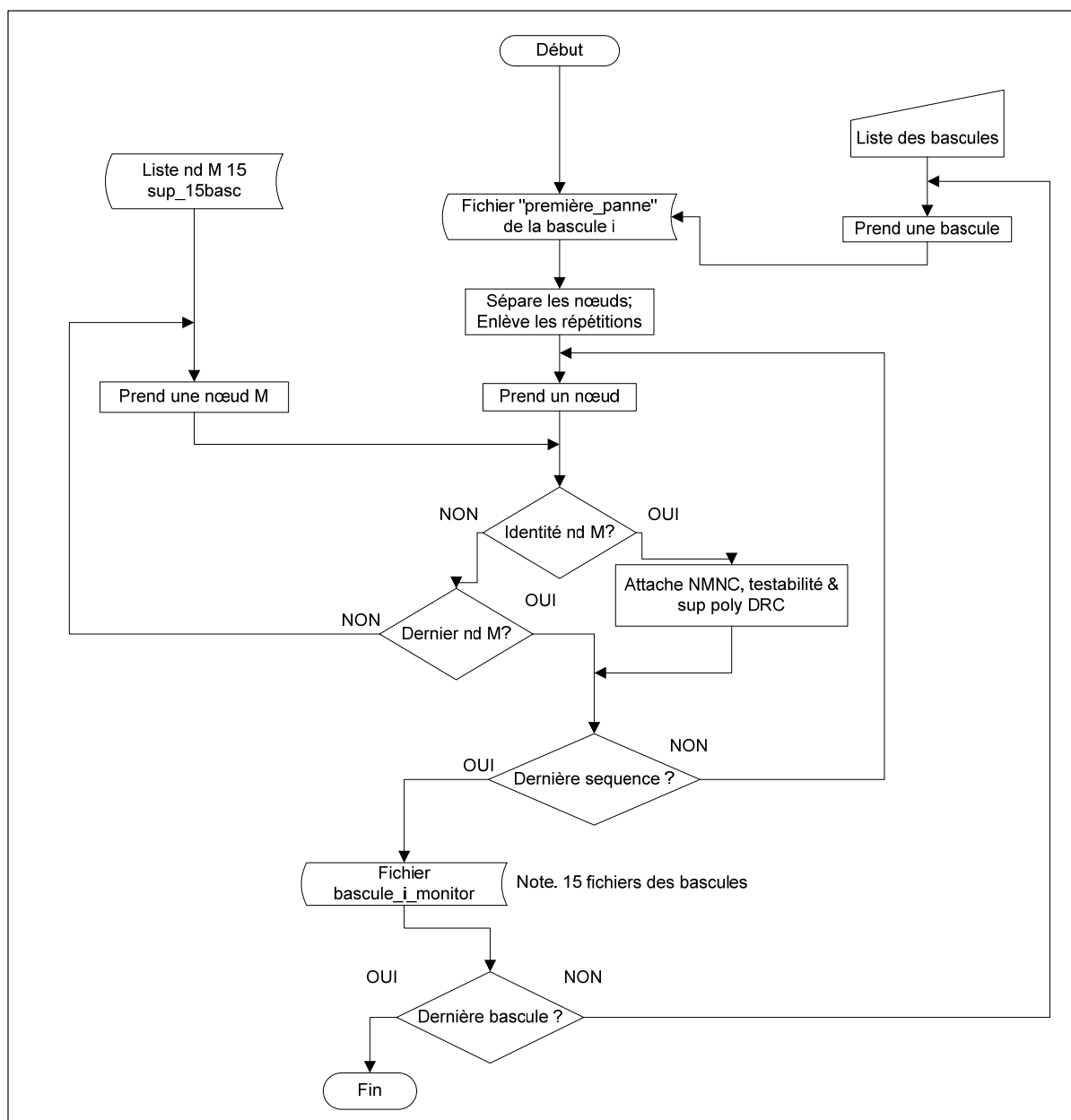


Figure-A XV-3 Traitement pour obtenir les nœuds couverts par chaque bascule conformément à la méthode 4c.

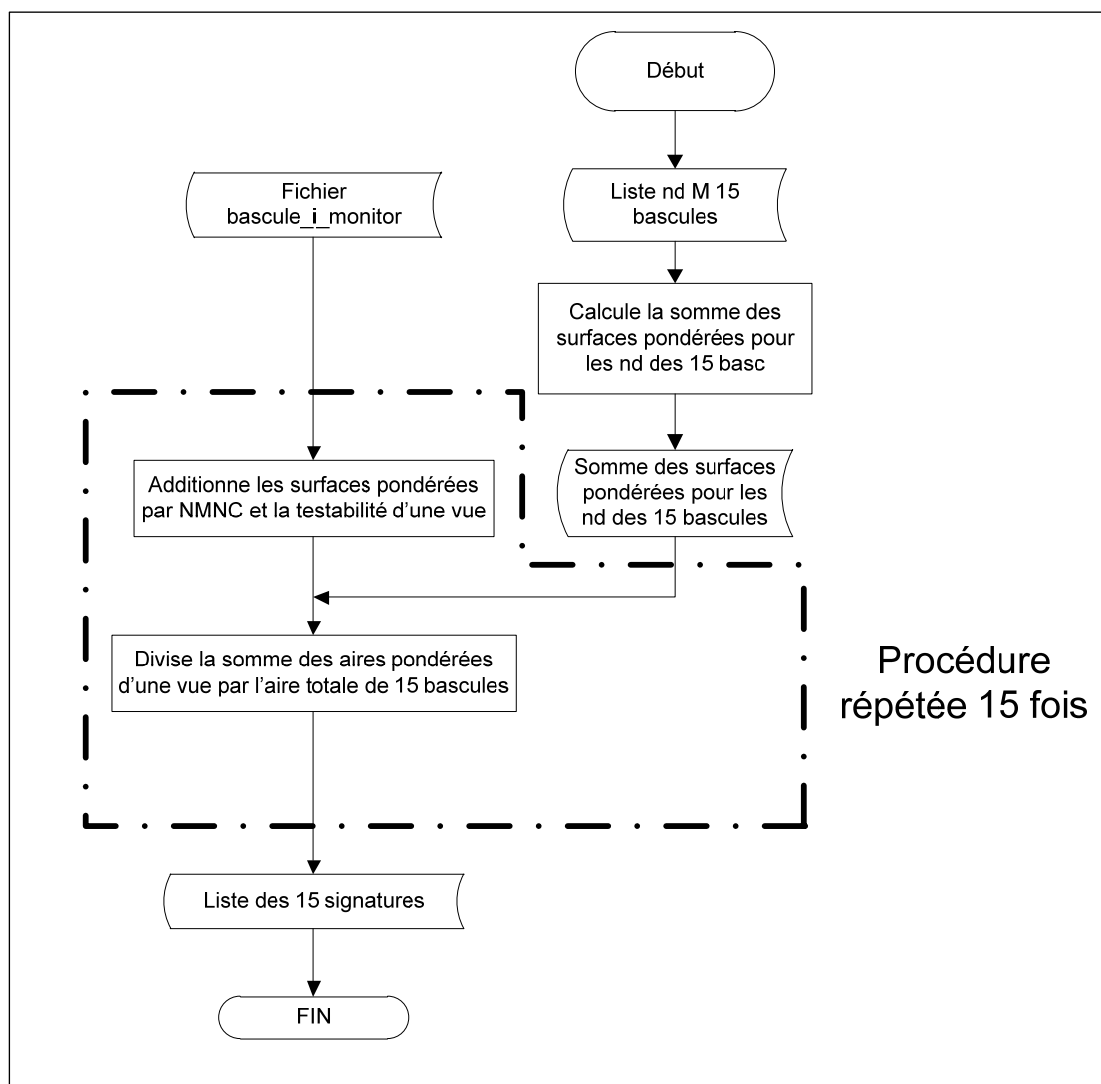


Figure-A XV-4 Traitement pour obtenir la signature conformément à la méthode 4c.

## ANNEXE XVI

### GRAPHE RÉCAPITULATIF DES SIGNATURES (VALEURS NUMÉRIQUES)

		Methode_1	Methode_2	Methode_3	Methode_4a	Methode_4b	Methode_4c
Signature	B1826	0,162018592	0,175260209	0,149669673	0,093704934	0,022925941	0,021284011
Signature	B1828	0,167330677	0,174337124	0,158713815	0,086213853	0,028297735	0,019629948
Signature	B1829	0,196547145	0,211067433	0,189806612	0,106247798	0,040764312	0,031461741
Signature	B1830	0,214254095	0,230290271	0,204927294	0,116291545	0,082719508	0,061619264
Signature	B1831	0,316069057	0,314835753	0,302957211	0,152345011	0,233625893	0,203495683
Signature	B1832	0,316069057	0,314049753	0,301875145	0,148150878	0,223613042	0,193908651
Signature	B1845	0,220894201	0,240101371	0,210751964	0,126362656	0,076283819	0,058242144
Signature	B1847	0,2833112	0,298321492	0,270718961	0,14792555	0,081682024	0,053357702
Signature	B1848	0,222222222	0,240469869	0,212302386	0,124531493	0,062323549	0,04414229
Signature	B1849	0,22664896	0,243830848	0,217012879	0,125309283	0,063160021	0,044740259
Signature	B1850	0,231961045	0,248819738	0,222439356	0,128139651	0,069226846	0,053795383
Signature	B1857	0,237715804	0,253488198	0,232105283	0,130929127	0,089232433	0,063449805
Signature	B1868	0,222664896	0,237715609	0,198683359	0,119312498	0,098336816	0,086001255
Signature	B1940	0,213368747	0,235814579	0,189046358	0,122901848	0,115545619	0,075796637
Signature	B1941	0,211598052	0,231997263	0,182667013	0,113882688	0,110664457	0,088397116

## BIBLIOGRAPHIE

1. C.Thibeault "Notes des cours - Les concepts et les techniques de conception de systèmes VLSI", ÉTS, 2009.
2. "Digital IC Design Flow - A Tutorial on RMC's Digital Design Flow", Collège Militaire Royal du Canada (Cadence University Alliance Program Member) Département de Génie Électrique et Informatique.
3. C. Schuermyer, K. Cota, R. Madge, and B. Benware, "Identification of systematic yield limiters in complex ASICs through volume structural test fail data visualization and analysis", in Proc. Int. Test Conf. (ITC), 2005.
4. Adhir Upadhyay, Vikas Gupta ``DFM-oriented test ensures better yield``, EE Times-India | January 16-31, 2008.
5. Geir Eide and Davide Appello, "Leveraging Diagnosis for Yield Analysis", Design Automation Conference, Anaheim, CA, 2010 June 13-18, Publisher: DAC.COM KNOWLEDGE CENTER ARTICLE.
6. Jeff Wilson, "2D and 3D Variability Analysis and Optimization", EDA Tech Forum, Technical Resource for the EE Community, 2009.
7. Shaun S. Gleason, Kenneth W. Tobin, Thomas P. Karnowski, Fred Lakhani, "Rapid Yield Learning Through Optical Defect and Electrical", Managed by Lockheed Martin Energy Research Corp. for the US. Department of Energy under contract DE-AC05-96OR22464.
8. R. Turakhia, M. Ward, S. Goel and B. Benware, "Bridging DFM Analysis and Volume Diagnostics for Yield Learning-A Case Study", IEEE Conference Proceedings 2009, Pages 167-172, ISBN: 1093-0167.
9. Kuo H. Wu, PhD, Marilyn Adan, ``When Following the Rules Is Not Quite Sufficient: The Need for New Post Route Analysis Techniques``, Synopsys, Inc., White Paper 2006.
10. Weste, N., Harris, D., *CMOS VLSI Design: A Circuits and Systems Perspective*, 3rd Edition, Addison Wesley, Reading (Mass.), 2005.
11. Manoj Gandhi, " The New Economics of Verification", Synopsys Insight, Issues 1, 2010.
12. J.Lawrence Carter, "The theory of signature testing for VLSI",ACMO-89791-067-2/82/O05/O066 1982.

13. J. A. Waicukauski, E. Lindbloom, " Failure diagnosis of structured VLSI", IEEE DESIGN & TEST OF COMPUTERS, 0740- 74 75 1891 0008-49\$1.000 1 989 IEEE.
14. Yu Huang, "Chain Diagnosis without Chain Patterns", Mentor Graphics Corporation, White Papers 2009.
15. Sujit Thomas Zacharias, "Algorithms for efficient extraction of faults in large VLSI circuits", A dissertation to the Faculty of the Graduate School of the State University of New York at Buffalo in partial fulfillment of the requirements for the degree of Doctor of Philosophy", August 2000.
16. David Abercrombie, Praveen Elakkumanan, Lars Liebmann, "Restrictive Design Rules and Their Impact on 22 nm Design and Physical Verification" Mentor Graphics, White papers, 2009.
17. Scott Davidson, "Characteristics of the ITC'99 Benchmark Circuit" March 1999, Benchmark Homepage, {"[www.cerc.utexas.edu/itc99-benchmbenchmarks/bench.html](http://www.cerc.utexas.edu/itc99-benchmbenchmarks/bench.html)"}.
18. Geir Eide, "Avoid throwing darts at a black hole by using diagnosis-driven yield analysis", Tessent and YieldInsight are trademarks of Mentor Graphics Corp, White Papers 2009.
19. Davide Appello, Alessandra Fudoli, Katia Giarda, Vincenzo Tancorre, Emil Gizdarski and Ben Mathew, "Understanding Yield Losses in Logic Circuits", IEEE Design & Test of Computers, May-June 2004, p.208-215.
20. Martin Keim, Nagesh Tamarapalli, Huaxing Tang, Manish Sharma, Janusz Rajske, Chris Schuermyer, Brady Benware, "A Rapid Yield Learning Flow Based on Production Integrated Layout-Aware Diagnosis", INTERNATIONAL TEST CONFERENCE, 2006 IEEE, paper 7.1
21. Jayanth Mekkoth, Murali Krishna, Jun Qian, Will Hsu, Chien-Hui Chen, Yuan-Shih Chen, Nagesh Tamarapalli, Wu-Tung Cheng, Jan Tofte, Martin Keim, "Yield Learning with Layout-aware Advanced Scan Diagnosis", Proceedings of the 32nd International Symposium for Testing and Failure Analysis November 12-16, 2006, p.412-418