



HAL
open science

Qualification et génération automatique de stimuli pour le test de systèmes sur puces (SoC) analogiques mixtes et RF.

Yves Joannon

► **To cite this version:**

Yves Joannon. Qualification et génération automatique de stimuli pour le test de systèmes sur puces (SoC) analogiques mixtes et RF.. Autre. Institut National Polytechnique de Grenoble - INPG, 2008. Français. NNT: . tel-00389783

HAL Id: tel-00389783

<https://theses.hal.science/tel-00389783>

Submitted on 29 May 2009

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

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

« Chercher n'est pas une chose et trouver une autre, mais le gain de la recherche, c'est la recherche même. »

Saint Grégoire de Nysse

Remerciements

Je souhaiterais remercier tous ceux qui m'ont côtoyé durant ces trois années de thèse. Tous, vous m'avez soutenu, guidé, diverti lors de cette *expérience* personnelle et professionnelle.

Plus particulièrement,

Je tiens à remercier Madame Chantal Robach, Directrice du Laboratoire de Conception et d'Intégration de Systèmes Grenoble INP, de m'avoir accueilli au sein du LCIS et d'avoir participé au co-encadrement de cette thèse.

Je tiens à exprimer ma reconnaissance à mon directeur de thèse ; Monsieur Smail Tedjini, Professeur à l'ESISAR Grenoble INP, pour avoir accepté de diriger cette thèse.

Mes remerciements vont tout particulièrement à Monsieur Jean-Louis Carbonero, Ingénieur ST Microelectronics Crolles, pour avoir encadré ma thèse en apportant une approche industrielle. Lors de ce projet, il m'a donné de précieux conseils et a apporté une critique constructive tant sur les axes de recherche que sur l'amélioration des publications et de ce mémoire de thèse.

J'adresse toute ma reconnaissance à Monsieur Vincent Berouille, maître de conférence à l'ESISAR Grenoble INP, pour avoir suivi régulièrement l'avancement de ma thèse. Je voudrais le remercier pour ses qualités humaines et ses connaissances, il m'a ainsi permis d'évoluer durant ces trois années de thèse et d'accomplir ce projet de recherche.

Je remercie Messieurs Ian O'Connor, Professeur dans l'Université de Lyon, et Michel Renovell, directeur de Recherche CNRS, pour avoir lu mon manuscrit en tant que rapporteurs et pour avoir apporté une autre vision sur mes travaux de recherche.

Je tiens aussi à remercier Messieurs Philippe Cauvet, Ingénieur NXP Semiconductors, et Yann Deval, Professeur à l'Université des Sciences et Technologies de Bordeaux, pour avoir bien voulu participer au jury de cette thèse.

Je voudrais aussi remercier tous ceux que j'ai côtoyés au quotidien durant ma thèse en commençant par Youssef (collègue de bureau) pour nos discussions diverses et variées autour de la machine à café. Je remercie aussi les membres du laboratoire : Carole, Christophe, Damien, Edouardo, Etienne, Fred, Jean-Luc, Laurent(s), Michel, Oum-el-ker, Pierre, Tan. J'adresse aussi mes remerciements aux doctorants qui participent à la vie et font le dynamisme de notre laboratoire :

Anthony, Anne-Laure, Audrey, Bao, Boussade, Fabrice, Fassely, Hind, Jean-Marie, Philippe, Rami (pour son support), Remi, Romain (pour nos ballades en moto), Thi-quinn, Vu.

Je fais part de toute mon amitié :

- A Larajasse pour les week-ends à la fois agités et récupérateurs : Alain, Anne-Marie, Boubou, Bruce, Claire, Claudie, Christophe, Christian, Cyril, Elodie, Emilie, Evelyne, Floriane, Jérôme, Michael, Nicolas, Pascal, Serge, Stéphanie, Violaine...

Parce que j'apprécie chaque week-end passé avec vous.

- Aux fous du ski pour leurs divertissements durant les longues nuits d'hivers : Bruno, Nicolas, Nunu, Miguel, Pof, Pierrick (s), Seb, Teuf...

Parce que je vais tous vous *pou...ir* !

- A la GPT pour les campings à Chanas, raclettes, mariage et autres : Fabien, Geo, Janfi, Pierre, Rémi, Seb, Tito, Yann et leurs copines.

Parce que cette amitié qui a débutée en 2001 reste forte malgré l'éloignement géographique.

- Aux motards pour les sorties hebdomadaires et ce tour de Corse : Belinda, Bol, Cyril, Jonjon, Cyril, Pierre-Emmanuel et ceux que j'ai déjà cités.

Parce que j'aime faire ces petites virées avec vous.

- Aux joueurs de poker pour les soirées en semaine et les débats interminables : Bruno, Christophe, Saad, Tito et Virginie.

Parce que c'était bien sympathique.

Pour terminer, je ne saurais oublier de remercier mes parents pour m'avoir permis d'en arriver là grâce à leur soutien et leurs encouragements, ma sœur Véronique et ainsi que mon frère Hervé.

Pour plus de clarté, la table des matières ci-dessous ne fait référence qu'aux chapitres traités. Elle est complétée par un sommaire détaillé fourni au début de chaque chapitre.

Chapitre I :

Introduction Générale. Page 7

Chapitre II :

Flot de conception de systèmes analogiques mixtes et radiofréquences. Page 11

Chapitre III :

Méthodes de validation de conception et de test de systèmes analogiques mixtes et RF. Page 35

Chapitre IV :

Modélisation comportementale d'un système W-CDMA. Page 57

Chapitre V :

Qualification de stimuli pour la validation de conception. Page 81

Chapitre VI :

Réutilisation des stimuli de la validation pour le test de production. Page 119

Chapitre VII :

Conclusion générale et perspectives. Page 151

Résumé

Page 156

CHAPITRE I

INTRODUCTION GENERALE

1.1. Contexte général

Les premiers phénomènes électriques ont été découverts il y a plusieurs siècles, mais l'électricité et l'électronique ont subi une véritable révolution seulement au cours du siècle dernier. Suite au développement des systèmes électriques, le terme électronique est apparu en 1904 avec l'invention du tube électronique. Puis, dans les années 50, la tendance à la miniaturisation a conduit au développement des technologies de la microélectronique. Depuis ces premiers travaux, l'utilisation de composants électroniques est devenue courante dans notre vie. Au quotidien, nous côtoyons un nombre croissant d'objets reposant sur cette technologie : téléphones portables, télévisions hautes définitions, ordinateurs, automobiles, appareils médicaux... Lors de la conception de ces produits, et en particulier dans le cas d'applications nomades, le concepteur cherche à intégrer le maximum de fonctionnalités dans ses circuits. Cela le conduit à réduire la taille de la partie électronique et ainsi à diminuer le poids et/ou le volume du produit final. L'objectif de cette intégration est également de réduire les coûts des produits électroniques mais la miniaturisation reste la ligne directrice principale suivie pour le développement des technologies futures.

En plus de la miniaturisation et du coût, les concepteurs ne doivent pas négliger la qualité des circuits. Ainsi, en plus de produire des circuits intégrés complexes, ils doivent être capables de les vérifier c'est-à-dire de s'assurer de leur fonctionnement. En particulier, dans les applications pour lesquelles des vies humaines sont en jeu, il est indispensable de garantir une très haute qualité à la fois de conception et de fabrication des circuits. On parle ainsi du "Zéro défaut" pour les produits impliqués dans la sécurité automobile.

Dans les années 90, l'intégration à très grande échelle a tout d'abord entraîné une augmentation de la complexité des circuits intégrés. Depuis quelques années, l'intégration de véritables systèmes

numériques, que l'on appelle communément Systèmes-sur-Puces (SoC¹) digitaux, est devenue une réalité ; différentes méthodes ont été mises en place afin de répondre aux problématiques de test de ces systèmes.

Les technologies de fabrication des systèmes analogiques mixtes et RF (AMS&RF²) sont très récentes. Les SoCs AMS&RF sont des systèmes hétérogènes composés de parties numériques, analogiques, mixtes, et RF. Un exemple de SoCs AMS&RF est un émetteur/récepteur RF. Il est composé d'une partie qui réalise la modulation/démodulation (analogique et RF), mais aussi d'une partie numérique capable de générer les séquences à envoyer et d'analyser les séquences reçues. Les méthodes pour réaliser le test matériel de ces systèmes n'ont pas évoluées aussi rapidement que les méthodes de conception et de fabrication. En conséquence, les fondeurs rencontrent de réelles difficultés pour s'assurer de la qualité de ces systèmes pour un coût raisonnable. Cette thèse développe des méthodes permettant de traiter la problématique du test des SoCs AMS&RF.

Comme tous les fabricants de systèmes microélectroniques, l'entreprise STMicroelectronics est fortement intéressée par cette problématique car elle est capable de concevoir ce type de systèmes intégrés hétérogènes. Toutefois, les méthodes de test actuellement utilisées ne sont pas optimales pour le test de ces systèmes. En effet, les principales solutions actuellement utilisées consistent à câbler en externe les entrées/sorties de chacun des blocs et ainsi à tester individuellement chaque bloc. Le problème de cette approche est qu'elle nécessite l'ajout de ports permettant le contrôle de chacun des blocs. Cette approche diminue les performances du système (délais et bruit additionnels) et est également coûteuse. Elle est donc difficilement applicable au test des systèmes complexes. Dans la littérature, plusieurs autres propositions de *Conception en Vue du Test (CVT)*, en anglais *Design-for-Test*, ont été faites afin d'accroître la contrôlabilité et l'observabilité de ces systèmes. Pour le moment, ces architectures de test ne couvrent pas la diversité des systèmes AMS&RF existants. Il est donc indispensable dans un premier temps d'optimiser les méthodes existantes et de mettre en place de nouvelles méthodes et outils permettant de tester les SoCs AMS&RF sans avoir recours à des composants additionnels pour le test.

1.2.Problématiques

Nous avons vu que l'augmentation de l'intégration et donc de la complexité des systèmes électroniques est réelle. La première problématique liée à cette évolution consiste à s'assurer que le système a été correctement conçu. En effet, les systèmes doivent supporter de plus en plus de fonctionnalités. Il est donc indispensable de mettre en place une méthode de validation de conception rigoureuse. L'objectif sera alors de s'assurer que le circuit développé respecte les spécifications définies dans le cahier des charges. Pour cela, il est en général nécessaire de définir un plan de vérification permettant de décrire les procédures de test permettant de vérifier la conception du système. Actuellement, il n'existe aucune méthode permettant de s'assurer que cette procédure de test est suffisante pour la validation de conception. Il est donc nécessaire de développer une méthode permettant d'évaluer ce plan de vérification.

Une fois que le circuit a été correctement conçu, il faut aussi s'assurer que celui-ci est correctement

¹ System on Chip

² Analogiques Mixtes et RadioFréquences.

fabriqué. Pour cela, il est indispensable de définir un programme de test capable de déterminer si le circuit testé est exempt de défaut physique. Des approches capables d'identifier des stimuli de test existent pour le test des circuits numériques mais elles ne sont pas adaptées au test des SoCs AMS&RF. Il est donc indispensable de proposer une méthode et des outils permettant d'assister l'ingénieur de test lors de la définition des programmes de test.

1.3. Présentation du plan du manuscrit de thèse

Ce mémoire est organisé en sept chapitres incluant la présente introduction et la conclusion. Le chapitre II présente les méthodes de conception ou flots de conception mises en place lors de la conception des SoCs AMS&RF. Les différents niveaux de description utilisés sont décrits ainsi que les langages de description et les outils de simulation associés.

Le chapitre III présente les deux principales finalités du test : vérification et validation de conception et test matériel. Les origines des défauts apparaissant durant la conception et durant la fabrication du circuit sont décrites et permettent de justifier les objectifs de ces deux types de test. Les méthodes de test utilisées pour définir les jeux de test pour ces deux finalités de test sont ensuite présentées. Les limites de ces méthodes pour le test des SoCs AMS&RF sont finalement abordées.

Le système intégré utilisé pour la validation de l'approche développée dans ce mémoire est présentée dans le chapitre IV. Le cas étudié est la partie réception d'un système W-CDMA³. L'architecture utilisée est développée suite à une rapide description de la technologie de modulation W-CDMA. Cette présentation de l'architecture permet de décrire le système au niveau comportemental en utilisant le langage VHDL-AMS. Cette description comportementale sera utilisée par la suite pour valider l'approche développée.

Dans le chapitre V, nous développerons la méthode de qualification des jeux de test utilisés pour la vérification des différentes étapes du flot de conception Top-Down. Cette méthode a pour objectif d'évaluer le plan de vérification, quelques soient les niveaux de modélisation associés, et plus particulièrement de qualifier les bancs de test qui sont nécessaire à la validation de conception.

Le chapitre VI est consacré à la présentation et à la validation d'une méthode et d'un outil logiciel (PLASMA : Plate-forme pour la qualification et la génération de stimuli de test pour les Systèmes Mixtes et Analogiques) permettant de qualifier les stimuli prédéfinis et de générer des stimuli complémentaires pour le test matériel. Contrairement au chapitre précédent, ces stimuli auront pour objectifs de détecter des défauts physiques pouvant apparaître lors de la fabrication du système.

Enfin, la conclusion résumera les travaux exposés dans ce mémoire et présentera les principales perspectives envisagées.

³ Wideband Code Division Multiple Access.

CHAPITRE II :

FLOT DE CONCEPTION DE SYSTEMES

ANALOGIQUES MIXTES ET RADIOFREQUENCES

| | |
|---|----|
| CHAPITRE II : | 11 |
| FLOT DE CONCEPTION DE SYSTEMES ANALOGIQUES MIXTES ET RADIOFREQUENCES | 11 |
| 2.1. Introduction | 14 |
| 2.2. Systèmes sur puce analogiques, mixtes et RF | 15 |
| 2.3. Flot de conception des SoCs AMS&RF | 15 |
| 2.3.1. Enjeux du flot de conception | 15 |
| 2.3.2. Conception des systèmes numériques et des systèmes analogiques | 16 |
| 2.3.2.1. Synthèse de SoCs numériques et analogiques | 16 |
| 2.3.2.2. Robustesse du système | 18 |
| 2.3.3. Flot de conception Bottom-Up pour les SoCs AMS&RF | 19 |
| 2.3.4. Flot de conception Top-Down pour les SoCs AMS&RF | 20 |
| 2.3.4.1. Définition de l'architecture | 21 |
| 2.3.4.2. Spécification des blocs | 21 |
| 2.3.4.3. Conception du schéma électrique | 22 |
| 2.3.4.4. Réalisation du placement routage | 22 |
| 2.3.5. Vérification de la conception par simulation | 23 |
| 2.4. Différents outils de simulation pour différents niveaux d'abstraction | 24 |
| 2.4.1. Simulation au niveau système ou niveau fonctionnel | 25 |
| 2.4.2. Simulation au niveau circuit | 25 |
| 2.4.2.1. Analyses statique (DC) | 26 |
| 2.4.2.2. Analyse transitoire (TRAN) | 26 |
| 2.4.2.3. Analyse fréquentielle (AC) | 26 |
| 2.4.2.4. Analyse périodique ou quasi-périodique (SST) | 26 |
| 2.4.2.5. Analyse transitoire de l'enveloppe (MODSST) | 26 |
| 2.4.3. Simulation au niveau bloc ou niveau comportemental | 27 |
| 2.5. Langages de description de SoCs AMS&RF | 28 |
| 2.5.1. VHDL | 28 |
| 2.5.2. VHDL-AMS | 29 |
| 2.6. Modélisation comportementale | 29 |
| 2.7. Conclusion | 30 |
| Bibliographie | 31 |

| | | |
|------------------|---|-----------|
| Fig. 2.1. | Diminution de la taille des transistors..... | 14 |
| Fig. 2.2. | Cycle de vie d'un produit. | 16 |
| Fig. 2.3. | Distribution normale et aire correspondant aux circuits défaillant..... | 19 |
| Fig. 2.4. | Flot de conception de SoCs AMS&RF. | 21 |
| Fig. 2.5. | Flot de conception multi-domaines pour des SoCs AMS&RF..... | 24 |
| Fig. 2.6. | Evolution des langages de description en fonction de l'intégration des transistors [2.15]. | 25 |

2.1.Introduction

Depuis l'apparition de l'électronique dans les années 60, celle-ci a subi une impressionnante évolution. En effet, la taille des transistors a été réduite d'un facteur dix mille et le prix du transistor s'est effondré. Cette révolution technologique a entraîné l'omniprésence de systèmes électroniques dans notre vie quotidienne. Ainsi, nous côtoyons tous les jours différents systèmes électroniques par le biais de l'utilisation de produits tel que : ordinateurs, téléphones portables, télévisions, baladeurs MP3, automobiles, appareils médicaux...

Le développement technologique de la microélectronique permet de concevoir des systèmes de plus en plus performants. Par exemple, la vitesse d'horloge devrait atteindre d'ici quinze ans 100GHz (Road map ITRS¹ 2007) contre 4GHz actuellement [2.2]. Le phénomène de miniaturisation est le second effet de cette révolution. Effectivement, les consommateurs de plus en plus amateurs de hautes technologies utilisent des systèmes nomades ultra-portables. Ainsi, l'évolution des technologies de fabrication permet actuellement de produire industriellement des transistors de 65nm de largeur de grille et pourrait atteindre les 20nm d'ici une vingtaine d'années (Fig. 2.1). La taille du transistor diminue de 15% par an et son coût de fabrication de 42%. Au final, les circuits intégrés sont de plus en plus complexes, intègrent de plus en plus de fonctionnalités et consomment de moins en moins d'énergie. D'après la loi de Moore, le nombre de transistors par puces est multiplié par deux tous les dix huit mois. Le nombre de transistors intégrés sur un même substrat est d'environ 500 millions en 2007 et pourrait atteindre les 12 milliards en 2019 d'après les estimations de l'ITRS [2.3]. Cette loi reste valable en 2007 mais pourrait être remise en cause dans le futur car les chercheurs sont confrontés à de nouvelles barrières physiques induites par la miniaturisation (travail à l'échelle de l'atome).

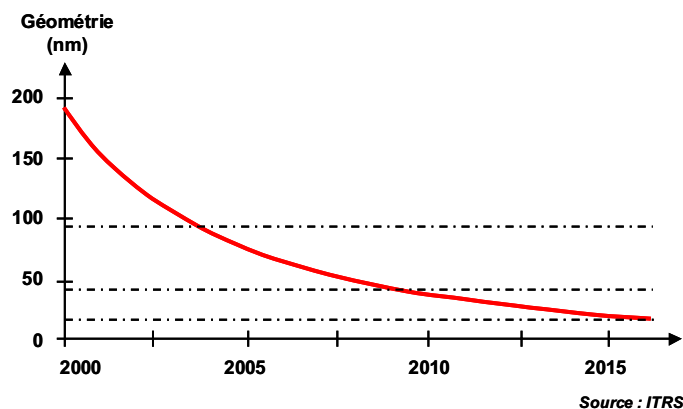


Fig. 2.1. Diminution de la taille des transistors.

La miniaturisation permet d'intégrer de plus en plus de fonctionnalités sur un même substrat (SoC²) ou dans un même boîtier (SiP³). Le nombre de SoCs fabriqués ne cesse d'augmenter. En 2004, le nombre de SoCs vendus s'est élevé à 900 millions et devrait atteindre les 2,2 milliards en 2009 [2.4]. Cette miniaturisation augmente la complexité de conception de ces systèmes. Il est donc

¹ International Technology Roadmap for Semiconductors.

² System on Chip.

³ System in Package.

indispensable d'utiliser des méthodes de conception adaptées à l'évolution de la microélectronique. Dans ce chapitre, nous présenterons les différentes approches utilisées lors de la conception de SoCs AMS&RF⁴. De ce fait, nous aborderons les particularités des différents domaines (numériques, analogiques et RF), les langages de description et les outils qui ont été développés afin de permettre la conception et la vérification de ces systèmes.

2.2. Systèmes sur puce analogiques, mixtes et RF

La notion de SoC est apparue dans les années 90 lorsque les concepteurs ont commencé à intégrer plusieurs composants sur un même substrat. Les premiers SoCs étaient purement numériques et sont une conséquence de l'intégration à grande échelle (VLSI⁵).

Les SoCs ont ensuite évolués en intégrant des parties analogiques mixtes et RF. En 2007, plus de 73% des SoCs sont composés de parties numériques et analogiques (source : IBS corp. [2.5]). La conception de SoCs AMS&RF présente cependant des caractéristiques différentes de celles des SoCs purement numériques et nécessite donc l'utilisation de méthodes de conception adaptées. Il ne suffit pas uniquement d'ajouter des parties analogiques ou RF à un système numérique, mais il faut être capable de fabriquer conjointement un système hétérogène complet. Par exemple, lorsque le système est homogène c'est-à-dire purement numérique ou purement analogique, une seule technologie est utilisée pour la fabrication du système : soit CMOS, soit Bipolaire [2.3]. Mais lors du développement de systèmes mixtes, il faut être capable d'intégrer ces deux types de transistor sur un même substrat. En fait, la plupart des circuits numériques sont fabriqués en technologie CMOS. Or, cette technologie utilisée seule conduit à des performances RF limitées. Ainsi, pour intégrer ces fonctions RF ultra-rapides aux circuits numériques CMOS, les concepteurs utilisent généralement une technologie mixte baptisée BiCMOS⁶. Cette technique a pour effet d'augmenter la complexité de fabrication des systèmes. En revanche, lorsqu'il n'est pas possible d'intégrer les différentes parties d'un système sur un même substrat, il est possible d'intégrer le tout dans un même boîtier, on parle alors de « System in Package » [2.6]. Cette technique d'intégration permet ainsi de diminuer la taille du système, mais augmente sa complexité de fabrication et plus particulièrement de son encapsulation (Packaging) dans un boîtier.

Le développement de tous les systèmes multi-domaines nécessite l'utilisation de méthodes de production et de conception rigoureuses et adaptées à leur complexité. Ainsi, dans la partie suivante, nous aborderons les différentes approches utilisées pour concevoir les SoCs AMS&RF.

2.3. Flot de conception des SoCs AMS&RF

2.3.1. Enjeux du flot de conception

Face à un marché en perpétuelle progression, les fabricants de SoCs doivent faire preuve d'une grande réactivité. Ils doivent ainsi continuellement améliorer leurs méthodes de conception et de fabrication afin de mettre leurs produits sur le marché le plus rapidement possible ; la contrainte du « Time To Market (TTM) » est au cœur de tout nouveau projet [2.7]. L'avance obtenue par un fondeur sur ses concurrents aura un impact significatif sur le nombre de produits vendus et donc sur

⁴ Analog and, Mixed Signal & Radio Frequency.

⁵ Very-Large-Scale Integration.

⁶ Bipolaire et CMOS.

les profits réalisés (Fig. 2.2). En effet, le premier produit mis sur le marché conservera son avance par rapport à ses concurrents tout au long de son cycle de vie. La Fig. 2.2 montre que dans le premier cas (courbe en trait continu), les produits ont pu être mis sur le marché plus tôt car le TTM1 est plus court. L'avance qui a été conservée tout au long du cycle de vie du produit a ainsi permis de générer plus de profits (aire hachurée) que dans le cas de la courbe en trait pointillé.

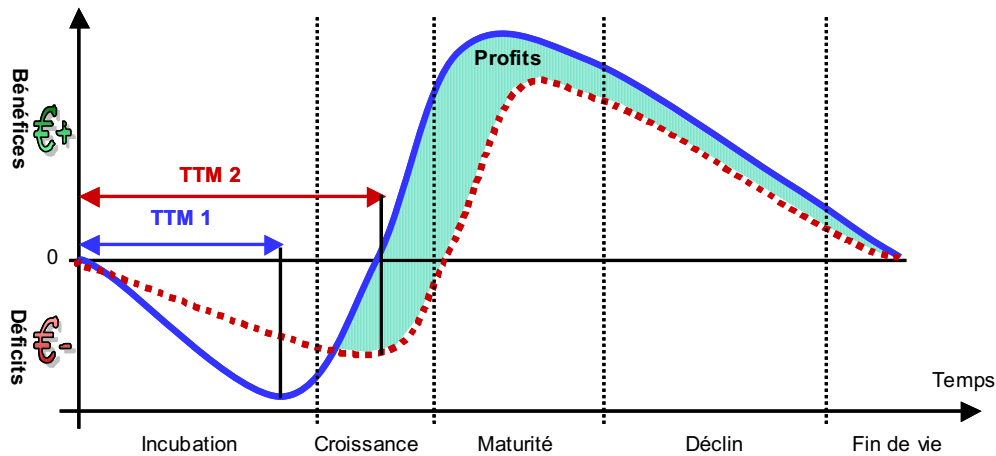


Fig. 2.2. Cycle de vie d'un produit.

Les facteurs permettant de diminuer le TTM sont liés aux capacités de conception et aux capacités de fabrication. La capacité de production est fortement liée aux savoir-faire de l'entreprise. Dans le domaine de la microélectronique, il s'agit par exemple des capacités d'un fondeur à fabriquer certains types de composants grâce à la maîtrise des technologies. Par exemple, pour la fabrication de SoCs AMS&RF, il est possible d'utiliser la technologie BiCMOS. Si le fondeur ne la maîtrise pas, il devra la développer ce qui aura pour effet d'augmenter le TTM. Dans la plupart des cas, la nouvelle tendance consiste à développer les technologies en partenariat avec plusieurs fondeurs afin de diminuer les coûts de recherche et de développement ; ces technologies sont donc généralement contrôlées par plusieurs fondeurs. Par conséquent, la maîtrise des technologies de fabrication en termes de robustesse et de qualité est indispensable mais n'est pas suffisante pour diminuer suffisamment le TTM. La réduction du TTM vis-à-vis de ses concurrents ne peut donc se faire que lors de la conception du système. Nous verrons dans la suite de ce chapitre comment il est possible de réduire les temps de conception en utilisant la méthodologie de conception Top-Down.

2.3.2. Conception des systèmes numériques et des systèmes analogiques

La conception de SoCs numériques demande quelques mois, elle est facilitée par l'utilisation de méthodes de conception rigoureuses et d'outils de conception adaptés. Par contre, la conception de SoCs analogiques peut nécessiter jusqu'à trois ou quatre modifications de la conception (« respins ») et plus d'un an et demi de conception [2.8]. Cette partie va nous permettre de mettre en évidence les particularités de chaque domaine et ainsi d'identifier les causes de l'écart entre les temps de conception de systèmes numériques et de systèmes analogiques.

2.3.2.1. Synthèse de SoCs numériques et analogiques

Comme précédemment cité, la notion de SoC est apparue dans les années 90 pour les systèmes purement numériques. L'intégration à grande échelle, permettant le développement de systèmes

VLSI⁷, a rapidement confronté les concepteurs de circuits numériques aux problèmes de conception de systèmes complexes. Des travaux de recherches dans le domaine numérique ont permis de mettre en place des méthodes de conception robustes et adaptées à ces systèmes. Ainsi, pour assister la conception, différents langages de description ont été standardisés afin de permettre la modélisation et la simulation à différents niveaux d'abstraction durant le flot de conception. Il est ainsi possible de décrire les systèmes au niveau physique (niveau transistor), au niveau porte, au niveau RTL⁸ et au niveau comportemental ou fonctionnel. Les axes de recherches se sont ensuite rapidement portés sur le passage automatique d'un niveau à un autre ; on parle de synthèse. La synthèse des SoCs numériques est automatisable car le domaine numérique possède des caractéristiques discrètes qui permettent de faciliter l'automatisation. Par exemple, la plupart des systèmes numériques sont construits à partir de cellules élémentaires (portes logiques définies dans des bibliothèques) et le nombre de cellules élémentaires différentes est limité ; il est alors possible de développer des algorithmes permettant de convertir des macro-blocs en cellules élémentaires. Une seconde raison permettant la synthèse automatique de SoCs numériques vient du fait que les entrées/sorties numériques manipulent des signaux possédant les mêmes propriétés (tension V_{HI} ⁹ et V_{LO} ¹⁰), il n'y a donc pas de risque d'incompatibilité de connexions entre deux blocs. Ainsi, des outils automatisant le passage entre ces différents niveaux d'abstraction (niveau comportemental, RTL, porte...) et permettant leur vérification ont été mis en place.

Lors de la conception de SoCs AMS&RF, les principales difficultés rencontrées sont identiques à celles évoquées pour le domaine numérique, mais il faut aussi faire face à de nouvelles difficultés. Des approches de conception semblables à celles du domaine numérique ont ainsi été mises en place. Des niveaux d'abstractions et des langages ont été définis et standardisés afin d'assister les concepteurs de ce type de systèmes. Ces niveaux de description et langages seront décrits dans la partie §2.4. Mais en plus de ces difficultés, le concepteur de circuits analogiques est confronté à d'autres problèmes rendant difficile la synthèse automatique. En effet, lors de la modélisation de SoCs AMS&RF, il est difficile de diviser le système en sous blocs élémentaires « standards » comme c'est le cas dans le domaine numérique où le système peut être décrit sous forme de portes logiques. Même s'il est possible de définir quelques blocs élémentaires grâce à l'utilisation de bibliothèques analogiques ou d'IP¹¹, ces ensembles de composants sont loin d'être suffisants et donc ne permettent pas de couvrir toutes les applications possibles. En plus de cette particularité, des problèmes concernant les interconnexions se posent lors de la conception de SoCs analogiques et RF. Si dans le domaine numérique, tous les signaux sont définis au même format, ce n'est pas le cas dans le domaine analogique et RF. En analogique, les blocs et les connexions dépendent des signaux qui leur sont appliqués. En fait, les blocs doivent alors être dimensionnés en fonction de caractéristiques des signaux (tension, courant, fréquence, ...). Des outils de modélisation et de simulation utilisant différents niveaux d'abstraction permettent d'assister le concepteur lors de la définition de son système, mais il n'existe pas d'outil commercialisé de synthèse automatique

⁷ Very-Large-Scale Integration.

⁸ Register Transfert Level.

⁹ Tension niveau logic 1.

¹⁰ Tension niveau logic 0.

¹¹ Intellectual Property.

comme dans le domaine numérique. Actuellement, le passage d'un niveau d'abstraction à un autre est en grande partie réalisé manuellement. Cette carence est un problème car la synthèse manuelle augmente sensiblement les risques d'erreur de conception.

La conception de SoCs AMS&RF intégrant à la fois la conception numérique et la conception analogique est de ce fait limitée par les contraintes du domaine analogique. Il est toutefois possible de décrire ces systèmes à différents niveaux d'abstraction, en utilisant des langages de description compatibles à la fois avec le domaine numérique et le domaine analogique [2.9]. Mais comme lors de la conception des systèmes analogiques, il n'est pas possible de passer automatiquement d'un niveau d'abstraction à un autre, ni même de faire une vérification automatique du système décrit.

2.3.2.2. Robustesse du système

Lors de la fabrication de circuits, des variations se produisent sur les paramètres du processus de fabrication. Dans le cas de la conception de SoCs numériques, cette notion n'était pratiquement pas prise en compte car les blocs élémentaires numériques étaient suffisamment robustes aux variations du processus de fabrication. Toutefois, comme la miniaturisation entraîne aujourd'hui la diminution des largeurs de grilles, les systèmes numériques deviennent de plus en plus sensibles à ces variations et la notion de robustesse commence ainsi à être considérée lors de la conception de systèmes numériques. Par contre, pour les SoCs analogiques, la partie analogique du circuit est très sensible à ces variations et il faut donc les prendre en compte dès la phase de conception du système, ce qui augmente ainsi la complexité de conception de systèmes analogiques. La robustesse ou insensibilité aux variations du processus de fabrication est une notion clef lors de la conception d'un système analogique. Il en découle que le rendement de fabrication ou son corollaire, le taux maximal de circuits défectueux, doit donc être fixé dès la conception. Par exemple, le cahier des charges peut exiger un nombre maximal de 100 circuits défectueux pour un million de circuits produits (100ppm). Cette valeur est fixée en fonction des attentes du client en termes de qualité, du coût du produit et du coût du test. Ainsi, ce nombre sera élevé si le fondeur peut se permettre de mettre place des moyens de test qui détectent tous les circuits défectueux. Par contre s'il souhaite effectuer le minimum de test il faudra concevoir un circuit robuste aux variations du processus de fabrication ce qui revient à imposer un nombre de circuits défectueux fabriqués très faible. Le concepteur doit donc définir son système en fonction des marges de tolérance sur les paramètres du processus de fabrication et en fonction du rendement à atteindre. Chacun des paramètres du processus est défini par une valeur typique P_{typ} , une ou deux valeurs limites P_{min} et P_{max} et le CPk¹². Ce CPk permet de qualifier la capacité du processus de fabrication ; il suppose que chaque paramètre suit une distribution gaussienne et est relié à la valeur de l'écart type (Eq. 2.1). L'ensemble de ces valeurs P_{min} , P_{max} et CPk permet alors de calculer la probabilité de fabriquer des circuits défectueux, cette valeur correspond à la somme des aires grises de la Fig. 2.3.

$$CPk = \frac{MIN(P_{max} - P_{typ}, P_{typ} - P_{min})}{3.\sigma} \quad (\text{Eq. 2.1})$$

¹² Capability Process.

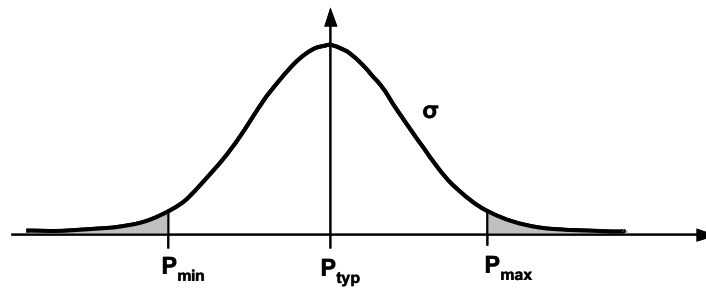


Fig. 2.3. Distribution normale et aire correspondant aux circuits défailant.

Après avoir présenté les particularités et les difficultés de la conception de SoCs numériques et analogiques, nous allons aborder les approches utilisées pour la conception des systèmes AMS&RF.

2.3.3. Flot de conception Bottom-Up pour les SoCs AMS&RF

L'approche classique utilisée pour réaliser un système simple consiste à concevoir individuellement les différents blocs de ce système. Chaque bloc est associé à une fonctionnalité et doit ainsi respecter un ensemble de spécifications fonctionnelles. Le terme spécification fonctionnelle signifie que seules les fonctionnalités sont définies et pas les imperfections provenant de phénomènes physiques parasites. Ainsi, chaque bloc est conçu et vérifié individuellement. Néanmoins, cette approche présente des limites lors de la conception de systèmes complexes. En effet, le problème majeur de cette approche est que le concepteur ne tient pas compte des interactions qui existent entre les différents blocs du système lors de la définition des spécifications. Une fois que chaque partie a été développée, les différentes parties sont assemblées afin de former le système complet. Lors de cette étape d'assemblage, il peut apparaître plusieurs problèmes : le premier vient du fait que les connexions ne sont pas compatibles car elles ne supportent pas le même format. Par exemple, le protocole de communication entre deux blocs est différent, le nombre de sorties d'un premier bloc devant être connecté à un second peut ne pas correspondre au nombre d'entrées du second. Le second problème provient de la complexité du système qui le rend dans sa totalité difficile à simuler et donc à vérifier. Le point critique de cette approche est qu'il n'est pas possible de s'apercevoir des erreurs de conception avant l'assemblage complet du système. En fait, lorsque l'on s'aperçoit d'une erreur, il faut recommencer toute la conception.

Un autre inconvénient de cette méthode est lié à la capacité de fabrication de circuits fonctionnels c'est-à-dire aux dérives de paramètres physiques qui peuvent apparaître lors de la fabrication. Comme la conception du système se fait blocs par blocs, il est difficile de considérer la robustesse globale du système.

Cette approche de conception Bottom-Up a évolué afin de considérer les problèmes de robustesse et les problèmes d'interactions qui peuvent exister entre les blocs analogiques. Pour faire face à ces problèmes, la première étape consiste à définir l'architecture du système en se plaçant au niveau fonctionnel. Cette étape permet tout d'abord de diviser le système en blocs distincts, de définir les marges de tolérances sur les paramètres fonctionnels (P_{typ} , P_{\min} , P_{\max} , CPk) et de définir les interactions entre blocs. Il est ainsi possible de spécifier exactement le nombre d'entrées sorties nécessaires et leur format (protocole, numérique, analogique, plage de fréquence...). Ensuite, de même que dans l'approche précédemment présentée, chacun des blocs est conçu individuellement

en tenant compte des spécifications sur les entrées sorties. Si cette approche permet de réduire les erreurs dues aux interconnexions, elle ne permet pas de résoudre le problème de simulation et de vérification du système complet. En effet, lors de l'assemblage des différents blocs décrits à un niveau d'abstraction proche du niveau physique, les temps de simulation nécessaires à la vérification d'un système complexe sont trop importants. La seule solution permettant de vérifier la conception consiste à faire des mesures directement sur le prototype. Si la conception n'est pas bonne il faudra alors la corriger et créer un nouveau prototype. Cette approche peut nécessiter des temps de développement très coûteux.

2.3.4. Flot de conception Top-Down pour les SoCs AMS&RF

Le flot de conception proposé dans cette partie utilise une approche Top-Down pour la partie conception et une approche Bottom-Up pour la validation ; ces deux principes seront détaillés dans la suite de ce paragraphe (Fig. 2.4) ([2.10] et [2.11]). Nous avons vu que l'approche précédemment proposée, consistant à définir l'architecture au niveau fonctionnel puis à synthétiser chacun des blocs individuellement, présente des limites lors de la validation car elle nécessite des temps de simulation trop longs. La faiblesse majeure de cette approche vient essentiellement du fait qu'il existe une différence trop importante entre le niveau fonctionnel et le niveau circuit. En fait, il faudrait pouvoir simuler un bloc nouvellement conçu en le plaçant au sein du système, c'est-à-dire réaliser une simulation où le bloc conçu est décrit au niveau le plus bas et où les autres blocs sont décrits à un niveau d'abstraction plus élevé. Pour cela, en plus de définir le nombre et le type des entrées sorties (au niveau fonctionnel), il faut également modéliser les caractéristiques électriques (impédances d'entrées sorties, coefficient de réflexion...) de chaque entrée sortie de chaque bloc afin de modéliser correctement le comportement du système. Il est donc nécessaire de modéliser les comportements électriques qui vont avoir un impact sur les blocs qui interagissent avec le bloc à vérifier. Pour cela, un niveau intermédiaire a été ajouté entre le niveau fonctionnel et le niveau circuit, il s'agit du niveau comportemental.

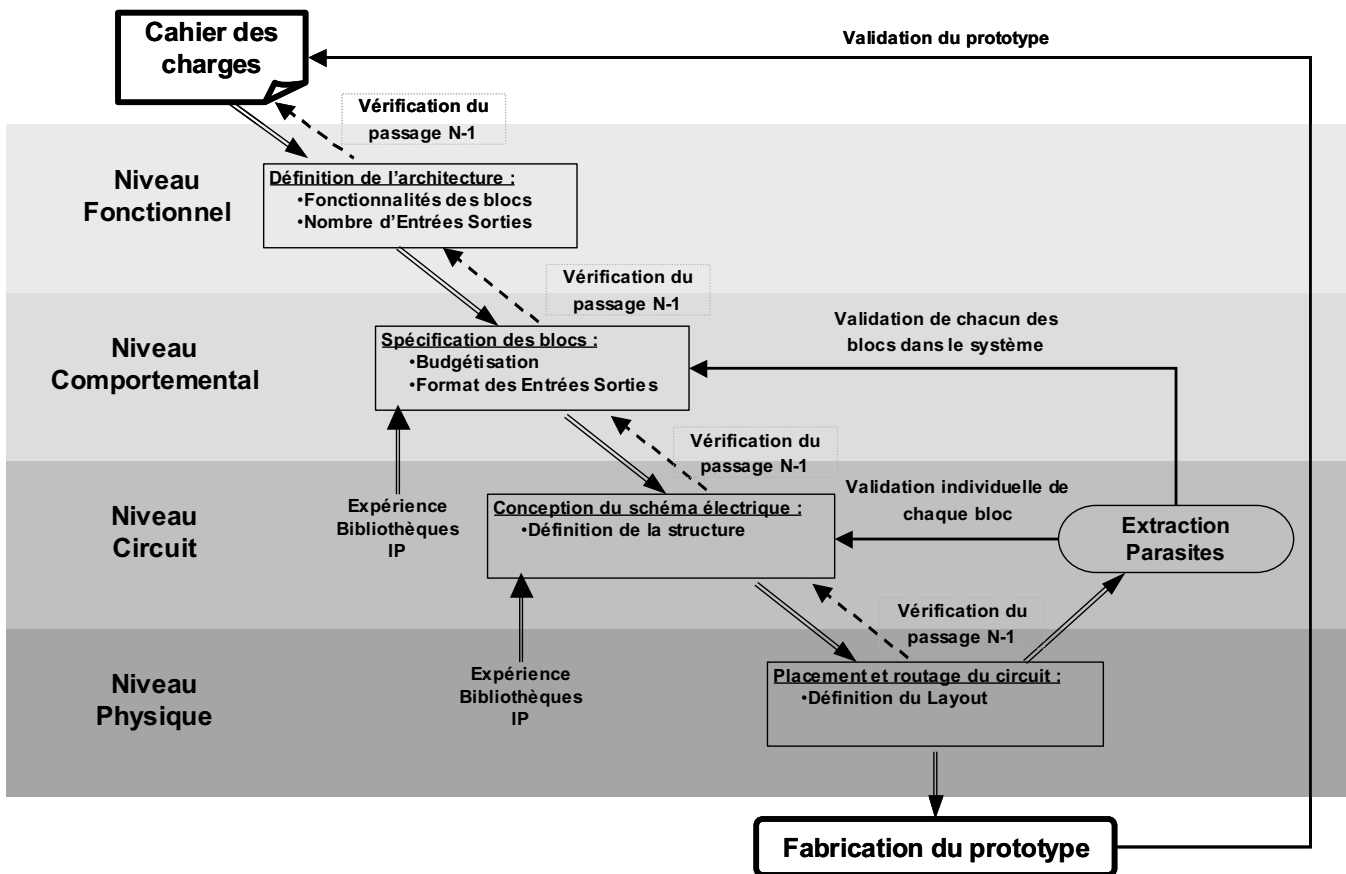


Fig. 2.4. Flot de conception de SoCs AMS&RF.

2.3.4.1. Définition de l'architecture

La première étape de conception consiste à définir l'architecture du système ; cette étape est réalisée au niveau le plus haut. Lors de la conception de systèmes complexes, cette étape permet de séparer les parties matérielles et les parties logicielles, c'est la conception conjointe (co-design) matérielle/logicielle. Cette partie ne sera pas abordée dans ce document car nous traitons uniquement la conception des systèmes matériels. La définition de l'architecture, réalisée le plus souvent par des ingénieurs systèmes, permet d'évaluer plusieurs architectures afin d'identifier la meilleure. La définition de l'architecture permet de diviser le système en plusieurs blocs ; chaque bloc réalise une fonction particulière qui pourra être développée et vérifiée individuellement. Le nombre d'entrées et de sorties des différents blocs est ensuite défini en fonction de leurs interactions avec les autres parties du système. Cette étape se termine en validant l'architecture c'est-à-dire en s'assurant que l'architecture respecte le cahier des charges. Dans [2.12], l'architecture d'un émetteur récepteur RF est développée ; celle-ci est vérifiée par la simulation du système et par la mesure de paramètres haut-niveau tel que le BER¹³.

2.3.4.2. Spécification des blocs

La seconde étape permet de décrire le système au niveau comportemental, elle est réalisée en procédant par raffinement du niveau fonctionnel (Fig. 2.4). Ce niveau comportemental va ainsi permettre de spécifier chacun des blocs avec plus de précision que lors de la modélisation fonctionnelle. En fait, tous les paramètres qui doivent être vérifiés sont définis à ce niveau ; il s'agit

¹³ Bit Error Rate : Taux de bits erronés lors de la modulation et démodulation d'une séquence numérique.

de paramètres décrivant le comportement du système qui peuvent avoir un effet sur les spécifications du cahier des charges. Ces paramètres sont identifiés à partir du plan de modélisation, cette notion est présentée dans la partie §2.3.5.

Ces paramètres comportementaux sont rattachés aux fonctionnalités et aux propriétés non-idéales du système. Les valeurs de ces différents paramètres sont spécifiées durant l'étape de budgétisation. Cette étape consiste à répartir les valeurs des paramètres fonctionnels et les paramètres électriques (non idéaux) dans les différents blocs du circuit. La budgétisation fixe la valeur typique P_{typ} de chaque paramètre, mais aussi leurs valeurs limites P_{min} P_{max} et donne ainsi une quantification de leur robustesse CPk. Comme cité précédemment (Fig. 2.3), ces valeurs limites et l'estimation normale de la distribution des paramètres vont permettre d'estimer le nombre de circuits défectueux qui risquent d'être fabriqués.

L'étape de budgétisation ne peut se faire que grâce à l'expérience du concepteur qui est capable de définir des valeurs réalistes pour les paramètres. En fait, le concepteur doit connaître les limites des blocs qui peuvent être réalisés ou faire appel à des bibliothèques de composants, des blocs existants (IP). Dans ce dernier cas, le concepteur réalise alors la budgétisation en fonction de composants qui existent déjà et donc qui ont été vérifiés lors de conception précédente, on parle alors de « reuse ».

2.3.4.3. Conception du schéma électrique

Suite à la vérification de la spécification du système au niveau comportemental, il est possible de réaliser la conception individuelle de chacun des blocs en respectant les spécifications « comportementales » définies précédemment (Fig. 2.4). Comme il a été énoncé dans la partie §2.3.2, cette étape peut être automatisée lors de la conception de blocs purement numériques dans la majorité des cas mais ne peut pas être automatisée pour les blocs analogiques ou RF. Cette étape est donc généralement réalisée manuellement en faisant appel aux compétences des concepteurs de circuits analogiques. Chacun des blocs est ensuite vérifié individuellement par simulation, c'est-à-dire que tous les paramètres spécifiés lors de la budgétisation sont vérifiés. Ces blocs doivent aussi être validés dans le système complet (Fig. 2.4). Pour cela, il faut réaliser une simulation faisant appel à différents niveaux d'abstraction (simulation multi-abstraction). En effet, comme toutes les interconnexions ont été modélisées au niveau comportemental avec une précision suffisante, alors il est possible de remplacer le bloc à valider par sa description composant (« schéma transistors ») et ainsi de simuler le système complet décrit à la fois aux niveaux comportemental et composant. Cette vérification est possible car il existe des langages de modélisation de systèmes matériels supportant la simulation multi-niveaux tel que le VHDL-AMS¹⁴ [2.13]; nous verrons ces différents langages de description matériel dans la partie §2.4.

2.3.4.4. Réalisation du placement routage

Une fois que les structures électriques des différents blocs ont été vérifiées, il est possible de réaliser le placement des composants et le routage des interconnexions (layout). Pour cela le concepteur utilise des outils d'aide au placement routage. Ces outils sont très performants pour les parties purement numériques mais nettement moins pour les parties analogiques et RF ; cette étape est donc finalisée manuellement pour les parties analogiques. La vérification du placement routage se fait par vérification automatique des règles de dessin. Toutefois, cette étape n'est pas suffisante,

¹⁴ Very high speed integrated circuits Hardware Description Language Analog and Mixed Signal.

il est aussi nécessaire d'évaluer les effets parasites qui apparaissent lors du routage (couplages entre lignes, courants de fuite...), or il n'existe pas d'outil de simulation permettant d'évaluer statiquement ces effets pour des systèmes analogiques complexes. Pour contourner ce problème, il est possible de procéder à une extraction des effets parasites. Cette étape consiste à générer un schéma électrique à partir du niveau physique et à modéliser les effets parasites en ajoutant des résistances, des inductances et des capacités parasites sur et entre les fils. Il est alors possible d'obtenir une description électrique réaliste du système car extraite du niveau inférieur (exemple d'outil : COLUMBUS-AMS). Cette description extraite peut être vérifiée de la même façon que lors de la vérification du schéma électrique : chaque bloc peut être vérifié individuellement ou dans le système complet modélisé à la fois aux niveaux comportemental et composant. Il existe également des outils permettant de réduire la complexité des descriptions extraites afin de diminuer les temps de simulation et de validation des circuits (JIVARO de EDXACT).

Une fois le placement routage validé et vérifié, il est possible de lancer la fabrication d'un prototype. L'étape ultime sera la validation du prototype. Pour cela, des mesures sont réalisées directement sur le prototype et permettent de s'assurer que le SoC conçu respecte le cahier des charges initial.

2.3.5. Vérification de la conception par simulation

L'automatisation de la synthèse logique pour les circuits numériques a été une évolution importante car elle a permis de réduire les temps de conception. Mais, cette automatisation permet également de réduire les risques d'erreur qui peuvent apparaître lors d'une synthèse manuelle.

Pour la vérification des systèmes analogiques, il est indispensable de mettre en place un plan de vérification. Cette notion sera détaillée dans le chapitre suivant, mais il est toutefois nécessaire de décrire ici son principe afin de pouvoir présenter les simulations réalisées lors de la conception de SoCs AMS&RF.

Le plan de vérification est composé d'un plan de simulation et d'un plan de modélisation définis pour chaque niveau de description ([2.8] et [2.14]). La première étape consiste à définir une partie ou une caractéristique spécifique qui doit être vérifiée. Le plan de simulation décrit la procédure de vérification ou le banc de test à mettre en œuvre. Le plan de modélisation détermine les effets qui doivent être modélisés et le niveau de description de chaque bloc en fonction de la caractéristique qui doit être vérifiée. Par exemple, lors de la vérification du gain total d'un système, il faut considérer les différents gains des blocs mais aussi leurs points de compression, leurs impédances d'entrées sorties, leurs coefficients de réflexion... Le modèle doit prendre en compte suffisamment d'effets afin d'assurer la pertinence et la qualité de la vérification ; à l'inverse, il ne doit pas trop compliquer la description car les temps de simulation seraient augmentés inutilement. Pour définir la description, il faut partir d'une description simple (souvent fonctionnelle) et ajouter les effets ou paramètres désirés. La définition des paramètres et donc du niveau de détail du modèle ne peut se faire qu'en connaissant parfaitement les imperfections des blocs et de leurs effets.

La mise en place de plans de vérification permet de réduire le TTM en définissant des procédures de vérification précises afin de permettre la conception de différents blocs en parallèle. En effet, le plan de vérification est accessible à tous les concepteurs. Ainsi, un concepteur de circuits

numériques n'ayant que peu de compétences dans le domaine analogique est capable d'utiliser le plan de vérification afin de s'assurer que les blocs numériques qu'il a synthés fonctionnent avec le système complet c'est-à-dire avec des parties à la fois numériques et analogiques. Le plan de vérification instaure ainsi un dialogue entre les concepteurs de circuits analogiques et numériques.

Cette partie §2.3 a montré l'importance de l'utilisation d'un flot de conception rigoureux pour la génération de SoCs AMS&RF. En effet, lors du développement de SoCs AMS&RF, la conception peut devenir très complexe, il est indispensable de procéder méthodiquement en utilisant plusieurs niveaux hiérarchiques. Ces niveaux permettent de raffiner progressivement le système en validant toute nouvelle description à partir du niveau précédent. La Fig. 2.5 reprend le flot de conception pour des SoCs AMS&RF, elle met ainsi en évidence la nécessité d'utiliser plusieurs niveaux de descriptions correspondant aux différents domaines : numérique, analogique et RF. La vérification de chaque niveau n'est possible que si les outils de simulation supportent la simulation multi-domaines. Nous allons présenter dans la partie suivante les différents niveaux existants ainsi que les outils de modélisation.

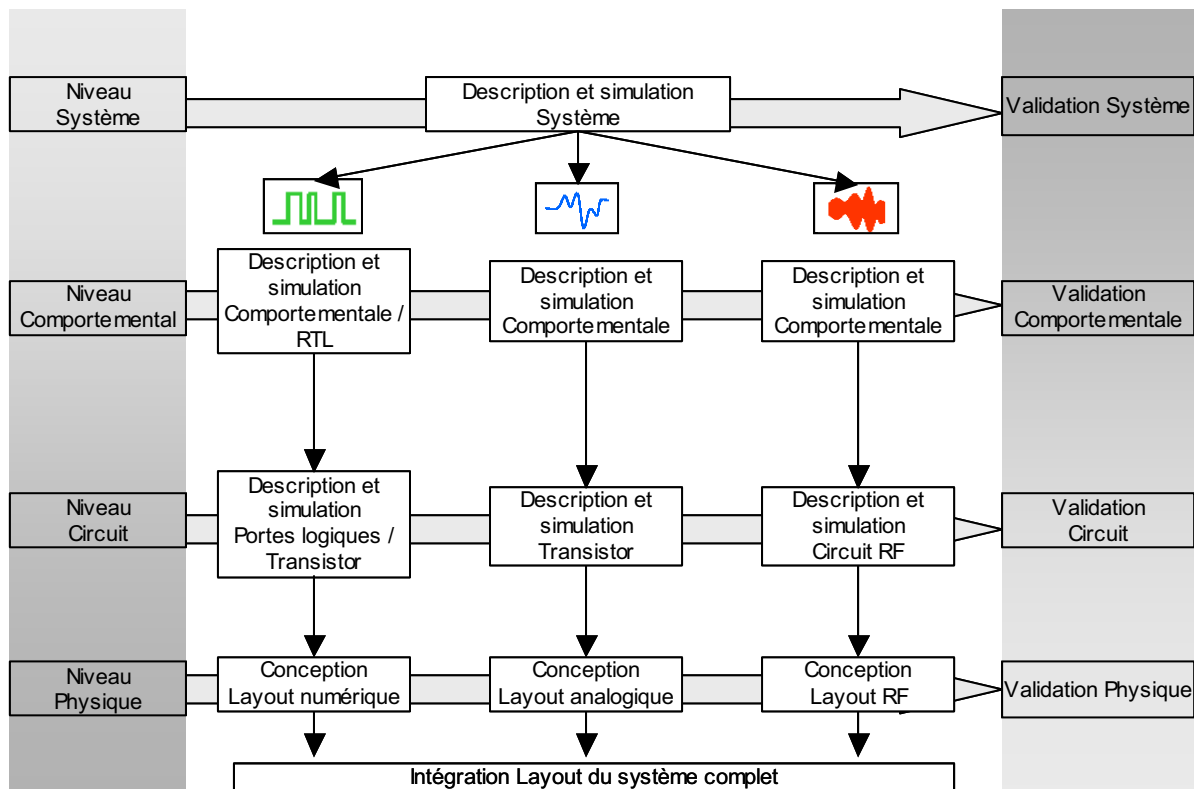


Fig. 2.5. Flot de conception multi-domaines pour des SoCs AMS&RF.

2.4. Différents outils de simulation pour différents niveaux d'abstraction

La Fig. 2.6 présente l'évolution des langages de description et l'augmentation des capacités technologiques en terme de densité d'intégration (courbe). On peut remarquer que la technologie évolue plus rapidement que les langages de description et méthodes permettant la conception de SoCs. Pour compenser cette différence, de nouveaux niveaux d'abstractions ont été spécifiés et de nouveaux langages de description ont été développés. Des outils de simulation leurs sont associés,

ils seront présentés dans la suite de cette partie.

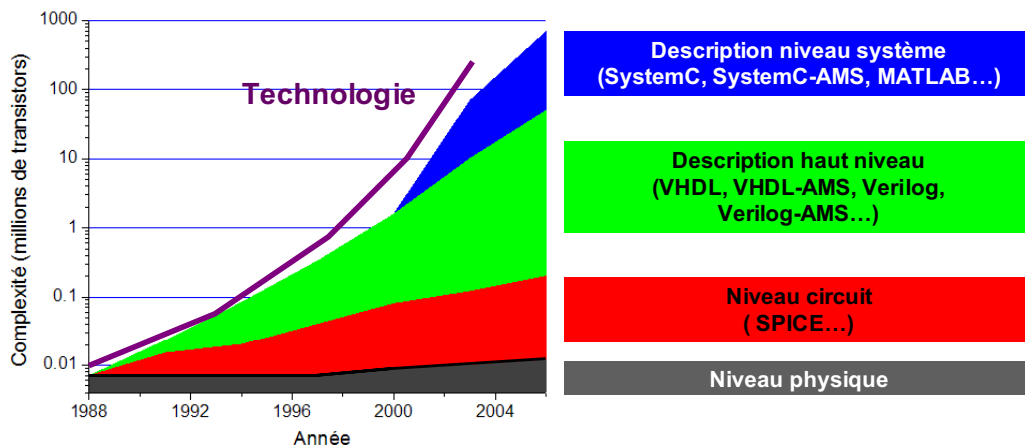


Fig. 2.6. Evolution des langages de description en fonction de l'intégration des transistors [2.15].

2.4.1. Simulation au niveau système ou niveau fonctionnel

Comme énoncé dans §2.3.4, la simulation au niveau fonctionnel est une étape indispensable du flot de conception. Cette étape permet d'évaluer différentes architectures avec des temps de simulation faibles. Il est ainsi possible d'identifier rapidement celles qui sont inutilisables et de déterminer l'architecture la plus adaptée à la réalisation du cahier des charges. Lors de cette étape de description fonctionnelle, le système est divisé en blocs. Ces blocs sont purement analogiques, numériques (mixtes seulement pour les convertisseurs AD, DA) ou logiciels. Un des outils de simulation système le plus utilisé est MatLab/Simulink (de MathWorks) [2.16], il s'agit d'un outil de calcul numérique qui permet de décrire le système sous forme d'équations mathématiques ([2.17], [2.18] et [2.19]). Cependant, il n'est pas nécessaire de définir manuellement toutes les équations mathématiques de chaque bloc car ce logiciel intègre des bibliothèques qui permettent l'implémentation rapide de blocs classiques. C'est ainsi que MathWorks a intégré dans son outil une boîte à outils RF composée de blocs RF mais aussi de sources et d'appareil de mesures. Il est ainsi facile de décrire les parties RF du SoC AMS&RF au niveau fonctionnel. Ptolemy est un autre outil de simulation au niveau système ([2.20], [2.21]). Contrairement à Matlab, il a été développé essentiellement pour la simulation de systèmes électroniques AMS&RF. Ainsi, il possède des descriptions de composants électriques plus complètes et des appareils de mesure plus performants.

Parmi les langages fonctionnels, il existe aussi des langages de descriptions tels que SystemC et SystemC-AMS qui permettent de considérer simultanément les aspects matériels et logiciels. Ces langages permettent, dans un premier temps, de modéliser le système complet, numérique avec SystemC ou analogique et mixte avec SystemC-AMS, et ensuite d'identifier les parties logicielles et matérielles [2.22].

2.4.2. Simulation au niveau circuit

La simulation de systèmes décrits au niveau circuit ou transistor est réalisée grâce à l'utilisation de simulateurs électriques. Pour cela, les parties analogiques et numériques du système sont décrites, dans une « netlist¹⁵ », par des composants élémentaires : transistors, résistances, inductances,

¹⁵ Fichier décrivant les connexions qu'il existe entre les composants du circuit.

capacités... Le simulateur résout alors les équations électriques décrites par les lois de Kirchhoff [2.23]. Le système d'équations obtenu est d'autant plus complexe que le système comprend de composants. A partir de ces équations, plusieurs types d'analyses permettent de calculer la réponse du circuit en fonction d'un type de simulation donnée. Ces analyses sont succinctement décrites dans ce qui suit.

2.4.2.1. *Analyses statique (DC)*

L'analyse statique est aussi appelée analyse DC, elle permet d'étudier le point de fonctionnement du circuit. Sa fonctionnalité est de calculer les tensions et les courants du système électrique lorsqu'il est en régime permanent.

2.4.2.2. *Analyse transitoire (TRAN)*

Cette analyse permet de déterminer la réponse temporelle du circuit pour une durée d'observation finie. Son principe de fonctionnement consiste à résoudre un système d'équations différentielles non-linéaires pour chaque pas sur la durée d'observation.

2.4.2.3. *Analyse fréquentielle (AC)*

L'analyse fréquentielle ou analyse AC permet de calculer la réponse fréquentielle du circuit obtenue à partir de stimuli sinusoïdaux. Cette analyse est basée sur la linéarisation des éléments non-linéaires du circuit autour du point de fonctionnement. Pour cela, il est nécessaire d'avoir préalablement réalisé une analyse DC.

Ces analyses sont adaptées pour l'analyse de systèmes analogiques. Elles sont d'autant plus longues que les systèmes sont complexes, mais surtout que le nombre de pas de simulation est important. Ce second point est souvent le cas lors de l'analyse de systèmes RF. En effet, dans ce type de système ce pas doit être court car il est lié à la période de la porteuse, mais la durée d'observation doit être longue (par rapport à ce pas) car elle est liée à la période du signal modulant. Pour cela, des analyses spécifiques à la simulation de systèmes RF ont été mises en place [2.9]. Ces analyses sont succinctement décrites dans les parties qui suivent.

2.4.2.4. *Analyse périodique ou quasi-périodique (SST)*

Ce type d'analyses, aussi appelées analyses en régime établi (Steady State), est une extension de l'analyse DC précédente pour les systèmes RF. La réponse du système en régime établi est obtenue en appliquant plusieurs stimuli périodiques en entrée du circuit. Cette analyse permet alors de visualiser le spectre fréquentiel en sortie du circuit. Il est ainsi possible par exemple de visualiser l'apparition de raies parasites et donc d'évaluer la distorsion du système. Ce type d'analyse permet d'observer uniquement des signaux quasi-périodiques.

2.4.2.5. *Analyse transitoire de l'enveloppe (MODSST)*

L'analyse par détection d'enveloppe (MODulation Steady State) est un compromis entre l'analyse transitoire qui est difficilement applicable à des systèmes RF et l'analyse périodique qui ne permet pas d'observer de signaux autres que quasi-périodiques. En se plaçant dans le domaine fréquentiel, cette analyse permet de calculer les raies situées autour de la fréquence spécifiée et donc de déterminer l'enveloppe du signal. Le temps de simulation est donc à peu près égal au temps de simulation de l'analyse périodique multipliée par le nombre de points nécessaires à la détection

d'enveloppe ; ce temps est largement inférieur à celui d'une analyse transitoire. Plus concrètement, cette analyse permet de simuler une transmission numérique et de visualiser les effets non-linéaires du système : intermodulation, saturation, temps de montée du signal...

Les cinq analyses présentées ci-dessus sont les principales utilisées, il en existe d'autres telles que l'analyse de bruit, l'analyse par calcul de la fonction de transfert ou l'analyse par calcul des paramètres S du système ; mais nous n'insisterons pas davantage sur ces analyses.

Le langage électrique permettant la description du système au niveau circuit électrique le plus connu et le plus utilisé est le langage SPICE. Il a été développé par l'université de Berkley [2.23]. Des exemples de simulateurs électriques capables de supporter ce niveau de description sont ELDO RF de Mentor Graphics ou Spectre RF de Cadence.

2.4.3. Simulation au niveau bloc ou niveau comportemental

La simulation de systèmes décrits au niveau comportemental présente des différences suivant qu'il s'agit de blocs numériques, analogiques ou RF. Si les parties analogiques et RF peuvent être simulés avec les mêmes types d'analyses que celles utilisées précédemment, l'analyse des parties numériques se fait avec une approche spécifique.

Dans le cas de la simulation de systèmes numériques ou de systèmes à temps discrets, l'analyse utilisée est basée sur les événements. La simulation événementielle permet de diminuer les temps de simulation en calculant la valeur des signaux sur les différents nœuds uniquement lors de l'apparition d'événement sur les entrées. Des langages de description de systèmes matériels ont été développés et standardisés afin de permettre ce type de simulation, il s'agit par exemple de Verilog ou de VHDL. Ce second sera détaillé dans la partie §2.5.1. Des exemples de simulateurs purement numériques sont ModelSIM de Mentor Graphics, Inscisive NC-sim de Cadence, VCS de Synopsys...[2.24]

Lors de la simulation de blocs analogiques ou RF, il n'est pas possible d'utiliser des simulateurs événementiels car les systèmes analogiques et RF sont des systèmes dits « à temps continus ». Les analyses utilisées sont alors les mêmes que celles employées lors de la simulation au niveau circuit (§2.4.2). De la même façon qu'en numérique, des langages de descriptions matériels purement analogiques (AHDL¹⁶) ont été développés dans un premier temps, un exemple de langage est Verilog-A.

Durant la conception de SoCs AMS&RF, il est nécessaire de simuler le système complet, c'est-à-dire de supporter l'interaction entre les blocs numériques, analogiques et RF. Pour cela des langages de description permettant de décrire à la fois des fonctions discrètes et continues ont ainsi été mis en place, c'est le cas de Verilog-AMS et de VHDL-AMS [2.28] (détaillé dans §2.5.2). Les simulateurs présents sur le marché permettant de supporter des systèmes AMS&RF sont : ADVance MS (Mentor Graphics), Virtuoso (Cadence), SMASH AMS (Dolphin Integration)*, Saber (Synopsys)...

Dans cette partie, plusieurs simulateurs utilisés à différents niveaux de modélisation ont été

¹⁶ Analog Hardware Description Languages.

* Partie RF en cours d'intégration dans l'outil.

présentés. Toutefois, les fabricants d'outils de simulation proposent de plus en plus de suites logicielles qui contiennent l'ensemble des outils de simulation nécessaires au cours du flot de conception. Il est ainsi possible de concevoir des SoCs AMS&RF avec la même suite logicielle, l'intérêt étant de diminuer les risques d'incompatibilité entre les différents outils de simulation. Les trois principaux environnements de simulation mixtes et RF sont : RFDE issu de l'alliance entre Cadence et Agilent, l'environnement ADMS RF de Mentor Graphics et la suite Ansoft Designer de la société Ansoft. Il existe aussi des passerelles logicielles permettant de faire le lien entre des simulateurs systèmes avec ces outils comme c'est le cas pour SME (MathWorks [2.26]) qui fait le lien entre Matlab/Simulink et ADMS [2.27].

2.5. Langages de description de SoCs AMS&RF

Lors de la présentation du flot de conception, nous avons mis en évidence la nécessité de décrire les systèmes à différents niveaux d'abstraction. Les principaux langages de description utilisés sont les langages mathématiques explicites, les langages orientés objet et les langages de type HDL.

Les « langages mathématiques » assurent la modélisation des systèmes grâce à l'utilisation d'équations mathématiques. L'outil Matlab est un exemple de logiciel de simulation basé sur ce type de langage.

Les langages de programmation orientée objet permettent de décrire l'architecture et/ou la structure du système. Ce type de langages utilise des bibliothèques de classes C++ permettant de modéliser les systèmes matériels discrets et/ou analogiques. L'utilisation de ces langages permet la conception de systèmes complexes qui demandent le développement conjoint de parties logicielles et matérielles (SystemC-AMS).

Les langages de description de systèmes matériels permettent de décrire l'architecture ou la structure des systèmes. Plusieurs langages de ce type ont été développés afin de décrire le système à différents niveaux d'abstraction. SPICE est un exemple de langage de description matériel permettant la modélisation principalement au niveau composant. Il existe aussi d'autres langages de description matériel supportant plusieurs niveaux d'abstraction, c'est le cas de VHDL-AMS, Verilog-AMS...[2.25].

Dans le cadre de notre étude, nous nous limiterons à la présentation et à l'utilisation du langage de description matériel haut niveau VHDL-AMS.

2.5.1. VHDL

Le VHDL¹⁷ est un langage de description de systèmes matériels qui permet de décrire et de simuler le comportement et la structure de circuits numériques [2.13]. Le langage VHDL est un standard IEEE¹⁸ depuis 1987 sous la dénomination IEEE Std. 1076-1987. Ce langage a été spécialement développé pour la conception d'ASICs¹⁹ et de circuits numériques intégrés complexes.

Une description VHDL est constituée de deux éléments de base : l'entité et l'architecture. L'entité d'une description permet de déclarer les interfaces qu'un circuit possède avec son environnement

¹⁷ Very high speed integrated circuits Hardware Description Language.

¹⁸ Institute of Electrical and Electronics Engineers.

¹⁹ Application-Specific Integrated Circuit.

extérieur. Elle spécifie ainsi les ports du système, le type des signaux utilisés (bit, booléen...) et leurs sens (entrée, sortie, entrée/sortie). La partie architecture de la description définit le fonctionnement interne du système. Les relations entre les différents signaux spécifiés dans l'entité sont définies dans l'architecture.

Le VHDL est un langage textuel qui fait implicitement appel à la notion de description schématique. VHDL supporte l'utilisation de niveaux hiérarchiques car une description VHDL peut devenir un composant élémentaire et être appelé par une autre description. Le principal avantage de ce langage est de pouvoir être utilisé tout au long du flot de conception car il supporte la description à différents niveaux d'abstraction. De plus, il est indépendant de la cible et du fondeur, il permet ainsi la réutilisation de blocs existants ou d'IP.

Ce langage de description est très largement appliqué pour la conception de systèmes numériques, une extension a été mise en place afin de permettre la conception de systèmes analogiques.

2.5.2. VHDL-AMS

Le langage VHDL-AMS est un langage de description de systèmes matériels analogiques et mixtes basé sur le VHDL [2.28]. Le VHDL-AMS a été standardisé en 1999 sous le standard IEEE Std. 1076-1999. VHDL-AMS comme VHDL permet de modéliser des systèmes à événements discrets, multi-abstractions, et hiérarchiques mais il permet aussi de modéliser des systèmes à temps continus et multi-domaines. En effet, il est capable de décrire les systèmes conservatifs électriques, mécaniques, thermiques et hydrauliques ([2.25], [2.29] et [2.30]).

Contrairement au VHDL qui ne supporte que l'analyse événementielle, VHDL-AMS permet également de modéliser et de résoudre les équations différentielles ordinaires non-linéaires continues par morceaux. Dans le cas de systèmes électriques, les quantités représentant les tensions et les courants permettent de caractériser l'énergie qui parcourt les terminaux et les nœuds internes du système. Les équations de Kirchhoff sont utilisées implicitement par le simulateur pour la résolution des équations liées aux composants électriques. VHDL-AMS supporte le calcul des intégrales et des dérivées sur les quantités, il est alors tout à fait possible de lier les différentes quantités sous forme d'équations différentielles afin de modéliser le système.

La principale difficulté pour des simulateurs mixtes est de synchroniser ces quantités à temps continu avec des signaux à temps discret, il existe pour cela des mécanismes permettant la synchronisation des noyaux de simulation numériques et analogiques.

Une extension VHDL-FD²⁰ a été développée afin de permettre la description des équations différentielles dans le domaine fréquentiel en faisant intervenir des instructions séquentielles. Cette extension permet de modéliser des systèmes RF non-linéaires. Le VHDL-FD n'est cependant pas normalisé, il est donc supporté par peu de simulateurs [2.31].

2.6. Modélisation comportementale

Après avoir abordé l'intérêt de la modélisation comportementale (§2.3.4) et les différents langages de description matériel (§2.5), nous allons présenter des approches permettant d'identifier les comportements qui doivent être modélisés dans une description comportementale.

²⁰ Frequency Domain.

La modélisation comportementale laisse à l'utilisateur plus de liberté dans le choix de ses modèles que la description fonctionnelle ou la description au niveau composant. En effet, dans la partie §2.3, nous avons vu que lors de la modélisation fonctionnelle, toutes les fonctionnalités et les paramètres fonctionnels définis par les spécifications doivent être décrits dans la description du système. A ce niveau, les imperfections dues aux phénomènes électriques ne sont pas considérées. Par opposition, lorsque l'on se place au niveau de modélisation composant, la plupart des phénomènes électriques sont pris en compte dans la description. Il n'y a donc pas d'ambiguïté sur ce qui doit être modélisé car la modélisation reproduit le schéma du circuit. Contrairement à ces deux niveaux, la modélisation comportementale laisse au concepteur le choix de ce qui doit être modélisé. Il faut donc être capable d'identifier les paramètres qui ont une influence sur le fonctionnement du système ; cette notion a été définie lors de la présentation du plan de vérification (§2.3.5). La définition de ce plan de vérification est difficile car elle fait appel aux savoir-faire des concepteurs. En effet, seuls les comportements électriques ou comportements non idéaux capables d'influer sur le fonctionnement global du système doivent être modélisés. Si trop de comportements différents sont décrits, les temps de simulations seront trop grands. Au contraire, si des comportements ayant une influence sur le fonctionnement du système ne sont pas modélisés alors le système ne sera pas vérifié correctement.

Il existe deux approches pour obtenir une description comportementale. La première est l'approche ascendante. Dans ce cas, la description est extraite à partir d'une description hiérarchique inférieure. Cette méthode ne peut s'appliquer que dans un flot Bottom-Up, c'est-à-dire lorsque le schéma électrique a été défini ; elle est ainsi utilisée pour la validation (Bottom-Up) de systèmes complexes. Son intérêt est de diminuer les temps de simulation du système complet [2.32].

La seconde approche est basée sur la modélisation descendante. Les descriptions obtenues avec cette approche sont utilisées durant les premières étapes du flot de conception Top-Down. Elles permettent de valider la définition de l'architecture, des spécifications et des interconnexions. La définition de ce type de description se fait à la fois à partir des connaissances des concepteurs systèmes et des concepteurs circuits. Dans [2.25] et [2.33], les auteurs proposent une méthode de modélisation qui consiste à décomposer la description en trois parties différentes : les entrées, les sorties et le comportement reliant les entrées aux sorties. Pour cela, la description doit identifier les variables d'entrées, appliquer la fonction de transfert qui permet de calculer les paramètres de sorties en fonction des entrées et enfin générer les signaux de sorties. Il est important de modéliser correctement le comportement des entrées et des sorties car le principal intérêt de la simulation comportementale est la vérification des interconnexions. Les descriptions comportementales générées par la méthode descendante sont dites génériques car elles sont développées indépendamment d'un composant ou d'un circuit de référence. Elles peuvent alors constituer des bibliothèques d'IP réutilisables pour la conception d'autres systèmes. Ainsi, il existe différentes bibliothèques d'IP en VHDL-AMS ; celles qui sont fournies avec les différents outils de simulation : Commlib_RF de MentorGraphics [2.34], adhLib de Cadence et des bibliothèques publiques proposées par BEAMS [2.35] et l'institut Fraunhofer de Dresde [2.11].

2.7. Conclusion

Ce chapitre nous a permis d'aborder les principes de conception utilisés pour des SoCs AMS&RF. Suite à la présentation de quelques chiffres permettant de mettre en évidence la révolution que subit

le monde de l'électronique ; la complexité et les enjeux de la conception de SoCs ont été mis en évidence. Ainsi, afin d'assister le concepteur lors de la création de systèmes hétérogènes complexes, des méthodes de conception rigoureuses ont été développées. Ces flots de conception ont été mis en place pour diminuer les temps de développement et donc des temps de mise sur le marché (Time To Market). Le développement et la standardisation de langage de modélisation matériel ont permis de diviser la conception en plusieurs étapes ou raffinements successifs. L'apparition de ces langages de description matériel a aussi facilité la simulation et donc la validation de conception des SoCs. Par exemple, le VHDL-AMS est un langage normalisé utilisé tout au long du flot de conception. Il permet entre autre de décrire les SoCs AMS&RF au niveau comportemental. Ce langage sera utilisé dans la suite afin de développer une méthode de qualification et de génération de stimuli de test à partir de ce niveau de modélisation.

Bibliographie

- [2.1] P. Leray, j. Weiss, "Technologies SoCs", Supélec Campus de Rennes, http://www.rennes.supelec.fr/ren/perso/jweiss/fpga/soc_jw/sld001.htm, Février 2002.
- [2.2] R. Barth, "Test and Test Equipement", http://www.itrs.net/Links/2007Winter/2007_Winter_Presentations/03_Test_2007_JP.pdf, Décembre 2007.
- [2.3] J. Hutchby, "Emerging Research Devices", ITRS Spring Public Conference, http://www.itrs.net/Links/2007Spring/Presentations/03_ERD_2007_Annecy.pdf, Avril 2007.
- [2.4] R. Krishnan, "System-on-a-Chip: Technology and Markets", <http://www.electronics.ca/presscenter/articles/48/1/Worldwide-System-on-Chip-SOC-Market-To-Reach-432-Billion-By-2009/Page1.html>, Novembre 2004.
- [2.5] K.S Kundert, "Principles of Top-Down mixed-signal Design", <http://www.designers-guide.org/Design/>, 2003.
- [2.6] B. Bottoms "Assembly and Packaging", ITRS Spring Public Conference, http://www.itrs.net/Links/2007Spring/Presentations/07_Assembly_2007_Annecy.pdf, Avril 2007.
- [2.7] M. Auguin, "Introduction à la conception de SoC", Laboratoire I3S Université de Nice Sophia Antipolis-CNRS, Ecole thématique 2003 Roscoff, 2003.
- [2.8] K.S Kundert, O. Zinke, "The Designer's Guide to Verilog-AMS", Edition Kluwer Academic Publishers, Juin 2004.
- [2.9] K.S Kundert, "Introduction to RF simulation and its application", IEEE Journal of Solid-State Circuits, Vol. 34, No. 9, <http://www.kenkundert.com/pubs.html>, September 1999.
- [2.10] K.S Kundert, "Principles of Top-Down mixed-signal Design", The Designer's Guide Community www.designers-guide.org, 2003.
- [2.11] R. Frevert, J. Haase, R. Jancke, U. Knöchel, P. Schwarz, R. Kakerow and M. Darianian "Modeling and Simulation for RF System Design", Springer, 2005.
- [2.12] S. Vitali, D. Laurentiis, N. Albertazzi, G. Agnelli, F. Rovatti, "Multi-standard simulation of WLAN/UMTS/GSM transceivers for analog front-end validation and design", IEEE International symposium on Wireless Communication Systems (ISWCS'04), pp 16 - 20, Septembre 2004.
- [2.13] <http://www.vhdl.org/>

- [2.14] R. Naiknaware, G. N. Nandakumar, S. R. Kasa, “Automatic Test Plan Generation for Analog and Mixed Signal Integrated Circuits using Partial Activation and High Level Simulation”, IEEE International Test Conference on Designing, Testing, and Diagnostics (ITC’93), pp 139 - 148, Octobre 1993.
- [2.15] E. Sicard et S. Ben Dhia, “Dessin et simulation de fonctions de base en CMOS 90nm”, Colloque sur l'Enseignement des Technologies et des Sciences de l'Information et des Systèmes CETSIS, pp 377, Novembre 2003.
- [2.16] Mathworks, “Matlab and Simulink”, <http://www.mathworks.com>.
- [2.17] N. Chandra, G. W. Roberts, Top-down analog design methodology using Matlab and Simulink”, International Symposium Circuits and Systems (ISCAS’01), pp 319 - 322, Mai 2001.
- [2.18] C. da Rocha Duarte, W.N. do Amaral Pereira, “Computational result for adaptive predistortion applied on linearization of RF power amplifiers”, IEEE Antennas and Propagation Society International Symposium, pp 209 - 212, Juillet 2005.
- [2.19] H. Jie, S. Y. Jun, K. Yongsup, Kim, “System-Level Time-Domain Behavioral Modeling for A Mobile WiMax Transceiver”, Behavioral Modeling and Simulation Workshop (BMAS’06), pp 138 - 143, Septembre 2006.
- [2.20] E. Lee, “Overview of the Ptolemy Project”, University of California, Berkeley, Juillet 2003.
- [2.21] Y. Queré, A. Manchec, C. Quendo, E. Rius, P. Bérardi, L. Crets, J. F. Favennec, C. Person et F. Le Penec, “ Simulation d'une chaîne de transmission numérique sous la plateforme ADS-AGILENT© ”, Colloque sur l'Enseignement des Technologies et des Sciences de l'Information et des Systèmes CETSIS, p349, Novembre 2003.
- [2.22] <http://www.systemc.org/home>.
- [2.23] L.W. Nagel, “SPICE2: A Computer program to simulate circuits”, Memorandum No. ERL-M520, Electronics Research Laboratory, University of California, 1975.
- [2.24] F. GAUTHIER, “Les simulateurs pour circuits numériques”, Electronique International, 2006, <http://www.electronique.biz/Pdf/ELM200602010166068.pdf>, pp 68, 2006.
- [2.25] R. Khouri, « Modélisation comportementale en VHDL-AMS du lien RF pour la simulation et l’optimisation des systèmes RFID UHF et micro-ondes », Thèse de l’INP Grenoble, Mai 2007.
- [2.26] Logiciel Mathlab de The Mathworks, http://www.mathworks.com/products/connections/product_main.html?prod_id=726&prod_name=ADVance%20MS .
- [2.27] B. Nicolle, W. Tatinian, J.-J. Mayol, J. Oudinot, G. Jacquemod, “Top-Down PLL Design Methodology Combining Block Diagram, Behavioral, and Transistor-Level Simulators”, Radio Frequency Integrated Circuits (RFIC) Symposium, pp 475 - 478, Juin 2007.
- [2.28] IEEE 1076.1 Working Group, “Analogue and mixed-signal extension to VHDL-AMS”, <http://www.eda.org/vhdl-ams>.
- [2.29] Y. Hervé, “VHDL-AMS Applications et enjeux industriels”, Cours et exercices corrigés, Dunod, 2002.
- [2.30] A. VACHOUX, "Analog and Mixed-Signal Systems Modeling", http://lsmwww.epfl.ch/design_languages/Model_Sys_Mix/Documents/ch2_vhdl-ams.pdf, notes de cours EPFL, 2005.
- [2.31] Logiciel Rincon de Ridgetop EDA Software, “RINCON™ Harmonic Balance Simulator “, http://www.ridgetop-group.com/docs/rincon_harmonic_balance_new.pdf, Janvier 2004.
- [2.32] H.R. Ghasemi, Z. Navabi, “A new synchronization algorithm for (VHDL-AMS) mixed signal simulation”, IEEE International Symposium on Communications and Information Technology (ISCIT’04), pp 1078 - 1083, Octobre, 2004.

- [2.33] A. Fakhfakh, “Contribution à la modélisation comportementale des circuits radio-fréquence”, thèse de l’université de Bordeaux I, Janvier 2002.
- [2.34] MENTOR GRAPHICS, “CommLib RF VHDL-AMS Library”, Manual, October 2005.
- [2.35] Association BEAMS (Behavioural Modelling of Analogue and Mixed Systems).
https://www.listes.u-bordeaux1.fr/www/info/beams_members.

CHAPITRE III :

METHODES DE VALIDATION DE CONCEPTION

ET DE TEST DE SYSTEMES ANALOGIQUES

MIXTES ET RF

| | |
|--|-----------|
| CHAPITRE III : | 35 |
| METHODES DE VALIDATION DE CONCEPTION ET DE TEST DE SYSTEMES ANALOGIQUES MIXTES ET RF | 35 |
| 3.1. Introduction | 37 |
| 3.2. Objectifs du test de SoCs AMS&RF | 37 |
| 3.2.1. Vérification et validation de conception | 38 |
| 3.2.1.1. Définitions | 38 |
| 3.2.1.2. Erreurs de conception | 39 |
| 3.2.1.3. Méthodes de vérification et de validation de conception | 40 |
| 3.2.2. Test de Production | 42 |
| 3.2.2.1. Définition | 42 |
| 3.2.2.2. Défauts de fabrication | 42 |
| 3.2.2.3. Modélisation des défauts : modèle de fautes | 43 |
| 3.2.2.4. Non détection de certains défauts | 43 |
| 3.3. Méthodes de génération de stimuli pour le test des SoCs AMS&RF | 44 |
| 3.3.1. Test fonctionnel | 44 |
| 3.3.2. Test des spécifications | 45 |
| 3.3.3. Test structurel | 45 |
| 3.3.3.1. Injection de fautes physiques | 46 |
| 3.3.3.2. Simulation de fautes | 47 |
| 3.3.3.3. Qualification des stimuli de test | 50 |
| 3.3.4. Test alternatif | 51 |
| 3.3.5. Application des stimuli de test | 52 |
| 3.4. Conception en vue du test | 52 |
| 3.5. Conclusion | 53 |
| Bibliographie | 54 |
| Fig. 3.1. Différents tests dans le flot de conception et cycle de vie de systèmes AMS&RF. | 38 |
| Fig. 3.2. Flot de vérification [3.8]. | 41 |
| Fig. 3.3. Génération et application de stimuli de test. | 46 |
| Fig. 3.4. Probabilité de détection de fautes PDF. | 51 |
| Fig. 3.5. Technique de test alternatif [3.33]. | 52 |

3.1. Introduction

Le chapitre précédent a mis en évidence les enjeux et les difficultés présents lors de la conception de SoCs AMS&RF. Des méthodes de conception ainsi que des outils d'aide à la conception ont été présentés. La problématique est maintenant de s'assurer que le système ainsi développé correspond aux attentes du client. Ceci est d'autant plus important que la conception de systèmes de plus en plus complexes augmente les risques d'erreurs.

Pour cela, il est indispensable de mettre en place des procédures de test pendant la conception et après la fabrication du circuit. On parlera de vérification ou de validation quand il s'agira de s'assurer de la cohérence des descriptions avec les spécifications et de test matériel quand il s'agira de s'assurer qu'un système a été correctement fabriqué.

Chacun de ces types de vérifications ou tests fait référence à des caractéristiques distinctes et précises du système, celles-ci sont abordées dans la première partie de ce chapitre. Nous définirons ainsi les termes adéquats propres à chacun de ces domaines puis nous aborderons les origines des erreurs entraînant la nécessité de réaliser ces tests. Les différentes méthodes permettant de tester le système seront ensuite présentées. Pour clore ce chapitre, nous aborderons les différentes approches de génération de stimuli de test et les méthodes permettant d'augmenter la testabilité du système.

3.2. Objectifs du test de SoCs AMS&RF

De manière générale, le test est l'action qui consiste à s'assurer que les caractéristiques d'un système correspondent aux performances qui ont été définies dans le cahier des charges. Cependant, le test présente différentes particularités suivant l'étape du flot où il est réalisé. La Fig. 3.1 met en évidence les différents types de test utilisés lors de chaque étape du cycle de vie de SoCs AMS&RF. En effet, s'il est exécuté durant la conception du système, les termes validation ou vérification sont alors employés, l'objectif est alors de s'assurer que le système développé correspond au cahier des charges. Lorsqu'il est réalisé durant l'étape de production, on parle de test de production. Ces deux finalités du test sont détaillées dans la suite de cette partie. D'autres tests peuvent aussi être appliqués afin de tester les systèmes dans leur environnement de fonctionnement, les termes employés sont test de maintenance, test sur site, test en ligne... mais ces notions ne seront pas abordées.

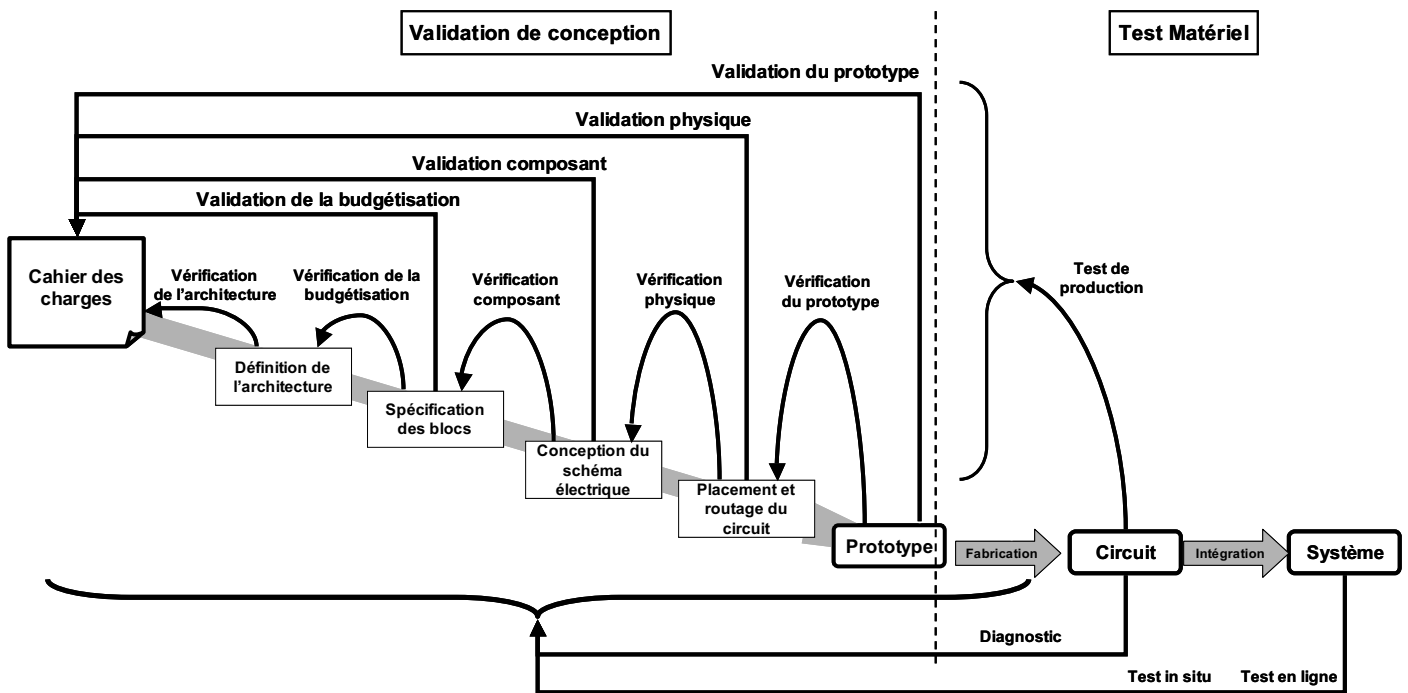


Fig. 3.1. Différents tests dans le flot de conception et cycle de vie de systèmes AMS&RF.

3.2.1. Vérification et validation de conception

La vérification et la validation sont des étapes importantes de la conception des systèmes intégrés. En effet, ces étapes peuvent représenter de 40% à 70% du coût des SoCs AMS&RF [3.1]. Une autre contrainte importante de la conception est le TTM¹. De ce fait, les concepteurs cherchent à développer le plus rapidement possible leur système, c'est-à-dire cherchent à le concevoir « du premier coup » sans erreur (« First Time Right Design »). Or, l'hétérogénéité des SoCs analogiques mixtes et RF augmente considérablement la difficulté de leur conception ; il est ainsi courant de devoir revenir en arrière (« re-spin ») durant les phases de conception. Cette pratique peut entraîner un TTM inacceptable. De ce fait, pour diminuer le TTM, il faut appliquer des méthodes de vérification et de validation de conception rigoureuses afin d'éviter ces « re-spins ». Comme il n'est pas possible de vérifier exhaustivement toutes les fonctions des systèmes, il faut aussi mettre en place des méthodes permettant d'évaluer la qualité des tests choisis. Il est ainsi nécessaire de recourir à des métriques qui permettent de déterminer si le système a été suffisamment vérifié. Dans cette thèse, différentes méthodes ont été développées afin de qualifier les tests effectués au cours de différentes étapes de conception et de fabrication. Ces méthodes prennent en compte la contrainte du temps d'exécution ; contrainte particulièrement présente dans le cadre de l'étude de systèmes complexes. Après avoir présenté les différences entre vérification et validation de conception, nous discuterons de l'origine des erreurs qui peuvent apparaître lors de la conception. Nous pourrions alors présenter le principe de fonctionnement des différentes méthodes de vérification capables de détecter ces erreurs.

3.2.1.1. Définitions

Les étapes de vérification et de validation de conception sont nécessaires tout au long du flot de conception des systèmes. En effet, ces étapes sont indispensables pour garantir l'absence d'erreur

¹ Time To Market.

lors de la conception en particulier de systèmes AMS&RF.

Les principes et les particularités du flot de conception des systèmes AMS&RF ont été détaillés dans le chapitre précédent (§2.3). La Fig. 3.1 rappelle la méthode utilisée pour la conception des SoCs AMS&RF, il s'agit d'une approche Top-Down pour la conception et Bottom-Up pour la validation [3.2].

Les notions de vérification et de validation sont différenciées par le fait qu'elles ne s'appliquent pas au même moment lors de la conception des systèmes.

Les étapes de *vérification* sont réalisées lors de la conception Top-Down (Fig. 3.1). Cette étape permet de répondre à la question : Construisons-nous bien le bon circuit ? [3.3] Elles permettent par exemple de s'assurer que les spécifications de chaque bloc composant le système ont correctement été définies, c'est-à-dire que le raffinement de la description initiale du système a été correctement réalisé. Pour cela, la description du système définie à un nouveau niveau d'abstraction est comparée avec la description du niveau précédent. Une procédure est donc mise en œuvre afin de vérifier chaque passage d'un niveau d'abstraction à un autre niveau plus détaillé.

La *validation* permet de répondre à la question : Avons-nous construit le bon circuit ? [3.3] Elle peut être appliquée dans le flot Top-Down afin de valider que chaque étape de conception respecte toujours le cahier des charges. Elle est le plus souvent appliquée lorsqu'il n'est pas possible de valider le système complet au niveau d'abstraction le plus bas. Dans ce cas, la validation de conception est mise en place dans un flot Bottom-Up. Ainsi, l'étape de validation est réalisée lorsque le système a complètement été développé c'est-à-dire lorsqu'il a été décrit au niveau physique (à la fois son implémentation matérielle et son placement/routage). Il est alors possible de valider la conception du système global par simulation de celui-ci à différents niveaux d'abstraction. Les différentes descriptions utilisées sont alors extraites à partir des descriptions de plus bas niveaux [3.4].

Pour valider la conception, on effectue souvent des mesures directement sur un prototype Silicium du système. Cette étape permet de valider ce qui n'a pas pu être validé par simulation à cause de temps de simulations trop élevés. Néanmoins, cette méthode est à la fois coûteuse et chronophage : des re-spins à ce niveau de la conception ont un impact direct sur le TTM.

3.2.1.2. Erreurs de conception

Avant de présenter les différentes méthodes utilisées pour la vérification et la validation de conception, il est important d'identifier les principales causes des erreurs pouvant apparaître lors de la conception de SoCs.

Les erreurs de conception sont principalement dues à un manque ou une mauvaise communication entre concepteurs. En effet, les ingénieurs de conception doivent développer les systèmes en répartissant les problèmes sur plusieurs équipes car :

- ces systèmes sont de plus en plus complexes et font appel à des compétences multiples,
- la conception de systèmes avec différentes équipes (numériques, analogiques et RF) travaillant en parallèle permet de diminuer le TTM.

Une incompréhension entre ces équipes, peut entraîner par exemple le développement de blocs qui sont incompatibles entre eux. Un autre type d'erreur humaine est dû à des erreurs individuelles qui entraînent une mauvaise définition d'une partie du système ; par exemple, un mauvais dimensionnement d'un paramètre.

Afin de limiter au maximum ces risques d'erreurs et de détecter les erreurs le plus tôt possible, il est indispensable d'utiliser des méthodes de conception rigoureuses. La conception doit alors être divisée en plusieurs étapes et chacune de ces étapes doit être complètement vérifiée avant de passer à l'étape suivante. Il faut aussi mettre en place des procédures permettant d'établir ou de standardiser des liens entre les différentes équipes afin de réduire le manque de communication [3.2].

3.2.1.3. Méthodes de vérification et de validation de conception

Le développement de méthodes de vérification et de validation de conception pour les circuits intégrés a suivi la même évolution que celle des méthodes et des outils de conception (§2.3.2). Ainsi, les méthodes de test ont initialement été développées pour des SoCs purement numériques puis elles ont évoluées pour le test de SoCs AMS&RF.

3.2.1.3.1. Analyse par simulation

L'analyse par simulation est une méthode de vérification et de validation couramment utilisée. Elle permet d'évaluer, à moindre coût, différentes solutions de conception en s'assurant qu'elles respectent les spécifications définies dans le cahier des charges. Initialement développée pour des SoCs purement numériques, l'apparition de simulateurs performants (multi-domaines, multi-niveaux) et de langages de description haut niveau et mixtes ont permis de l'étendre à l'analyse de SoCs AMS&RF (§2.4.).

Cette méthode permet de vérifier la conception des systèmes. Par exemple, la simulation du système avec des blocs décrits au niveau fonctionnel permet la vérification de l'architecture. Dans [3.5], l'auteur propose des bancs de test permettant la validation de conception au niveau système en effectuant des mesures fonctionnelles (qualité de la modulation, sensibilité, tension maximale de sortie, réjection des canaux adjacents...). De même, la simulation du système au niveau composant permet de vérifier la conception du schéma électrique.

De plus, l'apparition de simulateurs et de langages de description multi-niveaux a permis de simuler le système suivant à la fois différentes configurations et différents niveaux de précision. L'intérêt de la simulation à différents niveaux d'abstraction est de permettre la vérification de la conception d'un bloc en simulant la description extraite de ce bloc dans le système complet décrit à un niveau d'abstraction plus élevé ([3.6], [3.7]). Le choix du niveau de description dépend alors des caractéristiques qui doivent être vérifiées. La principale difficulté de cette approche est de déterminer les spécifications qui doivent être vérifiées et de définir les stimuli de test utilisés. Pour cela, un flot de vérification doit être mis en place afin d'identifier correctement ce qui doit être vérifié, il est présenté dans la partie suivante.

3.2.1.3.2. Plan de vérification

La définition d'un plan de vérification concerne une méthode appliquée lors de l'analyse de systèmes complexes. Cette technique a été développée afin de mettre en place une méthode de vérification rigoureuse qui permet de s'assurer que le système respecte les spécifications définies

dans le cahier des charges. Le principe a été rapidement abordé dans le chapitre précédent (§2.3.5). Dans cette partie, nous développerons les différentes étapes du plan de vérification.

Dans [3.8], Kundert décrit le principe du flot de vérification ; ce principe est représenté par le schéma Fig. 3.2. Le plan de vérification s'inscrit dans le flot de vérification. Ce plan est déterminé à partir du cahier des charges, mais il doit aussi tenir compte des risques pouvant apparaître lors de la conception afin de les éviter. Ces risques sont identifiés par les concepteurs et ont été présentés dans la partie §3.2.1.2. Le plan de vérification définit la manière de vérifier le système. Les plans de modélisation et de simulation permettent de spécifier le niveau de description de chacun des blocs, ils sont directement dérivés du plan de vérification. Ainsi, en définissant les caractéristiques d'un bloc à vérifier, il est possible de déterminer le niveau de modélisation de chacun des blocs simulés. Lorsque la vérification se fait sur un bloc précis, alors celui-ci est décrit au niveau composant. Les blocs n'ayant pas de rapport direct avec le bloc vérifié peuvent être décrits au niveau fonctionnel tandis que les blocs qui interagissent avec le bloc à vérifier sont décrits au niveau comportemental ou au niveau composant. Le plan de simulation définit quant-à lui le banc de test et les différentes configurations de simulation utilisées. Le bloc de génération du banc de test (Fig. 3.2) permet de mettre en place les simulations et de récupérer les résultats automatiquement. Ceux-ci peuvent alors être comparés aux résultats décrits dans le cahier des charges. Il est alors possible de déterminer si le système a été correctement conçu et respecte les spécifications du cahier des charges.

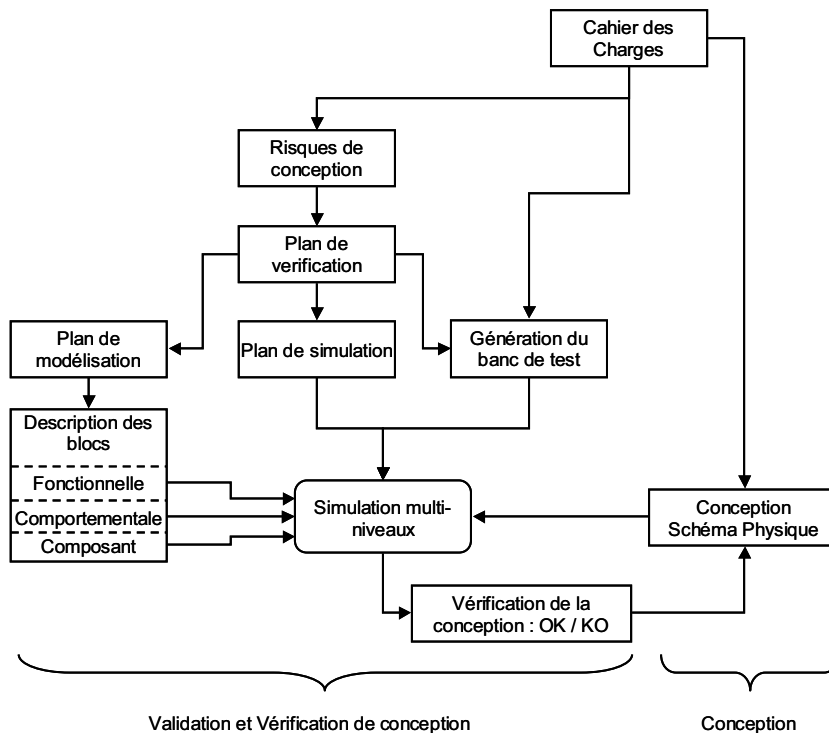


Fig. 3.2. Flot de vérification [3.8].

3.2.1.3.3. Analyse physique

Une des méthodes de validation de conception est l'analyse physique. Cette méthode est uniquement appliquée lors de l'étape de validation finale. Le principe consiste à mesurer directement sur un prototype les spécifications du circuit. Cette approche permet de tenir compte et de mesurer tous les effets du circuit. Cette méthode permet par exemple de mesurer les temps de propagation, les interférences... Avec cette approche, il est possible de mesurer presque instantanément des effets qui demanderaient plusieurs semaines de simulation. Toutefois, comme

énoncé précédemment, elle nécessite la fabrication d'un prototype et peut entraîner un surcoût de conception si ce premier prototype n'est pas bon.

Comme lors de la vérification par simulation, la définition des mesures à réaliser est un point important. En effet, s'il est facile d'identifier les spécifications à vérifier pour le cas de systèmes simples, il est impossible de vérifier toutes les configurations possibles et caractéristiques physiques de SoCs complexes.

3.2.2. Test de Production

3.2.2.1. Définition

Une fois que le système a été conçu et validé, il peut être produit à grande échelle. Toutefois, l'étape ultime avant de pouvoir vendre le circuit au client consiste à garantir que celui-ci ne contient pas de défauts : c'est l'objectif du test de production (Fig. 3.1). Le test de production permet de s'assurer que le composant électronique mis sur le marché donnera satisfaction au client en termes de fonctionnalités et de fiabilité. Sa fonction est donc de rejeter les circuits produits qui sont défectueux ou qui pourraient le devenir ([3.9] et [3.10]). Le test de production peut être réalisé avant la fin de la fabrication du circuit. En effet, il est possible de réaliser le test des composants directement sur la tranche de Silicium (wafer). Le wafer est alors testé en utilisant un outil de test automatique (ATE²) qui contient des ressources de test. Le « prober » est une partie de l'ATE qui permet de positionner la puce sous les pointes de test situées sur la carte à pointes. Ce prober permet de connecter grâce à la carte à pointes les appareils de mesures aux circuits présents sur le wafer. Une fois les circuits testés puis découpés, ils sont insérés dans leurs boîtiers, c'est l'encapsulation. Il est alors possible de procéder au test final du circuit afin d'identifier les circuits qui peuvent être vendus. Bien que le test final soit indispensable, il peut être intéressant de tester chaque circuit durant les différentes étapes de fabrication. En effet, plus un circuit défectueux est détecté tardivement, plus les pertes financières relatives à son rejet sont importantes car les coûts de mise en boîtier (transport, matière, main d'œuvre) ne sont pas négligeables. De plus, dans le cas de la fabrication de SiPs, le test final rejetterait non seulement la puce défectueuse mais toutes les autres puces contenues dans le boîtier.

Les contraintes du test de production ne sont pas les mêmes que celles de la validation ou de la vérification de conception. La principale contrainte de ce type de test est le coût du test. Ce coût peut atteindre jusqu'à 50% du prix de revient du circuit final [3.11]. En effet, le test de SoCs AMS&RF demande l'utilisation d'appareils de test complexes et performants donc coûteux. Par exemple, le prix d'un testeur AMS&RF est de l'ordre du million de dollars ce qui représente un coût horaire de 60\$. Le temps accordé au test d'un circuit est donc un facteur qui va influencer sur le coût du test car il va immobiliser le testeur pendant un certain temps. Ainsi lors de la définition de stimuli pour le test de production, il est indispensable de considérer le nombre de stimuli nécessaires et la difficulté à appliquer ces stimuli ou à effectuer des mesures sur le système. Pour résumer, le test de production doit être le plus efficace, le plus rapide, le moins coûteux tout en garantissant le niveau de qualité désiré.

3.2.2.2. Défauts de fabrication

Le test de production est utilisé pour détecter les défauts de fabrication. Dans cette section, nous ne

² Automatic Test Equipment.

parlons pas d'erreurs apparues lors de la conception mais de défauts physiques susceptibles d'apparaître lors de la fabrication [3.9]. Notons que les erreurs de conception ont théoriquement dû être détectées lors de la validation de conception.

Il existe deux types de défauts physiques : les défauts locaux et les défauts globaux.

Les défauts locaux, ou ponctuels (« spot defects »), peuvent être dus à la présence de poussières apparues sur les tranches de silicium ou sur les masques. Effectivement, ces poussières peuvent apparaître malgré les procédures mises en place. Ces défauts peuvent être modélisés par exemple sous forme de court-circuits, circuits-ouverts... Ils ont une probabilité d'apparition aléatoire.

Les défauts globaux sont liés aux variations des paramètres du processus de fabrication. En effet, lors de la fabrication de circuits, il peut se produire des dérives progressives sur les réglages des machines (four à diffusion, implantateur ionique...) entraînant une dérive sur des paramètres technologiques (épaisseur d'oxyde, largeur des lignes...) qui entraînent à leur tour des dérives sur des paramètres électriques (valeur des résistances, tensions de seuils...). Le problème pour détecter ce type de défauts est que ces variations peuvent être faibles et justes à l'extérieur de leurs plages de spécification. Contrairement au test structurel des circuits numériques, le test structurel des circuits analogiques repose essentiellement sur la détection de ce type de défauts. En effet, les systèmes numériques sont en général plus robustes aux variations paramétriques que les circuits analogiques.

3.2.2.3. Modélisation des défauts : modèle de fautes

Il est important de faire la différence entre un défaut et une faute. En fait, la faute modélise l'effet d'un défaut. Par exemple, l'apparition de défauts locaux peut entraîner les effets électriques suivant : court-circuits, circuits-ouverts mais également d'autres modifications de paramètres électriques structurels du circuit. Les fautes induites par ces défauts locaux peuvent être classées en deux types de fautes [3.12] :

- Les fautes catastrophiques correspondent à des court-circuits, circuits ouverts ou toutes variations importantes de paramètres niveau composant (par exemple, valeur d'une résistance, ou rapport w/l d'un transistor).
- Les fautes paramétriques correspondent à de faibles variations des paramètres niveau composant.

Evidemment, les variations globales des paramètres du processus de fabrication sont principalement modélisées par des fautes paramétriques.

3.2.2.4. Non détection de certains défauts

Si le test matériel repose uniquement sur la mesure des spécifications, il n'est pas alors toujours possible de détecter tous les défauts apparaissant lors de la fabrication. En effet, certains défauts qui engendrent des variations de paramètres électriques locaux en dehors de leur bande de tolérance ne produisent pas forcément de variations sur les paramètres des spécifications.

Ce phénomène est possible si l'on suppose que les bandes de tolérances des paramètres niveau composant ne sont pas parfaitement définies vis-à-vis des spécifications globales.

Ce phénomène apparaît généralement dès lors que l'on définit les fautes au niveau $n-1$ et qu'on observe leurs effets au niveau n . Par exemple, niveau composant pour les défauts et niveau circuit pour l'observation des spécifications.

Dans le cas particulier des systèmes composés de macro-composants, la définition des bandes de tolérance ou budgétisation des paramètres des macro-composants est une tâche très complexe. Très souvent les concepteurs choisissent de rendre ces systèmes « robustes » à des variations de paramètres de macro-composants en dehors de leurs bandes de tolérances. Ainsi, par le biais de la mesure des spécifications du système, de nombreuses variations locales resteront toujours indétectables.

3.3.Méthodes de génération de stimuli pour le test des SoCs AMS&RF

Dans la partie précédente, nous avons présenté des méthodes de test mises en place durant le flot de conception des SoCs. Ainsi, nous avons abordé les objectifs, les finalités et les caractéristiques propres à chacun des tests appliqués durant la conception et la fabrication des circuits [3.9].

Cette partie a pour objectif de présenter les principales méthodes utilisées pour générer des stimuli de test. La première méthode pour générer des stimuli consiste à utiliser le cahier des charges du système pour déterminer les fonctionnalités à tester, c'est le test fonctionnel.

La méthode de test des spécifications est sensiblement identique au test fonctionnel, mais elle utilise le cahier des charges de l'architecture du système (défini après budgétisation) pour générer des stimuli de test. Ces deux méthodes de génération de stimuli de test peuvent être appliquées pour la validation de conception ou le test de production.

La troisième méthode est le test structurel, elle permet de déterminer des stimuli de test en fonction de leurs capacités à détecter la présence de défauts physiques mais aussi d'erreurs de conception. En fait, même si cette approche est essentiellement associée au test de production, elle peut aussi être appliquée à la validation de conception car elle permet d'évaluer la capacité des stimuli à détecter des erreurs de conception.

Le test alternatif est la dernière méthode de test présentée, son principe consiste à mesurer les spécifications de manière indirecte afin d'optimiser les stimuli.

3.3.1. Test fonctionnel

Le test fonctionnel est une méthode de test largement utilisée pour le test de SoCs. L'objectif est de valider le fonctionnement du système et plus précisément de s'assurer que le système respecte les fonctionnalités décrites dans son cahier des charges du système. Dans [3.5], l'auteur propose de vérifier l'architecture développée en modélisant et en simulant un système RF au niveau fonctionnel. La vérification fonctionnelle est réalisée en procédant à un test fonctionnel c'est-à-dire en s'assurant que le système correspond aux caractéristiques définies par les standards de communication. Par exemple, lors du test fonctionnel d'un système W-CDMA, il faut s'assurer que celui-ci est capable de récupérer la séquence numérique d'un signal RF modulée au format IQ avec une probabilité d'erreur suffisamment faible (Taux Erreur Bit). Ainsi le signal modulant doit pouvoir être extrait lorsque le signal reçu est supérieur à une puissance minimale et lorsqu'il est perturbé par un bruit dont le niveau est inférieur ou égal à celui spécifié par la norme.

Lors du test fonctionnel, la génération de stimuli est réalisée, à partir du cahier des charges système, en déterminant des stimuli applicables sur le système c'est-à-dire en définissant le signal à envoyer sur les entrées du système et la réponse que l'on attend.

Cette approche est intéressante pour le test de systèmes simples, mais elle est plus difficilement

applicable pour des systèmes complexes comprenant un nombre important de fonctionnalités. En effet, la tendance à la miniaturisation conduit à l'intégration d'un nombre de fonctionnalités de plus en plus important et des fonctionnalités souvent de plus en plus complexes ; il est donc très difficile voire impossible de vérifier toutes ces fonctionnalités. De plus, seul le test de toutes les fonctionnalités avec l'ensemble des stimuli possibles (test exhaustif !) permettrait d'assurer la qualité du système.

3.3.2. Test des spécifications

Le test des spécifications est proche du test fonctionnel. L'objectif n'est plus ici d'évaluer les fonctionnalités du système, mais d'évaluer des caractéristiques définies à un niveau d'abstraction inférieur, par exemple, décrites dans le cahier des charges de l'architecture. Dans le cas d'un système W-CDMA, une des caractéristiques à évaluer est la réjection des fréquences parasites. Ces fréquences parasites sont induites par l'architecture matérielle (ex : raies parasites dues à l'oscillateur local) et ont une incidence sur le fonctionnement du système. En fait, si la réjection n'est pas suffisante, l'ACPR³ sera supérieur à la valeur définie par la norme et le système ne pourra pas extraire le signal modulant.

Le test de cette spécification permet de tester indirectement le fonctionnement du système. Pour permettre la définition des spécifications, le concepteur identifie dans un premier temps l'architecture à partir du cahier des charges du système. Il est ensuite capable de déterminer les caractéristiques qui sont importantes pour la spécification du système à partir de l'architecture matérielle précédemment déterminée. Ces caractéristiques sont alors spécifiées par une valeur typique et une marge de tolérance. La vérification de ces spécifications est donc essentielle pour s'assurer indirectement que le système peut réaliser les fonctions définies par le cahier des charges.

3.3.3. Test structurel

Le test structurel consiste à vérifier l'intégrité de chaque composant du système sans considérer ses fonctionnalités. L'objectif du générateur de stimuli est de détecter des fautes qui ont été insérées volontairement dans la structure du système. La Fig. 3.3 présente un schéma résumant la génération et l'application des stimuli de test par une approche structurelle [3.10]. La partie de gauche représente les étapes nécessaires à la génération mais surtout à la qualification des stimuli de test. Pour cela, des fautes sont injectées dans la description structurelle du circuit sous test. La description fautive est ensuite simulée grâce à l'utilisation d'un stimulus ; stimulus qui est généralement généré automatiquement. Les résultats de simulation ainsi obtenus permettent d'évaluer la qualité du stimulus c'est-à-dire de déterminer s'il est capable de détecter ces fautes. Le stimulus peut alors être utilisé pour le test du circuit afin de détecter des défauts physiques. Dans le cas du test matériel, le stimulus permettra de configurer les équipements de test pour permettre l'application d'excitations et de mesures du circuit afin de valider sa fabrication.

³ [Adjacent Channel Power Rejection.](#)

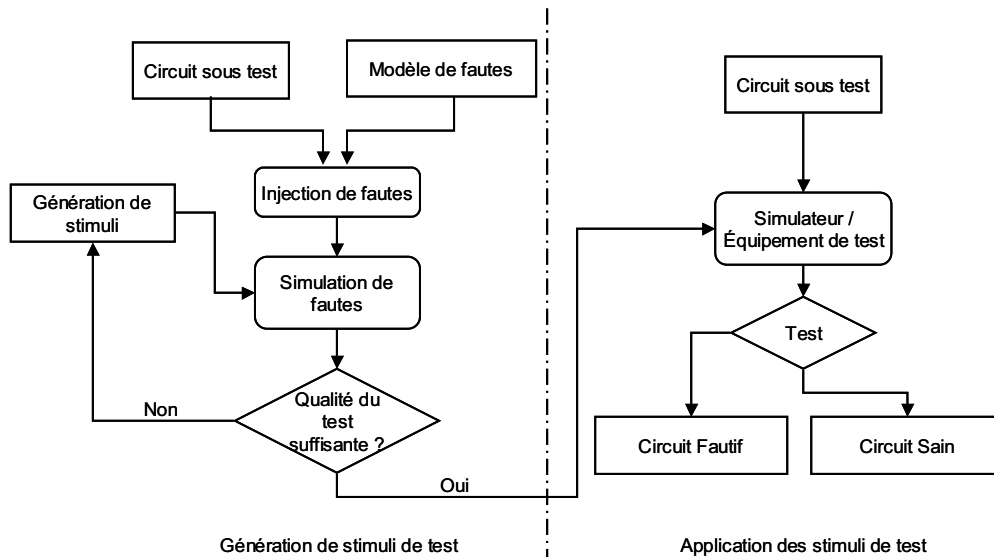


Fig. 3.3. Génération et application de stimuli de test.

Avant d'aborder les méthodes utilisées pour la génération de stimuli de test, il est important de définir la notion de stimuli. En fait, dans le domaine du test numérique, le terme vecteur est utilisé. Un vecteur est un ensemble de données binaires discrètes qui est appliqué aux entrées du circuit numérique sous test. Pour le test de systèmes AMS&RF, il faut également définir les différents paramètres numériques permettant de configurer le système. Par exemple, lors de l'utilisation d'un système RF, certaines caractéristiques peuvent être configurées comme le gain des amplificateurs; il s'agit dans ce cas de valeurs numériques. Hormis ces paramètres numériques, lors de la définition de stimuli de test pour les systèmes AMS&RF, il faut aussi définir les signaux continus qui sont envoyés sur certaines entrées du circuit. Pour cela, il est possible dans certains cas de discrétiser le signal analogique afin de l'appliquer au système. Le problème de cette approche est qu'elle n'est pas applicable pour des systèmes RF. Au lieu d'utiliser des signaux discrétisés, il est possible de spécifier uniquement les paramètres analogiques des signaux à appliquer. Ainsi lors de l'application d'une sinusoïde, il suffit de définir l'amplitude, la phase et la fréquence du signal. Lors de l'utilisation de signaux modulés, il est possible de définir la séquence numérique modulée et le type de modulation appliquée... Par conséquent, lors du test de SoCs AMS&RF, le terme stimuli est préféré : ce terme comprend les valeurs des paramètres de configuration du système et les valeurs des paramètres décrivant les signaux continus.

3.3.3.1. Injection de fautes physiques

Une méthode de génération de stimuli de test par injection de fautes consiste à injecter physiquement des fautes sur le circuit. La modification physique du circuit peut se faire par radiation ionique, par utilisation d'interférences électromagnétiques, par laser... Ces techniques sont très coûteuses à mettre en place et la miniaturisation des SoCs rend cette méthode d'injection impossible à mettre en place. Le développement d'outils de simulation a fortement diminué la pratique de cette approche, mais elle reste toutefois utilisée pour valider la conception de systèmes devant résister à des attaques physiques comme c'est le cas pour le développement de systèmes sécurisés.

3.3.3.2. Simulation de fautes

La simulation de fautes au niveau composant est la méthode la plus communément utilisée lors de la génération de stimuli de test pour des systèmes AMS&RF. Cette méthode a été mise en place dans les années 1970 afin de générer des stimuli pour le test de systèmes numériques [3.13]. Elle s'est fortement développée grâce à l'apparition d'outils de simulation performants c'est-à-dire adaptés à l'analyse de systèmes hétérogènes complexes. Dans [3.14], les auteurs présentent une méthode basée sur la simulation de fautes permettant d'identifier les mesures à réaliser pour le test d'un amplificateur faible bruit.

La simulation de fautes permet d'évaluer la qualité d'un stimulus pour détecter des fautes. Le principe de la simulation de fautes consiste à introduire une perturbation dans un système et à observer le comportement du système contenant la faute. Le système contenant la perturbation ou la faute est appelé *description fautive*. Son comportement est comparé à celui de la description d'origine ou *description saine*. L'objectif est d'identifier les stimuli qui permettent d'obtenir un comportement de la description fautive différent du comportement de la description saine. La détection d'une description fautive permet de montrer que la faute a été excitée et que son effet a été propagé vers une sortie.

La partie suivante présente les différentes méthodes utilisées pour injecter ces fautes.

3.3.3.2.1. Injection de fautes

L'injection de fautes peut se faire suivant trois principales méthodes : saboteur, mutation et commandes du simulateur [3.15].

La première méthode permettant d'injecter des fautes dans un système est basée sur l'utilisation de *saboteurs*. Un saboteur est un bloc spécifique qui est ajouté à la description d'origine. La fonction unique de ce bloc est d'injecter une ou plusieurs perturbations dans le système. L'intérêt de cette méthode repose sur la facilité d'insérer des saboteurs entre les différents blocs du système. Suivant le type de défaut modélisé, la perturbation peut être soit permanente soit transitoire. Dans le second cas, cela signifie que le saboteur est actif uniquement pendant un certain temps. Cette approche est intéressante mais permet uniquement d'injecter des fautes sur les connexions inter-blocs du système et ne permet pas d'injecter des fautes à l'intérieur de ces blocs. De plus, il est pratiquement impossible d'injecter des fautes de plus haut niveau. Différentes catégories de saboteurs numériques sont présentées dans [3.15] et [3.16], elles permettent de modéliser des défauts apparaissant sur les lignes d'interconnexions. Dans le domaine analogique, [3.17] propose d'utiliser un saboteur pour injecter des impulsions de courant à l'intérieur d'une PLL⁴.

La seconde méthode d'injection de faute est basée sur le principe de la *mutation*. Le mutant ou la description fautive est générée en modifiant le comportement de la description d'origine. La mutation peut se faire de deux manières différentes, la première consiste à modifier la structure du système en injectant des fautes dans la description d'origine [3.18]. Par exemple, il est possible d'ajouter des courts-circuits, des circuits-ouverts... La seconde approche consiste à modifier les valeurs des paramètres utilisés dans la description du système. Par exemple, la valeur d'une résistance peut être modifiée afin de modéliser une faute paramétrique.

⁴ Phase Locked Loop

La dernière méthode permettant l'injection de fautes est présentée dans ce paragraphe, son principe consiste à utiliser les « *commandes du simulateur* [3.15] » pour modifier les valeurs des signaux et des variables de la description. Par cette approche, l'injection de fautes peut se faire sur les nœuds externes du système (signaux en VHDL ou terminaux en VHDL-AMS). Il est alors possible de modifier le signal qui est envoyé sur la description du système. L'injection de fautes sur les signaux/terminaux reste toutefois limitée, le réel avantage de cette technique apparaît lors de la modification de paramètres génériques du circuit. En effet, la modification de ces paramètres permet de modifier les caractéristiques internes du circuit. Ainsi tout paramètre, défini comme paramètre générique dans la description VHDL-AMS, SPICE..., peut être modifié et donc modéliser une faute. Cette méthode permet d'injecter facilement des fautes sans modifier la description interne du circuit ou système. Cette approche est particulièrement intéressante lors de l'utilisation de langages de description matérielle compilée comme Verilog ou VHDL-AMS. En effet, il n'est pas nécessaire de recompiler la description pour chaque faute injectée. Cette économie de temps peut devenir relativement importante lors de l'analyse de systèmes complexes nécessitant l'injection de nombreuses fautes et la simulation d'un nombre important de stimuli.

3.3.3.2.2. *Modèle de fautes*

Le réalisme du modèle de fautes a un impact direct sur la qualité de l'évaluation des stimuli de test par simulation de fautes. Si dans le domaine numérique le modèle de fautes « collage à 0 » et « collage à 1 » est largement utilisé, il n'existe pas de modèle de fautes aussi reconnu dans le domaine analogique et RF.

Dans la partie §3.2.2.3, les deux principaux types de fautes, catastrophiques et paramétriques, ont été citées. Dans [3.18] et [3.19], des modèles de fautes catastrophiques sont utilisés pour la génération de stimuli pour le test matériel. L'injection de fautes catastrophiques se fait sur des descriptions définies au niveau composant et les types de fautes sont des courts-circuits et des circuits-ouverts placés entre les différentes interconnexions des résistances, transistors, capacités... Dans [3.20], l'auteur propose de qualifier les stimuli en utilisant des modèles de fautes paramétriques, c'est-à-dire en injectant des variations sur les paramètres du processus de fabrication (largeur, longueur, épaisseur d'oxyde...). L'injection de fautes sur des paramètres plus haut niveau a été développée dans [3.21], elle permet de diminuer les temps d'évaluation des stimuli en utilisant des descriptions moins coûteuses en temps de simulation.

Les modèles de fautes présentés dans le paragraphe précédent s'inscrivent dans un flot de conception Top-Down, c'est-à-dire que les fautes sont définies en fonction de la structure du système. Il existe une seconde méthode de modélisation de faute qui se positionne dans un flot Bottom-Up. Les fautes sont extraites à partir de la description bas-niveau. L'IFA⁵ permet d'identifier des fautes à partir du placement/routage du circuit et de conserver celles qui ont la plus forte probabilité d'apparaître [3.22]. Les trois principales étapes de l'analyse IFA sont : la définition de fautes déterminées à partir des données statistiques du processus de fabrication, l'extraction/conversion de fautes sur les grandeurs physiques en fautes électriques et la classification des fautes en fonction de leur probabilité d'apparition. Ces fautes induites sont alors réutilisées pour l'évaluation des stimuli de test. Dans [3.23], l'auteur utilise la méthode IFA pour définir un ensemble de fautes réalistes, c'est-à-dire ayant une probabilité d'apparition importante,

⁵ Inductive Faults Analysis

afin de réduire le nombre de stimuli nécessaire pour la vérification des performances du système. Le problème de cette approche est que la simulation doit se faire au niveau composant et nécessite des temps de simulation importants pour l'analyse de systèmes complexes. Pour résoudre ce problème, [3.24] propose de définir un modèle plus haut niveau en transposant les fautes obtenues par la méthode IFA au niveau comportemental. L'analyse des stimuli de test peut ainsi se faire au niveau comportemental et ainsi nécessiter des temps de simulation moins importants.

Parmi les deux méthodes descendante et montante permettant de définir des modèles de fautes, la seconde basée sur l'extraction permet d'obtenir des modèles de fautes précis et réalistes mais nécessite l'utilisation d'une description physique du système. Elle ne peut donc pas être mise en place avant la réalisation de ce niveau physique. Par opposition, la première méthode permet d'obtenir rapidement des modèles de fautes, mais en contre partie elle présente une précision réduite. Toutefois, dans ce dernier cas, il est parfois possible de réutiliser les modèles de fautes extraits lors de la conception de précédents systèmes.

3.3.3.2.3. Définition des réponses saines

Lorsqu'on utilise la méthode d'injection et de simulation de fautes, le comportement des descriptions fautives doit être analysé afin de déterminer si le stimulus a permis de détecter la faute [3.20]. Pour cela, il faut comparer la *réponse fautive* obtenue par simulation de la description fautive avec la *réponse saine*. Lors du test fonctionnel, la réponse saine est déterminée directement à partir du cahier des charges. Mais dans le cas de l'utilisation de méthodes de test structurel, rien ne permet de connaître la réponse du système sain car le stimulus évalué n'est pas forcément défini dans le cahier des charges. La seule solution consiste alors à simuler la description saine afin d'obtenir une réponse saine propre à chaque stimulus évalué. Lorsque la réponse fautive et la réponse saine sont suffisamment différentes, il est possible de conclure que le stimulus utilisé permet de détecter la faute.

Toutefois, l'utilisation d'une seule réponse saine n'est pas suffisante. En effet, il existe une infinité de circuits sains, une plage de réponse saine doit donc être utilisée pour la comparaison avec le comportement fautif. Pour cela, il existe plusieurs possibilités, la première consiste à définir des marges de tolérance à partir de la réponse d'une seule description saine. La définition de ces marges de tolérance est toutefois difficile car elle diffère suivant le stimulus, la mesure, la robustesse de la conception...

La seconde approche consiste à tenir compte des marges de tolérances des paramètres du système définies lors de sa conception. En effet, lors de la conception, des valeurs limites sont déterminées pour chaque paramètre. Par exemple, au niveau fonctionnel, les valeurs minimum et maximum du gain seront définies. Au niveau composant, une résistance sera définie à +/-20%... Ces marges de tolérances impliquent qu'il existe une infinité de combinaisons de paramètres permettant de décrire un système sain. Ainsi, comme il n'est pas possible de simuler l'ensemble des descriptions saines, [3.25] propose une approche considérant les pires-cas, pour cela il faut combiner les valeurs minimales et maximales de chaque paramètre. Cette méthode génère 2^n descriptions saines où n est le nombre de paramètres modélisés ; évidemment cette méthode ne peut pas être utilisée dans le cas des systèmes complexes car elle nécessiterait des temps de simulations trop importants. De plus, cette approche suppose que le système est monotone c'est-à-dire suppose que les relations entre les paramètres et les mesures sont strictement croissantes ou décroissantes et donc que les mesures

minimales ou maximales sont obtenues quand les paramètres sont à leurs valeurs limites. Une approche moins simplificatrice consiste à diviser chaque réponse en intervalles monotones [3.25]. Cette approche est plus précise mais augmente encore le nombre de combinaisons possibles.

Des approches statistiques ont également été développées afin de diminuer le nombre de simulation des descriptions saines. Une analyse de Monte Carlo permet de générer aléatoirement des descriptions. Cette génération se fait en considérant les distributions statistiques des paramètres physiques et permet ainsi d'obtenir une population représentative de la réalité. Il est ensuite possible de définir des limites sur les mesures de cette population [3.20]. Des méthodes d'estimation gaussienne ont été mises en place dans [3.26] afin de diminuer le nombre de simulations. Le principe consiste à faire l'hypothèse que la mesure obtenue, pour un stimulus, suit une distribution gaussienne. Il est alors possible de diminuer le nombre de descriptions à simuler en estimant la mesure à partir d'un échantillon. Dans [3.27], l'auteur propose de définir les limites de détection en fonction du nombre de circuits sains que l'on s'autorise à rejeter. Comme la métrique est exprimée en nombre de circuits sains rejetés sur un million, l'auteur utilise des estimations gaussiennes pour augmenter artificiellement la taille de la population et permet ainsi de simuler seulement mille descriptions saines.

3.3.3.3. Qualification des stimuli de test

La qualification des stimuli de test est une étape importante de la génération des stimuli de test. Elle permet de déterminer si chaque stimulus est pertinent pour la détection de fautes ou s'il faut poursuivre la génération afin de garantir une meilleure qualité du test.

La *couverture de fautes* ou le *score de mutation* correspondent au rapport entre le nombre de fautes ou mutants détectés et le nombre total de fautes ou mutants injectés. Ces deux métriques sont très employées pour qualifier les stimuli générés dans le domaine numérique [3.28] et analogique [3.29].

Toutefois, lors de l'injection de fautes paramétriques, la couverture de faute seule n'est pas suffisante car l'amplitude de la variation sur chaque paramètre a un effet sur la détection des fautes. Ainsi une très forte variation est plus facile à détecter qu'une faible variation ; or cette notion n'est pas prise en compte dans cette métrique. Pour limiter ce problème, [3.24] propose d'injecter des variations paramétriques à différentes valeurs : 5%, 10% jusqu'à 25% de la valeur typique. Le problème de cette approche est qu'elle augmente le nombre de descriptions fautives à simuler.

De plus, lors de la qualification des stimuli, il faut aussi tenir compte de variations simultanées de plusieurs paramètres. En effet, suivant que l'on injecte des variations sur les autres paramètres ou pas, il sera plus ou moins facile de détecter une faute. Pour résoudre ce problème, [3.26] propose de calculer la *Probabilité de Détection de Fautes (PDF)* en calculant la surface nommée PDF dans la Fig. 3.4. Cette surface correspond à la probabilité de réellement détecter la faute lorsque les autres paramètres varient. En fait, comme les paramètres de la description saine et de la description fautive peuvent varier, l'auteur suppose que les réponses saine et fautive suivent une distribution normale. Il réalise alors une estimation gaussienne de la réponse saine de référence D_{ref} et de la réponse fautive D_f . La Fig. 3.4 illustre ce principe. L'axe des abscisses représente une mesure obtenue pour un stimulus. Plus les réponses sont distinctes, plus le PDF est proche de 100%. Cette mesure permet de qualifier la détection de la faute. Dans [3.30], cette métrique permet de diminuer les risques d'erreur de tri (circuits fautifs conservés, circuits sains jetés) en cherchant à identifier le stimulus et la limite de détection optimale.

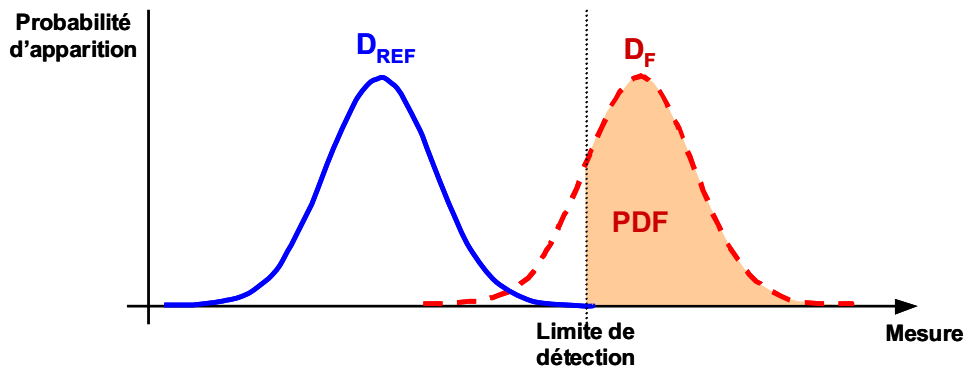


Fig. 3.4. Probabilité de détection de fautes PDF.

Pour le test de production, il existe en particulier deux métriques. La première métrique permet de quantifier le pourcentage de circuits sains qui ont été rejetés lors du test ; on parle de Perte de Rendement (ou Yield Loss) [3.32]. La seconde métrique représente parmi les circuits qui passeront le test, ceux qui peuvent être défectueux ; on parle alors de Taux de défauts (ou de Defect Level) [3.32]. Ces deux métriques permettent donc de quantifier l'erreur qui est faite lors du test de production.

3.3.4. Test alternatif

Cette méthode de test propose une solution principalement appliquée au test de production. L'approche utilisée est différente des méthodes présentées précédemment. D'une part, le test fonctionnel de systèmes AMS&RF est de plus en plus difficile à mettre en place car il faut tester de plus en plus de fonctionnalités et le temps nécessaire pour tester toutes ces fonctionnalités devient trop long. D'autre part, le test des spécifications est parfois lui aussi mal adapté au test de production car il demande l'utilisation d'équipements de test coûteux. De plus, la technique de simulation de fautes peut ne pas être applicable parce que les temps de simulation peuvent être très longs. Le test alternatif propose ainsi une nouvelle approche essayant de répondre à ces difficultés.

Le principe du test alternatif consiste à prédire les spécifications du système en effectuant des mesures alternatives moins coûteuses en temps de simulation et en équipement de test ([3.33], [3.34]). Pour cela, des variations sont injectées sur les paramètres du système ; celles-ci entraînent des variations à la fois sur les spécifications et des variations sur les mesures alternatives (Fig. 3.5). Les résultats de ces simulations permettent de définir des relations entre les paramètres physiques P et les spécifications S ($S=f(P)$) et entre les paramètres physiques P et les mesures alternatives M ($M=g(P)$). Ainsi à partir de ces relations et des définitions des plages de tolérances sur les paramètres physiques P , il est possible de déterminer les plages de tolérances à la fois sur les spécifications S et sur les mesures M . Ensuite, une méthode mathématique MARS [3.35] basée sur l'utilisation de méthode de régression statistique permet de déterminer la relation entre les mesures M et les spécifications S ($M=h(S)$). Cette étape permet donc de prédire les spécifications à partir de mesures alternatives. L'optimisation des stimuli se fait en comparant la réponse du système simulé (spécifications simulées) avec celle du système prédit à partir de la mesure (spécifications estimées). Ainsi, le stimulus généré est optimal lorsque l'erreur sur l'estimation des spécifications à partir de la mesure est minimale.

Cette approche alternative est intéressante pour les points précisés ci-dessus, mais nécessite la

simulation du système au niveau composant en injectant de nombreuses variations sur les paramètres physiques. Il est donc indispensable d'exécuter un nombre de simulations suffisamment important afin d'obtenir un modèle estimé suffisamment précis. Par conséquent, l'analyse de systèmes complexes peut devenir très longue, voire impossible.

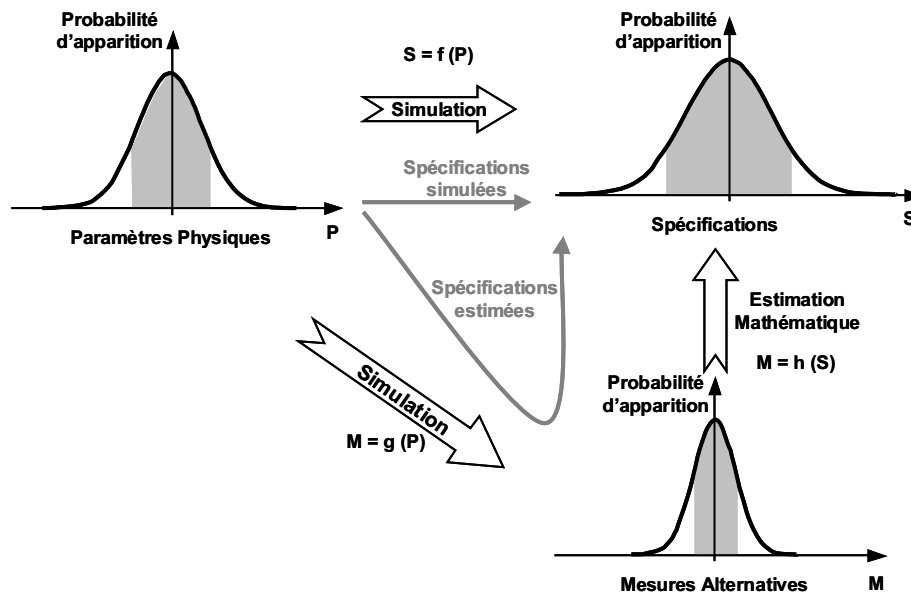


Fig. 3.5. Technique de test alternatif [3.33].

3.3.5. Application des stimuli de test

Après avoir développé dans la partie précédente les différentes méthodes permettant de générer des stimuli de test, il reste maintenant à appliquer les stimuli pour permettre le test matériel du circuit.

La première approche consiste à appliquer les stimuli directement sur le circuit afin de permettre soit la vérification du prototype soit le test de production. Le test du circuit se fait alors de façon matérielle.

Une autre approche permettant de tester un système électronique consiste à intégrer des modules de test à l'intérieur du circuit. Cette approche est uniquement dédiée au test matériel : test de production ou test en ligne et sur site. L'objectif est alors de tester le circuit en utilisant le minimum de ressources externes. Ainsi, l'utilisation de BIST⁶ consiste à implanter matériellement des générateurs de test (ou stimuli) et/ou des analyseurs de test à l'intérieur du circuit. Cette approche permet aux circuits de s'autotester en s'appliquant et en analysant automatiquement des stimuli de test et leurs résultats. Mais bien plus que cela, elle permet aussi d'augmenter la contrôlabilité et l'observabilité des systèmes.

3.4. Conception en vue du test

Nous avons vu dans le chapitre précédent que les systèmes développés sont de plus en plus complexes et intègrent de plus en plus de fonctionnalités. D'une part ces systèmes demandent des méthodes de conception rigoureuses et adaptées, et d'autre part la testabilité matérielle de ces systèmes doit aussi être prise en compte. En effet, suite à la conception d'un système, les concepteurs peuvent se rendre compte que celui-ci n'est pas testable. Dans ce cas, il ne sera pas

⁶ Built In Self Test.

possible de garantir la qualité de la fabrication de ce système. Il est donc indispensable de considérer la testabilité matérielle dès les étapes de conception. On parle alors de CVT⁷ (ou en anglais de DfT⁸). Ce principe consiste à concevoir le système en cherchant à assister, simplifier et réduire le temps du test matériel.

La DfT incite les ingénieurs de conception à définir l'architecture du système en considérant l'impact de leurs choix sur le test matériel. Cette méthode est plus répandue pour le test de SoCs digitaux. La première approche consiste à ajouter des points d'observation ou de contrôle (au sens commande) dans le système. L'intérêt est alors de pouvoir diviser le système en blocs et ainsi d'observer et de contrôler directement chaque bloc afin de tester leur fonctionnement individuellement. Cette approche pourrait sembler facile à mettre en œuvre mais l'ajout de matériel (même une simple ligne) sur une ligne RF peut être un véritable casse-tête pour le concepteur [3.36].

Une seconde approche permettant d'augmenter la testabilité d'un système consiste à intégrer des ressources à l'intérieur même du circuit. Le bloc ajouté est appelé BIST, son intérêt est alors de transformer un signal en une signature. Le BIST peut aussi fournir un signal « go/no go » qui permet de déterminer directement si le système est fonctionnel ou défaillant. Ainsi, le test de la partie analogique d'un SoC AMS&RF peut se faire de manière transparente comme pour du test purement numérique c'est-à-dire en positionnant le système en mode test par l'intermédiaire d'un bit de contrôle et en récupérant la réponse du test par un autre bit. L'utilisation de BIST est spécifique à chaque système. Le développement de techniques de BIST est en évolution mais reste pour l'instant peu développé surtout pour les circuits RF.

3.5. Conclusion

Nous avons présenté dans ce chapitre le test de SoCs AMS&RF. Lors du développement d'un système complexe, il est nécessaire de mettre en place des procédures permettant de s'assurer que le système respecte les spécifications attendues par le client. Ainsi la vérification et la validation sont les premiers types de tests utilisés, ils permettent de vérifier que le système a été conçu correctement et qu'il respecte les spécifications définies dans le cahier des charges. Une fois que la conception du système a été suffisamment vérifiée, la fabrication du circuit à grande échelle est lancée. Lors de cette étape de fabrication, des défauts peuvent apparaître à cause de la présence d'impuretés ou de dérives des paramètres du processus de fabrication. Ces défauts doivent être détectés par le test de production afin de valider l'étape de fabrication. Des méthodes permettant de définir des stimuli de test efficaces ont été présentées dans ce chapitre. Ces méthodes consistent soit à tester les fonctionnalités du système, soit à détecter la présence de fautes, soit à trouver des stimuli simplifiés permettant de tester indirectement les fonctionnalités. Dans l'approche que nous proposerons dans les chapitres suivants, nous privilégierons la génération et la qualification de stimuli de test à partir de la simulation de descriptions de plus haut niveau que celles habituellement utilisées ; cette approche permettra de diminuer les temps de calcul et ainsi de traiter des systèmes AMS&RF complexes.

⁷ Conception en vue du test.

⁸ Design for Test.

Bibliographie

- [3.1] P. Rashinkar, P. Paterson, L. Singh, "System-on-a-chip Verification Methodology and Techniques", Edition Kluwer Academic Publishers, 2001.
- [3.2] K.S Kundert, O. Zinke, "The Designer's Guide to Verilog-AMS", Edition Kluwer Academic Publishers, Juin 2004.
- [3.3] J.-P. Fournier, "Vérification versus validation", Laboratoire didactique informatique <http://www.infeig.unige.ch/support/se/lect/prg/tst/node1.html>, Septembre 2002.
- [3.4] W.H. Kao, L. Chi-Yuan, R. Singh, "Parasitic extraction: current state of the art and future trends", Proceedings of the IEEE Volume 89, Issue 5, pp729 - 739, Mai 2001.
- [3.5] S. Vitali, D. Laurentiis, N. Albertazzi, G. Agnelli, F. Rovatti, "Multi-standard simulation of WLAN/UMTS/GSM transceivers for analog front-end validation and design", International Symposium on Wireless Communication Systems (ISWCS'04), pp 16- 20, Septembre 2004.
- [3.6] M. Sida, R. Ahola, D. Wallner, "Bluetooth transceiver design and simulation with VHDL-AMS", IEEE Circuits and Devices Magazine, pp 11 - 14, Mars 2003.
- [3.7] E. Liu, H. Chang, A. Sangiovanni-Vincentelli, "Analog system verification in the presence of parasitics using behavioral simulation", Design Automation Conference (DAC'93), pp 159 - 163, Juin 1993.
- [3.8] K. Kundert, H. Chang, "Verification of Complex Analog and RF IC Designs", Proceedings of the IEEE, pp 622 - 639, Mars 2007.
- [3.9] C. Landrault, "Test de circuits et de systèmes intégrés (Traité EGEM, série Electronique et micro-électronique)", Edition Lavoisier, Mai 2004.
- [3.10] A. Bounceur, "Plateforme CAO pour le test de circuits mixtes", Thèse de l'INP Grenoble, Avril 2007.
- [3.11] M. Burns, W. R. Gordon, "Introduction to Mixed-Signal IC Test & Measurement", Edition Oxford University Press, Janvier 2001.
- [3.12] G. Devarayanadurg, M. Soma, P. Goteti, S.D. Huynh, "Test set selection for structural faults in analog IC's", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, pp 1026 - 1039, Juillet 1999.
- [3.13] R. M. McClure, "Fault simulation of digital logic utilizing a small host machine", Proceedings of the 9th workshop on Design automation (DAC'72), pp 104 - 110, Juin 1972.
- [3.14] R. Kheriji, V. Danelon, J.L. Carbonero, S. Mir, "Optimising test sets for a low noise amplifier with a defect-oriented approach", Design Automation and Test in Europe (DATE'05), pp 170 - 171, Mars 2005.
- [3.15] J. Gracia, J.C. Baraza, D. Gil, P.J. Gil, "Comparison and Application of Different VHDL-Based Fault Injection Techniques", IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT'01), pp 233 - 241, 2001.
- [3.16] E. Jenn, J. Arlat, M. Rimén, J. Ohlsson, J. Karlsson, "Fault injection into VHDL models: the MEFISTO tool", International Symposium on Fault-Tolerant Computing, pp 66 - 75, Juin 1994.
- [3.17] R. Leveugle, A. Ammari, "Early SEU fault injection in digital, analog and mixed signal circuits: a global flow", Design, Automation and Test in Europe Conference and Exhibition, pp 590 - 595, Février 2004.
- [3.18] E. Acar, S. Ozev, "Defect-based RF testing using a new catastrophic fault model", IEEE International Test Conference (ITC'05), pp 421 - 429, Novembre 2005.

- [3.19] L. Milor, V. Visvanathan, "Detection of catastrophic faults in analog integrated circuits", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, pp 114 - 130, Février 1989.
- [3.20] K. Saab, N. Ben-Hamida, B. Kaminska, "Parametric Fault Simulation and test vector generation", Design, Automation and Test in Europe Conference and Exhibition (DATE'00), pp 650 - 656, September 2000.
- [3.21] P. Wilson, Y. Kilic, J. Ross, M. Zwolinski, A. Brown, "Behavioural Modeling of Operational amplifier Faults using analogue Hardware Description Languages (HDLs)", Fifth IEEE International Workshop on Behavioral Modeling and Simulation (BMAS'2001), pp 106 - 112, Octobre 2001.
- [3.22] J.P. Shen, W. Maly, F.J. Ferguson, "Inductive Fault Analysis of MOS Integrated Circuits", IEEE Design & Test of Computers, pp 13 - 16, Décembre 1985.
- [3.23] Z. Jaworski, M. Niewczas, W. Kuzmicz, "Extension of Inductive Fault Analysis to Parametric Faults in Analog Circuits with Application to Test Generation," IEEE VLSI Test Symposium (VTS'97), pp 172 - 175, Mai 1997.
- [3.24] Liu Fang, S. Ozev, "Fast hierarchical process variability analysis and parametric test development for analog/RF circuits", IEEE International Conference on Computer Design (ICCD'05), pp 161 - 168, Octobre 2005.
- [3.25] M. W. Tian, C.-J. R. Shi, "Worst Case Tolerance Analysis of Linear Analog Circuits Using Sensitivity Bands", IEEE Transactions on Circuits and Systems, Vol. 47, No. 8, Août 2000.
- [3.26] A. Khouas, "Simulation de Fautes et Optimisation des Tests de Production pour les Circuits Analogiques avec Prise ne Compte des Tolérances", Thèse de doctorat, Université de Paris VI, 2000.
- [3.27] A. Bounceur, S. Mir, L. Rolindez, E. Simeu, "A CAT Platform for Analog and Mixed-Signal Test Evaluation and Optimization", IEEE European Test Conference (ETC'06), pp 217-222, Mai 2006.
- [3.28] Y. Serrestou, V. Beroulle, C. Robach, "Functional Verification of RTL Designs driven by Mutation Testing metrics", 10th Euromicro Conference On Digital System Design, pp 222-227, Août 2007.
- [3.29] A. Milne, D. Taylor, K. Naylor, "Assessing and comparing fault coverage when testing analogue circuits", IEE Proceedings Circuits, Devices and Systems, pp 1 - 4, Février 1997.
- [3.30] K. Saab, N. Ben-Hamida, B. Kaminska, "Parametric fault simulation and test vector generation", Design, Automation and Test in Europe Conference (DATE'00), pp 650 - 656, Septembre 2000.
- [3.31] Y.-B. Sha; M.-S. Lee; C.-N. J. Liu, "On code coverage measurement for Verilog-A", IEEE International High-Level Design Validation and Test Workshop, pp 115 - 120, 2004.
- [3.32] S. Sunter, N. Nagi, "Test metrics for analog parametric faults", VLSI Test Symposium (VTS'99), pp 226 - 234, Avril 1999.
- [3.33] P.N. Variyam, S. Cherubal and A. Chatterjee, "Prediction of Analog Performance Parameters Using Fast Transient Testing", IEEE Trans. CAD of Integrated Circuits and Systems, pp 349 - 361, 1992.
- [3.34] A. Halder, S. Bhattacharya, A. Chatterjee, "Automatic Multitone Alternate Test Generation For RF Circuits Using Behavioral Models", International Test Conference 2003 (ITC'03), pp 665 - 673, 2003.
- [3.35] J. H. Friedman, "Multivariate adaptive regression splines," Ann. Statistics, vol. 19, no. 1, pp 1 - 141.
- [3.36] J. Ferrario, R. Wolf, S. Moss, M. Slamani, "A Low-Cost Test Solution for Wireless Phone RFICs", IEEE Communications Magazine, VOL 41PART 9, pp 82 - 89, 2003.

CHAPITRE IV :

MODELISATION COMPORTEMENTALE

D'UN SYSTEME W-CDMA

| | |
|---|----|
| CHAPITRE IV : | 57 |
| MODELISATION COMPORTEMENTALE D'UN SYSTEME W-CDMA | 57 |
| 4.1. Introduction | 60 |
| 4.2. Description du système W-CDMA | 60 |
| 4.2.1. Présentation générale de la technologie W-CDMA | 60 |
| 4.2.2. Spécification des paramètres électriques d'après le standard UMTS | 63 |
| 4.2.3. Architecture de la partie réception du système W-CDMA | 64 |
| 4.3. Description du modèle comportemental | 65 |
| 4.4. Modélisation comportementale des paramètres | 66 |
| 4.4.1. Modélisation et validation du gain | 66 |
| 4.4.1.1. Modélisation | 66 |
| 4.4.1.2. Validation de la description | 67 |
| 4.4.2. Modélisation et validation du point de compression | 68 |
| 4.4.2.1. Modélisation | 68 |
| 4.4.2.2. Vérification de la description | 69 |
| 4.4.3. Modélisation et validation de l'intermodulation d'ordre trois | 70 |
| 4.4.3.1. Modélisation | 70 |
| 4.4.3.2. Vérification de la description | 72 |
| 4.4.4. Modèle comportemental de l'amplificateur | 74 |
| 4.5. Validation du modèle comportemental du système W-CDMA | 75 |
| 4.5.1. Validation fonctionnelle | 75 |
| 4.5.2. Réjection des raies parasites | 77 |
| 4.6. Validation du modèle comportemental à partir de mesures sur le circuit | 78 |
| 4.7. Conclusion | 79 |
| Bibliographie | 80 |

| | | |
|-------------------|---|-----------|
| Fig. 4.1. | Répartition des bandes de fréquence. | 61 |
| Fig. 4.2. | Division FDD et TDD. | 62 |
| Fig. 4.3. | Etalement du signal. | 62 |
| Fig. 4.4. | Modulation QPSK (4 états). | 63 |
| Fig. 4.5. | Schéma de la partie réception du système W-CDMA. | 65 |
| Fig. 4.6. | Mesure du gain d'un amplificateur. | 67 |
| Fig. 4.7. | Validation de la modélisation du gain d'un amplificateur. | 68 |
| Fig. 4.8. | Modélisation du point de compression. | 69 |
| Fig. 4.9. | Mesure du point de compression de l'amplificateur. | 69 |
| Fig. 4.10. | Validation de la modélisation du point de compression d'un amplificateur. | 70 |
| Fig. 4.11. | Réponse fréquentielle et réponse en puissance de l'intermodulation d'ordre trois. | 71 |
| Fig. 4.12. | Mesure du point d'intersection d'ordre trois de l'amplificateur. | 72 |
| Fig. 4.13. | Validation de la modélisation du point intersection d'intermodulation d'ordre trois IIP3. | 73 |
| Fig. 4.14. | Modèle comportemental décrit en VHDL-AMS d'un amplificateur. | 75 |
| Fig. 4.15. | Démodulation IQ par le système W-CDMA. | 76 |
| Fig. 4.16. | Démodulation du signal modulé IQ. | 77 |
| Fig. 4.17. | Validation de la réjection de raies parasites. | 78 |
| Fig. 4.18. | Simulation de la réjection des raies parasites. | 78 |
| Fig. 4.19. | Schéma du routage du système W-CDMA. | 79 |
| Fig. 4.20. | Simulation et mesure de l'intermodulation d'ordre 3 en fonction du gain VGA1. | 79 |

4.1. Introduction

L'objectif de ce chapitre est de présenter le modèle comportemental du système W-CDMA¹ utilisé comme cas d'étude tout au long de cette thèse. Ce modèle comportemental a été développé dans un flot de conception Top-Down comme décrit dans le chapitre II (§2.3.4). Ainsi le modèle a été généré à partir de sa description fonctionnelle et raffiné grâce à l'insertion des comportements non-idéaux dans les blocs ou macro-composants qui le constituent. La description du système à ce niveau d'abstraction permet de réaliser la budgétisation des paramètres fonctionnels et électriques de chaque bloc, c'est-à-dire de convertir les paramètres systèmes en paramètres comportementaux liés à chacun des blocs. Une première validation de cette budgétisation est abordée à la fin de ce chapitre.

Le modèle comportemental développé ici est réutilisé dans la suite afin de valider notre approche pour la qualification et la génération des stimuli de test. Nous verrons alors que le choix du niveau de détail de ce modèle aura une influence sur la qualité des stimuli de test générés.

Le SoC² étudié est la partie réception d'un système W-CDMA, il comprend des parties analogiques, mixtes et RF. Dans un premier temps, afin de mettre en évidence quels sont les effets importants à modéliser, la technologie W-CDMA est présentée. Ensuite, l'architecture proposée pour la partie réception du système est décrite. Cette architecture représente le système sous la forme de plusieurs blocs. Il est alors possible de raffiner le modèle fonctionnel de chacun de ces blocs et ainsi de décrire le système au niveau comportemental. Enfin, des méthodes permettant de modéliser en VHDL-AMS certaines caractéristiques électriques et/ou RF sont présentées. Ces méthodes permettent de générer les descriptions VHDL-AMS des différents blocs du système. Pour clore ce chapitre, nous comparons les fonctionnalités et les caractéristiques globales du modèle comportemental du système avec les spécifications définies par la norme. Nous comparons aussi le comportement de notre modèle avec des mesures réalisées sur le circuit. Ces deux étapes nous permettent de valider la modélisation comportementale du système W-CDMA.

4.2. Description du système W-CDMA

Avant d'aborder la modélisation comportementale du système utilisé, nous allons faire une rapide présentation de la modulation W-CDMA.

4.2.1. Présentation générale de la technologie W-CDMA

La technologie W-CDMA est une extension du codage CDMA³. Cette dernière est une méthode d'accès multiple par codage qui permet le partage des ressources temporelles et fréquentielles c'est-à-dire d'utiliser au même moment le même canal de fréquence. La modulation CDMA a été développée par l'armée Américaine pour le radio guidage afin de protéger les transmissions contre le brouillage et l'interception ennemie. Le codage est basé sur l'étalement de spectre, son principe consiste à remplacer chaque bit par une séquence de bits. Tandis que les méthodes de transmission

¹ Wide Bande Code Division Multiple Acces.

² System on Chip.

³ Code Division Multiple Access.

RF existantes (par exemple, le GSM⁴) sont basées sur l'accès multiple en divisant les canaux dans les domaines temporel et fréquentiel ; la technologie CDMA a pour avantage de permettre le partage d'un même canal (temporel et fréquentiel) en codant le signal. Ainsi, le codage permet de reconstituer le signal d'origine sans erreur et d'inhiber les perturbations des autres transmissions utilisant le même canal. Evolution de la technologie CDMA, la modulation W-CDMA reprend les mêmes fonctionnalités. Les principales modifications sont l'élargissement du spectre de fréquence et l'utilisation de paquets de données, ces évolutions permettent le transfert de données de grandes tailles. Le W-CDMA a été développé pour une utilisation en téléphonie mobile, il définit la couche physique du standard de communication UMTS⁵.

L'UMTS est une des technologies de la téléphonie mobile de troisième génération (3G). Elle a pour objectif d'être compatible avec tous les réseaux de seconde génération du monde et de supporter des fonctions multimédia (Transfert haut débit) [4.1]. Ainsi, l'UMTS définit un débit minimal de 144 kbit/s et permet d'atteindre des débits pouvant aller jusqu'à 2Mbits/s. Les bandes de fréquences allouées par l'UMTS sont 1920-1980 MHz pour la partie émission et 2110-2170 MHz pour la partie réception (Fig. 4.1). Les canaux de communication sont limités à 5MHz, c'est-à-dire que la plage d'émission du téléphone portable (1920MHz à 1980MHz) est divisée en 12 canaux. Il en est de même pour la plage de la partie réception.

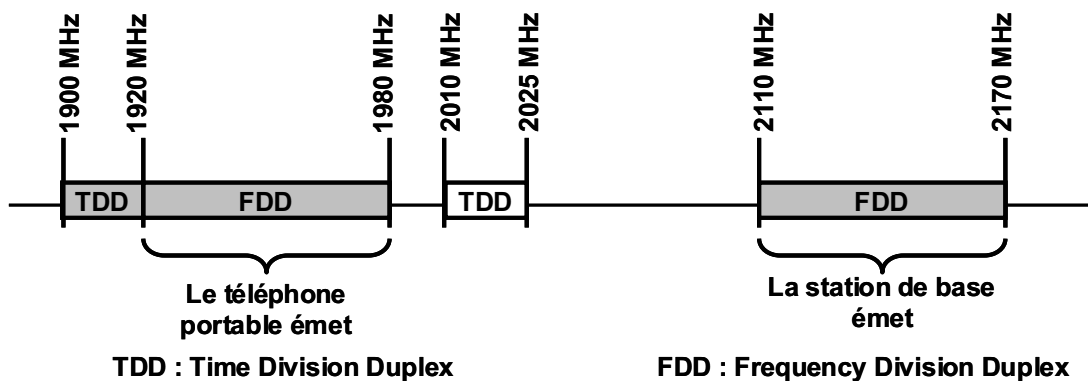


Fig. 4.1. Répartition des bandes de fréquence.

Les termes de FDD et de TDD définissent le format de l'accès multiple (Fig. 4.1), ils permettent de répartir les canaux soit dans le domaine temporel (TDD : Time Division Duplex) soit dans le domaine fréquentiel (FDD : Frequency Division Duplex) (Fig. 4.2). Ainsi les bandes FDD composées de canaux de 5MHz sont appropriées pour la téléphonie « bas débit » et les bandes TDD de 20MHz sont dédiées au transfert de données plus haut débit. Les flèches ascendantes et descendantes (Fig. 4.2) indiquent le sens de la communication (flèche descendante : communication de la station de base vers le téléphone).

⁴ Global System Mobiles.

⁵ Universal Mobil Telecommunications System.

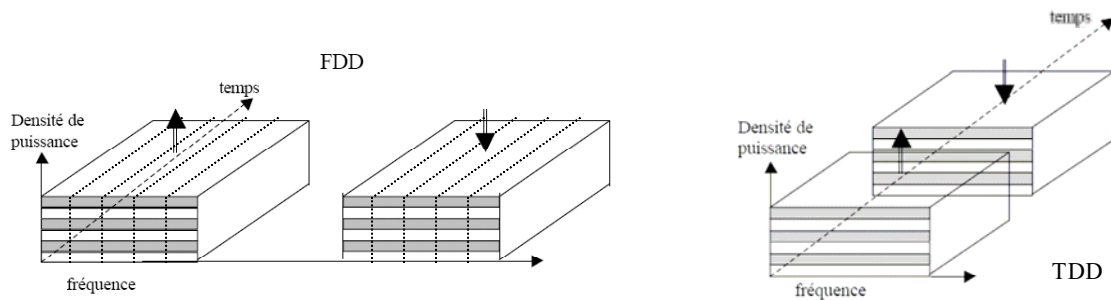


Fig. 4.2. Division FDD et TDD.

Le codage utilisé par la technologie W-CDMA est basé sur l'étalement du spectre, celui-ci se fait en deux étapes. La première, dite de canalisation ou d'étalement (spreading), transforme chaque symbole de données en une séquence. Le chip, ou code d'étalement, est l'élément de base de la séquence qui est multipliée au symbole (Fig. 4.3). La seconde étape, dite de brouillage (scrambling), s'applique aux chips. Cette étape est aussi nécessaire pour séparer les différentes données d'applications (transfert de données, téléphonie...) issues d'une même source, et pour séparer différentes stations de base et mobiles.

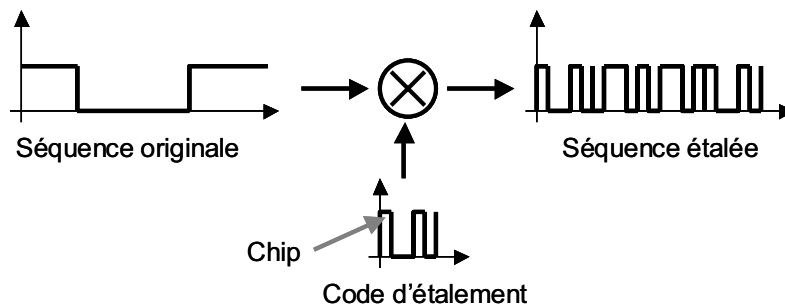


Fig. 4.3. Etalement du signal.

Nous n'insisterons pas d'avantage sur les techniques de codages de la technologie W-CDMA car elles font partie du traitement en bande de base. Cette partie traitement numérique ne sera pas décrite car elle s'écarte du sujet de ce mémoire.

La séquence numérique encodée doit ensuite être modulée afin de pouvoir être transportée vers un récepteur distant. Le type de modulation utilisée pour convertir les séquences numériques en signaux analogiques est la modulation QPSK⁶. Cette modulation utilise la modulation d'amplitude et de phase. Le codage QPSK peut être représenté sous forme d'un diagramme de constellation à quatre points situés à équidistance autour du cercle unitaire (Fig. 4.4). Le signal est ainsi constitué de deux composantes, la partie I (In-Phase) et la partie Q (Quadrature-phase). Avec quatre phases, ce type de modulation permet de coder deux bits par symboles [4.3]. Ainsi lors de la démodulation d'un signal codé QPSK, nous obtenons une donnée sur la voie I et une donnée sur la voie Q. La modulation QPSK est définie par l'équation suivant :

$$s(t) = A_0 \cos(2\pi f_c + (2i - 1) \cdot \frac{\pi}{4}) \text{ avec } i = 1, 2, 3, 4. \quad (\text{Eq. 4.1})$$

Avec A_0 l'amplitude du signal, et f_c la fréquence de la porteuse.

⁶ Quadrature Phase Shift Keying.

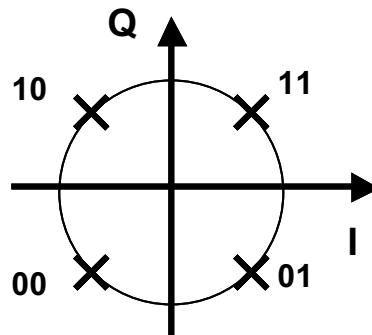


Fig. 4.4. Modulation QPSK (4 états).

4.2.2. Spécification des paramètres électriques d'après le standard UMTS

Les spécifications techniques de la technologie W-CDMA sont définies dans le document [4.4]. Dans cette partie, nous ne citerons que les principaux paramètres (utiles pour la modélisation VHDL-AMS). Le standard définit tout d'abord les différentes plages de fréquences présentées précédemment (Fig. 4.1) (Réception, Emission, FDD, TDD) ainsi que les espaces fréquentiels entre les canaux (5MHz). Le second type de paramètre spécifié correspond aux puissances minimales devant être démodulées par la partie réception ou aux puissances maximales pouvant être émises (en classe 1 : $P_{OUT} = 33\text{dBm}$). Cette puissance maximale admise dépend de nombreux critères tels que le mode de fonctionnement, le type de signal, le débit... Le gabarit du spectre d'émission est ensuite spécifié en définissant la valeur de l'ACLR⁷. Lorsqu'un signal est démodulé par un système, les caractéristiques non idéales du système vont déformer le signal et créer des lobes adjacents. L'ACLR permet de caractériser les non-linéarités du système en calculant le rapport de puissance des canaux adjacents sur la puissance du canal principal. Le standard UMTS définit pour cet ACLR une valeur maximale. Par exemple, pour un système de « classe 3 » (la norme UMTS classe les systèmes en fonction de leur puissance d'émission) et pour une mesure d'ACLR avec des canaux secondaires à $\pm 5\text{MHz}$, l'ACLR limite est fixé à 33dB. Un autre type de paramètres spécifié concerne l'EVM⁸. Il s'agit d'une mesure qui permet de qualifier la distorsion d'amplitude et de phase sur le diagramme de constellation des états (Fig. 4.4). L'EVM, exprimée en pourcentage, est définie comme le rapport entre le module du vecteur d'erreur et le module du vecteur de référence. La norme spécifie un nombre de symboles minimum permettant de faire cette mesure. Par exemple, l'EVM maximale est fixée à 17,5% pour un signal d'entrée ayant été préalablement dégradé par un filtrage RRC⁹ spécifique ; l'utilisation de filtre RRC correspond à un protocole classique permettant de mesurer l'EVM. D'autres paramètres tel que le BER¹⁰ ou le DPCH¹¹ sont aussi utilisés pour spécifier les comportements « limites » que peuvent avoir les systèmes développés afin d'utiliser le standard UMTS pour de la communication sans fil.

⁷ Adjacent Channel Leakage power Ratio.

⁸ Error Vector Magnitude.

⁹ Root Raised Cosine.

¹⁰ Bit Error Rate.

¹¹ Dedicated Physical Channel.

4.2.3. Architecture de la partie réception du système W-CDMA

Le standard [4.4] évoqué dans la partie précédente est le document de référence des concepteurs. En effet, cet ensemble de spécifications permet de définir l'architecture du système. Les concepteurs divisent le système en blocs lors de la définition de l'architecture puis ils déterminent les caractéristiques et les valeurs des paramètres des différents blocs en utilisant la technique de budgétisation présentée dans le chapitre II (§2.3.4). Par exemple, l'ACLR, décrite dans la partie précédente, doit être répartie sur les différents blocs du système. Cette étape de budgétisation est loin d'être triviale car l'ACLR doit être converti en d'autres paramètres tels que le point de compression, les produits d'intermodulations, les couplages parasites... Le choix des paramètres spécifiés se fait en déterminant ceux qui influent sur le comportement du système lors de la validation de conception. Le concepteur doit ensuite choisir la distribution finale des paramètres en fonction des performances des blocs qui sont à sa disposition comme décrit dans le chapitre II (§2.3.3).

L'architecture utilisée pour la partie réception du système W-CDMA est une architecture RF classique à Fréquence Intermédiaire nulle (zéro FI), on parle d'architecture de réception à conversion directe. La technologie zéro FI consiste à convertir le signal RF directement en signal basse-fréquence. Contrairement aux systèmes à FI non nulle, elle permet de diminuer la taille du circuit car elle ne réalise qu'une seule conversion à une fréquence inférieure et donc nécessite l'utilisation d'un seul mélangeur pour transposer le signal vers les fréquences basses.

Cette technologie présente toutefois l'inconvénient de conserver un offset parasite qui va perturber la démodulation et risque d'entraîner des problèmes de saturation des étages amplificateurs. Pour compenser cet effet, un système de compensation d'offset est utilisé. Son principe consiste à mesurer l'offset et à l'annuler en utilisant des blocs de compensation configurables. La valeur totale du gain de la chaîne de réception est configurée en fonction de la puissance du signal reçue à l'entrée du système ; plusieurs amplificateurs à gains variables sont présents dans notre système pour permettre cette fonctionnalité. Ces deux fonctions (compensation d'offset et gain ajustable) sont réalisées à l'aide de registres numériques qui reçoivent la valeur à appliquer sur le système. Les registres sont gérés par un système de contrôle qui est inclus dans la partie en bande de base ; cette fonctionnalité externe à notre SoC ne sera pas développée durant cette étude.

L'architecture de la partie réception est représentée sur Fig. 4.5. Elle est constituée d'un amplificateur faible bruit (LNA¹²). La fonction principale de ce bloc est d'amplifier le signal RF en ajoutant le minimum de bruit. Ce premier étage d'amplification est primordial car la dégradation du rapport signal sur bruit du signal d'entrée, de très faible puissance, ne pourra plus être corrigée par les étages suivants qui ne peuvent que dégrader encore, certes dans une moindre mesure, le signal utile.

Le bloc suivant est le filtre RF, il s'agit d'un filtre passe bande qui permet de supprimer les fréquences indésirables situées à l'extérieur de la bande utile du signal. Lors de la conception, il a été choisi d'utiliser un filtre externe afin de diminuer la complexité du SoC. De plus, même si le filtre utilisé est un système externe qui augmente la taille du produit final, celui-ci utilise une

¹² Low Noise Amplifier.

technologie à ondes acoustiques de surfaces (SAW¹³) qui est plus performante que les filtres intégrés classiques. Cette technologie permet de réaliser des filtres RF à bandes étroites d'ordres élevés (supérieurs à 80) qui seraient difficiles à mettre en œuvre avec des éléments discrets.

L'élément suivant est constitué de deux mélangeurs, il permet de convertir le signal RF en deux signaux I et Q basse-fréquence comme définis dans la partie §4.2.1. Pour cela, chaque mélangeur multiplie le signal RF avec le signal provenant de l'oscillateur local. Cette multiplication revient à soustraire la fréquence de l'oscillateur local de la fréquence du signal d'entrée et donc à récupérer le signal démodulé. L'oscillateur local fournit deux signaux déphasés de 90° qui sont appliqués aux mélangeurs. La sortie du premier mélangeur fournira alors le signal de la voie I et celle du second celui de la voie Q (Fig. 4.5). Les signaux provenant de l'oscillateur local sont générés à partir d'une PLL¹⁴. Les mélangeurs intègrent une fonction filtrage passe bas qui est liée à leur structure interne.

Le compensateur d'offset est situé directement après le mélangeur, il permet de supprimer comme précédemment expliqué l'offset parasite dû à la conversion directe. Ce bloc doit être situé avant l'étage d'amplification secondaire car la compensation d'offset a pour objectif d'éviter la saturation de ces amplificateurs.

Le système est ensuite composé de deux étages d'amplification chacun constitué d'un amplificateur à gain variable et d'un filtre passe bas. Ces étages ont pour effet d'amplifier le signal démodulé en fonction de la valeur contenu dans les registres de contrôle et de supprimer les fréquences parasites.

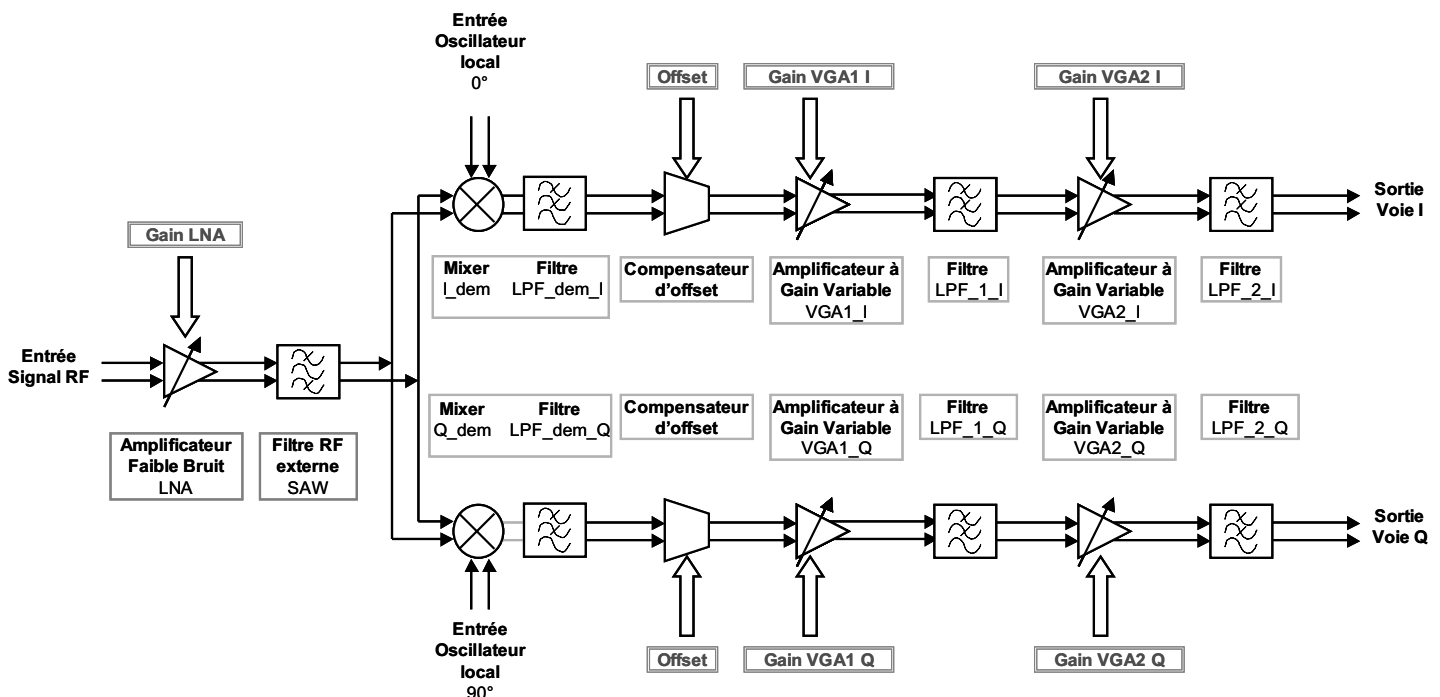


Fig. 4.5. Schéma de la partie réception du système W-CDMA.

4.3. Description du modèle comportemental

Nous allons maintenant présenter la modélisation comportementale d'un système W-CDMA utilisé pour évaluer la méthode de test développée dans ce manuscrit. Dans le chapitre II (§2.4), le flot de

¹³ Surface Acoustic Waves.

¹⁴ Phase Locked Loops.

conception et les différents niveaux d'abstraction utilisés pour décrire un système électrique ont été présentés. Le principe consiste à identifier les paramètres qui devront être vérifiés afin de respecter le standard UMTS [4.4], ces paramètres seront alors modélisés. La norme définit une puissance minimale de réception en dessous de laquelle la démodulation ne peut se faire correctement. Il est ainsi possible de définir le gain total de la chaîne de réception en considérant la tension minimale de sortie. Pour vérifier le gain total de la chaîne, il faut donc modéliser les différents amplificateurs qui la constituent, mais aussi les impédances d'entrées et de sorties ainsi que les coefficients de réflexion qui risquent d'atténuer les signaux. Il faut aussi considérer les différentes atténuations introduites par les filtres, le point de compression des amplificateurs... Pour vérifier l'ACLR, il faut définir les différentes non-linéarités du système : point de compression, intermodulations d'ordre 2 et 3, couplages parasites, fréquence de coupure des filtres. Et pour terminer sur les paramètres importants pour la modélisation comportementale, nous avons vu que l'offset parasite peut entraîner la saturation des amplificateurs et donc dégrader le signal ; il est important de modéliser les offsets des différents blocs afin de vérifier le bon dimensionnement du compensateur d'offset.

Pour conclure ce paragraphe, nous pouvons dire que le modèle comportemental est composé à la fois de paramètres fonctionnels et de paramètres électriques. Ces derniers paramètres caractérisent les comportements non-idéaux des différents blocs du système. Les paramètres fonctionnels sont les gains des différents blocs et leurs fréquences de coupure. Les paramètres électriques sont le point de compression, les intermodulations d'ordre 2 et 3, le couplage entre fils, les offsets, les impédances et les coefficients de réflexion. Pour résumer, nous pouvons dire que les paramètres utilisés pour la définition du modèle comportemental sont les paramètres qui ont été spécifiés durant la définition du plan de vérification et qui permettent de décrire les spécifications des différents blocs.

4.4. Modélisation comportementale des paramètres

Les paramètres comportementaux identifiés dans la partie précédente sont ceux présents dans notre modèle comportemental du système W-CDMA. Chaque bloc du système est décrit au niveau comportemental en utilisant le langage de description VHDL-AMS [4.6]. Les paramètres définis dans la partie précédente sont présents dans la plupart des blocs du système ; nous avons donc choisi de décrire non pas la modélisation de chaque bloc, mais plutôt la modélisation de chaque paramètre. La modélisation correspond à la transposition d'un paramètre à l'aide d'équations de Kirchhoff [4.7]. Cette conversion sera développée dans cette partie pour plusieurs paramètres. La validation de la modélisation de chaque paramètre est ensuite effectuée par simulation. Les modèles ont été développés grâce à l'outil de Mentor GraphiCs ADvanced MS RF [4.8] et plus particulièrement grâce à l'utilisation de la bibliothèque associée Commlib RF [4.9].

4.4.1. Modélisation et validation du gain

4.4.1.1. Modélisation

Le gain est un des paramètres le plus simple à modéliser dans un système électrique. Ce paramètre permet de spécifier le niveau d'amplification qui est réalisé sur le signal d'entrée. Suivant le type de composant décrit, il correspond soit à un gain en tension, soit un gain en puissance. Dans la plupart des cas il est défini en dB ; ce paramètre doit donc le plus souvent être converti en un coefficient linéaire A avant d'être inséré dans l'équation liant la tension de sortie à la tension d'entrée (Eq. 4.2).

$$V_{out} = A.V_{in} \quad (\text{Eq. 4.2})$$

Avec A suivant que le paramètre Gain soit exprimé en tension ou en puissance :

$$\text{Gain en tension : } A = \sqrt{10^{\frac{\text{Gain}}{20}}} \quad (\text{Eq. 4.3})$$

$$\text{Gain en puissance : } A = \sqrt{10^{\frac{\text{Gain}}{10}} \cdot \frac{R_{in}}{R_{out}}} \quad (\text{Eq. 4.4})$$

Dans (Eq. 4.3), le paramètre Gain est un gain en tension, le calcul du paramètre A liant la tension d'entrée avec la tension de sortie se détermine directement. Le gain est généralement défini en tension lorsque le bloc fonctionne à basse-fréquence. Dans ce cas, il n'y a pas d'atténuation due à l'adaptation d'impédance quand l'impédance d'entrée est très faible (idéalement nulle) et quand l'impédance de sortie est très grande (idéalement infinie). Dans (Eq. 4.4), le paramètre Gain est un gain en puissance. Il faut donc utiliser les impédances d'entrées et de sorties pour transformer la puissance en tension. Le gain est défini en puissance lorsqu'il s'agit de systèmes RF car des phénomènes de réflexion apparaissent. Dans ce cas, l'adaptation d'impédance doit être respectée afin de transmettre le maximum de puissance.

4.4.1.2. Validation de la description

La validation de la modélisation du gain consiste à intégrer le paramètre « Gain » modélisé dans la description d'un amplificateur classique et à extraire la valeur du paramètre modélisé à partir de la réalisation de mesures. Le banc de test est constitué de l'amplificateur modélisé par les résistances d'entrée et de sortie adaptées à 50Ω et le coefficient linéaire d'amplification (A), d'une source mono-ton adaptée à 50Ω , et d'une charge de 50Ω (Fig. 4.6). Le gain est défini en mesurant la puissance du signal entrant et la puissance du signal sortant. Lors de cette mesure, nous avons choisi de faire varier la puissance d'entrée afin de montrer qu'un amplificateur idéal possède un gain constant quelque soit la puissance d'entrée. La simulation se fait avec l'outil ADvanced MS RF et le type d'analyse est une analyse en régime établi (SST). Ce type d'analyse est suffisant car le système doit être dans un état stable et que l'observation de la puissance à la fréquence du fondamental suffit.

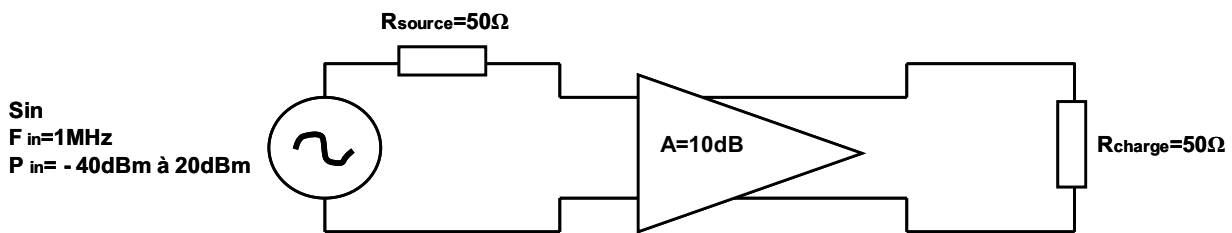


Fig. 4.6. Mesure du gain d'un amplificateur.

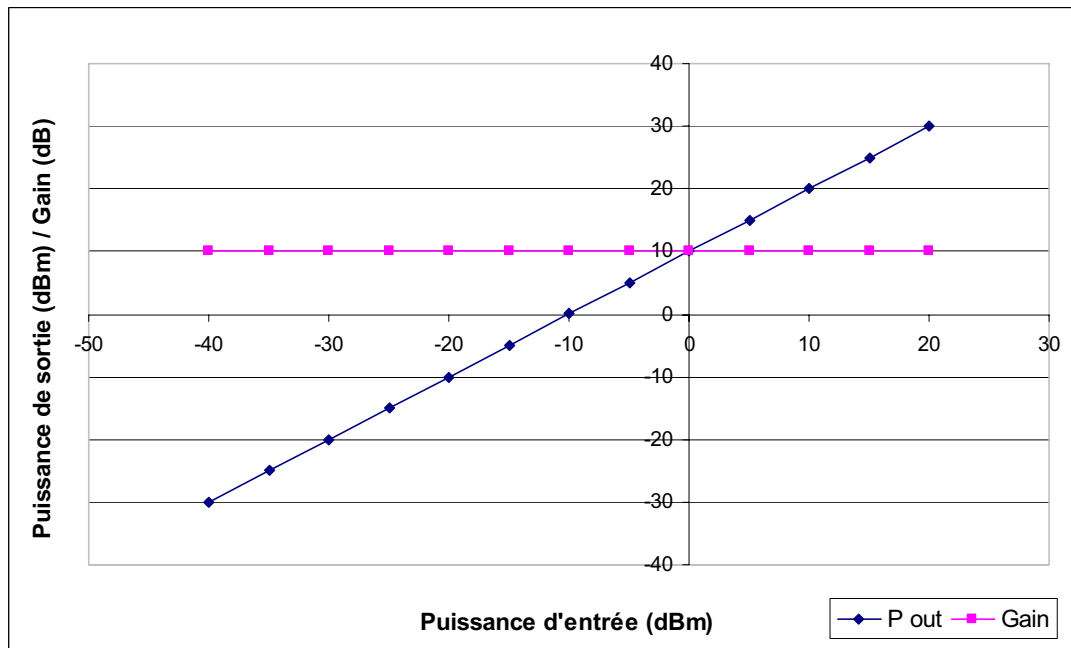


Fig. 4.7. Validation de la modélisation du gain d'un amplificateur.

Cet amplificateur idéal a été simulé avec un signal sinusoïdal de 1MHz et des puissances d'entrée différentes allant de -40dBm à 20dBm . La puissance de sortie ainsi que le gain sont mesurés pour chaque puissance d'entrée. La Fig. 4.7 nous permet de valider la modélisation du gain de l'amplificateur car le gain mesuré vaut 10dB quelque soit la puissance d'entrée. Les équations (Eq. 4.3) et (Eq. 4.4) permettent ainsi de modéliser le gain d'un système électrique.

4.4.2. Modélisation et validation du point de compression

4.4.2.1. Modélisation

Le point de compression permet de caractériser la saturation d'un système. Cette valeur spécifie la puissance d'entrée maximale qui entraîne une puissance de sortie située à 1dB au dessous de la puissance de sortie théorique (c'est-à-dire valeur de sortie sans saturation ni autres effets non-linéaires).

La modélisation du point de compression se fait en ajoutant une saturation sur le signal de sortie. Lorsque la puissance d'entrée est supérieure au point de compression, elle est limitée à la valeur du point de compression. La principale étape consiste à convertir le point de compression exprimé en dBm en tension de saturation. Pour cela, nous utilisons les équations suivantes :

$$P_{t_compression_lineaire} = \sqrt{10^{\frac{P_{t_compression}(dBm)-30}{10}}} \quad (\text{Eq. 4.5})$$

$$V_{saturation} = \sqrt{2.R.P_{t_compression_lineaire}} \quad (\text{Eq. 4.6})$$

La Fig. 4.8 représente une partie de la description du paramètre « point de compression à 1dB ». La constante « $v_{saturation}$ » est calculée à partir des équations (Eq. 4.5), (Eq. 4.6) et à partir de la valeur du point de compression spécifiée par le concepteur.

```

-- Compression point at 1dB
if (v_interne >= v_saturation) use
    v_out == v_saturation;
elsif (v_comprese <= -v_saturation) use
    v_out == -v_saturation;
else
    v_out == v_interne;
end use;

```

Fig. 4.8. Modélisation du point de compression.

Dans le cas où le point de compression n'est pas spécifié, il faut tout de même insérer un comportement de saturation liée à la tension d'alimentation du système. Ainsi, la même syntaxe est utilisée pour faire saturer le signal de sortie lorsque celui-ci est supérieur à la tension d'alimentation.

4.4.2.2. Vérification de la description

La vérification est réalisée de la même façon que dans la partie §4.4.1.2. La mesure du point de compression se fait en faisant varier la puissance d'entrée et en mesurant le gain de l'amplificateur (Fig. 4.9). Le point de compression est extrait en déterminant la puissance d'entrée qui engendre un gain situé à -1dB du gain théorique.

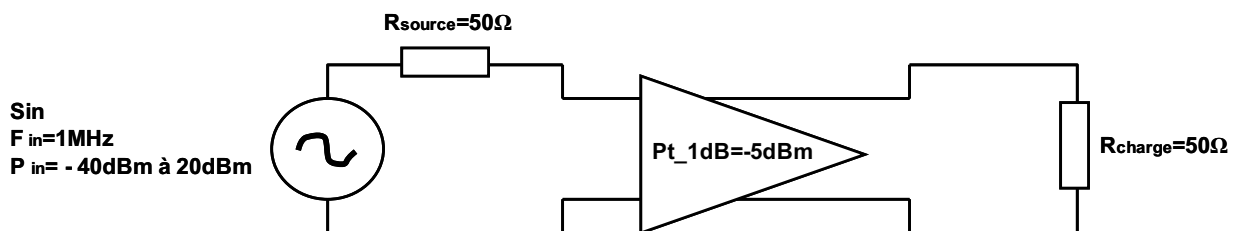


Fig. 4.9. Mesure du point de compression de l'amplificateur.

Lors de la simulation, la puissance d'entrée varie de -40dBm à 20dBm, le pas a été diminué autour de -5dBm afin de pouvoir déterminer avec plus de précision la valeur du gain et donc du point de compression. D'après la Fig. 4.10, pour une puissance d'entrée de -5dBm, le gain mesuré est de 8.98dB. Le gain de l'amplificateur a été défini comme dans la partie précédente à 10dB, nous sommes donc au point de compression à -1dB. Ces résultats de simulation nous permettent ainsi de valider la modélisation VHDL-AMS du point de compression pour l'amplificateur décrit dans la Fig. 4.8.

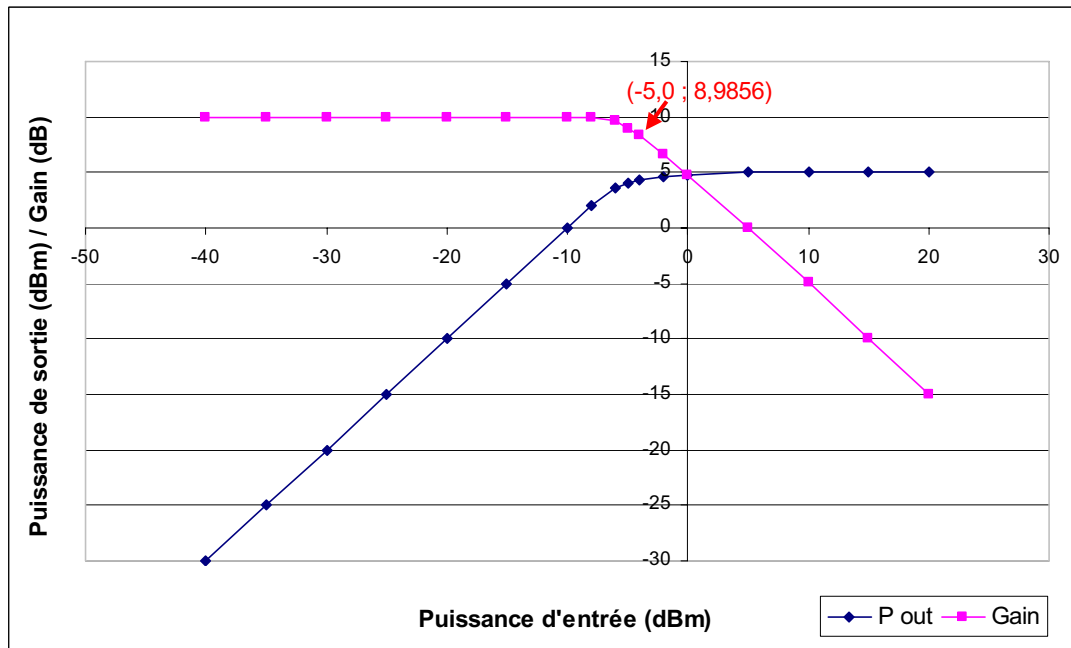


Fig. 4.10. Validation de la modélisation du point de compression d'un amplificateur.

4.4.3. Modélisation et validation de l'intermodulation d'ordre trois

4.4.3.1. Modélisation

Pour terminer la présentation des paramètres modélisés, nous traiterons le cas de l'intermodulation d'ordre trois. En effet, dans le cas des systèmes non-linéaires, des raies parasites peuvent apparaître à différentes fréquences multiples de la fréquence fondamentale : ce sont les harmoniques. Les harmoniques d'ordres supérieures à trois ne sont pas traitées car leur puissance n'est généralement pas suffisamment élevée pour avoir un impact sur le signal fondamental. Au contraire, les harmoniques d'ordre trois sont importantes car elles entrent souvent dans la bande passante du système et perturbent alors le signal utile.

L'intermodulation d'ordre trois est un paramètre qui permet de modéliser les effets de la non-linéarité d'un système électrique. En effet, l'équation liant le signal d'entrée au signal de sortie n'est plus définie simplement comme dans (Eq. 4.2) mais devient :

$$V_{out} = A.V_{in} + A_2.V_{in}^2 + A_3.V_{in}^3 + \dots \quad (\text{Eq. 4.7})$$

Dans notre cas, nous considérons que le paramètre A_2 est nul car nous ne modélisons pas l'effet de l'intermodulation d'ordre deux. Pour voir l'effet de la non-linéarité dans la bande passante du système, nous devons utiliser en entrée un signal bi-ton comme défini ci-dessous :

$$V_{in} = V_1.\cos w_1 t + V_2.\cos w_2 t \quad (\text{Eq. 4.8})$$

En remplaçant V_{in} dans (Eq. 4.7) et avec A_2 nul nous obtenons :

$$V_{out} = A.(V_1.\cos w_1 t + V_2.\cos w_2 t) + A_3.(V_1.\cos w_1 t + V_2.\cos w_2 t)^3 \quad (\text{Eq. 4.9})$$

En développant le second terme, nous obtenons :

$$\begin{aligned}
(V_1 \cdot \cos w_1 t + V_2 \cdot \cos w_2 t)^3 &= V_1^3 \cdot \left(\frac{3 \cdot \cos w_1 t}{4} + \frac{\cos 3w_1 t}{4} \right) + V_2^3 \cdot \left(\frac{3 \cdot \cos w_2 t}{4} + \frac{\cos 3w_2 t}{4} \right) \\
&+ V_1^2 \cdot V_2 \cdot \left[\frac{3}{2} \cdot \cos w_2 t + \frac{3}{4} \cdot \cos(2 \cdot w_1 - w_2) t + \frac{3}{4} \cdot \cos(2 \cdot w_1 + w_2) t \right] \\
&+ V_1 \cdot V_2^2 \cdot \left[\frac{3}{2} \cdot \cos w_1 t + \frac{3}{4} \cdot \cos(2 \cdot w_2 - w_1) t + \frac{3}{4} \cdot \cos(2 \cdot w_2 + w_1) t \right]
\end{aligned} \tag{Eq. 4.10}$$

La factorisation de l'équation (Eq. 4.9) nous permet d'obtenir la réponse d'un signal bi-ton amplifié par un système non linéaire :

$$\begin{aligned}
V_{out} &= \left(A \cdot V_1 + A_3 \cdot \left(V_1^3 \cdot \frac{3}{4} + V_1 \cdot V_2^2 \cdot \frac{3}{2} \right) \right) \cdot \cos w_1 t \\
&+ \left(A \cdot V_2 + A_3 \cdot \left(V_2^3 \cdot \frac{3}{4} + V_1^2 \cdot V_2 \cdot \frac{3}{2} \right) \right) \cdot \cos w_2 t \\
&+ A_3 \cdot V_1^2 \cdot V_2 \cdot \frac{3}{4} \cdot \cos(2 \cdot w_1 - w_2) t \\
&+ A_3 \cdot V_1 \cdot V_2^2 \cdot \frac{3}{4} \cdot \cos(2 \cdot w_2 - w_1) t \\
&+ A_3 \cdot V_1 \cdot V_2 \cdot (V_1 + V_2) \cdot \frac{3}{4} \cdot \cos(2 \cdot w_1 + w_2) t
\end{aligned} \tag{Eq. 4.11}$$

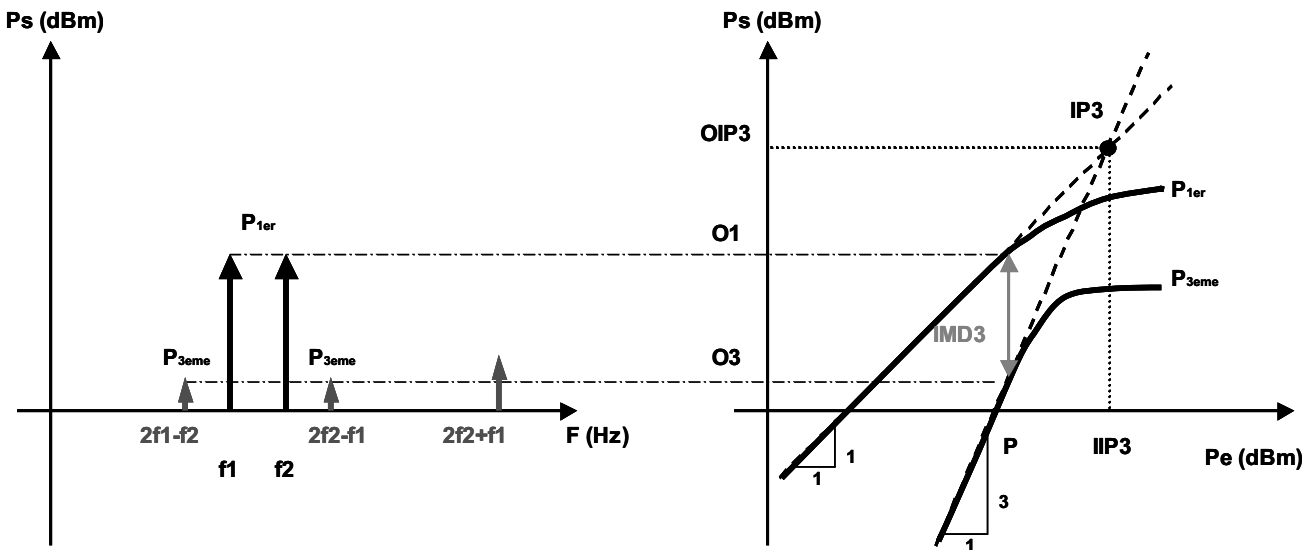


Fig. 4.11. Réponse fréquentielle et réponse en puissance de l'intermodulation d'ordre trois.

En sortie du système non-linéaire, nous obtenons un signal composé de cinq raies de fréquence (Fig. 4.11) et équation (Eq. 4.11). Nous pouvons constater que l'intermodulation d'ordre trois modifie la valeur des raies fondamentales (w_1 et w_2) en ajoutant le terme $A_3 \cdot \left(V_1^3 \cdot \frac{3}{4} + V_1 \cdot V_2^2 \cdot \frac{3}{2} \right)$. Il faut aussi noter que si les fréquences w_1 et w_2 sont proches alors les raies situées à $2w_1 - w_2$ et $2w_2 - w_1$ se trouvent dans la bande passante (proches de w_1 et w_2) et vont perturber le signal (Fig. 4.11).

L'IMD3¹⁵ est définie comme le rapport entre la puissance de la raie fondamentale sur la puissance

¹⁵ Intermodulation distortion from the third harmonic.

des raies dues à la non-linéarité d'ordre trois. L'intermodulation peut aussi être définie sous le terme IIP3¹⁶, il s'agit de la puissance d'entrée qui permet d'obtenir une puissance de la raie due à l'ordre 3 identique à la puissance du fondamental (Fig. 4.11). En réalité, à cause de la compression ce point n'existe pas, mais il peut être représenté en considérant une extrapolation des courbes de puissance de sortie de la fréquence fondamentale (P_{1er}) et de l'ordre trois (P_{3eme}) [4.10]. Au point d'intersection, nous avons :

$$A.V_1 + A_3.(V_1^3 \cdot \frac{3}{4} + V_1.V_2^2 \cdot \frac{3}{2}) = A_3.V_1^2.V_2 \cdot \frac{3}{4} \quad (\text{Eq. 4.12})$$

L'IIP3 étant la puissance d'entrée correspondant au point d'intersection, nous définissons

$$V_1 = V_2 = \sqrt{2.R.10^{\frac{IIP3-30}{10}}} \quad (\text{Eq. 4.13})$$

En posant

$$A.V_1 \gg A_3.(V_1^3 \cdot \frac{3}{4} + V_1.V_2^2 \cdot \frac{3}{2}) \quad (\text{Eq. 4.14})$$

Nous obtenons finalement :

$$A_3 = \frac{2}{3} \cdot \frac{A}{R.10^{\frac{IIP3-30}{10}}} \quad (\text{Eq. 4.15})$$

L'intermodulation d'ordre trois est modélisée en utilisant la valeur de l'IIP3, elle est ensuite convertie en paramètre d'amplification A_3 comme démontré ci-dessus (Eq. 4.15).

4.4.3.2. Vérification de la description

De même que lors de la description des paramètres précédents, nous allons valider cette modélisation en simulant un amplificateur possédant une intermodulation d'ordre trois. Le banc de test est semblable aux précédentes parties, nous avons cependant modifié la source en utilisant une source bi-tons.

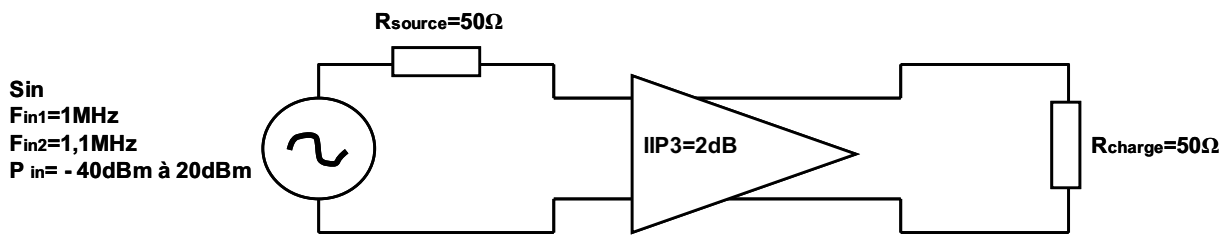


Fig. 4.12. Mesure du point d'intersection d'ordre trois de l'amplificateur.

La validation de la modélisation de l'IIP3 est effectuée en simulant le système avec des puissances d'entrée variables. La mesure graphique de l'IIP3 se fait en traçant les asymptotes de la puissance de sortie mesurée à la fréquence de la raie fondamentale (F_{in1}) et à la fréquence des raies d'ordre 3 définies à $2F_{in1}-F_{in2}$. La valeur de l'IIP3 mesurée est la puissance d'entrée (axes des abscisses) correspondant au point d'intersection des deux asymptotes, ce point situé à 2dBm (Fig. 4.13) correspond exactement à la valeur de l'IIP3 défini dans notre modèle.

¹⁶ Input Intercept Point from the third harmonic.

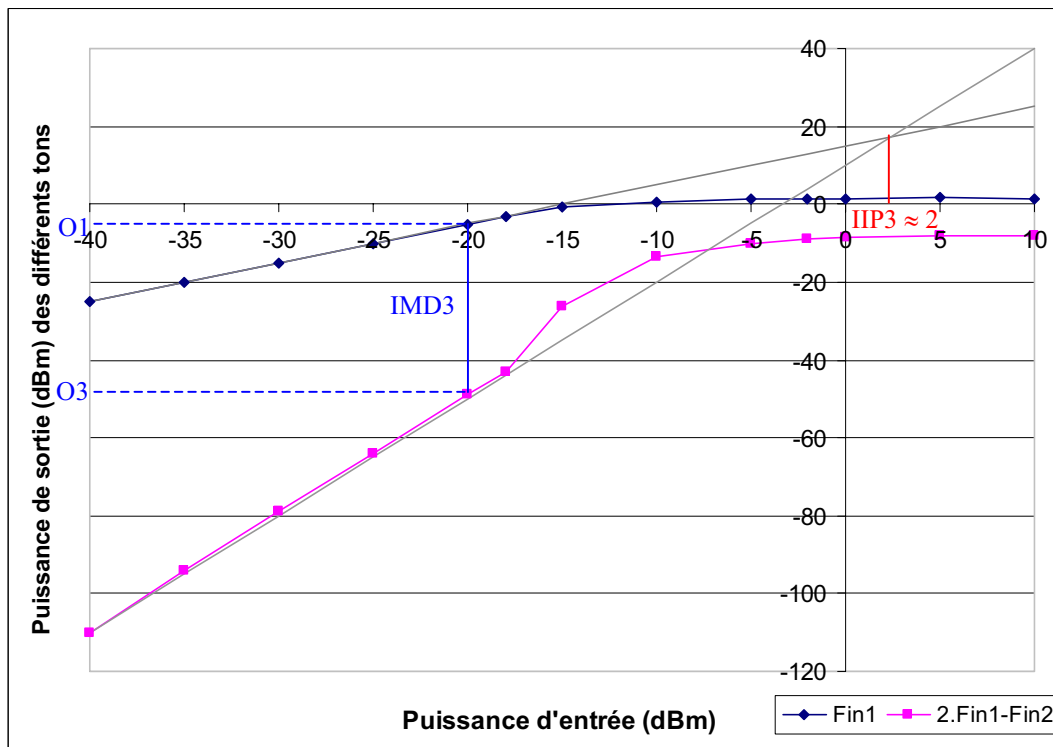


Fig. 4.13. Validation de la modélisation du point d'intersection d'intermodulation d'ordre trois IIP3.

L'utilisation de la méthode des asymptotes demande de simuler le système avec différentes puissances d'entrées c'est-à-dire de faire plusieurs simulations. Il est toutefois possible de mesurer l'IIP3 pour une seule puissance d'entrée mais de façon indirecte, il faut d'abord mesurer la puissance de la raie fondamentale et la puissance de la raie due à l'intermodulation d'ordre trois, nous obtenons ainsi l'IMD3. Le point d'intersection se calcule en définissant les équations linéaires des courbes de puissance. Par définition de l'ordre un et trois, nous pouvons dire que la pente de l'ordre trois est trois fois plus élevée que celle du fondamental (Fig. 4.13). Nous en déduisons les équations suivantes :

$$Y_{1er} = (X - P) + O_1 \quad (\text{Eq. 4.16})$$

$$Y_{3eme} = (3X - P) + O_3 \quad (\text{Eq. 4.17})$$

Au point d'intersection $Y_{1er} = Y_{3eme}$:

$$(X - P) + O_1 = (3X - P) + O_3 \quad (\text{Eq. 4.18})$$

$$X = \frac{O_1 - O_3 - P}{2} \quad (\text{Eq. 4.19})$$

$$IIP3 = \frac{IMD3}{2} - P$$

A partir des résultats de simulation pour une puissance d'entrée de -20dBm , nous pouvons déterminer la valeur de l'IIP3 de l'amplificateur (Fig. 4.13).

$$IIP3 = \frac{-5.166 + 49}{2} - (-20) = 1.917\text{dBm} \quad (\text{Eq. 4.20})$$

Ces résultats de simulation nous ont permis de valider la modélisation de l'intermodulation d'ordre trois. La méthode appliquée pour la modélisation de l'intermodulation d'ordre 2 est sensiblement identique et ne sera pas développée.

4.4.4. Modèle comportemental de l'amplificateur

Avec les développements présentés dans les parties précédentes, il est possible d'élaborer un modèle comportemental d'un amplificateur caractérisé par un gain en puissance, un point de compression, une intermodulation d'ordre 3. Le bloc est aussi constitué de paramètres tels que les impédances d'entrée et de sortie, les coefficients de réflexion et la fréquence de coupure... La modélisation des autres paramètres comportementaux ne sera pas développée car le principe de modélisation est toujours le même. La Fig. 4.14 présente la description VHDL-AMS de l'amplificateur.

```

-----
--
-- BLOCK      : Amplifier
-- VHDL-AMS   : amplifieur.vhd
-----
-- LIBRARY
--
Library IEEE;
Library IEEE_proposed;
Use IEEE.MATH_REAL.ALL;
Use IEEE_proposed.electrical_systems.all;
Use IEEE.STD_LOGIC_1164.ALL;
-----
-- ENTITY
ENTITY amplifieur IS

    GENERIC ( g_Zin : real := 50.0;      -- Input Impedance
              g_Zout : real := 50.0;    -- Output Impedance
              g_f0 : real := 2.68e6;    -- Cut off frequency 2nd Order Filter
              g_Gain : real := 15.0;    -- Power Gain (dB)
              g_S11 : real := -10.0;    -- Input Return Losses (dB)
              g_S22 : real := -8.0;     -- Output Return Losses (dB)
              g_P_compression : real := -9.0; -- 1dB Input Compression Point (dBm)
              g_IIP3 : real := 1.0      -- Input IP3 (dBm)
            );
    PORT ( terminal t_in_p, t_out_p : electrical);
END ENTITY amplifieur;
-----
-- ARCHITECTURE
ARCHITECTURE bhv OF amplifieur IS

Quantity v_in across i_in through t_in_p;
Quantity v_out across i_out through t_out_p;
Quantity q_v_filter, q_v_gain, q_v_compression, q_v_out_imp : real;

Constant c_S11_real : real := 10.0**(g_S11/20.0);
Constant c_S22_real : real := 10.0**(g_S22/20.0);
Constant c_num : real_vector := (1.0 , 0.0 );
Constant c_den : real_vector := (1.0 , 2.0/(MATH_2_PI*g_f0) ,
2.0/((MATH_2_PI*g_f0)*(MATH_2_PI*g_f0)) ,
1.0/((MATH_2_PI*g_f0)*(MATH_2_PI*g_f0)*(MATH_2_PI*g_f0)));
Constant c_a1 : real := 2.0 * 10.0**(g_Gain/20.0) * sqrt(g_Zout/g_Zin);
Constant c_P_compression_real : real := 10.0**((-30.0+g_P_compression-
2.0)/10.0);
Constant c_V_compression : real := sqrt(2.0*c_P_compression_real*g_Zout);
Constant c_a3 : real := - 2.0/ 3.0 * c_a1 / (g_Zout * 10.0**((g_IIP3-
30.0)/10.0));

```

```

BEGIN
-----
---
-- Coefficient of reflexion and Characteristic Impedance
i_in == v_in / g_Zin * (1.0 + c_S11_real ) / (1.0 - c_S11_real);
i_out == q_v_out_imp/g_Zout * (1.0 + c_S22_real ) / (1.0 - c_S22_real);
-----
--
-- Filter
q_v_filter == v_in'ltf(c_num,c_den);
-----
--
-- Compression point
if (q_v_filter >= c_V_compression) use
    q_v_compresse == c_V_compression;
elsif (q_v_filter <= -c_V_compression) use
    q_v_compresse == -c_V_compression;
else
    q_v_compresse == v_in;
end use;
-----
--
-- Gain
q_v_gain == c_a1* q_v_compresse + c_a3 * (q_v_compresse ** 3.0) ;
v_out == q_v_gain + q_v_out_imp;

END ARCHITECTURE bhv;
-----

```

Fig. 4.14. Modèle comportemental décrit en VHDL-AMS d'un amplificateur.

Les différents blocs du système W-CDMA tel que le LNA, le mélangeur, les amplificateurs à gain variables ont été modélisés de la même façon que pour l'amplificateur. Les valeurs des paramètres de chaque bloc sont déterminées à partir des spécifications définies lors de la conception du système et plus précisément lors de la budgétisation. Les différents blocs sont assemblés et permettent de modéliser le système complet décrit au niveau comportemental. La prochaine étape consiste à valider la modélisation du système W-CDMA utilisé.

4.5. Validation du modèle comportemental du système W-CDMA

Dans cette partie, des résultats de simulation permettant de vérifier que le système développé respecte les spécifications et donc de valider la conception du système sont exposés. Le choix de la valeur des paramètres des différents blocs est ainsi validé.

4.5.1. Validation fonctionnelle

La première étape consiste à valider la fonctionnalité principale de la partie réception du système W-CDMA c'est-à-dire la démodulation du signal reçu. Pour cela, nous générons un signal modulé IQ. Cette modulation se fait en utilisant une source disponible avec ADvanced MS RF. Celle-ci permet de générer une séquence numérique préalablement définie et de la moduler au format QPSK à la fréquence désirée.

La validation de la partie réception est réalisée en comparant la séquence numérique d'entrée avec les signaux observés sur les sorties. En sortie du système nous obtenons un signal analogique basse-fréquence. Le banc de test utilisé est le suivant :

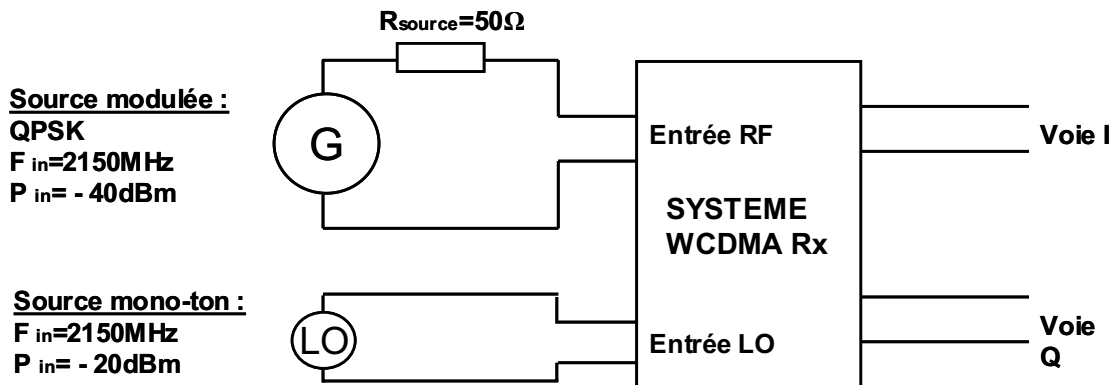


Fig. 4.15. Démodulation IQ par le système W-CDMA.

Lors de cette simulation, il n'est pas possible d'utiliser une simple analyse en régime établi (SST), car l'analyse aux fréquences fondamentales et aux harmoniques n'est pas suffisante pour pouvoir observer correctement le signal démodulé. Une analyse plus complexe utilisant la détection d'enveloppe est nécessaire, celle-ci permet de déterminer le spectre fréquentiel autour des différentes fréquences des fondamentales et des harmoniques. Sous l'outil de Mentor Graphics, cette analyse s'appelle la MODSST. Les courbes a) et b) de la Fig. 4.16 représentent les séquences numériques des voies respectives I et Q qui sont utilisées par la source modulée pour générer le signal RF injecté à l'entrée de la partie réception du système W-CDMA. La courbe c) représente le signal RF modulé au format QPSK appliqué à l'entrée du système. Les courbes d) et e) correspondent aux signaux démodulés sur les voies I et Q à la sortie des mélangeurs (Fig. 4.5). En sortie du système, nous pouvons observer les signaux I et Q démodulés (Fig. 4.16 f) et g)) ; en les comparant avec a) et b), nous pouvons conclure que la démodulation a permis de retrouver les séquences d'origines sans erreur. Cette simulation nous permet ainsi de valider fonctionnellement la modélisation de la partie réception du système W-CDMA.

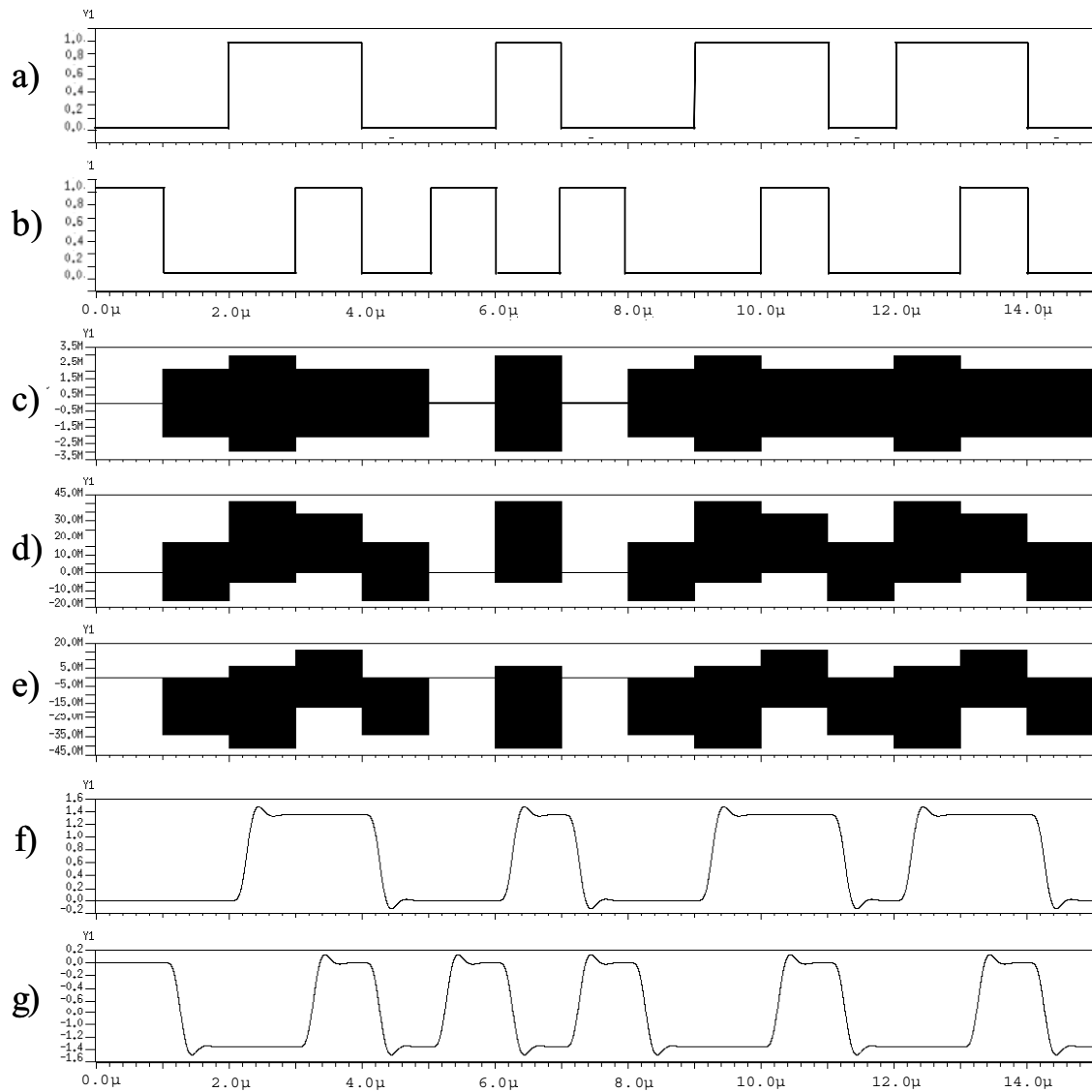


Fig. 4.16. Démodulation du signal modulé IQ.

4.5.2. Réjection des raies parasites

La seconde étape de validation consiste à évaluer la capacité du système à rejeter des raies parasites situées hors de la bande passante du signal. Pour cela nous utilisons une source multi-tons, la fréquence du signal fondamental est fixée à 2111MHz et celle des raies parasites sont fixées à 2113MHz et 2115MHz (Fig. 4.17). Le type d'analyse utilisé est une analyse en régime établi, elle permet de mesurer la puissance des raies parasites sur les sorties I et Q afin de vérifier leur atténuation. Cette étape va aussi permettre de valider le contrôle de gain. Pour cela, nous faisons varier la valeur du gain du bloc VGA1 et nous observons la puissance du signal de sortie. La Fig. 4.18 représente les tensions mesurées correspondantes à la raie fondamentale (1MHz) et aux raies parasites (3MHz et 5MHz). Tout d'abord, nous pouvons valider l'amplification à gain variable car celle-ci permet d'obtenir une plage d'amplification de 30dB c'est-à-dire un signal allant de -40dB à -10dBV pour la raie située à 1MHz. Ensuite, nous constatons que les raies parasites fixées à 2113MHz et 2115MHz ayant la même puissance initiale que la raie fondamentale (-50dBm) sont atténuées de 8dB et 29dB par rapport au fondamental. Cette simulation permet de valider la conception du système car les raies parasites sont suffisamment atténuées afin de respecter les

spécifications décrites par la norme concernant l'ACPR [4.4].

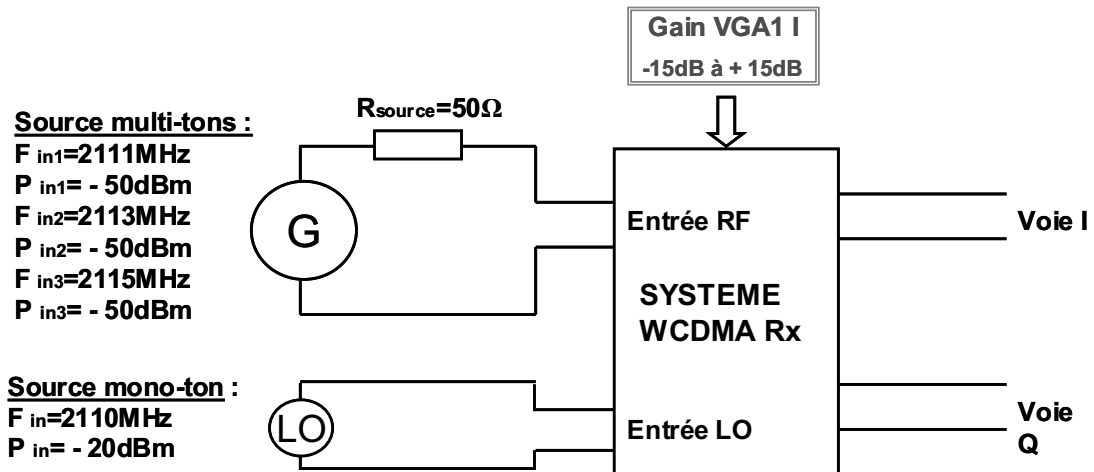


Fig. 4.17. Validation de la réjection de raies parasites.

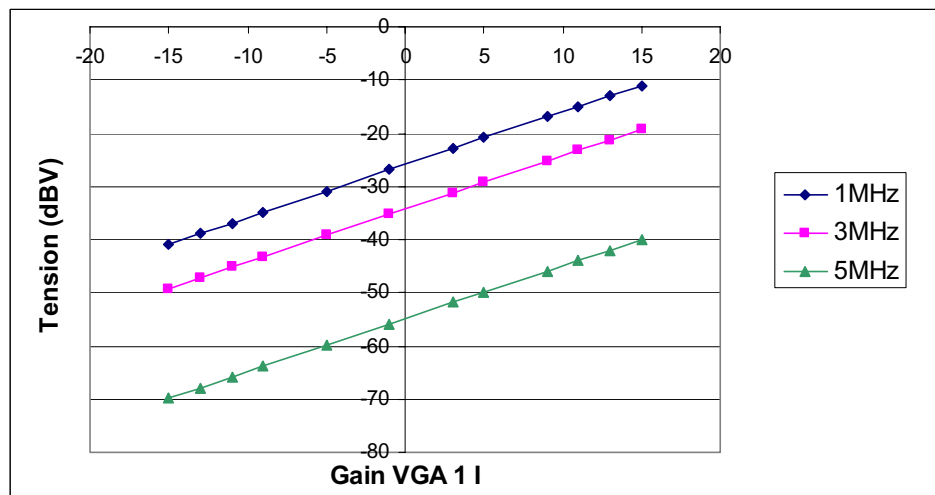


Fig. 4.18. Simulation de la réjection des raies parasites.

Dans cette partie, nous avons présenté la modélisation complète de la partie réception du système W-CDMA. Les différents blocs préalablement validés ont été assemblés afin de créer le système final. Un sous-ensemble des simulations effectuées pour la validation du design du système W-CDMA a été présenté ; dans [4.11], d'autres résultats de simulation sont présentés.

4.6. Validation du modèle comportemental à partir de mesures sur le circuit

L'objectif de cette partie est de montrer que le modèle comportemental présenté dans ce chapitre ne modélise pas uniquement le comportement spécifié par le cahier des charges comme présenté dans la partie ci-dessus, mais modélise aussi le comportement réel du circuit. Pour cela, nous avons repris les résultats de la validation du prototype c'est-à-dire les résultats de mesures réalisées directement sur le circuit réel. Le système a été conçu dans une approche Top-Down et développé avec la technologie BICMOS RF 0,25 μm . Le schéma du routage (layout) est présenté sur la figure (Fig. 4.19).

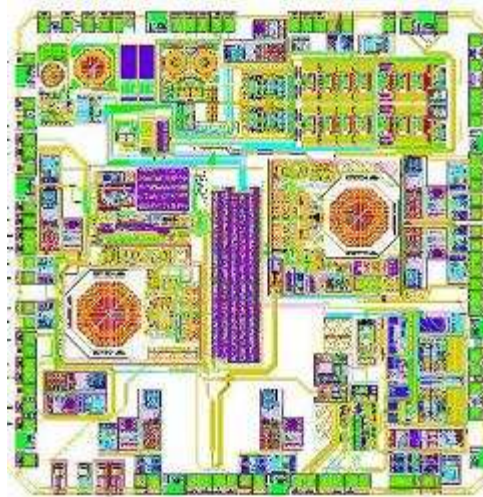


Fig. 4.19. Schéma du routage du système W-CDMA.

Nous allons présenter, à titre d'exemple, une des mesures qui ont permis de valider notre modèle ; il s'agit de la validation de l'IMD3 (Fig. 4.20). Le synoptique du banc de test utilisé est sensiblement le même que celui présenté sur la Fig. 4.17, les signaux générés par les sources multi-tons sont légèrement différents. La source reliée à l'entrée RF génère un signal bi-tons : $F_1=2123.4\text{MHz}$, $P_1=-40\text{dBm}$ et $F_2=2133.4\text{MHz}$, $P_2=-40\text{dBm}$ et la source reliée à l'entrée LO génère un signal mono-ton à 2112.4MHz . La mesure de l'IMD3 se fait en mesurant la puissance à 1MHz qui correspond à l'harmonique d'ordre 3 ($2F_1-F_2-F_{LO}$) et en faisant varier la valeur du gain du bloc VGA1. La Fig. 4.20 présente la réponse des simulations et les résultats obtenus lors de mesures sur le circuit réel. Cette figure montre que la valeur de l'IMD3 est la même à $\pm 3\%$.

Cette étape nous permet ainsi de valider la modélisation comportementale de notre système W-CDMA à partir de comparaison avec des mesures réalisées directement sur circuits.

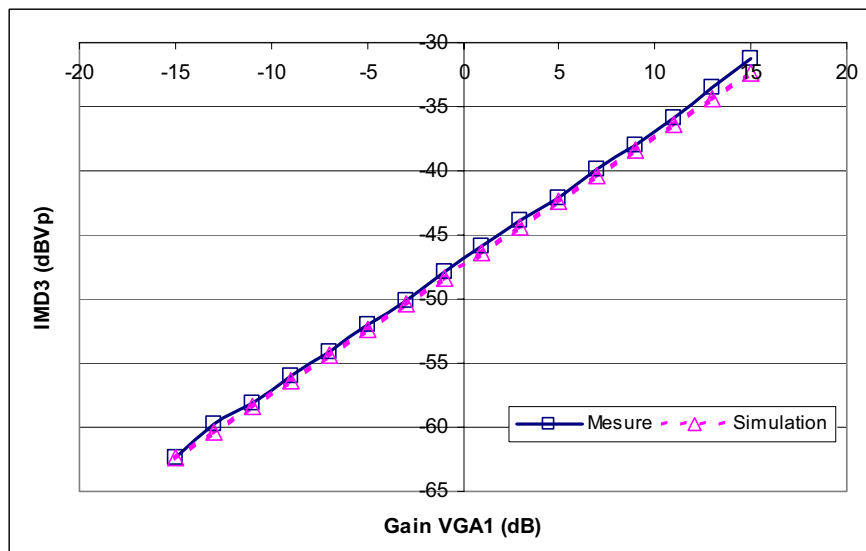


Fig. 4.20. Simulation et mesure de l'intermodulation d'ordre 3 en fonction du gain VGA1.

4.7. Conclusion

Dans ce chapitre, nous avons abordé la modélisation comportementale d'un système W-CDMA. Cette application RF sera reprise pour le développement de méthodes de test et plus particulièrement pour la génération de stimuli de test. Cette étape de modélisation est importante

pour le développement de méthodes de test basées sur l'injection et la simulation de fautes car le modèle utilisé a un impact direct sur la qualité des stimuli générés. Après une présentation rapide de la technologie W-CDMA, nous avons présenté la modélisation comportementale du SoC. Ce chapitre a été clos par la présentation de résultats de simulation permettant de valider le modèle du système en s'assurant que celui-ci modélise exactement les spécifications définies dans le cahier des charges et le comportement mesuré sur le système physique.

Bibliographie

- [4.1] M. Terre, "UMTS", Cours du Conservatoire National des Arts et Métiers, 2004.
- [4.2] S. Bauzac, "UMTS, la couche physique de l'interface radio", Cours Université de Savoie, 2003.
- [4.3] J. Auvray, "Les Techniques De Multiplexage", Université Pierre et Marie Curie IST SETI, 2001.
- [4.4] 3rd Generation Partnership Project; Technical Specification Group Radio Access Networks; User Equipment (UE) radio transmission and reception (FDD), <http://www.3gpp.org>, 2004.
- [4.5] K. S. Kundert, O. Zinke, "The Designer's guide to Verilog AMS", Cadence, pp 13-33, Juin 2004.
- [4.6] J.J. Charlot, H. Levy, N. Milet-Lewis, T. Zimmer, "VHDL-AMS for mixed technology and mixed signal, an overview", IEEE Mediterranean Conference on Control and Automation (MED01), Juin 2001.
- [4.7] L.W. Nagel, "SPICE2: A Computer program to simulate circuits", Memorandum No. ERL-M520, Electronics Research Laboratory, University of California, 1975.
- [4.8] "ADvance MSTM Reference Manual", Mentor Graphics, <http://www.mentor.com>
- [4.9] Mentor Graphics, "CommLib RF VHDL-AMS Library", Manual, Octobre 2005.
- [4.10] Ken KUNDERT, "Accurate and Rapid Measurement of IP₂ and IP₃", The Designer's Guide Community, www.designers-guide.org/Analysis/intercept-point.pdf, Mai 2002.
- [4.11] Y. Joannon, V. Berouille, R. Khouri, C. Robach, S. Tedjini, J-L. Carbonero, "Behavioral modeling of W-CDMA transceiver with VHDL-AMS language", Design and Diagnostics of Electronic Circuits and Systems (DDECS'06), pp. 113 - 118, Avril 2006.

CHAPITRE V :

QUALIFICATION DE STIMULI

POUR LA VALIDATION DE CONCEPTION

| | |
|---|-----|
| CHAPITRE V : | 81 |
| QUALIFICATION DE STIMULI POUR LA VALIDATION DE CONCEPTION | 81 |
| 5.1. Introduction | 84 |
| 5.2. Intérêt de la qualification de la vérification de conception | 84 |
| 5.3. Qualification des stimuli de vérification par injection et simulation de fautes | 86 |
| 5.3.1. PLASMA : outil d'aide à la génération de stimuli de test | 88 |
| 5.3.1.1. Simulation automatique | 89 |
| 5.3.1.2. Traitement des résultats de simulation | 91 |
| 5.3.1.3. Optimisation des temps de simulation | 91 |
| 5.3.2. Descriptions saines | 91 |
| 5.3.2.1. Echantillonnage pires-cas « simplifié » | 92 |
| 5.3.2.2. Echantillonnage par estimation gaussienne | 94 |
| 5.3.3. Modèle de fautes | 99 |
| 5.3.4. Métrique de qualification | 101 |
| 5.4. Résultats de qualification de stimuli pour la validation de conception | 103 |
| 5.4.1. Définition des paramètres comportementaux, choix des descriptions saines et fautives | 103 |
| 5.4.2. Stimuli de test | 104 |
| 5.4.3. Evaluation des plages de mesures de la population saine | 105 |
| 5.4.3.1. Définition des plages de mesures par échantillonnage pires-cas « simplifié » | 105 |
| 5.4.3.2. Définition des plages de mesures par estimation gaussienne | 106 |
| 5.4.4. Evaluation du modèle de fautes et de la métrique | 108 |
| 5.4.5. Résultats de qualification des stimuli | 111 |
| 5.4.5.1. Couverture de fautes | 111 |
| 5.4.5.2. Taux Relatif de Couverture des Paramètres | 114 |
| 5.4.5.3. Temps de simulation | 114 |
| Conclusion | 115 |
| Bibliographie | 115 |
| Annexes | 116 |

| | | |
|-------------------|--|------------|
| Fig. 5.1. | Qualification de stimuli dans le plan de vérification. | 85 |
| Fig. 5.2. | Flot de conception et de flot de vérification. | 86 |
| Fig. 5.3. | Injection et simulation de fautes. | 87 |
| Fig. 5.4. | Définition des plages de mesures pour la détection de fautes. | 87 |
| Fig. 5.5. | Vérification de conception de l'architecture. | 88 |
| Fig. 5.6. | Principe de fonctionnement de PLASMA. | 89 |
| Fig. 5.7. | Fichier de simulation. | 90 |
| Fig. 5.8. | Plages de mesure des descriptions saines. | 93 |
| Fig. 5.9. | Distribution gaussienne des paramètres du circuit. | 95 |
| Fig. 5.10. | Définition des plages de mesures de modèles sains. | 95 |
| Fig. 5.11. | Probabilité cumulée de la mesure déterminée à partir d'un échantillon de descriptions saines. | 99 |
| Fig. 5.12. | Définition de la faute paramétrique. | 102 |
| Fig. 5.13. | Paramètres comportementaux décrits dans le système W-CDMA. | 104 |
| Fig. 5.14. | Probabilité cumulée de l'estimation gaussienne de la population saine à partir de différents échantillons. | 107 |
| Fig. 5.15. | Erreur relative de l'estimation gaussienne pour les différents échantillons. | 107 |
| Fig. 5.16. | Plage de mesure des descriptions saines et fautives. | 110 |
| Fig. 5.17. | Limite de détection. | 110 |
| Fig. 5.18. | Détection des fautes sur les fréquences de coupures. | 113 |
| Fig. 5.19. | Détection des fautes sur les intermodulations. | 114 |
| Tab. 5.1. | Tableau de correspondance limite des descriptions saines (h) et nombre de circuits rejetés sur un million (PPM). | 96 |
| Tab. 5.2. | Extrait du tableau des spécifications de la partie réception du système W-CDMA. | 104 |
| Tab. 5.3. | Définition d'un stimulus de test utilisé. | 105 |
| Tab. 5.4. | Définition des plages de mesures à partir de l'échantillonnage pires-cas simplifié. | 106 |
| Tab. 5.5. | Définition des plages de mesures à partir de l'estimation gaussienne. | 106 |
| Tab. 5.6. | Définition des stimuli de test utilisés. | 109 |
| Tab. 5.7. | Plages de mesures des descriptions saines et fautives et limites de détection. | 110 |
| Tab. 5.8. | Résultats de détection de circuits fautifs. | 112 |
| Tab. 5.9. | Tableau des valeurs du quantile suivant la loi de Student. | 116 |
| Tab. 5.10. | Tableau des valeurs du quantile suivant la loi du Khi-deux. | 117 |

5.1. Introduction

Ce chapitre a pour objectif de présenter une méthode de qualification des stimuli utilisés pour valider la conception de systèmes AMS&RF. En fait, dans le flot de conception, la définition de stimuli permettant de valider le système est un point clé. C'est pourquoi, nous proposons de qualifier les stimuli définis dans le plan de vérification (§3.2.1.) afin de déterminer leur capacité à détecter d'éventuelles erreurs de conception. Dans notre approche, nous considérons que l'architecture a été définie et vérifiée au niveau fonctionnel grâce à l'utilisation de méthodes de vérification classiques relatives à ce niveau. Quelques-unes de ces méthodes basées sur les mesures de BER¹, EVM²... ont été présentées dans la partie 3.2.1. du chapitre III. Nous cherchons maintenant à vérifier que les paramètres de chaque bloc du système ont été correctement spécifiés lors de l'étape de budgétisation, c'est-à-dire lors de la définition des valeurs limites des paramètres des différents blocs. Pour cela, nous proposons d'utiliser une méthode largement répandue dans le domaine numérique : il s'agit de la simulation de fautes. Le principe consiste à créer des circuits fautifs en injectant des fautes dans la description originale du circuit et à déterminer la capacité de chaque stimulus à détecter ces fautes. Lors de l'analyse par simulation de fautes, il est nécessaire de définir un modèle de fautes pertinent car ce modèle aura un impact direct sur la qualité de l'évaluation des stimuli. Dans le domaine analogique, l'analyse par simulation de fautes demande aussi la définition de différentes descriptions saines servant de référence et représentant la variabilité normale du circuit ; la comparaison entre les réponses saines et fautives est alors possible et permet la détection de fautes.

Après la présentation de notre technique pour l'injection de fautes, nous aborderons le principe de fonctionnement de notre simulateur de fautes. Les notions nécessaires à l'injection de fautes : descriptions saines, modèle de fautes et métrique utilisée pour la qualification des stimuli seront alors détaillées. Finalement, la qualification des stimuli générés pour la validation de conception qui est proposée dans ce chapitre sera validée par la présentation de résultats de qualification obtenus pour un système AMS&RF : récepteur W-CDMA.

5.2. Intérêt de la qualification de la vérification de conception

Les enjeux et les méthodes appliquées pour la vérification de conception ont été abordés dans le chapitre III. La nécessité d'utiliser un flot de vérification rigoureux a été mis en évidence afin de garantir la qualité de conception de SoCs AMS&RF. Le flot de vérification est mis en oeuvre grâce à la définition d'un plan de vérification (§3.2.1.3.2.). Ce plan est défini par les ingénieurs de conception et de vérification, il permet d'identifier les caractéristiques du système qui doivent être vérifiées. La définition du plan de simulation est l'étape suivante (Fig. 5.1), elle a pour objectif de définir les bancs de test qui permettent de vérifier les caractéristiques identifiées lors de la définition du plan de vérification. Toutefois, lors de l'utilisation du flot de vérification, il n'existe pas d'étapes permettant de s'assurer que les caractéristiques à vérifier (définies par l'ingénieur de vérification) sont suffisantes et que le plan de simulation permet de vérifier exactement chacune de ces

¹ Bit Error Rate.

² Error Vector Magnitude.

caractéristiques. Les risques sont réels car ces étapes sont définies manuellement et font appel à l'appréciation des ingénieurs de vérification. Il est donc indispensable de mettre en place une méthode permettant de qualifier le plan de vérification et le plan de simulation.

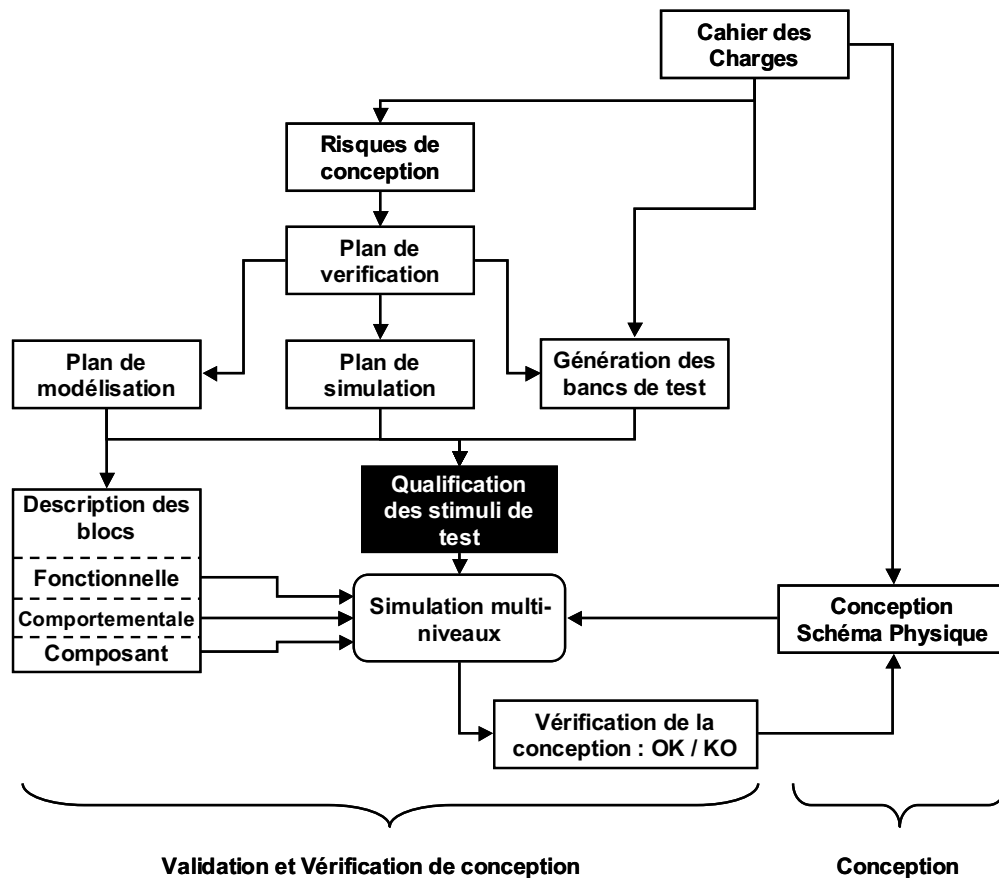


Fig. 5.1. Qualification de stimuli dans le plan de vérification.

La Fig. 5.1 reprend le principe du flot de vérification décrit précédemment (§3.2.1.3.2). Une étape a été ajoutée avant la simulation du système afin de qualifier les stimuli de test. Cette étape utilise le banc de test qui définit les composants externes ajoutés au circuit afin de permettre sa vérification, mais aussi les données issues du plan de simulation qui définissent ce qui doit être vérifié. Cette étape de qualification utilise aussi le plan de modélisation afin de déterminer le niveau de modélisation des blocs du système utilisés pour la qualification des stimuli de test. Ces trois parties sont étroitement liées et permettent de définir la simulation à exécuter. Pour réaliser la qualification, nous proposons d'utiliser une méthode basée sur la simulation de fautes. L'intérêt de cette méthode est de s'assurer que les bancs de test et le plan de simulation permettent de vérifier toutes les caractéristiques définies par le plan de vérification. Nous proposons alors de mettre en place, dans le flot de vérification, une étape d'analyse de banc de test ou plus précisément une étape d'analyse des stimuli utilisés lors de la vérification (Fig. 5.1). Ainsi, un modèle de fautes est proposé et permet d'évaluer la capacité de chaque stimulus à identifier ces fautes. La méthode utilisée sera développée dans les parties suivantes.

La qualification de la vérification de conception présente un intérêt particulier dans une approche de conception en vue du test (DfT¹). En effet, l'approche proposée permet de s'assurer que les stimuli

¹ Design for Testability.

définis par le plan de simulation ont permis de vérifier les caractéristiques définies par le plan de vérification. Si les résultats de la qualification révèlent que toutes les caractéristiques n'ont pas pu être détectées alors soit les stimuli sont insuffisants, soit le système n'est pas testable. Si un manque de testabilité est mis en évidence, cette étape de qualification pourra inciter le concepteur à ajouter des blocs DfT afin d'augmenter la testabilité du système (§3.4).

Le diagramme Fig. 5.2 représente les relations qui existent entre le flot de conception et le flot de vérification. Ainsi, à chaque étape de conception, un plan de vérification est mis en place. Il est donc indispensable d'évaluer et de qualifier le plan de vérification car celui-ci permettra à son tour de valider chaque étape de conception.

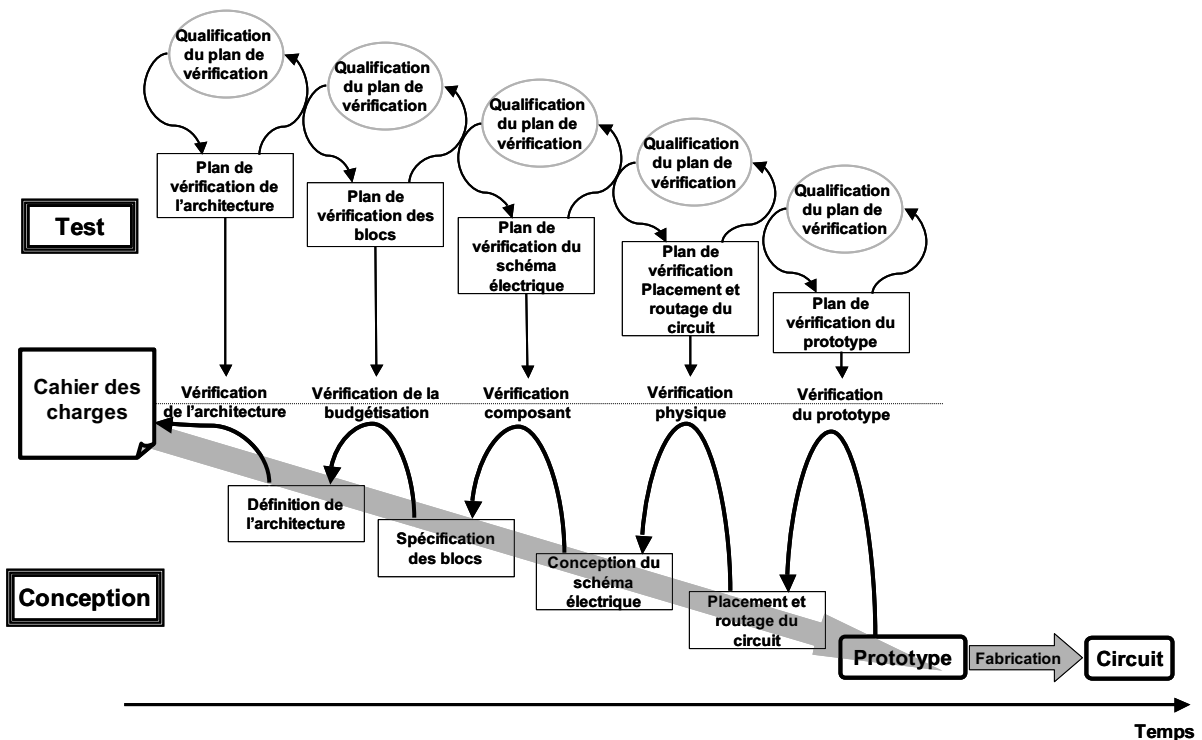


Fig. 5.2. Flot de conception et de flot de vérification.

Dans la suite de ce chapitre, nous proposons de mettre en place cette approche lors de la validation de la budgétisation c'est-à-dire lors de la définition des spécifications des blocs du système (Fig. 5.2). En effet, il existe déjà plusieurs techniques permettant de vérifier la définition de l'architecture (§2.4.1) et d'autres permettant de vérifier la description niveau composant c'est-à-dire la conception du schéma électrique (§2.4.2) mais peu de méthodes de vérification des spécifications des blocs du SoC. Ainsi, l'ajout d'un niveau d'abstraction comportemental dans le flot de conception Top-Down (§2.3.4) nécessite la définition de nouvelles étapes de vérification de conception et plus particulièrement de vérification de la budgétisation des spécifications des blocs.

5.3. Qualification des stimuli de vérification par injection et simulation de fautes

La méthode de qualification des stimuli de test que nous proposons est basée sur l'injection et la simulation de fautes ; son principe consiste à comparer les résultats de simulation des *descriptions fautes* avec ceux des *descriptions saines* (Fig. 5.3). Pour cela, des descriptions fautes sont générées en injectant des fautes dans la description originale du système, la définition du modèle de

fautes utilisé sera donnée dans la partie 5.3.3. L'objectif est alors de détecter toutes les descriptions fautives. La qualification des stimuli se fait en identifiant leur capacité à stimuler et à rendre observable les effets des paramètres budgétisés.

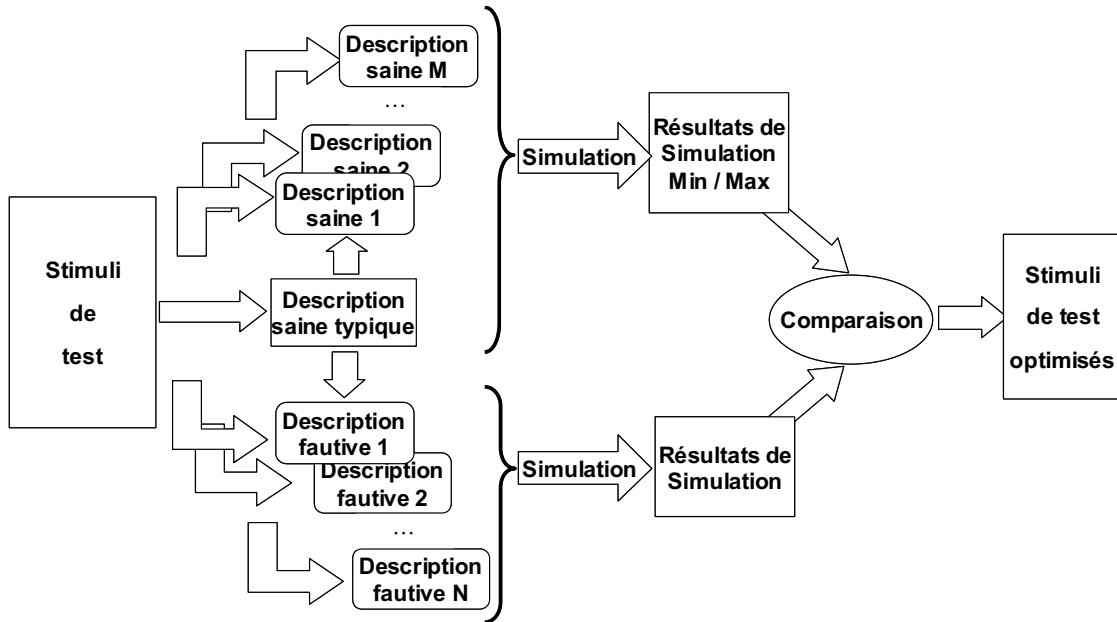


Fig. 5.3. Injection et simulation de fautes.

Pour chaque stimulus, plusieurs descriptions saines sont simulées (Fig. 5.3). Des plages des descriptions saines sont ainsi déterminées pour chaque mesure à partir des résultats de simulation minimum et maximum (Fig. 5.4). La réponse de chaque description fautive est ensuite comparée avec la plage précédemment déterminée. Si la réponse de la description fautive est à l'extérieur, cela signifie que le stimulus évalué permet d'activer et de propager l'effet de cette faute et par conséquent qu'il est capable de détecter au moins cette faute. Par exemple, la Fig. 5.4 met en évidence la détection de la faute injectée dans la description fautive n°2 mais pas de celle injectée dans la description fautive n°1.

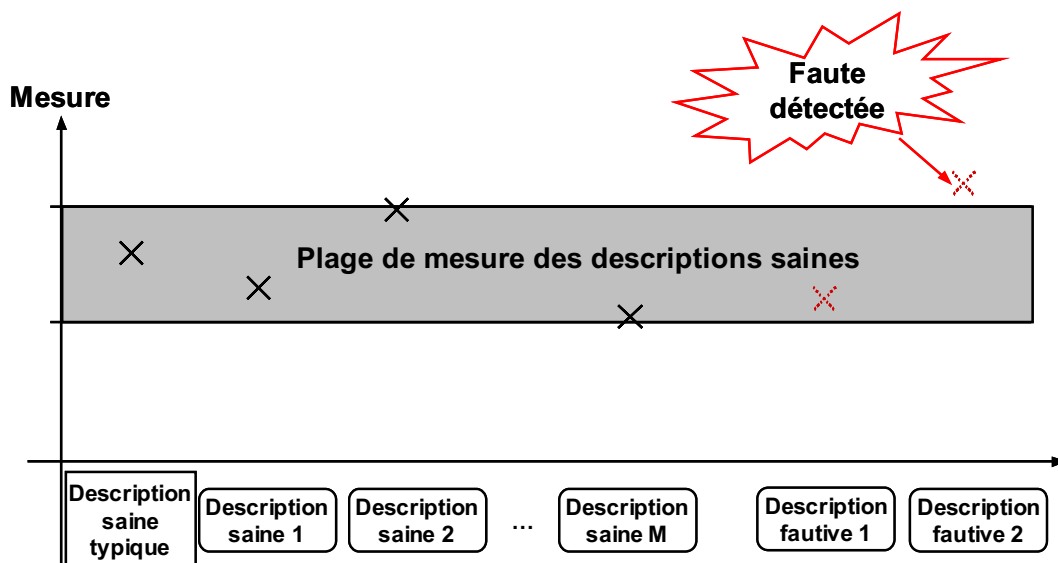


Fig. 5.4. Définition des plages de mesures pour la détection de fautes.

Dans l'approche développée, nous cherchons à vérifier l'étape de budgétisation des spécifications

de chaque bloc contenu dans l'architecture du système (Fig. 5.2). Pour cela, il est nécessaire de comparer le cahier des charges initial du système vis-à-vis de l'architecture définie. Cette comparaison n'est pas immédiate. Il est tout d'abord nécessaire d'établir un modèle fonctionnel exécutable du système (Oracle). Puis, les résultats de simulation de cet oracle sont comparés aux résultats de simulation de l'architecture comme illustré sur la Fig. 5.5.

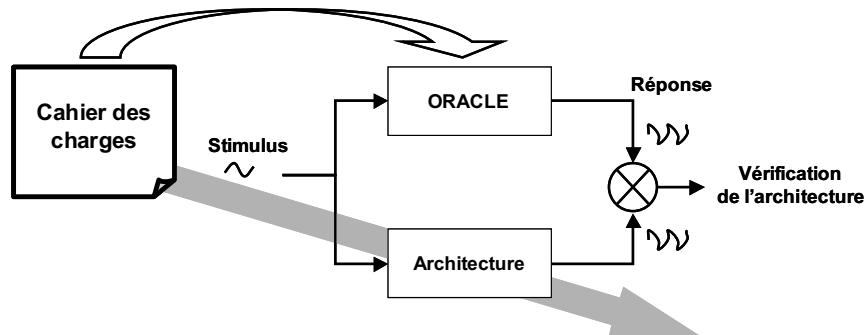


Fig. 5.5. Vérification de conception de l'architecture.

Il faut donc déterminer les stimuli qui vont permettre de comparer les réponses de nos descriptions saines avec celles de l'oracle. Pour cela, nous proposons de qualifier les stimuli avec les descriptions saines comme référence et d'identifier tout stimulus permettant de détecter des variations de spécifications ; cette notion liée au modèle de fautes est décrite dans le paragraphe 5.3.3. Il faudra ensuite recalculer les limites de détection sur les mesures en utilisant le cahier des charges du système ou l'oracle afin d'obtenir les limites correspondant exactement au cahier des charges.

Généralement, les stimuli sont déterminés à partir du plan de vérification, plan de vérification lui-même construit à partir des spécifications contenues dans le cahier des charges. Ainsi, pour chaque stimulus, le concepteur connaît la valeur de la mesure. Par exemple, le plan de vérification précise la valeur du gain à vérifier pour une fréquence donnée. Ceci implique que des stimuli « alternatifs », c'est-à-dire n'ayant pas de relations avec le plan de vérification et donc pas de relations avec la description de l'oracle ne peuvent pas être appliqués pour la validation du système. Par exemple, la mesure du gain à une fréquence extérieure au domaine des fréquences fixé par le plan de vérification ne permet pas de vérifier l'architecture du système.

5.3.1. PLASMA : outil d'aide à la génération de stimuli de test

Lors de la qualification d'un stimulus, il faut exécuter autant de simulations qu'il y a de descriptions saines et de descriptions fautives. Un outil d'aide à l'évaluation des stimuli de test a ainsi été développé afin d'automatiser cette étape. Il permet ainsi de générer des fichiers nécessaires à la description du banc de test, à l'exécution des simulations et à l'analyse de leurs résultats.

L'outil PLASMA est défini comme une **PLA**te-forme de génération de stimuli de test pour les **S**ystèmes **M**ixtes **A**nalogiques et RF, son principe de fonctionnement est décrit par la Fig. 5.6. Il permet d'évaluer le plan de simulation en qualifiant les stimuli de test utilisés dans le flot de vérification et il permet de plus de générer des stimuli pour le test de production. Cette dernière fonctionnalité sera exposée dans le chapitre suivant. PLASMA a été développé afin de permettre l'exécution automatique de simulations AMS&RF et l'analyse des résultats ainsi que la génération du stimuli de test. Cet outil est associé au logiciel de simulation ADVance MS RF de MENTOR

GRAPHICS. PLASMA met en place automatiquement les simulations en générant au préalable les données nécessaires à la simulation (fichier de simulation) à partir de la description du système (VHDL-AMS), des types d'analyses utilisées pour la simulation et de la liste de stimuli à qualifier (Fig. 5.6). Une fois que toutes les simulations des descriptions saines et fautives ont été effectuées, les résultats sont analysés afin de qualifier le stimulus. Ce cycle est ré-itéré autant de fois qu'il y a de stimuli à évaluer.

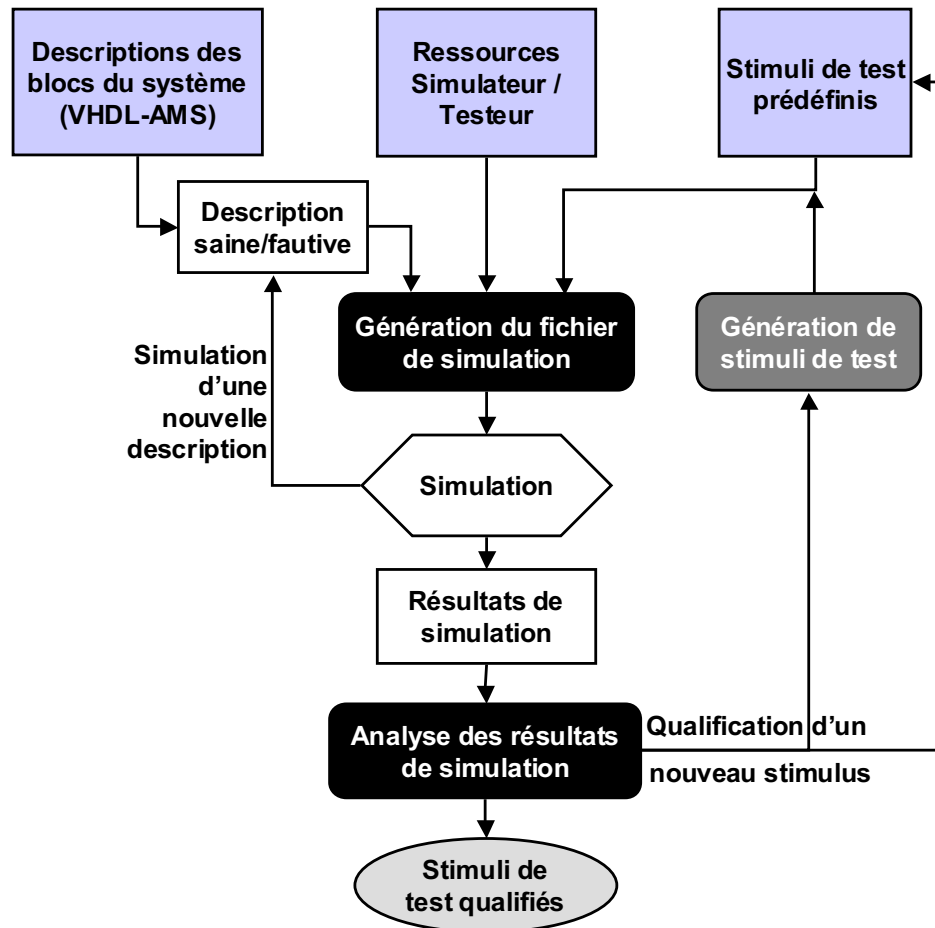


Fig. 5.6. Principe de fonctionnement de PLASMA.

5.3.1.1. Simulation automatique

L'évaluation des stimuli de test est réalisée grâce à l'exécution de simulations. Pour cela, l'outil PLASMA met en place des bancs de test permettant d'effectuer des mesures virtuelles sur les descriptions saines et les descriptions fautives (Fig. 5.6). Ces bancs de test permettront alors de déterminer la capacité des stimuli à détecter des fautes. Pour permettre la simulation, PLASMA décrit le banc de test dans le fichier de simulation (Fig. 5.7). La première partie de cette description fait appel au modèle original du système, elle permet de nommer les différents terminaux et de définir la valeur des paramètres génériques du système. La seconde partie définit les signaux d'entrées ou autrement dit le stimulus appliqué sur la description. Le banc de test est ensuite spécifié, il permet d'ajouter des composants externes au circuit sous test tels que des résistances de charges, tension de référence (GND, VCC)... Les dernières parties définissent le type d'analyse qui doit être appliqué (§2.4.2.) et les mesures réalisées sur les différents terminaux.

```

*****
* Implémentation de la description à simuler
*****
.model top(bhv) macro lang=vhdlams lib=Hekla_lib

Y_MT top(bhv) port : T1_MT, T2_MT, T3_MT, T4_MT, T5_MT,
T6_MT, T7_MT
+ param :
+ PC1 = 1.500000e+01
...
+ PL1 = 3.750000e+02
+ PL2 = 3.750000e+02
...
*****
* Définition des signaux d'entrées
*****
.param fund_1=2.141400e+09
.param fund_2=2.140000e+09

V_S1_MT T1_MT 0 rport=5.000000e+01 iport=1 FOUR fund_1 MA
(1) 2.236067e-05 0.000000e+00
V_S2_MT T2_MT T3_MT FOUR fund_2 MA (1) 1.000000e-01
0.000000e+00
*****
* Définition du banc de test
*****
V_gnd_MT_1 T3_MT 0 0.0
R_MT_4 T4_MT T5_MT 1.000000e+06
R_MT_6 T6_MT T7_MT 1.000000e+06
*****
* Définition du type d'analyse
*****
.sst fund1=fund_1 nharm1=5
+ fund2=fund_2 nharm2=5
*****
* Définition des mesures
*****
.printfile fsst VM (T4_MT, T5_MT) file="MT.data"
.printfile fsst VM (T6_MT, T7_MT) file="MT.data"

```

Fig. 5.7. Fichier de simulation.

L'implémentation des descriptions saines et fautives dans la description du banc de test se fait à partir de la même description comportementale VHDL-AMS originale du circuit. La méthode utilisée pour définir les descriptions saines et fautives est basée sur « l'utilisation des commandes du simulateur » (§3.3.3.2.1.). Grâce à l'utilisation de cette méthode, il est possible de modéliser des descriptions saines et fautives en faisant uniquement varier les valeurs des paramètres comportementaux modélisés dans la description originale et définis comme paramètres génériques. Ainsi, suivant qu'il s'agit d'une description saine ou d'une description fautive, la valeur du paramètre est située à l'intérieur ou à l'extérieur des spécifications. Cette méthode d'injection de fautes basée sur les commandes du simulateur permet d'éviter de recompiler la description VHDL-AMS à chaque implémentation. Pour pouvoir utiliser les commandes du simulateur pour l'injection de fautes, les paramètres comportementaux du système doivent être définis comme paramètres génériques dans la description VHDL-AMS.

Les sources permettant de définir les signaux d'entrées sont choisies en fonction des ressources disponibles (Fig. 5.6). L'utilisateur peut ainsi spécifier le type de source et donc indirectement le type d'analyse qu'il pourra utiliser lors de la simulation ou les appareils de mesures qu'il pourra utiliser lors de la validation du prototype. Des classes de sources ont été définies, cela signifie que les sources sont classées en fonction de leur disponibilité et de leur coût. Ainsi, l'utilisation de sources sinusoïdales étant moins coûteuse en temps de simulation par rapport à l'utilisation d'un générateur de signal haute-fréquence modulé, cette classe est donc prioritaire lors de la génération de stimuli. Cette fonctionnalité permet de considérer le temps de qualification des stimuli mais elle a été mise en place dans l'objectif principal de diminuer le coût du test de production. Son intérêt sera mis en évidence dans le chapitre suivant.

Pour terminer, le fichier de simulation définit le type d'analyse qui doit être réalisé lors de la

simulation. Ce paramètre est directement lié aux types de sources utilisées. En effet, lors de l'utilisation de sources sinusoïdales, une analyse en régime établi est suffisante. Par contre lors de l'utilisation de sources modulées, il faut mettre en place une analyse par détection d'enveloppe plus coûteuse en temps de simulation. Ces différents types d'analyse ont été présentés dans le chapitre II.

5.3.1.2. *Traitement des résultats de simulation*

Une fois l'ensemble des simulations réalisées, nous obtenons autant de fichiers de résultats de simulation qu'il y a de descriptions simulées. Chacun de ces fichiers est composé des résultats de simulation obtenus sur les différents terminaux du circuit. Par exemple, lors d'une analyse en régime établi, nous obtenons la tension pour les différentes fréquences analysées lors de la simulation (ex : Mesures des tensions entre les terminaux T4_MT et T5_MT Fig. 5.7). L'outil PLASMA permet de déterminer les plages de réponse des descriptions saines utilisées pour la comparaison des réponses des descriptions fautives comme représenté dans la Fig. 5.4. Si celles-ci sont situées à l'extérieur de la plage des descriptions saines, cela signifie que le stimulus utilisé permet de détecter la faute injectée. Il est alors possible de qualifier le stimulus évalué (Fig. 5.6) en appliquant une métrique qui sera définie dans la partie 5.3.4.

5.3.1.3. *Optimisation des temps de simulation*

Lors de la qualification de stimuli de test, PLASMA doit permettre de qualifier un nombre très important de stimuli dans des temps réalistes (≈ 23.000 descriptions simulées pour une centaine de stimuli qualifiés), son architecture a donc été développée de façon à réduire les temps d'évaluation des stimuli.

La première technique permettant de diminuer les temps d'évaluation consiste à injecter des fautes en utilisant les commandes du simulateur afin de supprimer l'étape de compilation des descriptions coûteuses en temps. Cette approche a été présentée précédemment dans 5.3.1.1.

La seconde technique consiste à générer des simulations parallèles. Pour cela plusieurs descriptions sont implémentées dans un même fichier de simulation. Comme l'étape de chargement des descriptions dans l'outil de simulation (ADVanced MS RF) est coûteuse en temps d'exécution, il est ainsi possible de diminuer le temps final d'évaluation des stimuli.

Les principales fonctionnalités de l'outil PLASMA pour l'évaluation des stimuli de test ont été définies. Cet outil pilote le simulateur de Mentor Graphics ADMS RF, ainsi il bénéficie des propriétés du simulateur telle que la possibilité d'utiliser un système décrit à l'aide de plusieurs langages de description (SPICE, VHDL, VHDL-AMS). Pour cette raison le système peut ainsi être décrit dans PLASMA à différents niveaux d'abstractions (niveau circuit, comportemental, et fonctionnel). Ainsi, comme ADMS RF, PLASMA peut traiter des systèmes multi-domaines et multi-disciplinaires (analogique, numérique, RF, micromécaniques ... (§2.5.2.)).

Après avoir présenté le concept général de la simulation de fautes et les principales fonctionnalités de PLASMA, nous allons aborder la définition des différentes descriptions utilisées et de la métrique nécessaire à l'évaluation des stimuli.

5.3.2. *Descriptions saines*

Dans la partie précédente (Fig. 5.3 et Fig. 5.4), nous avons vu que les descriptions saines sont

indispensables à la détection de fautes et donc à l'évaluation des stimuli, nous allons maintenant préciser comment ces descriptions sont générées. Une description saine est définie comme une description dont tous les paramètres sont situés à l'intérieur de leur plage de tolérance. Ces plages sont spécifiées au cours de la conception du système et plus particulièrement lors de la définition des blocs.

Lors de la définition des plages de mesures des descriptions saines, il est indispensable de procéder à un échantillonnage. En effet, même si les spécifications sont bornées, il s'agit pour les paramètres analogiques de valeurs continues ; il existe donc une infinité de descriptions saines. De plus, la complexité croissante des systèmes fait que les descriptions incluent de plus en plus de paramètres. Une stratégie d'échantillonnage est donc indispensable pour permettre l'évaluation de stimuli de test de qualité dans des temps de simulation raisonnables. Dans les parties suivantes, nous allons présenter deux méthodes permettant de déterminer un ensemble représentatif de la population saine.

5.3.2.1. Echantillonnage pires-cas « simplifié »

La première méthode permettant de définir un échantillon de la population des descriptions saines consiste à utiliser les valeurs pires-cas, pour cela il faut combiner les valeurs minimales et maximales de chaque paramètre (§3.3.3.2.3.). Dans le cas de la partie réception du système W-CDMA, cela reviendrait à générer $2^{23} \approx 8$ millions de descriptions saines ; évidemment cette méthode ne peut pas être utilisée pour ce cas d'étude car elle demanderait des temps de simulations trop importants. L'utilisation de cette méthode basée sur les valeurs limites des paramètres nécessite de faire l'hypothèse que le système est monotone, cela signifie que les relations entre les paramètres et les mesures sont strictement croissantes (resp. décroissantes) et donc que la mesure minimale/maximale est obtenue quand le paramètre est à une de ses valeurs limites.

Une variante de cette méthode permet de diminuer le nombre de descriptions saines à simuler au détriment d'une perte de précision sur la définition des plages de mesures des circuits sains. Cette méthode consiste à placer tous les paramètres à leur valeur typique sauf un qui est défini à une de ses valeurs limites. Cette simplification implique que l'on suppose qu'un paramètre va avoir une influence plus importante sur la variation d'une mesure que les autres paramètres. Le positionnement de chaque paramètre à sa valeur limite va ainsi permettre d'obtenir une plage de mesure d'un échantillon de la population qui est forcément incluse dans la plage de mesure de la population saine totale (Fig. 5.8). Dans le cas du récepteur W-CDMA (Chapitre IV), l'utilisation de cette approche permet de diminuer le nombre de descriptions saines à simuler à $2 \times 23 = 46$ descriptions.

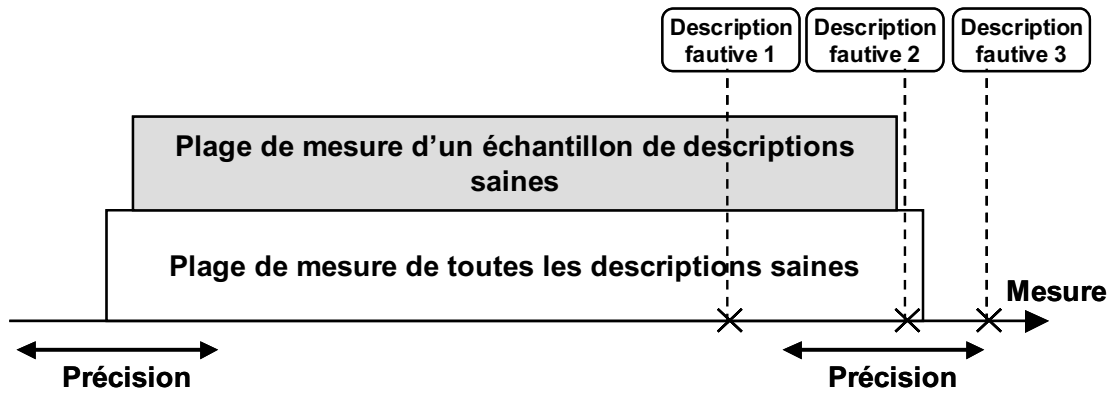


Fig. 5.8. Plages de mesure des descriptions saines.

Nous allons maintenant justifier cette approche et les hypothèses faites en montrant que la précision sur la définition des plages de mesures des descriptions saines n'est pas indispensable pour pouvoir qualifier les stimuli de tests. En effet, si la différence entre les plages de mesure de la population échantillonnée et totale est très faible devant la précision des modèles, des simulations, des appareils de mesures... ; il n'est pas nécessaire de simuler un nombre de descriptions saines plus important (Fig. 5.8). Comme il est très difficile de quantifier la précision des plages de mesures obtenues, nous présenterons dans 5.4.3.1 des résultats de simulations et de définition des plages de mesures obtenues pour un cas d'étude donné.

Lors de l'évaluation de stimuli de test, nous cherchons à identifier ceux qui sont inutiles, c'est-à-dire qui ne permettent pas de détecter des fautes, afin de pouvoir les supprimer lors de la réduction des stimuli de test. Pour un stimulus, toute description fautive non-détectée à partir de la plage d'un échantillon de la population saine (un paramètre au pires-cas) ne sera jamais détectée par ce stimulus car cette plage est forcément inférieure à la plage de la population saine totale (ex : Description fautive 1. Fig. 5.8). Par contre dans le cas où la description fautive est détectée, le problème de précision issu de l'échantillonnage peut devenir pénalisant. En effet, les descriptions fautives 2 et 3 de la Fig. 5.8 sont détectées lorsque l'on utilise un échantillon de la population saine, mais en réalité seule la description fautive 3 pourrait être détectée si on considérait la population totale de circuits sains.

Pour faire abstraction des problèmes de précision évoqués ci-dessus, nous choisissons de raisonner en relatif c'est-à-dire en comparant un stimulus par rapport à un autre. En effet, l'utilisation d'une méthode permettant de définir exactement les fautes que l'on est capable de détecter pour un ensemble de stimuli n'est pas réaliste car elle nécessite la simulation d'un nombre trop important de descriptions (8 millions contre 47 dans notre cas d'étude). Ainsi, lors de l'évaluation, nous ne cherchons pas à obtenir une précision absolue sur les métriques mais nous comparons la qualité de différents stimuli entre eux. En utilisant le même échantillon de descriptions saines pour l'évaluation de plusieurs stimuli, la qualification des stimuli permet de faire abstraction de la précision absolue qui est très difficilement quantifiable lors de l'utilisation de méthodes d'échantillonnage comme celle proposée.

Ainsi, en utilisant un mode d'échantillonnage compatible avec des temps de simulation raisonnables, il n'est pas possible de quantifier avec une précision absolue des grandeurs telles que

les pertes de rendement, le taux de défauts... mais seulement de dire si un stimulus est meilleur qu'un autre vis-à-vis ces grandeurs.

5.3.2.2. Echantillonnage par estimation gaussienne

5.3.2.2.1. Définition des descriptions saines simulées

La méthode présentée dans cette partie permet de diminuer le nombre de descriptions saines en tenant compte des pertes de précision (dues à l'échantillonnage) sur la définition des plages de mesures des circuits sains. Cette approche est basée sur l'analyse de Monte Carlo. Lorsqu'il est difficile de connaître les plages de mesures des descriptions saines ou autrement dit les réponses pour toutes les combinaisons de paramètres, l'analyse de Monte Carlo permet de définir ces plages en procédant par tirages aléatoires des paramètres du circuit. Dans notre cas, les descriptions saines sont générées en tirant aléatoirement, mais dans les plages de spécifications, les valeurs des paramètres présents dans la description (Paramètres génériques). Le tirage utilisé est un tirage gaussien dont les caractéristiques de distribution des paramètres (moyenne et écart-type) sont déterminées à partir des spécifications définies lors de la conception du système. Ces caractéristiques ont été abordées dans le chapitre II partie 2.3.2.2. et sont représentées dans la figure (Fig. 5.9). En effet, lors de la spécification de la valeur typique et des valeurs limites des différents paramètres, le concepteur tient compte des variations qui peuvent apparaître lors de la fabrication du circuit, il fait ainsi apparaître la notion de robustesse par la définition du « Capability Process » (Eq. 5.1) et partie 2.3.2.2.). Cette caractéristique permet d'estimer la distribution que suit chaque paramètre lors du processus de fabrication (Fig. 5.9). Pour cela, il suppose que la distribution des paramètres suit une loi gaussienne et définit sa moyenne (valeur typique), ses valeurs limites et son CPk. L'équation (Eq. 5.1) relie les différentes caractéristiques de chaque paramètre et permet de déterminer l'écart type.

$$\sigma = \frac{\text{MIN}(P_{\max} - P_{\text{typ}}, P_{\text{typ}} - P_{\min})}{3.CPk} \quad (\text{Eq. 5.1})$$

Ces caractéristiques (typ, min, max, CPk) sont donc reprises pour le tirage aléatoire de la population saine. Ainsi, l'analyse de Monte Carlo permet d'obtenir une représentation réaliste de la population saine, mais pour cela un nombre de tirages important est nécessaire. En effet, pour l'analyse de systèmes comprenant un grand nombre de paramètres, il faut un nombre de tirage supérieur au million si les précisions des métriques utilisées sont de l'ordre des Parties Par Million (ppm). Par exemple, dans [5.1], l'auteur propose une approche permettant de définir les plages de mesures des circuits sains en spécifiant un nombre de circuits sains que l'on s'autorise à rejeter sur un million de circuits testés. Pour cela, il simule 1000 circuits et augmente virtuellement le nombre de circuits par l'utilisation d'un modèle mathématique. Dans notre cas, nous ne cherchons pas à définir des métriques absolues, mais nous avons toutefois besoin d'avoir des plages de mesures saines proches de la réalité. Pour cela, nous monterons par expérience et pour un circuit précis que le nombre minimum de descriptions saines non rejetées sur un million peut être calculé en vérifiant que les mesures obtenues suivent une loi gaussienne.

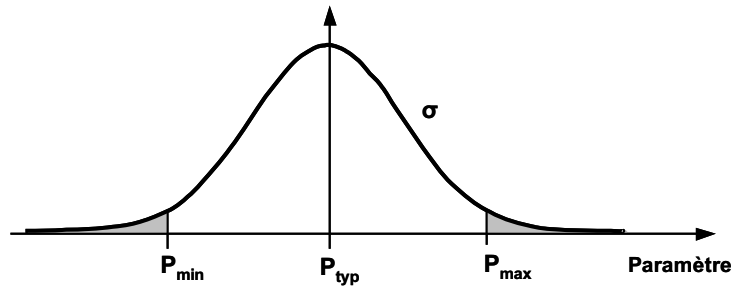


Fig. 5.9. Distribution gaussienne des paramètres du circuit.

5.3.2.2.2. Définition des plages de mesures des circuits sains

Pour la définition des plages de mesures des circuits sains, nous proposons de définir des limites à partir de la moyenne et de l'écart type de la population saine (Fig. 5.10). Pour cela, nous supposons que les mesures de la population saine suivent une distribution gaussienne.

Les limites sur les mesures sont définies par les équations suivantes :

$$X_{\min} = m - h \cdot \sigma \quad (\text{Eq. 5.2})$$

$$X_{\max} = m + h \cdot \sigma \quad (\text{Eq. 5.3})$$

Les valeurs de m et σ sont respectivement la moyenne et l'écart-type de la population totale des circuits sains. La valeur de h permet de positionner la limite des mesures saines en fonction de l'écart type et du nombre de circuits sains que l'on s'autorise à rejeter sur un million (Surface grise Fig. 5.10). Tab. 5.1 permet de lier la valeur de h au nombre de circuits sains que l'on accepte de rejeter.

Les valeurs de m et σ correspondent aux caractéristiques de la population saine totale. Comme les contraintes de temps de simulation ne nous permettent pas de simuler toutes les descriptions saines, nous proposons donc d'estimer les plages de mesures de la population saine en utilisant un échantillon de cette population. Pour cela nous estimons tout d'abord la moyenne et l'écart type de l'ensemble de la population saine en calculant la moyenne et l'écart type empirique de l'échantillon et en ajoutant un intervalle de confiance puis nous définissons les valeurs des plages de mesures des circuits sains à partir des équations (Eq. 5.2) et (Eq. 5.3) ou m et σ sont remplacés par les valeurs estimées.

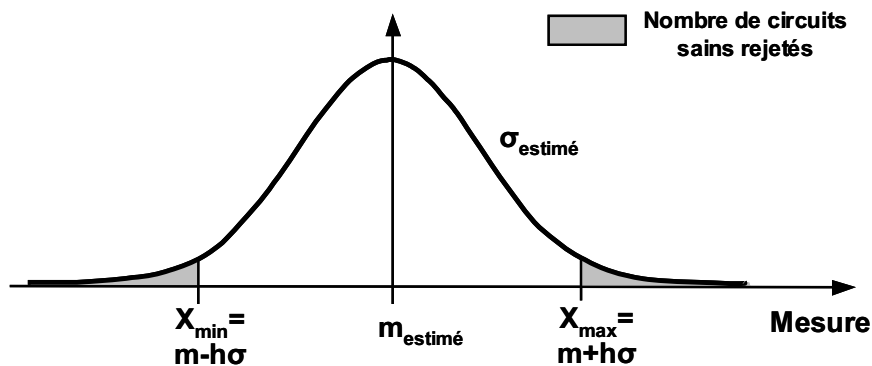


Fig. 5.10. Définition des plages de mesures de modèles sains.

| Limite des modèles sains h | Circuits concernés % | Circuits rejetés (PPM) |
|-------------------------------|-------------------------|---------------------------|
| 1 | 0,84 | 158655 |
| 1,2 | 0,884930268 | 115070 |
| 1,4 | 0,919243289 | 80757 |
| 1,6 | 0,945200711 | 54799 |
| 1,8 | 0,964069734 | 35930 |
| 2 | 0,98 | 22750 |
| 2,2 | 0,986096601 | 13903 |
| 2,4 | 0,991802471 | 8198 |
| 2,6 | 0,995338778 | 4661 |
| 2,8 | 0,997444809 | 2555 |
| 3 | 0,9987 | 1350 |
| 3,2 | 0,999312798 | 687 |
| 3,4 | 0,999663019 | 337 |
| 3,6 | 0,999840854 | 159 |
| 3,8 | 0,999927628 | 72 |
| 4 | 0,999968 | 32 |
| 4,2 | 0,999986646 | 13,35409733 |
| 4,4 | 0,999994583 | 5,416953054 |
| 4,6 | 0,999997885 | 2,114643376 |
| 4,8 | 0,999999206 | 0,794352669 |
| 5 | 0,99999971 | 0,29 |
| 5,2 | 0,9999999 | 0,0998344 |
| 5,4 | 0,99999967 | 0,033396123 |
| 5,6 | 0,99999989 | 0,010746217 |
| 5,8 | 0,99999997 | 0,003326052 |
| 6 | 0,99999999 | 0,00099 |

Tab. 5.1. Tableau de correspondance limite des descriptions saines (h) et nombre de circuits rejetés sur un million(PPM).

D'après la loi des grands nombres, la moyenne estimée converge vers la moyenne réelle, il s'agit donc d'une estimation sans biais. Il est néanmoins nécessaire d'utiliser des intervalles de confiance afin de mesurer l'imprécision de l'estimation des paramètres de la loi normale et de considérer l'erreur due à la diminution du nombre d'échantillons. On définit X_1, \dots, X_n comme étant les résultats de simulations obtenus pour n descriptions saines. Si X_1, \dots, X_n sont indépendantes et de loi $N(m, \sigma^2)$, on sait que l'estimateur sans biais de la moyenne m est $m_{\text{estimé}}$. L'intervalle de confiance I pour le paramètre m de la loi normale $N(m, \sigma^2)$ est défini par (Eq. 5.4) ([5.3]). Le paramètre α représente le seuil de confiance, c'est-à-dire la probabilité que la moyenne de la population complète se situe à l'intérieur de l'intervalle de confiance I.

$$I = \left[m_{\text{estimé}} - \frac{\sigma}{\sqrt{n}} u_{\alpha}, m_{\text{estimé}} + \frac{\sigma}{\sqrt{n}} u_{\alpha} \right], \quad (\text{Eq. 5.4})$$

ou u_{α} est tel que $P[|U| > u_{\alpha}] = \alpha$ lorsque U suit la loi $N(0,1)$.

$$\text{Avec } m_{\text{estimé}} = \frac{1}{n} \cdot \sum_{i=1}^n x_i, \quad (\text{Eq. 5.5})$$

ou n est le nombre d'échantillons et X la mesure obtenue.

Il faut maintenant définir la valeur U_α pour définir l'intervalle I . Dans notre cas, l'écart type σ de la population totale n'est pas connue, il est donc nécessaire de l'estimer car il intervient lors de la définition de l'intervalle de confiance I sur la moyenne. Pour cela, $\sigma_{\text{estimé}}$ est défini comme étant l'écart type calculé à partir de l'échantillon :

$$\sigma_{\text{estimé}}^2 = \frac{1}{n-1} \cdot \sum_{i=1}^n (x_i - m_{\text{estimé}})^2, \quad (\text{Eq. 5.6})$$

ou n est le nombre d'échantillons et X la mesure obtenue.

Comme notre variable aléatoire X suit une loi normale, la quantité $n \cdot \frac{\sigma^2}{\sigma_{\text{estimé}}^2}$ suit une loi du χ^2 à $n - 1$ degrés de liberté. En posant une nouvelle variable Y définie par $Y = \frac{(m_{\text{estimé}} - m)\sqrt{n}}{\sigma_{\text{estimé}}}$, celle-ci suit alors une loi de Student à $n - 1$ degrés de liberté [5.4].

D'après la loi de Student, l'intervalle de confiance sur la moyenne estimée devient alors :

$$I = \left[m_{\text{estimé}} - \frac{\sigma_{\text{estimé}}}{\sqrt{n}} t_{\alpha}^{n-1}, m_{\text{estimé}} + \frac{\sigma_{\text{estimé}}}{\sqrt{n}} t_{\alpha}^{n-1} \right], \quad (\text{Eq. 5.7})$$

ou α est le seuil de l'intervalle de confiance et t_{n-1} correspondant au risque α qui est lu dans la table de Student pour $n - 1$ degrés de liberté (Tab. 5.9 en annexes).

L'intervalle I représente l'intervalle de confiance sur la moyenne estimée à partir de l'échantillonnage. Comme nous souhaitons définir des plages de mesures des circuits sains en fonction de l'écart-type de la population échantillonnée ((Eq. 5.2) et (Eq. 5.3)), il est nécessaire de définir un intervalle de confiance sur $\sigma_{\text{estimé}}$. La définition de l'intervalle de confiance J pour l'écart type se fait en considérant la loi du χ^2 à $n - 1$ degrés de liberté [5.4].

$$J = \left[\frac{n\sigma_{\text{estimé}}^2}{\kappa_{1-\frac{\alpha}{2}}^2(n-1)}, \frac{n\sigma_{\text{estimé}}^2}{\kappa_{\frac{\alpha}{2}}^2(n-1)} \right], \quad (\text{Eq. 5.8})$$

ou $\kappa_{1-\frac{\alpha}{2}}^2(n-1)$ et $\kappa_{\frac{\alpha}{2}}^2(n-1)$ sont les quantiles d'ordres $1 - \frac{\alpha}{2}$ et $\frac{\alpha}{2}$ de loi du χ^2 à $n - 1$ degrés de liberté [5.4].

On obtient :

$$P[\kappa^2 < k] = P\left[\frac{\sigma_{\text{estimé}}^2}{\sigma^2} < k\right] = \alpha, \quad (\text{Eq. 5.9})$$

ou k est lu dans la table du χ^2 à $n - 1$ degrés de liberté (Tab. 5.10 en annexes) [5.4].

Au final, il est possible d'estimer des valeurs minimales et maximales à partir de l'estimation gaussienne d'un échantillon de la population saine en faisant apparaître les intervalles de confiance I (Eq. 5.7) et J (Eq. 5.8) dans (Eq. 5.2) et (Eq. 5.3).

$$X_{\min} = m_{\text{estimé}} - \left(\frac{\sigma_{\text{estimé}}}{\sqrt{n}} t_{\alpha}^{n-1} + h \sqrt{\frac{n}{\kappa_{1-\frac{\alpha}{2}}^2 (n-1)}} \sigma_{\text{estimé}} \right), \quad (\text{Eq. 5.10})$$

ou n est le nombre d'échantillons, $m_{\text{estimé}}$ est définie par (Eq. 5.7) et $\sigma_{\text{estimé}}$ par (Eq. 5.8) et h est relatif au nombre de circuits sains rejetés.

$$X_{\max} = m_{\text{estimé}} + \left(\frac{\sigma_{\text{estimé}}}{\sqrt{n}} t_{\alpha}^{n-1} + h \sqrt{\frac{n}{\kappa_{1-\frac{\alpha}{2}}^2 (n-1)}} \sigma_{\text{estimé}} \right), \quad (\text{Eq. 5.11})$$

ou n est le nombre d'échantillons, $m_{\text{estimé}}$ est définie par (Eq. 5.7) et $\sigma_{\text{estimé}}$ par (Eq. 5.8) et h est relatif au nombre de circuits sains rejetés.

Ainsi, PLASMA calcul la moyenne estimée (Eq. 5.5) et l'écart type estimé (Eq. 5.6) pour chaque mesure à partir des résultats de simulation de l'échantillon de la population saine simulée. Lors de l'estimation, on s'autorise un seuil de confiance $\alpha=5\%$, l'outil détermine alors la valeur de $t_{\alpha; n-1}$ à partir de Tab. 5.9 (en annexes) ou n est le nombre de descriptions saines simulées. La valeur de κ^2 se détermine de la même façon pour $\alpha=5\%$ et à partir de Tab. 5.10 (en annexes). D'après (Eq. 5.10) et (Eq. 5.11), il est alors possible de définir les valeurs minimum et maximum estimées pour une mesure à partir de n descriptions saines et donc de définir une plage de mesure nécessaire à la détection des descriptions fautives.

La difficulté est maintenant de déterminer le nombre minimal de descriptions saines nécessaire pour déterminer les plages de mesures de la population saine. La définition de ce nombre est difficile car elle va dépendre du stimulus utilisé, de la mesure et du circuit.

5.3.2.2.3. Définition du nombre de modèles sains par étude de la variation des plages de mesures

Une première approche consiste à définir le nombre de descriptions saines en considérant la variation des plages de mesures obtenues. Lorsqu'une plage de mesure ne varie plus c'est-à-dire qu'elle est sensiblement identique entre un nombre de descriptions saines n et n', cela signifie que le nombre de descriptions saines est suffisant. Cette méthode est valable si on suppose que la plage de mesure tend asymptotiquement vers la plage de mesure réelle lorsque le nombre de descriptions saines augmente. Il est alors possible de mettre en place une méthode dynamique qui permet de déterminer le nombre de descriptions minimales. Pour cela, il faut commencer par simuler une seule description saine puis augmenter le nombre de descriptions tant que les plages de mesures saines estimées varient. Cette approche dynamique est difficile à mettre en place car l'évolution de la plage estimée dépend du stimulus et de la mesure ; ainsi le nombre minimal de descriptions saines peut être différent pour différents stimuli et pour différentes mesures. De plus, lors de l'utilisation de cette approche dynamique, il faut définir une relation entre n et n'. La relation la plus simple consiste à dire que $n'=n+1$. Cependant, pour une valeur de n importante, la variation de la plage de mesure entre n et n' peut être faible ce qui va conduire à l'arrêt des simulations même si la plage déterminée est très éloignée de la plage de la population saine totale. Il est peut-être préférable d'utiliser une relation $n'=10n$, mais cela demande un nombre de simulation qui croit rapidement. Les deux critères : détermination d'un couple stimulus/mesure et définition de la relation entre n et n' font que la mise en place d'une approche permettant de définir dynamiquement le nombre de descriptions saines est difficile à mettre en place. Cette méthode n'a donc pas été mise en place.

5.3.2.2.4. Définition du nombre de modèles sains par étude de la « normalité »

Une seconde approche permettant de qualifier la précision de la plage estimée consiste à étudier la « normalité » de la distribution [5.2]. L'objectif est de s'assurer que la population saine totale suit une loi gaussienne ; elle pourra alors ainsi être estimée par une distribution gaussienne obtenue à partir d'un sous-échantillon. Le principe consiste à comparer la probabilité cumulée de la mesure estimée à partir d'un nombre réduit de descriptions (estimation gaussienne $m_{\text{estimé}}$, $\sigma_{\text{estimé}}$) avec la probabilité cumulée d'un nombre important de descriptions saines simulées servant de référence, la Fig. 5.11 illustre cette approche. Par exemple, la probabilité cumulée de 1000 descriptions saines pour une mesure donnée est représentée en trait continu, elle sert de référence. La courbe composée de carrés représente l'estimation gaussienne déterminée à partir de la simulation de 50 descriptions saines. L'étude de la « normalité » consiste à comparer ces deux courbes. Lorsque l'erreur est suffisamment faible, il est possible de conclure que le nombre de descriptions saines est suffisant pour estimer la population saine réelle et que la population réelle suit une loi gaussienne.

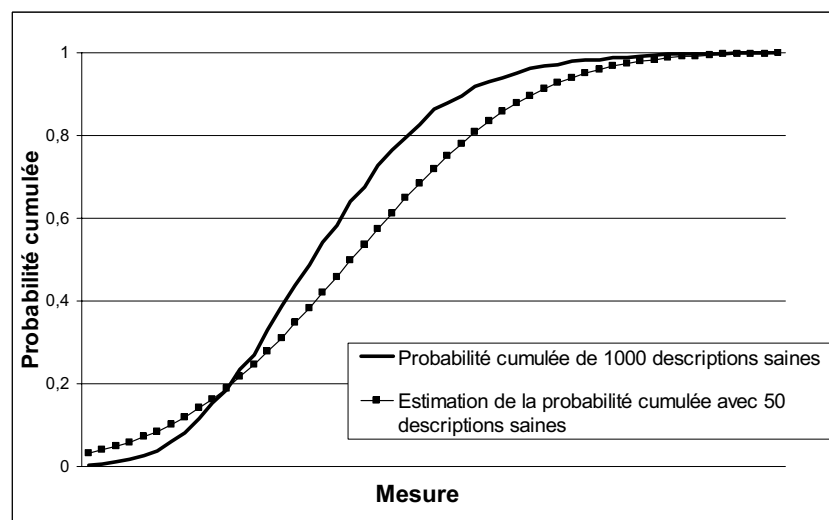


Fig. 5.11. Probabilité cumulée de la mesure déterminée à partir d'un échantillon de descriptions saines.

Dans nos travaux, nous n'avons pas implémenté la méthode permettant de définir dynamiquement le nombre minimal de descriptions saines, mais nous proposons d'étudier la variation des plages de mesures pour différents échantillons de descriptions saines et d'étudier la « normalité » des réponses de l'échantillon utilisé pour un stimulus et une mesure (§5.4.3.2). Ces approches permettent ainsi de vérifier la qualité de l'hypothèse faite sur la précision de la plage de mesure obtenue et sur la « normalité » de la population saine.

5.3.3. Modèle de fautes

Le modèle de fautes est un point clé de l'évaluation des stimuli de test par la simulation de fautes, l'objectif est de déterminer le stimulus qui va entraîner une réponse de la description fautive différente de celles des descriptions saines. La génération des descriptions fautives se fait en utilisant une méthode d'injection de fautes basée sur « l'utilisation des commandes du simulateur » (§3.3.3.2.1). Dans la partie §5.3.1.3, nous avons justifié le choix de cette approche car elle permet de diminuer les temps nécessaires à la qualification des stimuli. Cette méthode consiste à modifier une partie de la description afin de modéliser un comportement fautif. Lors de la définition des

descriptions fautives, nous supposons que l'architecture du système a été vérifiée au niveau fonctionnel, nous cherchons maintenant à valider l'étape de budgétisation des spécifications des blocs c'est-à-dire à s'assurer que les différents blocs et que les paramètres ont été correctement définis. Ainsi, l'architecture ayant été préalablement validée, nous n'injectons pas de fautes modifiant la structure du système ; l'injection de fautes paramétriques (§3.2.2.3) est donc suffisante pour valider la budgétisation des paramètres de chaque bloc.

Les descriptions fautives peuvent être créées soit en injectant une faute unique, soit en injectant plusieurs fautes simultanément. Pour injecter plusieurs fautes, une des méthodes utilisée au niveau structurel consiste à tenir compte de la variabilité de tous les paramètres et de procéder à un tirage aléatoire (cf. [5.1] et [5.5]). Les descriptions obtenues sont classées comme fautives si au moins un des paramètres est situé à l'extérieur de spécifications. Cette approche demande la génération d'un nombre important de descriptions et donc des temps de simulation longs. L'injection d'une faute unique est justifiée car on suppose que lors de la conception d'un système, le concepteur a de faibles chances de faire plusieurs erreurs. Cependant, si plusieurs erreurs sont présentes dans le système on fait l'hypothèse que leurs effets vont s'additionner dans la plupart des cas et entraîner un comportement fautif qui sera équivalent à l'effet d'une faute unique c'est-à-dire détectable voire même plus facilement détectable. Cette hypothèse sera validée dans le chapitre suivant. Dans le cas rare où les fautes multiples se compensent et n'entraînent aucun comportement fautif alors ces fautes ne seront jamais détectées.

Les descriptions fautives sont définies en utilisant la description originale dans laquelle tous les paramètres sont à leur valeur typique et en faisant varier un seul paramètre à l'extérieur de ses spécifications. Certes il serait possible de faire aussi varier les paramètres non-fautifs à l'intérieur de leurs plages de spécifications comme dans [5.5], mais cette approche nécessite la simulation d'un nombre important de descriptions et entraîne des temps de simulation très importants. Cette simplification suppose que l'on fait l'hypothèse que le paramètre muté va avoir une influence plus importante sur la variation d'une mesure que les autres paramètres. Il est toutefois nécessaire de quantifier l'erreur de précision qui est faite sur la métrique lorsque l'on fait cette hypothèse c'est-à-dire lorsque l'on utilise un modèle de fautes ayant une seule valeur modifiée et tous les autres définies à la valeur typique. Comme il a été défini lors de la justification du choix des descriptions saines, la méthode de qualification proposée est relative c'est-à-dire qu'elle permet de comparer des stimuli ensembles mais ne permet pas de quantifier en absolu le nombre de circuits défectueux détectés pour chaque stimulus. Notre approche permet cependant de déterminer si un stimulus est capable de détecter une variation d'un paramètre à l'extérieur de ses limites. Pour valider l'hypothèse faite, il est nécessaire de déterminer l'effet de la variation des autres paramètres sur la variation de la mesure de la description saine et donc sur la détection. L'objectif est alors de montrer que l'effet des variations est faible mais surtout qu'il est sensiblement identique quelque soit le stimulus utilisé et donc qu'il est tout à fait possible de comparer des stimuli entre eux en faisant cette hypothèse. Des résultats de simulation seront présentés dans la partie 5.4.4 pour un cas d'étude donné et un nombre réduit de stimuli.

Il reste maintenant à choisir la valeur à donner au paramètre fautif, le choix de la valeur est directement lié à la métrique utilisée pour la qualification, ce calcul sera donc développé dans la partie suivante (§5.3.4).

Cette partie a permis de présenter notre modèle de fautes, celui-ci modélise des erreurs de conception qui doivent être détectés par les stimuli utilisés lors de la vérification. Comme ce modèle de fautes est basé sur la variation des paramètres modélisés dans la description, la précision de la description et plus particulièrement le choix des paramètres modélisés ont un impact direct sur la qualité de l'évaluation des stimuli. Dans le chapitre IV, nous avons présenté la modélisation comportementale d'un système RF. Le choix du niveau de modélisation comportemental a été justifié afin de faire un compromis entre précision de la description et temps de simulation. Lors de la définition de la description comportementale, il est nécessaire d'identifier les paramètres à modéliser (fonctionnels et électriques) pour réaliser des vérifications efficaces (§3.2.1.3). Dans notre cas, il s'agit donc naturellement des paramètres qui ont été budgétisés lors de la conception et qui permettent de décrire à la fois les fonctionnalités et les comportements non-idéaux des blocs du système.

5.3.4. Métrique de qualification

La définition de la métrique de qualification est une partie importante pour l'évaluation des stimuli de test car elle a un impact direct sur la précision de la qualification. La métrique doit être capable de quantifier la capacité d'un stimulus à détecter une faute ou plus exactement une variation d'un paramètre hors des plages spécifiées lors de l'étape de conception. La métrique est ainsi directement liée à la valeur de la faute détectée, nous commencerons donc par présenter le calcul de la valeur de la faute.

Comme énoncé dans la partie précédente, notre modèle de fautes consiste à faire varier la valeur de chaque paramètre à l'extérieur de sa plage de spécifications. Le choix de cette valeur est important ; si celle-ci est très éloignée des limites alors la faute est facilement détectable car elle entraîne un comportement différent de celui des descriptions saines. Dans ce cas, même des stimuli de qualité médiocre seront capables de détecter cette faute. Par opposition, le choix d'une valeur fautive proche de la limite des spécifications est difficilement détectable car le comportement est proche du comportement sain et la robustesse du design risque de masquer les comportements fautifs. La plupart des stimuli ne pourront pas détecter de différence et dans ce cas aucun stimulus ne sera conservé lors de l'optimisation. L'approche utilisée pour éviter ce problème consiste à utiliser des paramètres fautifs à valeurs variables, c'est-à-dire que l'on modifie itérativement la valeur d'un paramètre fautif afin de déterminer la limite de détection de chaque stimulus. La Fig. 5.12 illustre le choix de la valeur du paramètre fautif, l'axe des abscisses représente la valeur d'un des paramètres du système. Par définition, lorsque la valeur du paramètre est située à l'intérieur des spécifications (P_{typ} , P_{min} , P_{max}), il s'agit d'un circuit sain. Ainsi tout circuit qui se trouve à l'extérieur de la zone gris-claire (Fig. 5.12 en bas à gauche) correspond aux circuits fautifs. Lors de la qualification, l'objectif est de trouver la variation minimale du paramètre fautif. Pour cela, nous devons définir une zone bornée par une des limites des spécifications P_{max} ou P_{min} et respectivement une valeur P_{max_fault} ou P_{min_fault} . Cette valeur P_{max_fault} (resp. P_{min_fault}) correspond à la valeur maximale (resp. minimale) du paramètre fautif acceptable. Nous supposons qu'une erreur de conception au-delà de ces limites a peu de chance de se produire sans que le concepteur ne se rende compte de son erreur ; si toutefois c'est le cas nous admettons que la détection de variation inférieure à cette limite permet de détecter une variation supérieure à cette valeur de faute car nous supposons que le système est monotone. Le principe consiste ensuite à déterminer la limite de détection en procédant par

itérations successives. La première valeur de la faute est définie au centre de la zone « Circuits fautifs ». Une dichotomie permet alors d'obtenir la limite de détection en diminuant la variation du paramètre fautif si la description fautive a été détectée (cas 1 et 2 Fig. 5.12) ou en augmentant la variation si la faute n'a pas été détectée (cas 3 et 4 Fig. 5.12). Au final, nous obtenons la valeur limite de détection P_{lim} qui correspond à la variation minimale de valeur du paramètre fautif qui peut être détectée (quand tous les autres paramètres sont à leur valeur typique (Hypothèse faite dans §5.3.3)) grâce au stimulus évalué.

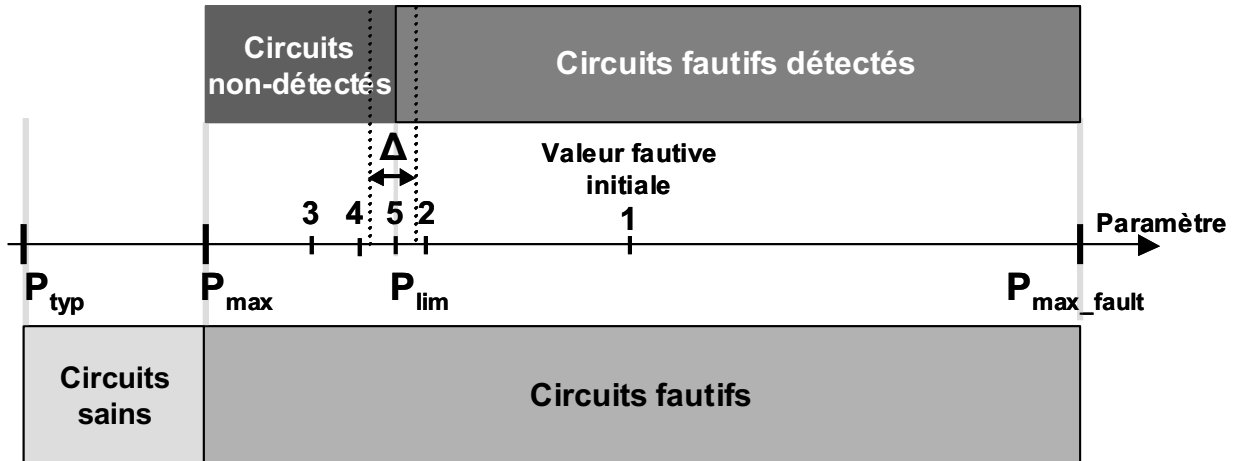


Fig. 5.12. Définition de la faute paramétrique.

La détermination de la valeur limite de détection (P_{lim}) permet de diviser l'aire « Circuits fautifs » en deux parties (Fig. 5.12 en haut). La première partie correspond aux valeurs de paramètres des descriptions fautives qui sont détectées par le stimulus évalué : cette partie est nommée «Circuits fautifs détectés ». La seconde partie correspond aux descriptions fautives qui n'ont pas pu être détectées : ce sont les « Circuits non-détectés ». Il devient alors possible de qualifier facilement les stimuli évalués en utilisant ces deux aires. L'objectif est d'obtenir un nombre de « Circuits non-détectés » nul, ce qui revient à dire que n'importe quelle variation du paramètre considéré à l'extérieur de ses spécifications est détectée par ce stimulus. Nous pouvons définir pour chaque paramètre P_i une métrique de qualification. Le « Taux Relatif de Couverture des Paramètres » est défini par les formules suivantes pour les bornes supérieures et inférieures :

$$TRCP_{P_i} = 1 - \frac{(P_{lim_P_i} - P_{max_P_i})}{(P_{max_fault_P_i} - P_{max_P_i})} \quad (\text{Eq. 5.12})$$

$$TRCP_{P_i} = 1 - \frac{(P_{min_P_i} - P_{lim_P_i})}{(P_{min_P_i} - P_{min_fault_P_i})} \quad (\text{Eq. 5.13})$$

Le $TRCP_{P_i}$ permet de qualifier la capacité d'un stimulus à détecter une variation du paramètre P_i ; cette valeur est bornée entre [0 ; 1]. Lorsqu'elle atteint 100%, alors cela signifie que le stimulus permet de détecter la moindre variation du paramètre fautif P_i à l'extérieur de ses spécifications.

Le terme « relatif » est utilisé dans la métrique car la valeur du $TRCP_{P_i}$ obtenue dépend de la valeur P_{max_fault} ou P_{min_fault} choisie. En effet, dans l'équation (Eq. 5.12), nous pouvons constater que la valeur P_{max_fault} (resp. P_{min_fault} (Eq. 5.13)) a un impact direct sur la métrique, il faut donc correctement choisir cette valeur afin de ne pas définir une métrique trop optimiste ou au contraire trop pessimiste. Dans notre cas, nous avons choisi de lier cette valeur à partir de la variation tolérée

par le concepteur, elle est donc proportionnelle à la plage de spécification comme définie en (Eq. 5.14).

$$P_{\max_fault} = k.(P_{\max} - P_{typ}) \quad (\text{Eq. 5.14})$$

$$P_{\min_fault} = k.(P_{typ} - P_{\min}) \quad (\text{Eq. 5.15})$$

La définition de P_{\max_fault} et P_{\min_fault} en fonction des plages de spécifications n'enlève pas le caractère relatif de la métrique car il reste toujours à définir la valeur de k . Nous proposons de définir cette valeur de manière à ce qu'elle soit grande par rapport à l'erreur induite par l'hypothèse faite lorsque l'on suppose que tous les paramètres non fautifs sont laissés à leur valeur typique. Cette erreur de précision est représentée par l'intervalle Δ de la Fig. 5.12. Suivant que les variations des paramètres non fautifs (à l'intérieur de leurs spécifications) se combinent de façon favorable, il sera possible d'obtenir une limite de détection plus proche de la limite des spécifications c'est-à-dire que le stimulus sera capable de détecter de faibles variations et donc calculera un $TRCP_{Pi}$ plus élevé (limite en pointillé de gauche sur la Fig. 5.12). Si au contraire, les paramètres se combinent dans un cas défavorable, la limite de détection sera plus éloignée de la limite de détection obtenue quand tous les paramètres sont à leur valeur typique. Cela signifie que l'on pense pouvoir détecter de faibles variations, mais qu'en réalité la combinaison des autres paramètres peut masquer cette détection. La limite de détection réelle est donc plus éloignée (limite en pointillé de droite sur la Fig. 5.12). Nous proposons de définir la valeur de k en fonction de cette erreur de précision Δ représentée sur la Fig. 5.12. En effet, comme cette erreur de précision peut fausser la qualification de stimuli, le choix de k se fera afin que la partie « Circuits fautifs » soit 100 fois supérieure à l'erreur engendrée par notre hypothèse simplificatrice ce qui correspond à une erreur de 1%. Ainsi, il faut calculer Δ pour un stimulus donné et définir la valeur de k égale à 100Δ .

Dans une approche de qualification relative, la valeur de k n'est pas essentielle car on cherche à comparer le $TRCP_{Pi}$ de différents stimuli (calculés pour une même valeur de k). Il est donc possible d'utiliser cette métrique pour faire de l'optimisation de stimuli de test en procédant par comparaison par rapport à d'autres stimuli. Nous montrerons dans §5.4.4, pour un cas donné comment est évaluée l'imprécision sur la description fautive et donc comment est définie la valeur de k .

5.4. Résultats de qualification de stimuli pour la validation de conception

Cette partie présente des résultats de simulation permettant de valider notre approche de qualification des stimuli pour la validation de conception par simulation de fautes comportementales. La méthode proposée est appliquée sur la partie réception du système W-CDMA présentée dans le chapitre IV.

5.4.1. Définition des paramètres comportementaux, choix des descriptions saines et fautives

La description de la partie réception est décrite par 23 paramètres comportementaux. Ces différents paramètres sont représentés sur le schéma bloc de la partie réception de la Fig. 5.13. Chacun de ces paramètres est défini par une valeur typique et une ou deux valeurs limites ; Tab. 5.2 présente les caractéristiques d'un échantillon de paramètres. Lorsque pour une limite, la valeur dans ce tableau est notée « X », cela signifie qu'il n'y a pas de limite pour ce paramètre. Dans ce cas, aucune valeur limite n'est critique pour le fonctionnement du système. Par exemple, lors de la spécification du point de compression, une valeur minimale doit être définie afin de fixer la non-linéarité maximale

autorisée. La définition d'une valeur du point de compression maximale n'a pas de sens car cela voudrait dire que l'utilisateur se fixe des contraintes sur un paramètre qui n'est pas critique.

Au final, d'après notre modèle de fautes décrit précédemment, nous obtenons un ensemble de 36 descriptions fautives qui devront être détectées par les stimuli lors de la validation. Ce nombre correspond à 10 fautes pour les 10 paramètres spécifiés par une seule limite et à 26 fautes pour les 13 paramètres spécifiés par 2 limites.

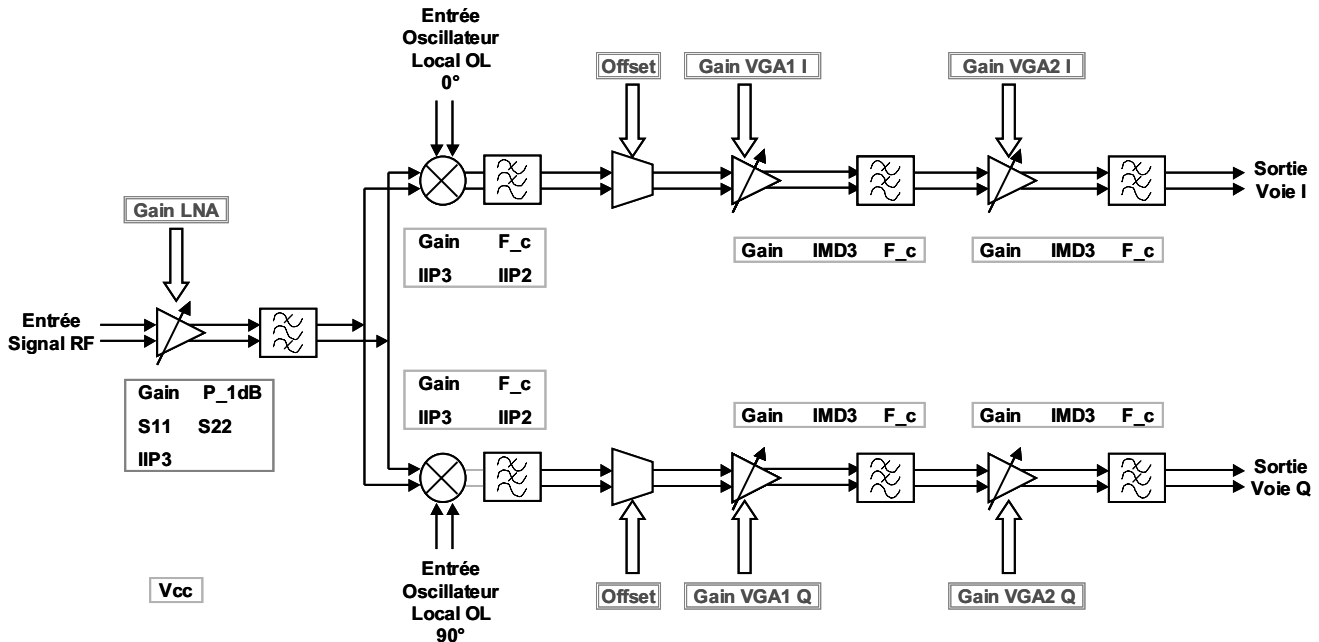


Fig. 5.13. Paramètres comportementaux décrits dans le système W-CDMA.

| Nom | Val_typ | Val_min | Val_max |
|------------------|---------|---------|---------|
| LNA_Gain (dB) | 13,85 | -15 | 15 |
| VGA1_I (dB) | 0 | -15 | 15 |
| S11_LNA (dB) | -14,47 | X | -10,25 |
| S22_LNA (dB) | -6,1 | X | -1 |
| P_comp_LNA (dBm) | -8,834 | -18,4 | X |
| IIP3_LNA (dBm) | -0,24 | -5,03 | X |
| Gain_IQ_dem (dB) | 22 | 20 | 24 |

Tab. 5.2. Extrait du tableau des spécifications de la partie réception du système W-CDMA.

5.4.2. Stimuli de test

Dans notre cas d'étude, les stimuli de test ont été déterminés manuellement afin de valider la conception du circuit. Ces stimuli permettent de vérifier les caractéristiques du circuit définies dans le cahier des charges en faisant varier les puissances et les fréquences des signaux en entrée. De plus, ils vérifient le fonctionnement du système pour différentes configurations c'est-à-dire pour différentes valeurs de gain et d'offset (§4.2.3. et Tab. 5.3). Il est également nécessaire d'évaluer la capacité du système à rejeter des fréquences parasites. Pour cela, les signaux d'entrées sont définis à des fréquences situées hors du domaine de fonctionnement afin de représenter ces fréquences parasites et la mesure se fait de manière à identifier le rejet de ces raies parasites. Par exemple, si on

cherche à vérifier la fréquence de coupure d'un filtre, on va injecter un signal à la limite de cette fréquence et on va mesurer la puissance de signal à la sortie du filtre ; il sera alors possible de vérifier que celle-ci a bien été rejetée.

Comme, il n'est pas possible de simuler le système avec tous les stimuli définis manuellement lors de la validation de conception c'est-à-dire validant la conception dans toutes les configurations possibles et sur toute sa plage de fonctionnement ; un échantillon des stimuli a été retenu afin de permettre la validation du circuit. Lors de cette évaluation, nous avons identifié un nombre réduit de stimuli et choisi de privilégier les stimuli mono-ton et bi-tons afin d'avoir des temps de simulation raisonnables. Ainsi, nous avons utilisé 114 stimuli que nous avons qualifiés grâce à PLASMA afin de déterminer leur capacité à détecter des fautes comportementales.

La notion de stimuli a été définie dans la partie §3.3.3. du chapitre III. Un stimulus est composé de deux parties, nous allons maintenant détailler les paramètres constituant ces parties. La première partie comprend les paramètres configurables tels que la valeur des amplificateurs à gains variables. La seconde partie spécifie les propriétés des signaux appliqués sur les entrées du système : type de signal utilisé (simple-ton, multi-tons, modulé) et ses caractéristiques (puissance, phase et fréquence). Un exemple de stimulus est présenté dans Tab. 5.3.

| Configuration du système | | | | | Signaux d'entrées | | | | | |
|--------------------------|-------------|-------------|-------------|-------------|-------------------|-------|-----------|-----------|-------|-----------|
| | | | | | Entrée RF | | | Entrée OL | | |
| Gain LNA | Gain VGA1_I | Gain VGA1_Q | Gain VGA2_I | Gain VGA2_Q | Puissance | Phase | Fréquence | Puissance | Phase | Fréquence |
| 15dB | -5dB | 5dB | 0dB | 0dB | -30dBm | 0° | 2141MHz | -20dBm | 0° | 2140MHz |

Tab. 5.3. Définition d'un stimulus de test utilisé.

5.4.3. Evaluation des plages de mesures de la population saine

Dans cette partie, nous allons valider, pour un circuit donné, l'hypothèse qui a été faite lors de la définition des plages de mesures des descriptions saines (§5.3.2). Nous proposons d'utiliser un échantillon de la population saine pour déterminer les plages de mesures des circuits sains car il est impossible de simuler toutes les descriptions saines. Dans cette partie, nous quantifions l'erreur de précision qui est faite lors de la définition de la plage de mesure des modèles sains à partir d'un échantillon. Pour cela, nous qualifions les plages de mesures saines obtenues par l'utilisation des différentes méthodes présentées précédemment (§5.3.2.1 et §5.3.2.2) par rapport à une plage de mesure déterminée à partir d'une population saine de référence. Ces plages sont définies pour un stimulus donné (Tab. 5.3) et pour la mesure de la tension à 1MHz du signal démodulé de la sortie voie I (Fig. 5.13).

La plage de référence est déterminée en simulant un nombre important de descriptions saines générées aléatoirement suivant une distribution gaussienne. Nous choisissons de simuler 20000 descriptions saines et nous supposons que ce nombre est suffisant pour avoir une plage de mesure représentative de la population saine totale. Ce nombre a été choisi afin d'être le plus important possible mais pour une durée de simulation raisonnable d'environ 10 heures.

5.4.3.1. Définition des plages de mesures par échantillonnage pires-cas « simplifié »

Les plages de mesures sont déterminées en identifiant les mesures minimales et maximales obtenues par simulation de l'échantillon de référence et de l'échantillonnage pires-cas simplifié

(§5.3.2.1). Tab. 5.4 montre les résultats de simulations. La première ligne représente la plage de mesure de référence, c'est-à-dire celle de la population saine totale (20000 descriptions saines). La plage de référence est comprise entre 5,6mV et 9,2mV. La plage de mesure obtenue par l'utilisation d'un échantillon de population pires-cas simplifié a permis d'obtenir une plage de mesure qui est proche de la plage de mesure de référence avec une erreur de 7,5%.

| Echantillon de descriptions saines | | Mesure Min (mV) | Mesure Max (mV) | Erreur relative par rapport à la plage de référence (%) | Temps de simulation (min) |
|------------------------------------|-------|-----------------|-----------------|---|---------------------------|
| Référence | 20000 | 5,6 | 9,2 | | 600 |
| Pire cas simplifié | 36 | 5,7 | 9,03 | 7,5 | 6 |

Tab. 5.4. Définition des plages de mesures à partir de l'échantillonnage pires-cas simplifié.

5.4.3.2. Définition des plages de mesures par estimation gaussienne

Lors de l'utilisation de l'estimation gaussienne, les plages de mesures ne sont pas déterminées à partir des valeurs minimales et maximales, mais elles sont déterminées en fonction du nombre de circuits sains que l'on accepte de rejeter (§5.3.2.2.2). La plage de référence est donc différente de celle utilisée dans la partie précédente car elle n'est plus obtenue en prenant les mesures minimales et maximales, mais elle est déterminée en calculant la moyenne et l'écart type des 20000 mesures et en déterminant la limite par exemple à 4σ (Eq. 5.2) et (Eq. 5.3)). Les limites de mesures ainsi obtenues sont de 5,8mV et 8,7mV.

Pour les autres lignes du tableau, la définition des plages de mesure a été réalisée à partir d'un échantillon de la population saine et d'une estimation gaussienne (§5.3.2.2). Les moyennes et les écarts types de chaque échantillon sont calculés et les plages de mesures sont estimées à partir des équations (Eq. 5.10) et (Eq. 5.11). Nous pouvons constater que l'erreur de précision diminue lorsque le nombre d'échantillons augmente. Ainsi à partir de 100 modèles sains, l'erreur sur l'approximation par rapport à la plage de référence est inférieure à 3%.

| Echantillon de descriptions saines | | Mesure Min à -4σ (mV) | Mesure Max à $+4\sigma$ (mV) | Erreur relative par rapport à la plage de référence (%) | Temps de simulation (min) |
|------------------------------------|-------|------------------------------|------------------------------|---|---------------------------|
| Estimation gaussienne | 20000 | 5,8 | 8,7 | 0,00 | 600 |
| | 10 | 5,97 | 7,78 | 37,59 | 2 |
| | 30 | 6,26 | 8,41 | 25,86 | 6 |
| | 50 | 6,13 | 8,52 | 17,59 | 10 |
| | 100 | 5,74 | 8,72 | 2,76 | 20 |

Tab. 5.5. Définition des plages de mesures à partir de l'estimation gaussienne.

Pour valider la méthode d'estimation gaussienne, nous allons utiliser le quantile normale [5.2] afin de s'assurer que la population saine totale suit une loi gaussienne et peut donc être estimée par une gaussienne déterminée à partir d'un échantillon des descriptions saines. La méthode a été présentée dans §5.3.2.2.4, elle consiste à comparer la probabilité cumulée de la population saine totale avec la probabilité cumulée estimée à partir de l'échantillon. La Fig. 5.14 représente les différentes probabilités cumulées, la courbe noire représente la population saine de référence (20000 descriptions saines simulées). Les moyennes et les écarts types sont estimés à partir de plusieurs

échantillons (10, 30, 50, 100, 1000). Les probabilités cumulées estimées par la loi gaussienne sont représentées dans la Fig. 5.14. Les courbes de la Fig. 5.15 représentent l'erreur relative qui est faite lorsque l'on estime la probabilité cumulée de la population saine à partir d'une estimation gaussienne.

Ces résultats nous permettent de déduire que la population saine totale suit une loi normale car l'erreur diminue lorsque le nombre d'échantillon augmente c'est-à-dire lorsque l'échantillon tend vers la population totale. Si la limite sur l'erreur permettant d'affirmer que la population saine totale suit une distribution normale est fixée à 15%, on peut en déduire qu'il faut au minimum simuler 50 descriptions saines tirées aléatoirement pour obtenir une estimation « correcte » de la population totale et donc de définir des plages de mesures proches des plages de mesures de la population saine totale.

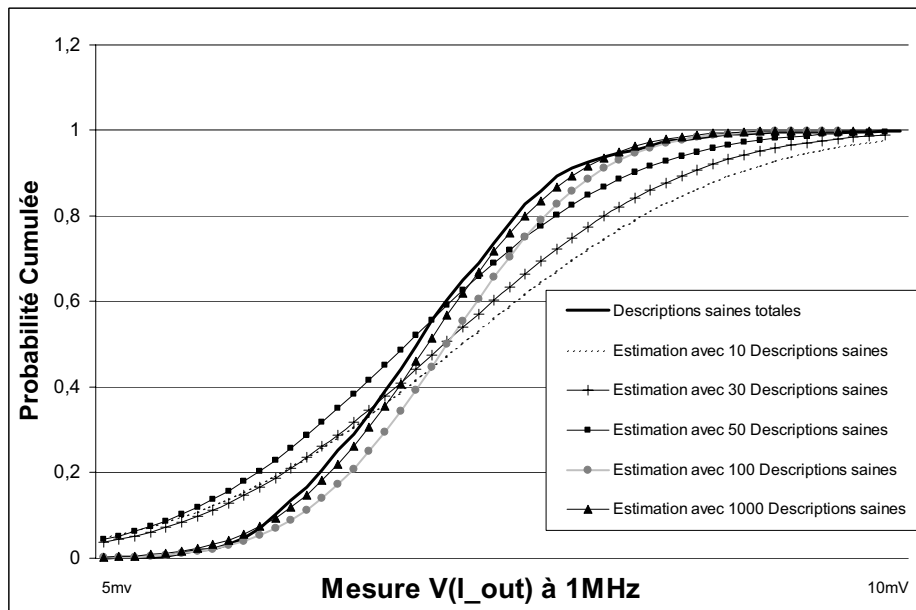


Fig. 5.14. Probabilité cumulée de l'estimation gaussienne de la population saine à partir de différents échantillons.

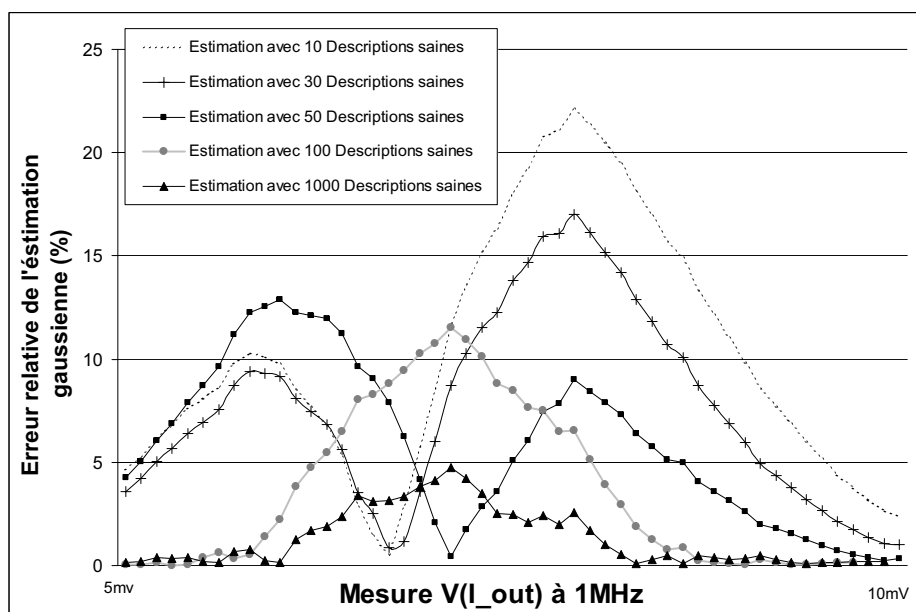


Fig. 5.15. Erreur relative de l'estimation gaussienne pour les différents échantillons.

Ces résultats de simulation obtenus pour un cas donné, c'est à dire pour un circuit RF simulé avec un stimulus et en observant une mesure ont montré que l'erreur de précision sur les plages de mesures peut être faible mais n'est pas négligeable. En effet, l'erreur sur la plage de mesure engendrée par un échantillonnage pires-cas simplifié ou un échantillonnage par estimation gaussienne est de 7,5% et respectivement 3%; il faut donc les prendre en compte. Comme énoncé dans §5.3.2.1, l'erreur de précision des modèles, des simulations, des appareils de mesures doit aussi être considérée lors de la définition des plages de mesures. En effet, il n'est pas utile d'avoir une grande précision sur la définition des plages de mesures si celle-ci est très faible par rapport aux erreurs citées précédemment. Il est difficile de déterminer l'erreur due au manque de précision des modèles mais la précision des appareils de mesures utilisés lors de la vérification du prototype est plus facile à obtenir et est d'environ 5%. L'erreur engendrée par l'utilisation d'un échantillon de population est donc du même ordre de grandeur et a un effet non négligeable sur la qualification. Cependant, comme nous évaluons les stimuli à partir d'une métrique relative, l'erreur d'estimation est la même pour les différents stimuli évalués à partir du même échantillon de descriptions saines. Il est donc tout à fait possible de comparer les résultats de qualifications de plusieurs stimuli entre eux.

5.4.4. Evaluation du modèle de fautes et de la métrique

Nous allons maintenant valider l'hypothèse faite sur notre modèle de fautes. En effet dans §5.3.3, nous avons considéré que l'injection de variation unique dans la description était suffisante pour la qualification de stimuli de test. Pour cela, nous avons fait l'hypothèse que l'erreur faite sur la limite de détection (Δ de Fig. 5.12) est faible lorsque nous ne considérons pas les variations des autres paramètres à l'intérieur de leur plage de spécification.

Dans cette partie, nous présentons des résultats de simulation pour un circuit et pour un paramètre fautif donné. Ces résultats permettent dans un premier temps de déterminer l'erreur de précision (Δ) puis de montrer que celle-ci est sensiblement identique pour différents stimuli. Il est alors possible de comparer la qualité de plusieurs stimuli entre eux.

La première étape permettant l'évaluation de l'hypothèse consiste à déterminer la plage de mesure des descriptions saines pour un stimulus et pour une mesure (surface blanche Fig. 5.16). La limite de détection (P_{lim}) de la variation sur un paramètre fautif est ensuite calculée lorsque tous les autres paramètres sont à leur valeur typique. L'étape suivante permet de déterminer la plage de mesure des descriptions fautives quand le paramètre fautif est à sa limite de détection (P_{lim}) et que les autres paramètres varient dans leur plage de spécification (surface grise Fig. 5.16). Par l'hypothèse faite sur le modèle de fautes, nous avons considéré que la variation du paramètre fautif à la valeur P_{lim} est détectée quelque soit les variations sur les autres paramètres. Or, si nous comparons les plages de mesures nous pouvons constater qu'elles se chevauchent (Fig. 5.16) ; cela signifie que lorsque le paramètre fautif est à sa limite de détection (P_{lim}) il y a des descriptions fautives qui ne sont pas détectées. La description fautive ayant permis d'obtenir la mesure la plus importante et qui est incluse dans la plage des descriptions saines est alors appelée la description pires-cas (Max Fig. 5.16). En réutilisant cette description fautive pires-cas et en faisant varier le paramètre fautif jusqu'à ce qu'il devienne détectable, il est possible d'obtenir une nouvelle limite de détection (P_{lim_min} Fig. 5.17). Au delà de cette limite, on est sûr de détecter la variation du paramètre fautif quelque soit la

combinaison des autres paramètres à l'intérieur de leur plage de spécification ; on parle alors de la description fautive pires-cas pouvant être détectée. Le même raisonnement est suivi pour la définition de la limite de détection meilleur cas (P_{lim_max} Fig. 5.17). Il suffit de prendre la description fautive meilleur cas c'est-à-dire ayant permis d'obtenir la mesure la plus éloignée de la plage des descriptions saines (Min Fig. 5.16) et de calculer la limite de détection meilleure cas (P_{lim_max}) en faisant varier le paramètre fautif jusqu'à ce qu'il ne soit plus détectable. Au final, la distance entre les limites de détection meilleure (P_{lim_max}) et pires-cas (P_{lim_min}) permet de déterminer l'erreur de précision Δ (Fig. 5.12 et Fig. 5.17).

Dans notre cas, nous cherchons à comparer l'erreur de précision faite pour différents stimuli, nous appliquons cette méthode pour trois stimuli donnés (Tab. 5.6). La liste de stimuli est composée de deux stimuli mono-ton et d'un stimulus bi-tons.

| Stimulus n° | Configuration du système | | | | | Signaux d'entrées | | | | | |
|----------------|--------------------------|----------------|----------------|----------------|----------------|-------------------|-------|-----------|-----------|-------|-----------|
| | | | | | | Entrée RF | | | Entrée OL | | |
| | Gain LNA | Gain VGA1_I | Gain VGA1_Q | Gain VGA2_I | Gain VGA2_Q | Puissance | Phase | Fréquence | Puissance | Phase | Fréquence |
| 1 | 15dB | -1dB | -1dB | 23dB | 23dB | -40dBm | 0° | 2168,6MHz | -20dBm | 0° | 2167,6MHz |
| 2 | -15dB | 15dB | 15dB | 23dB | 23dB | -20dBm | 0° | 2113,4MHz | -20dBm | 0° | 2112,4MHz |
| 3 | -15dB | 15dB | 15dB | 23dB | 23dB | -30dBm | 0° | 2151MHz | -20dBm | 0° | 2140MHz |
| | | | | | | -20dBm | 0° | 2150MHz | -20dBm | 0° | 2000MHz |

Tab. 5.6. Définition des stimuli de test utilisés.

Par exemple, l'utilisation du stimulus n°1 nous permet d'obtenir une plage des descriptions saines obtenue pour une mesure de la voie I à 1MHz. Ce stimulus nous permet de détecter une variation de la fréquence de coupure du filtre LPF1_I ($F_{c_LPF1_I}$ Fig. 5.13 et Tab. 5.2) à partir de 3.28MHz (P_{lim}). Il est alors possible de déterminer la plage de mesure des descriptions fautives lorsque la fréquence de coupure est à la limite de détection (7,14mV à 7,17mV Fig. 5.16). Cette plage est obtenue en simulant un échantillon de 1000 descriptions fautives tirées aléatoirement c'est-à-dire que le paramètre fautif est fixé à sa limite de détection (3.28MHz) et que tous les autres paramètres varient à l'intérieur de leurs plages de spécifications. La Fig. 5.16 représente sur l'axe des abscisses (tension mesurée à 1MHz) les plages des descriptions saines et fautives. On peut remarquer que les plages des descriptions saines et fautives se chevauchent, cela signifie que des descriptions fautives pour $F_{c_LPF1_I} = 3.28\text{MHz}$ ne sont pas détectées. Comme énoncé ci-dessus, il existe donc une autre limite de détection pires-cas ($P_{lim_min}=3.08\text{MHz}$ cf. Fig. 5.17 et Tab. 5.7) qui correspond à la combinaison des paramètres entraînant le cas le moins favorable (description fautive pires-cas). La nouvelle limite (P_{lim_min}) ainsi obtenue permet de garantir que la description fautive sera détectée quelque soit les combinaisons des différents paramètres à l'intérieur de leur plage de spécification. Réciproquement, il est possible de déterminer la limite de détection meilleur cas ($P_{lim_max}=3.52\text{MHz}$ cf. Fig. 5.17 et Tab. 5.7), c'est-à-dire la variation minimum du paramètre fautif pour qu'au moins une description fautive (combinaison meilleure cas) soit détectée.

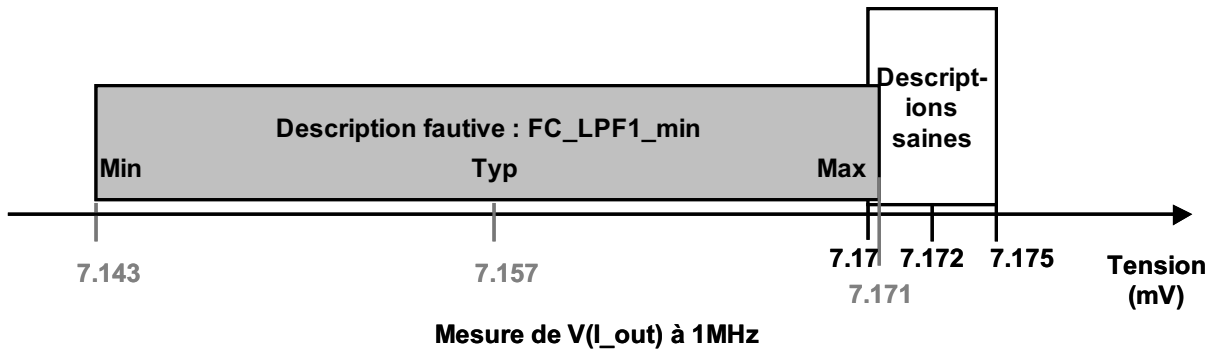


Fig. 5.16. Plage de mesure des descriptions saines et fautives.

La même manipulation a été faite pour les deux autres stimuli (Tab. 5.6). Dans un premier temps, les plages de mesures des descriptions saines et fautives ont été déterminées (cf. Colonnes « Mesures » Tab. 5.7). L’objectif est alors de transposer les plages obtenues sur les mesures en plages définies sur le paramètre fautif afin de pouvoir comparer les différents stimuli entre eux (dans notre cas : Fréquence de coupure du filtre LPF1_I (Colonnes « Limites de détections » Tab. 5.7)). La méthode de transposition a été définie ci-dessus lors de la considération des descriptions fautives pires-cas et meilleur cas. Les Fig. 5.18 et Fig. 5.17 reprennent les différents résultats obtenus lors de la simulation. Nous pouvons en conclure que le stimulus n°2 permet d’obtenir la limite de détection la plus proche des limites de spécifications et donc le TRCP le plus élevé. Au final, ce stimulus sera qualifié comme le plus efficace des trois.

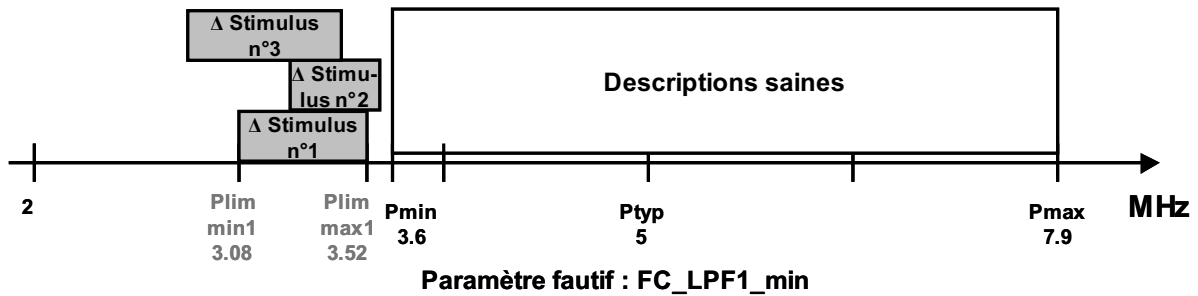


Fig. 5.17. Limite de détection.

| Stimulus n° | Mesures | | | | | | Paramètre Fc_LPF1 (MHz) | | | | | Erreur Relative à la plage des spécifications Δ (%) |
|-------------|---------------------|--------|--------|----------------------|--------|--------|-------------------------|-------|----------------------|----------|----------|---|
| | Descriptions saines | | | Descriptions Fautive | | | Spécifications | | Limites de détection | | | |
| | Min | Typ | Max | Min | Typ | Max | P Min | P Typ | Plim Min | Plim Typ | Plim Max | |
| 1 | 7,170 | 7,173 | 7,175 | 7,143 | 7,157 | 7,172 | 3,60 | 5,00 | 3,08 | 3,28 | 3,52 | 31,429 |
| 2 | 14,303 | 14,311 | 14,319 | 14,242 | 14,268 | 14,297 | 3,60 | 5,00 | 3,18 | 3,38 | 3,55 | 26,429 |
| 3 | 6,252 | 6,254 | 6,258 | 6,190 | 6,250 | 6,253 | 3,60 | 5,00 | 3,03 | 3,20 | 3,49 | 32,857 |

Tab. 5.7. Plages de mesures des descriptions saines et fautives et limites de détection.

L’intérêt de cette simulation est de montrer que l’erreur Δ engendrée par l’hypothèse faite sur notre modèle de fautes est sensiblement identique quelque soit le stimulus utilisé. La valeur de Δ est calculée à partir des limites de détection pires-cas et meilleur cas, elle est exprimée en fonction des spécifications. La valeur de Δ est représentée dans la dernière colonne de Tab. 5.7 et est de l’ordre de 30%. L’erreur de précision est sensiblement identique, il est donc possible de comparer les résultats de qualification de chaque stimulus entre eux en faisant cette hypothèse sur notre modèle de fautes.

Notre modèle de fautes consistant à injecter une seule variation sur le paramètre fautif et aucune sur les autres paramètres entraîne une erreur sur la limite de détection d'environ 30%. Comme énoncé dans §5.3.4, il est possible de rendre cette erreur négligeable sur la métrique (TRCP) en définissant le paramètre k à 100Δ . Par la suite, k sera fixé à 30 afin que l'hypothèse faite sur le modèle de faute n'ait pas d'impact sur la métrique.

5.4.5. Résultats de qualification des stimuli

La liste des 114 stimuli de test est qualifiée grâce l'outil PLASMA. Cette analyse nécessite un nombre important de simulations (≈ 23.000 simulations).

A la première itération de la dichotomie (§5.3.3), la valeur fautive du paramètre P_i est fixée au centre de l'aire des circuits fautifs (Fig. 5.12), la valeur P_{\max_fault} (resp. P_{\min_fault}) est définie d'après l'équation (Eq. 5.14) avec $k=30$ (§5.4.4). Ensuite, 5 itérations de dichotomie permettent d'approcher la limite de détection en tenant compte des détections précédentes (Fig. 5.12). Pour la définition des plages de mesures des descriptions saines, nous avons choisi d'utiliser l'échantillonnage par estimation gaussienne présenté en §5.3.2.2. Les plages des mesures sont définies en estimant les moyennes et les écarts types à partir de 50 descriptions saines. Les limites des plages de mesures des descriptions saines sont fixées d'après les équations (Eq. 5.10) et (Eq. 5.11) ou $h=4$. La valeur de h est déterminée d'après la partie §5.3.2.2.1 et la Fig. 5.9, on s'autorise ainsi de rejeter un nombre de circuits sains de 31 sur un million.

5.4.5.1. Couverture de fautes

Les résultats ainsi obtenus à partir de PLASMA permettent de qualifier les stimuli de test. Le premier résultat concerne la couverture de fautes ; parmi 36 descriptions fautives, notre ensemble de stimuli permet de détecter 21 fautes (Tab. 5.8). Le taux de couverture de fautes (CF) (défini dans la partie §3.3.3.3.) ainsi obtenu est de $21/36=58\%$; ce résultat nous permet de conclure que l'ensemble des stimuli analysés ne permet pas de détecter toutes les variations sur les paramètres comportementaux. Ce résultat nous montre tout d'abord que la liste des stimuli évaluée n'est pas suffisante. D'après Tab. 5.8, nous pouvons constater que toutes les fautes sur les fréquences de coupures supérieures et que plusieurs fautes sur la non-linéarité (IIP3, IIP2) n'ont pas été détectées. Nous allons maintenant expliquer les causes de ces non-détections.

| Paramètre Comportementaux | Valeur limite | Détection | N° Stimulus | Valeur fautive détectée | TRC _{PPi} |
|---------------------------|---------------|------------|-------------|-------------------------|--------------------|
| S11_LNA | -1,03E+01 | X | 1 | -9,89E+00 | 0,997 |
| S22_LNA | -1,00E+00 | | | | 0 |
| P_comp_LNA | -1,84E+01 | X | 108 | -2,59E+01 | 0,938 |
| IIP3_LNA | -5,03E+00 | X | 17 | -4,62E+01 | 0,656 |
| Gain IQ_dem | 2,00E+01 | X | 1 | 1,56E+00 | 0,631 |
| Gain IQ_dem | 2,40E+01 | X | 1 | 2,56E+01 | 0,969 |
| IIP3 IQ_dem | -1,50E+00 | | | | 0 |
| IIP2 IQ_dem | 3,90E+01 | | | | 0 |
| Fc LPF I | 3,60E+06 | X | 1 | 1,09E+06 | 0,928 |
| Fc LPF I | 7,90E+06 | | | | 0 |
| Fc LPF Q | 3,60E+06 | X | 1 | 1,09E+06 | 0,928 |
| Fc LPF Q | 7,90E+06 | | | | 0 |
| IMD3_VGA1_I | -1,07E+02 | | | | 0 |
| IMD3_VGA1_Q | -1,07E+02 | | | | 0 |
| Fc_VGA1_I | 2,60E+06 | X | 49 | 1,54E+06 | 0,438 |
| Fc_VGA1_I | 2,72E+06 | | | | 0 |
| Fc_VGA1_Q | 2,60E+06 | X | 49 | 1,54E+06 | 0,438 |
| Fc_VGA1_Q | 2,72E+06 | | | | 0 |
| IMD3_VGA2_I | -1,13E+02 | | | | 0 |
| IMD3_VGA2_Q | -1,13E+02 | | | | 0 |
| Fc_VGA2_I | 2,82E+06 | X | 49 | 1,60E+06 | 0,563 |
| Fc_VGA2_I | 2,98E+06 | | | | 0 |
| Fc_VGA2_Q | 2,82E+06 | X | 49 | 1,60E+06 | 0,563 |
| Fc_VGA2_Q | 2,98E+06 | | | | 0 |
| V_supply | 2,50E+00 | | | | 0 |
| V_supply | 2,70E+00 | | | | 0 |
| LNA_Gain | -2,00E+00 | X | 1 | -3,5625 | 0,969 |
| LNA_Gain | 2,00E+00 | X | 1 | 3,5625 | 0,969 |
| VGA1_I | 2,00E-01 | X | 1 | -1,91875 | 0,625 |
| VGA1_I | 2,00E-01 | X | 1 | 1,91875 | 0,625 |
| VGA1_Q | 2,00E-01 | X | 1 | -1,91875 | 0,625 |
| VGA1_Q | 2,00E-01 | X | 1 | 1,91875 | 0,625 |
| VGA2_I | 2,00E-01 | X | 1 | -1,91875 | 0,625 |
| VGA2_I | 2,00E-01 | X | 1 | 1,91875 | 0,625 |
| VGA2_Q | 2,00E-01 | X | 1 | -1,91875 | 0,625 |
| VGA2_Q | 2,00E-01 | X | 1 | 1,91875 | 0,625 |
| | CF | 21/36=0,58 | | TRCP | 0,416 |

Tab. 5.8. Résultats de détection de circuits fautifs.

Pour commencer, nous allons développer le cas de détection de la fréquence de coupure $F_{c_VGA1_I}$ des filtres passe-bas de la partie en bande de base (Fig. 5.13). Lorsque le signal est démodulé, la fréquence du signal fondamental entrant dans le filtre VGA1 (Fig. 5.13) correspond à la différence des fréquences de ces deux signaux ($F_{RF}-F_{OL}$). Les flèches (1 à 5) de la Fig. 5.18 représentent différents stimuli mono-ton qui peuvent être injectés à l'entrée du filtre. La détection de la faute injectée sur la fréquence inférieure ($F_{c_min_faute}$) est facilement détectable car il suffit que la fréquence du stimulus soit supérieure à $F_{c_min_faute}$ pour permettre la détection. Ainsi tous les stimuli sauf le n°1 permettent de détecter cette faute. Par contre, à partir des résultats de qualification obtenus (Tab. 5.8), nous constatons que la faute injectée sur la limite supérieure de ce paramètre n'est jamais détectée ($F_{c_faute_max}$). En effet, tant que la fréquence du signal d'entrée est inférieure à la limite maximum (F_{c_max}) des descriptions saines il n'est pas possible de détecter la faute $F_{c_faute_max}$ (stimulus 1, 2, 3, 4 Fig. 5.18). Il faut donc que la fréquence du stimulus appliqué à l'entrée du filtre soit supérieure à F_{c_max} . En se plaçant au niveau système, cela implique que l'écart de fréquences ($F_{RF}-F_{OL}$) doit être supérieur à F_{c_max} . Tout stimulus ne respectant pas ce critère ne permettra pas de détecter une variation de la fréquence de coupure supérieure à l'extérieur des spécifications. Ainsi, pour augmenter le nombre de détection de fautes,

il faut utiliser un stimulus dont l'écart de fréquence entre les deux signaux d'entrées (RF et OL) est supérieur à la valeur maximale de la fréquence de coupure.

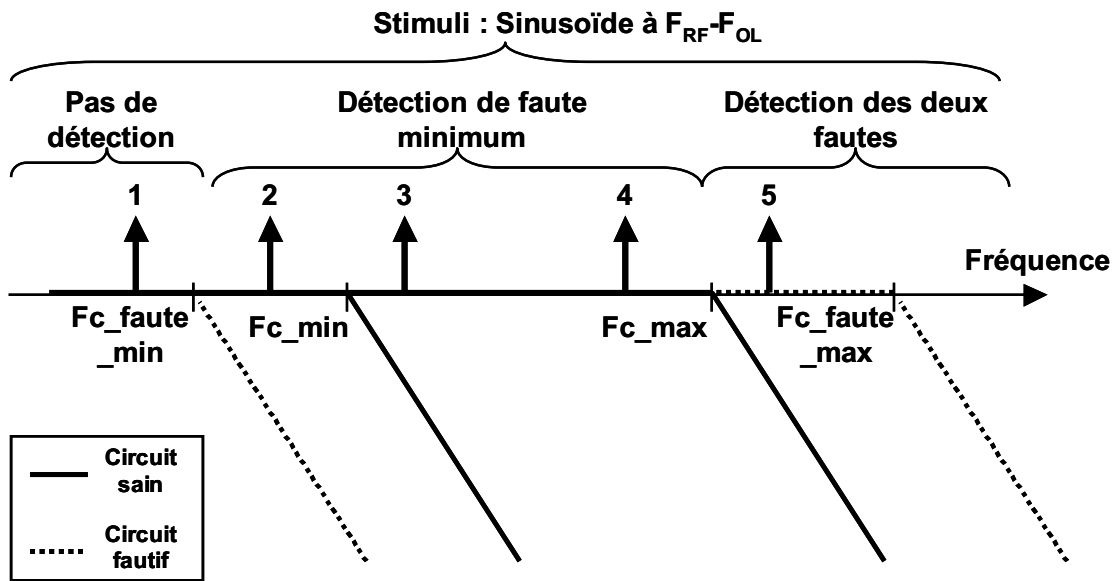


Fig. 5.18. Détection des fautes sur les fréquences de coupures.

Le second type de paramètres non détectés concerne les paramètres modélisant la non-linéarité et l'intermodulation. La Fig. 5.19 présente la non-linéarité et l'intermodulation. La courbe « P1er » représente la relation entre la puissance d'entrée et la puissance de sortie de la raie fondamentale. Tant que la puissance d'entrée est faible (stimulus 1 et 2 Fig. 5.19), l'effet de la saturation modélisé par le point de compression ne pourra pas être observé, il faut donc que le signal ait une puissance d'entrée suffisante pour mettre en évidence la différence de saturation du gain entre le circuit sain (trait continu) et le circuit fautif (trait pointillé) (stimulus 3 Fig. 5.19). Pour la détection de faute sur l'intermodulation d'ordre 3, il faut être capable de mesurer la puissance de l'harmonique représentant l'ordre 3 afin de pouvoir calculer l'IMD3 (Eq. 4.19 dans la partie §4.4.3.1. du chapitre IV). Le problème est que si la puissance du signal d'entrée est trop faible est donc que la puissance de la raie d'ordre 3 est inférieure la résolution du simulateur ou de l'appareil de mesure (Limite de sensibilité), il ne sera pas possible de déterminer l'intermodulation d'ordre 3. Pour le stimulus n°1 (Fig. 5.19), on ne peut pas mesurer les raies d'ordre 3 des circuits sains et fautifs ; il n'est donc pas possible de détecter la faute. Il faut donc que la puissance du stimuli d'entrée soit suffisante (stimulus n°3 Fig. 5.19) pour pouvoir mettre en évidence les comportements non-linéaires (Points de compression, Intermodulation d'ordres 2 et 3) afin de pouvoir détecter des fautes injectées sur ces paramètres.

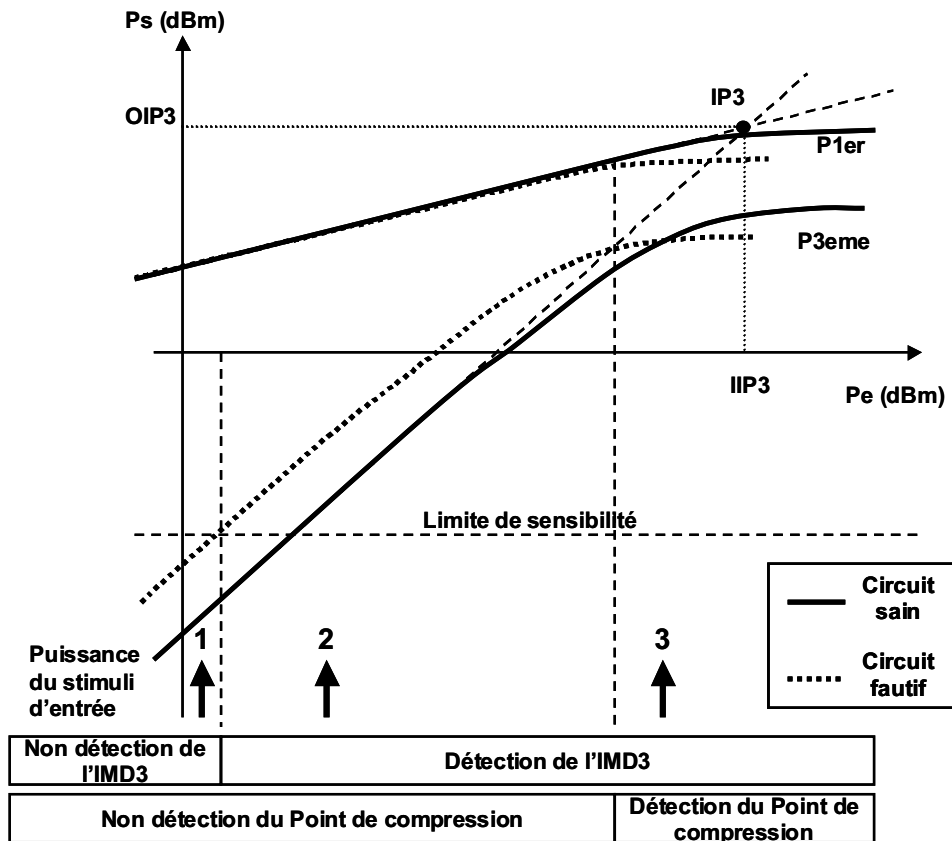


Fig. 5.19. Détection des fautes sur les intermodulations.

5.4.5.2. Taux Relatif de Couverture des Paramètres

La seconde métrique utilisée permet de qualifier plus précisément les stimuli de test car elle permet de différencier la détection si la valeur de la faute est plus ou moins éloignée des limites des spécifications. L'ensemble des simulations réalisées permet de calculer un $TRCP_{P_i}$ (Eq. 5.12) pour chaque paramètre et pour chaque stimulus (Fig. 5.6), les résultats sont présentés dans Tab. 5.8. L'analyse des différents $TRCP_{P_i}$ permet de classer les stimuli en fonction de leurs capacités à détecter de faibles variations et ainsi de conserver que les stimuli ayant le $TRCP_{P_i}$ le plus élevé. La colonne intitulée « N° Stimulus » donne le numéro du stimulus qui a permis d'obtenir le $TRCP_{P_i}$ maximal. Au final, nous obtenons un $TRCP$ moyen de 41% calculé à partir d'un ensemble de stimuli de test composé de seulement 4 stimuli. Ces résultats nous montrent qu'il est possible de réduire l'ensemble des stimuli initiaux et qu'il est nécessaire d'utiliser d'autres stimuli afin de valider la budgétisation de ce système car toutes les variations comportementales n'ont pas été détectées. Nous reviendrons sur l'intérêt de la réduction de l'ensemble des stimuli dans le chapitre suivant.

L'article [5.6] montre des résultats obtenus pour la qualification d'un autre ensemble de stimuli. La liste utilisée est issue de celle définie par les concepteurs mais a été complétée à partir des constatations faites précédemment sur la détection des fréquences de coupures et des paramètres de non-linéarité. Cet article met aussi en évidence des résultats obtenus à partir de la génération automatique de stimuli de test ; cette partie sera abordée dans le chapitre VI.

5.4.5.3. Temps de simulation

Le temps de simulation doit être considéré lors de la qualification des stimuli de test car il a un

impact sur la définition des stimuli de test et donc sur le TTM. En fait, le niveau de modélisation et le modèle de fautes ont été choisis afin d'obtenir des temps de simulation et d'évaluation des stimuli de test raisonnables, c'est-à-dire réalistes vis-à-vis des temps de conception du système. La qualification des 114 stimuli précédents a duré 38 heures sur un PC équipé d'un Pentium IV fonctionnant à 3GHz, et d'un 1Go de RAM. Ce temps de simulation reste faible devant le temps nécessaire à la conception du système qui peut prendre plusieurs mois.

Conclusion

Dans ce chapitre, une méthode d'évaluation de stimuli de test a été développée. Durant la conception des systèmes, la validation est une étape importante qui permet de s'assurer que le circuit est conçu sans erreur c'est-à-dire qu'il correspond aux spécifications définies dans le cahier des charges. Cette validation se fait généralement en utilisant des stimuli qui vont permettre de détecter les erreurs de conception. Nous proposons de mesurer la qualité de la validation de conception en évaluant les stimuli utilisés. Pour cela, nous avons mis en place une méthode d'injection et de simulation de fautes. Le principe consiste à créer volontairement des descriptions fautives et à s'assurer que les stimuli détectent un comportement différent de celui des descriptions saines. Pour cela, l'outil développé (PLASMA) permet d'automatiser les simulations et surtout de qualifier chaque stimulus en évaluant leur capacité à détecter des variations de paramètres comportementaux. Suite à la définition du modèle de fautes, plusieurs approches ont été proposées de manière à définir un sous-ensemble de descriptions saines afin de comparer leurs réponses avec les descriptions fautives devant être détectées. L'approche proposée a été appliquée sur un système RF (partie réception d'un système W-CDMA) afin de la valider. Les résultats ont permis d'évaluer la capacité des stimuli de test à détecter des fautes c'est-à-dire de déterminer le taux de couverture de fautes d'un ensemble de stimuli de test, le TRCP... et aussi d'identifier les stimuli redondants.

Bibliographie

- [5.1] A. Bounceur, S. Mir, L. Rolindez, E. Simeu, "A CAT Platform for Analog and Mixed-Signal Test Evaluation and Optimization", IEEE European Test Conference (ETC'06), pp. 217-222, May 2006.
- [5.2] R. L. Wardrop, "Normal Quantile Plots and Statistical Inference", Tutorials: University of Wisconsin-Madison, Mars 1999.
- [5.3] M. R. Spiegel, "Théorie et Applications de la statistique", Edition Série Schaum, 1991.
- [5.4] J.-M. Jolion, "Probabilités et Statistique", cours INSA, <http://rfv.insa-lyon.fr/~jolion/STAT/poly.html>, Juillet 2001.
- [5.5] E. Acar, S. Ozev, "Parametric test development for RF circuits targeting physical fault locations and using specification-based fault definitions", IEEE/ACM International conference on Computer-aided design, pp 73 – 79, November 2005.
- [5.6] Y. Joannon, V. Beroulle, C. Robach, S. Tedjini, J-L. Carbonero, "Qualification of Behavioral Level Design Validation for AMS&RF SoCs", IFIP International Conference On Veru Large Scale Integration (VLSI SOC'07), Octobre 2007.

Annexes

| Degré de liberté | Seuil de l'intervalle de confiance α | | | | | | | | | | |
|------------------|---|-------|-------|-------|-------|-------|-------|-------|---------|---------|---------|
| | 0,25 | 0,2 | 0,15 | 0,1 | 0,05 | 0,025 | 0,01 | 0,005 | 0,002 5 | 0,001 0 | 0,000 5 |
| 1 | 1 | 1,376 | 1,963 | 3,078 | 6,314 | 12,71 | 31,82 | 63,66 | 127,3 | 318,3 | 636,6 |
| 2 | 0,816 | 1,061 | 1,386 | 1,886 | 2,92 | 4,303 | 6,965 | 9,925 | 14,09 | 22,33 | 31,6 |
| 3 | 0,765 | 0,978 | 1,25 | 1,638 | 2,353 | 3,182 | 4,541 | 5,841 | 7,453 | 10,21 | 12,92 |
| 4 | 0,741 | 0,941 | 1,19 | 1,533 | 2,132 | 2,776 | 3,747 | 4,604 | 5,598 | 7,173 | 8,61 |
| 5 | 0,727 | 0,92 | 1,156 | 1,476 | 2,015 | 2,571 | 3,365 | 4,032 | 4,773 | 5,893 | 6,869 |
| 6 | 0,718 | 0,906 | 1,134 | 1,44 | 1,943 | 2,447 | 3,143 | 3,707 | 4,317 | 5,208 | 5,959 |
| 7 | 0,711 | 0,896 | 1,119 | 1,415 | 1,895 | 2,365 | 2,998 | 3,499 | 4,029 | 4,785 | 5,408 |
| 8 | 0,706 | 0,889 | 1,108 | 1,397 | 1,86 | 2,306 | 2,896 | 3,355 | 3,833 | 4,501 | 5,041 |
| 9 | 0,703 | 0,883 | 1,1 | 1,383 | 1,833 | 2,262 | 2,821 | 3,25 | 3,69 | 4,297 | 4,781 |
| 10 | 0,7 | 0,879 | 1,093 | 1,372 | 1,812 | 2,228 | 2,764 | 3,169 | 3,581 | 4,144 | 4,587 |
| 11 | 0,697 | 0,876 | 1,088 | 1,363 | 1,796 | 2,201 | 2,718 | 3,106 | 3,497 | 4,025 | 4,437 |
| 12 | 0,695 | 0,873 | 1,083 | 1,356 | 1,782 | 2,179 | 2,681 | 3,055 | 3,428 | 3,93 | 4,318 |
| 13 | 0,694 | 0,87 | 1,079 | 1,35 | 1,771 | 2,16 | 2,65 | 3,012 | 3,372 | 3,852 | 4,221 |
| 14 | 0,692 | 0,868 | 1,076 | 1,345 | 1,761 | 2,145 | 2,624 | 2,977 | 3,326 | 3,787 | 4,14 |
| 15 | 0,691 | 0,866 | 1,074 | 1,341 | 1,753 | 2,131 | 2,602 | 2,947 | 3,286 | 3,733 | 4,073 |
| 16 | 0,69 | 0,865 | 1,071 | 1,337 | 1,746 | 2,12 | 2,583 | 2,921 | 3,252 | 3,686 | 4,015 |
| 17 | 0,689 | 0,863 | 1,069 | 1,333 | 1,74 | 2,11 | 2,567 | 2,898 | 3,222 | 3,646 | 3,965 |
| 18 | 0,688 | 0,862 | 1,067 | 1,33 | 1,734 | 2,101 | 2,552 | 2,878 | 3,197 | 3,61 | 3,922 |
| 19 | 0,688 | 0,861 | 1,066 | 1,328 | 1,729 | 2,093 | 2,539 | 2,861 | 3,174 | 3,579 | 3,883 |
| 20 | 0,687 | 0,86 | 1,064 | 1,325 | 1,725 | 2,086 | 2,528 | 2,845 | 3,153 | 3,552 | 3,85 |
| 21 | 0,686 | 0,859 | 1,063 | 1,323 | 1,721 | 2,08 | 2,518 | 2,831 | 3,135 | 3,527 | 3,819 |
| 22 | 0,686 | 0,858 | 1,061 | 1,321 | 1,717 | 2,074 | 2,508 | 2,819 | 3,119 | 3,505 | 3,792 |
| 23 | 0,685 | 0,858 | 1,06 | 1,319 | 1,714 | 2,069 | 2,5 | 2,807 | 3,104 | 3,485 | 3,767 |
| 24 | 0,685 | 0,857 | 1,059 | 1,318 | 1,711 | 2,064 | 2,492 | 2,797 | 3,091 | 3,467 | 3,745 |
| 25 | 0,684 | 0,856 | 1,058 | 1,316 | 1,708 | 2,06 | 2,485 | 2,787 | 3,078 | 3,45 | 3,725 |
| 26 | 0,684 | 0,856 | 1,058 | 1,315 | 1,706 | 2,056 | 2,479 | 2,779 | 3,067 | 3,435 | 3,707 |
| 27 | 0,684 | 0,855 | 1,057 | 1,314 | 1,703 | 2,052 | 2,473 | 2,771 | 3,057 | 3,421 | 3,69 |
| 28 | 0,683 | 0,855 | 1,056 | 1,313 | 1,701 | 2,048 | 2,467 | 2,763 | 3,047 | 3,408 | 3,674 |
| 29 | 0,683 | 0,854 | 1,055 | 1,311 | 1,699 | 2,045 | 2,462 | 2,756 | 3,038 | 3,396 | 3,659 |
| 30 | 0,683 | 0,854 | 1,055 | 1,31 | 1,697 | 2,042 | 2,457 | 2,75 | 3,03 | 3,385 | 3,646 |
| 40 | 0,681 | 0,851 | 1,05 | 1,303 | 1,684 | 2,021 | 2,423 | 2,704 | 2,971 | 3,307 | 3,551 |
| 50 | 0,679 | 0,849 | 1,047 | 1,299 | 1,676 | 2,009 | 2,403 | 2,678 | 2,937 | 3,261 | 3,496 |
| 60 | 0,679 | 0,848 | 1,045 | 1,296 | 1,671 | 2 | 2,39 | 2,66 | 2,915 | 3,232 | 3,46 |
| 80 | 0,678 | 0,846 | 1,043 | 1,292 | 1,664 | 1,99 | 2,374 | 2,639 | 2,887 | 3,195 | 3,416 |
| 100 | 0,677 | 0,845 | 1,042 | 1,29 | 1,66 | 1,984 | 2,364 | 2,626 | 2,871 | 3,174 | 3,39 |
| 120 | 0,677 | 0,845 | 1,041 | 1,289 | 1,658 | 1,98 | 2,358 | 2,617 | 2,86 | 3,16 | 3,373 |
| ∞ | 0,674 | 0,842 | 1,036 | 1,282 | 1,645 | 1,96 | 2,326 | 2,576 | 2,807 | 3,09 | 3,291 |

Tab. 5.9. Tableau des valeurs du quantile suivant la loi de Student

| Degré de liberté n | Seuil de l'intervalle de confiance α | | | | | | | | | | | | |
|-----------------------|---|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| | 0,995 | 0,99 | 0,975 | 0,95 | 0,9 | 0,75 | 0,5 | 0,25 | 0,1 | 0,05 | 0,025 | 0,01 | 0,005 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0,102 | 0,455 | 1,323 | 2,706 | 3,841 | 5,024 | 6,635 | 7,879 |
| 2 | 0,01 | 0,02 | 0,051 | 0,103 | 0,211 | 0,575 | 1,386 | 2,773 | 4,605 | 5,991 | 7,378 | 9,21 | 10,597 |
| 3 | 0,072 | 0,115 | 0,216 | 0,352 | 0,584 | 1,213 | 2,366 | 4,108 | 6,251 | 7,815 | 9,348 | 11,345 | 12,838 |
| 4 | 0,207 | 0,297 | 0,484 | 0,711 | 1,064 | 1,923 | 3,357 | 5,385 | 7,779 | 9,488 | 11,143 | 13,277 | 14,86 |
| 5 | 0,412 | 0,554 | 0,831 | 1,145 | 1,61 | 2,675 | 4,351 | 6,626 | 9,236 | 11,07 | 12,833 | 15,086 | 16,75 |
| 6 | 0,676 | 0,872 | 1,237 | 1,635 | 2,204 | 3,455 | 5,348 | 7,841 | 10,645 | 12,592 | 14,449 | 16,812 | 18,548 |
| 7 | 0,989 | 1,239 | 1,69 | 2,167 | 2,833 | 4,255 | 6,346 | 9,037 | 12,017 | 14,067 | 16,013 | 18,475 | 20,278 |
| 8 | 1,344 | 1,646 | 2,18 | 2,733 | 3,49 | 5,071 | 7,344 | 10,219 | 13,362 | 15,507 | 17,535 | 20,09 | 21,955 |
| 9 | 1,735 | 2,088 | 2,7 | 3,325 | 4,168 | 5,899 | 8,343 | 11,389 | 14,684 | 16,919 | 19,023 | 21,666 | 23,589 |
| 10 | 2,156 | 2,558 | 3,247 | 3,94 | 4,865 | 6,737 | 9,342 | 12,549 | 15,987 | 18,307 | 20,483 | 23,209 | 25,188 |
| 11 | 2,603 | 3,053 | 3,816 | 4,575 | 5,578 | 7,584 | 10,341 | 13,701 | 17,275 | 19,675 | 21,92 | 24,725 | 26,757 |
| 12 | 3,074 | 3,571 | 4,404 | 5,226 | 6,304 | 8,438 | 11,34 | 14,845 | 18,549 | 21,026 | 23,337 | 26,217 | 28,3 |
| 13 | 3,565 | 4,107 | 5,009 | 5,892 | 7,042 | 9,299 | 12,34 | 15,984 | 19,812 | 22,362 | 24,736 | 27,688 | 29,819 |
| 14 | 4,075 | 4,66 | 5,629 | 6,571 | 7,79 | 10,165 | 13,339 | 17,117 | 21,064 | 23,685 | 26,119 | 29,141 | 31,319 |
| 15 | 4,601 | 5,229 | 6,262 | 7,261 | 8,547 | 11,037 | 14,339 | 18,245 | 22,307 | 24,996 | 27,488 | 30,578 | 32,801 |
| 16 | 5,142 | 5,812 | 6,908 | 7,962 | 9,312 | 11,912 | 15,338 | 19,369 | 23,542 | 26,296 | 28,845 | 32 | 34,267 |
| 17 | 5,697 | 6,408 | 7,564 | 8,672 | 10,085 | 12,792 | 16,338 | 20,489 | 24,769 | 27,587 | 30,191 | 33,409 | 35,718 |
| 18 | 6,265 | 7,015 | 8,231 | 9,39 | 10,865 | 13,675 | 17,338 | 21,605 | 25,989 | 28,869 | 31,526 | 34,805 | 37,156 |
| 19 | 6,844 | 7,633 | 8,907 | 10,117 | 11,651 | 14,562 | 18,338 | 22,718 | 27,204 | 30,144 | 32,852 | 36,191 | 38,582 |
| 20 | 7,434 | 8,26 | 9,591 | 10,851 | 12,443 | 15,452 | 19,337 | 23,828 | 28,412 | 31,41 | 34,17 | 37,566 | 39,997 |
| 21 | 8,034 | 8,897 | 10,283 | 11,591 | 13,24 | 16,344 | 20,337 | 24,935 | 29,615 | 32,671 | 35,479 | 38,932 | 41,401 |
| 22 | 8,643 | 9,542 | 10,982 | 12,338 | 14,041 | 17,24 | 21,337 | 26,039 | 30,813 | 33,924 | 36,781 | 40,289 | 42,796 |
| 23 | 9,26 | 10,196 | 11,689 | 13,091 | 14,848 | 18,137 | 22,337 | 27,141 | 32,007 | 35,172 | 38,076 | 41,638 | 44,181 |
| 24 | 9,886 | 10,856 | 12,401 | 13,848 | 15,659 | 19,037 | 23,337 | 28,241 | 33,196 | 36,415 | 39,364 | 42,98 | 45,559 |
| 25 | 10,52 | 11,524 | 13,12 | 14,611 | 16,473 | 19,939 | 24,337 | 29,339 | 34,382 | 37,652 | 40,646 | 44,314 | 46,928 |
| 26 | 11,16 | 12,198 | 13,844 | 15,379 | 17,292 | 20,843 | 25,336 | 30,435 | 35,563 | 38,885 | 41,923 | 45,642 | 48,29 |
| 27 | 11,808 | 12,879 | 14,573 | 16,151 | 18,114 | 21,749 | 26,336 | 31,528 | 36,741 | 40,113 | 43,195 | 46,963 | 49,645 |
| 28 | 12,461 | 13,565 | 15,308 | 16,928 | 18,939 | 22,657 | 27,336 | 32,62 | 37,916 | 41,337 | 44,461 | 48,278 | 50,993 |
| 29 | 13,121 | 14,256 | 16,047 | 17,708 | 19,768 | 23,567 | 28,336 | 33,711 | 39,087 | 42,557 | 45,722 | 49,588 | 52,336 |
| 30 | 13,787 | 14,953 | 16,791 | 18,493 | 20,599 | 24,478 | 29,336 | 34,8 | 40,256 | 43,773 | 46,979 | 50,892 | 53,672 |

Tab. 5.10. Tableau des valeurs du quantile suivant la loi du Khi-deux

CHAPITRE VI :

REUTILISATION DES STIMULI DE LA VALIDATION POUR LE TEST DE PRODUCTION

| | |
|--|-----|
| <i>CHAPITRE VI :</i> | 119 |
| <i>REUTILISATION DES STIMULI DE LA VALIDATION POUR LE TEST DE PRODUCTION</i> | 119 |
| 6.1. Introduction | 122 |
| 6.2. Génération de stimuli pour le test de production | 122 |
| 6.3. Application de la méthode de qualification pour le test de production | 124 |
| 6.3.1. Descriptions saines | 124 |
| 6.3.1.1. Echantillonnage pires-cas « simplifié » | 125 |
| 6.3.1.2. Echantillonnage par estimation gaussienne | 125 |
| 6.3.2. Modèle de fautes | 128 |
| 6.3.2.1. Origine des fautes | 128 |
| 6.3.2.2. Définition du type de fautes | 128 |
| 6.3.2.3. Définition du niveau d'abstraction pour l'injection de fautes paramétriques | 129 |
| 6.3.2.4. Précision du modèle de fautes | 129 |
| 6.3.3. Métrique de qualification de stimuli | 130 |
| 6.4. Définition des stimuli | 132 |
| 6.4.1. Définition des classes de stimuli | 132 |
| 6.4.2. Réutilisation des stimuli de la vérification | 132 |
| 6.5. Résultats de qualification de stimuli pour le test de production | 133 |
| 6.6. Evaluation du modèle de fautes comportementales | 136 |
| 6.6.1. Choix du modèle de fautes de référence | 136 |
| 6.6.1.1. Description du LNA | 136 |
| 6.6.1.2. Modèle de fautes « niveau composant » | 137 |
| 6.6.2. Qualification à partir d'un modèle de faute au niveau composant | 137 |
| 6.6.3. Comparaison des TRCP | 138 |
| 6.7. Définition de plages de mesures plus précises | 141 |
| 6.8. Génération de stimuli | 143 |
| 6.8.1. Perspective pour la génération : Approche déterministe | 144 |
| 6.9. Conclusion | 146 |
| Bibliographie | 146 |
| Annexes | 147 |

| | | |
|-------------------|--|------------|
| Fig. 6.1. | Approche traditionnelle du flot de conception et de vérification. | 123 |
| Fig. 6.2. | Réutilisation des stimuli de la vérification pour le test de production. | 124 |
| Fig. 6.3. | Définition des plages de mesures de la population saine. | 125 |
| Fig. 6.4. | Définition des plages de mesures des modèles sains. | 127 |
| Fig. 6.5. | Définition de la valeur du paramètre fautif pour le test de production. | 130 |
| Fig. 6.6. | Schéma électrique de l'amplificateur faible bruit (LNA). | 137 |
| Fig. 6.7. | Comparaison des TRCP. | 139 |
| Fig. 6.8. | Probabilité cumulée de l'estimation gaussienne de la population saine à partir de différents échantillons. | 142 |
| Fig. 6.9. | Erreur relative de l'estimation gaussienne pour les différents échantillons. | 142 |
| Fig. 6.10. | Principe de génération de stimuli déterministes | 145 |
| | | |
| Tab. 6.1. | Classification des stimuli en fonction du coût pour la simulation et pour le test de production. | 132 |
| Tab. 6.2. | Evaluation d'un ensemble de 120 stimuli à partir d'un modèle de fautes comportementales. | 135 |
| Tab. 6.3. | Taux Relatif de Couverture Paramétrique Structurel. | 140 |
| Tab. 6.4. | Ensemble des 114 stimuli monoton utilisés pour lorsde la qualification. | 149 |
| Tab. 6.5. | Ensemble des 6 stimuli bitons utilisés pour lors de la qualification. | 149 |
| Tab. 6.6. | Génération et qualification de stimuli pour un système W-CDMA à partir d'un modèle de fautes comportementales. | 150 |

6.1. Introduction

Le test de production est l'étape de test qui suit la fabrication des circuits électroniques. Cette étape permet de décider de rejeter un circuit considéré comme défectueux. La détermination de stimuli pour ce test, stimuli également appelés vecteurs de test, est difficile à mettre en place car elle doit permettre de tester un circuit de manière efficace tout en respectant de fortes contraintes de coût.

En fait, les capacités d'intégrations des SoCs conduisent au développement de systèmes de plus en plus complexes. Cette évolution a eu pour conséquence d'augmenter la part du coût du test dans le coût total des circuits. Ce coût est donc une contrainte majeure qui doit être considérée lors de la définition des stimuli de test. Il est en effet indispensable de réduire le nombre de stimuli nécessaires au test du circuit et de considérer principalement des stimuli nécessitant des ressources « testeur » simples et donc peu coûteuses.

La complexité des systèmes accroît aussi la difficulté pour identifier des stimuli pertinents. En conséquence, dans ce chapitre, nous proposons une approche pour évaluer les stimuli préalablement utilisés lors de la validation de conception et lors de la caractérisation des circuits. En fait, cette méthode d'évaluation offre à l'ingénieur de test un ensemble réduit de stimuli associés à leur capacité de détection. Ainsi, après avoir développé, dans le chapitre précédent, une méthode permettant d'évaluer les stimuli de test nécessaires à la validation de SoCs AMS&RF, nous proposons de réutiliser cette approche pour définir les stimuli nécessaires au test de production.

Dans ce chapitre, nous aborderons tout d'abord les enjeux de la génération de stimuli pour le test de production. Suite à cela, nous présenterons le modèle de fautes et la métrique utilisés lors de l'évaluation du test. Puis nous présenterons et validerons les résultats de qualification obtenus pour le test de la partie réception du système W-CDMA. Finalement, nous aborderons, en perspectives, la génération de stimuli de test permettant de compléter le sous-ensemble de stimuli proposé à partir des résultats de la qualification.

6.2. Génération de stimuli pour le test de production

Pour le test de production des SoCs AMS&RF, les vecteurs ou stimuli de test sont généralement difficiles à générer. Dans la plupart des cas, ces stimuli sont générés afin de vérifier les fonctions du système. Cette approche est limitée par le fait qu'il est impossible de tester de manière exhaustive toutes les configurations possibles. En fait, lors du test de systèmes complexes, les vecteurs appliqués permettent de s'assurer que les fonctions du système sont opérationnelles.

Pour remédier aux problèmes d'exhaustivité du test fonctionnel et du test des spécifications, des méthodes de test structurel ont été développées (présentées dans le chapitre III). Le test structurel consiste à identifier les stimuli de test qui permettent de détecter des fautes injectées dans le système. Plutôt que de tester toutes les fonctions ou spécifications pour toutes les configurations possibles, cette méthode propose de détecter les fautes structurelles modélisant les défauts pouvant apparaître lors de la fabrication du circuit. Ces fautes étant basées sur la structure du système et non pas sur ses fonctionnalités, il est possible de déterminer un ensemble de stimuli plus simple et moins coûteux que celui défini par le test fonctionnel ou par le test des spécifications. Toutefois, la génération des stimuli avec une approche structurelle s'avère difficile et contraignante à mettre en

place pour les systèmes AMS&RF. En effet, cette méthode basée sur l'injection de fautes à bas-niveau (niveau composant) demande l'analyse d'un nombre de fautes important et nécessite donc des temps de simulation longs. Le programme de test est alors déterminé à partir des stimuli définis dans le plan de vérification du prototype. Le schéma décrivant les étapes de conception d'un circuit (AMS&RF) introduit dans le chapitre V (Fig. 5.2) est ici complété (Fig. 6.1) avec la partie production afin de mettre en évidence la notion de TTM. Evidemment, avec une approche structurée, le programme de test ne peut être défini que lorsque le schéma structurel et le placement routage du circuit ont été entièrement validés. Comme les temps de développement et de validation de programme de test pour des systèmes AMS&RF sont longs ; ils ont donc une incidence directe sur le TTM.

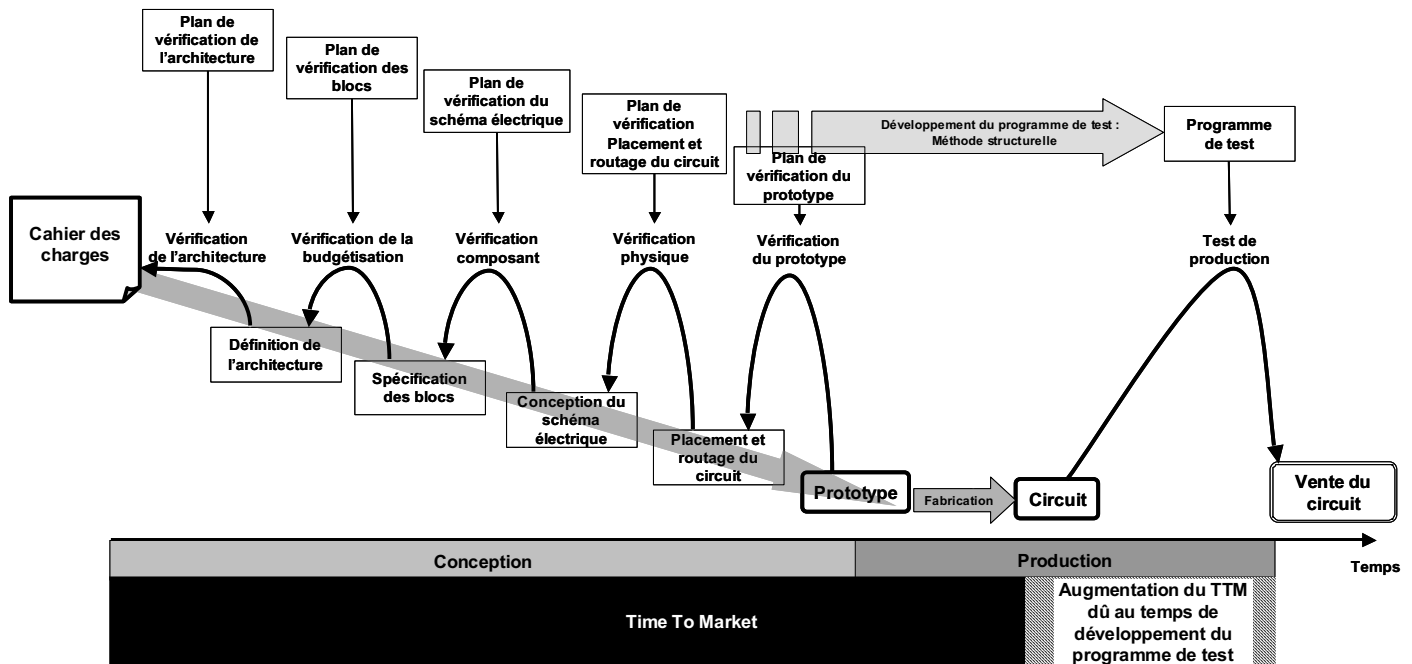


Fig. 6.1. Approche traditionnelle du flot de conception et de vérification.

L'approche développée dans ce chapitre permet de réutiliser, pour le test de production, certains stimuli qui ont été générés lors des différentes étapes de vérification du système et en particulier les stimuli générés pour vérifier la description comportementale du système. Cet ensemble de stimuli de test pourra être utilisé pour générer le programme de test final. Toutefois cet ensemble n'est pas toujours suffisant pour assurer la qualité de test voulue car les modèles utilisés (modèles du système et modèle de fautes) sont trop imprécis. Dans ce cas, cet ensemble devra donc être complété avec des approches plus bas niveau qui sera par exemple réalisée au niveau composant.

Le principe de la méthode développée est illustré par la Fig. 6.2, elle est issue de la Fig. 6.1. Dans le chapitre V, nous avons mis en évidence la nécessité de qualifier les stimuli utilisés lors de la vérification de conception afin de garantir la qualité de conception de SoCs AMS&RF. Nous proposons de réutiliser les stimuli générés lors de la conception du système pour le test de production (symbolisé par la flèche horizontale Fig. 6.2). Cette approche permet de diminuer le TTM (Fig. 6.2). La génération du programme de test consiste alors à convertir les stimuli « virtuels » en un programme de test exécutable par les équipements de test.

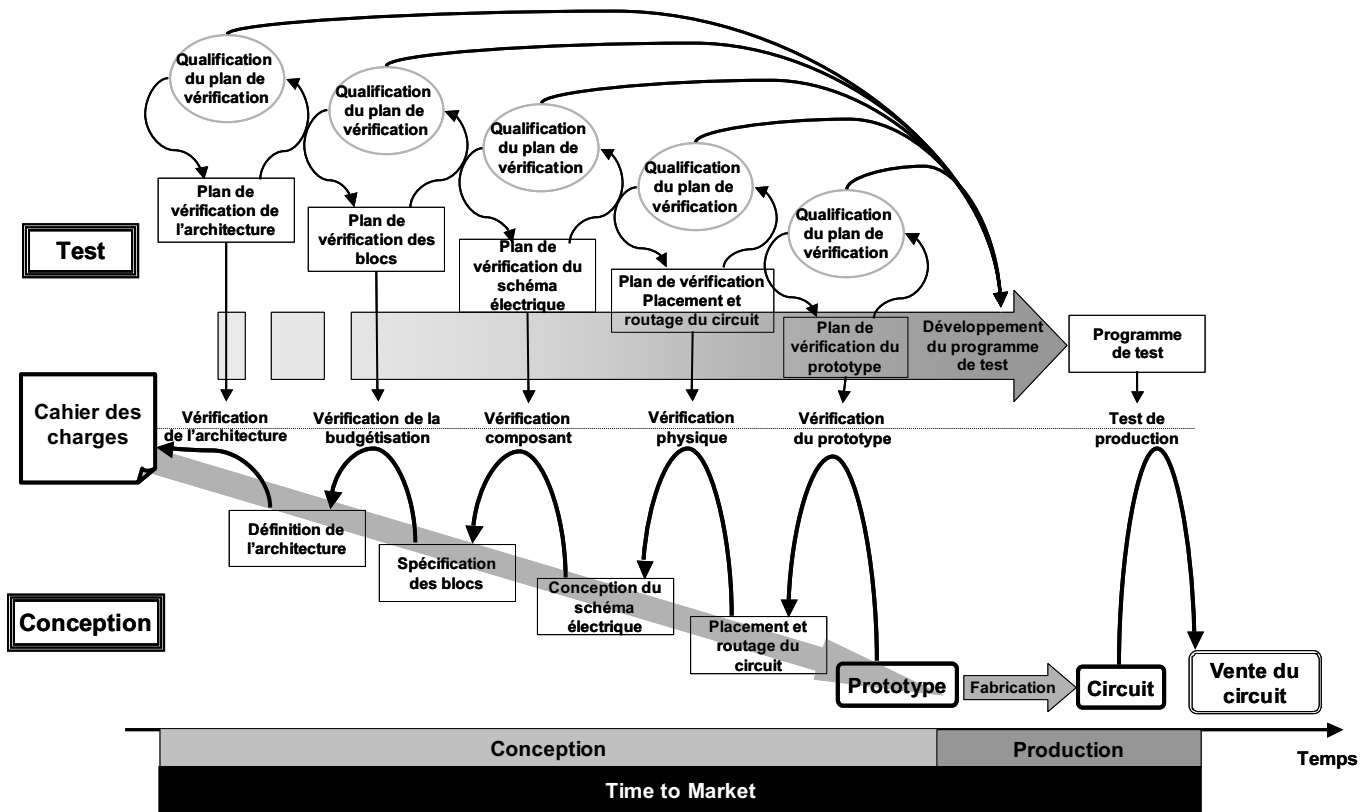


Fig. 6.2. Réutilisation des stimuli de la vérification pour le test de production.

6.3. Application de la méthode de qualification pour le test de production

Le chapitre précédent a mis en évidence l'intérêt d'utiliser une méthode de qualification de stimuli pour la validation de conception. Cette méthode permet d'identifier les stimuli redondants, et également de s'assurer que tous les paramètres importants sont couverts. Ce chapitre va montrer que cette approche de qualification peut être réutilisée pour le test de production afin de fournir à l'ingénieur de test un premier ensemble de stimuli de test. Les stimuli alors utilisés sont capables de détecter les fautes qui sont injectées au niveau comportemental ; le choix du modèle de fautes est développé dans §6.3.2. Dans la suite, nous discuterons également des éléments utiles à la qualification que nous proposons : descriptions saines, modèle de fautes et métrique. Ces éléments ont déjà été présentés dans le chapitre V mais doivent être adaptés au test de production.

6.3.1. Descriptions saines

Dans §5.3, nous avons discuté de la nécessité d'utiliser un ensemble fini de descriptions saines. La comparaison entre les réponses obtenues avec les descriptions fautives et les descriptions saines (Fig. 5.4) permet de qualifier la capacité des stimuli à détecter des fautes. Lors de la vérification, l'objectif principal est d'identifier des stimuli qui permettent de détecter des variations de paramètres hors de leur plage de spécification. Comme représenté dans la Fig. 6.2, la vérification peut être mise en place pour les différents niveaux d'abstraction. L'objectif du test de production est légèrement différent puisqu'il cherche à détecter uniquement des fautes modélisant des défauts pouvant apparaître lors de la fabrication du circuit. Il faut ainsi être capable de déterminer le nombre de fautes détectées (Couverture de fautes §3.3.3.3) et l'erreur qui est faite sur le nombre de circuits mal triés (Pertes de rendement et Taux de défauts §3.3.3.3). La définition des plages de mesures pour le test de production est directement liée aux points cités précédemment car elle va

permettre de prendre la décision de jeter ou non le circuit. La précision des plages de mesures saines obtenues par l'échantillonnage des descriptions saines est donc un critère essentiel lors de la qualification et de la génération des stimuli.

Les différentes méthodes utilisées pour définir des plages de mesures saines ont déjà été abordées lors de la qualification de stimuli pour la validation de conception (§5.3.2), mais nous allons en rediscuter ci-dessous afin d'identifier celles qui sont les mieux adaptées au test de production.

6.3.1.1. Echantillonnage pires-cas « simplifié »

L'utilisation de l'approche d'échantillonnage pires-cas « simplifié » (§5.3.2.1) est intéressante en terme de nombre de descriptions saines simulées car elle ne nécessite l'utilisation que de seulement $2 \cdot p$ descriptions (p : nombre de paramètres). D'après les résultats présentés dans §5.4.3, l'échantillonnage pires-cas permet d'obtenir des plages suffisamment grandes pour contenir la population saine totale. Cet échantillonnage est toutefois peu adapté au test de production car il ne considère pas la probabilité d'apparition des circuits sains. En effet, par cette approche, les circuits pires-cas sont considérés lors de la détection des circuits fautifs même s'ils ont une très faible probabilité d'apparaître. La Fig. 6.3 représente, par la courbe de gauche, la distribution sur une mesure de l'ensemble des circuits sains lorsque les paramètres varient dans les plages de spécifications. La courbe de droite représente la distribution des circuits fautifs contenant une faute et dont les autres paramètres du circuit varient dans leurs plages de tolérances. La Fig. 6.3 montre qu'aux limites de la distribution des circuits sains (limite supérieure dans le cas représenté), des circuits fautifs ne sont pas détectés alors que la probabilité d'avoir des circuits sains est pratiquement nulle. Il est donc nécessaire de considérer la probabilité d'apparition des circuits sains pour déterminer les plages de mesures des circuits sains et ainsi permettre la détection de circuits fautifs ayant un comportement (mesure) proches des circuits sains (Fig. 6.3).

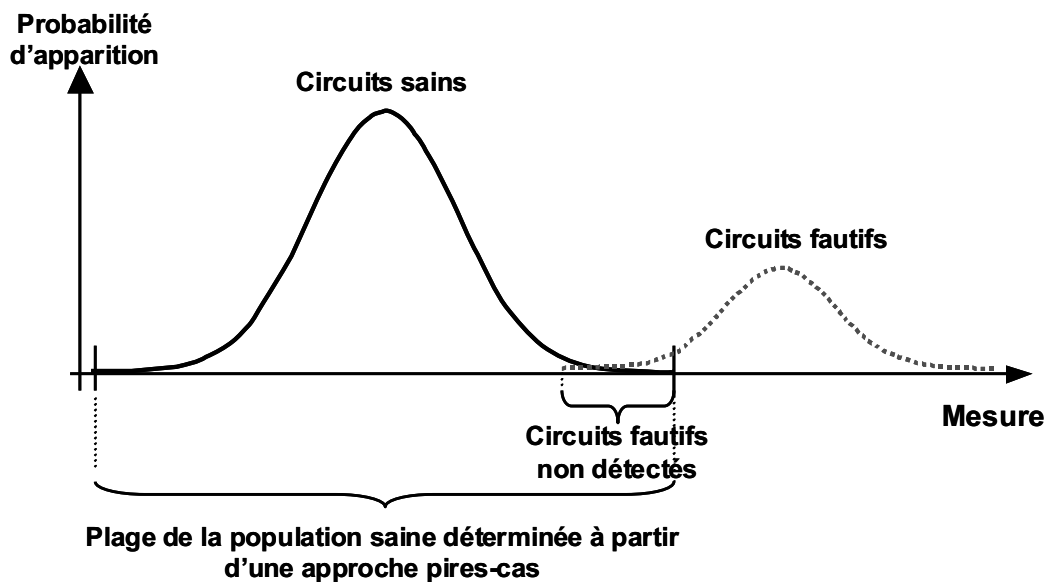


Fig. 6.3. Définition des plages de mesures de la population saine.

6.3.1.2. Echantillonnage par estimation gaussienne

L'approche d'échantillonnage de la population saine par estimation gaussienne développée dans la partie §5.3.2.2 est plus adaptée au test de production que l'approche précédente car elle permet de faire un compromis en les circuits sains rejetés et les circuits fautifs non détectés. Ainsi, en

supposant que tous les paramètres du système suivent une distribution gaussienne, elle permet d'estimer à la fois le nombre de circuits sains rejetés et le nombre de circuits défaillants non rejetés. Cette hypothèse de distribution gaussienne est souvent faite par le concepteur lors de la spécification des valeurs limites des paramètres et de la valeur du CPk qui est directement lié à l'écart type d'une distribution gaussienne (Eq. 2.1). L'échantillon obtenu par tirage aléatoire gaussien permet d'estimer l'espérance et l'écart type des réponses saines de manière suffisamment précise.

6.3.1.2.1. Estimation des plages de mesures des descriptions saines

Afin de pouvoir définir les plages de mesures des descriptions saines en fonction du nombre de circuits sains rejetés (comme présentée par la suite dans la Fig. 6.4), nous supposons que les mesures de l'ensemble des descriptions saines suivent une distribution gaussienne. L'utilisation d'un échantillon de descriptions saines et l'extrapolation des résultats de mesures permettent alors d'obtenir une estimation des mesures sur la population totale. L'hypothèse de distribution Gaussienne des mesures sera vérifiée dans la suite. La moyenne et l'écart type de chaque réponse de la population saine peuvent alors être estimés à partir de l'échantillon représentant la population saine (§5.3.2.2). Il suffit ensuite de déterminer la plage $[X_{\min}; X_{\max}]$ pour chaque mesure des descriptions saines à partir des équations (Eq. 6.1) et (Eq. 6.2) développées dans la partie §5.3.2.2 du chapitre précédent.

$$X_{\min} = m_{\text{estimé}} - \left(\frac{\sigma_{\text{estimé}}}{\sqrt{n}} t_{\alpha}^{n-1} + h \sqrt{\frac{n}{\kappa_{1-\frac{\alpha}{2}}^2 (n-1)}} \sigma_{\text{estimé}} \right), \quad (\text{Eq. 6.1})$$

où n est le nombre d'échantillons, $m_{\text{estimé}}$ est défini par (Eq. 5.7) et $\sigma_{\text{estimé}}$ par (Eq. 5.8) et h est relatif au nombre de circuits sains rejetés.

$$X_{\max} = m_{\text{estimé}} + \left(\frac{\sigma_{\text{estimé}}}{\sqrt{n}} t_{\alpha}^{n-1} + h \sqrt{\frac{n}{\kappa_{1-\frac{\alpha}{2}}^2 (n-1)}} \sigma_{\text{estimé}} \right), \quad (\text{Eq. 6.2})$$

où n est le nombre d'échantillons, $m_{\text{estimé}}$ est défini par (Eq. 5.7) et $\sigma_{\text{estimé}}$ par (Eq. 5.8) et h est relatif au nombre de circuits sains rejetés.

Dans ces deux équations (Eq. 6.1) et (Eq. 6.2), la valeur de h permet de définir le nombre maximum de descriptions saines que l'on s'autorise à rejeter (sur un million) lors du test de production (Fig. 6.4). Cette valeur spécifie à quelle distance (en écart-type σ) est positionnée la limite de chaque mesure saine, elle est donc directement liée au nombre de circuits sains qui peuvent être rejetés lors du test (Tab. 5.1).

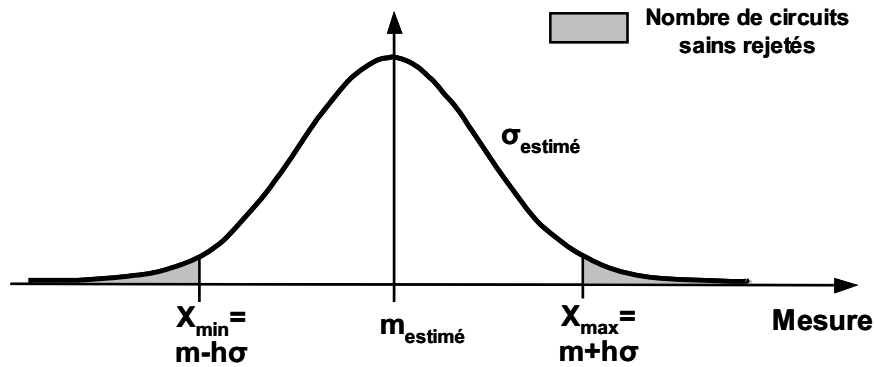


Fig. 6.4. Définition des plages de mesures des modèles sains.

6.3.1.2.2. Test du quantile normal

Comme présenté ci-dessus, les limites de détections sont définies en fonction du nombre de circuits sains que nous nous autorisons à rejeter sur un million de circuits testés. Pour convertir ce nombre en valeur de h (Eq. 6.1), nous avons supposé que la distribution suit une loi gaussienne. Or, l'incertitude engendrée par cette hypothèse n'est pas négligeable et peut se répercuter sur le nombre de circuits sains que l'on s'autorise à jeter. Ainsi, afin de réduire les temps de calcul mais en conservant un niveau de précision suffisant, nous procédons en deux phases :

Dans un premier temps, nous utilisons une population saine réduite pour la qualification de l'ensemble des stimuli de test. Cette étape permet de faire une estimation grossière mais rapide des stimuli et ainsi d'identifier les couples stimuli/mesures qui permettent de détecter au moins une faute.

Puis dans un second temps, les réponses de ces stimuli sont analysées plus finement en utilisant un échantillon plus grand pour la population saine. C'est ainsi qu'en connaissant le stimulus et la mesure ayant permis la détection d'au moins une faute, nous proposons de diminuer l'erreur de précision sur l'estimation en utilisant le test du quantile normal pour garantir la « normalité » de la mesure (§5.3.2.2.4). Cette étape ne peut être réalisée qu'après avoir défini les couples stimuli/mesures optimums car elle serait difficile à mettre en place et consommatrice en temps si elle était appliquée pour tous les stimuli et pour toutes les mesures. Le nombre de descriptions saines nécessaires à la définition des plages de mesures est déterminé en fixant l'erreur maximale acceptable sur l'estimation de la probabilité cumulée. Dans §5.4.3.2. nous avons fixé arbitrairement cette erreur à 10%. Le choix d'une valeur plus faible (5% par exemple) aura pour effet d'augmenter la précision de l'estimation gaussienne (en augmentant le nombre de descriptions saines nécessaires à l'estimation des plages de mesures) et donc d'augmenter la qualité de l'évaluation des stimuli. Les limites de détections nécessaires à la décision de rejet du circuit ainsi que le nombre de circuits sains rejetés seront alors plus précis.

La partie §6.7 reprend des résultats présentés dans [6.3] permettant d'augmenter la précision des plages de mesures saines en vérifiant que le nombre de descriptions utilisées pour estimer la plage est suffisant.

6.3.1.2.3. Détermination des pertes de rendement

D'après la Fig. 6.4, les plages de mesures saines sont déterminées à partir du nombre de circuits sains que l'on accepte de rejeter sur un million (cf. Pertes de Rendement/Yield Loss §3.3.3.3). Or

ces pertes de rendement sont définies pour un stimulus, il est donc nécessaire de les concaténer afin d'obtenir une notion de pertes de rendement « globale » c'est-à-dire liée à l'ensemble des stimuli qui seront utilisés lors du test de production. Ainsi, afin de pouvoir spécifier le nombre de circuits sains total sur un million que l'on accepte de rejeter, nous devons faire l'hypothèse que les mesures sont indépendantes. En fait, si les mesures sont indépendantes (cas le plus défavorable pour notre métrique), le nombre total de circuits sains rejetés pour l'ensemble des stimuli (Pertes de rendement) est égal à la somme des probabilités déterminées pour chaque mesure. Par contre si les mesures ne sont pas indépendantes, il faut déterminer les corrélations qu'il existe entre les différentes mesures. Cette étape complexifie la définition des plages de mesures. Suite à la qualification d'un ensemble de stimuli, les stimuli redondants ont été supprimés, cela signifie que ceux qui restent ne détectent pas exactement les mêmes fautes. On peut donc en conclure que les couples stimuli/mesures conservés sont dans la plupart des cas indépendants. Nous considérons donc que le nombre final de circuits sains rejetés sur un million est égal à la somme du nombre de circuits sains rejetés pour chaque stimulus.

6.3.2. *Modèle de fautes*

La définition du modèle de fautes est une étape essentielle de la qualification des stimuli car elle a un impact direct sur la précision de l'évaluation (§5.3.3) et sur la qualité des stimuli générés.

6.3.2.1. *Origine des fautes*

Comme on l'a vu dans le chapitre III, l'origine des fautes est différente suivant qu'il s'agit du test de production ou de la validation de conception. En effet, lors de la validation, on cherche à s'assurer que les spécifications de chaque bloc ont été déterminées sans erreur. On considère que toutes les erreurs d'écriture ou erreurs humaines sont équiprobables et ont lieu uniquement sur les paramètres spécifiés donc sur les paramètres comportementaux à vérifier. Au contraire, lors du test de production, on cherche à détecter des défauts physiques qui peuvent apparaître lors du processus de fabrication ; il s'agit donc d'erreurs qui affectent des paramètres bas-niveau. La probabilité d'apparition de ces défauts est alors liée aux capacités du processus de fabrication (§3.2.2.2).

6.3.2.2. *Définition du type de fautes*

Les fautes utilisées en test matériel pour modéliser les défauts physiques peuvent être soit des fautes catastrophiques ou soit des fautes paramétriques (§3.2.2.3).

Les variations des paramètres de fabrication sont des défauts qui peuvent apparaître lors de la production de circuits, ils sont modélisés grâce à l'utilisation de fautes paramétriques. La probabilité d'apparition de ces défauts dépend de la robustesse du processus de fabrication.

Les autres fautes généralement utilisées sont des fautes catastrophiques, elles modélisent principalement des défauts locaux. Ces fautes catastrophiques ne sont pas considérées dans notre modèle de fautes car nous supposons que dans la plupart des cas elles entraîneront un dysfonctionnement du système important donc facilement détectable à partir de stimuli capables de détecter les faibles variations paramétriques. Nous considérons donc que l'utilisation de fautes paramétriques est suffisante car une bonne couverture de fautes paramétriques permettrait de couvrir les fautes catastrophiques qui ont un effet sur le fonctionnement et les caractéristiques du circuit.

6.3.2.3. Définition du niveau d'abstraction pour l'injection de fautes paramétriques

Nous avons vu dans le chapitre III partie §3.3.2 qu'il existe de nombreux travaux sur le test matériel qui utilisent l'injection de fautes au niveau composant. L'inconvénient de cette méthode est qu'elle est difficilement applicable au test de systèmes complexes car trop coûteuse en temps de simulation. Un autre problème de l'injection de fautes au niveau composant est qu'elle engendre un nombre important de fautes car les variations doivent être injectées sur de très nombreux paramètres [6.4]. Pour une approche système, il est donc indispensable de se placer à un niveau de modélisation supérieur et ainsi de modéliser le comportement fautif haut niveau induit par un ou plusieurs défauts physiques. Pour cela, nous proposons d'utiliser le niveau comportemental. Dans notre cas, nous avons choisi de réutiliser uniquement les paramètres spécifiés lors de la conception du système car ils ont été déterminés à partir du cahier des charges et permettent alors de modéliser les comportements du système ; mais surtout parce que se sont ces spécifications qui sont disponibles dans le plan de vérification (§3.2.1).

6.3.2.4. Précision du modèle de fautes

L'utilisation d'un niveau d'abstraction supérieur est intéressante en termes de temps de simulation, mais il faut s'assurer que ce modèle de fautes est suffisant pour la qualification de stimuli pour le test de production. Nous avons choisi de reprendre le même modèle de fautes que celui décrit dans le chapitre V (§5.3.3) pour la validation de conception. Ainsi, une description fautive est obtenue en faisant varier un seul paramètre à l'extérieur de ses spécifications et en fixant tous les autres paramètres à leur valeur typique. A partir de ce modèle de fautes, il est possible de déterminer la capacité de chaque stimulus à détecter des variations des spécifications à l'extérieur de leurs plages de spécification. Le TRCP présenté dans §5.3.4 pourra alors être utilisé pour qualifier des stimuli utilisés pour le test de production (§6.3.3).

L'hypothèse simplificatrice faite sur le modèle de fautes (variation fautive unique) entraîne une perte de précision sur la qualification des stimuli et plus particulièrement sur le calcul du $TRCP_{P_i}^1$ (Eq. 5.12). Nous allons aborder ci-dessous trois cas de figures possibles permettant de voir l'impact de l'hypothèse sur la précision de la métrique.

Dans un premier cas idéal, la présence d'un défaut physique entraîne la variation d'un seul paramètre comportemental. Ainsi, mis à part l'erreur de précision Δ engendrée par l'hypothèse faite sur le modèle de fautes où tous les paramètres sont à leur valeur typique sauf le paramètre fautif (§5.3.4 et §5.4.4), le TRCP correspond exactement à la valeur de la limite de détection du paramètre fautif.

Cependant, la présence d'un défaut physique peut entraîner une variation significative de plusieurs paramètres comportementaux à la fois. Ce phénomène peut alors se traduire suivant deux cas : un cas favorable et un cas défavorable.

Dans un cas moins favorable, les variations des paramètres comportementaux peuvent se combiner et produire une variation sur une mesure qui devient plus facilement détectable c'est-à-dire pour une plus faible variation du paramètre fautif évalué (P). Lors du calcul du TRCP, nous considérons la variation d'un seul paramètre P or si pour modéliser le défaut physique précédent nous faisons varier plusieurs paramètres en même temps, la faute serait détectée pour une variation plus faible de

¹ Taux Relatif de Couverture Paramétrique du Paramètre i.

P et le TRCP serait donc plus élevé. Dans ce cas, notre métrique est pessimiste car en réalité le stimulus est capable de détecter une variation fautive plus faible que celle calculée.

Respectivement, dans un cas défavorable, il se peut qu'un défaut physique entraîne la variation de plusieurs paramètres comportementaux et ces variations peuvent, contrairement au cas précédent, se compenser. L'effet sur une mesure sera alors plus difficilement détectable. Lors du calcul du TRCP par une variation paramétrique unique sur P, un stimulus sera capable de détecter une petite variation du paramètre P alors qu'en réalité l'effet est compensé par les variations d'autres paramètres ; il faut donc que le paramètre fautif P varie plus pour être détecté. Dans ce cas, notre métrique est optimiste, le TRCP réel est donc plus faible.

Dans la partie 6.6, nous chercherons à évaluer l'erreur globale faite sur le TRCP ainsi nous déterminerons si notre métrique est pessimiste ou optimiste ([6.5] et [6.6]).

6.3.3. Métrique de qualification de stimuli

La métrique utilisée lors de la qualification et lors de la génération de stimuli pour le test de production est sensiblement la même que celle présentée dans le chapitre précédent (§5.3.4). Elle est déterminée à partir de la limite de détection (P_{lim}) (Eq. 5.12) et permet de quantifier la plus petite variation détectable. Notre modèle de fautes consiste à faire varier un paramètre à l'extérieur de ses limites de spécifications, il a été présenté dans §6.3.2.4. L'objectif est alors de déterminer la variation minimale (P_{lim}) qui peut être détectée par le stimulus évalué (Fig. 6.5). Pour chaque stimulus, il sera possible de classer les « Circuits fautifs » en « Circuits fautifs non-détectés » et « Circuits fautifs détectés ». Comme c'était le cas précédemment, une dichotomie est utilisée afin d'approcher cette limite (Fig. 6.5) et de déterminer la capacité de chaque stimulus à détecter de faibles variations fautives.

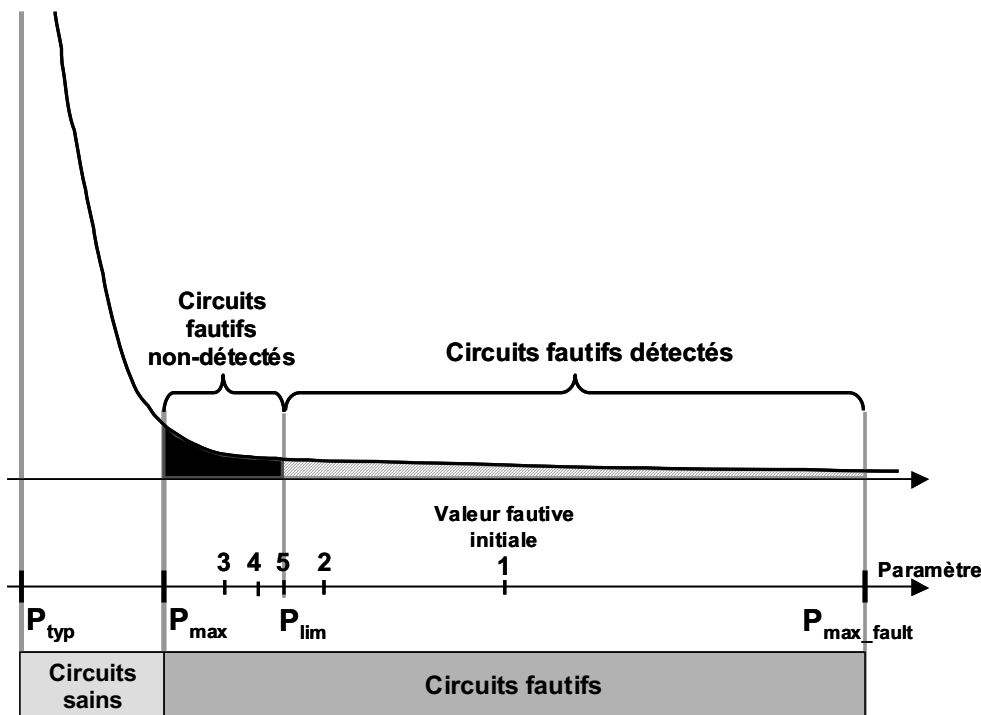


Fig. 6.5. Définition de la valeur du paramètre fautif pour le test de production.

La principale différence entre la validation de conception et le test de production vient de l'hypothèse faite sur la probabilité d'apparition des fautes (§6.3.2.1 et §6.3.2.2). Si lors de la

validation de conception, la probabilité que le concepteur définisse la valeur d'un paramètre à l'extérieure des plages de tolérance est considérée comme suivant une distribution uniforme, ce n'est plus le cas pour les défauts apparus en production. Pour cela, dans § 6.3.1, nous avons décrit l'hypothèse que nous faisons : chaque paramètre suit une distribution gaussienne. Cette distribution est spécifiée par le concepteur lors de la définition des paramètres grâce à P_{typ} , P_{min} , P_{max} et CPk . Ainsi lors de l'utilisation de fautes paramétriques pour le test de production, les probabilités d'apparitions ne sont plus uniformes mais estimées par une distribution gaussienne.

La Fig. 6.5 représente cette distribution gaussienne, seule la queue de la gaussienne est dessinée car c'est uniquement cette partie qui nous intéresse (Côté max des paramètres sur la figure, la même analyse peut être faite côté min mais n'est pas représentée). Il est alors possible de réutiliser la même métrique que pour la vérification mais en considérant une distribution gaussienne. Le $TRCP_{Pi}$ est défini par les équations suivantes :

$$TRCP_{Pi} = \frac{\text{Nombre_Circuits_fautifs_détectés}}{\text{Nombre_Circuits_fautifs}} \quad (\text{Eq. 6.3})$$

Lors de l'utilisation d'une distribution uniforme, le nombre de circuits détectés est déterminé par :

$$\text{Nombre_Circuits_fautifs_détectés} = P_{lim} - P_{max_fault} \quad (\text{Eq. 6.4})$$

ou P_{max_fault} est fixé comme dans §5.3.4.

$$\text{Nombre_Circuits_fautifs_détectés} = P_{min_fault} - P_{lim} \quad (\text{Eq. 6.5})$$

ou P_{min_fault} est fixé comme dans §5.3.4.

Mais lors de l'utilisation d'une distribution gaussienne, il faut calculer la surface hachurée (Fig. 6.5). Pour cela, il faut intégrer l'équation de la fonction gaussienne de P_{lim} à P_{max_fault} . L'équation (Eq. 6.6) permet de calculer le nombre de circuits fautifs détectés. Le même calcul peut être réalisé sur la borne min.

$$\text{Nombre_Circuits_fautifs_détectés} = \frac{1}{\sigma\sqrt{2\pi}} \int_{P_{lim}}^{P_{max_fault}} e^{-\frac{(x-P_{typ})^2}{2\sigma^2}} dx \quad (\text{Eq. 6.6})$$

$$\text{Avec } \sigma = \frac{P_{max} - P_{typ}}{3.CPK} \quad (\text{Eq. 6.7})$$

Il est alors possible de calculer le nombre de circuits fautifs à partir de (Eq. 6.6) (pour $x = P_{max}$) et enfin de déterminer le $TRCP_{Pi}$ à partir de (Eq. 6.3).

Les valeurs P_{min_fault} ou P_{max_fault} sont fixées de manière à être suffisamment éloignées de la limite de détection afin de ne pas fausser la métrique. Contrairement à ce que l'on avait observé pour la méthode appliquée à la validation de conception, à partir d'une valeur suffisamment importante, ces paramètres n'ont plus d'impact sur la métrique. En fait, comme la probabilité cumulée d'une gaussienne est bornée, la surface comprise entre P_{max_fault} et $+\infty$ (Reps. $-\infty$ et P_{min_fault}) devient faible lorsque P_{max_fault} (Reps. P_{min_fault}) s'éloigne de la limite des spécifications P_{max} (Resp. P_{min}). En se plaçant par exemple à 6σ , la surface est de 0,0009PPM (Tab. 5.1). Pour des distributions gaussiennes, les valeurs de P_{min_fault} et de P_{max_fault} n'ont plus guère d'incidence sur la métrique lorsqu'elles sont suffisamment éloignées de la limite des spécifications.

6.4. Définition des stimuli

6.4.1. Définition des classes de stimuli

Cette partie a pour objectif de définir les caractéristiques des stimuli utilisés pour le test de production. En effet, les contraintes considérées lors de la génération de stimuli pour le test de production sont différentes de celles de la validation. Ainsi, lors de la validation, le concepteur veut s'assurer que le système développé correspond rigoureusement à toutes les spécifications. Pour cela, il est prêt à réaliser un nombre important de simulations, ou de mesures dans le cas de la caractérisation d'un prototype, même si elles nécessitent des temps d'exécution importants. Pour le test de production, le coût du test est un critère majeur. Ce coût est principalement lié au coût des équipements utilisés lors du test ; il est donc indispensable de privilégier l'utilisation de stimuli simples qui nécessitent l'utilisation de ressources testeurs faible coût. Pour cela nous avons succinctement décrit dans §5.3.3.1 le fait que l'outil PLASMA utilise des classes de stimuli (Tab. 6.1). Ces classes permettent de regrouper les stimuli en fonction de leur complexité c'est-à-dire en fonction des ressources testeurs nécessaires. Par exemple, dans Tab. 6.1, un stimulus composé d'un signal RF modulé au format W-CDMA est de classe 4 car il nécessite l'utilisation d'un générateur de signaux RF modulé (RF IQ) ; ce stimulus sera très coûteux à mettre en œuvre lors du test de production. Tab. 6.1 a été défini pour le test de la partie réception du système W-CDMA. Cette classification des stimuli doit être établie en fonction du système testé et des ressources testeurs disponibles.

Le type d'analyse a aussi été spécifié dans le tableau des classes de stimuli car il a un impact direct sur le temps nécessaire à l'évaluation ou à la génération des stimuli lors de l'analyse par simulation. En effet, même si le temps nécessaire à l'évaluation et la génération des stimuli n'est pas le critère le plus important durant le test de production, il faut que ces temps soient raisonnables par rapport au temps de développement du circuit. Par exemple, dans Tab. 6.1, l'utilisation d'analyse en régime établi (SST §2.4.2.4) avec des signaux simple-ton est privilégiée car elle demande des temps de simulation largement moins importants (d'un facteur ≈ 20) que l'analyse transitoire de l'enveloppe (MODSST §2.4.2.5). Ainsi, lors de la qualification et lors de la génération, les stimuli de classe 1 seront privilégiés.

| Classe des stimuli | Simulation | | | Physique | |
|--------------------|----------------------|---------|--|--------------------------|----------------------|
| | Générateur | Analyse | Mesure | Source | Mesure |
| 1 | Simple ton | SST | Fréquentielle aux fréquences harmoniques | Générateur sinusoïdale | Analyseur de Spectre |
| 2 | Double tons | SST | Fréquentielle aux fréquences harmoniques | Générateur multi sources | Analyseur de Spectre |
| 3 | Tiple tons | SST | Fréquentielle aux fréquences harmoniques | Générateur multi sources | Analyseur de Spectre |
| 4 | Signal modulé W-CDMA | MODSST | Mesure temporelle | Modulateur RF IQ | Analyseur de Spectre |

Tab. 6.1. Classification des stimuli en fonction du coût pour la simulation et pour le test de production.

6.4.2. Réutilisation des stimuli de la vérification

Les stimuli utilisés pour le test de production sont principalement déterminés à partir de ceux utilisés lors de la validation de conception et lors de la caractérisation du système. Les stimuli utilisés lors de la caractérisation permettent de mesurer les spécifications mais aussi d'autres

caractéristiques du système certes qui ne sont pas critiques, sinon elles auraient été spécifiées dans le cahier des charges, mais qui sont intéressantes pour valider le fonctionnement du système. Elles sont donc très potentiellement capables de détecter des défauts physiques, nous choisissons donc de les utiliser pour la qualification.

Les stimuli obtenus pour la vérification et la caractérisation peuvent donc être réutilisés pour le test de production. Après avoir enlevé ceux qui ne peuvent pas être appliqués lors du test de production pour des problèmes de coût et/ou de complexité de mise en œuvre, il est nécessaire de réduire cet ensemble de stimuli afin de supprimer les stimuli redondants. La méthode de qualification de stimuli proposée dans ce chapitre permet de remplir cette tâche car elle permet de déterminer la capacité de chaque stimulus à détecter une variation fautive des paramètres du circuit à tester.

Afin de respecter les contraintes fortes en termes de coût du test de production, nous proposons d'optimiser les stimuli en identifiant ceux qui sont capables de détecter les plus faibles variations fautives et en considérant de préférence les stimuli de classe faible (§6.4.1). Ainsi, il est possible de réduire le coût du test de production en conservant uniquement les stimuli ayant permis d'obtenir les $TCPR_{Pi}$ (Eq. 6.3) maximum.

Dans le chapitre précédent (§5.4.5), nous avons qualifié un sous ensemble des stimuli utilisés lors de la validation de conception. Dans la partie qui suit, un autre sous ensemble de stimuli utilisés lors de la validation de conception est repris afin d'être qualifié pour le test de production.

6.5. Résultats de qualification de stimuli pour le test de production

Nous allons présenter dans cette partie des résultats de qualification obtenus lors de l'évaluation de stimuli pour le test de la partie réception d'un système W-CDMA. Nous montrerons alors que la méthode de qualification de stimuli pour la validation peut être réutilisée pour le test de production en procédant aux adaptations développées dans les parties précédentes (§6.3.1, §6.3.2 et §6.3.3).

Le jeu de test utilisé lors de la qualification est composé d'un ensemble de 120 stimuli (114 stimuli mono-ton et 6 stimuli bi-tons, Tab. 6.4 et Tab. 6.5 en annexes). Il s'agit d'un échantillon des stimuli utilisés lors de la validation de conception et lors de la caractérisation du circuit. L'ensemble des stimuli évalué est sensiblement identique à celui utilisé dans §5.4.5 du chapitre précédent.

Les descriptions fautives sont définies en faisant varier un paramètre à l'extérieur de ses plages de spécifications comme décrit dans §6.3.2. Pour le cas étudié (partie réception du système W-CDMA), 36 descriptions fautives sont utilisées lors de l'évaluation.

Les 50 descriptions saines utilisées pour la définition des plages de mesures sont générées à partir d'un tirage aléatoire gaussien : la valeur de chaque paramètre est déterminée à partir d'une distribution gaussienne dont la moyenne et l'écart type sont issues des valeurs fixées lors de la conception (§6.3.1.2).

L'évaluation des 120 stimuli a été effectuée en procédant à 10 itérations de dichotomie. Le nombre d'itérations a été augmenté par rapport aux résultats du chapitre précédent afin d'obtenir une limite de détection plus précise. En effet, pour la métrique basée sur une distribution gaussienne (§6.3.3), une faible variation peut entraîner une forte variation du TRCP, il faut donc que la limite de détection soit suffisamment précise. L'évaluation des 120 stimuli dans ces conditions a duré environ 40 heures sur un PC équipé d'un Pentium IV fonctionnant à 3GHz, et d'un 1Go de RAM.

La CF^2 obtenue à partir de cet ensemble de stimuli est de 91,6%, c'est-à-dire que 33 descriptions fautives ont été détectées parmi les 36 descriptions fautives définies par notre approche (Tab. 6.2).

Les $TRCP_{PCi}^3$ sont calculés en fonction des limites de détection obtenues pour chaque stimulus. Comme présenté dans §6.3.3, la métrique utilisée est différente de la métrique définie lors de la validation de conception car elle prend en compte la probabilité d'apparition des fautes. Le $TRCP_{PCi}$ est calculé en tenant compte de marges de tolérances (Valeurs typiques, minimum et maximum) et de la robustesse (CPk) spécifié lors de la conception du système (Fig. 6.5, (Eq. 6.3) et (Eq. 6.6)). Les résultats de qualification des stimuli nous montrent que certains paramètres sont détectés pour de très faibles variations à l'extérieur de leurs plages de spécifications. Par exemple, la variation fautive du paramètre S11 est détectée lorsque la valeur de la faute est fixée à -10,248dB alors que la limite de spécification est à -10,25dB (Tab. 6.2). Le $TRCP_{PCi}$ ainsi obtenu est de 99%, le stimulus évalué permet donc de détecter une très faible variation sur S11. A l'opposé, il y a des paramètres qui sont moins bien détectés. Par exemple, la fréquence de coupure du bloc LPF_I (Fc_LPF_I) ne peut être détectée que lorsque la valeur de la fréquence de coupure est inférieure à 1,27MHz alors que la limite des spécifications est à 2,6MHz (Tab. 6.2). La valeur du $TRCP_{PCi}$ alors calculée est nulle car la limite de détection est trop éloignée de la limite des spécifications. Comme la distribution utilisée pour le calcul du TRCP n'est plus uniforme mais gaussienne (§6.3.3), alors si la limite de détection est trop éloignée de la limite des spécifications le nombre de circuits fautifs détectés devient nul (aire hachurée Fig. 6.5) et le $TRCP_{PCi}$ aussi. Les raisons de cette non détection ont été exposé dans le chapitre précédent. Toutefois, même si le $TRCP_{PCi}$ calculé est nul, il est important de conserver le stimulus qui a détecté cette (importante) variation car ce stimulus pourra quand même détecter des fautes catastrophiques liées à cette spécification.

Le TRCP total obtenu en calculant la moyenne de tous les $TRCP_{PCi}$ est de 60% (Tab. 6.2). Cette valeur nous montre que l'ensemble des stimuli utilisés détecte des variations fautives mais que celles-ci restent éloignées des limites de spécifications.

² Taux de Couverture de Fautes.

³ Taux Relatif de Couverture de Paramètre Paramètre_Comportemental_i.

| Paramètre Comportementaux | Valeur limite | Détection | N° Vecteur | Valeur fautive détectée | TRCPC _i |
|---------------------------|---------------|-------------|------------|-------------------------|--------------------|
| S11 LNA | -10,25 | X | 1 | -10,248 | 0,99 |
| S22 LNA | -1,598 | | | | |
| P_comp LNA | -18,4 | X | 108 | -18,437 | 0,91 |
| IIP3 LNA | -11,2146 | X | 108 | -11,2149 | 0,99 |
| Gain IQ dem | 20 | | | | |
| Gain IQ dem | 24 | X | 1 | 24,059 | 0,75 |
| IIP3 IQ dem | -1,5 | X | 108 | -1,529 | 0,66 |
| IIP2 IQ dem | 39 | X | 47 | 5,449 | 0,00 |
| Fc LPF I | 3,60E+06 | X | 1 | 1,27E+06 | 0,00 |
| Fc LPF I | 7,90E+06 | X | 1 | 7,98E+06 | 0,49 |
| Fc LPF Q | 3,60E+06 | X | 1 | 1,27E+06 | 0,00 |
| Fc LPF Q | 7,90E+06 | X | 1 | 7,98E+06 | 0,49 |
| IMD3 VGA1 I | -107 | X | 108 | -107,21 | 0,41 |
| IMD3 VGA1 Q | -107 | X | 108 | -107,21 | 0,41 |
| Fc VGA1 I | 2,60E+06 | X | 1 | 2,60E+06 | 0,58 |
| Fc VGA1 I | 2,72E+06 | X | 1 | 2,72E+06 | 0,58 |
| Fc VGA1 Q | 2,60E+06 | X | 1 | 2,60E+06 | 0,58 |
| Fc VGA1 Q | 2,72E+06 | X | 1 | 2,72E+06 | 0,58 |
| IMD3 VGA2 I | -113 | X | 108 | -113,23 | 0,41 |
| IMD3 VGA2 Q | -113 | X | 108 | -113,23 | 0,41 |
| Fc VGA2 I | 2,82E+06 | X | 1 | 2,82E+06 | 0,58 |
| Fc VGA2 I | 2,98E+06 | X | 1 | 2,98E+06 | 0,58 |
| Fc VGA2 Q | 2,82E+06 | X | 1 | 2,82E+06 | 0,58 |
| Fc VGA2 Q | 2,98E+06 | X | 1 | 2,98E+06 | 0,58 |
| V supply | 2,5 | X | 17 | 2,497 | 0,41 |
| V supply | 2,7 | | | | |
| LNA Gain | -5,26 | X | 1 | -5,258 | 0,99 |
| LNA Gain | 5,26 | X | 1 | 5,258 | 0,99 |
| VGA1 I | -0,20 | X | 1 | -0,205 | 0,75 |
| VGA1 I | 0,20 | X | 1 | 0,205 | 0,75 |
| VGA1 Q | -0,20 | X | 1 | -0,205 | 0,75 |
| VGA1 Q | 0,20 | X | 1 | 0,205 | 0,75 |
| VGA2 I | -0,20 | X | 1 | -0,205 | 0,75 |
| VGA2 I | 0,20 | X | 1 | 0,205 | 0,75 |
| VGA2 Q | -0,20 | X | 1 | -0,205 | 0,75 |
| VGA2 Q | 0,20 | X | 1 | 0,205 | 0,75 |
| | CF | 33/36=0,916 | | TRCP | 0,60 |

Tab. 6.2. Evaluation d'un ensemble de 120 stimuli à partir d'un modèle de fautes comportementales.

Ce travail nous montre que si nous avons uniquement mesuré la couverture de fautes (>90%) de cet ensemble de stimuli, nous aurions pu conclure que ces stimuli étaient suffisants. En calculant le TRCP, nous constatons que celui-ci est de 60%, c'est-à-dire que les stimuli évalués sont capables de détecter des variations de paramètres fautifs, mais que celles-ci sont éloignées des limites des spécifications. Elles peuvent donc entraîner un nombre de circuits fautifs non détectés qui n'est pas négligeable. Il est donc nécessaire de compléter notre jeu de test en générant de nouveaux stimuli qui vont permettre de détecter de plus faibles variations fautives.

Pour le test de production, nous avons vu que le nombre de stimuli appliqués a un impact direct sur le coût du test. Il est donc nécessaire de réduire le nombre de stimuli de test en ne conservant que

les stimuli qui ont permis d'atteindre les meilleurs $TRCP_{PCi}$. Parmi les 120 stimuli évalués, l'outil de qualification PLASMA permet de mettre en évidence un sous-ensemble de 4 stimuli capables de détecter les 33 modèles fautifs pour un TRCP total de 60%.

6.6. Evaluation du modèle de fautes comportementales

L'objectif de cette partie est d'évaluer notre approche de qualification des stimuli de test. Elle permet en particulier d'évaluer la qualité du modèle de fautes utilisé. Lors de la qualification, nous avons choisi d'injecter des variations paramétriques au niveau comportemental (§6.3.2.4). Nous proposons maintenant de comparer ce modèle de fautes haut niveau avec un modèle de fautes plus bas niveau pouvant être considéré comme la référence pour cette évaluation. L'objectif sera alors de déterminer l'erreur qui est faite sur la qualification des stimuli de test.

6.6.1. Choix du modèle de fautes de référence

La première étape de cette évaluation consiste à définir un modèle de fautes représentatif des défauts physiques pouvant apparaître lors de la fabrication du circuit ; ce modèle de fautes doit donc être proche du niveau physique. Nous utiliserons et considérerons comme référence le niveau composant.

Le système est donc maintenant décrit au niveau composant : les différents paramètres utilisés sont les valeurs des résistances, inductances, capacités et les paramètres des transistors. Nous ne modélisons que l'amplificateur faible bruit (LNA⁴) au niveau composant car il n'est pas possible de simuler tout le système au niveau composant. Les autres blocs sont modélisés au niveau comportemental. Ainsi, seules les fautes sur le LNA sont considérées dans cette partie, mais la même expérience pourrait être mise en place sur n'importe quel autre bloc du système. Lorsque le LNA est modélisé au niveau composant et que les autres blocs sont modélisés au niveau comportemental, la qualification de l'ensemble des 120 stimuli nécessite 50 heures de simulations.

6.6.1.1. Description du LNA

La description du LNA au niveau composant a été développée lors de la conception du système W-CDMA. Toutes les descriptions niveau composant ont été vérifiées lors de la conception du schéma électrique (Fig. 6.2). Le schéma électrique du LNA est décrit dans la Fig. 6.6. Il est composé d'une structure classique cascode (Q1 et Q2) et d'éléments passifs permettant l'adaptation d'impédance en entrée (L3, C1 et C4) et en sortie (L5, C6 et C7).

⁴ Low Noise Amplifier.

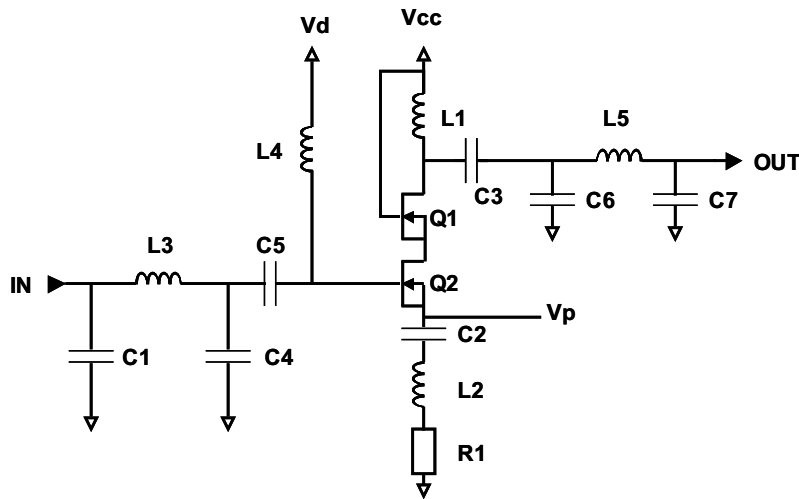


Fig. 6.6. Schéma électrique de l'amplificateur faible bruit (LNA).

6.6.1.2. Modèle de fautes « niveau composant »

Le modèle de fautes que nous choisissons d'utiliser au niveau composant est identique à notre modèle de fautes au niveau comportemental : une faute correspond à une variation d'un paramètre d'un des composants du circuit à l'extérieur de sa plage de spécification, les autres paramètres restant à leurs valeurs typiques. Nous supposons que ce modèle de fautes modélise avec suffisamment de précision des défauts physiques et peut servir de référence pour l'évaluation de la précision du TRCP_{PCi}. Il est possible de faire cette hypothèse car dans §3.3.2 décrivant le test structurel, nous avons montré que de nombreux travaux basés sur l'injection de fautes au niveau composant permettent de définir des stimuli efficaces pour le test de production. Le modèle de fautes au niveau composant consistant à faire varier individuellement chaque paramètre servira donc de référence.

L'outil PLASMA peut également être utilisé pour réaliser cette injection de fautes au niveau composant. Il permet ainsi de déterminer, pour l'ensemble des stimuli évalués, les limites de détection sur chaque paramètre des composants. Les valeurs typiques, extrêmes et le CPk nécessaires au calcul des limites de détection ont été spécifiées par les concepteurs lors de la conception du schéma électrique. Elles ont été définies en fonction de l'architecture du système et des capacités technologiques de production c'est-à-dire en tenant compte des variations du processus de fabrication.

Nous avons vu dans §5.3.1 que l'outil de simulation ADMS RF (Mentor Graphics) supporte différents niveaux de simulation (composant, comportemental) et langages (Spice, VHDL-AMS). Comme l'outil PLASMA utilise ce simulateur, il hérite de ses propriétés. Il est donc possible d'utiliser PLASMA pour réaliser cette expérience en modifiant le niveau de description du LNA.

6.6.2. Qualification à partir d'un modèle de faute au niveau composant

L'ensemble des 120 stimuli de test qualifiés au niveau comportemental dans §6.5 (Tab. 6.4 et Tab. 6.5) est maintenant qualifié à partir de fautes injectées au niveau composant. L'objectif est de montrer que les stimuli considérés comme les plus efficaces au niveau comportemental restent les mêmes au niveau composant.

Parmi les 34 fautes injectées sur le LNA décrit au niveau composant, l'ensemble des stimuli évalués

à permis de détecter 23 descriptions fautives. Le taux de couverture de fautes pour le bloc LNA est de 67% (Tab. 6.3).

Le $TRCP_{PSi}^5$ est représenté dans l'avant dernière colonne de Tab. 6.3. La valeur du TRCP moyen pour les paramètres du LNA est égale à 21%. Ce résultat montre encore une fois que l'ensemble des stimuli de test n'est pas suffisant pour permettre la détection de fautes décrites au niveau composant. Ces résultats sont principalement dus au fait que lors de la définition du niveau composant, les concepteurs ajoutent de la robustesse au circuit soit parce qu'il est difficile de créer un circuit suivant exactement les paramètres comportementaux mais surtout parce qu'ils préfèrent garder des marges de tolérances (cette notion a été présentée dans §2.3.2.2). Par précaution, ils choisissent une architecture plus robuste. Cet ajout de robustesse a pour effet de rendre la détection de variations niveau composant plus difficile et d'avoir un impact sur notre métrique.

L'utilisation d'un modèle de fautes au niveau comportemental entraîne un CF et un TRCP qui sont différents de ceux obtenus à partir d'un modèle de fautes défini au niveau composant. Par exemple, nous avons obtenu un CF de 83% (5 fautes détectées sur 6 injectées) pour les paramètres du LNA lors de l'utilisation du niveau comportemental alors que le CF est de 67% au niveau composant. En fait, ces deux résultats ne peuvent pas être comparés directement car ils correspondent à deux niveaux différents. Dans la partie suivante, nous allons présenter une méthode permettant de faire le lien entre la métrique obtenue au niveau comportementale et la métrique obtenue au niveau composant.

Les résultats intéressants de cette étude montrent que les stimuli n°1 et 108 (Tab. 6.4 et Tab. 6.5 en annexes) ont permis de détecter les plus faibles variations injectées sur les paramètres niveau composant du LNA. En reprenant les résultats présentés dans Tab. 6.2, nous constatons que ces même stimuli ont été identifiés comme les plus efficaces pour la détection des paramètres comportementaux du LNA. Ces résultats ont été obtenus lors de la qualification du même ensemble de 120 stimuli mais avec une approche comportementale.

6.6.3. Comparaison des TRCP

Pour évaluer notre approche, nous proposons de transposer les $TRCP_{PCi}$ calculés sur des paramètres comportementaux en $TRCP_{PSi_T}^6$ calculés sur des paramètres structurels (Fig. 6.7). Il sera alors possible de comparer les nouveaux $TRCP_{PSi_T}$ avec ceux obtenus lors de l'injection de fautes niveau composant $TRCP_{PSi}$ (§6.6.2 et partie droite de la Fig. 6.7).

⁵ Taux Relatif de Couverture de Paramètre Paramètre_Structurel_i

⁶ Taux Relatif de Couverture de Paramètre Paramètre_Structurel_i_Tansposé

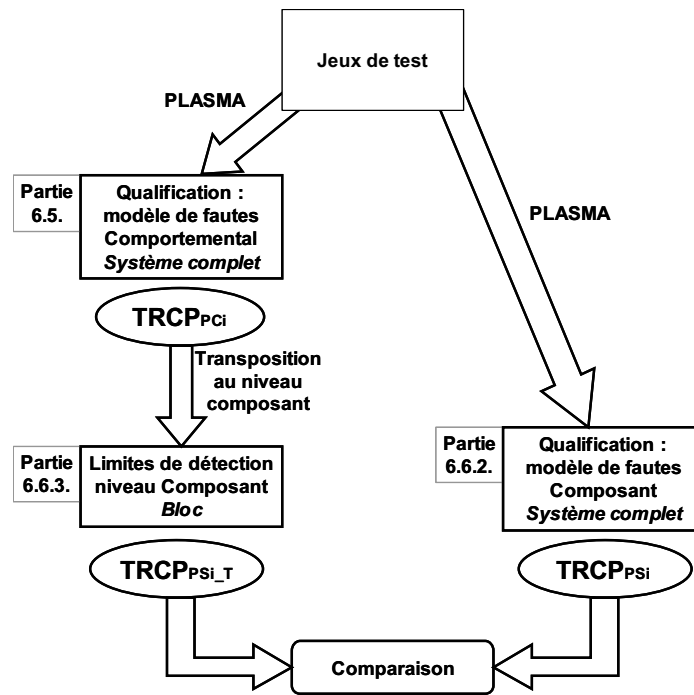


Fig. 6.7. Comparaison des TRCP.

Pour transposer le $TRCP_{Pi}$ obtenu au niveau comportemental (6.5 et partie gauche de la Fig. 6.7) vers le niveau composant, nous proposons de faire une transposition pour chaque bloc en simulant chaque bloc individuellement. Dans notre cas, la transposition a uniquement été effectuée sur le LNA. Ce bloc est décrit au niveau comportemental par cinq paramètres : gain, S11, S22, point de compression, et IIP3. D'après les résultats de la qualification présentés dans Tab. 6.2, l'ensemble des stimuli évalués a permis de détecter une variation fautive du gain inférieure à 8,6dB et supérieure à 19,12dB, de S11 à partir de -10,248dB, du point de compression 1dB à partir de -18,437dBm et de l'IIP3 à partir de -11,2149dBm. Le paramètre S22 n'a pas été détecté par l'ensemble de 120 stimuli évalués.

Afin de déterminer le $TRCP_{Psi_T}$ au niveau composant, il est maintenant nécessaire de calculer les limites de détections sur les paramètres structurels. Pour cela nous simulons le LNA seul et nous mesurons chacun des paramètres comportementaux détectés (Fig. 6.7). Les paramètres structurels sont modifiés individuellement jusqu'à ce que la limite de détection du paramètre comportementale soit atteinte. Tab. 6.3 montre les limites de détection obtenues pour chacun des paramètres structurels. Par exemple, le point de compression du LNA est à sa limite de détection (-18,437dBm, Tab. 6.2) lorsque la valeur du composant C7 est à 8,99nF. Lorsque la valeur de l'inductance L5 est à 4,427nH ; le gain du LNA est à sa limite de détection inférieure (8,6dB)...

Dans Tab. 6.3, on peut constater que pour certains paramètres comme la capacité C1 il n'a pas été possible de déterminer des limites de détection. Cela signifie que ces paramètres n'ont pas d'influence sur les paramètres comportementaux définis.

| Spécifications | | | | Modèle de fautes comportemental transposé | | | | | | | Modèle de fautes Composant | | Erreur TRCP _{PSi} | |
|----------------|---------|----------|----------|---|---------------------|---------------------|---------------------|---------------------|------------------------------|-----------------------|----------------------------|---------------------|----------------------------|--------|
| | | | | Gain inférieur | Gain supérieur | S11 | IIP1 | IIP3 | Limite de détection minimale | | | | | |
| Nom | Typique | Limite | CPk | Limite de détection | Limite de détection | Limite de détection | Limite de détection | Limite de détection | Limite de détection | TRCP _{PSi_T} | Limite de détection | TRCP _{PSi} | | |
| R1 | Min | 2,50E+00 | 1,75E+00 | 1,33 | | | | | | | | | | |
| R1 | Max | 2,50E+00 | 3,25E+00 | 1,33 | | | 2,241E+01 | | | 2,241E+01 | 0 | 8,370E+00 | 0 | |
| C1 | Min | 1,00E-19 | 9,00E-20 | 1,33 | | | | | | | | | | |
| C1 | Max | 1,00E-19 | 1,10E-19 | 1,33 | | | | | | | | | | |
| C2 | Min | 1,00E-06 | 9,00E-07 | 1,33 | | | | | | | | | | |
| C2 | Max | 1,00E-06 | 1,10E-06 | 1,33 | | | | | | | | | | |
| C3 | Min | 1,00E-12 | 9,00E-13 | 1,33 | 2,737E-14 | | | | | 2,737E-14 | 0 | 5,723E-14 | 0 | |
| C3 | Max | 1,00E-12 | 1,10E-12 | 1,33 | | | | | | | | | | |
| C4 | Min | 3,25E-13 | 2,90E-13 | 1,33 | | | 1,843E-13 | 2,506E-13 | | 2,506E-13 | 0 | 2,571E-13 | 2,99E-10 | 100,00 |
| C4 | Max | 3,25E-13 | 3,61E-13 | 1,33 | | | 3,833E-13 | | | 3,833E-13 | 1,02E-06 | 3,784E-13 | 3,45E-05 | 97,04 |
| C5 | Min | 1,00E-11 | 9,00E-12 | 1,33 | | | 5,248E-13 | | 1,000E-15 | 5,248E-13 | 0 | 1,910E-12 | 0 | |
| C5 | Max | 1,00E-11 | 1,10E-11 | 1,33 | | | | | | | | | | |
| C6 | Min | 1,40E-12 | 1,26E-12 | 1,33 | 1,253E-12 | | | | | 1,253E-12 | 0,418 | 1,253E-12 | 0,423 | 1,25 |
| C6 | Max | 1,40E-12 | 1,54E-12 | 1,33 | | 1,541E-12 | | | | 1,541E-12 | 0,921 | 1,540E-12 | 0,984 | 6,38 |
| C7 | Min | 1,00E-11 | 9,00E-12 | 1,33 | | | | 8,998E-12 | | 8,998E-12 | 0,968 | 8,998E-12 | 0,968 | 0,00 |
| C7 | Max | 1,00E-11 | 1,10E-11 | 1,33 | 1,146E-11 | | | | | 1,146E-11 | 7,88E-05 | 1,384E-11 | 0 | |
| L1 | Min | 1,00E-06 | 9,50E-07 | 1,33 | | 3,223E-08 | | | | 3,223E-08 | 0 | 5,420E-08 | 0 | |
| L1 | Max | 1,00E-06 | 1,05E-06 | 1,33 | | | | | | | | | | |
| L2 | Min | 2,50E-09 | 2,38E-09 | 1,33 | | | 1,423E-09 | 2,319E-09 | 3,784E-10 | 2,319E-09 | 1,23E-04 | 5,842E-10 | 0 | 100,00 |
| L2 | Max | 2,50E-09 | 2,63E-09 | 1,33 | 4,782E-09 | | 5,689E-09 | | | 4,782E-09 | 0 | 3,859E-09 | 0 | |
| L3 | Min | 1,06E-08 | 1,01E-08 | 1,33 | | | 7,881E-09 | 8,333E-09 | | 8,333E-09 | 0 | 7,865E-09 | 0 | |
| L3 | Max | 1,06E-08 | 1,11E-08 | 1,33 | | | 1,201E-08 | | | 1,201E-08 | 0 | 1,185E-08 | 0 | |
| L4 | Min | 1,00E-06 | 9,50E-07 | 1,33 | | | 3,572E-08 | | 1,000E-15 | 3,572E-08 | 0 | 7,178E-08 | 0 | |
| L4 | Max | 1,00E-06 | 1,05E-06 | 1,33 | | | | | | | | | | |
| L5 | Min | 4,30E-09 | 4,26E-09 | 1,33 | | | | 4,257E-09 | 4,151E-09 | 4,257E-09 | 0,968 | 4,257E-09 | 0,992 | 2,44 |
| L5 | Max | 4,30E-09 | 4,34E-09 | 1,33 | 4,427E-09 | | | | | 4,427E-09 | 0 | 4,634E-09 | 0 | |
| LM X1 | Min | 2,40E-07 | 2,16E-07 | 1,33 | | | | 2,158E-07 | | 2,158E-07 | 0,869 | 2,160E-07 | 0,984 | 11,68 |
| LM X1 | Max | 2,40E-07 | 2,67E-07 | 1,33 | | | | | | | | | | |
| WF X1 | Min | 1,00E-05 | 8,00E-06 | 1,33 | | | | 7,996E-06 | 6,813E-06 | 7,996E-06 | 0,968 | 7,998E-06 | 0,984 | 1,63 |
| WF X1 | Max | 1,00E-05 | 1,20E-05 | 1,33 | 1,376E-05 | | | | | 1,376E-05 | 9,98E-10 | 1,417E-05 | 0 | 100,00 |
| LM X2 | Min | 2,40E-07 | 2,16E-07 | 1,33 | | | | 2,160E-07 | | 2,160E-07 | 0,968 | 2,160E-07 | 0,984 | 1,63 |
| LM X2 | Max | 2,40E-07 | 2,67E-07 | 1,33 | | | | | | | | 3,161E-07 | 0 | |
| WF X2 | Min | 1,00E-05 | 8,00E-06 | 1,33 | | | 5,487E-06 | 7,996E-06 | | 7,996E-06 | 0,968 | 7,996E-06 | 0,968 | 0,00 |
| WF X2 | Max | 1,00E-05 | 1,20E-05 | 1,33 | | | | | | | | 1,410E-05 | 3,36E-12 | 100,00 |
| | | | | | | | | | | | CF | TRCP | | |
| | | | | | | | | | | | 23/34= 67% | 21% | | |

Tab. 6.3. Taux Relatif de Couverture Paramétrique Structurel.

Lors de la transposition du TRCP comportemental au TRCP niveau composant, nous conservons la variation minimale de chacun des paramètres structurels ayant été obtenue pour qu'au moins un des paramètres comportementaux soit à la limite de détection calculée dans §6.5 (cf. Colonne n°10 : « Limite de détection minimale » Tab. 6.3). La limite de détection des paramètres structurels permet alors de déterminer le TRCP_{PSi_T} qui est le transposé du niveau comportemental au niveau composant. Il est maintenant possible de comparer le TRCP_{PSi_T} obtenu à partir d'un modèle de faute haut niveau (comportemental) (cf. TRCP_{PSi_T} Tab. 6.3) avec le TRCP_{PSi} obtenu à partir d'un modèle de faute niveau composant (cf. TRCP_{PSi} Tab. 6.3).

Nous pouvons constater que pour environ un tiers des cas, aucune variation n'a été détectée sur les paramètres structurels : R1_min, C1, C2, C3_max, L1_max... Ces paramètres n'ont aucun effet sur les paramètres comportementaux définis pour la simulation du LNA seul et aucun effet sur les sorties du système lors de la simulation de tout le système (LNA décrit au niveau composant et les autres blocs décrits au niveau comportemental). Il faut tout de même noter que les variations des paramètres au niveau composant sont bornées par les valeurs P_{min_fault} et P_{max_fault} (§5.3.4) et donc par le coefficient k fixé à 30 (Eq. 5.14. et Eq. 5.15.). Une très forte variation, correspondant à une faute catastrophique, serait certainement détectable. De toute façon, si la valeur de k avait été augmentée, certes on aurait pu détecter une variation mais celle-ci aurait été tellement éloignée des limites de spécifications que la valeur du TRCP calculé aurait été nulle.

Un second cas de figure correspond à des variations qui ont été détectées, mais ces variations sont tellement larges que la valeur du TRCP calculée est proche de zéro (C4, C7_max, L2_min...) ou nulle (R1_max, C3_max, L1_min, L3, L4...). Nous ne considérerons donc pas ces valeurs car elles sont nulles.

Au final, seule une partie des paramètres structurels a pu être détectée à partir de notre ensemble de stimuli de test (Tab. 6.3 lignes des paramètres grisées). Ainsi, lors de l'interprétation des résultats, seuls les paramètres grisés seront considérés. La dernière colonne de Tab. 6.3 représente l'erreur qui est faite entre le $TRCP_{PSi_T}$ et le $TRCP_{PSi}$. Si on considère uniquement les paramètres structurels détectés (lignes grisées), l'erreur obtenue est inférieure à 10%, cela signifie que lorsque l'on utilise un modèle de faute haut niveau (niveau comportemental) et que nous cherchons à le convertir à un niveau plus communément utilisé (niveau composant), nous sommes capables de déterminer les limites des variations détectables avec une erreur raisonnable (10%).

Dans tous les cas, le $TRCP_{PSi_T}$ est inférieur au $TRCP_{PSi}$, cela signifie que lors de l'injection de fautes paramétriques comportementales uniques, nous détectons une variation plus tardivement que si nous avons utilisé un modèle de fautes défini au niveau composant entraînant plusieurs variations sur les paramètres comportementaux (§6.3.2.4). En réalité, un défaut physique entraîne la modification de plusieurs paramètres comportementaux ; dans notre cas, leurs effets s'ajoutent et rendent la faute détectable pour une plus faible variation (§6.3.2.4). Notre métrique est donc plutôt pessimiste.

6.7. Définition de plages de mesures plus précises

Après avoir montré dans §6.5 qu'il est possible de réduire le nombre de stimuli tout en conservant la même couverture de fautes et le même TRCP, nous allons vérifier que la mesure permettant la détection des circuits fautifs respecte l'hypothèse que nous avons faite sur la distribution gaussienne de cette mesure. Dans §6.3.1.2, nous avons supposé que la mesure suit une distribution gaussienne afin de réduire le nombre de descriptions saines nécessaires à la définition des plages de mesures. Pour un stimulus défini comme le plus efficace pour la détection de fautes comportementales, nous allons vérifier la normalité de la mesure et nous allons déterminer le nombre de circuits sains minimal permettant la mise en place d'une estimation gaussienne précise.

La courbe intitulée : « Descriptions saines de référence » (trait noir Fig. 6.8) représente la distribution cumulée de la population saine de référence. Cette distribution a été obtenue pour un tirage aléatoire des paramètres du système suivant une distribution gaussienne définie par les marges de tolérance et le CPk spécifiés par les concepteurs (§6.3.1.2). Cette distribution a été obtenue à partir de la simulation de 5000 descriptions saines et a durée 6 heures.

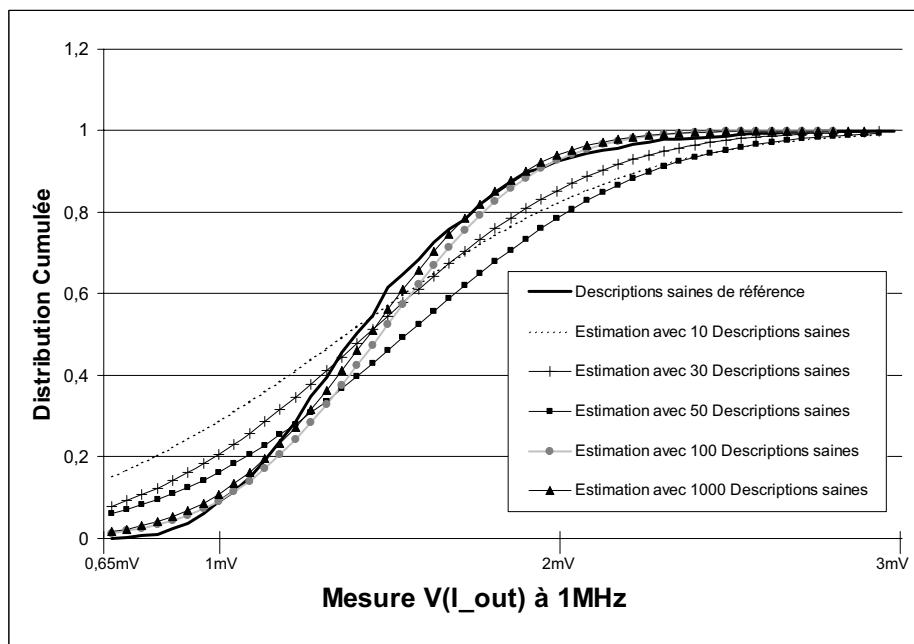


Fig. 6.8. Probabilité cumulée de l'estimation gaussienne de la population saine à partir de différents échantillons.

La distribution cumulée de référence est alors estimée en calculant la valeur moyenne et l'écart type de la population saine à partir de plusieurs échantillons. De la même façon que dans §5.4.3.2, il est possible de déterminer l'erreur qui est faite lors de l'estimation. La Fig. 6.9 représente l'erreur qui est faite sur la distribution cumulée lors de l'estimation par un échantillon. Par exemple, lorsque l'on utilise un sous-ensemble de la population saine composée de 1000 descriptions, il est possible d'estimer la distribution de la population totale avec une erreur maximale inférieure à 5% (courbe avec triangles Fig. 6.9).

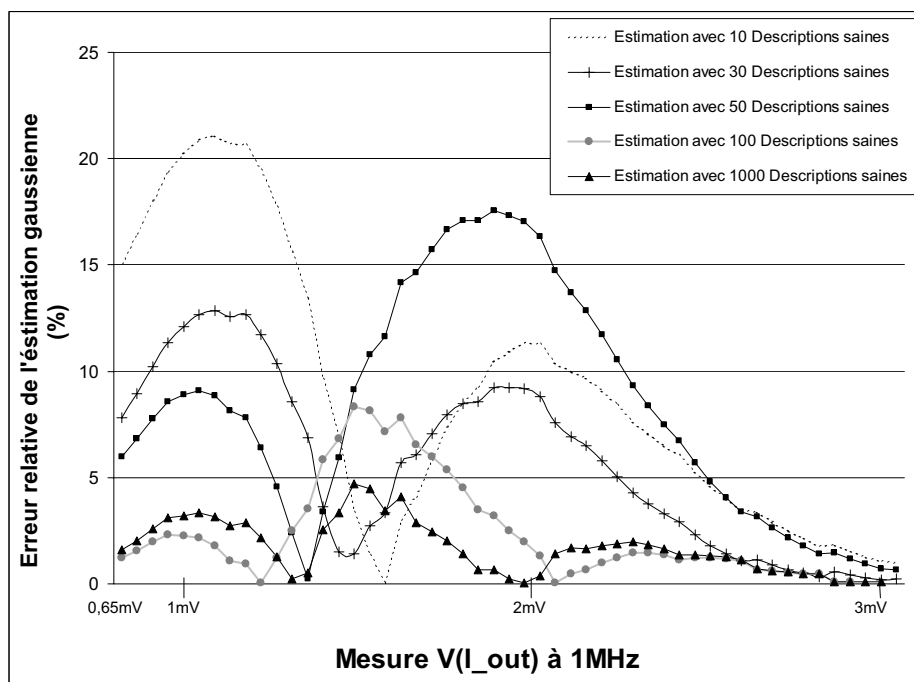


Fig. 6.9. Erreur relative de l'estimation gaussienne pour les différents échantillons.

Les valeurs de la moyenne et de l'écart type obtenues à partir de 1000 descriptions saines pour ce couple stimulus/mesure sont respectivement 1,48mV et 0,379mV. D'après les équations (Eq. 6.1) et

(Eq. 6.2), en prenant $t_{n-1}^{\alpha}=2,576$ (Tab. 5.9.), $\kappa_{1-\alpha/2}^2=18,493$ (Tab. 5.10.) et $h=4$ (Fig. 6.4 et Fig. 5.10.), la tension minimum de la plage des descriptions saines est 1.096mV et la tension maximum est 1.863mV.

Ainsi, afin d'obtenir une précision suffisante sur les plages de mesures des circuits sains, il est nécessaire d'utiliser 1000 descriptions saines. L'évaluation d'un stimulus à partir de 1000 descriptions saines a duré environ 1 heure 30, cette évaluation ne peut donc être appliquée que sur un nombre réduit de stimuli. Il est alors indispensable de faire une première évaluation avec un nombre réduit de circuits sains afin d'identifier les stimuli capables de détecter de faibles variations fautives puis d'évaluer de nouveau les stimuli avec un nombre de descriptions saines plus important. Ainsi dans notre cas (§6.5) il faut re-évaluer 4 stimuli afin d'obtenir des plages de mesures précises. Il faut donc 6 heures pour déterminer les plages de mesures des circuits sains nécessaires pour décider du rejet des circuits testés.

Nous avons choisi de fixer la valeur de h à 4 car d'après Tab. 5.1. cela signifie que l'on accepte de rejeter 31 circuits sains sur un million pour chaque stimuli. Comme les résultats de §6.5 ont permis de conserver 4 couples stimuli/mesures, si nous supposons que ces couples sont indépendants (hypothèse faite dans §6.3.1.2) l'utilisation de cet ensemble réduit rejettera au maximum 124 circuits sains sur un million.

6.8. Génération de stimuli

Nous avons vu dans §6.4 que la première étape permettant la définition de stimuli pour le test de production consiste à réutiliser les stimuli appliqués lors de la validation de conception et lors de la caractérisation du circuit. Cet ensemble de stimuli permet de détecter certains défauts (§6.6), mais n'est pas suffisant. Il est donc nécessaire de compléter cet ensemble de stimuli afin de permettre un test de production plus efficace c'est-à-dire capable de détecter des défauts physiques en considérant la couverture des spécifications. Pour cela, nous proposons de développer des méthodes de génération de stimuli de test capables de compléter les stimuli de test qualifiés précédemment (§6.3 et §6.5).

Lors de la mise en place d'une méthode de génération de stimuli, il faut tout d'abord identifier les paramètres permettant de décrire ces stimuli. Dans §3.3.3, nous avons défini la notion de stimulus. Un stimulus est composé des paramètres permettant de configurer le circuit (gain, offset) et des signaux appliqués sur les entrées du circuit. Un ensemble des signaux pouvant être appliqués au système étudié est présenté dans Tab. 6.1. Les principaux signaux pouvant être appliqués sont de type : sinusoïdal, continu, créneaux, modulé... Leurs caractéristiques doivent être spécifiées. Par exemple, dans le cas de la définition de stimuli composés des signaux mono-ton (Classe 1 Tab. 6.1), il est nécessaire de spécifier la fréquence, l'amplitude et la phase du signal. Les plages de ces paramètres sont déterminées en fonction des ressources testeurs mises à la disposition de l'ingénieur de test mais aussi en fonction des plages de fonctionnement du circuit sous test.

Pour permettre la détection des circuits fautifs, il faut aussi définir la mesure à effectuer. Dans notre cas, les mesures effectuées sur les circuits dépendent essentiellement des stimuli appliqués ; Tab. 6.1 montre les relations entre les sources et les mesures. Par exemple, nous mesurerons la puissance du signal aux différentes fréquences des harmoniques lors de l'utilisation de sources multi-tons (Classe 1 à 3 Tab. 6.1).

Après avoir identifié les différents paramètres définissant le stimulus, il est nécessaire de déterminer les valeurs des paramètres des signaux afin de permettre la génération des stimuli. Pour cela, il existe plusieurs approches de génération automatique. L'approche la plus simple consiste à définir les valeurs des paramètres en procédant par tirage aléatoire uniforme. Cette approche est la plus simple à mettre en place, mais elle nécessite des temps de génération longs. En effet, comme les stimuli sont générés aléatoirement il faut en qualifier un nombre important avant de trouver un stimulus efficace. Cette approche a donc un impact sur le temps de génération de stimuli. Mais comme au final seul les stimuli ayant permis d'obtenir les meilleurs $TRCP_{Pi}$ seront conservés pour le test de production, l'utilisation de cette approche n'aura pas d'effet sur le temps de test de production.

Tab. 6.6 (en annexes) représente les résultats de qualification de stimuli de test générés automatiquement pour le test de la partie réception du système W-CDMA. Lors de cette étape, 41 stimuli ont été générés aléatoirement et leur évaluation a nécessité 21 heures de simulation. Ces résultats montrent que la totalité des 36 fautes injectées a été détectée (Tab. 6.6 en annexes). On peut constater que les paramètres qui n'avaient pas été détectés dans §6.5 : S22, Gain_IQ_dem, IIP2_IQ_dem, Fc_LPF_I, Fc_LPF_Q et V_supply ont été détectés mais pour une variation fautive importante. En effet, leurs $TRCP_{Pi}$ alors calculés sont pratiquement nuls (Tab. 6.6). Finalement, on constate que cet ensemble de stimuli généré de manière aléatoire est plus efficace pour la détection de variations paramétriques comportementales que l'ensemble de 120 stimuli utilisé précédemment car il permet d'obtenir un TRCP total de 79% contre 60% dans §6.5. Les stimuli générés aléatoirement permettent de détecter plus de fautes car la puissance des signaux utilisée est légèrement plus élevée. En effet, le générateur aléatoire détermine des valeurs à partir de la plage de fonctionnement du circuit. Il va donc générer des stimuli de puissances faibles, mais aussi de puissances élevées. Dans le second cas, le système se trouve en limite de saturation, il est alors possible de mettre en évidence des comportements non-linéaires (§5.4.5.1). De plus, les signaux d'entrées utilisés lors de la vérification ont une différence de fréquence constante ($F_{RF}-F_{OL}=1\text{MHz}$ §5.4.5.1) ; comme le générateur aléatoire va déterminer des stimuli ayant des fréquences variables, il va ainsi permettre la détection de fautes injectées sur les fréquences de coupures.

D'après le tableau de résultats (Tab. 6.6 en annexes) certains paramètres sont détectés mais pour une valeur du paramètre fautif très éloignée des limites de spécifications (TRCP pratiquement nul), il faudrait donc s'intéresser particulièrement à leur détection afin de détecter une variation fautive plus faible. Nous allons aborder dans la partie qui suit une approche qui permet de traiter ce problème.

6.8.1. Perspective pour la génération : Approche déterministe

Afin de diminuer le nombre de stimuli à évaluer par rapport à l'approche de génération aléatoire et surtout afin d'orienter l'outil vers la détection de fautes particulières, il est possible d'utiliser un algorithme de génération déterministe.

Le principe des algorithmes déterministes consiste à faire varier une entrée (E_d Fig. 6.10) (dans notre cas : un paramètre du stimulus) et d'observer une sortie (S_d Fig. 6.10) (dans notre cas : une mesure) afin de voir comment S_d varie en fonction de E_d . L'objectif est de trouver E_d qui permet d'obtenir une variation sur une mesure S_d afin que la détection de la description fautive puisse être

effective c'est-à-dire que la mesure fautive soit à l'extérieur de la plage des mesures saines.

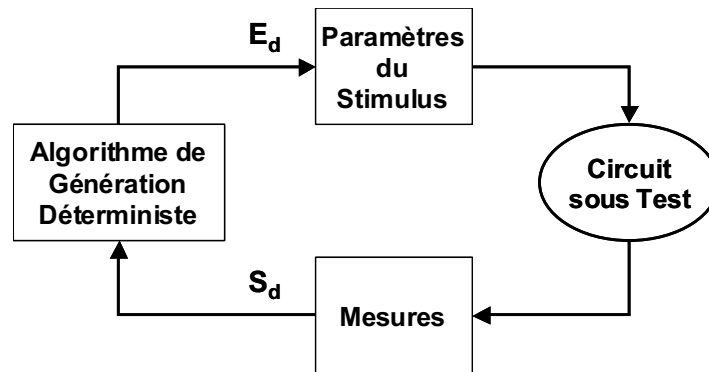


Fig. 6.10. Principe de génération de stimuli déterministes

Par la suite, nous allons aborder les différentes notions utiles afin de mettre en évidence les particularités à prendre en compte lors de l'implémentation d'une approche déterministe.

La première étape permettant la génération déterministe consiste à identifier la description fautive que l'algorithme doit chercher à détecter. Lorsque l'on a identifié une faute qui doit absolument être détectée ou lorsqu'il ne reste plus qu'une description fautive à détecter, il n'y a pas de problème. Mais dans le cas contraire, il est nécessaire de faire un choix. Ce choix peut rapidement être fait en considérant en priorité la faute ayant la plus forte probabilité d'apparaître. Pour cela il suffit d'utiliser les caractéristiques de robustesse définies par les concepteurs et plus précisément d'identifier le paramètre dont la valeur du CPk est la plus faible car une variation fautive sur ce paramètre aura la plus forte probabilité d'apparaître.

Une des difficultés consiste à déterminer la mesure (S_d) pour laquelle on va chercher à faire sortir la réponse fautive de la plage des mesures saines. Lors de la qualification d'un stimulus, on utilise toutes les mesures qui peuvent être faites (Choix des terminaux à mesurer : entrées, sorties ; Type de mesure : à une fréquence spécifique...) afin d'identifier la détection de descriptions fautives. Lors de l'utilisation d'un algorithme déterministe, il faut déterminer une sortie (S_d) à considérer pour permettre la mise en place de la génération déterministe (Il n'est pas possible de considérer toutes les sorties car elles peuvent présenter un comportement différent et dans ce cas il ne sera pas possible de déterminer comment le paramètre d'entrée E_d doit varier).

Pour finir, il est nécessaire de déterminer comment varient les E_d : linéairement/exponentiel, croissant/décroissant, dichotomie... car ce choix va avoir un impact sur le temps de convergence. Il faut aussi tenir compte des corrélations qui peuvent exister entre les différents paramètres des sources.

Comme nous avons pu le montrer, la génération automatique de stimuli par une approche déterministe fait appel à plusieurs particularités qui ne sont pas triviales. Dans cette section, nous avons abordé uniquement les perspectives et les points à résoudre dans un futur travail. La génération déterministe a été implémentée dans l'outil PLASMA mais n'a pas été développée dans cette thèse car elle doit être améliorée afin de permettre une génération plus optimale. La finalité est ainsi de proposer aux ingénieurs de test, un outil PLASMA permettant non seulement la qualification de stimuli de test déterminés lors d'une étape précédente, mais aussi la possibilité de compléter automatiquement l'ensemble de stimuli de test proposés.

6.9. Conclusion

Afin de développer un outil permettant d'assister les ingénieurs de test à choisir des stimuli pour le test de production, nous avons proposé dans ce chapitre de réutiliser l'évaluation des stimuli initialement développée pour la qualification des stimuli utilisés pour la validation de conception. Après avoir exposé le concept général de la méthode développée, nous avons tout d'abord présenté une méthode permettant de réduire l'échantillon de la population saine. Nous avons ensuite montré que cette méthode d'évaluation des stimuli est adaptée au test de production. En particulier, la vérification de l'hypothèse de normalité a été faite sur les mesures. Le choix du modèle de fautes a ensuite été présenté et validé pour une approche de test de production en montrant, pour un cas donné, que les résultats obtenus étaient sensiblement identiques à ceux obtenus lors d'une approche d'injection de fautes à bas niveau (niveau composant). Finalement, nous avons abordé une méthode de génération automatique de stimuli de test basée sur la génération aléatoire. Cette méthode a permis d'obtenir de bons résultats mais pourra être complétée par l'utilisation de méthodes de génération déterministe. Ce second point se présente comme un développement complémentaire qui pourra être réalisé pour compléter nos résultats de recherche. Il permettra de fournir à l'ingénieur de test un outil encore plus performant, capable de qualifier des stimuli de test préalablement utilisés lors de la conception du circuit mais aussi capable de compléter cet ensemble en proposant de nouveaux stimuli tout en considérant toujours la contrainte du coût du test.

Bibliographie

- [6.1] A. Khouas, "Simulation de Fautes et Optimisation des Tests de Production pour les Circuits Analogiques avec Prise en Compte des Tolérances", Thèse de doctorat, Université de Paris VI, 2000.
- [6.2] J.-M. Jolion, "Probabilités et Statistique", <http://rfv.insa-lyon.fr/~jolion/STAT/poly.html>, Juillet 2001.
- [6.3] Y. Joannon, V. Beroulle, C. Robach, S. Tedjini, J-L. Carbonero, "Decreasing Test Qualification Time in AMS & RF Systems", IEEE Design & Test of Computers, Special Issue on Design and Test of RFIC Chips, Février 2008.
- [6.4] E. Acar, S. Ozev, "Parametric test development for RF circuits targeting physical fault locations and using specification-based fault definitions", IEEE/ACM International conference on Computer-aided design, pp 73 – 79, November 2005.
- [6.5] Y. Joannon, V. Beroulle, C. Robach, S. Tedjini, J-L. Carbonero, "Using of Behavioral level AMS&RF Simulation for Validation Test Set Optimization", IEEE International Mixed-Signals Testing Workshop, pp 66 - 71, Juin 2007.
- [6.6] Y. Joannon, V. Beroulle, C. Robach, S. Tedjini, J-L. Carbonero, "Choice of a high level fault model for the Optimization of Validation Test Set reused for Manufacturing Test", IEEE Wireless Test Workshop, pp 62 - 67, Mai 2007.

Annexes

| Stimulus n° | Configuration du système | | | | | Signaux d'entrées | | | | | |
|----------------|--------------------------|---------------------|---------------------|---------------------|---------------------|-------------------|--------------|--------------------|----------------|--------------|--------------------|
| | | | | | | Entrée RF | | | Entrée OL | | |
| | Gain LNA (dB) | Gain VGA1_I (dB) | Gain VGA1_Q (dB) | Gain VGA2_I (dB) | Gain VGA2_Q (dB) | Tension (V) | Phase (°) | Fréquence (MHz) | Tension (V) | Phase (°) | Fréquence (MHz) |
| 1 | 15 | -15 | -15 | 23 | 23 | 2,24E-05 | 0 | 2168,6 | 0,1 | 0 | 2167,6 |
| 2 | 15 | -13 | -13 | 23 | 23 | 2,24E-05 | 0 | 2168,6 | 0,1 | 0 | 2167,6 |
| 3 | 15 | -11 | -11 | 23 | 23 | 2,24E-05 | 0 | 2168,6 | 0,1 | 0 | 2167,6 |
| 4 | 15 | -9 | -9 | 23 | 23 | 2,24E-05 | 0 | 2168,6 | 0,1 | 0 | 2167,6 |
| 5 | 15 | -7 | -7 | 23 | 23 | 2,24E-05 | 0 | 2168,6 | 0,1 | 0 | 2167,6 |
| 6 | 15 | -5 | -5 | 23 | 23 | 2,24E-05 | 0 | 2168,6 | 0,1 | 0 | 2167,6 |
| 7 | 15 | -3 | -3 | 23 | 23 | 2,24E-05 | 0 | 2168,6 | 0,1 | 0 | 2167,6 |
| 8 | 15 | -1 | -1 | 23 | 23 | 2,24E-05 | 0 | 2168,6 | 0,1 | 0 | 2167,6 |
| 9 | 15 | 1 | 1 | 23 | 23 | 2,24E-05 | 0 | 2168,6 | 0,1 | 0 | 2167,6 |
| 10 | 15 | 3 | 3 | 23 | 23 | 2,24E-05 | 0 | 2168,6 | 0,1 | 0 | 2167,6 |
| 11 | 15 | 5 | 5 | 23 | 23 | 2,24E-05 | 0 | 2168,6 | 0,1 | 0 | 2167,6 |
| 12 | 15 | 7 | 7 | 23 | 23 | 2,24E-05 | 0 | 2168,6 | 0,1 | 0 | 2167,6 |
| 13 | 15 | 9 | 9 | 23 | 23 | 2,24E-05 | 0 | 2168,6 | 0,1 | 0 | 2167,6 |
| 14 | 15 | 11 | 11 | 23 | 23 | 2,24E-05 | 0 | 2168,6 | 0,1 | 0 | 2167,6 |
| 15 | 15 | 13 | 13 | 23 | 23 | 2,24E-05 | 0 | 2168,6 | 0,1 | 0 | 2167,6 |
| 16 | 15 | 15 | 15 | 23 | 23 | 2,24E-05 | 0 | 2168,6 | 0,1 | 0 | 2167,6 |
| 17 | -15 | -15 | -15 | 23 | 23 | 2,24E-04 | 0 | 2168,6 | 0,1 | 0 | 2167,6 |
| 18 | -15 | -13 | -13 | 23 | 23 | 2,24E-04 | 0 | 2168,6 | 0,1 | 0 | 2167,6 |
| 19 | -15 | -11 | -11 | 23 | 23 | 2,24E-04 | 0 | 2168,6 | 0,1 | 0 | 2167,6 |
| 20 | -15 | -9 | -9 | 23 | 23 | 2,24E-04 | 0 | 2168,6 | 0,1 | 0 | 2167,6 |
| 21 | -15 | -7 | -7 | 23 | 23 | 2,24E-04 | 0 | 2168,6 | 0,1 | 0 | 2167,6 |
| 22 | -15 | -5 | -5 | 23 | 23 | 2,24E-04 | 0 | 2168,6 | 0,1 | 0 | 2167,6 |
| 23 | -15 | -3 | -3 | 23 | 23 | 2,24E-04 | 0 | 2168,6 | 0,1 | 0 | 2167,6 |
| 24 | -15 | -1 | -1 | 23 | 23 | 2,24E-04 | 0 | 2168,6 | 0,1 | 0 | 2167,6 |
| 25 | -15 | 1 | 1 | 23 | 23 | 2,24E-04 | 0 | 2168,6 | 0,1 | 0 | 2167,6 |
| 26 | -15 | 3 | 3 | 23 | 23 | 2,24E-04 | 0 | 2168,6 | 0,1 | 0 | 2167,6 |
| 27 | -15 | 5 | 5 | 23 | 23 | 2,24E-04 | 0 | 2168,6 | 0,1 | 0 | 2167,6 |
| 28 | -15 | 7 | 7 | 23 | 23 | 2,24E-04 | 0 | 2168,6 | 0,1 | 0 | 2167,6 |
| 29 | -15 | 9 | 9 | 23 | 23 | 2,24E-04 | 0 | 2168,6 | 0,1 | 0 | 2167,6 |
| 30 | -15 | 11 | 11 | 23 | 23 | 2,24E-04 | 0 | 2168,6 | 0,1 | 0 | 2167,6 |
| 31 | -15 | 13 | 13 | 23 | 23 | 2,24E-04 | 0 | 2168,6 | 0,1 | 0 | 2167,6 |
| 32 | -15 | 15 | 15 | 23 | 23 | 2,24E-04 | 0 | 2168,6 | 0,1 | 0 | 2167,6 |
| 33 | 15 | 15 | 15 | -7 | -7 | 2,24E-05 | 0 | 2113,4 | 0,1 | 0 | 2112,4 |
| 34 | 15 | 15 | 15 | -5 | -5 | 2,24E-05 | 0 | 2113,4 | 0,1 | 0 | 2112,4 |
| 35 | 15 | 15 | 15 | -3 | -3 | 2,24E-05 | 0 | 2113,4 | 0,1 | 0 | 2112,4 |
| 36 | 15 | 15 | 15 | -1 | -1 | 2,24E-05 | 0 | 2113,4 | 0,1 | 0 | 2112,4 |
| 37 | 15 | 15 | 15 | 1 | 1 | 2,24E-05 | 0 | 2113,4 | 0,1 | 0 | 2112,4 |
| 38 | 15 | 15 | 15 | 3 | 3 | 2,24E-05 | 0 | 2113,4 | 0,1 | 0 | 2112,4 |
| 39 | 15 | 15 | 15 | 5 | 5 | 2,24E-05 | 0 | 2113,4 | 0,1 | 0 | 2112,4 |
| 40 | 15 | 15 | 15 | 7 | 7 | 2,24E-05 | 0 | 2113,4 | 0,1 | 0 | 2112,4 |
| 41 | 15 | 15 | 15 | 9 | 9 | 2,24E-05 | 0 | 2113,4 | 0,1 | 0 | 2112,4 |
| 42 | 15 | 15 | 15 | 11 | 11 | 2,24E-05 | 0 | 2113,4 | 0,1 | 0 | 2112,4 |
| 43 | 15 | 15 | 15 | 13 | 13 | 2,24E-05 | 0 | 2113,4 | 0,1 | 0 | 2112,4 |
| 44 | 15 | 15 | 15 | 15 | 15 | 2,24E-05 | 0 | 2113,4 | 0,1 | 0 | 2112,4 |
| 45 | 15 | 15 | 15 | 17 | 17 | 2,24E-05 | 0 | 2113,4 | 0,1 | 0 | 2112,4 |
| 46 | 15 | 15 | 15 | 19 | 19 | 2,24E-05 | 0 | 2113,4 | 0,1 | 0 | 2112,4 |
| 47 | 15 | 15 | 15 | 21 | 21 | 2,24E-05 | 0 | 2113,4 | 0,1 | 0 | 2112,4 |
| 48 | 15 | 15 | 15 | 23 | 23 | 2,24E-05 | 0 | 2113,4 | 0,1 | 0 | 2112,4 |
| 49 | 15 | 15 | 15 | -7 | -7 | 2,24E-05 | 0 | 2141,4 | 0,1 | 0 | 2140 |
| 50 | 15 | 15 | 15 | -5 | -5 | 2,24E-05 | 0 | 2141,4 | 0,1 | 0 | 2140 |
| 51 | 15 | 15 | 15 | -3 | -3 | 2,24E-05 | 0 | 2141,4 | 0,1 | 0 | 2140 |
| 52 | 15 | 15 | 15 | -1 | -1 | 2,24E-05 | 0 | 2141,4 | 0,1 | 0 | 2140 |
| 53 | 15 | 15 | 15 | 1 | 1 | 2,24E-05 | 0 | 2141,4 | 0,1 | 0 | 2140 |
| 54 | 15 | 15 | 15 | 3 | 3 | 2,24E-05 | 0 | 2141,4 | 0,1 | 0 | 2140 |
| 55 | 15 | 15 | 15 | 5 | 5 | 2,24E-05 | 0 | 2141,4 | 0,1 | 0 | 2140 |
| 56 | 15 | 15 | 15 | 7 | 7 | 2,24E-05 | 0 | 2141,4 | 0,1 | 0 | 2140 |
| 57 | 15 | 15 | 15 | 9 | 9 | 2,24E-05 | 0 | 2141,4 | 0,1 | 0 | 2140 |
| 58 | 15 | 15 | 15 | 11 | 11 | 2,24E-05 | 0 | 2141,4 | 0,1 | 0 | 2140 |
| 59 | 15 | 15 | 15 | 13 | 13 | 2,24E-05 | 0 | 2141,4 | 0,1 | 0 | 2140 |
| 60 | 15 | 15 | 15 | 15 | 15 | 2,24E-05 | 0 | 2141,4 | 0,1 | 0 | 2140 |
| 61 | 15 | 15 | 15 | 17 | 17 | 2,24E-05 | 0 | 2141,4 | 0,1 | 0 | 2140 |
| 62 | 15 | 15 | 15 | 19 | 19 | 2,24E-05 | 0 | 2141,4 | 0,1 | 0 | 2140 |
| 63 | 15 | 15 | 15 | 21 | 21 | 2,24E-05 | 0 | 2141,4 | 0,1 | 0 | 2140 |
| 64 | 15 | 15 | 15 | 23 | 23 | 2,24E-05 | 0 | 2141,4 | 0,1 | 0 | 2140 |
| 65 | 15 | 15 | 15 | -7 | -7 | 2,24E-05 | 0 | 2168,6 | 0,1 | 0 | 2167,6 |
| 66 | 15 | 15 | 15 | -5 | -5 | 2,24E-05 | 0 | 2168,6 | 0,1 | 0 | 2167,6 |
| 67 | 15 | 15 | 15 | -3 | -3 | 2,24E-05 | 0 | 2168,6 | 0,1 | 0 | 2167,6 |
| 68 | 15 | 15 | 15 | -1 | -1 | 2,24E-05 | 0 | 2168,6 | 0,1 | 0 | 2167,6 |
| 69 | 15 | 15 | 15 | 1 | 1 | 2,24E-05 | 0 | 2168,6 | 0,1 | 0 | 2167,6 |
| 70 | 15 | 15 | 15 | 3 | 3 | 2,24E-05 | 0 | 2168,6 | 0,1 | 0 | 2167,6 |

| | | | | | | | | | | | |
|-----|----|-----|-----|----|----|----------|---|--------|-----|---|--------|
| 71 | 15 | 15 | 15 | 5 | 5 | 2,24E-05 | 0 | 2168,6 | 0,1 | 0 | 2167,6 |
| 72 | 15 | 15 | 15 | 7 | 7 | 2,24E-05 | 0 | 2168,6 | 0,1 | 0 | 2167,6 |
| 73 | 15 | 15 | 15 | 9 | 9 | 2,24E-05 | 0 | 2168,6 | 0,1 | 0 | 2167,6 |
| 74 | 15 | 15 | 15 | 11 | 11 | 2,24E-05 | 0 | 2168,6 | 0,1 | 0 | 2167,6 |
| 75 | 15 | 15 | 15 | 13 | 13 | 2,24E-05 | 0 | 2168,6 | 0,1 | 0 | 2167,6 |
| 76 | 15 | 15 | 15 | 15 | 15 | 2,24E-05 | 0 | 2168,6 | 0,1 | 0 | 2167,6 |
| 77 | 15 | 15 | 15 | 17 | 17 | 2,24E-05 | 0 | 2168,6 | 0,1 | 0 | 2167,6 |
| 78 | 15 | 15 | 15 | 19 | 19 | 2,24E-05 | 0 | 2168,6 | 0,1 | 0 | 2167,6 |
| 79 | 15 | 15 | 15 | 21 | 21 | 2,24E-05 | 0 | 2168,6 | 0,1 | 0 | 2167,6 |
| 80 | 15 | 15 | 15 | 23 | 23 | 2,24E-05 | 0 | 2168,6 | 0,1 | 0 | 2167,6 |
| 81 | 15 | 15 | 15 | 23 | 23 | 2,24E-05 | 0 | 2113,4 | 0,1 | 0 | 2112,4 |
| 82 | 15 | 15 | 15 | 23 | 23 | 2,24E-05 | 0 | 2115,4 | 0,1 | 0 | 2114,4 |
| 83 | 15 | 15 | 15 | 23 | 23 | 2,24E-05 | 0 | 2117,4 | 0,1 | 0 | 2116,4 |
| 84 | 15 | 15 | 15 | 23 | 23 | 2,24E-05 | 0 | 2119,4 | 0,1 | 0 | 2118,4 |
| 85 | 15 | 15 | 15 | 23 | 23 | 2,24E-05 | 0 | 2121,4 | 0,1 | 0 | 2120,4 |
| 86 | 15 | 15 | 15 | 23 | 23 | 2,24E-05 | 0 | 2123,4 | 0,1 | 0 | 2122,4 |
| 87 | 15 | 15 | 15 | 23 | 23 | 2,24E-05 | 0 | 2125,4 | 0,1 | 0 | 2124,4 |
| 88 | 15 | 15 | 15 | 23 | 23 | 2,24E-05 | 0 | 2127,4 | 0,1 | 0 | 2126,4 |
| 89 | 15 | 15 | 15 | 23 | 23 | 2,24E-05 | 0 | 2129,4 | 0,1 | 0 | 2128,4 |
| 90 | 15 | 15 | 15 | 23 | 23 | 2,24E-05 | 0 | 2131,4 | 0,1 | 0 | 2130,4 |
| 91 | 15 | 15 | 15 | 23 | 23 | 2,24E-05 | 0 | 2133,4 | 0,1 | 0 | 2132,4 |
| 92 | 15 | 15 | 15 | 23 | 23 | 2,24E-05 | 0 | 2135,4 | 0,1 | 0 | 2134,4 |
| 93 | 15 | 15 | 15 | 23 | 23 | 2,24E-05 | 0 | 2137,4 | 0,1 | 0 | 2136,4 |
| 94 | 15 | 15 | 15 | 23 | 23 | 2,24E-05 | 0 | 2139,4 | 0,1 | 0 | 2138,4 |
| 95 | 15 | 15 | 15 | 23 | 23 | 2,24E-05 | 0 | 2141,4 | 0,1 | 0 | 2140,4 |
| 96 | 15 | 15 | 15 | 23 | 23 | 2,24E-05 | 0 | 2143,4 | 0,1 | 0 | 2142,4 |
| 97 | 15 | 15 | 15 | 23 | 23 | 2,24E-05 | 0 | 2145,4 | 0,1 | 0 | 2144,4 |
| 98 | 15 | 15 | 15 | 23 | 23 | 2,24E-05 | 0 | 2147,4 | 0,1 | 0 | 2146,4 |
| 99 | 15 | 15 | 15 | 23 | 23 | 2,24E-05 | 0 | 2149,4 | 0,1 | 0 | 2148,4 |
| 100 | 15 | 15 | 15 | 23 | 23 | 2,24E-05 | 0 | 2151,4 | 0,1 | 0 | 2150,4 |
| 101 | 15 | 15 | 15 | 23 | 23 | 2,24E-05 | 0 | 2153,4 | 0,1 | 0 | 2152,4 |
| 102 | 15 | 15 | 15 | 23 | 23 | 2,24E-05 | 0 | 2155,4 | 0,1 | 0 | 2154,4 |
| 103 | 15 | 15 | 15 | 23 | 23 | 2,24E-05 | 0 | 2157,4 | 0,1 | 0 | 2156,4 |
| 104 | 15 | 15 | 15 | 23 | 23 | 2,24E-05 | 0 | 2159,4 | 0,1 | 0 | 2158,4 |
| 105 | 15 | 15 | 15 | 23 | 23 | 2,24E-05 | 0 | 2161,4 | 0,1 | 0 | 2160,4 |
| 106 | 15 | 15 | 15 | 23 | 23 | 2,24E-05 | 0 | 2163,4 | 0,1 | 0 | 2162,4 |
| 107 | 15 | 15 | 15 | 23 | 23 | 2,24E-05 | 0 | 2165,4 | 0,1 | 0 | 2164,4 |
| 108 | 15 | 15 | 15 | 23 | 23 | 2,24E-05 | 0 | 2166,4 | 0,1 | 0 | 2165,4 |
| 109 | 15 | -10 | -10 | -7 | -7 | 2,24E-03 | 0 | 2168,6 | 0,1 | 0 | 2167,6 |
| 110 | 15 | 0 | 0 | -7 | -7 | 2,24E-03 | 0 | 2168,6 | 0,1 | 0 | 2167,6 |
| 111 | 15 | 10 | 10 | -7 | -7 | 2,24E-03 | 0 | 2168,6 | 0,1 | 0 | 2167,6 |
| 112 | 15 | 10 | 10 | 0 | 0 | 2,24E-03 | 0 | 2168,6 | 0,1 | 0 | 2167,6 |
| 113 | 15 | -10 | -10 | 7 | 7 | 2,24E-03 | 0 | 2168,6 | 0,1 | 0 | 2167,6 |
| 114 | 15 | 0 | 0 | 7 | 7 | 2,24E-03 | 0 | 2168,6 | 0,1 | 0 | 2167,6 |

Tab. 6.4. Ensemble des 114 stimuli monoton utilisés pour lorsde la qualification.

| Stimulus n° | Configuration du système | | | | | Signaux d'entrées | | | | | | | | |
|-------------|--------------------------|------------------|------------------|------------------|------------------|-------------------|---|--------|-----------|---|--------|-------------|-----------|-----------------|
| | Gain LNA (dB) | Gain VGA1_I (dB) | Gain VGA1_Q (dB) | Gain VGA2_I (dB) | Gain VGA2_Q (dB) | Entrée RF | | | Entrée OL | | | Tension (V) | Phase (°) | Fréquence (MHz) |
| 115 | -15 | -15 | -15 | 23 | 23 | 2,24E-03 | 0 | 2123,4 | 2,24E-03 | 0 | 2133,4 | 2,24E-03 | 0 | 2112,4 |
| 116 | -15 | 15 | 15 | 23 | 23 | 2,24E-03 | 0 | 2123,4 | 2,24E-03 | 0 | 2133,4 | 2,24E-03 | 0 | 2112,4 |
| 117 | -15 | 15 | 15 | 23 | 23 | 1,99E-02 | 0 | 1923,4 | 2,82E-03 | 0 | 2045 | 1,99E-02 | 0 | 2167,4 |
| 118 | -15 | 15 | 15 | 23 | 23 | 1,99E-02 | 0 | 1923,4 | 1,41E-03 | 0 | 2045 | 1,99E-02 | 0 | 2167,4 |
| 119 | -15 | 15 | 15 | 23 | 23 | 7,07E-03 | 0 | 2151 | 7,07E-03 | 0 | 2150 | 7,07E-03 | 0 | 2140 |
| 120 | -15 | 15 | 15 | 23 | 23 | 2,24E-04 | 0 | 2151 | 2,24E-04 | 0 | 2150 | 2,24E-04 | 0 | 2140 |

Tab. 6.5. Ensemble des 6 stimuli bitons utilisés pour lors de la qualification.

| Paramètre Comportementaux | Valeur limite | Détection | N° Vecteur | Valeur fautive détectée | TRCPCi |
|---------------------------|---------------|-----------|------------|-------------------------|--------|
| S11 LNA | -10,2499 | X | 20 | -10,2496 | 1,00 |
| S22 LNA | -1,598 | X | 21 | -2,48E-01 | 0,04 |
| P_comp LNA | -18,4 | X | 21 | -18,402 | 0,99 |
| IIP3 LNA | -11,2146 | X | 21 | -11,2148 | 0,99 |
| Gain IQ dem | 20 | X | 21 | 7,493 | 0,00 |
| Gain IQ dem | 24 | X | 20 | 24,007 | 0,96 |
| IIP3 IQ dem | -1,5 | X | 20 | -1,504 | 0,95 |
| IIP2 IQ dem | 39 | X | 20 | 5,603 | 0,00 |
| Fc LPF I | 3,60E+06 | X | 20 | 1,31E+06 | 0,00 |
| Fc LPF I | 7,90E+06 | X | 20 | 7,91E+06 | 0,92 |
| Fc LPF Q | 3,60E+06 | X | 20 | 1,31E+06 | 0,00 |
| Fc LPF Q | 7,90E+06 | X | 20 | 7,91E+06 | 0,92 |
| IMD3 VGA1 I | -107 | X | 21 | -107,03 | 0,90 |
| IMD3 VGA1 Q | -107 | X | 20 | -107,03 | 0,90 |
| Fc VGA1 I | 2,60E+06 | X | 20 | 2,60E+06 | 0,93 |
| Fc VGA1 I | 2,72E+06 | X | 20 | 2,72E+06 | 0,93 |
| Fc VGA1 Q | 2,60E+06 | X | 20 | 2,60E+06 | 0,93 |
| Fc VGA1 Q | 2,72E+06 | X | 20 | 2,72E+06 | 0,93 |
| IMD3 VGA2 I | -113 | X | 20 | -113,03 | 0,90 |
| IMD3 VGA2 Q | -113 | X | 20 | -113,03 | 0,90 |
| Fc VGA2 I | 2,82E+06 | X | 20 | 2,82E+06 | 0,93 |
| Fc VGA2 I | 2,98E+06 | X | 20 | 2,98E+06 | 0,93 |
| Fc VGA2 Q | 2,82E+06 | X | 20 | 2,82E+06 | 0,93 |
| Fc VGA2 Q | 2,98E+06 | X | 20 | 2,98E+06 | 0,93 |
| V supply | 2,5 | X | 20 | 2,500 | 0,90 |
| V supply | 2,7 | X | 20 | 2,700 | 0,00 |
| LNA Gain | -5,2554 | X | 20 | -5,2557 | 1,00 |
| LNA Gain | 5,2554 | X | 20 | 5,2557 | 1,00 |
| VGA1 I | -0,20 | X | 20 | -0,201 | 0,96 |
| VGA1 I | 0,20 | X | 20 | 0,201 | 0,96 |
| VGA1 Q | -0,20 | X | 20 | -0,201 | 0,96 |
| VGA1 Q | 0,20 | X | 20 | 0,201 | 0,96 |
| VGA2 I | -0,20 | X | 20 | -0,201 | 0,96 |
| VGA2 I | 0,20 | X | 20 | 0,201 | 0,96 |
| VGA2 Q | -0,20 | X | 20 | -0,201 | 0,96 |
| VGA2 Q | 0,20 | X | 20 | 0,201 | 0,96 |
| | CF | 36/36=1 | | TRCP | 0,79 |

Tab. 6.6. Génération et qualification de stimuli pour un système W-CDMA à partir d'un modèle de fautes comportementales.

CHAPITRE VII :

CONCLUSION GENERALE ET PERSPECTIVES

7.1. Conclusion générale

Les fabricants de systèmes microélectroniques sont capables de concevoir des systèmes des plus en plus complexes afin de répondre aux attentes de leurs clients. Devant cette augmentation croissante de la complexité des SoCs AMS&RF, il est nécessaire de garantir une qualité de conception et de fabrication de ces systèmes. Les travaux de recherches présentés dans ce mémoire de thèse se montrent comme une contribution à l'amélioration de ce point. La thèse ainsi présentée dans ce mémoire s'inscrit dans une problématique très actuelle d'aide à la détermination de stimuli pour le test des systèmes électroniques hétérogènes.

Dans le chapitre II, nous avons présenté un état de l'art des méthodes utilisées lors de la conception des systèmes AMS&RF. Durant la conception de ces systèmes, différents niveaux d'abstractions sont mis en place, ils permettent aux concepteurs de passer du cahier des charges à la description finale du système en procédant par raffinements successifs. En parallèle à ces nouvelles méthodes de conception, des langages de description et des simulateurs associés ont été développés. Des analyses dédiées à la simulation des SoCs AMS&RF (analyse en régime établi, détection d'enveloppe) ont été développées afin de permettre la simulation de ces systèmes dans des temps de simulation raisonnables et une précision suffisante. Grâce aux langages de description haut niveau, il est possible de simuler le système complet à différents niveaux de précision et ainsi de vérifier les étapes de la conception. Dans ce chapitre, nous avons insisté plus particulièrement sur le langage VHDL-AMS, langage de description matériel normalisé, car il a été utilisé pour la modélisation du système électronique.

Les différentes finalités du test des systèmes électroniques sont présentées dans le chapitre III. Les

méthodes utilisées lors de la définition de stimuli ont ensuite été exposées en fonction des différents objectifs de test. Leurs avantages et leurs inconvénients ont ainsi été mis en évidence pour leur application à des SoCs AMS&RF.

La chapitre IV présente la description du système RF utilisé pour évaluer l'approche de génération des jeux de test proposée. Le système utilisé est la partie réception d'un système W-CDMA. Après avoir abordé cette technologie de modulation W-CDMA, nous avons présenté l'architecture développée lors de la conception de ce système. Ce système a été conçu suivant une approche Top-Down. Ainsi, la définition de l'architecture au niveau fonctionnel a tout d'abord permis de diviser le système en blocs fonctionnels (amplification, filtrage...). Ensuite, ces blocs ont été spécifiés à la fois par des paramètres fonctionnels et électriques déterminés lors de l'étape de budgétisation. Nous avons alors décrit le système au niveau comportemental afin de vérifier que la définition de l'architecture et la budgétisation avaient été conçues de manière à respecter le cahier des charges. Des résultats de simulations réalisés sur le système décrit en VHDL-AMS ont alors permis de valider cette étape de conception.

Pour améliorer la validation de conception, nous avons proposé dans le chapitre V d'associer au plan de vérification une étape de qualification des stimuli de test. En effet, le plan de vérification définit les tests nécessaires à la vérification, mais sans s'assurer que ces tests permettent de vérifier toutes les spécifications du système. Nous avons donc développé une méthode permettant d'assister le concepteur lors de la définition des jeux de test. Cette méthode est basée sur l'injection de fautes paramétriques dans la description VHDL-AMS du circuit à vérifier. La métrique mise en place permet de qualifier chacun des vecteurs en fonction de leur capacité à détecter de faibles variations paramétriques à l'extérieure des plages de tolérances définies par le concepteur. Il est alors possible de qualifier la capacité de détection de variations fautives de l'ensemble des stimuli proposés lors de la vérification et ainsi de valider l'ensemble des stimuli utilisés. Il sera alors possible de compléter cet ensemble afin de procéder à une vérification plus complète.

Enfin, le chapitre VI propose de réutiliser la méthode développée précédemment mais pour le test matériel. L'origine des défauts n'est plus l'erreur humaine qui a lieu au cours de la conception mais les variations sur les paramètres du processus de fabrication ou l'apparition de défauts locaux. Grâce aux données statistiques du procédé de fabrication, il est possible d'estimer (en faisant l'hypothèse de normalité du circuit) la probabilité d'avoir un circuit fautif et ainsi estimer le nombre de circuits sains qui sera rejetés ou le nombre de circuits fautifs qui pourra être détecté par l'ensemble des stimuli utilisés. Afin de diminuer les temps de simulation nécessaires à l'évaluation des stimuli, une approche de test haut niveau a été proposée. Pour cela, un modèle de fautes reposant sur les paramètres comportementaux du système a été développé. En utilisant ce modèle de fautes, il est possible d'évaluer la capacité d'un stimulus à détecter une variation plus ou moins faible sur les paramètres comportementaux du circuit. Grâce à la présentation de résultats de simulation obtenus pour un système RF (partie réception du système W-CDMA), nous avons montré que la détection de variations paramétriques sur des paramètres comportementaux permet d'estimer assez précisément les détections des variations sur les paramètres structurels. Ces résultats nous ont montrés, pour un cas donné, que notre modèle de fautes haut niveau est adapté à la qualification de stimuli pour le test matériel. Cette méthode de test permet de diminuer d'un facteur 10 les temps d'évaluation des stimuli. La méthode de qualification basée sur un modèle de fautes haut niveau a ensuite été comparée avec une approche réalisée au niveau composant. C'est ainsi que

les mêmes stimuli de test ont été appliqués pour la détection de fautes paramétriques sur les composants d'un bloc du système (LNA). Ces résultats montrent que l'approche proposée au niveau comportementale permet de détecter des variations sur les paramètres niveau composant pour des temps de simulation raisonnables. Mais ces résultats montrent aussi qu'il est possible de faire une analyse plus fine en décrivant une partie du système au niveau composant et le reste au niveau comportemental.

Pour clore ce chapitre, le principe de la génération automatique de stimuli de test a été présentée.

7.2. Perspectives

Dans cette thèse, nous avons mis en place une méthode de qualification des stimuli de test utilisés lors de la conception de systèmes AMS&RF. L'outil développé (PLASMA) permet ainsi de qualifier automatiquement un ensemble de stimuli de test défini par le concepteur. A l'issue de cette analyse, le concepteur est capable d'évaluer la qualité du test de son système. En effet, si de larges variations fautives sur les paramètres du système n'ont pas été détectées, alors le test est incomplet. Afin de pouvoir détecter plus facilement les défauts physiques à l'issue de la fabrication, les concepteurs de systèmes AMS&RF devront dans le futur utiliser des méthodes de CVT (§3.4).

Une première perspective de travail consisterait à mettre en place une méthode permettant d'indiquer aux concepteurs où et comment insérer les blocs de test additionnels afin d'optimiser le test. Pour cela, il faudra définir une bibliothèque de blocs de test (générateurs et analyseurs) capables de tester les principaux paramètres électriques et fonctionnels des systèmes AMS&RF : gain, coefficients de réflexion, non-linéarité, fréquences de coupures... La qualification des stimuli de test réalisée par PLASMA permettra alors d'aider le concepteur à identifier quels éléments de test doivent être ajoutés au système afin de le rendre plus testable.

Un second axe de recherche consisterait à définir et utiliser un modèle de fautes plus précis lors de la qualification des stimuli. En effet, dans cette thèse, nous avons choisi de définir le modèle de fautes dans une approche Top-Down, c'est-à-dire sans avoir connaissance de l'implantation physique du circuit final. Cette approche à l'avantage de permettre de définir le modèle de fautes avant d'avoir complètement terminé la conception du système. Par contre, cette approche conduit à générer un modèle de fautes modélisant moins précisément les défauts physiques pouvant apparaître lors de la fabrication du circuit. Le travail consisterait à déterminer des modèles de fautes haut niveau extraits à partir de fautes injectées à bas niveau. Les modèles de fautes ainsi définis en Bottom-Up seront alors plus précis et plus représentatifs des défauts pouvant apparaître lors de la fabrication du circuit. Ces modèles de fautes pourraient ainsi être directement associés aux *hard IP* (*IP liées à une description structurelle*), afin d'être réutilisés lors de la qualification des stimuli des systèmes impliquant ces IPs.

Une troisième perspective s'inscrit dans un projet en cours de développement chez ST Microelectronics. L'objectif de ce projet est de simuler le programme de test en considérant les appareils et équipements utilisés lors du test. Pour cela, les modèles comportementaux des appareils de test, des cartes de test et des sondes ont été développés en VHDL-AMS ; le programme de test est alors simulé avec tout son environnement de test pour valider son fonctionnement. Une amélioration notable consisterait à utiliser PLASMA dans le cadre de ce projet. En effet, d'une part, la qualification des stimuli de test serait plus précise puisque les imperfections de l'environnement

de test seraient alors considérées lors de l'évaluation et d'autre part, une véritable optimisation et pré-qualification du programme de test pourrait être conduite avant l'arrivée même des premiers prototypes.

Qualification et génération automatiques de stimuli pour le test de systèmes sur puces (SoC) analogiques mixtes et RF

Résumé :

L'augmentation de la complexité des systèmes hétérogènes a conduit à l'apparition d'une nouvelle méthode de conception et de validation. Cette approche qui s'appuie sur un flot de conception descendant s'inspire des méthodes utilisées dans le domaine numérique.

Dans cette thèse, nous avons développé une approche permettant de qualifier le plan de vérification utilisé lors de la validation de conception et d'améliorer le test matériel des systèmes AMS&RF. L'originalité de notre approche est d'utiliser une description comportementale du système pour la qualification et la génération des stimuli de test. En effet, les méthodes de test actuelles utilisant les descriptions niveau composant ne sont pas adaptées au test de systèmes complexes.

En utilisant ce concept, au cours de cette thèse la plateforme PLASMA, PLAteforme pour la qualification et la génération de stimuli pour test de Systèmes Mixtes et Analogiques, a été développée. Cette plateforme est en particulier conçue pour le test des systèmes AMS&RF. PLASMA a été validée sur un émetteur/récepteur W-CDMA intégré conçu par ST Microelectronics.

Mots clés : Systèmes hétérogènes, SoCs AMS&RF, Description comportementale, VHDL-AMS, Validation de conception, Test de production, Qualification de stimuli de test, Modèle de fautes haut niveau.

Qualification and automatic generation of stimuli for the test of analog mixed signal and RF Systems on Chip (SoC)

Abstract :

The increasing complexity of heterogeneous systems leads to the development of a new design and validation method. This approach, re-used from digital domain, is based on Top-Down design flow.

In this thesis, we have developed a method to qualify the verification plan apply during the validation of the design and to upgrade the manufacturing test of AMS&RF systems. In our developed approach, we propose to use the behavioral description of the system for the qualification and generation of test stimuli. In fact, the previously developed test methods use component level description for the generation of stimuli but these approaches are not adapted for the test of complex AMS&RF systems because they involves long simulation times.

By using this behavioural approach, a platform PLASMA has been developed during this thesis. PLASMA qualifies and generates test stimuli for AMS&RF systems by using behavioral description. PLASMA is validated by studying an integrated W-CDMA transceiver developed by ST Microelectronics.

Keywords: Heterogeneous systems, SoCs Ams&RF, Behavioral descriptions, VHDL-AMS, Design validation, Manufacturing test, test stimuli qualification, high abstraction level fault model.