



**HAL**  
open science

# Méthodes de conception des circuits intégrés analogiques et mixtes - Perspectives sur les systèmes électroniques en interaction avec le vivant

Noëlle Lewis

► **To cite this version:**

Noëlle Lewis. Méthodes de conception des circuits intégrés analogiques et mixtes - Perspectives sur les systèmes électroniques en interaction avec le vivant. Electronique. Université Bordeaux 1, 2010. tel-01015800

**HAL Id: tel-01015800**

**<https://theses.hal.science/tel-01015800>**

Submitted on 27 Jun 2014

**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.

# MEMOIRE DE RECHERCHE

En vue de l'obtention du diplôme  
d'HABILITATION A DIRIGER DES RECHERCHES

Spécialité : Electronique

NOËLLE LEWIS

Maître de Conférences, Université BORDEAUX 1  
Laboratoire de l'Intégration du Matériau au Système, IMS

---

**Méthodes de conception des circuits intégrés analogiques et mixtes –  
Perspectives sur les systèmes électroniques en interaction avec le vivant**

---

Soutenance le 9 Avril 2010

Après avis des rapporteurs :

Mme	Valérie VIGNERAS	Professeur	IMS, IPB, Bordeaux
M.	Patrick GARDA	Professeur	LIP6, Université Paris VI
M.	Ian O'CONNOR	Professeur	INL, ECL, Lyon

Devant la commission d'examen composée de :

Mme	Sylvie RENAUD	Professeur	IMS, IPB, Bordeaux
Mme	Valérie VIGNERAS	Professeur	IMS, IPB, Bordeaux
M.	Abdelhamid BENAZZOUZ	Directeur de Recherche	MAC, INSERM, Bordeaux
M.	Patrick GARDA	Professeur	LIP6, Université Paris VI
M.	Ian O'CONNOR	Professeur	INL, ECL, Lyon
M.	Serge SCOTTI	Ingénieur	STMicroelectronics, Grenoble
M.	Alain VACHOUX	Maître de Conférences	LSM, EPFL, Lausanne
M.	Bernard VEYRET	Directeur de Recherche	IMS, CNRS, EPHE, Bordeaux



# TABLE DES MATIERES

<b>INTRODUCTION</b>	<b>1</b>
<b>METHODOLOGIE DE MODELISATION DES CIRCUITS ANALOGIQUES</b>	<b>5</b>
<b>1. Contexte et définitions</b>	<b>5</b>
1.1. Modèle - Modélisation	5
1.2. Conception hiérarchique	6
1.3. Modèle et niveau d'abstraction	7
1.4. Technique de modélisation	8
<b>2. Méthodes de modélisation</b>	<b>9</b>
2.1. Modélisation en phase Bottom-Up	10
2.1.1. Modèle basé sur la connaissance	10
2.1.2. Modèles basés sur l'identification	20
2.2. Modélisation en phase Top-Down	21
2.2.1. Démarche systématique	22
2.2.2. Application	24
<b>3. Modèles multi-niveaux d'abstraction</b>	<b>26</b>
3.1. Macro-modèle d'Amplificateur Opérationnel	26
3.1.1. Sous-circuits de base	26
3.1.2. Générateur automatique de macromodèle	28
3.1.3. Exemples d'utilisation	29
3.2. Modèles VHDL-AMS imbriqués	30
3.2.1. Principe	30
3.2.2. Application	32
<b>4. Conclusion</b>	<b>34</b>
<b>MODELISATION DE SOURCES DE BRUIT POUR SIMULATION TEMPORELLE</b>	<b>35</b>
<b>1. Introduction</b>	<b>35</b>
1.1. Contexte de l'étude	35
1.2. Etat de l'art	36
1.3. Résumé des objectifs et moyens d'étude	39
<b>2. Méthode</b>	<b>39</b>
2.1. Principe et hypothèses	39
2.1.1. Décomposition asymptotique de la densité spectrale de bruit	39
2.1.2. Génération d'un bruit en $1/f^n$ avec n entier positif	40
2.2. Synthèse de l'intégrateur d'ordre $\frac{1}{2}$ - Application au bruit en $1/f$	42
2.3. Implémentation du filtre et performances	45
2.4. Généralisation	46
<b>3. Applications et résultats</b>	<b>48</b>
3.1. Bruits de tension élémentaires	48
3.1.1. Bruit blanc	48
3.1.2. Bruit en $1/f$	50
3.1.3. Bruit combiné en $1/f$ et blanc	52
3.2. Bruit des oscillateurs	52
3.2.1. Modèle d'une sinusoïde bruitée	53
3.2.2. Modèle d'une horloge avec jitter	56
3.3. Modélisation du bruit dans une chaîne de réception	59
<b>4. Conclusion</b>	<b>60</b>

<b>CONCEPTION ANALOGIQUE A BASE D'IPS</b>	<b>63</b>
<b>1. Introduction</b>	<b>63</b>
1.1. Conception de systèmes sur puce numériques	63
1.2. Problématique des blocs IPs analogiques et état de l'art	66
1.3. Exposé du cadre et des objectifs de ce travail	70
<b>2. Conception de Firm IPs pour l'ingénierie neuromorphique</b>	<b>71</b>
2.1. Hiérarchie, primitives et spécifications	71
2.2. Contenu formel d'un IP	73
2.3. Conception d'un Firm IP	74
2.3.1. Procédure de conception	74
2.3.2. Modélisation	76
2.3.3. Caractérisation en vue de la réutilisation	76
<b>3. Flot de conception basé sur la réutilisation</b>	<b>77</b>
3.1. Organisation générale du flot	77
3.2. Structure de la base de données et algorithme d'exploration	78
3.3. Aide à la réutilisation	81
3.3.1. Diagnostic de réutilisation	81
3.3.2. Migration des spécifications	81
3.3.3. Migration technologique	82
3.4. Vérification	84
<b>4. Conclusion</b>	<b>86</b>
<b>ACQUISITION ET CONTROLE DE L'ACTIVITE BIOELECTRIQUE</b>	<b>87</b>
<b>1. Contexte local</b>	<b>87</b>
<b>2. Contexte scientifique et problématique</b>	<b>89</b>
2.1. Introduction	89
2.2. Le signal nerveux	89
2.3. Technologie de l'interface neuro-électrique	91
2.4. Circuits de détection de l'information	93
2.5. Stimulation contrôlée	97
2.6. Traitement temps réel	98
<b>3. Projets et stratégie</b>	<b>99</b>
3.1. Projet STN Oscillations	99
3.2. Projet HYRENE	100
3.3. Projet NeuroEM	102
3.4. Stratégie	103
3.4.1. Méthodologie de conception des systèmes hybrides vivant/électronique	103
3.4.2. Pluridisciplinarité et ancrage dans les thématiques de recherche reconnues	105
3.4.3. Transfert et politique de valorisation	106
<b>4. Conclusion</b>	<b>107</b>
<b>CONCLUSION</b>	<b>109</b>
<b>REFERENCES BIBLIOGRAPHIQUES</b>	<b>113</b>
<b>LISTE DES PUBLICATIONS DE L'AUTEUR</b>	<b>121</b>

# INTRODUCTION

Le présent mémoire propose une synthèse de mes activités de recherche sur la période 1993 – 2009 ainsi que mon projet pour les années à venir.

Ces travaux se sont déroulés au laboratoire de l'*Intégration des Composants et Systèmes Electroniques IXL*, initialement dirigé par le Pr. J.P. Dom, et devenu aujourd'hui le laboratoire de l'*Intégration du Matériau au Système IMS*, dirigé par le Pr. P. Fouillat. Mon parcours est resté associé à la thématique générale de la **Conception Optimisée des Circuits Analogiques et Mixtes**, même si ses étapes successives l'ont rattaché à différentes équipes du laboratoire, au fil de son histoire. Ainsi, d'un point de vue institutionnel, mes travaux ont été hébergés par :

- l'équipe *Caractérisation et Modélisation* des Pr. J.P. Dom puis T. Zimmer, durant mon DEA et ma thèse,
- l'équipe *Modélisation Comportementale des Circuits et Systèmes* du Pr. H. Lévi, durant la période 1998 – 2004,
- l'équipe *Ingénierie des Systèmes Neuromorphiques (ISN)* du Pr. S. Renaud, durant la période 2004 – 2009.

Mon intégration dans l'équipe ISN a eu lieu à l'occasion de l'encadrement de la thèse de T. Lévi, qui avait pour objectif d'appliquer un concept innovant en conception analogique aux circuits développés par celle-ci. C'est avec un réel intérêt pour les activités de l'équipe ISN en lien avec les **Neurosciences**, que j'ai décidé d'approfondir mes connaissances sur le sujet, en demandant un Congé pour Recherche et Conversion Thématique qui m'a été accordé sur 2 ans, de 2007 à 2009. Tirant profit de cet aménagement de service, je suis en mesure aujourd'hui de déposer mon dossier d'Habilitation à Diriger des Recherches et d'animer à partir du 1<sup>er</sup> Janvier 2010 une nouvelle équipe centrée sur l'**Interaction des Systèmes Electroniques avec le Vivant**, nommée EIBio.

Le contexte scientifique général de ce mémoire est celui des **Méthodes et Outils de Conception des Circuits Intégrés Analogiques et Mixtes**. La tâche de conception de fonctions électroniques sur circuit intégré ne peut être aujourd'hui menée à bien sans l'aide d'outils informatiques, véritables plate-formes logicielles d'aide à la conception. La complexité croissante des circuits intégrés nécessite une amélioration constante des méthodes

et outils de conception, afin de réduire au maximum le temps de développement, tout en assurant la fiabilité des circuits fabriqués. Un paradoxe connu dans l'industrie du semi-conducteur sous le nom de *Design Productivity Gap* met en avant l'écart entre le nombre de transistors technologiquement disponibles sur une puce et la possibilité de les assembler convenablement pour assurer une fonctionnalité. Cela revient à comparer l'efficacité des technologies de conception à celle des technologies de fabrication. Depuis le début de l'ère des circuits intégrés, ce *Design Productivity Gap* ne cesse de croître et les recommandations de l'*ITRS (International Technology Roadmap for Semiconductors)* reviennent constamment sur la nécessité de proposer de nouvelles solutions de conception automatisée (*EDA, Electronic Design Automation*).

Ajoutons à cela la montée en puissance des composants intégrés analogiques et mixtes (*AMS, Analog and Mixed Systems*), sur lesquels cohabitent, au sein d'un même substrat, des fonctions numériques et analogiques. La différence de maturité des processus de conception analogique et numérique apparaît alors nettement et c'est la partie analogique qui requiert le plus grand effort de conception. En effet, alors que la synthèse de circuits numériques est aujourd'hui entièrement automatisée, la conception analogique souffre d'un manque de formalisme défavorable à son automatisation. La variation continue des signaux, la multitude des facteurs de performances, leur interdépendance, leur sensibilité vis-à-vis de fluctuations physiques peuvent être citées comme des freins inhérents à la nature analogique des circuits.

Ainsi, mes travaux de recherche ont été effectués dans l'objectif général d'**optimiser le flot de conception analogique**, c'est-à-dire d'automatiser un maximum de tâches aujourd'hui réalisées manuellement. Cela nécessite avant tout de proposer des **méthodes et formalismes** qui permettront de passer à la dimension d'**outils logiciels de conception**.

Les 3 premiers chapitres de ce mémoire s'inscrivent pleinement dans cet objectif et déclinent ma contribution selon les 3 axes suivants :

- le **chapitre 1** est consacré à une réflexion générale sur les **Méthodes de Modélisation des Circuits Analogiques et Mixtes**, menée de façon transversale sur l'ensemble de mon travail d'encadrement doctoral,
- le **chapitre 2** s'attache à une question de modélisation particulière et non résolue dans les simulateurs électriques actuels, il s'agit de la **Modélisation de Sources de Bruit pour Simulation Temporelle** ; cela met en œuvre une technique de génération de modèles basée sur les filtres d'ordre non entier, en partenariat avec des chercheurs en Automatique de l'équipe CRONE,

- le **chapitre 3** met la **réutilisation** au cœur du processus de conception et propose d'appliquer le concept d'**IP** (*Intellectual Property*), connu et efficace en conception de systèmes numériques, à la conception de systèmes intégrés analogiques ; un paradigme de **conception analogique à base d'IPs** y est développé et appliqué à la **conception de circuits neuromimétiques**.

Le **chapitre 4** fait l'exposé de mon projet de recherche qui s'articule autour des **Systèmes Electroniques en Interaction avec le Vivant**. Ce dernier chapitre commence par introduire le contexte pluridisciplinaire et la problématique des **Systèmes d'Acquisition et de Contrôle de l'Activité Bioélectrique**, et s'achève par la description du cadre stratégique des projets de l'équipe E $\ell$ Bio.





# METHODOLOGIE DE MODELISATION

## DES CIRCUITS ANALOGIQUES ET MIXTES

### 1. Contexte et définitions

#### 1.1. Modèle - Modélisation

Un modèle est « une représentation en langage mathématique d'un aspect de la réalité »<sup>1</sup>, il s'agit d'une représentation abstraite d'une réalité physique. La modélisation consiste à trouver une loi mathématique représentative du comportement d'un système et la vérification de la vraisemblance de cette loi se fait par comparaison avec données de référence. Ces données de référence proviennent de mesures avant tout, et parfois de simulations réalisées à partir de modèles déjà validés. Cette loi mathématique ne révèle nécessairement qu'un « aspect de la réalité » qui dépend du cadre d'utilisation du système étudié. Dans le contexte de la simulation électrique des circuits et systèmes, un modèle doit en outre être exécutable. Ainsi, les principaux critères de qualité d'un modèle sont :

- la *précision* : elle caractérise la capacité du modèle à couvrir un grand nombre de contextes d'utilisation ; plus le modèle est précis, plus le nombre d'équations et de paramètres est important. Lorsque l'on simplifie le modèle en ne prenant pas en compte tous les effets de second ordre, on élève son degré d'*abstraction*,
- la *rapidité* d'exécution : elle dépend de la technique d'implémentation des équations, du simulateur et de la précision aussi.

Le développement d'un modèle peut se décomposer en plusieurs étapes : on appellera *méthode de modélisation* l'étape qui cherche un jeu d'équations descriptives du comportement d'un circuit ; ces équations étant connues, on appellera plutôt *technique de modélisation* l'étape consistant à implémenter ce jeu d'équations dans un style donné voire un langage. La technique de modélisation dépend fortement du type de comportement observé (continu, discret, logique) mais également du *niveau hiérarchique* auquel on se place pour observer un système.

---

<sup>1</sup> Giorgio ISRAEL, *La Mathématisation du réel*, Editions du Seuil, 1996.

## 1.2. Conception hiérarchique

Les systèmes aujourd'hui intégrés sur une puce étant d'une grande complexité, la hiérarchisation de la conception est une nécessité. On commence par concevoir et valider un algorithme ou une architecture au moyen de blocs fonctionnels, puis on descend progressivement dans le détail des blocs, jusqu'à la conception de circuits élémentaires au niveau transistor ou portes logiques. Le problème initial étant la traduction du cahier des charges en un circuit intégré fonctionnel, cette approche revient à décomposer le problème en sous-problèmes etc. et donne lieu à plusieurs *niveaux hiérarchiques*. Les deux sens de parcours de cette hiérarchie, descendant (*Top-Down*) et ascendant (*Bottom-Up*), correspondent à différentes phases de la conception du système. Chaque niveau hiérarchique est caractérisé par un ensemble d'entités permettant de décrire la *topologie* du système : on parle aussi de *primitives* et de *description structurelle* réalisée à partir de ces primitives.

La fig.1 donne un exemple de système décomposé en 3 niveaux hiérarchiques ; au **niveau système** la topologie est décrite au moyen de blocs fonctionnels, au **niveau bloc** les primitives sont des circuits, enfin au **niveau circuit** les primitives sont des transistors. Une autre manière de décrire un niveau hiérarchique est de donner un ensemble de *performances*. Lors d'un projet de conception, cette hiérarchie est contrainte au plus haut niveau ainsi qu'au plus bas niveau : en haut ce sont les *spécifications système* (données « client ») qui fixent les performances, en bas ce sont les *contraintes technologiques* qui imposent les marges de conception.

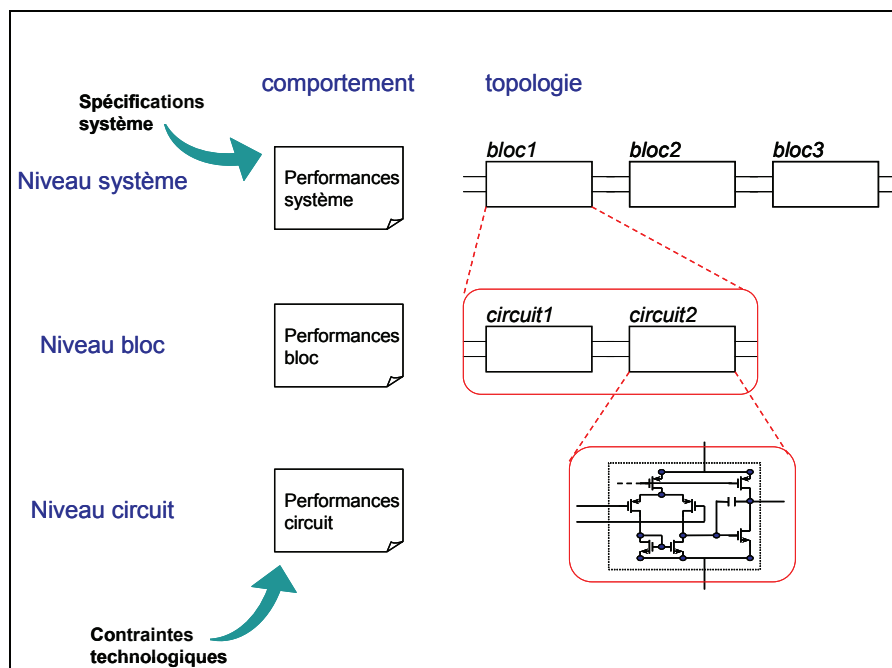


Figure 1 : Hiérarchie de la conception

On appelle *flot de conception* la succession des décisions permettant de passer d'un niveau hiérarchique à un niveau adjacent. Le parcours Top-Down du flot de conception consiste à propager et distribuer les spécifications système vers les niveaux inférieurs, jusqu'au schéma transistor de chaque circuit ; on parle aussi de *synthèse électrique*. Le parcours Bottom-Up consiste à *vérifier* que l'implémentation électrique, (voire physique si on tient compte de parasites de layout) permet bien de réaliser un système conforme aux spécifications.

### 1.3. Modèle et niveau d'abstraction

Les modèles jouent un rôle essentiel dans ce flot de conception. Au plus bas niveau, le concepteur manipule des primitives de type transistor, pour lesquelles modèles et valeurs des paramètres sont donnés. Par contre, il lui incombe souvent de développer les modèles des primitives des niveaux hiérarchiques supérieurs. L'apparition des langages de description des systèmes matériels (*Hardware Description Languages*), tels que Verilog, VHDL et leurs extensions à l'analogique Verilog-AMS, VHDL-AMS, ont permis de décrire des modèles de circuit à différents niveaux d'abstraction, simulables dans un environnement de conception traditionnel, c'est-à-dire compatibles avec un simulateur analogique<sup>2</sup> et/ou logique<sup>3</sup>. En général, les modèles sont de plus en plus abstraits lorsque l'on monte dans la hiérarchie et de plus en plus précis lorsque l'on descend. Une classification exhaustive des niveaux d'abstraction est proposée dans [VAC02] et reprise ici :

- Niveau causal : prise en compte de la simple relation de cause à effet
- Niveau discret :
  - Description de cycles
  - **Description dirigée par évènements**
- Niveau continu :
  - Description de type flot de signal (blocs fonctionnels unidirectionnels)
  - **Description de type conservatif** (blocs a-directionnels, prise en compte de la conservation de l'énergie associée à chaque port, lois de Kirchhoff)

Ces niveaux d'abstractions ne sont pas tous simulables par un simulateur électrique (analogique ou logique) ; les hauts niveaux d'abstraction sont plutôt décrits en langage C ou

---

<sup>2</sup> Résolution des équations de Kirchhoff, méthode conservant l'énergie du réseau.

<sup>3</sup> Technique de simulation basée sur l'évaluation de fonctions logiques et la propagation d'évènements. On parle de simulation dirigée par évènements.

C++, System C, Matlab ou sa version graphique Simulink. Les travaux recensés ici ne concernent que les modèles simulables par un simulateur électrique, ce qui correspond aux comportements dirigés par événements ou bien continus conservatifs.

En conception de circuits analogiques, on appelle généralement *modèle comportemental* tout modèle décrivant le comportement électrique d'un circuit autrement que par une description structurelle au niveau transistor. On trouve souvent une terminologie plus précise qui distingue principalement 3 niveaux d'abstraction [SHV95] :

- Niveau *fonctionnel* : relations entrée/sortie, représentation de la fonction idéale,
- Niveau *comportemental* : description prenant en compte les caractéristiques de la réalisation physique, ses non-idéalités,
- Niveau circuit : description avec les primitives de plus bas niveau (transistors), on parle aussi de niveau transistor.

En résumé, ces modèles sont tous des modèles de comportement, mais considérés à des niveaux d'abstraction différents. L'intérêt des ces différents niveaux d'abstraction est de pouvoir être utiles à différents niveaux hiérarchiques du flot de conception, dans le meilleur compromis précision/rapidité.

#### **1.4. Technique de modélisation**

La technique de modélisation concerne plutôt le « style » dans lequel le modèle est implémenté.

En analogique, les premiers modèles comportementaux ont été développés dans les années 70-80 et ne bénéficiaient pas encore des langages de description matérielle<sup>4</sup>. La technique de modélisation est alors contrainte par l'usage de primitives SPICE (transistors, diodes, composants passifs élémentaires, sources de tension ou courant contrôlées). Ces premiers modèles comportementaux prennent alors la forme de schémas équivalents, issus de la simplification du schéma d'origine ou d'une construction empirique. Ce type de modèle comportemental porte le nom de *macromodèle* et le précurseur est le macromodèle d'amplificateur opérationnel de Boyle [BPC74]. Le paragraphe **3.1.** décrira un modèle développé selon cette technique.

L'avènement des langages de description matérielle a permis de modéliser les circuits directement en implémentant des *équations différentielles algébriques* (adaptées aux

---

<sup>4</sup> La norme VHDL-AMS a été finalisée en 1999.

comportements analogiques continus dans le temps) ou des *algorithmes dirigés par évènements* (adaptés aux comportements des circuits à temps discret).

Enfin, on trouve également une technique de modélisation ne passant pas par la recherche d'équations comportementales mais qui consiste à modéliser sous la forme de tableaux. On décrit alors les relations entrées/sorties numériquement, pour des valeurs bien précises des entrées et avec un degré de paramétrage variable.

Le présent chapitre aborde la question des méthodes et techniques de modélisation des circuits analogiques et mixtes, telles qu'elles ont été abordées dans les 3 thèses suivantes : ma thèse [LEW97] et celles de A. Fakhfakh [FAK02] et G. Monnerie [MON05]. Les domaines d'application envisagés concernent essentiellement les circuits radio-fréquence et les circuits neuromorphiques.

## **2. Méthodes de modélisation**

On s'intéresse ici au processus partant de l'analyse du comportement d'un circuit et aboutissant à un système d'équations ou un algorithme descriptifs. Ce processus diffère selon les phases de la conception d'un système : en phase descendante on parle de *raffinement* des modèles, dans le sens où ils vont vers des niveaux de précision croissants (grand nombre de variables d'état et de performances modélisées), en phase montante on parle à l'inverse de *simplification* de modèles.

Une illustration est proposée fig.2. Si on considère par exemple l'oscillateur local d'un système de réception RF, on ne le modélise pas avec le même niveau de précision à tous les niveaux hiérarchiques du système. Une simple relation sinusoïdale peut suffire au *niveau système* ; au *niveau bloc* on peut augmenter la précision du modèle et s'intéresser à l'architecture de l'oscillateur (type PLL) pour en détailler les performances ; puis au *niveau circuit*, on s'intéressera à l'architecture et aux performances de chacun des blocs de la PLL. Il n'y a qu'au niveau circuit que l'on a une vue détaillée de l'implémentation physique des différents composants du système. Ainsi, au cours de la conception d'un système, le processus de modélisation prendra des aspects différents selon que l'on se trouve en phase Top-Down ou en phase Bottom-Up. En résumé, on peut dire que le compromis précision/rapidité des modèles penchera en faveur de la rapidité en phase Top-Down et plutôt en faveur de la précision en phase Bottom-Up.

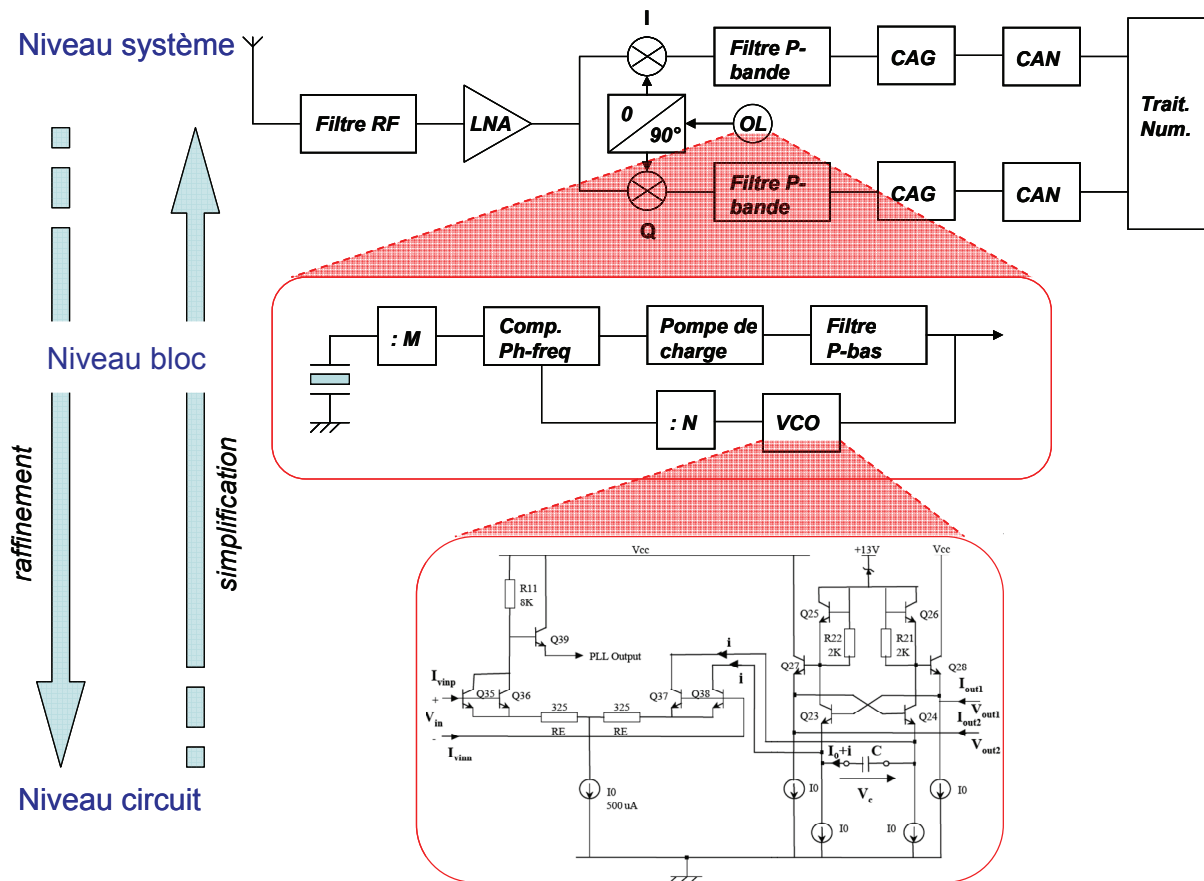


Figure 2 : Processus de modélisation selon les niveaux hiérarchiques de la conception

## 2.1. Modélisation en phase Bottom-Up

En phase de conception Bottom-Up, le point de départ est un circuit dimensionné pour un objectif précis. Il s'agit d'en extraire un modèle comportemental, nécessairement plus abstrait que la description au niveau transistor, mais capable de propager les performances du circuit réalisé vers les niveaux hiérarchiques supérieurs. Cela est illustré fig.2.

Deux approches sont alors possibles [GIE02] : une approche « boîte blanche » pour extraire un modèle *de connaissance* ou une approche « boîte noire » pour extraire un modèle *par identification*.

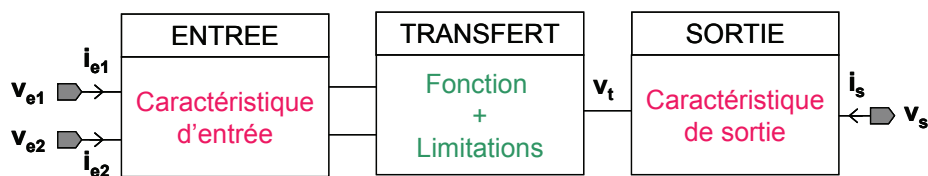
### 2.1.1 Modèle basé sur la connaissance

La première approche est basée sur la connaissance de la structure interne du circuit (topologie) et de lois électriques simples : modèle de premier ordre pour les transistors (Ebers-Moll ou MOS niveau 1) et lois de Kirchoff. Il s'agit d'effectuer un calcul symbolique et d'exprimer les performances du circuit à partir des paramètres clés de ses constituants. On aboutit à un système d'équations qui pourra éventuellement être simplifié.

Le principal avantage est que le modèle ainsi développé garde un sens physique et peut même être prédictif, si le domaine de validité du système est clairement connu. Ces modèles permettent souvent une meilleure compréhension du comportement du circuit et ont en ce sens une vertu « pédagogique ». Les inconvénients sont que l'obtention du système d'équation est une tâche qui se complexifie rapidement avec la taille du circuit et qu'elle est plus facilement applicable aux circuits linéaires. Notons qu'il existe une famille de simulateurs dits symboliques pouvant réaliser cette étape [BOR98], [GIS91], mais ces simulateurs restent très marginaux par rapport aux simulateurs numériques de type SPICE.

Cette méthode a été appliquée dans ma thèse pour développer une bibliothèque de modèles de circuits pour systèmes neuromorphiques et dans la thèse d'A. Fakhfakh pour modéliser les blocs d'une boucle à verrouillage de phase analogique (PLL560B).

Le modèle est systématiquement structuré en 3 sous-blocs comme montré fig.3.



**Figure 3 : Architecture d'un modèle comportemental**

Le modèle comportemental doit être fidèle y compris dans la prise en compte des interfaces, c'est pourquoi il doit décrire non seulement la fonction principale du circuit (relation entrées/sorties) assorties de ces éventuelles imperfections mais également les caractéristiques d'entrée et de sortie du circuit. Les modèles développés dans ces travaux ont été implémentés dans divers langages de description matérielle (HDL-A, Verilog-A, VHDL-AMS), en suivant la même décomposition comme développé fig.4. Si la caractéristique de transfert du circuit relie les tensions d'entrée/sortie, on parle de circuit en mode tension, si elle relie les courants d'entrée/sortie, on parle de circuit en mode courant. Certains circuits sont mixtes de ce point de vue, citons les amplificateurs de transconductance qui sont en mode tension à l'entrée et en mode courant en sortie.

Notons qu'il n'est pas toujours possible d'obtenir une expression symbolique explicite comme c'est le cas des expressions de la fig.4, surtout si l'on souhaite modéliser le circuit avec une précision importante. Ceci n'est pas un problème car un simulateur analogique est conçu pour résoudre les équations implicites.

Notons également que les caractéristiques d'entrée et de sortie d'un circuit analogique sont souvent non-linéaires mais peuvent cependant être linéarisées autour d'un point de



repos ; dans ce cas, les fonctions  $f(\cdot)$  et  $h(\cdot)$  font simplement intervenir les résistances d'entrée et de sortie.

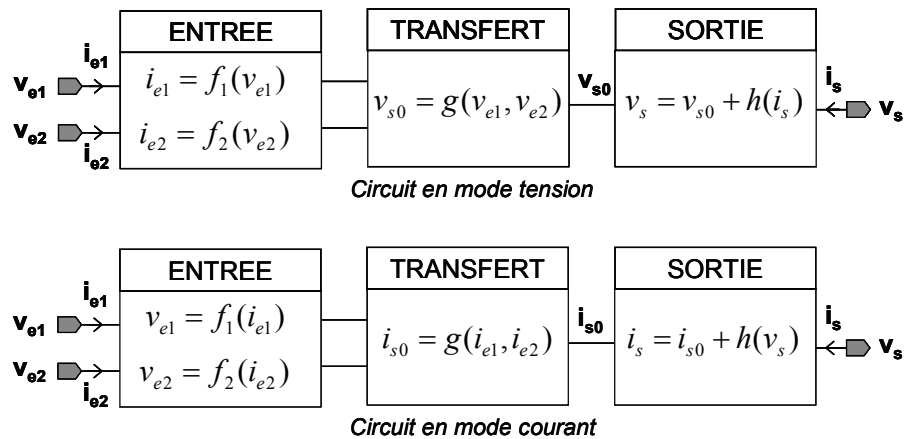


Figure 4 : Expression générique d'un modèle comportemental

a) Modèles comportementaux pour circuits neuro-mimétiques

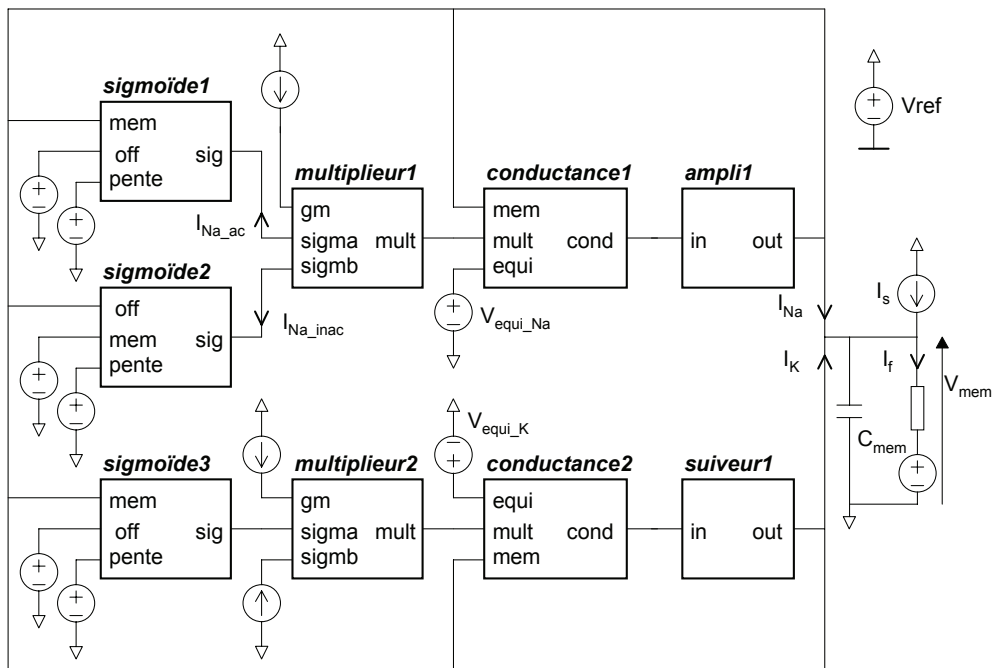


Figure 5 : Schéma bloc d'un neurone analogique

La fig.5 représente le schéma bloc d'un neurone analogique, biologiquement réaliste, conçu dans la thèse d'A. Laflaquière [LAF98]. Les blocs analogiques **sigmoïde**, **multiplieur**, **conductance** et **suiveur/ampli** ont été conçus pour reproduire, dans leur comportement électrique, les équations du modèle de neurone de Hodgkin-Huxley [HOH52]. Les modèles comportementaux ont été écrits en HDL-A, langage précurseur du standard VHDL-AMS.

Les tableaux qui suivent reprennent de manière condensée les équations comportementales extraites du schéma de ces 4 blocs ; une fois ces équations écrites en HDL-

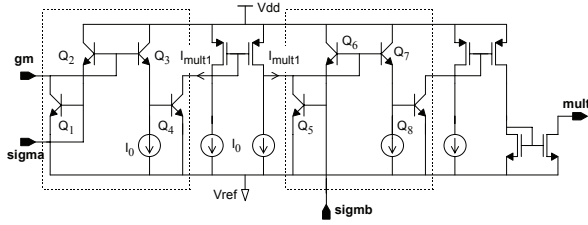
A, les modèles sont évalués et ajustés par comparaison à des simulations au niveau transistor, certains résultats sont reportés ici sous forme de courbes.

bloc sigmoïde	
Schéma transistor	Modèle comportemental
	$I_{mem} = I_{off} = I_{pente} = 0$ $I_{dc} = \frac{I_0}{1 + \exp[-f(V_{pente})(V_{mem} - V_{off})]}$ $f(V_{pente}) = \frac{Rcg_1}{V_T} \left( \frac{1}{2} + \frac{g_2}{2I_0} V_{pente} \right)$ $\tau \frac{dI_{sig0}}{dt} = I_{dc} - I_{sig0}$ $I_{sig} = I_{sig0} + \frac{V_{sig}}{Rs}$
Tests comparatifs	
<p><b>Caractéristique de transfert statique</b></p> <p>◆ Δ ◆ : circuit complet          — : modèle</p>	
<p><b>Réponse transitoire</b></p> <p>◆ : circuit complet          — : modèle</p>	

**Tableau 1 : Modèle comportemental du bloc sigmoïde – équations et caractérisation**

## bloc multiplieur

### Schéma transistor



### Modèle comportemental

$$V_{\sigma} \approx V_T \ln \left( \frac{I_{gm}}{I_S} \right)$$

$$V_{gm} \approx V_T \ln \left( \frac{-I_{\sigma}}{I_S} \right) + V_T \ln \left( \frac{I_{gm}}{I_S} \right)$$

$$V_{\sigma mb} \approx V_T \ln \left( \frac{I_{mult1}}{I_S} \right)$$

$$I_{mult0} = \frac{I_{gm} I_{\sigma} I_{\sigma mb}}{I_0^2}$$

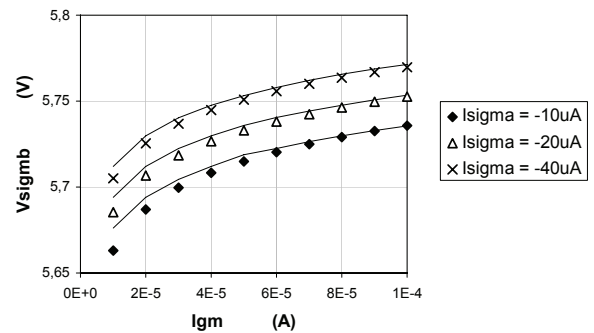
$$I_{mult} = I_{mult0} + \frac{V_{mult}}{R_s}$$

### Tests comparatifs

#### Caractéristique d'entrée

χ Δ ◆ : circuit complet

— : modèle



#### Caractéristique de transfert

χ Δ ◆ : circuit complet

— : modèle

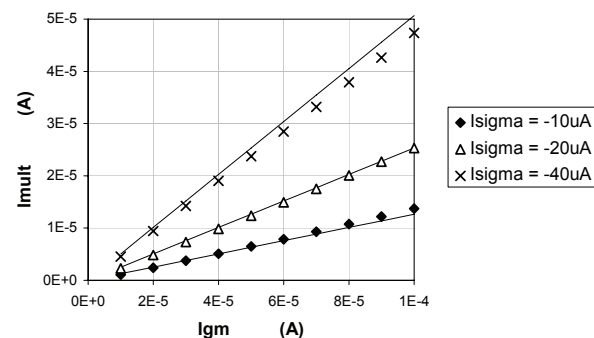


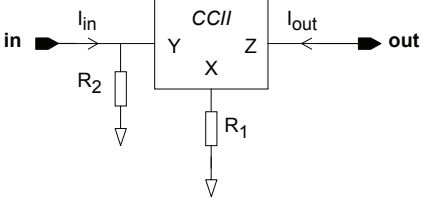
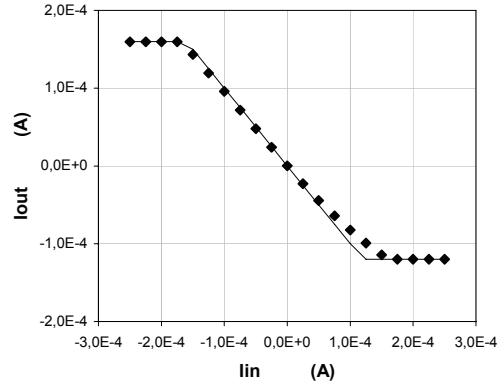
Tableau 2 : Modèle comportemental du bloc multiplieur – équations et caractérisation

**bloc conductance**

Schéma transistor	Modèle comportemental
	$I_{equi} = I_{mem} = 0$ $V_{mult} = V_0 + aI_{mult}$ $I_{cond0} = 2 \frac{g_1}{I_0} I_{max} (V_{mem} - V_{equi})$ $I_{cond} = I_{cond0} + \frac{V_{cond}}{R_s}$
Tests comparatifs	
<p><i>Caractéristique de transfert</i></p> <p> <math>\chi \Delta \blacklozenge</math> : circuit complet              — : modèle           </p>	

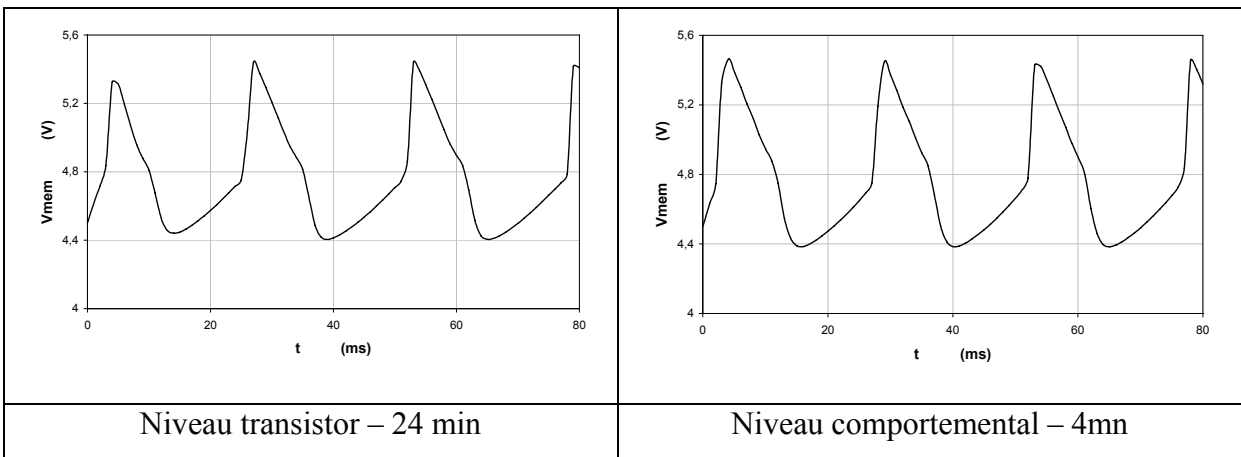
**Tableau 3 : Modèle comportemental du bloc conductance – équations et caractérisation**

**bloc ampli-suiveur**

Schéma transistor	Modèle comportemental
	$V_{in} = R_2 I_{in}$ <p>Si <math>I_{in} &lt; IL</math>, alors <math>I_{out0} = IOL</math>          Si <math>I_{in} &gt; IH</math>, alors <math>I_{out0} = IOH</math>          Sinon <math>I_{out0} = -\frac{R_2}{R_1} I_{in}</math>.</p> $I_{out} = I_{out0} + \frac{V_{out}}{R_s}$
Tests comparatifs	
<p>Caractéristique de transfert</p> <p>⊗ Δ ◆ : circuit complet          — : modèle</p>	

**Tableau 4 : Modèle comportemental du bloc ampli/suiveur – équations et caractérisation**

Une simulation du système complet, (soit l'équivalent d'un neurone de Hodgkin-Huxley), a été menée en assemblant ces 4 blocs ; les résultats sont donnés fig.6.



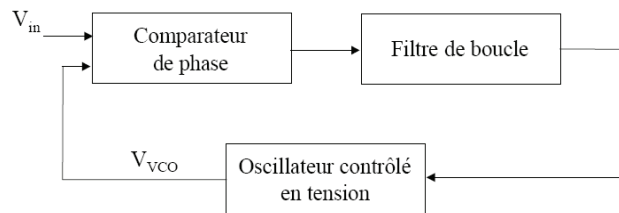
**Figure 6 : simulation transitoire du neurone – courbes et durées de simulation**

Les temps de simulation ont été comparés sur une station SUN SPARC5. La simulation du système du niveau transistor dure 24 mn si l'ensemble des courants et tensions du circuit sont sauvegardés, 17 mn si l'on ne sauvegarde que les courants et tensions externes aux blocs. La simulation avec les modèles HDL-A dure quant à elle 4 mn, soit un gain de temps d'un facteur 6 par rapport à la simulation niveau transistor la plus complète.

### ***b ) Modèles comportementaux pour PLL analogique 560B***

La même méthode a été mise en œuvre dans une partie de la thèse d'A. Fakhfakh [FAK02], pour modéliser une boucle à verrouillage de phase standard identifiée sous le numéro PLL560B [GHL01].

La boucle à verrouillage de phase (Phase Locked Loop, PLL) est un asservissement de phase, circuit fondamental des systèmes de communications. Il est utilisé notamment dans des applications de restitution de porteuse, de démodulation de fréquence, de synthèse de fréquence. Une PLL est constituée d'un circuit comparateur de phase et d'un oscillateur contrôlé en tension (Voltage Controlled Oscillator, VCO) ; le circuit comparateur de phase est lui-même composé d'un circuit de type multiplieur ou comparateur logique suivi d'un filtre passe-bas (fig.7).

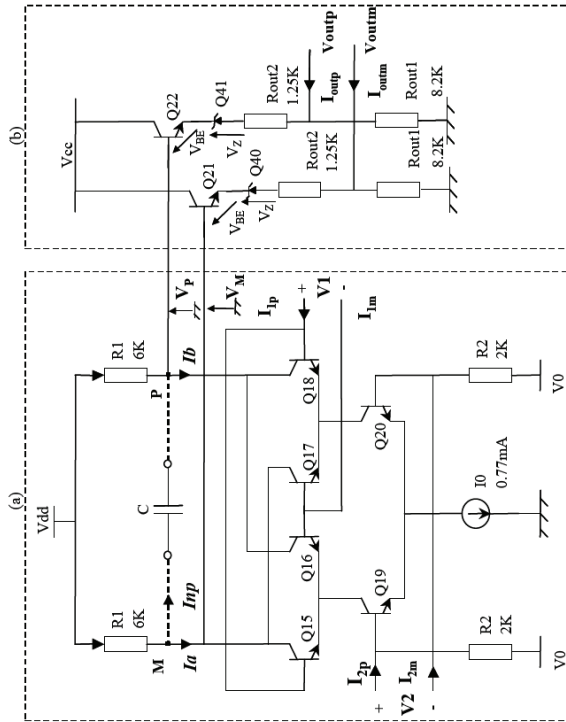


**Figure 7 : Architecture d'une PLL**

Dans la PLL560B, c'est une cellule de Gilbert munie d'une capacité de sortie qui joue le rôle du comparateur de phase et l'oscillateur est à relaxation. Les modèles comportementaux ont été écrits en Verilog-A. Les tableaux qui suivent reprennent de manière condensée les modèles extraits du schéma de ces 4 blocs ; les modèles sont évalués et ajustés par comparaison à des simulations au niveau transistor, certains résultats sont reportés ici sous forme de courbes.

## bloc comparateur de phase

### Schéma transistor



### Modèle comportemental

```

module phcomp (in2p, in2n, in1p, in1n, outp, outn) ;
electrical in2p, in2n, in1p, in1n, outp, outn;
parameter real I0=0.741m, Vdd=13, fc=300,
Vshift=7.096, v0=3.69, Rout=1178, Beta=110,
Rv2=2K, Kdiv=0.868, R1=6K;
real I19, I20, I15, I16, I17, I18, Ia, Ib, C ;
electrical v1, v2, v3, v4, outp0, outn0;
analog begin
C = 1 / (4*`PI*R1*fc) ;
V(v2) <+ V(in2p) - V(in2n) ;
V(v1) <+ V(in1p) - V(in1n) ;
I19 = I0 / (1+exp(-V(v2)/$vt)) ;
I20 = I0 / (1+exp(V(v2)/$vt)) ;
I15 = I19 / (1+exp(V(v1)/$vt)) ;
I16 = I19 / (1+exp(-V(v1)/$vt)) ;
I17 = I20 / (1+exp(-V(v1)/$vt)) ;
I18 = I20 / (1+exp(V(v1)/$vt)) ;
/ input characteristic /
I(in2p) <+ I19/Beta + (V(in2p)-v0) / Rv2 ;
I(in2n) <+ I20/Beta + (V(in2n)-v0) / Rv2 ;
I(in1p) <+ (I16+I17)/Beta ;
I(in1n) <+ (I15+I18)/Beta ;
/ transfer characteristic /
Ia = I15 + I17;
Ib = I16 + I18;
V(v3) <+ Vdd - R1*(Ib+I(v3,v4)) ;
V(v4) <+ Vdd - R1*(Ia-I(v3,v4)) ;
I(v3,v4) <+ C * ddt(V(v3,v4));
V(outp0) <+ Kdiv * (V(v3) - Vshift);
V(outn0) <+ Kdiv * (V(v4) - Vshift);
/ output characteristic /
V(outp) <+ V(outp0) + Rout*I(outp);
V(outn) <+ V(outn0) + Rout*I(outn);
end
endmodule

```

### Tests comparatifs

#### Caractéristique de transfert :

tension différentielle de sortie en fonction du déphasage entre les 2 signaux d'entrée

Pour une durée d'analyse égale à 50 périodes d'entrées et pour un déphasage égal à  $\pi/6$ , le temps de la simulation comportementale est de 12.55 s. Celui de la simulation du schéma transistor est de 35.15 s. On obtient par conséquent un gain en simulation égale à 2.8.

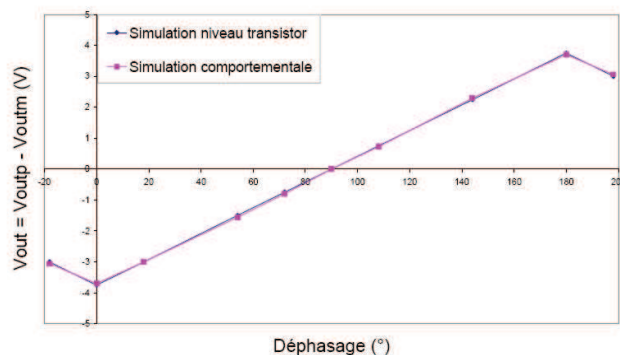
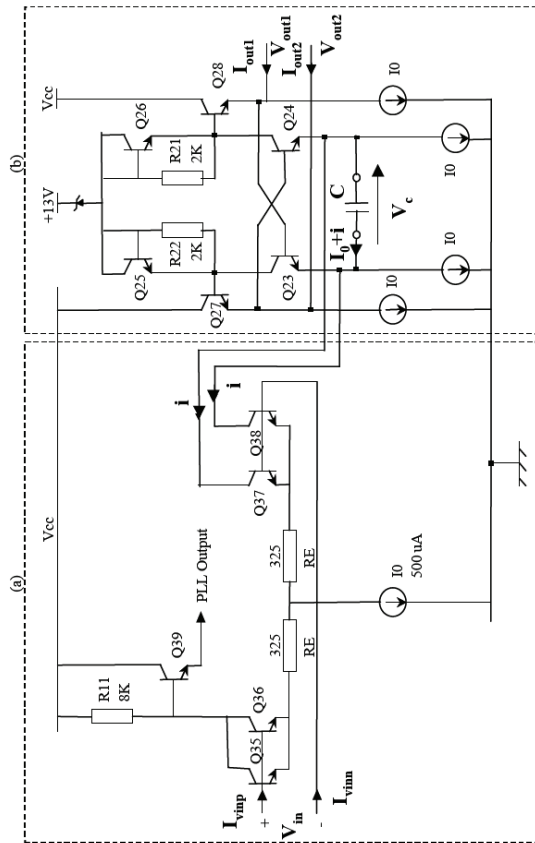


Tableau 5 : Modèle comportemental du bloc comparateur – équations et caractérisation

**bloc vco**

**Schéma transistor**



**Modèle comportemental**

```

module vco (inp, inn, outp, outn);
electrical inp, inn, outp, outn;
parameter real f0 = 100K, vdd=6.8, I0=500u,
RE=325, Beta1=125, Beta2=98;
real i,ic0, CI, ic,icc, c, vbe, vin, voutp,
voutn;
electrical vc;
analog begin
@ (initial_step) begin
CI = 0;
icc = 110u;
voutp = vdd - 2 * vbe;
voutn = vdd - vbe;
end
ic0=I0/4;
vbe = 0.854;
c = (I0+ic0)/(4*vbe*f0);
vin = V(inn) - V(inp);
/ input characteristic /
I(inp) <+ (I0/Beta1) / (1+exp(2*vin/(RE*I0)));
I(inn) <+ (I0/Beta2) / (1+exp(-2*vin/(RE*I0)));
/ transfer characteristic /
i = I0 / (2*(1+exp(-2*vin/(RE*I0))));
ic = I0 + i;
V(vc) <+ idt(icc/c, CI);
if (f0 > 100M)
begin
voutp = vdd - 2 * vbe;
voutn = vdd - vbe;
end
else
begin
if (icc >= 0)
begin
voutp = vdd - 2 * vbe;
voutn = vdd - vbe;
if (V(vc) >= vbe) icc = -ic;
end
if (icc < 0)
begin
voutp = vdd - vbe;
voutn = vdd - 2 * vbe;
if (V(vc) <= -vbe) icc = ic;
end
end
end
/ output characteristic /
V(outp) <+ voutp+3e9*I(outp)*I(outp)*I(outp) -
1e6*I(outp)*I(outp)+266*I(outp);
V(outn) <+ voutn+3e9*I(outn)*I(outn)*I(outn) -
1e6*I(outn)*I(outn)+266*I(outn);
end
endmodule

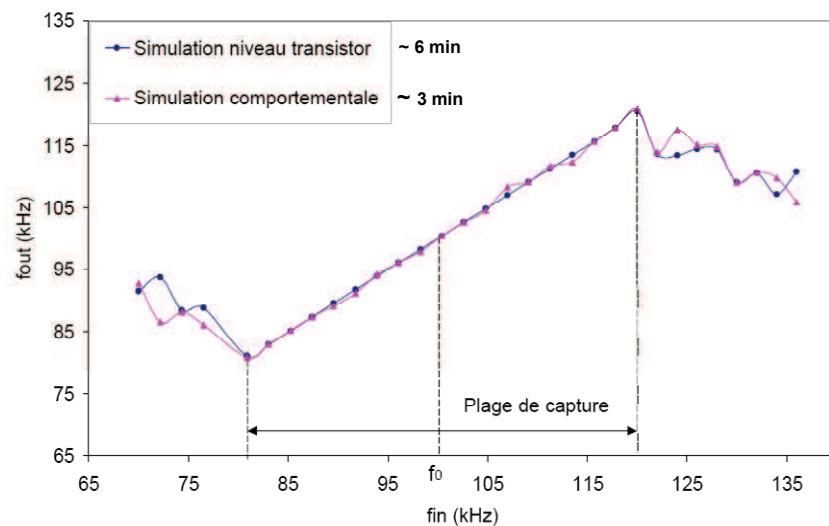
```



Tests comparatifs	
<p>Caractéristique de transfert : fréquence de sortie en fonction de la tension d'entrée</p>	

**Tableau 6 : modèle comportemental du bloc vco – équations et caractérisation**

En guise d'illustration du comportement du système, une simulation de la plage de capture de la PLL est proposée fig.8.



**Figure 8 : simulation de la plage de capture de la PLL – courbes et durée**

La reproduction des principales performances de la PLL est assurée par cette modélisation tout en gagnant un facteur 2 sur la durée de simulation.

### 2.1.2 Modèles basés sur l'identification

Un modèle extrait *par identification* (Approche « boîte noire » décrite par G. Gielen [Gie02]), s'appuie sur l'acquisition des caractéristiques externes du circuit, sans tenir compte de sa structure interne. Les données de caractérisation se présentent donc sous forme tabulaire et il s'agit de trouver une expression mathématique reproduisant le bon comportement, par une méthode d'interpolation.

Ce type de méthode a été expérimenté au cours de ma thèse dans le cas particulier de l'identification du comportement fréquentiel d'un amplificateur opérationnel (AOP) [MZD96]. A partir d'une simulation AC du schéma électrique de l'AOP, sa réponse fréquentielle est obtenue sous la forme d'un tableau de points. La méthode d'identification fréquentielle de Levy [LEV59] a été programmée dans Matlab et utilisée pour modéliser la fonction de transfert sous la forme

$$H(s) = \frac{A_0 + A_1s + A_2s^2 + \dots + A_ns^n}{B_0 + B_1s + B_2s^2 + \dots + B_ms^m}$$

ou bien sous forme factorisée faisant apparaître les pôles et zéros :

$$H(s) = \frac{(s - z_1)(s - z_2) \dots (s - z_n)}{(s - p_1)(s - p_2) \dots (s - p_n)}$$

Chacun de ces pôles et zéros a ensuite été modélisé par un sous-circuit SPICE, dans le cadre de la construction modulaire d'un macromodèle d'AOP. Cet aspect est développé au paragraphe **3.1.** (Tableau 8).

Une autre voie a été proposée en collaboration avec l'équipe de Marie-Minerve Louërat du département SOC du LIP6, au cours d'un stage de Master 2 encadré par R. Iskander [MED09]. Le but de ce stage était d'étudier les atouts de la technique des plans d'expériences, pour la génération automatique d'un modèle de performances d'OTA. Un outil d'exploration de performances a été développé selon le principe des plans d'expériences (dans ce cas, expérience = simulation) puis ces performances ont été modélisées par des équations polynômiales.

Le principal avantage de ces méthodes basées sur l'identification est qu'elles présentent un fort potentiel d'automatisation ; l'inconvénient est que les paramètres du modèle perdent leur sens physique, ce qui n'est pas le cas de la méthode « symbolique » précédente, basée sur la connaissance.

## **2.2. Modélisation en phase Top-Down**

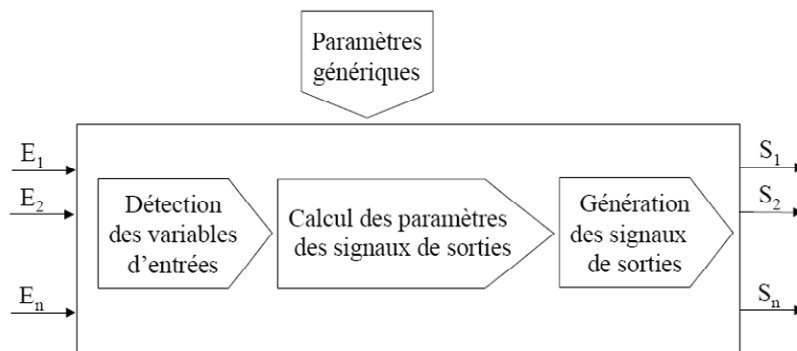
En phase de conception Top-Down, le point de départ est une liste de spécifications, le schéma électrique étant alors inconnu. Il s'agit donc de construire un modèle représentant la fonction du bloc considéré, réglable au moyen de paramètres de performances. Ce type de

modèle est utilisé en amont de la conception, pour poser les bases de l'architecture du système et la valider par une première série de simulations. Les spécifications du système sont alors distribuées sur chaque bloc. Ces modèles peuvent être qualifiés de *fonctionnels*, ils sont moins précis que les modèles *comportementaux* extraits en phase Bottom-Up ; ils doivent permettre d'explorer différentes architectures de système et de figer les spécifications de chaque bloc pour la phase ultérieure. Dans le compromis rapidité/précision des modèles, la rapidité est prédominante dans cette phase, ainsi on pensera à privilégier les descriptions dirigées par événements, quand c'est possible.

Dans sa thèse, A. Fakhfakh a réfléchi à une approche systématique pour le développement de modèles fonctionnels au moyen des langages VHDL-AMS et Verilog-A. Dans le cadre de la modélisation de circuits pour la synthèse de fréquence, il a établi une liste de sous-fonctions récurrentes, à partir desquelles peuvent être construits tous les modèles [FAK02].

### 2.2.1 Démarche systématique

Pour fixer les choses, cette démarche systématique est décomposée en 3 étapes comme illustré par la fig.9. Dans une première étape, les variables d'entrées du modèle sont détectées ; on appelle variable d'entrée une grandeur électrique caractéristique du signal d'entrée et qui intervient dans la description du comportement (cela peut être une tension, un courant, une fréquence instantanée, etc.). La deuxième étape consiste à calculer les paramètres caractéristiques des signaux de sorties, à partir des variables d'entrées et des paramètres génériques (il s'agit de l'expression de la « fonction » du circuit). Enfin, dans une dernière partie, on génère les signaux de sortie (par transformation des signaux d'entrée ou bien de façon autonome comme pour les oscillateurs).



**Figure 9 : structure d'un modèle fonctionnel**

Pour faciliter le développement de modèles fonctionnels selon cette construction modulaire, une bibliothèque de sous-fonctions VHDL-AMS ou Verilog-A est proposée par A. Fakhfakh dans sa thèse [FAK02], avec l'aide des travaux de DEA de G. Monnerie [MON01].

Les sous-fonctions du module d'entrée réalisent la détection de :

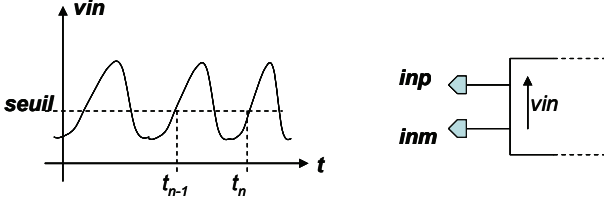
- tension ou courant
- front montant ou descendant
- durée d'impulsion
- fréquence instantanée.

Les sous-fonctions du module de sortie génèrent :

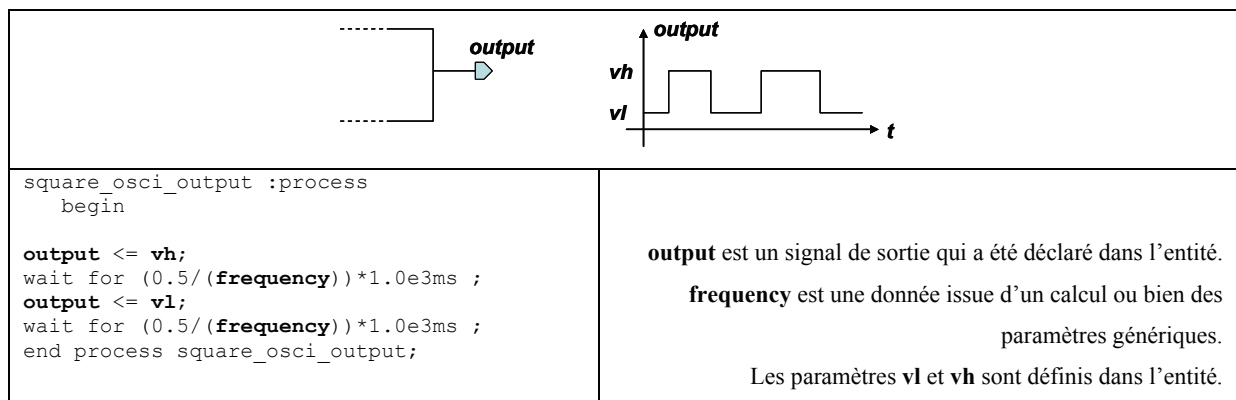
- un signal sinusoïdal d'amplitude et fréquence données
- un signal oscillateur carré d'amplitude et fréquence données.

La description du module central relevant de la fonction spécifique du circuit est laissée au soin de l'utilisateur. Dans le cas VHDL-AMS, les sous-fonctions d'entrée/sortie sont proposées soit pour un port analogique (*objet quantity*), soit pour un port numérique (*objet signal*).

A titre d'exemple, le listing list.1 est la description VHDL-AMS du module d'entrée « détecteur de fréquence », le listing list.2 celui du module de sortie « générateur d'oscillateur carré ».

	<p>Les ports <b>inp</b> et <b>inm</b> ont été définis dans l'entité.</p> <p>Mémorisation de l'instant du front précédent.</p> <p>Détection d'un front montant en entrée ; <b>seuil</b> est un paramètre générique défini dans l'entité.</p> <p>Mémorisation de l'instant du front actuel.</p> <p>Calcul de l'écart temporel au front montant précédent.</p> <p>Déduction de la fréquence ; elle sera mise à jour à chaque front montant. Précaution prise pour le 1<sup>er</sup> front ; <b>f0</b> est une fréquence « par défaut ».</p>
<pre>quantity vin across iin through inp to inm; signal frequency : real;  detection_frequence : process   variable tn, tn_1, period : real := 0.0; begin    tn_1 := tn;    wait until vin'above(seuil) = true;    tn := now;    period := tn - tn_1;    if period &lt; 0 then     frequency &lt;= f0;   else     frequency &lt;= 1.0/period;   end if;  end process detection_frequence;</pre>	

**List.1 : Code VHDL-AMS pour la détection de fréquence d'entrée**



**List.2 : Code VHDL-AMS pour la génération d'un signal carré**

Au moyen de ces 2 exemples, on peut notamment construire un modèle de diviseur de fréquence : il ne manque qu'à préciser la relation entre la fréquence détectée en entrée et celle générée en sortie. On peut ainsi décrire de manière abstraite les diviseurs fractionnaires, dont le rapport de division  $N$  ne serait pas un entier, avant de descendre dans la topologie. Le module oscillateur peut également servir à décrire un VCO, dans ce cas la fréquence de sortie sera reliée au niveau de tension appliqué en entrée, etc.

### 2.2.2 Application

Cette méthode a été appliquée au développement d'une bibliothèque de circuits typiquement utilisés en synthèse de fréquence. Le tableau suivant récapitule l'ensemble des modèles de cette bibliothèque.

<i>Comparaison de phase</i>	Comparateur phase-fréquence
	Pompe de charge
<i>Filtrage</i>	Filtre passe bas
	Filtre passe haut
	Filtre passe bande
	Filtre coupe bande
<i>Division de fréquence</i>	Diviseur $N$
	Diviseur $N/N+M$
	Accumulateur
	Diviseur fractionnaire
<i>Oscillateurs</i>	Oscillateur contrôlé en tension
	Oscillateurs synchrones

**Tableau 7 : bibliothèque de modèles pour la synthèse de fréquence**

A titre d'exemple, un synthétiseur de fréquence fractionnaire, en cours de développement dans l'équipe de *Conception RF* de Y. Deval, a été modélisé en assemblant

ces blocs, selon le schéma de la fig.10 ; ses spécifications correspondent à l'émission/réception de norme UMTS.

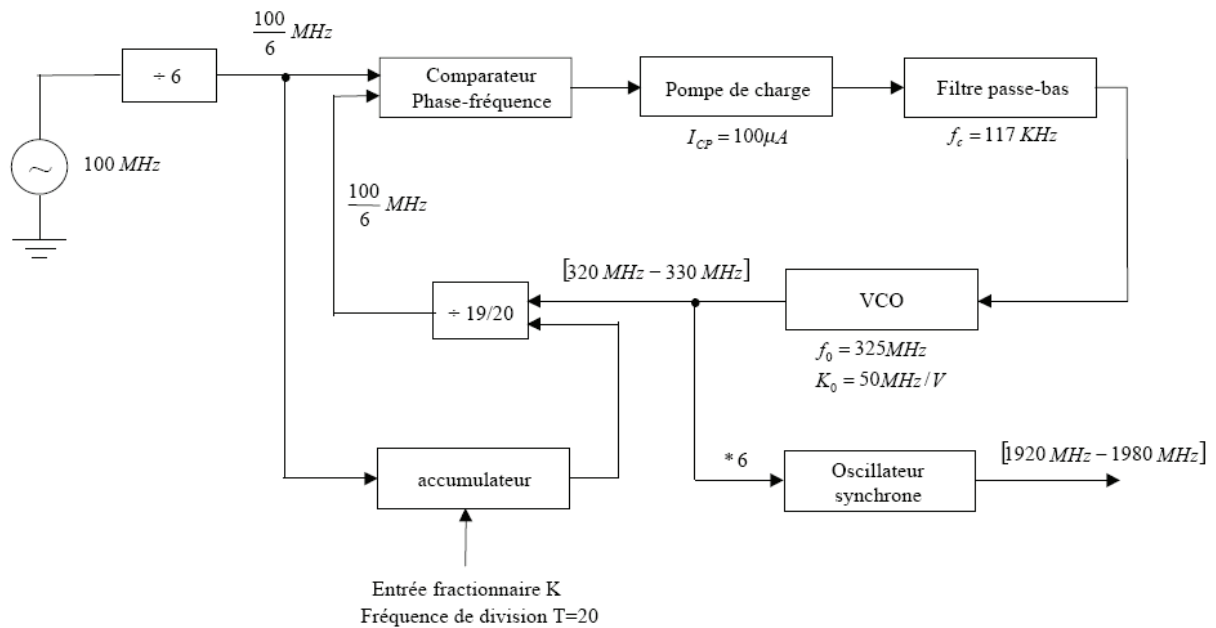


Figure 10 : architecture du synthétiseur de fréquence fractionnaire

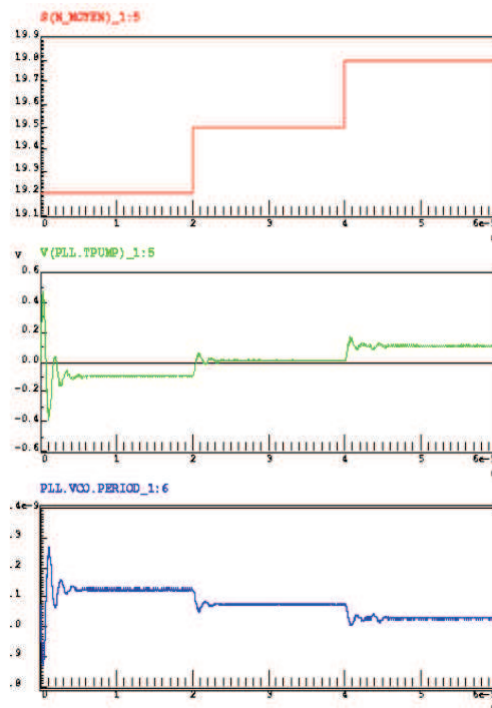


Figure 11 : Résultats de simulation du synthétiseur fractionnaire

La fig.11 illustre les réponses du modèle de synthétiseur fractionnaire suite à un changement de la valeur du paramètre K (4, 10, 16), ce qui induit un changement dans la valeur du rapport de division moyen  $N_{FRAC}$  (19.2, 19.5, 19.80). La première courbe montre la variation du rapport de division  $N_{FRAC}$  ; la deuxième illustre le courant à la sortie de la pompe

de charge (ce qui correspond à la sortie du filtre de boucle) et la troisième courbe montre la variation de la période de sortie du VCO suite au changement du rapport de division. Les résultats obtenus prouvent le bon fonctionnement du synthétiseur et donc confirme la bonne modélisation de chaque bloc.

### **3. Modèles multi-niveaux d'abstraction**

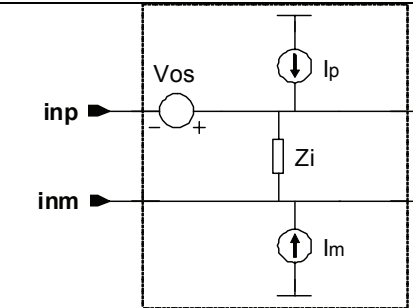
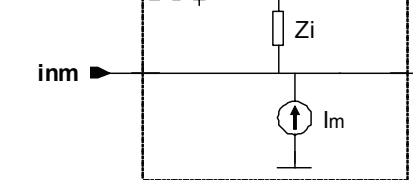
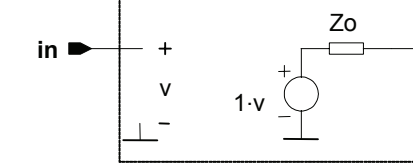
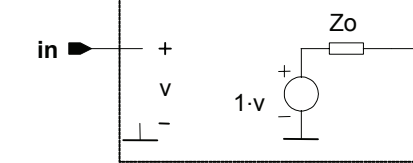
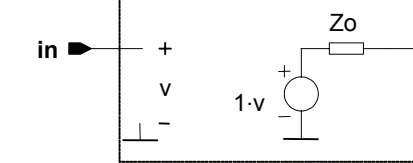
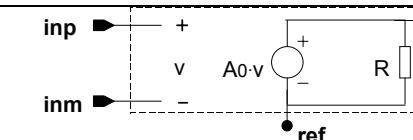
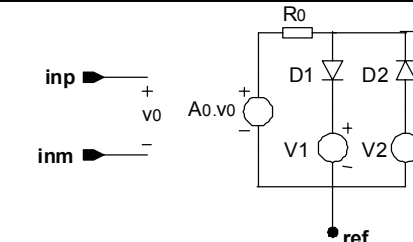
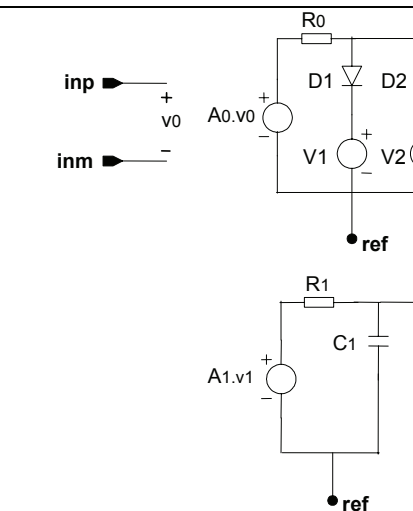
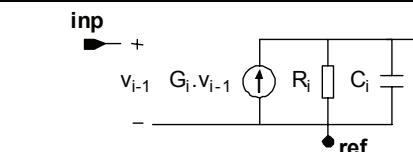
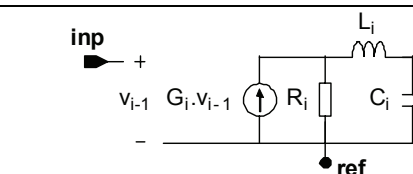
On a vu précédemment que la hiérarchisation de la conception d'un système était incontournable et que les modèles n'avaient pas les mêmes propriétés selon le niveau hiérarchique considéré. Afin d'assurer une certaine fluidité dans le flot de conception, nous avons imaginé différentes manières de concevoir des modèles multi-niveaux d'abstraction ; ainsi l'utilisateur pourra choisir le degré de précision « à la carte », en fonction des besoins dans le processus de conception. Cette question a été abordée au cours de ma thèse et de celle de G. Monnerie.

#### **3.1. Macro-modèle d'Amplificateur Opérationnel**

Les modèles exposés ici ont été développés durant ma thèse [LEW97], ce sont des macromodèles SPICE. Ces macromodèles sont construits de manière modulaire ; chaque *module*, est indépendant des autres et simule une seule performance. Cela permet de concevoir un macromodèle d'Amplificateur Opérationnel (AOP) prenant en compte uniquement les caractéristiques utiles du circuit et d'aboutir ainsi à un niveau de précision variable et ajusté au besoin. La procédure de réglage de ces modules ainsi qu'un plan de caractérisation de l'AOP ont été explicités dans ma thèse, c'est un point essentiel lorsque l'on voudra ensuite ajuster un modèle à des caractéristiques mesurées.

##### **3.1.1 Sous-circuits de base**

Il s'agit de sous-circuits SPICE qui serviront par la suite à décrire un modèle d'AOP multi-niveaux. Leurs caractéristiques sont résumées dans le tableau suivant.

module	schéma sous-circuit	caractéristiques	paramètres
ENTREE		- entrée différentielle - résistance d'entrée - courants de polarisation - tension d'offset	Ri Ip, Im Vos
ENTREE_RC		- entrée différentielle - résistance et capacité d'entrée - courants de polarisation - tension d'offset	Ri, Ci Ip, Im Vos
SORTIE		- suiveur de tension - résistance de sortie	Ro
SORTIE_RC		- suiveur de tension - résistance et capacité de sortie	Ro, Co
SORTIE_RL		- suiveur de tension - résistance et inductance de sortie	Ro, Lo
AMPLI		- gain en tension statique	A0
AMPLI_SAT		- gain en tension statique - tensions de saturation	A0 VH, VL
AMPLI_SAT_P		- gain en tension statique - tensions de saturation - pôle dominant	A0 VH, VL f
POLE		- gain statique unitaire - pôle	f
POLE_COMP		- gain statique unitaire - pôles complexes conjugués	f <sub>r</sub> , f <sub>c</sub>



ZERO		- gain statique unitaire - zéro	f
ZERO_POS		- gain statique unitaire - zéro à partie réelle positive	f

**Tableau 8 : Sous-circuits SPICE pour macromodèle d'AOP**

### 3.1.2 Générateur automatique de macromodèle

A partir des sous-circuits précédents, il est possible de générer automatiquement un modèle d'AOP de précision voulue et dont les paramètres sont ajustés selon des performances saisies par l'utilisateur. Le programme GENE\_AUTO a été développé en C à cet effet. Cinq niveaux de modélisation ont été prévus, du plus élémentaire au plus complet. Le tableau 9 en dresse la liste en associant à chaque niveau les caractéristiques simulées par le macromodèle.

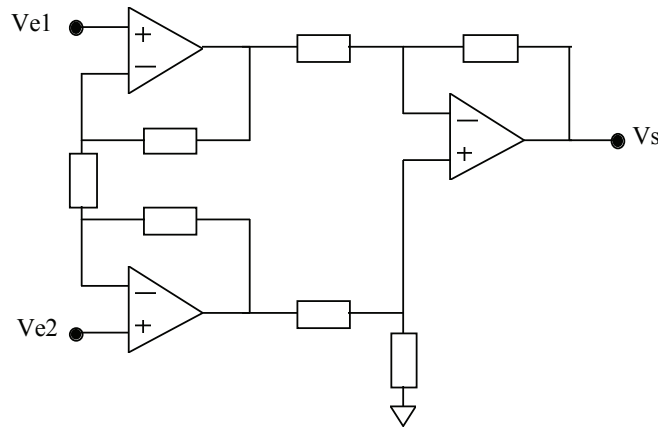
NIVEAU	caractéristiques simulées	
NIVEAU 1	<ul style="list-style-type: none"> <li>résistance d'entrée <b>R<sub>i</sub></b></li> <li>tension d'offset <b>V<sub>os</sub></b></li> <li>courants de polarisation <b>I<sub>p</sub></b> et <b>I<sub>m</sub></b></li> <li>résistance de sortie <b>R<sub>o</sub></b></li> </ul>	<ul style="list-style-type: none"> <li>gain différentiel statique <b>Ad0</b></li> <li>produit gain-bande passante <b>GB</b></li> <li>tensions de saturation <b>V<sub>H</sub></b> et <b>V<sub>L</sub></b></li> </ul>
NIVEAU 2	NIVEAU 1 +	
	<ul style="list-style-type: none"> <li>capacité d'entrée <b>C<sub>i</sub></b></li> <li>marge de phase <b>D<sub>phi</sub></b></li> </ul>	<ul style="list-style-type: none"> <li>slew-rate <b>SR<sub>p</sub></b> et <b>SR<sub>m</sub></b></li> <li>capacité (inductance) de sortie <b>C<sub>o</sub></b> (<b>L<sub>o</sub></b>)</li> </ul>
NIVEAU 3	NIVEAU 2 (excepté le paramètre <b>D<sub>phi</sub></b> ) +	
	<ul style="list-style-type: none"> <li>pôles et zéros haute fréquence <b>f<sub>p2</sub></b>, <b>f<sub>p3</sub></b>, ..., <b>f<sub>z1</sub></b>, <b>f<sub>z2</sub></b>, ...</li> </ul>	
NIVEAU 4	NIVEAU 3 +	
	<ul style="list-style-type: none"> <li>résistance d'entrée en mode commun <b>R<sub>ic</sub></b></li> <li>capacité d'entrée en mode commun <b>C<sub>ic</sub></b></li> <li>taux de réjection du mode commun statique <b>CMRR</b></li> </ul>	
NIVEAU 5	NIVEAU 4 +	
	<ul style="list-style-type: none"> <li>taux de réjection des alimentations <b>PSRRDD</b> et <b>PSRRSS</b></li> </ul>	

**Tableau 9: Différents niveaux du macromodèle d'AOP**

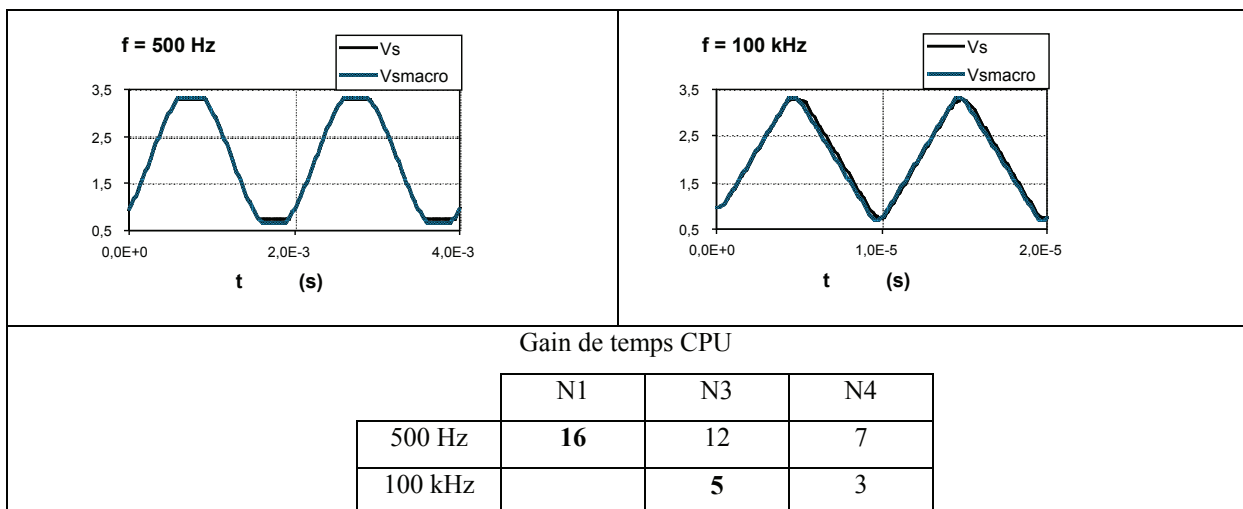
### 3.1.3 Exemples d'utilisation

Le macromodèle ainsi généré a été validé à travers différents exemples, notamment des amplificateurs d'instrumentation et de type chopper, architectures utilisant plusieurs AOP, mais on ne reprendra ici qu'une illustration de l'intérêt de choisir le niveau de modélisation en fonction du régime de fonctionnement estimé de l'AOP.

Ainsi, considérons l'amplificateur d'instrumentation représenté fig.12. L'AOP utilisé dans cette architecture est un circuit intégré full-custom réalisé au laboratoire par Y. Deval et J.B. Begueret. Il a été complètement caractérisé et son macromodèle extrait du Niveau 1 au Niveau 5. On applique alors une tension sinusoïdale différentielle à l'entrée du système et une étude comparative de la qualité et de la durée des simulations de type transitoire est effectuée. La fig.13 donne les principaux résultats [MZD97].



**Figure 12 : schéma électrique de l'amplificateur d'instrumentation**



**Figure 13 : résultats de simulation de l'amplificateur d'instrumentation**

La simulation du système pour  $f = 500$  Hz est suffisamment précise avec le macromodèle de Niveau 1 et le gain de temps est de 16 par rapport à une simulation au niveau

transistor. Les niveaux de modélisation 3 et 4 réalisent respectivement des gains de temps de 12 et 7 sans apporter de précision supplémentaire. Pour  $f = 100$  kHz, l'influence du slew-rate nécessite le passage au Niveau 2, avec un gain de temps de 5.

### 3.2. Modèles VHDL-AMS imbriqués

Les modèles présentés dans ce paragraphe suivent la même intention de modélisation multi-niveaux mais tirent profit du langage VHDL-AMS. Ils proviennent d'une idée originale de G. Monnerie qui va être développée ci-après ; son principe a été appliqué à une bibliothèque de modèles pour la simulation d'une chaîne de réception RF.

#### 3.2.1 Principe

L'objectif est toujours d'avoir un degré de précision variable et adapté au besoin, tout en gardant une compatibilité broche à broche avec le schéma électrique au niveau transistor. Les modèles sont composés de plusieurs couches imbriquées comme des *poupées russes*<sup>5</sup>. Chaque modèle fait appel à un sous-modèle de complexité moindre, comme illustré ci-dessous.

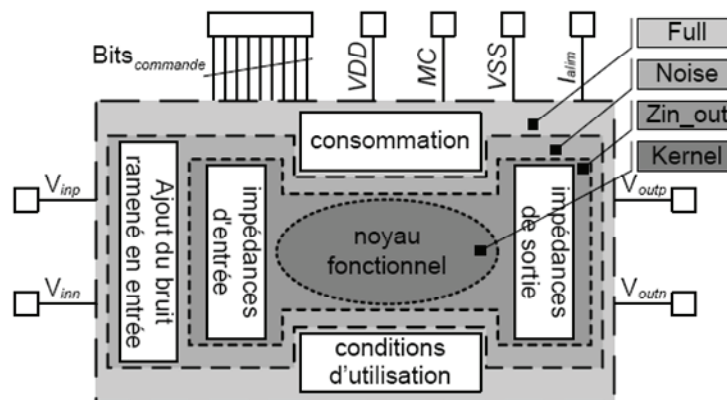


Figure 14 : Principe de construction d'un modèle *en poupées russes*

Le noyau fonctionnel *Kernel* regroupe les fonctions minimales du composant, il est utilisé pour les simulations globales de la chaîne de réception. Ce noyau est appelé par le modèle supérieur *Zin\_out* où sont prises en compte les impédances d'accès, lui-même appelé par un modèle *Noise* incluant le bruit ramené en entrée. Enfin, le niveau le plus complet, *Full*, fait appel à tous ces modèles imbriqués hiérarchiquement ; il rajoute la vérification de l'intervalle de validité des tensions/courants d'alimentation.

Le listing ci-après donne le code VHDL-AMS typique de ces architectures imbriquées.

<sup>5</sup> Terminologie originale de G. Monnerie.

<pre> ENTITY MODELE IS   GENERIC( noise : REAL := 0.8e-9; Tsmp : REAL := 1.0e-6);   PORT ( TERMINAL VDD, VSS, MC : ELECTRICAL;         TERMINAL Ialim : ELECTRICAL;         TERMINAL INP, INN : ELECTRICAL;         TERMINAL OUTP, OUTN : ELECTRICAL;         SIGNAL Commande : IN BIT_VECTOR (2 DOWNTO 0)); END ENTITY MODELE; </pre>	<p>Déf. des paramètres et connexions :</p> <ul style="list-style-type: none"> <li>- alimentation en tension et courant</li> <li>- entrée différentielle</li> <li>- sortie différentielle</li> <li>- bit de commande</li> </ul>
<pre> ----- FULL ----- ARCHITECTURE behaviour_full OF MODELE IS   QUANTITY V_alim ACROSS I_alim THROUGH VDD TO VSS;   QUANTITY V_mc ACROSS MC TO VSS;   QUANTITY V_bias ACROSS I_bias THROUGH Ialim TO VSS;   SIGNAL time : BIT := '0'; BEGIN  Supply_verif :PROCESS ( V_alim'ABOVE(2.9),V_alim'ABOVE(3.6),I_bias'ABOVE(80.0e-6), I_bias'ABOVE(120.0e-6),V_mc'ABOVE(v_alim*0.525), V_mc'ABOVE(v_alim*0.475),time)   BEGIN   IF NOW &gt; 0.0 THEN     ASSERT (v_alim &gt; 2.9)     REPORT "VOLTAGE SUPPLY UNDER 2.9V" SEVERITY WARNING;     ASSERT (v_alim &lt; 3.6)     REPORT "VOLTAGE SUPPLY OVER 3.6V" SEVERITY WARNING;     ASSERT (I_bias &gt; 80.0e-6)     REPORT "CURRENT SUPPLY TOO LOW" SEVERITY WARNING;     ASSERT (I_bias &lt; 120.0e-6)     REPORT "CURRENT SUPPLY TOO HIGH" SEVERITY WARNING;     ASSERT (V_mc &lt; v_alim*0.525)     REPORT "COMMON VOLTAGE SUPPLY TOO HIGH" SEVERITY WARNING;     ASSERT (V_mc &gt; v_alim*0.475)     REPORT "COMMON VOLTAGE SUPPLY TOO LOW" SEVERITY WARNING;   ELSE time &lt;= '1' AFTER 1fs;   END IF; END PROCESS;  I_bias ==V_bias/10.0e+3;  M : ENTITY MODELE (behaviour_noise)   GENERIC MAP ( noise =&gt;noise, Tsmp=&gt;Tsmp)   PORT MAP ( VDD=&gt;VDD, VSS=&gt;VSS, MC=&gt;MC, Ialim=&gt;Ialim,             INP=&gt;INP, INN=&gt;INN, OUTP=&gt;OUTP, OUTN=&gt;OUTN,             Commande =&gt; Commande); END ARCHITECTURE behaviour_full; </pre>	<p>Description de l'architecture de plus haut niveau <i>behaviour_full</i>.</p> <p>Vérification des alimentations. Utilisation d'un signal (time) décalé d'une 1fs pour que la vérification s'effectue après l'analyse dc initiale.</p> <p>Modélisation du courant consommé.</p> <p>Instanciation du modèle de niveau inférieur : réutilisation de la même entité et appel à l'architecture <i>behaviour_noise</i>.</p>
<pre> ----- NOISE ----- ARCHITECTURE behaviour_noise OF MODELE IS   TERMINAL INP_noise, INN_noise : ELECTRICAL;   QUANTITY Vp_noise ACROSS Ip_noise THROUGH INP TO INP_noise;   QUANTITY Vn_noise ACROSS In_noise THROUGH INN TO INN_noise;   SIGNAL White_n : REAL := 0.0;  BEGIN noise_addition : PROCESS   VARIABLE white : REAL_VECTOR (0 TO 4) := (11.0, 4862.0,   214.0, 546.0, 1105.0));   BEGIN   white := white_noise_Vrms(1.0/Tsmp,noise,white);   White_n &lt;= white(0);   WAIT FOR (Tsmp); END PROCESS noise_addition;  Vp_noise == +White_n'RAMP(Tsmp/10.0,Tsmp/10.0)/2.0; Vn_noise == -White_n'RAMP(Tsmp/10.0,Tsmp/10.0)/2.0;  M : ENTITY MODELE (behaviour_Zin_out)   GENERIC MAP ( noise =&gt;noise, Tsmp=&gt;Tsmp)   PORT MAP ( VDD=&gt;VDD, VSS=&gt;VSS, MC=&gt;MC, Ialim=&gt;Ialim,             INP=&gt;INP_noise, INN=&gt;INN_noise,             OUTP=&gt;OUTP, OUTN=&gt;OUTN, Commande =&gt; Commande); END ARCHITECTURE behaviour_noise; </pre>	<p>Description de l'architecture <i>behaviour_noise</i>, introduisant un bruit blanc ramené en entrée.</p> <p>Le principe de modélisation du bruit dans le domaine temporel est décrit en détail dans le chapitre 2.</p> <p>Instanciation du modèle de niveau inférieur : réutilisation de la même entité et appel à l'architecture <i>behaviour_Zin_out</i>.</p>

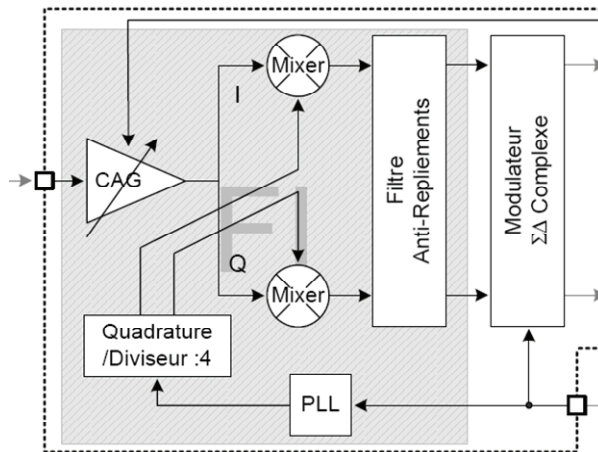
<pre> ----- Zin/Zout ----- ARCHITECTURE behaviour_Zin_out OF MODELE IS   CONSTANT Cin : REAL := 0.6e-12;   CONSTANT Cout : REAL := 0.8e-12;   CONSTANT Rout : REAL := 900.0;   QUANTITY Vin ACROSS Iin THROUGH INP TO INN;   QUANTITY Vinp ACROSS Iinp THROUGH INP TO electrical_ground;   QUANTITY Vinn ACROSS Iinn THROUGH INN TO electrical_ground;   QUANTITY Vout ACROSS Iout, Icout THROUGH OUTP TO OUTN; BEGIN    Iinp == Cin*Vinp'DOT;   Iinn == Cin*Vinn'DOT;   Iin == (Cin/2.0)*Vin'DOT;   Icout == Cout *Vout'DOT;   Vout == Rout *Iout;  M : ENTITY MODELE (behaviour_kernel)   GENERIC MAP ( noise =&gt;noise, Tsmpl=&gt;Tsmpl)   PORT MAP ( VDD=&gt;VDD, VSS=&gt;VSS, MC=&gt;MC, Ialim=&gt;Ialim,     INP=&gt;INP, INN=&gt;INN, OUTP=&gt;OUTP, OUTN=&gt;OUTN,     Commande =&gt; Commande); END ARCHITECTURE behaviour_Zin_out; </pre>	<p>Description de l'architecture <i>behaviour_Zin_out</i>, introduisant les impédances d'accès.</p> <p>Modélisation des impédances d'accès.</p> <p>Instanciation du modèle de niveau inférieur : réutilisation de la même entité et appel à l'architecture <i>behaviour_kernel</i>.</p>
<pre> ----- KERNEL ----- ARCHITECTURE behaviour_kernel OF MODELE IS BEGIN   --- a completer selon les circuits considérés --- END ARCHITECTURE behaviour_kernel; </pre>	

**List.3 : Code VHDL-AMS d'un modèle multi-niveaux à architectures imbriquées**

Il s'agit d'une syntaxe très originale et compacte, apte à couvrir les différents besoin en modélisation tout au long du flot de conception.

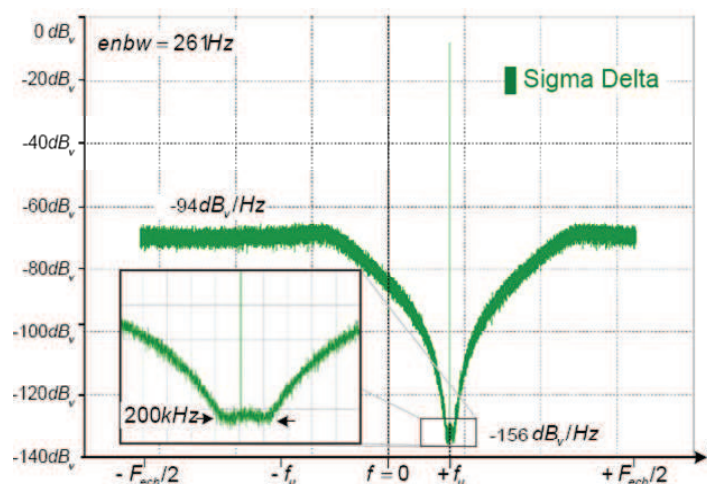
### 3.2.2 Application

Ce principe de modélisation VHDL-AMS a été appliqué à une bibliothèque de circuits servant à simuler une architecture de récepteur RF, développé par la société EADS – TELECOM et sur laquelle G. Monnerie devait évaluer l'influence de différentes sources de bruit. Le schéma de cette architecture, de type conversion à faible fréquence intermédiaire (*Low-IF*), polyphasée [GU05], est représenté fig.15. Les caractéristiques précises de ce récepteur se trouvent au début du chapitre 2. Notre intérêt s'est porté sur les blocs suivants : amplificateur à gain contrôlé (CAG), 2 mélangeurs (Mixer), un filtre anti-repliement, un générateur d'horloges en quadrature lui-même commandé par une PLL ainsi qu'un convertisseur  $\Sigma\Delta$  passe-bande. Les modèles précis de ces blocs sont en annexe de la thèse de G. Monnerie [MON05], ils ont été utilisés, à différents niveau de précision, pour simuler une partie ou bien l'ensemble de la chaîne de réception. A titre d'exemple, nous montrerons un résultat de simulation concernant le modulateur  $\Sigma\Delta$  [LMS04].



**Figure 15 : Partie de l'architecture du récepteur RF à étudier**

Ce modulateur est du second ordre et traite simultanément les voies I et Q. Il comporte deux paires de CAN – CNA, un filtre à capacités commutées complexe et un générateur d'horloges. Chacun des blocs a été caractérisé au niveau transistor, notamment en terme de bruit équivalent ramené en entrée. Cela permet d'avoir une estimation théorique du bruit en entrée du modulateur égale à 156 dB/Hz. Signalons que ce modulateur est à temps discret et qu'il est prohibitif de le simuler au niveau transistor (~ une journée de simulation pour obtenir quelques périodes en sortie). Une simulation du modulateur a donc été menée avec ses sous-blocs décrits en VHDL-AMS ; dans un 1<sup>er</sup> temps, chaque bloc était muni de sa propre contribution de bruit (modèle *behaviour\_noise*) puis dans un 2<sup>ème</sup> temps, ces blocs ont été modélisés sans leur contribution de bruit (modèle *behaviour\_Zin\_out*) alors que ce bruit était globalement ramené en entrée du modulateur. Une analyse temporelle sur 1048576 échantillons, suivie d'une analyse spectrale par périodogramme (131072 sous-fenêtres entrelacées à 50%, Blackman7) donne le résultat de la fig.16.



**Figure 16 : Simulation du spectre en sortie du modulateur ΣΔ**

Ce résultat de simulation montre un comportement attendu, soit une élévation du plancher de bruit du modulateur jusqu'à un niveau de 156 dB/Hz, ce qui a ensuite été corroboré par les mesures réalisées sur la puce.

#### **4. Conclusion**

Ce chapitre propose une synthèse de mes travaux personnels ou d'encadrement sur le thème de la modélisation des circuits analogiques, vue sous l'aspect méthodologique. Le but n'est pas simplement de développer des bibliothèques de modèles applicatives mais de définir des méthodes qui sont généralisables. Cette volonté de formalisme permet deux choses : d'enseigner ces méthodes à de futurs ingénieurs ou chercheurs<sup>6</sup> et aussi de concevoir ou de poser les bases d'outils automatiques visant à optimiser le flot de conception analogique.

Ces travaux ont été valorisés par deux articles et de nombreuses conférences. Certains illustrent l'aspect méthodologique appliqué à des circuits RF [ZMF99], [MIZ99], [FML00], [MMF01], [FLL02] ou neuromorphiques [MCL99], [MCZ99]; d'autres abordent la modélisation multi-niveaux de l'AOP [MZD94], [MZD97] [MZD96]. Cette approche de modélisation modulaire s'avère propice à la *réutilisation* d'un modèle dans des applications variées et ce thème sera davantage développé dans le chapitre 3 (thèse de T. Levi [LEV07]). D'autre part, dans le domaine RF, un des circuits clés est l'oscillateur et des questions spécifiques de modélisation ont été abordées pour ce circuit : un modèle d'Oscillateur Synchrone a été développé sur la base des travaux de thèse de F. Badets [FMB02] et des études préliminaires sur l'introduction du bruit dans ces modèles ont été menées [FMB02], [FMD01]. Ce thème de modélisation du bruit sera davantage développé au chapitre 2 (thèse de G. Monnerie [MON05]).

---

<sup>6</sup> Un transfert vers la formation a été réalisé à partir de ces travaux sous la forme d'une initiation à la conception hiérarchique et au langage VHDL-AMS dispensée en Master 2 EEA et en Licence Pro Microélectronique/Microsystèmes, sous une forme simplifiée.

# MODELISATION DE SOURCES DE BRUIT

## POUR SIMULATION TEMPORELLE

### 1. Introduction

#### 1.1. Contexte de l'étude

Ce chapitre fait état d'un travail initié par Guillaume Monnerie lors de sa thèse (thèse CIFRE IXL/EADS-TELECOM, 2002-2005 [MON05]). La question abordée est celle de la prise en compte du bruit lors de la simulation d'un système mixte. Plus précisément, le cadre posé par le partenaire industriel est la conception de la chaîne de réception RF d'un mobile à usage professionnel (*PMR : Private or Professional Mobile Radio*)<sup>1</sup> et l'évaluation de l'impact de différentes sources de bruit sur une architecture de récepteur, dans sa partie implémentée en temps discret. L'architecture retenue par la société EADS - TELECOM est de type conversion à faible fréquence intermédiaire (*Low-IF*), polyphasée [GU05], son schéma est représenté fig.1.

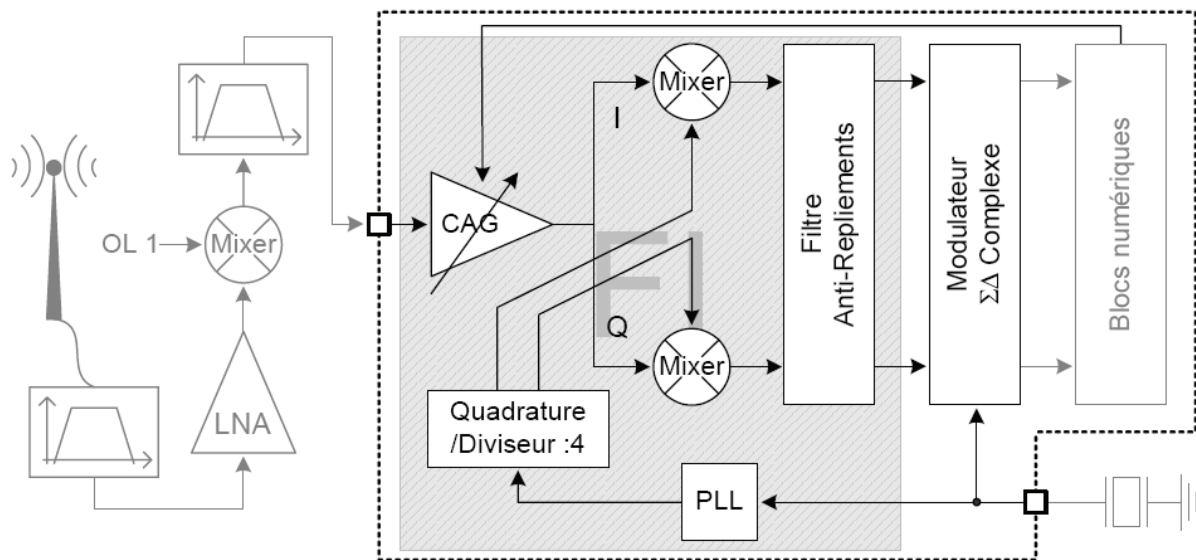


Figure 1 : Schéma de principe de la chaîne de réception RF étudiée

<sup>1</sup> Le sigle *PMR* désigne un ensemble de réseaux et systèmes de radiocommunication spécialement adaptés à un usage professionnel (compagnies de taxi, police, pompiers). Les caractéristiques particulières de ces réseaux sont notamment la communication « point à multi-point », de vastes domaines couvertures, l'utilisation des bandes de fréquences VHF ou UHF.



Les circuits encadrés en pointillés sont intégrés sur un ASIC qui a servi de démonstrateur pour la thèse.

L'architecture à faible fréquence intermédiaire est couramment utilisée et constitue une étape vers la radio logicielle dont l'idéal est une architecture entièrement reconfigurable, obtenue en ramenant le traitement numérique le plus près possible de l'antenne. Dans l'architecture étudiée (fig.1), la succession de deux mélangeurs, l'un RF (Mixer +  $OL_1$ ) et l'autre polyphasé (Mixer, I, Q + PLL), ramène le signal RF autour d'une porteuse à faible fréquence. Un filtre passe-bande complexe joue le rôle de filtre anti-repliement et le convertisseur analogique numérique (CAN) est de type  $\Sigma\Delta$  passe-bande complexe, à temps discret, et le reste du traitement est effectué numériquement.

A l'entrée de la puce, la fréquence porteuse appliquée à un amplificateur à gain contrôlé (CAG) est de 73.125MHz. La fréquence du second oscillateur local (sortie de PLL divisée par 4) est de 71.5MHz, si bien que les signaux en quadrature de phase sont à la fréquence 1.625MHz. La conversion analogique/numérique s'effectue à la fréquence d'échantillonnage de 13MHz, soit 8 fois la fréquence du signal d'entrée.

## 1.2. Etat de l'art

Quels sont les approches traditionnelles et verrous rencontrés pour la simulation de l'influence du bruit sur un tel circuit ?

Pour un circuit analogique, l'évaluation du bruit dans un contexte linéaire peut-être menée au moyen d'une simulation de type *noise analysis*, accessible dans tout simulateur analogique standard. Il s'agit en fait d'une extension de l'analyse fréquentielle linéaire (*AC analysis*), prenant en compte les paramètres des sources de bruit présentes dans le modèle de chaque transistor. Ce type de simulation est utile pour évaluer le bruit d'un circuit analogique de taille restreinte, typiquement un amplificateur fonctionnant autour d'un point de polarisation. Dans une chaîne de réception RF, seul l'amplificateur faible bruit (*LNA*) peut être analysé de cette façon.

Pour mieux évaluer les caractéristiques des circuits et systèmes radiofréquence, les simulateurs analogiques ont été pourvus de nouveaux algorithmes d'analyse appropriés. Dans SpectreRF par exemple [CADRF], on trouve l'analyse *PNOISE* (*Periodic Noise analysis*) qui est utile pour évaluer le bruit de circuits non-linéaires, tels que les mélangeurs ou les oscillateurs dans certaines conditions. Il faut en effet que ces circuits aient un point de polarisation qui évolue de manière périodique [KUN03] ; ils produisent alors un bruit qualifié

de cyclo-stationnaire<sup>2</sup> [PHK00]. Ceci n'est pas le cas pour un modulateur  $\Sigma\Delta$  [TEK99]. Pour les oscillateurs, qu'ils soient de référence ou asservis dans une PLL, le bruit de phase est un paramètre important à simuler car il dégrade les performances du système. Le bruit de phase relève d'un processus cumulatif (non cyclo-stationnaire) produisant une dérive permanente de la phase et l'analyse *PNOISE* est en général inadéquate pour la simulation du bruit de phase proche de la fréquence centrale [KUN03]. Dans le meilleur des cas, cette analyse pourra servir de caractérisation du bruit d'un bloc afin de paramétrer un modèle comportemental.

Notre système étant globalement mixte, incluant un modulateur  $\Sigma\Delta$  à temps discret, et compte-tenu des difficultés rencontrées dans les outils de simulation conventionnels, nous sommes intéressés aux techniques possibles de simulation temporelle du bruit.

Les travaux de Demir *et al.* [DLS96] vont dans ce sens. Les différentes sources de bruit<sup>3</sup> affectant les composants de la microélectronique sont prises en compte ; elles entrent habituellement dans le modèle petit signal de ces composants, Demir les adapte pour qu'elles puissent être intégrées au modèle grand signal et entrer en jeu dans les simulations temporelles d'un simulateur analogique standard (SPICE). Il propose alors une modification importante des algorithmes de l'analyse transitoire de SPICE<sup>4</sup>, sur la base de la théorie des équations différentielles stochastiques. Demir recrée ainsi un nouveau simulateur capable d'évaluer le bruit pris en compte au niveau transistor ; en fin de simulation, les propriétés statistiques de chaque potentiel de nœud sont fournies, et ce en fonction du temps. L'énorme inconvénient de cette technique est qu'elle n'est accessible qu'à un utilisateur expert capable d'arranger les algorithmes internes de SPICE. D'autre part, puisqu'il s'agit d'une simulation au niveau transistor, elle n'est réalisable que sur des circuits de petite taille et ne conviendrait pas à l'étude d'un récepteur RF entier. Nous retenons cependant un point intéressant qui concerne la possibilité de synthétiser un bruit en  $1/f$  à partir d'un bruit blanc et de la somme de plusieurs spectres Lorentziens, dont les pôles sont distribués uniformément sur un intervalle de fréquence donné.

Une approche convenant davantage à notre application concerne la modélisation de circuits bruités en régime temporel mais à partir d'une description comportementale. C'est

---

<sup>2</sup> Un bruit cyclo-stationnaire a des propriétés statistiques ne variant pas d'un cycle à l'autre, cela se produit par exemple lorsqu'un grand signal est appliqué à un circuit non-linéaire.

<sup>3</sup> Bruits thermique, de grenaille et en  $1/f$ .

<sup>4</sup> Version libre du simulateur fondateur développé à l'Université de Berkeley, téléchargeable sur internet.

ainsi que certaines idées de Demir ont été reprises par Kundert, pour la modélisation du bruit des oscillateurs ou PLLs [KUN06].

Le bruit dans les oscillateurs a pour conséquence une déviation aléatoire de la phase. On parle de bruit de phase lorsque l'on s'intéresse aux caractéristiques fréquentielles de ce phénomène ou de jitter lorsque l'on caractérise l'incertitude des instants de transition d'un signal oscillant binaire. Kundert fait d'abord une synthèse théorique des diverses caractéristiques du bruit dans les oscillateurs ou PLLs, puis propose des méthodes pour extraire ces caractéristiques sur les blocs d'une PLL, au moyen de simulateurs standards (Spectre, SpectreRF). Enfin, une technique de modélisation du bruit en régime temporel (jitter) est proposée et implémentée dans le langage de description comportemental analogique de Cadence, Verilog-A. La modélisation comportementale permet d'imposer directement les instants de transition du signal de sortie d'un bloc. Ces instants sont ainsi soumis à une déviation issue d'une séquence aléatoire gaussienne. Les modèles d'oscillateurs bruités développés dans la thèse de G. Monnerie s'inspirent en partie de ce principe. Notons que les bruits considérés dans les exemples sont des bruits blancs gaussiens et aucune piste n'est donnée pour la modélisation du bruit en  $1/f$ , que nous tenons à inclure dans nos modèles.

En ce qui concerne la génération d'un bruit discret ayant une densité spectrale en  $1/f$ , différentes approches existent dans la littérature [KAS95], [GUG01] : qu'il s'agisse de voies passant par l'étude de processus aléatoires particuliers<sup>5</sup> ou par la recherche de systèmes linéaires différentiels aptes à produire un bruit de comportement spectral en  $1/f$ , elles ont en commun d'être le résultat d'agrégation de cellules élémentaires du 1<sup>er</sup> ordre, à coefficients constants ou non. C'est cette approche qui a été choisie, car plus proche de notre culture d'électronicien. De plus, comme la problématique rejoint le cadre de la dérivation non-entière, l'expertise de l'équipe CRONE<sup>6</sup> de l'IMS a facilité certains développements décrits ci-après. Dans le contexte de la simulation des circuits et systèmes RF, deux références utilisent une approche similaire [PMR03], [SFB05].

### **1.3. Résumé des objectifs et moyens d'étude**

Ayant d'analyser les performances d'un système très complexe, en particulier le rapport signal à bruit de toute la chaîne de réception, la modélisation comportementale de chaque bloc

---

<sup>5</sup> Comme les processus fractionnaires Browniens ou les processus auto-similaires.

<sup>6</sup> Commande Robuste d'Ordre Non-Entier.

est indispensable. Cette modélisation devra prendre en compte les effets des bruits de sources variées et aboutir à une représentation simulable temporellement, naturellement adaptée au traitement en signaux échantillonnés. Les modèles des blocs de la chaîne sont développés en VHDL-AMS, les modèles de sources de bruit également. Notons qu'en ce qui concerne les sources de bruit, le sous-ensemble VHDL suffit dans la plupart des cas. Les paramètres des modèles peuvent être extraits de mesures ou de simulations bas-niveau effectuées sur chaque bloc individuellement. Les relations entre paramètres du modèle et paramètres mesurables sont données systématiquement.

## 2. Méthode

### 2.1. Principe et hypothèses

#### 2.1.1 Décomposition asymptotique de la densité spectrale de bruit

Le bruit en électronique provient de mouvements aléatoires microscopiques ayant différentes origines [GHL01]. La physique sous-jacente à ces phénomènes n'est pas ici notre propos mais plutôt les caractéristiques mesurables de ces bruits. L'électronicien est particulièrement attaché aux caractéristiques fréquentielles d'un bruit  $x(t)$ , visibles sur sa densité spectrale de puissance  $S_X(f)$ . Les dispositifs électroniques sont sujets, principalement, aux bruits blancs (bruit thermique, bruit de grenaille) et au bruit en  $1/f$  (*flicker noise*). Lorsque ces dispositifs (transistors, composants passifs) sont arrangés pour former un circuit, ces différents bruits se combinent. Par exemple, dans un circuit oscillateur, les contributions de type bruit blanc et bruit en  $1/f$  des transistors subissent globalement une intégration et donnent un spectre de puissance présentant des portions en  $1/f^3$  et  $1/f^2$  autour de la fréquence fondamentale [KUN06]. La méthode de modélisation du bruit développée ici part donc de l'hypothèse que tout spectre de bruit peut être décomposé en régions adjacentes de la forme  $S(f) = 1/f^n$ ,  $n$  étant un entier positif.

On cherchera ensuite à reproduire un tel spectre de bruit en additionnant les contributions de trois densités spectrales élémentaires  $S_0(f)$ ,  $S_1(f)$  et  $S_2(f)$  comme indiqué fig.2.

Initialement, il revient donc à l'utilisateur de fournir une telle approximation asymptotique du spectre qu'il souhaite modéliser. Cela n'a pas été utilisé ici mais un algorithme d'identification de fonctions de transfert permettrait d'effectuer cette étape automatiquement. Si les densités spectrales  $S_0(f)$ ,  $S_1(f)$  et  $S_2(f)$  sont produites à partir de sources aléatoires indépendantes, on trouvera une élévation de 3 dB aux fréquences de coupures, en sommant toutes les contributions.

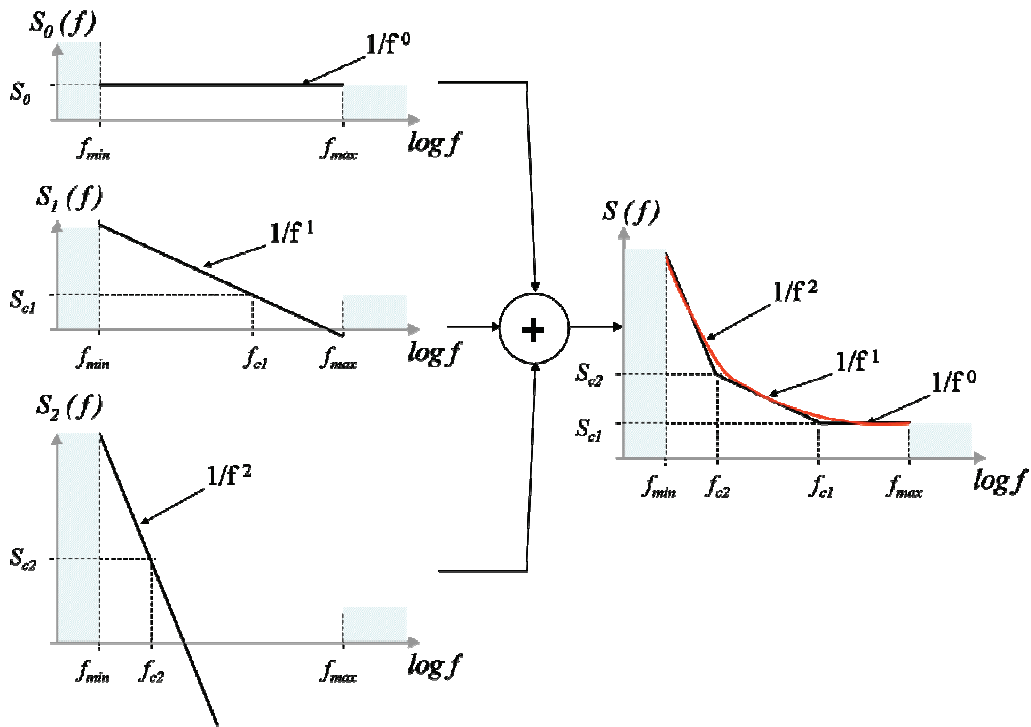


Figure 2 : Décomposition asymptotique d'un spectre de bruit

### 2.1.2 Génération d'un bruit en $1/f^n$ avec n entier positif

On peut obtenir un bruit de densité spectrale  $S_n(f) = S_0/(f^n)$  en appliquant un bruit blanc  $S_0$  à un filtre de fonction de transfert  $H_n(p) = (1/p)^{n/2}$ . Cela est illustré fig.3.

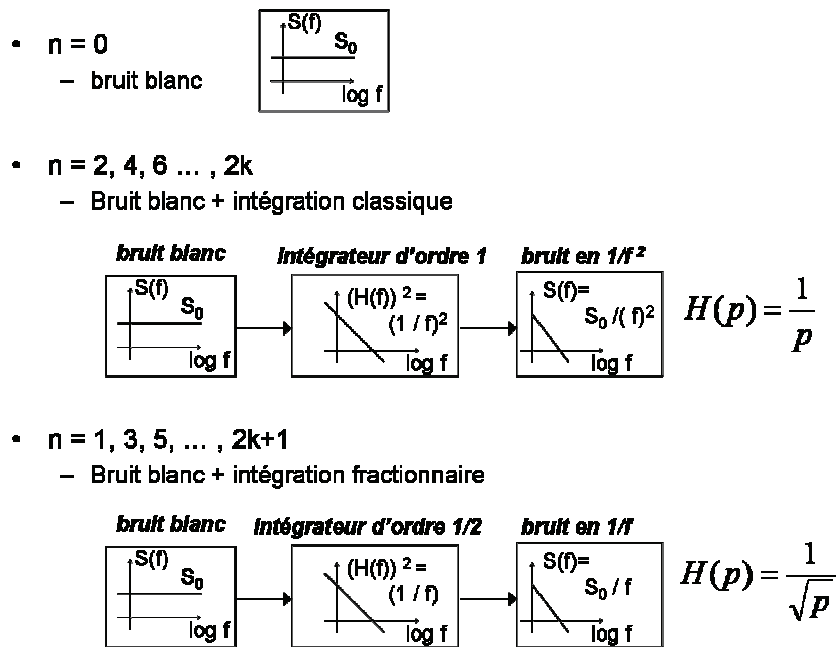


Figure 3 : Principe de génération de bruit en  $1/f^n$ , pour  $n = 0$ , n pair et n impair

### Le cas n = 0 :

Il correspond à une source de bruit blanc de densité spectrale  $S_0$ . Ce type de bruit peut être reproduit au moyen d'un générateur pseudo-aléatoire comme on en trouve classiquement dans les bibliothèques de langage de programmation ou de modélisation. Ils délivrent des séquences de nombres indépendants de répartition gaussienne ou uniforme, selon les cas. On préférera un générateur aléatoire de répartition gaussienne, centrée en 0 et d'écart-type  $\sigma$ , parce que cette densité de probabilité est plus souvent caractéristique des signaux physiques et parce que la relation de passage entre loi de probabilité et densité spectrale de puissance est simple ( $\sigma^2$  représente la puissance moyenne totale du signal et donc  $S_0 \Delta f$  si  $\Delta f$  est la bande de fréquence utile). Toutefois, rien n'empêche de prendre pour base un bruit blanc issu d'autres types de générateur aléatoire.

Dans notre application de signaux échantillonnés, le générateur aléatoire gaussien sera cadencé à la fréquence d'échantillonnage du système,  $f_e$ . Ainsi, si l'on veut un bruit blanc de niveau  $S_0$  (en dB par Hz), sur l'intervalle de Nyquist  $[-f_e/2, f_e/2]$ , il faudra ajuster l'écart-type selon la formule :

$$\sigma = \sqrt{f_e 10^{\frac{S_0 \text{ dB/Hz}}{10}}} \quad (1)$$

### Le cas n pair :

Comme le décrit la fig.3, lorsqu'une source de bruit blanc de densité spectrale  $S_0$  est filtrée par un intégrateur d'ordre 1, soit de fonction de transfert  $H(p) = 1/p$ , on obtient une source de bruit de densité spectrale  $S(f) = S_0 |H(p)|^2$ . Par conséquent, une cascade d'intégrateurs d'ordre 1 doit permettre de modéliser tous les cas où n est pair.

### Le cas n impair :

Selon le même principe, pour obtenir une source de bruit en  $1/f$ , c'est un intégrateur d'ordre  $1/2$  qu'il faut. Les autres cas, n impair  $\geq 3$ , peuvent alors être construits en ajoutant des intégrateurs simples.

Les opérateurs disponibles en VHDL-AMS, ainsi que dans la plupart des langages, concernent la dérivation et l'intégration entières. C'est ici que l'on rejoint la problématique de la synthèse d'un filtre d'ordre fractionnaire, bien connue du département LAPS de notre laboratoire [OLM00, OUS95].

## 2.2. Synthèse de l'intégrateur d'ordre 1/2 - Application au bruit en 1/f

Le problème posé est celui de la synthèse d'un filtre de la forme

$$H(p) = \frac{A}{\sqrt{p}}, \text{ au moins sur un intervalle de fréquence } [f_{min}; f_{max}].$$

En pratique, cela peut-être obtenu par une fonction de transfert telle que  $H_i(p)$  :

$$H_i(p) = H_0 \frac{\left(1 + \frac{p}{\omega_H}\right)^{1/2}}{\left(1 + \frac{p}{\omega_L}\right)^{1/2}} \text{ avec } \begin{cases} \omega_H = 2\pi f_H \\ \omega_L = 2\pi f_L \end{cases}$$

D'un point de vue asymptotique, dans l'intervalle  $[f_L, f_H]$  on a :

$$H_i(p) \approx H_0 \frac{1}{\left(\frac{p}{\omega_L}\right)^{1/2}} \approx \frac{H_0 \sqrt{\omega_L}}{\sqrt{p}}$$

La synthèse de la fonction de transfert  $H_i(p)$  au moyen d'opérateurs de dérivée entière repose sur une cascade récursive de N filtres élémentaires qui aboutit à la fonction de transfert  $H_r(p)$  :

$$H_r(p) = H_0 \frac{\prod_{j=1}^N \left(1 + \frac{p}{\omega'_j}\right)}{\prod_{j=1}^N \left(1 + \frac{p}{\omega_j}\right)} \quad \lim_{N \rightarrow \infty} H_r(p) = H_i(p)$$

Les pôles  $\omega_j$  et zéros  $\omega'_j$  sont répartis selon une distribution récursive comme le développe la théorie sous-jacente [OLM00, OUS95], afin que  $H_r(p)$  approche au mieux  $H_i(p)$ . Une illustration est fournie fig.4. Le résultat caractéristique de cette technique est la présence d'ondulations plus ou moins marquées, sur le module comme la phase de  $H_r(p)$ . Plus le nombre de cellules N est grand, meilleure est l'approximation, toutefois de trop grandes valeurs pourraient aboutir à des temps de simulation trop longs et à l'utilisation de grandes ressources mémoire.

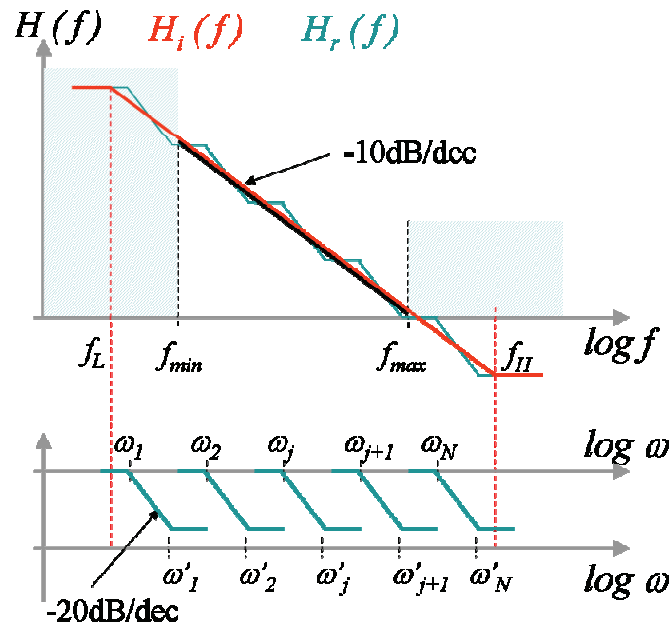


Figure 4 : Principe de la synthèse de l'intégrateur d'ordre  $\frac{1}{2}$

La synthèse du filtre consiste alors à calculer les paramètres  $\omega_j$ ,  $\omega'_j$ ,  $\omega_H$ ,  $\omega_L$ ,  $H_0$  et  $N$ , étant données les spécifications du bruit en  $1/f$  à générer, soient son intervalle de validité  $[f_{min}, f_{max}]$  et les coordonnées d'un point  $(f_{cl}, S_{cl})$  dans cet intervalle.

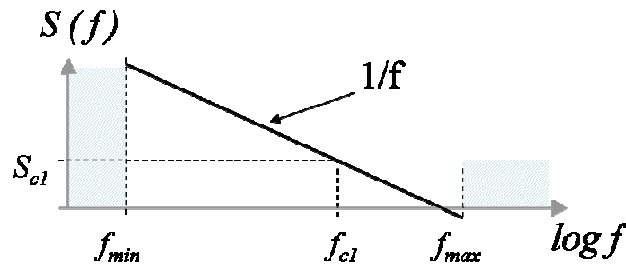


Figure 5 : Caractéristiques spectrales du bruit à générer

Pour s'assurer d'avoir le comportement souhaité sur  $[f_{min}; f_{max}]$ , et éviter les effets de bord aux fréquences  $f_H$  et  $f_L$ , on prend une marge confortable d'une décade soit :

$$f_H = 10f_{max} \quad (2)$$

$$f_L = \frac{f_{min}}{10} \quad (3)$$

$N$  est choisi par l'utilisateur selon le même critère d'erreur ; en pratique, dans nos exemples où l'intervalle  $[f_{min}; f_{max}]$  couvre 3 décades, les valeurs allant de 5 à 7 ont suffi.

Le calcul des pôles  $\omega_j$  et zéros  $\omega'_j$  se fait par les formules de récurrence suivantes [OUS95] :

$$\gamma = \left( \frac{\omega_L}{\omega_H} \right)^{\frac{1}{2N}} \quad (4)$$



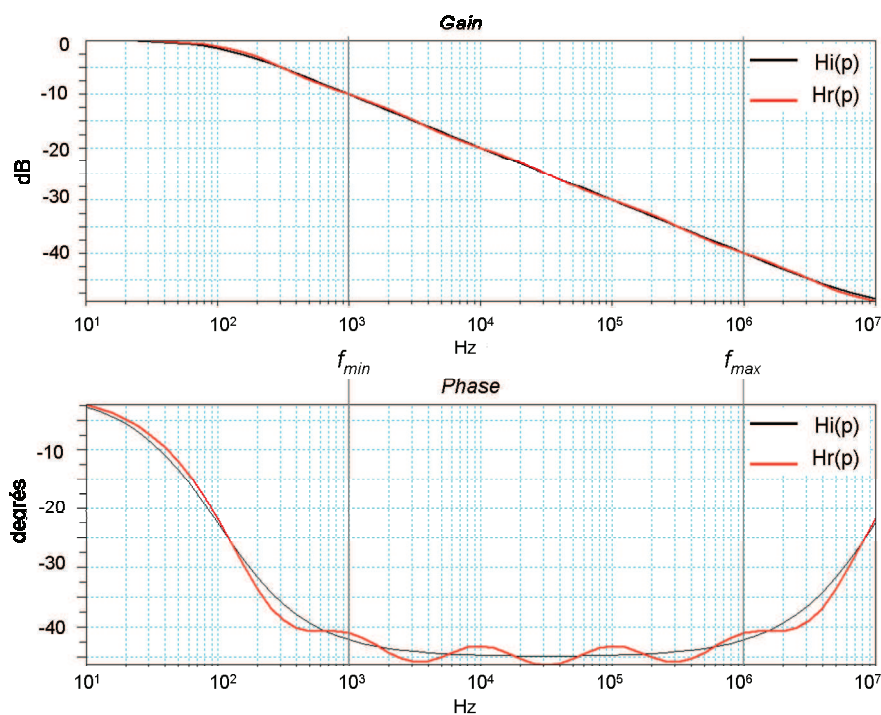
$$\frac{\omega'_{j+1}}{\omega'_j} = \frac{\omega_{j+1}}{\omega_j} = \gamma^2 \quad (3-5)$$

$$\omega'_1 = \sqrt{\gamma} \omega_H \quad (3-6)$$

On a, de manière symétrique,  $\omega_N = \sqrt{\gamma} \omega_L$  mais cette dernière relation est superflue pour le calcul de tous les  $\omega_j$  et  $\omega'_j$ .

Dans l'exemple ci-dessous, on applique cette méthode de synthèse pour les paramètres d'entrée suivants :

$$f_{min} = 1 \text{ kHz} \quad f_{max} = 1 \text{ MHz} \quad H_0 = 1 \quad N = 5$$



**Figure 6 : Diagramme de Bode du filtre synthétisé pour les paramètres**

$$f_{min} = 1 \text{ kHz}, f_{max} = 1 \text{ MHz}, H_0 = 1 \text{ et } N = 5.$$

On observe bien la présence d'ondulations et une pente moyenne de -10 dB/décade pour le gain, dans l'intervalle  $[f_{min}; f_{max}]$ . L'erreur relative  $\frac{|H_r - H_i|}{H_i}$  n'excède pas 2,5% pour le gain et 3,5% pour la phase. Ces erreurs maximales passent respectivement à 0,65% et 1,1% pour  $N = 7$ . Cela n'a pas été utilisé ici mais un article intéressant traite de la possibilité d'effectuer la synthèse avec une précision voulue [GUG01].

Maintenant, appliquons ce filtre à un bruit blanc de densité spectrale  $S_0$ , afin de générer le bruit en  $1/f$  représenté fig.5. En sortie du filtre on a :

$$S(f) = S_0 |H(p)|^2 \approx S_0 H_0^2 \left| \frac{\left(1 + \frac{p}{\omega_H}\right)}{\left(1 + \frac{p}{\omega_L}\right)} \right|^2$$

ce qui, dans l'intervalle  $[f_{min}; f_{max}]$  se simplifie en :

$$S(f) \approx S_0 H_0^2 \frac{1}{\left|\frac{p}{\omega_L}\right|^2} \approx S_0 H_0^2 \left(\frac{f_{min}}{10f}\right)$$

En prenant systématiquement  $H_0 = 1$  et sachant que l'on veut  $S(f_{cl}) = S_{cl}$ , on trouve :

$$S_{cl} = S_0 \left(\frac{f_{min}}{10f_{cl}}\right) \quad \text{soit} \quad S_{cl \text{ dB}} = S_0 \text{ dB} + 10 \log\left(\frac{f_{min}}{10f_{cl}}\right) \quad \text{et donc :}$$

$$S_0 \text{ dB / Hz} = S_{cl \text{ dB / Hz}} + 10 \log\left(\frac{10f_{cl}}{f_{min}}\right) \quad (7)$$

Le calcul de  $S_0$  permet donc d'ajuster le niveau de bruit en sortie du filtre, conformément aux spécifications.

Les équations numérotées de (1) à (7), ainsi que le choix du nombre N en fonction de la précision souhaitée, suffisent à synthétiser complètement le bruit en  $1/f$  décrit fig.5.

### 2.3. Implémentation du filtre et performances

Les développements précédents, en particulier la synthèse du filtre, ont été faits en temps continu alors que notre domaine d'application est celui de systèmes à temps discret. Les signaux manipulés sont échantillonnés à la fréquence  $f_e$ , fréquence qui cadence par ailleurs la source de bruit blanc gaussien se trouvant au cœur de la méthode. L'implémentation du filtre de fonction de transfert  $H_r(p)$  se fera donc après numérisation et obtention d'une équation de récurrence qui sera appliquée en sortie de la source de bruit blanc.

Avant cela, on a fait le choix de décomposer  $H_r(p)$  en éléments simples, de façon à transformer le produit de N fonctions de transfert du 1<sup>er</sup> ordre en une somme. En appliquant les formules mathématiques relatives à cette décomposition, on a :

$$H_r(p) = H_0 \frac{\prod_{j=1}^N \left(1 + \frac{p}{\omega'_j}\right)}{\prod_{j=1}^N \left(1 + \frac{p}{\omega_j}\right)} = c_0 + \sum_{i=1}^N \frac{c_i}{\omega_j + p}$$

$$\text{avec } c_0 = H_0 \frac{\prod_{j=1}^N \omega_j}{\prod_{j=1}^N \omega'_j} \quad \text{et} \quad c_i = c_0 \frac{\prod_{j=1}^N (\omega'_j - \omega_i)}{\prod_{j=1}^N (\omega_j - \omega_i)} .$$

L'intérêt de cette décomposition est qu'elle permet d'implémenter une équation de récurrence beaucoup moins complexe.

L'implémentation du filtre se fera en effet numériquement car nous sommes dans un contexte de circuits à temps discret, de fréquence d'échantillonnage  $f_e$ . La numérisation du filtre est faite ensuite au moyen de la transformée de Tustin :

$$p = 2f_e \frac{(1 - Z^{-1})}{(1 + Z^{-1})}$$

qui aboutit à une fonction de transfert en  $Z$  :

$$H_r(Z) = c_0 + \sum_{i=1}^N \frac{a_i + a'_i Z^{-1}}{b_i + b'_i Z^{-1}} \quad \text{avec} \quad \begin{cases} a_i = a'_i = c_i / f_e \\ b_i = 2 + \omega_i / f_e \\ b'_i = b_i - 4 \end{cases}$$

Finalement, la relation de récurrence du filtre s'écrit :

$$s_n = c_0 e_n + \sum_{i=1}^N f_{i,n} \quad \text{où} \quad f_{i,n} = \frac{1}{b_i} [a_i e_n + a'_i e_{n-1} - b'_i f_{i,n-1}] \quad (8)$$

Les différents coefficients  $c_0$ ,  $a_i$ ,  $a'_i$ ,  $b_i$  et  $b'_i$  étant fixés, le calcul d'un échantillon  $s_n$  requiert  $N+2$  additions et 5 multiplications, ainsi que la mémorisation de l'état précédent pour 2 variables,  $e_n$  et  $f_{i,n}$ .

En comparaison, la forme précédente de  $H_r(p)$ , avant la décomposition en éléments simples, aurait nécessité une mémorisation des échantillons antérieurs jusqu'à l'ordre  $N$  et un nombre plus important d'opérations.

## 2.4. Généralisation

Imaginons une densité spectrale de bruit telle que celle représenté fig.7. Les fréquences de coupures  $f_{c1}$ ,  $f_{c2}$  et  $f_{c3}$  séparent les différentes régions du spectre ; en ces points,  $S_{c1}$ ,  $S_{c2}$  et

$S_{c3}$  sont respectivement les niveaux de bruit de la représentation asymptotique. Ce spectre présente 4 régions distinctes et sa modélisation fera donc appel à 4 générateurs aléatoires gaussiens, indépendants entre eux, dont les densités spectrales sont notées  $S_0$ ,  $S_{0_1}$ ,  $S_{0_2}$  et  $S_{0_3}$ .<sup>7</sup>

Les cas du bruit blanc et du bruit en  $1/f$  ont déjà été exposés plus haut.

Lorsqu'il s'agit de générer un bruit en  $1/f^2$  dans un intervalle donné  $[f_{c3}, f_{c2}]$ , on aura recours à un bruit blanc gaussien de densité spectrale  $S_{0_2}$ , appliqué à un intégrateur ordre 1, donc en en  $1/p$ . La transformée de Tustin sera également utilisée pour numériser ce filtre :

$$s_n = s_{n-1} + \frac{1}{2f_e}(e_n + e_{n-1})$$

Le bruit blanc  $S_{0_2}$  servira à ajuster le niveau de bruit final pour que  $S(f) = S_{0_2} |1/p|^2$  passe par le point  $(f_{c2}, S_{c2})$ . Ce qui donne :

$$S_{0_2 \text{ dB/Hz}} = S_{c2 \text{ dB/Hz}} + 20 \log(2\pi f_{c2}) \quad (9)$$

Pour le bruit en  $1/f^3$ , sur l'intervalle  $[f_{\min}, f_{c3}]$ , on fera se succéder un intégrateur d'ordre  $1/2$  et un intégrateur d'ordre 1, avec :

$$S_{0_3 \text{ dB/Hz}} = S_{c3 \text{ dB/Hz}} + 10 \log\left(\frac{10f_{c3}}{f_{\min}}\right) + 20 \log(2\pi f_{c3}) \quad (10)$$

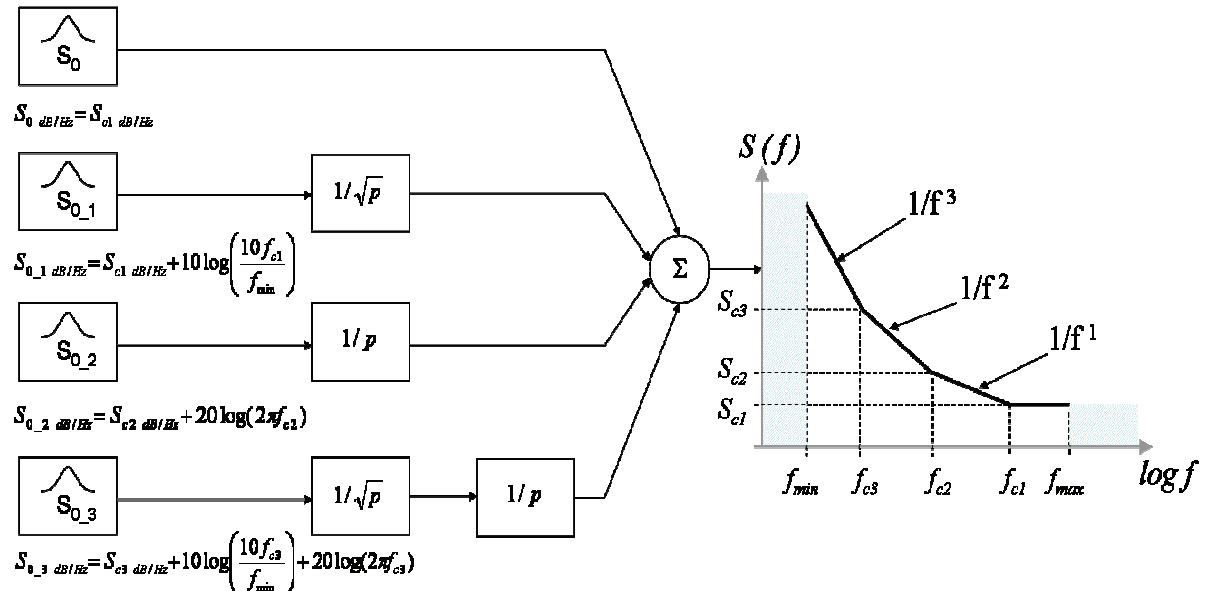


Figure 7 : Synoptique de la génération d'un bruit complexe

La fig.7 résume le principe général ; les blocs élémentaires utiles à la méthode sont donc des générateurs aléatoires gaussiens, des intégrateurs d'ordre 1 et d'ordre  $1/2$ . Les données  $f_{\min}$ ,

<sup>7</sup> C'est l'initialisation de ces générateurs aléatoires qui garantira leur indépendance.

$f_{max}$  et  $f_e$  étant constantes, on réutilisera toujours le même intégrateur d'ordre  $\frac{1}{2}$ , quelles que soient ses occurrences dans le synoptique. Il en va de même pour l'intégrateur d'ordre 1. Selon ce principe, seuls les niveaux des générateurs aléatoires gaussiens permettent d'ajuster le gabarit spectral voulu.

Ainsi le principe de construction modulaire, avec modules réutilisables, est encore respecté dans cette application.

### **3. Applications et résultats**

Le précédent paragraphe a exposé comment on pouvait produire une séquence aléatoire de nombres réels, de période d'échantillonnage fixée et de densité spectrale de puissance donnée. Selon les applications, nous allons voir que cette séquence aléatoire peut prendre une dimension physique adaptée au problème : tension, phase, temps, etc.

#### **3.1. Bruits de tension élémentaires**

Les modèles exposés et caractérisés ci-après posent les bases de l'implémentation de la méthode. La séquence aléatoire est appliquée à *signal* (au sens VHDL, donc de nature événementielle), à valeurs réelles. Ces modèles délivrent donc ce qu'on peut considérer comme des bruits de tension.

##### **3.1.1 Bruit blanc**

En matière de générateur pseudo-aléatoire, le langage VHDL propose la fonction la fonction UNIFORM qui fournit des nombres aléatoires avec une répartition uniforme entre 0 et 1. Une technique classique existe pour transformer un signal aléatoire uniforme en un signal gaussien, il s'agit de la transformée de Box-Muller. Le principe utilise deux processus aléatoires  $u_1$  et  $u_2$ , de répartition uniforme sur  $[0,1[$ , et construit le processus gaussien  $g$  de moyenne  $m$  et d'écart-type  $\sigma$  de la façon suivante :

$$g = m + \sigma \sqrt{-2 \ln(u_1)} \sin(2\pi u_2)$$

On choisit de générer un bruit blanc avec les paramètres d'entrée suivants :

$$S_0 = -50 \text{ dB/Hz}$$

$$f_e = 20 \text{ MHz}$$

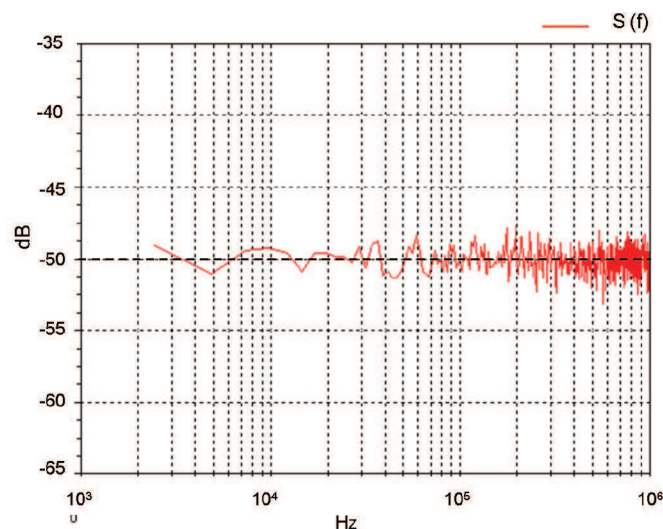
$$N_e = 131072, \text{ nombre total d'échantillons analysés.}$$

Le listing list.1 donne un extrait du modèle VHDL de cette source, avec commentaires.

<pre> ENTITY bruit_blan IS   GENERIC (     Fsmpl : REAL := REAL(2e+007) ;     S0_dB_Hz : REAL := REAL(-50) );   PORT (SIGNAL Nout : OUT REAL := 0.0); END ENTITY bruit_blan;  ARCHITECTURE behavioral OF bruit_blan IS   CONSTANT sigma : REAL := Sqrt(Fsmpl*10.0**(S0_dB_Hz/10.0)); BEGIN   PROCESS     VARIABLE unif_1, unif_2 : REAL ;     VARIABLE seed_1 : INTEGER := 31 ;     VARIABLE seed_2 : INTEGER := 986 ;     VARIABLE seed_3 : INTEGER := 31006 ;     VARIABLE seed_4 : INTEGER := 74090 ;      BEGIN      UNIFORM(seed_1,seed_2,unif_1);     UNIFORM(seed_3,seed_4,unif_2);      Nout &lt;= sigma*Sqrt(-2.0*LOG(unif_1))*SIN(MATH_2_PI*unif_2);      WAIT FOR (1.0/Fsmpl);      END PROCESS ; END ARCHITECTURE behavioral; </pre>	<p>Passage des paramètres caractéristiques.</p> <p>Conversion d'unité.</p> <p>Nombres entiers utilisés pour initialiser les 2 générateurs pseudo-aléatoires.</p> <p>Appel aux 2 générateurs pseudo-aléatoires.</p> <p>Application de Box-Muller.</p> <p>Attente d'une période.</p> <p>Retour au début du processus.</p>
--	---

**List.1 : Modèle VHDL d'une source de bruit blanc (extrait)**

Une simulation est effectuée sur  $N_e$  échantillons et la sortie de la source est récupérée sous le logiciel libre de calcul numérique SCILAB [SCI] pour analyse spectrale. On estime la densité spectrale de puissance par un algorithme de périodogramme moyenné sur 16 sous-fenêtres de  $M = 8192$  échantillons chacune, entrelacées à 50%, avec fenêtre rectangulaire (fig.8).



**Figure 8 : Densité spectrale de puissance de la source de bruit blanc**

### 3.1.2 Bruit en $1/f$

Afin de vérifier la procédure de synthèse d'un bruit en  $1/f$  décrite au paragraphe 2.2., on choisit de générer un bruit avec les paramètres d'entrée suivants :

$$f_{cl} = 100 \text{ kHz}, S_{fcl} = -50 \text{ dB/Hz}$$

$$f_e = 20 \text{ MHz}, f_{min} = 1 \text{ kHz}, f_{max} = 1 \text{ MHz}, N = 7$$

$$N_e = 131072.$$

Le listing list.2 donne un extrait du modèle VHDL de cette source, avec commentaires.

Pour vérifier le bon comportement de cette source de bruit, un périodogramme est effectué, toujours dans les mêmes conditions. Le résultat (fig.9) valide la méthode.

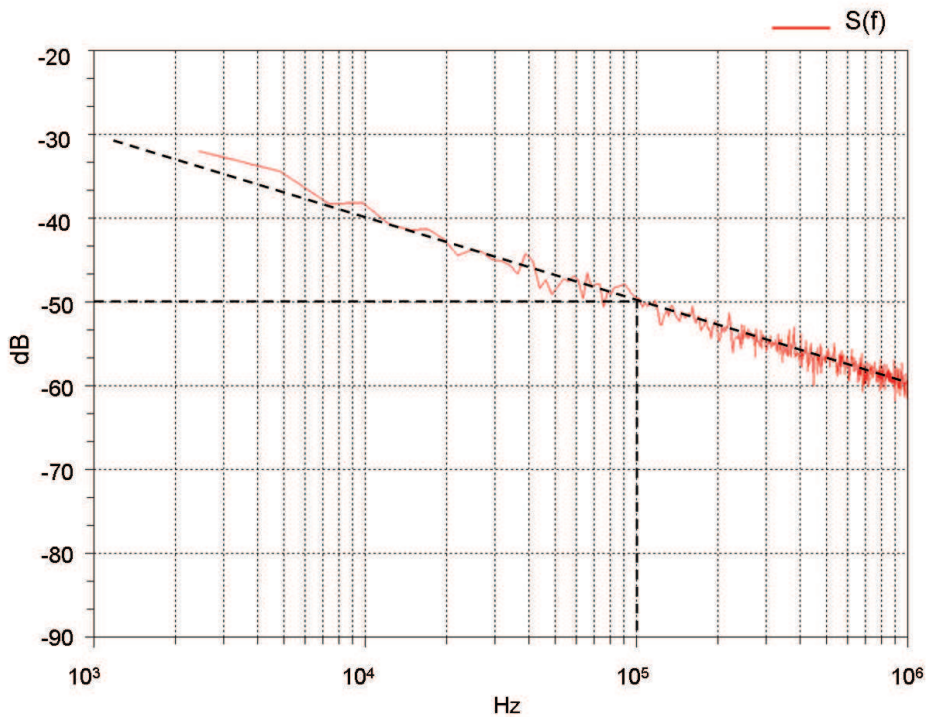


Figure 9 : Densité spectrale de puissance de la source de bruit en  $1/f$

<pre> ----- ENTITY bruit_un_sur_f IS     GENERIC (         Fsmpl : REAL := REAL(2e+007) ;         Fc1 : REAL := REAL(1e+005) ;         Sc1_dB_Hz : REAL := REAL(-50));     PORT( SIGNAL Nout: OUT REAL := 0.0); END ENTITY bruit_un_sur_f; -----  ARCHITECTURE behavioral OF bruit_un_sur_f IS  -- modele genere pour Fmin = 1e+003 et Fmax = 1e+006 avec N = 7 -- -- reprendre la procedure si l'un de ces parametres change --  CONSTANT S0_dB_Hz : REAL := Sc1_dB_Hz + 10*log10(Fc1/100.0); CONSTANT sigma : REAL := SQRT(Fsmpl*10.0**(S0_dB_Hz/10.0));  BEGIN  PROCESS VARIABLE unif_1, unif_2 : REAL ; VARIABLE seed_1 : INTEGER := 47 ; VARIABLE seed_2 : INTEGER := 1480 ; VARIABLE seed_3 : INTEGER := 46509 ; VARIABLE seed_4 : INTEGER := 61136 ;  VARIABLE en : REAL := 0.0; VARIABLE en_1 : REAL := 0.0; VARIABLE fn : REAL_VECTOR(1 TO 7) := (0.0,0.0,0.0,0.0,0.0,0.0,0.0); VARIABLE fn_1 : REAL_VECTOR(1 TO 7) := (0.0,0.0,0.0,0.0,0.0,0.0,0.0);  BEGIN  -- Génération du bruit blanc -- UNIFORM(seed_1,seed_2,unif_1); UNIFORM(seed_3,seed_4,unif_2); en := sigma*SQRT(-2.0*LOG(unif_1))*SIN(MATH_2_PI*unif_2);  -- Filtrage du bruit blanc -- en_1 := en; fn_1(7) := fn(7); fn_1(6) := fn(6); fn_1(5) := fn(5); fn_1(4) := fn(4); fn_1(3) := fn(3); fn_1(2) := fn(2); fn_1(1) := fn(1);  fn(1) := (0.34305) * ((0.00241028) * en + (0.00241028) * en_1 -(-1.08497) * fn_1(1)); fn(2) := (0.459418) * ((0.0012012) * en + (0.0012012) * en_1 -(-1.82334) * fn_1(2)); fn(3) := (0.491616) * ((0.000539446) * en + (0.000539446) * en_1 -(-1.96589) * fn_1(3)); fn(4) := (0.498359) * ((0.000238418) * en + (0.000238418) * en_1 -(-1.99341) * fn_1(4)); fn(5) := (0.499682) * ((0.00010582) * en + (0.00010582) * en_1 -(-1.99873) * fn_1(5)); fn(6) := (0.499939) * ((4.89208e-005) * en + (4.89208e-005) * en_1 -(-1.99975) * fn_1(6)); fn(7) := (0.499988) * ((3.09417e-005) * en + (3.09417e-005) * en_1 -(-1.99995) * fn_1(7));  Nout &lt;= 0.00316228 * en + fn(1) + fn(2) + fn(3) + fn(4) + fn(5) + fn(6) + fn(7);  WAIT FOR (1.0/Fsmpl);  END PROCESS ;  END ARCHITECTURE behavioral; </pre>	<p>Passage des paramètres caractéristiques.</p> <p>Calibration de la source de bruit blanc en entrée du filtre.</p> <p>Nombres entiers utilisés pour initialiser les 2 générateurs pseudo-aléatoires.</p> <p>Initialisation des variables de l'algorithme de récurrence du filtre.</p> <p>Appel aux 2 générateurs pseudo-aléatoires. Application de Box-Muller.</p> <p>Mémorisation des échantillons de l'état précédent.</p> <p>Calcul des termes servant dans l'équation de récurrence.</p> <p>Implémentation de l'équation de récurrence définie en (3-8)</p> <p>Attente d'une période.</p> <p>Retour au début du processus.</p>
--	---

**List.2 : Modèle VHDL d'une source de bruit en 1/f (extrait)**



### 3.1.3 Bruit combiné en 1/f et blanc

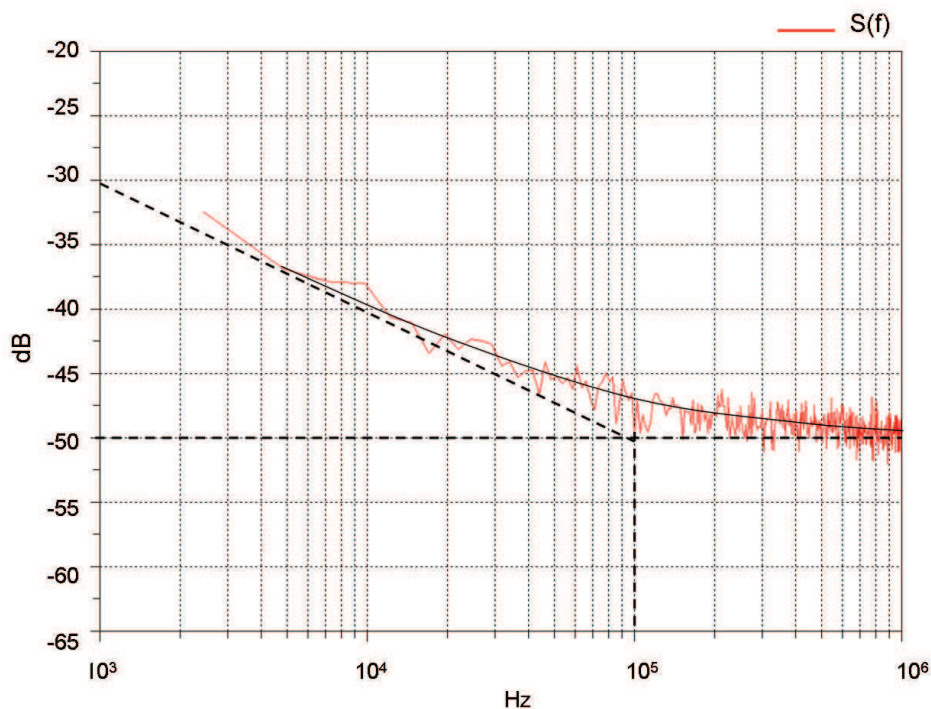
On souhaite vérifier ici le principe de superposition d'une source de bruit blanc et d'une source de bruit en  $1/f$ , indépendantes, pour aboutir à un bruit combiné présentant 2 régions de ce type. Les paramètres d'entrée du bruit à générer sont :

$$f_{cl} = 100 \text{ kHz}, S_{f_{cl}} = -50 \text{ dB/Hz}$$

$$f_e = 20 \text{ MHz}, f_{min} = 1 \text{ kHz}, f_{max} = 1 \text{ MHz}, N = 7$$

$$N_e = 131072.$$

En ce qui concerne le modèle VHDL, il fait simplement appel aux deux blocs décrits juste avant et dûment paramétrés (selon les formules données fig.7), dont on somme les sorties. Le résultat du périodogramme est montré fig.10, il confirme l'ensemble de la méthode. On trouve comme convenu un écart d'environ 3 dB par rapport au spectre asymptotique, à la fréquence  $f_{cl}$ .



**Figure 10 : Densité spectrale de puissance de la source de bruit combiné blanc +  $1/f$**

### 3.2. Bruit des oscillateurs

Les oscillateurs sont des circuits clés dans les systèmes de radiocommunication car ils assurent la synchronisation des différentes fonctions ou servent de référence de fréquence. L'impact du bruit électronique sur ces circuits se traduit en une variation aléatoire de l'amplitude et de la phase de l'oscillation. Cette dernière est connue sous le nom de bruit de

phase (caractéristique fréquentielle) ou bien encore de jitter (caractéristique temporelle), qui sont deux métriques du même phénomène. Les deux exemples choisis dans ce paragraphe permettent de couvrir les deux visions.

### 3.2.1 Modèle d'une sinusoïde bruitée

Dans un oscillateur idéal, la fréquence d'oscillation  $f_0$ , ainsi que l'amplitude  $A$  sont fixes et la tension de sortie peut s'écrire :

$$v_0(t) = A \cos(\varphi(t)) = A \cos(2\pi f_0 t + \varphi_0)$$

où  $\varphi(t)$  représente la phase instantanée,  $\varphi_0$  la phase à l'origine, fixe et souvent choisie nulle (c'est ce qu'on considère par la suite).

En réalité, amplitude et phase sont tous deux sujets à des variations aléatoires, ce qui peut être représenté par une déviation de tension  $da(t)$  et une déviation de phase  $d\varphi(t)$ .

$$v(t) = A(1 + da(t)) \cos(2\pi f_0 t + d\varphi(t))$$

D'un point de vue pratique, on n'a pas directement accès (mesure à l'analyseur de spectre) aux densités spectrales de puissances  $S_{d\varphi}(f)$  ou  $S_{da}(f)$  mais plutôt à  $S_v(f)$ , densité spectrale du bruit de tension en sortie de l'oscillateur. Les deux contributions, bruit d'amplitude et bruit de phase, sont alors mélangées et il n'est pas aisé de les séparer. Le bruit de phase est le paramètre le plus critique car la nature non-linéaire d'un oscillateur tend à corriger en permanence l'erreur d'amplitude [KUN06], alors que la phase peut dériver de manière significative et entraîner des phénomènes de désynchronisation. Nous nous intéressons dans les exemples choisis au bruit de phase uniquement.

D'un point de vue théorique, la densité spectrale de puissance  $S_{d\varphi}(f)$  présente en général deux régions caractéristiques autour de la fréquence fondamentale  $f_0$  : une région en  $1/f^3$  et une région en  $1/f^2$ , issue de l'intégration des bruits de tension élémentaires en  $1/f$  et blanc. Dans le cas où  $d\varphi(t)$  reste petit, on peut écrire :

$$v(t) \approx A \cos(2\pi f_0 t) - A \sin(2\pi f_0 t) (d\varphi(t))$$

En termes de densité spectrale, cela donne :

$$S_v(f) = \left[ \frac{A^2}{4} \delta(f - f_0) + \frac{A^2}{4} \delta(f + f_0) \right] + \left[ \frac{A^2}{4} S_{d\varphi}(f - f_0) + \frac{A^2}{4} S_{d\varphi}(f + f_0) \right]$$

En pratique, on observe seulement la partie du spectre pour  $f > 0$ , appelé spectre unilatéral:

$$S_v(f)_{f>0} = \left[ \frac{A^2}{4} \delta(f + f_0) \right] + \left[ \frac{A^2}{4} S_{d\varphi}(f + f_0) \right]$$

Le 1<sup>er</sup> terme correspond à la puissance de la porteuse idéale, le 2<sup>ème</sup> terme correspond à l'influence du bruit de phase. C'est ce dernier terme qui nous intéresse. En opérant un changement d'origine autour de la porteuse, soit  $f = f_m + f_0$ , et en se plaçant à  $f_m \neq 0$ , on a :

$$S_v(f_m + f_0)_{f>0} = \frac{A^2}{4} S_{d\phi}(f_m)_{f>0}$$

Le terme  $A^2/4$  correspond à la puissance de l'oscillation non bruitée. On a alors accès à  $S_{d\phi}(f)_{f>0}$  (grandeur souvent notée  $\mathcal{L}(f_m)$ ) en normalisant  $S_v(f)$  par rapport à la puissance de la porteuse,  $P_0$  :

$$S_{d\phi}(f_m) = \frac{S_v(f_m + f_0)}{P_0}$$

L'unité utilisée étant alors le dBc/Hz.  $P_0$  prendra la valeur  $A^2/4$  s'il s'agit d'un spectre unilatéral « tronqué » (on supprime simplement le spectre pour les fréquences négatives) ; il prendra la valeur  $A^2/2$  s'il s'agit d'un spectre unilatéral « replié » (on replie le spectre des fréquences négatives sur le spectre des fréquences positives, cela permet de conserver la puissance totale).

En résumé, le modèle implémente l'équation :

$$v(t) = A \cos(2\pi f_0 t + d\phi(t))$$

L'excès de phase  $d\phi(t)$  étant généré selon le principe résumé fig.11.

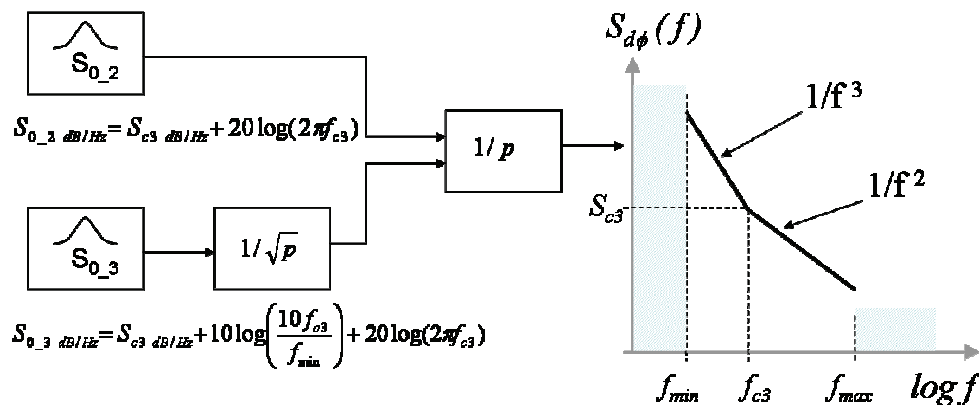


Figure 11 : Schéma fonctionnel de la génération du bruit d'excès de phase  $S_{d\phi}(f)$

Le modèle développé ici produit une tension prenant des valeurs réelles « pseudo-continues », en effet l'excès de phase est issu d'une séquence aléatoire échantillonnée, à la fréquence fixe  $f_e$ . Une simulation a été menée avec les paramètres suivants :

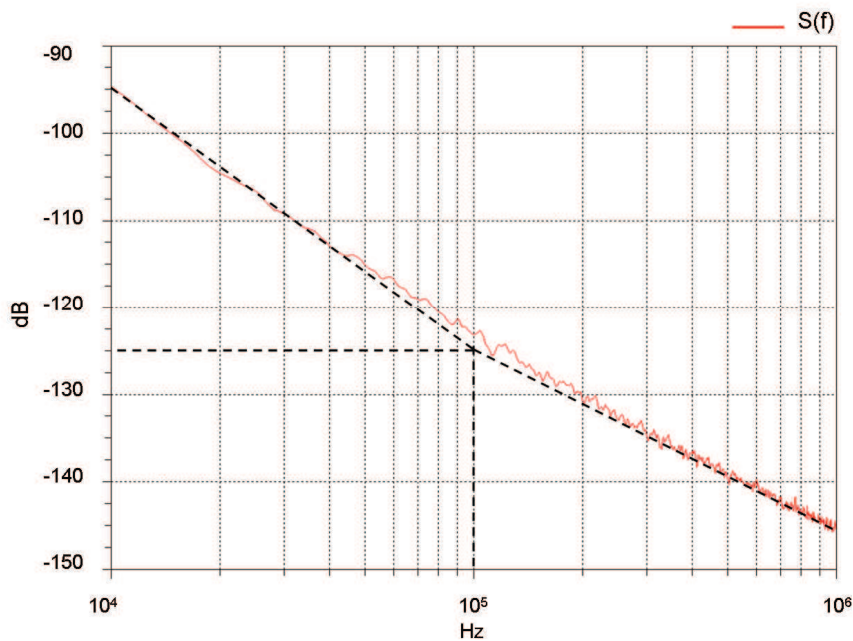
$$f_0 = 200 \text{ kHz}, A = 14,14 \text{ V } (A^2/2 = 100), \phi_0 = 0$$

$$f_e = 2 \text{ MHz}, f_{c3} = 100 \text{ kHz} \text{ et } S_{c3} = -125 \text{ dBc/Hz}$$

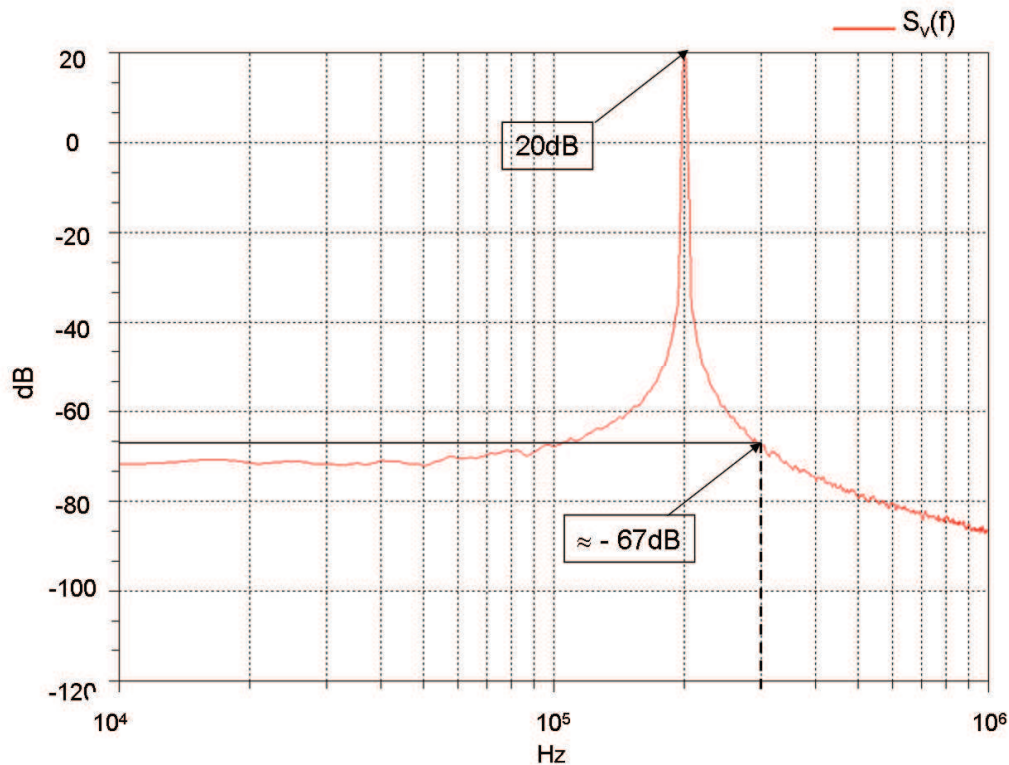
Le résultat de la simulation a été analysé, sur un nombre total de  $N_e = 1048576$  échantillons, au moyen d'un périodogramme moyenné, sur 32 sous-fenêtres de  $M = 32768$  échantillons chacune, entrelacées à 50%, avec fenêtre de Blackman7.

La fig.12 représente le spectre de l'excès de phase généré, il est estimé à l'aide d'un périodogramme directement corrigé par le facteur de bande équivalente de bruit, comme c'est le cas pour tous les exemples précédents.

La fig.13 représente le spectre de tension en sortie de l'oscillateur, il est estimé à l'aide d'un périodogramme qui n'est pas corrigé par le facteur de bande équivalente de bruit afin de ne pas perturber l'amplitude de la raie fondamentale. La bande équivalente de bruit de l'algorithme de FFT est de 6425.6 Hz.



**Figure 12 : Spectre du bruit d'excès de phase  $d\varphi(t)$**



**Figure 13 : Spectre en puissance de la tension de sortie  $v(t)$**

On observe bien un pic de 20 dB à 200 kHz correspondant à la porteuse. Notons que l'échelle des fréquences est logarithmique, ce qui ne permet pas d'observer la symétrie du spectre autour de  $f_0$ . A une fréquence d'offset  $f_m = 100$  kHz de la porteuse on trouve une puissance de -67 dB, qui doit être corrigée par la bande équivalente de bruit, soit  $S_v(f_m = 100 \text{ kHz}) \approx -67 - 10\log(6425.6) = -67 - 38 = -105$  dB. Lorsque l'on rapporte cette valeur à la puissance de la porteuse, on trouve bien  $S_{d\phi}(f_m = 100 \text{ kHz}) \approx -125$  dB.

### 3.2.2 Modèle d'une horloge avec jitter

Dans le cas d'une horloge idéale, les instants de transition  $t_n$  sont équidistants d'un intervalle correspondant à la période idéale  $T$ . Dans le cas d'une horloge avec jitter ces instants subissent une déviation  $\Delta t_n$  :

$$t_n = nT + \Delta t_n$$

La séquence des  $\Delta t_n$  représente ce qu'on appelle le jitter absolu [LEE02]. Si on le mesure plutôt en radian :

$$\Delta \phi_n = \frac{2\pi}{T} \Delta t_n \quad (11)$$

on a alors le jitter de phase absolu.

Ce jitter de phase absolu est une version discrète de l'excès de phase  $d\phi(t)$  du cas sinusoïdal précédent et leurs densités spectrales de puissance sont identiques [LEE02], [KUN06]. C'est la donnée de  $S_{d\phi}(f)$  qui est utilisée pour caractériser le modèle ici présenté, même si d'autres métriques permettant de caractériser le jitter existent [LEE02] [FAK02]. On peut aussi mesurer la période sur un cycle et étudier la déviation de cette période par rapport au cas idéal :

$$T_n = t_n - t_{n-1} = nT + \Delta t_n - (n-1)T - \Delta t_{n-1} = T + \Delta t_n - \Delta t_{n-1}$$

$$\Delta T_n = T_n - T = \Delta t_n - \Delta t_{n-1} \quad (12)$$

C'est cette dernière équation qui est utilisée dans le modèle, car il est basé sur l'utilisation de l'instruction WAIT qui attend une durée égale à une demi-période pour faire basculer la sortie de l'oscillateur. Les paramètres du modèle sont donc extraits de la donnée de la densité spectrale de bruit de phase  $S_{d\phi}(f)$  mesuré, caractérisant la séquence des excès de phase  $\Delta\phi_n$ . L'écart de période est alors calculé en combinant les équations (11) et (12). La séquence des  $\Delta\phi_n$  est alors générée selon le principe exposé au paragraphe 2. et appliqué au modèle d'horloge dont le code est donné list.3.

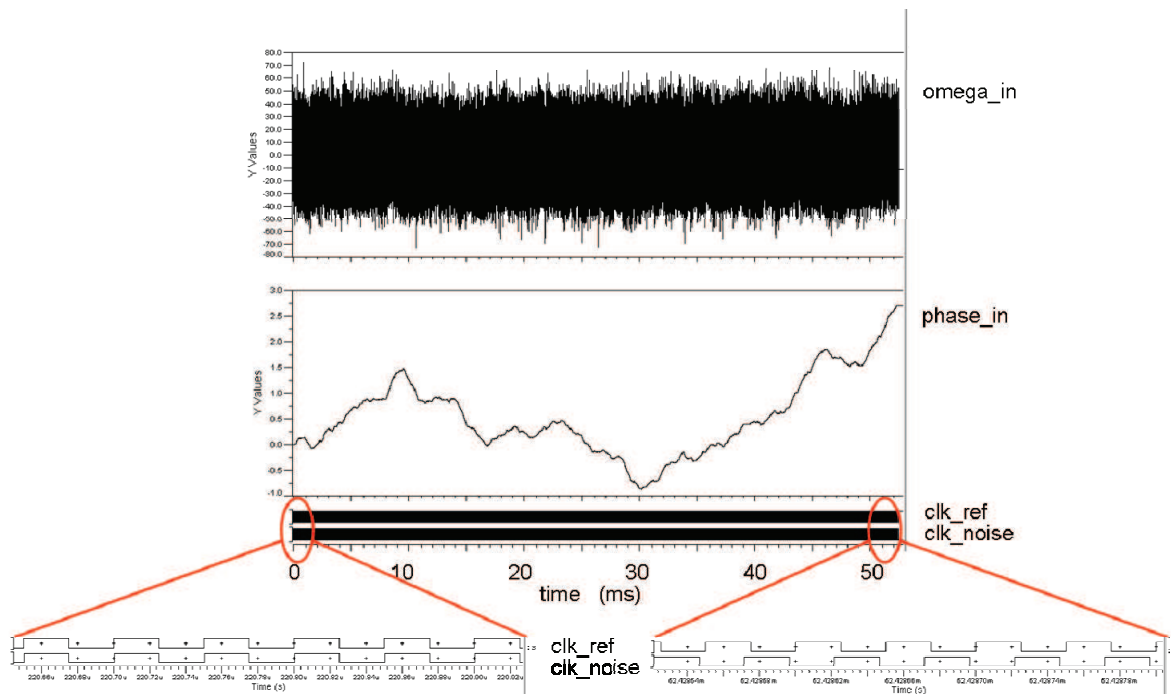
Dans l'exemple proposé, le modèle reçoit en entrée le signal `phase_in` issu d'un bloc qui génère une séquence aléatoire dont le spectre est en  $1/f^3$  puis  $1/f^2$ . On utilise dans cet exemple le même gabarit de bruit de phase que dans l'exemple sinusoïdal précédent et le principe de génération du signal `phase_in` est donc représenté fig.12.

Une simulation a été menée sur un nombre total de  $N_e = 1048576$  échantillons, la fréquence idéale de l'horloge est de 20 MHz, tout comme la fréquence de génération du jitter.

La fig.14 donne le résultat de cette simulation. Le signal de sortie de l'horloge avec jitter est `clk_noise`, on le compare à un signal d'horloge idéal appelé `clk`. En fait de ce graphe sont également représentés le signal déviation de phase `phase_in` (qui est produit à une cadence de 20 MHz, selon la fig.12) et sa dérivée `omega_in` (il s'agit de l'entrée de l'intégrateur  $1/p$  de la fig.12). On constate bien l'effet cumulatif du jitter en comparant avec le signal de l'horloge idéale : son effet est peu visible en début de simulation, il devient important au voisinage de 50 ms, ce qui représente plus d'un million de cycles. C'est un effet typique du bruit en  $1/f^n$ .

<pre> ----- ENTITY digital_random_pulse IS GENERIC (      frequency : REAL := 20.0e+6 ); PORT ( phase_in : IN REAL ;       Npulse: INOUT BIT := '0'); END ENTITY digital_random_pulse; -----  ARCHITECTURE behaviour OF digital_random_pulse IS BEGIN Output_generation : PROCESS   VARIABLE Phi, Phi_1 : REAL := 0.0;   VARIABLE waiting_time : REAL := 0.0; BEGIN   Phi_1 := Phi ;   Phi := phase_in ;    ASSERT ((1.0+(Phi-Phi_1)/MATH_2_PI)/2.0/frequency &gt; 0.0)   REPORT "NEGATIV DEMI-PERIOD" SEVERITY warning;    IF (1.0+(Phi-Phi_1)/MATH_2_PI)/2.0/frequency &gt; 0.0 THEN     waiting_time := (1.0+(Phi-Phi_1)/MATH_2_PI)/2.0/frequency;   END IF;    Npulse &lt;= '1'; WAIT FOR waiting_time;   Npulse &lt;= '0'; WAIT FOR waiting_time; END PROCESS Output_generation; END ARCHITECTURE behaviour; </pre>	<p>Initialisation des variables utilisée dans le process.</p> <p>Mémorisation de l'écart de phase précédent.</p> <p>Vérification de la positivité de la demi-période</p> <p>Calcul de la demi-période</p> <p>Basculements de la sortie sur un cycle</p>
--	---

**List.3 : Modèle VHDL d'une horloge avec jitter (extrait)**



**Figure 14 : Résultats de la simulation de l'horloge avec jitter**



### 3.3. Modélisation du bruit dans une chaîne de réception

Revenons sur la chaîne de réception décrite fig.1. Les travaux menés pendant la thèse de G. Monnerie étaient destinés à estimer une de ses performances, à savoir le rapport signal à bruit SNR. Au moyen des modèles bruits d'origines variées décrits plus haut et ayant au préalable développé des modèles comportementaux en VHDL-AMS des différents blocs constituant cette chaîne, une simulation système a pu être entreprise.

La société EADS disposait d'une mesure du SNR, réalisée sur la puce, elle est représentée fig.15. Un signal sinusoïdal est appliqué à l'entrée de la puce (entrée du bloc CAG) et la sortie est mesurée après le modulateur  $\Sigma\Delta$ . La mesure du SNR est effectuée pour différentes valeurs de l'amplitude du signal d'entrée jusqu'à la pleine échelle du convertisseur. Sur la fig.15, on représente le SNR en fonction de la variable  $X = 10\log(P_{in}/P_{FS})$ ,  $P_{in}$  étant la puissance du signal d'entrée et  $P_{FS}$  celle du signal pleine échelle.

Il est facile d'évaluer théoriquement le bruit thermique des différents blocs et d'expliquer ainsi le plateau de 96,5 dB pour les faibles amplitudes du signal d'entrée. Par contre elle ne suffit pas à expliquer la dégradation du SNR pour les fortes amplitudes.

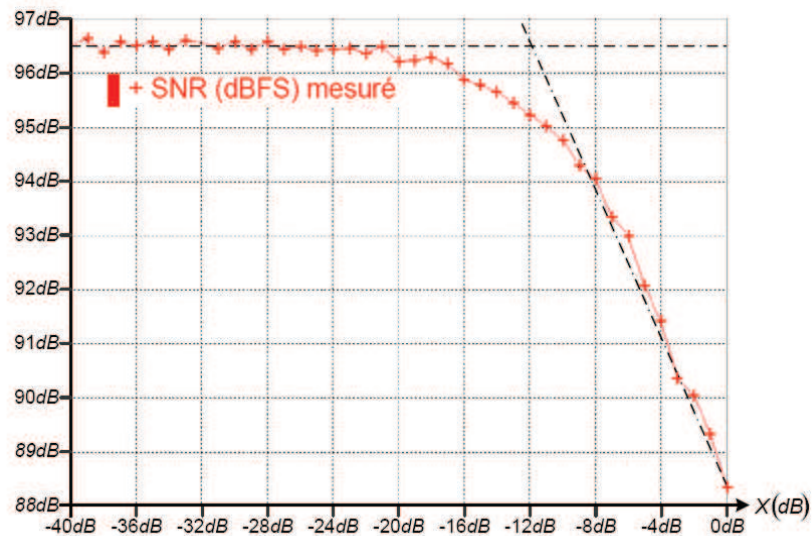
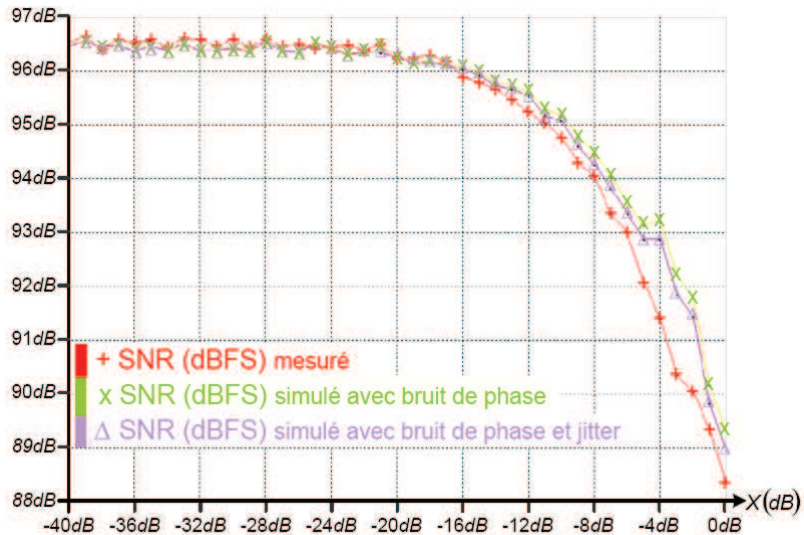


Figure 15 : SNR mesuré de la chaîne de réception

Trois autres contributions ont pu être introduites au moyen des modèles précédents. Le jitter interne du modulateur qui a été extrait d'un banc de mesure et d'un protocole développé à l'IMS, dans l'équipe de D. Dallet, le bruit de phase de la PLL et celui du générateur produisant le signal d'entrée. On peut alors estimer le SNR par simulation et le résultat se trouve fig.16.





**Figure 16 : Comparaison du SNR mesuré avec le SNR simulé**

Une conclusion remarquable de ces travaux indique que la contribution en bruit prépondérante est celle du bruit de phase du générateur d'entrée et ne remet donc pas en cause l'architecture de l'ASIC.

#### **4. Conclusion**

La question des méthodes de modélisation a été traitée dans le chapitre précédent, dans un objectif de généralisation et de formalisme. Le chapitre 2 s'est quant à lui concentré sur un point précis : la modélisation de sources de bruit paramétrables et adaptées à une simulation électrique temporelle. Ce besoin est très vite apparu dans le cadre de la conception et de la simulation des circuits et systèmes radio-fréquence. La thèse de G. Monnerie, en contrat CIFRE avec EADS-telecom, a permis de développer une méthode pour générer des sources de bruit génériques, paramétrables, permettant de couvrir les principaux gabarits de densité spectrale de puissance rencontrés dans le domaine RF.

Les travaux présentés dans ce chapitre ont fait l'objet de cinq communications [MLD03], [MLL04], [LMS04], [LML05], [LML06]. La majorité des développements se trouve dans la thèse de G. Monnerie [MON05] et le rapport de stage de L. Lewis [LEW04]. La thèse d'A. Fakhfakh [FAK02] avait posé les bases la modélisation des circuits RF et du bruit de phase des oscillateurs [FMD01], [FMB02].

Le stage de L. Lewis [LEW04] a permis de passer du formalisme à la génération automatique des modèles de bruit en VHDL. Un logiciel a ainsi été développé sous SCILAB, permettant à la fois la génération automatique des modèles après saisie des paramètres caractéristiques et l'analyse spectrale des signaux simulés sous ADVanceMS.

Cette méthode est aujourd'hui exploitée au laboratoire, dans l'équipe de D. Dallet par G. Eynard (contrat post-doc), pour simuler le bruit de phase dans les synthétiseurs de fréquence. Un article de conférence a été soumis [ELD10].



# CONCEPTION ANALOGIQUE

## A BASE D'IPS

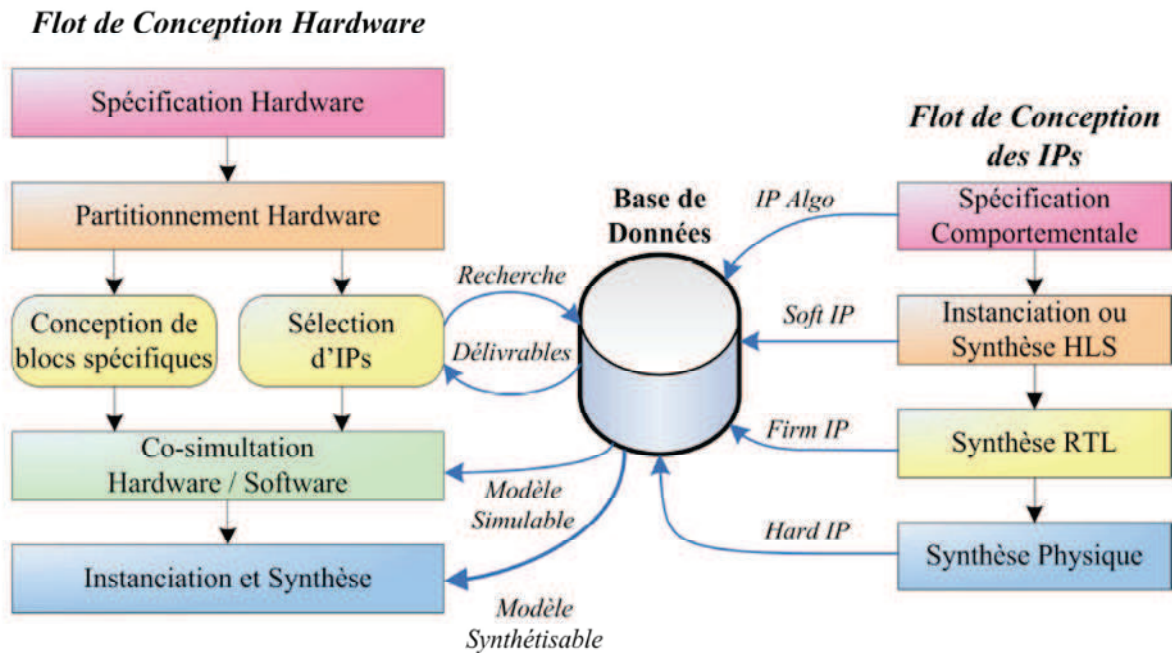
### 1. Introduction

Les précédents chapitres ont traité des méthodes et techniques de modélisation dans le cadre du flot de conception analogique et mixte ; des solutions ont été proposées pour assister le concepteur dans les phases Top-Down ou Bottom-Up de la conception. Au-delà de cet aspect modélisation, le présent chapitre pose la question du processus de conception dans son ensemble. Dans un premier temps, nous reviendrons sur les raisons qui font que, dans les systèmes sur puce analogiques et mixtes (SoC-AMS), les parties analogiques requièrent le plus gros effort de conception, tout en occupant une surface souvent bien inférieure à celle des parties numériques. Pour cela, nous commencerons par résumer les avancées historiques et points forts actuels en conception numérique : la notion d'*IP* y tient rôle majeur, en tant que composant *pré-conçu*. Ensuite, nous verrons quelles sont les pistes pour élargir cette notion au domaine analogique, pour finalement introduire les choix qui sont à la base de la thèse de T. Lévi [LEV07].

#### 1.1. Conception de systèmes sur puce numériques

Dans le domaine des circuits numériques, le flot de conception est aujourd'hui largement automatisé, notamment dans le processus de synthèse, partant de la description d'une architecture haut-niveau et aboutissant au schéma électrique au niveau portes logiques ainsi qu'au placement/routage. La première révolution dans ce domaine s'est faite avec l'effort de standardisation donnant les langages de description VHDL ou Verilog, qui sont au cœur de tous les outils CAO commerciaux. C'est aussi grâce à eux que des méthodologies de réutilisation ont pu être développées en conception numérique [SWM06], et la notion d'*Intellectual Property* (IP) est désormais incontournable sur le marché du semi-conducteur. Ces IPs représentent des blocs logiciels ou matériels préconçus, pouvant être réutilisés, resynthétisés, échangés, voire vendus, d'où cette notion de propriété intellectuelle qui est finalement restée comme label. Cette seconde révolution correspond à l'ère du SoC (System on a Chip), voire du NoC (Network on a Chip) qui est une conséquence directe des progrès

constants en densité d'intégration mais qui pose plus que jamais la question des méthodes de conception, pour parvenir à gérer la complexité accrue de ces « super puces ». Ainsi, *réutilisation* devient le maître-mot pour la conception des SoCs, et des efforts de standardisation ont été faits en ce qui concerne les formats d'échange, de documentation, également pour la protection de la propriété intellectuelle de ces IPs [VSIA].

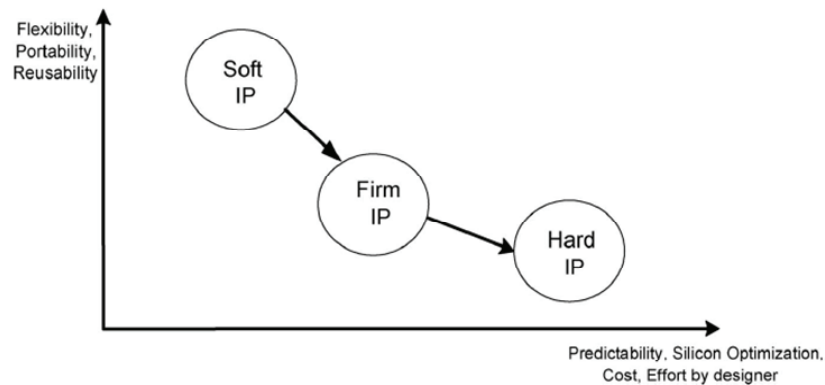


**Figure 1 : Flot de conception numérique à base d'IPs [LEG05]**

La fig.1 illustre le flot de conception typiquement utilisé en conception de systèmes numériques. Il repose sur une base de données d'IPs ; à droite sont représentées les étapes de conception des IPs, à gauche les étapes de conception du SoC. Les primitives du concepteur de SoC sont ces IPs, qu'il manipule, assemble en une architecture devant répondre aux spécifications.

La conception d'un bloc IP diffère de la conception d'un ASIC à application unique et cela peut nécessiter jusqu'à 5 fois plus de temps de développement [SWM06]. Les trois grandes étapes de conception d'un IP numérique sont : (i) sa spécification et documentation, (ii) son implémentation avec des techniques de description standards, (iii) sa vérification fonctionnelle ou comportementale. Dans le domaine numérique, c'est la troisième étape qui s'avère la plus consommatrice de temps (jusqu'à 50% du temps de conception global) car il s'agit d'anticiper sur toute la gamme des applications possibles et donc d'assurer une couverture de test proche de 100%.

Un IP numérique peut prendre différentes formes selon la manière dont le développeur/vendeur souhaite fournir ce bloc. Ainsi on trouve trois catégories d'IP : les *Soft*, *Firm* ou *Hard* IPs. La fig.2 illustre le type de compromis réalisé par chaque catégorie d'IP.



**Figure 2 : Différents types de blocs IP [SWM06]**

Les *Soft IPs* se présentent sous la forme de descriptions comportementales en VHDL. Ils peuvent ainsi être synthétisés au niveau portes dans de multiples applications ; il en découle un niveau de flexibilité, portabilité et « réutilisabilité » maximal mais l'impossibilité de prédire les principales performances (fréquence, consommation, etc.) qui dépendent essentiellement du processus de fabrication et de l'application elle-même.

Les *Hard IPs* se présentent sous la forme de layouts spécifiquement optimisés pour une application. Ils possèdent ainsi des performances prédictibles mais leur réutilisabilité s'en trouve réduite, on peut les rapprocher des cellules pré-caractérisées utilisées en conception *semi-custom*.

Les *Firm IPs* réalisent un compromis intermédiaire, ils se présentent souvent comme partiellement optimisées, avec des sous-blocs RTL « pré-synthétisés » à partir d'une bibliothèque générique, un plan de masse donnant le placement relatif des sous-blocs ; le concepteur a ainsi la possibilité de peaufiner l'optimisation pour ses propres besoins.

Jusqu'à récemment la plupart des blocs IPs étaient fournis sous la forme de *Hard IPs* vendus avec des modèles comportementaux pour faciliter l'étape de vérification dans la conception du SoC ; c'était le cas des cœurs de processeurs ARM ou MIPS. A présent, les *Soft IPs* leur sont préférés, les plus répandus vont des blocs d'interface (USB, UART, PCI) aux blocs réseaux (ATM, Ethernet), en passant par les microcontrôleurs et microprocesseurs [SWM06].

Pour replacer ces importants progrès de productivité dans leur contexte initial, on peut se rappeler qu'à la base le traitement numérique du signal relève d'une abstraction drastique

consistant à représenter une information sur 2 états ; il en découle l'application des règles de l'algèbre de Boole et la possibilité de construire des fonctions plus ou moins élaborées à l'aide d'un petit nombre de primitives, inchangées depuis les débuts de l'électronique numérique et robustes aux évolutions technologiques. C'est ainsi que la hiérarchie des fonctions de traitement numérique a pu s'établir et se conforter au fil du temps et que les méthodes de synthèse ont pu être automatisées.

## **1.2. Problématique des blocs IPs analogiques et état de l'art**

L'explosion du marché des applications sans fil a engendré des besoins en intégration de systèmes mixtes, où parties analogiques et numériques coexistent sur le même substrat ; ainsi 60% des circuits intégrés actuels incluent des blocs de traitement analogiques [CFG06]. Dans l'idéal, le flot de conception analogique devrait être calqué sur le flot de conception numérique, ce qui est loin d'être le cas aujourd'hui. Soulignons que le contexte initial est bien différent et les relations à la base de la conception analogique sont loin d'être aussi simples que celles de l'algèbre de Boole ; le concepteur doit faire face à des relations fonctionnelles fortement non-linéaires, paramétrées par des variables technologiques et sensibles à différents parasites (température, bruit, mis-match, etc.). Le traitement analogique du signal, s'il s'avère plus performant en fréquence et en consommation, relève d'une conception plus ardue, liée à l'expérience unique du concepteur. En conséquence, il n'existe pas de consensus sur le processus de conception analogique et les fonctions primitives peuvent être très variées selon les cas : conception en mode tension ou en mode courant, conception Bipolaire ou CMOS, conception MOS en forte ou faible inversion, etc.

En dépit de ce contexte initial nettement différent, le domaine de la conception analogique a intérêt à s'inspirer de l'évolution méthodologique de la conception numérique. Dans un premier temps, sont apparus les langages standards de description étendus à l'analogique, VHDL-AMS puis Verilog-AMS : la synthèse automatique n'était pas à l'ordre du jour, de loin, mais il devenait évident que la modélisation comportementale était une des manières de gérer la complexité de la conception. Dans un deuxième temps, puisque la conception de SoC revient aujourd'hui à intégrer et à faire communiquer des circuits préconçus plus ou moins ajustables, la question des IPs analogiques s'est posée.

Est-il possible de construire un flot de conception de SoC-AMS similaire à celui de la fig.1 ? Sous quelle forme doivent être archivés les IPs analogiques ? Comment doivent-ils être intégrés et ajustés dans le processus de conception SoC ?

### *Hard IP ?*

Parmi les opérateurs typiques en conception analogique, on peut citer les amplificateurs opérationnels, amplificateurs de transconductance, convertisseurs A/N et N/A, les PLL, les filtres, les références de tension et modules RF. Ces blocs sont traditionnellement conçus et optimisés pour une application spécifique, en tentant de réduire au maximum la consommation et l'aire : la tendance naturelle est donc au *Hard IP*. C'est d'ailleurs ainsi que l'envisageait le consortium VSIA<sup>1</sup> en 2001 dans son guide des étapes de conception et de documentation d'un IP AMS [VSI01].

Dans ce cas, la question de la migration vers d'autres technologies ou d'autres applications se résout par une re-conception plutôt qu'une ré-utilisation. Il s'agit donc d'introduire de la flexibilité dans l'IP analogique pour le faire passer à l'état de *Firm IP*, voire de *Soft IP*.

### *Soft IP ?*

A quoi ressemblerait un *Soft IP* en analogique ? Il est tout-à-fait possible de décrire dans un langage de conception HDL la fonctionnalité souhaitée d'un bloc analogique. Le problème est qu'il n'existe aujourd'hui aucun processus de synthèse universel délivrant un schéma électrique dimensionné pour une application. Cela demande tout d'abord de sélectionner un mode de conception, des primitives, une topologie, puis de procéder au dimensionnement. Un outil capable d'assister l'étape de sélection de topologie serait une importante avancée pour la synthèse analogique. Citons les travaux prometteurs qui avaient été entrepris par Y. Deval dans sa thèse [DEV94] et la méthode d'analyse par intervalles proposée par J. Michel [PEPS], [MIS08] qui pourrait rapidement confirmer l'existence d'une solution au problème de dimensionnement, amenant ainsi à sélectionner ou bien éliminer certaines topologies. Enfin, notons que pour pouvoir effectuer une véritable sélection de topologie, il faut avoir au préalable archivé un ensemble de solutions documentées un minimum.

### *Firm IP ?*

A mi-chemin entre le *Soft* et le *Hard IP*, c'est le format *Firm IP* qui semble le plus approprié à l'analogique, pas seulement un layout figé ni une description purement logicielle

---

<sup>1</sup> Virtual Socket Interface Alliance : consortium d'industriels du semi-conducteur et de la CAO, fondé en 1996 dont la mission était de conforter le développement d'IP pour la conception des SoCs au travers de métriques et formats standards.



synthétisable, mais un circuit basé sur un schéma électrique, suffisamment paramétré et documenté pour pouvoir être réutilisé dans plusieurs applications.

Un certain nombre d'initiatives existent sur de thème de la réutilisation dans le flot de conception analogique et les efforts de ces précurseurs peuvent se structurer en points fondamentaux :

- 1) Le contenu du « paquetage IP »
- 2) La méthodologie de conception d'un IP
- 3) La méthodologie de réutilisation de l'IP dans le flot de conception système.

A propos du point 1) :

La question concerne les données caractéristiques de l'IP, ses paramètres d'ajustement, ses formats de description et de documentation adaptés à toutes les étapes du flot de conception (simulation d'architecture, dimensionnement, vérification, dessin des masques).

On peut citer les travaux de l'équipe de I. O'Connor du LEOM [CTR06]. Le contenu préconisé pour un IP analogique est le suivant :

- un ensemble « vues » : les ports d'entrée/sortie, le modèle fonctionnel, le modèle structurel
- un ensemble de paramètres : paramètres de performance, paramètres de conception, paramètres physiques
- un ensemble de relations entre ces paramètres :
  - la méthode d'évaluation ou relation pour passer des paramètres physiques aux paramètres de performance
  - la méthode d'extraction qui est aussi une méthode d'évaluation mais à partir de simulations et non pas analytique
  - la méthode de synthèse pour passer des paramètres de performance aux paramètres de conception
  - la méthode de distribution des contraintes pour formaliser le passage des spécifications de l'IP aux spécifications de ses sous-blocs.

Ces relations contiennent certainement l'expertise de conception et leur expression reste la partie la plus délicate de la conception de l'IP. Elles peuvent prendre une forme analytique dans une approche basée sur la connaissance, une forme procédurale quand le formalisme des étapes est clair, une forme itérative dans une approche basée sur la simulation et

l'optimisation, etc. Cette définition de l'IP analogique a été appliquée à des circuits d'amplification d'une chaîne de photo-réception CMOS et utilisée pour démontrer la possibilité d'une synthèse hiérarchique automatisée dans le domaine des systèmes optiques d'interconnexion.

Concernant le point 2) :

Il s'agit de guider le concepteur sur les étapes de développement d'un bloc réutilisable. L'effort à fournir est supérieur à celui de la conception d'un bloc à usage unique et c'est d'ailleurs une des raisons qui font que les méthodologies de réutilisation ne sont pas si répandues en analogique. Formuler l'expérience de conception pour qu'elle soit réutilisable n'est pas une tâche aisée et une des clés est la formalisation du processus de dimensionnement.

On peut citer ici les travaux de l'équipe de M.M. Louërat du LIP6 [NGU06] autour de l'outil CAIRO+, véritable plateforme de conception d'un IP réutilisable. En premier lieu, le concepteur est amené à décrire son circuit sous la forme d'un schéma électrique, d'un ensemble de paramètres pour le dimensionnement et d'un plan de masse donnant le placement relatif des cellules de layout. CAIRO+ possède un moteur de dimensionnement et de dessin des masques, paramétrable en fonction de la technologie, à condition que le concepteur ait exprimé de façon procédurale les étapes du dimensionnement. Cette tâche est elle-même assistée par une méthode de synthèse développée dans la thèse de R. Iskander [ISK08]. Cette équipe est une des rares à aborder la question de la génération automatique de layout dans l'esprit de la réutilisation.

Concernant le point 3) :

La question est de savoir comment l'IP va être intégré dans le flot de conception système. L'étendue de ce flot, depuis l'exploration d'architecture jusqu'à la vérification, confirme l'intérêt de modéliser l'IP à différents niveaux d'abstraction. Un autre aspect concerne l'archivage des IPs dans une base de données interrogeable ainsi que l'algorithme de recherche dans cette base de données. Enfin, lorsqu'un Firm IP est sélectionné, comment procéder à son ultime optimisation : c'est le *Firm IP hardening flow* qui dépend du formalisme adopté pour l'IP.

Dans le cas du LIP6, la description de l'IP dans CAIRO+ est paramétrée et mise en équation à un tel degré que la génération du Hard IP est automatique et ne recourt

pratiquement pas à l'optimisation. Des exemples marquants se trouvent dans la thèse de P. Nguyen-Tuong [NGU06].

Dans le cas du LEOM, le processus de synthèse est modélisé avec le langage UML et est inséré dans une plate-forme de conception (RUNE II) disposant de multiples outils de simulation et d'optimisation [TOG04].

D'autres équipes de chercheurs ont contribué au développement des méthodologies de réutilisation, on trouvera un état de l'art plus complet dans la thèse de T. Lévi [LEV07]. J'ai choisi de citer ici les travaux de chercheurs avec qui nous avons eu l'occasion de collaborer à travers l'AS<sup>2</sup> « SoC-AMS » [AS] ou encore le PEPS<sup>3</sup> « Unified IC Sizing Approach » [PEPS].

### **1.3. Exposé du cadre et des objectifs de ce travail**

Les développements et résultats présentés dans ce chapitre proviennent du travail de thèse de T. Lévi [LEV07]. Le sujet a été suscité par les réflexions menées dans le cadre de l'Action Spécifique SoC-AMS soutenue par le CNRS en 2003, sous la direction de P. Fouillat et P. Garda. J'ai animé le sous-programme SP3 « Méthodes et outils pour la conception des SoC-AMS », dont les conclusions se trouvent dans le rapport final [AS] et ont été présentées au Workshop du RTP SoC à Lalonde les Maures en mai 2004. Cette thèse s'inscrit pleinement dans la suite de cette action et a été financée par un contrat BDI.

L'originalité repose notamment sur le cadre d'application qui est la conception de circuits et systèmes neuromorphiques, tels que les développe l'équipe ISN<sup>4</sup> du laboratoire IMS. L'expérience en ingénierie neuromorphique de cette équipe avait déjà permis de poser clairement les bases de la hiérarchie du système et les primitives de la conception, ce qui constituait un terrain favorable au développement d'un flot de conception basé sur la réutilisation.

Parmi les points critiques à aborder et cités au paragraphe précédent, nous avons contribué à :

---

<sup>2</sup> Action Spécifique CNRS.

<sup>3</sup> Projet Exploratoire Puridisciplinaire CNRS.

<sup>4</sup> ISN : Ingénierie des Systèmes Neuromorphiques.

- la définition du contenu d'un IP, en réinvestissant notamment l'aspect modélisation multi-niveaux
- la méthode de conception d'un Firm IP, à partir d'un Hard IP
- l'organisation des IPs en une base de données
- la méthodologie de réutilisation d'un IP dans le flot de conception SoC.

## **2. Conception de Firm IPs pour l'ingénierie neuromorphique**

### **2.1. Hiérarchie, primitives et spécifications**

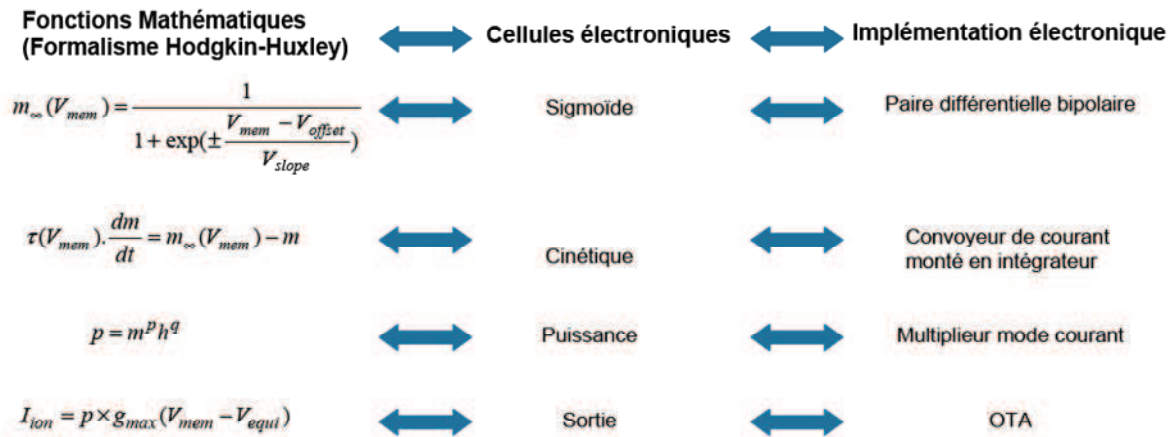
L'ingénierie neuromorphique est un domaine de recherche tourné vers la conception et la fabrication de réseaux de neurones artificiels imitant le comportement des réseaux de neurones biologiques. Les ASICs développés par l'équipe ISN sont basés sur un modèle électro-physiologique du neurone issu du formalisme Hodgkin et Huxley [HOH52] ; ils sont typiquement utilisés comme émulateurs en Neurosciences Computationnelles, pour explorer les principes d'adaptation et de transport de l'information dans le cerveau [LBA06], [LER07], [LBR09].

Le principe de construction modulaire des circuits neuromorphiques fait déjà partie du travail de thèse de Sylvain [SAI04]. On peut considérer qu'il s'agit d'une première version de la bibliothèque d'IPs, sous la forme de *Hard IPs*, réalisés en technologie AMS BiCMOS 0.35  $\mu\text{m}$ . En effet, le choix est fait de focaliser la conception des ASICs sur la répétition d'un nombre fini de cellules élémentaires (**sigmoïde**, **cinétique**, **puissance**, **sortie**, voir fig.3) ; chaque cellule est un opérateur de calcul analogique et l'assemblage de ces opérateurs permet de résoudre, en temps réel<sup>5</sup>, les équations du formalisme de Hodgkin et Huxley.

La fig.3 indique la correspondance entre les fonctions mathématiques du modèle de neurone de Hodgkin et Huxley (HH) et les cellules analogiques utilisées pour leur réalisation. On peut noter le choix d'une conception en mode courant, la seule tension véhiculée dans le système étant une image directe du potentiel de membrane  $V_{\text{mem}}$ , s'établissant entre l'intérieur et l'extérieur du neurone.

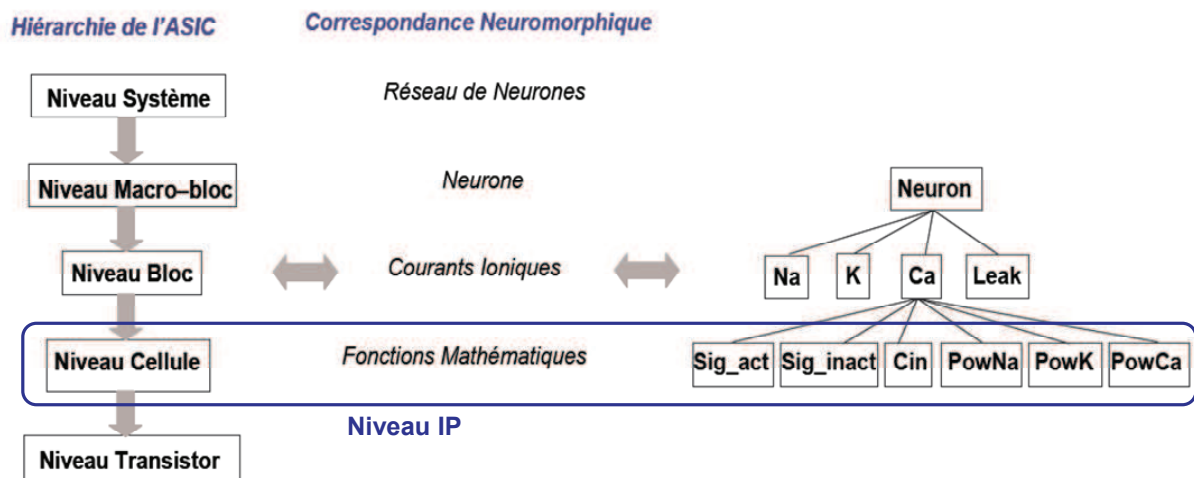
---

<sup>5</sup> Le terme « temps réel » signifie ici que le circuit se comporte avec exactement la même dynamique que le réseau de neurone biologique.



**Figure 3 : Réalisation électronique des équations du modèle de HH**

A partir de ces opérateurs de base, il est possible de construire la hiérarchie d'un système intégré émulant un réseau de neurone biologiquement réaliste ; cette hiérarchie est décrite fig.4. Le niveau IP correspond aux fonctions mathématiques récurrentes et les niveaux hiérarchiques N+1 sont des descriptions structurelles faites à partir des primitives du niveau N.



**Figure 4 : Hiérarchie d'un ASIC neuromimétique**

Les spécifications d'un tel système sont un peu atypiques, en regard de la conception analogique traditionnelle. En effet, nos collaborateurs du domaine des Neurosciences sont intéressés par :

- le nombre de neurones qu'il est possible de faire interagir (défini au *Niveau Système*)
- le nombre de courant ioniques considéré pour chaque neurone (défini au *Niveau Macro-bloc*)
- la possibilité de paramétrer à loisir ces courants ioniques (paramètres électrophysiologiques définis au *Niveau Bloc* et propagés directement au *Niveau Cellule*).

En effet, la variété des comportements dynamiques du neurone observés en biologie est liée au nombre de courants ioniques pris en compte dans le modèle et au jeu de paramètres électro-physiologiques associé. Un exemple de spécifications est donné ci-dessous : il s'agit d'un réseau de 2 neurones, le Neurone 1 doit avoir 3 courants ioniques (*Na*, *K*, *Leak*) et 2 synapses, le Neurone 2 doit avoir 4 courants ioniques (*Na*, *K*, *Leak*, *Slow conductance*) et 1 synapse.

Courants ioniques	Voffset (mV)	Vpente (mV)	Tau (ms)	Vequi (mV)	gmax ( $\mu$ S)
<b>Neurone 1</b>					
Na	-37	7.2	0.03	50	110
K	-37	11.38	3	-100	22
Leak				50	0.22
Synapse 1			5	52	0.1
Synapse 2			4	45	0.1
<b>Neurone 2</b>					
Na	-41	5.7	0.08	80	70
K	-31	13.81	5	-120	13
Leak				50	0.31
Slow conductance	-35	11.4	8	-700	50
Synapse 1			4	10	0.1

**Tableau 1 : Exemple de spécifications d'un ASIC neuromimétique**

Pour chacun des courants ioniques considérés, les spécifications sont données sous la forme d'une liste de valeurs pour les paramètres électro-physiologiques  $V_{offset}$ ,  $V_{slope}$ ,  $Tau$ ,  $V_{equi}$  et  $g_{max}$ . Pour garantir une certaine reconfigurabilité des ASICs si ce jeu de paramètres venait à changer, les performances recherchées pour les cellules élémentaires répertoriées fig.3 s'expriment en termes de dynamique : dynamique d'entrée en tension pour la fonction **sigmoïde**, dynamique de constante de temps pour la fonction **cinétique** (typiquement  $[\tau, 50\tau]$ ) ou de dynamique d'entrée en tension et dynamique de  $g_{max}$  pour la fonction **sortie** (typiquement  $[g_{max}, 50 g_{max}]$ ).

## 2.2. Contenu formel d'un IP

Chaque IP possède 7 vues informatives récapitulées dans le Tableau 2. Les vues **Symbole**, **Schéma** et **Layout** existaient déjà dans la bibliothèque de *Hard IPs* initiale.

Nom des vues :	Description / Rôle	Format de description
<b>Symbole</b>	Visualise la fonction	Graphique
<b>Connectique</b>	Vérifie les connexions entre les différents blocs	VerilogA
<b>Fonctionnelle</b>	Modélise le comportement électrique idéal	VerilogA
<b>Comportementale</b>	Modélise le comportement électrique non idéal, extrait du schéma	VerilogA
<b>Schéma</b>	Présente le schéma au niveau transistor	Graphique / netlist
<b>Layout</b>	Dessin des masques dans la technologie cible	Graphique / GDS-II
<b>Documentation</b>	Voir ci-dessous...	Texte

**Tableau 2 : Liste des vues contenues dans un IP**

Des vues correspondant à des modèles de différents niveaux d'abstraction (**connectique**, **fonctionnelle** et **comportementale**) ont été rajoutées. Le fichier **documentation** contient l'expertise et les données de la conception telles que :

- la procédure suivie pour la conception
- les paramètres de performances et paramètres de conception du circuit
- les relations comportementales entre paramètres de performances et paramètres de conception
- la caractérisation du bloc en vue de la réutilisation
- les informations concernant le layout déjà conçu (technologie, tension d'alimentation, surface et coût)

### 2.3. Conception d'un Firm IP

La conception « en vue de la réutilisation » ne s'arrête pas à la conception du schéma électrique dimensionné et du layout (version *Hard IP*), elle consiste dans notre cas à concevoir les modèles hiérarchiques, à caractériser le circuit et à renseigner le fichier de documentation. C'est cette méthodologie qui est ici précisée en prenant pour exemple le bloc **sigmoïde**.

#### 2.3.1 Procédure de conception

Comme précisé fig.3, la fonction mathématique du formalisme de HH à imiter la suivante :

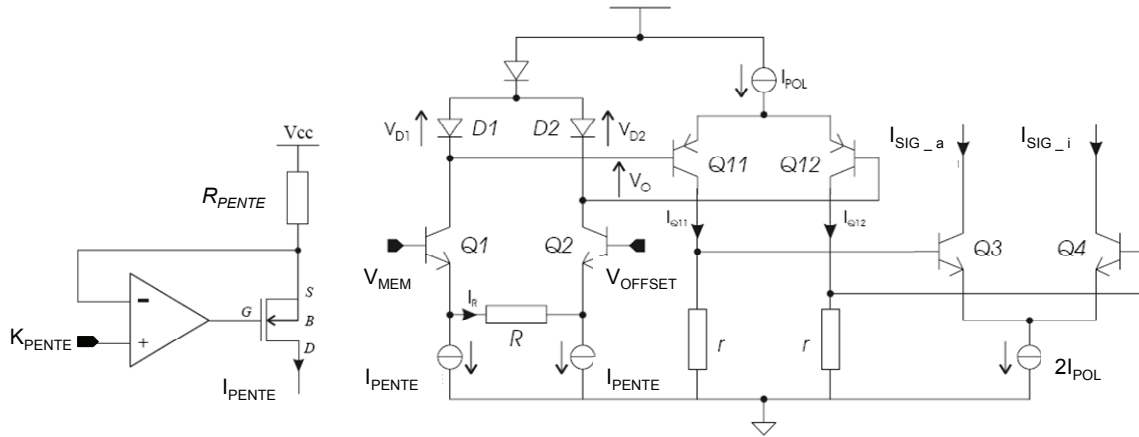
$$m_{\infty}(V_{mem}) = \frac{1}{1 + e^{\frac{-(V_{mem} - V_{offset})}{V_{pente}}}}$$

Dans la pratique, on choisit de représenter la variable  $m_{\infty}$  par un courant normalisé par  $2I_{pol}$ .

L'équation fonctionnelle électrique devient alors :

$$I_{m_x} = \frac{2I_{pol}}{1 + e^{\frac{-(V_{mem}-V_{offset})}{V_{pente}}}}$$

La topologie choisie pour émuler cette fonction est basée sur une paire différentielle qui prend directement en entrée les tensions  $V_{mem}$  et  $V_{offset}$  ; le contrôle du paramètre de pente  $V_{pente}$  du modèle se fait par l'intermédiaire d'un convertisseur courant-tension, piloté par la tension  $K_{pente}$ . Le schéma est donné ci-après :



**Figure 5 : Schéma électrique du bloc sigmoïde**

Un premier sous-bloc et a pour équation fonctionnelle :

$$I_{pente} = \frac{V_{cc} - K_{pente}}{R_{pente}}$$

Le second sous-bloc produit la fonction sigmoïdale, pour l'activation ( $I_{sig\_a}$ ) et l'inactivation ( $I_{sig\_i}$ ) d'un canal ionique :

$$I_{sig\_a} = \frac{2I_{pol}}{1 + e^{\frac{r \cdot I_{pol} \cdot (V_{mem} - V_{offset})}{R \cdot U_T \cdot I_{pente}}}} \quad I_{sig\_i} = \frac{2I_{pol}}{1 + e^{\frac{r \cdot I_{pol} \cdot (V_{mem} - V_{offset})}{R \cdot U_T \cdot I_{pente}}}}$$

Ainsi le réglage du paramètre  $V_{pente}$  se fait au moyen de l'équation suivante :

$$V_{pente} = \frac{R U_T}{r I_{pol}} \frac{V_{cc} - K_{pente}}{R_{pente}}$$

Nous pouvons ainsi extraire les paramètres utiles à cette procédure de conception. Les paramètres de « performances », c'est-à-dire ceux sur lesquelles vont porter les spécifications, sont :  $V_{mem}$ ,  $V_{offset}$  et  $V_{pente}$  (ou bien  $K_{pente}$  qui lui est linéairement relié). Les paramètres de conception, c'est-à-dire les variables d'ajustement du schéma électrique sont :  $R_{pente}$ ,  $R$ ,  $r$  et  $I_{pol}$ . Les équations précédentes permettent de relier analytiquement un jeu de paramètres à un autre, selon une méthode basée sur la connaissance.



### 2.3.2 Modélisation

Les modèles sont envisagés à 3 niveaux d'abstraction et reprennent en cela les méthodes exposées au **Chapitre 1** de ce mémoire ; un développement succinct de cet aspect sera donc fait ici. Les modèles ont été écrits en VerilogA dans le but unique de conserver l'environnement habituel des concepteurs, à savoir la plateforme CAO de Cadence.

En reprenant l'exemple du bloc **sigmoïde**, le *modèle fonctionnel* reprend les équations du paragraphe précédent, les impédances d'entrées/sorties étant idéales. Le *modèle comportemental* provient d'une analyse plus détaillée du schéma transistor, y adjoint les impédances d'entrées/sorties ainsi que des effets non-idéaux tels que le gain de l'AOP dans le convertisseur tension-courant et le coefficient de recopie des miroirs de courant. Le *modèle connectique* est un modèle envisagé pour la vérification système très-haut niveau ; il s'agit d'un bloc quasiment vide d'un point de vue fonctionnel, il est simplement chargé de détecter en cours de simulation un problème de connexion entre blocs. Le principe revient à vérifier que les tensions et courants appliqués au bloc appartiennent à un intervalle plausible ; si c'est le cas, la sortie prend une valeur nominale sinon elle prend la valeur zéro.

### 2.3.3 Caractérisation en vue de la réutilisation

La caractérisation de l'IP concerne l'étude et la mesure des performances en général. Dans le cadre de la réutilisation, elle prend un sens un peu différent car il s'agit de savoir dans quelles conditions le bloc pourra subir une migration de spécifications. Nous avons précédemment vu que les grandeurs sur lesquelles vont porter les spécifications de la cellule **sigmoïde** sont les tensions d'entrée  $V_{mem}$ ,  $V_{offset}$  et  $K_{pente}$ . L'IP a été dimensionné une première fois pour garantir un bon fonctionnement pour des valeurs précises de ces tensions. Dans quelle mesure garde-t-il sa fonctionnalité pour un autre jeu de valeurs ? Pour déterminer cela, une analyse systématique a été effectuée pour déterminer les intervalles de  $V_{mem}$ ,  $V_{offset}$  et  $K_{pente}$  dans lesquels la fonction sigmoïdale est conservée. Cette étude est menée en effectuant une série de simulations (AC, DC, transitoire et Monte-Carlo) sur le schéma électrique dimensionné selon les spécifications initiales.

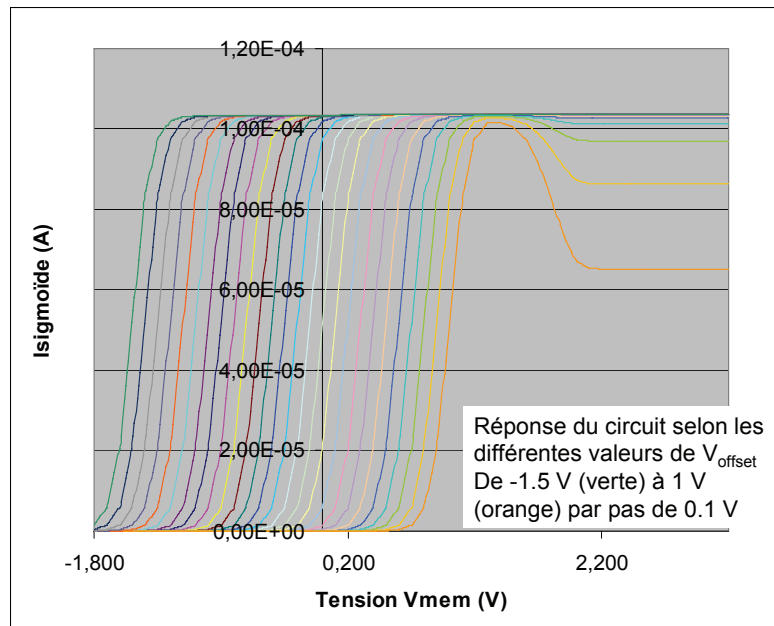
A titre d'exemple, les intervalles de validité de tensions d'entrée du bloc **sigmoïde** sont :

Parameters de performances	Minimum	Maximum
$V_{mem}$	1 V	4.5 V
$V_{offset}$	1.6 V	3.2 V
$K_{pente}$	1.7 V	4.7 V

**Tableau 3 : Domaine de validité du bloc sigmoïde**

Cela signifie simplement que l'IP **sigmoïde** pourra être réutilisé sans modification dans sa conception, à condition que les spécifications sur  $V_{mem}$ ,  $V_{offset}$  et  $K_{pente}$  soient comprises dans ces intervalles.

Notons que les procédures précises de caractérisation de ces intervalles de validité sont données dans le fichier *documentation*. Le cas du bloc **sigmoïde** est traité intégralement dans la thèse de T. Lévi [LEV07]. En guise d'illustration la Figure 6 représente les réponses du circuit ( $I_{sig\_a}$ ) à un balayage DC de la tension  $V_{mem}$ , pour plusieurs valeurs de  $V_{offset}$ .



**Figure 6 : Réponse DC du circuit sigmoïde pour plusieurs valeurs de  $V_{offset}$**

Ces réponses sont comparées à la réponse idéale de la fonction sigmoïdale, en mesurant l'erreur quadratique moyenne. L'intervalle de validité en  $V_{offset}$  est alors déterminé afin que cette erreur ne dépasse pas 1 %. Ces traitements ont été automatisés par un programme en langage Perl permettant aussi de récupérer les données de caractérisation dans un fichier Excel.

### **3. Flot de conception basé sur la réutilisation**

#### **3.1. Organisation générale du flot**

Les 4 blocs IP récurrents dans la conception des ASICs ont été conçus, modélisés et caractérisés selon la démarche décrite au paragraphe précédent. La question abordée maintenant concerne l'organisation de ces IPs en une base de données qui sera le pivot du flot de conception d'un ASIC. Un organigramme typique de ce flot est donné fig.7.

En amont de la conception, les modèles fonctionnels présents dans la base de données servent à valider une architecture paramétrée par les spécifications ; la base de données est alors interrogée afin de savoir si des IPs peuvent être réutilisés. Si oui, les différentes vues contenues dans l'IP sont récupérées pour les étapes finales de la conception ; si non, nous verrons comment les informations de réutilisation peuvent servir à ajuster un IP aux besoins spécifiques.

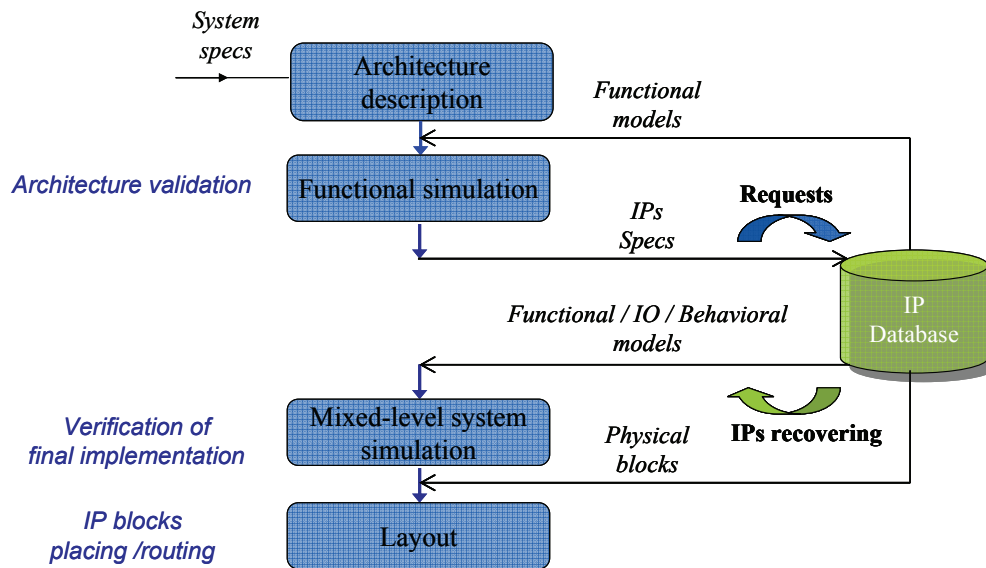


Figure 7 : Flot de conception système préconisé

### 3.2. Structure de la base de données et algorithme d'exploration

Les IPs sont rassemblés en une base de données SQL<sup>6</sup> hébergée sur un serveur Apache [BOC00]. Dix tables ont été nécessaires pour la décrire car non seulement chaque IP et ses données de caractérisation y ont été répertoriés, mais également la hiérarchie du système décrite fig.4, de façon à pouvoir faire une exploration hiérarchique descendante. La fig.8 suivante est une copie d'écran de la table se saisie du contenu d'un IP.

Cette base de données est incluse dans une plateforme d'exploitation qui permet de transformer les spécifications (saisies sur un poste distant par le « client ») en requêtes (envoyées au poste serveur du concepteur). Cela a nécessité l'emploi des langages PHP<sup>7</sup> [RIG06], HTML et XML<sup>8</sup> [ABS00].

<sup>6</sup> SQL : Structured Query Language.

<sup>7</sup> PHP est un langage de développement de scripts, pouvant être intégré à du code HTML de pages Web.

<sup>8</sup> XML est un langage de balisage générique servant essentiellement à stocker/transférer des données structurées en champs arborescents.

phpMyAdmin

Accueil

































































































reseauneuronnes (10)

reseauneuronnes

- blocks
- current\_level
- function\_level
- io\_final
- ip
- ipcurrent\_links
- ipneuron\_links
- ipsystem\_links
- lowerblocklinks
- neuron\_level

Base de données reseauneuronnes - Table ip sur le serveur localhost

Structure Afficher SQL Sélectionner Insérer Exporter Opérations Vider Supprimer

Champ	Type	Attributs	Null	Défaut	Extra	Action
<input type="checkbox"/> name_ip	varchar(30)		Non			     
<input type="checkbox"/> Level	int(11)		Non	0		     
<input type="checkbox"/> values	int(11)		Non	0		     
<input type="checkbox"/> symbol	varchar(50)		Non			     
<input type="checkbox"/> connectical	varchar(50)		Non			     
<input type="checkbox"/> functional_model	varchar(50)		Non			     
<input checked="" type="checkbox"/> behavioural_model	varchar(50)		Non			     
<input type="checkbox"/> schematic	varchar(50)		Non			     
<input type="checkbox"/> layout	varchar(50)		Non			     
<input type="checkbox"/> technology	varchar(30)		Non			     
<input type="checkbox"/> supply_voltage	float		Non	0		     
<input type="checkbox"/> Nb_inputs	float		Non	0		     
<input type="checkbox"/> Nb_outputs	float		Non	0		     
<input type="checkbox"/> layout_area	float		Non	0		     
<input type="checkbox"/> ip_cost	float		Non	0		     
<input type="checkbox"/> datasheet	varchar(50)		Non			     



Tout cocher / Tout décocher Pour la sélection:  

Figure 8 : Saisie d'écran de l'interface de gestion de la table « IP »

La fig.9 est une saisie d'écran du formulaire de spécifications qu'est amené à remplir le « client ».

## Summary of the system specifications of levi

---

You already have an account

---

You want to have 2 neurons in your network

---

### Neuron 1

---

Cmem\_bio in uF/cm2

Area of neuron 1 in cm2

---

Please choose your ionic currents:

- Sodium
- Potassium
- Leak
- Slow conductance 1
- Slow conductance 2

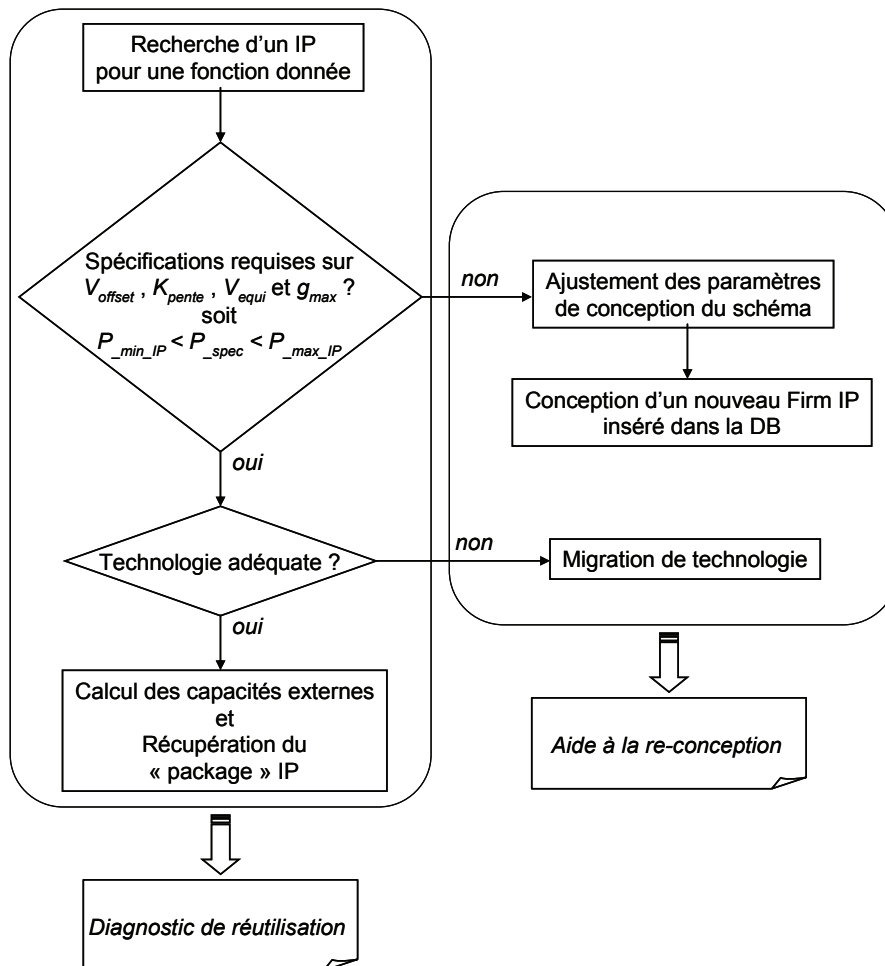
How many multisynapses for this neuron do you want ?

Figure 9 : Formulaire de saisie des spécifications

Un script PHP est ensuite chargé d'extraire les données de ce formulaire de spécifications, de les organiser dans un fichier texte et de produire un compte-rendu pour le client. Toujours en PHP, ces données de spécifications sont transformées en un algorithme de requêtes vers la base de données SQL. Les diagrammes d'activité UML de ces 2 étapes sont précisés dans la thèse de T. Lévi.

Les critères mis en avant pour l'exploration de la base de données d'IP sont la technologie souhaitée et les spécifications, données sous la forme de valeurs des paramètres  $V_{offset}$ ,  $K_{pente}$ ,  $Tau$ ,  $V_{equi}$ ,  $g_{max}$ , comme le montre l'exemple du Tableau 1. La réutilisabilité de chaque IP de la base de données est caractérisée sous la forme d'intervalles de validité  $[P_{min}, P_{max}]$  pour les paramètres  $V_{offset}$ ,  $K_{pente}$ ,  $V_{equi}$  et  $g_{max}$ ; quant au paramètre de la cinétique,  $Tau$ , il dépend d'une valeur de capacité externe, connectée au bloc **cinétique**.

Ainsi, l'algorithme d'exploration de la base de données d'IP est basé sur le principe illustré par l'organigramme suivant :



**Figure 10 : Principe de l'exploration de la base de données d'IP**

### 3.3. Aide à la réutilisation

#### 3.3.1 Diagnostic de réutilisation

L'exploration une fois terminée, l'interface HTML renvoie au concepteur un tableau, appelé diagnostic de réutilisation, dans lequel sont identifiés : les IPs correspondant aux spécifications et à la technologie choisie (en vert sur la fig.11) et les IPs ayant un défaut de spécification ou de technologie (en rouge sur la fig.11).

neuron 1						
	Activation	Inactivation	activation kinetic	inactivation kinetic	Power	Output Level
Sodium	■	■	■	■	■	■
Potassium	■		■		■	■
Leak						■
Synapse 1			■			■
Synapse 2			■			■

neuron 2						
	Activation	Inactivation	activation kinetic	inactivation kinetic	Power	Output Level
Sodium	■	■	■	■	■	■
Potassium	■		■		■	■
Leak						■
Slow conductance 1	■	■	■	■	■	■
Synapse 1			■			■

■	IP found
■	IP not found

Figure 11 : Exemple de diagnostic de réutilisation pour la conception d'un ASIC

Les parties suivantes abordent les deux cas de figure pouvant provoquer un échec de l'exploration, c'est-à-dire un problème concernant les spécifications ou bien la technologie de fabrication.

#### 3.3.2 Migration des spécifications

L'exemple illustré fig.11 indique que les spécifications de la fonction **sigmoïde** d'inactivation du courant Sodium du neurone 1 n'ont pas pu être satisfaites lors de l'exploration. Le concepteur doit alors concevoir un nouvel IP mais, si possible, en ajustant les paramètres de conception de l'IP de la base de données dont les performances sont « les plus proches » des spécifications  $P_{spec}$ . Comme la « couverture des performances » est exprimée sous la forme d'intervalles de validité pour un paramètre  $P$ , il s'agit donc de procéder à une nouvelle exploration de la base de données en minimisant la distance :

$$P_{spec} - \frac{P_{max\_IP} - P_{min\_IP}}{2}$$

Une fois trouvé cet IP « le plus proche », l'équation reliant les paramètres de conception aux paramètres de performances, donnée dans la vue *documentation*, permettra d'ajuster

rapidement le dimensionnement du schéma pour satisfaire les spécifications. Le concepteur devra alors concevoir un nouveau layout, caractériser ce nouveau bloc (son domaine de validité en particulier) et ainsi alimenter la base de données avec un nouvel IP. Les différents modèles n'ont pas besoin d'être redéveloppés puisqu'il appartient à la même famille fonctionnelle que l'IP d'origine. Un exemple a été traité dans [LLT08].

### 3.3.3 Migration technologique

Ce besoin peut arriver lorsqu'un IP de la base de données rentre bien dans les spécifications mais que le concepteur souhaite le réaliser dans une technologie différente de sa technologie d'origine. Ce propos n'a pas été traité de façon systématique pour tous les blocs IP précédents, mais une piste méthodologique a été étudiée sur le cas d'un OTA, bloc récurrent en conception analogique et rentrant dans la conception de la fonction **sortie** des circuits neuromorphiques.

Le problème consiste donc à partir d'un circuit dimensionné dans une technologie, de conserver la même topologie, de conserver les performances et de calculer facilement les nouvelles dimensions lorsqu'on change de technologie, d'où le terme de redimensionnement. Tout d'abord, une étude de l'état de l'art a été effectuée [GSC02], [ASA02], [SCK06]. Ces travaux ont pour point commun de baser les équations de redimensionnement sur le modèle ACM<sup>9</sup> du transistor MOS, qui reste peu usité, que ce soit dans les simulateurs ou pour les estimations faites en général par le concepteur. Les travaux présentés dans [SCK06] arrivent à un degré d'automatisation total en adjoignant diverses optimisations à la méthode analytique proposée par [GSC02]. Mais finalement, le principal inconvénient de toutes ces méthodes concerne l'aire du circuit redimensionné puisque que l'aire active du circuit est multipliée par 2 ou 4, lorsque la migration s'effectue vers une technologie pourtant plus fine que la technologie de départ.

La méthode proposée dans la thèse de T. Lévi utilise le modèle de niveau 1 du transistor MOS afin d'obtenir une première estimation du redimensionnement et une optimisation succède à cette première phase si nécessaire. Les détails ont été exposés dans plusieurs communications [LLT06], [LLT07c], [LTL07b] et un article [LTL09], et la méthode appliquée avec succès à un OTA ainsi qu'à un oscillateur en anneau, pour évaluer son impact

---

<sup>9</sup> ACM : Advance Compact Model ; ce modèle de transistor MOS est une version simplifiée du modèle EKV.

sur des circuits linéaires et non-linéaires. Nous ne donnerons ici qu'un exemple concernant un OTA.

La méthode peut être ramenée à trois étapes :

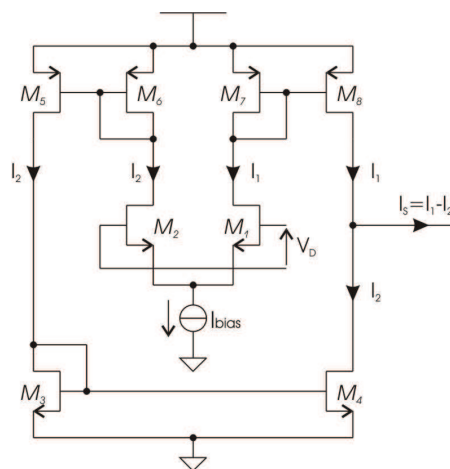
- 1<sup>ère</sup> étape : Définition des performances à conserver
- 2<sup>ème</sup> étape : Calcul des facteurs de redimensionnement technologiques
- 3<sup>ème</sup> étape : Définition de la stratégie et calcul des facteurs de redimensionnement électriques et géométriques.

La 1<sup>ère</sup> étape pose les objectifs du redimensionnement. La 2<sup>ème</sup> est une simple application numérique qui dépend de la technologie ciblée. La 3<sup>ème</sup> étape rassemble les équations disponibles pour calculer les facteurs de redimensionnement inconnus. Elle requiert un choix stratégique de la part du concepteur, afin d'avoir un nombre d'équations suffisant pour déterminer les facteurs de redimensionnement et ainsi les nouvelles tailles des transistors.

La fig.12 donne la définition des facteurs de redimensionnement qui sont les variables utilisées dans la méthode. La fig.13 reproduit le schéma électrique de l'OTA.

Parameters	Scaling Factors	
Supply Voltage : $V_{DD}$	$K_V = V_{DD2} / V_{DD1}$	} <b>Technological Factors</b>
Oxide Capacitance : $C_{OX}$	$K_{OX} = C_{OX2} / C_{OX1}$	
Mobility : $\mu_0$	$K_\mu = \mu_{02} / \mu_{01}$	} <b>Electrical Factor</b> Depends on the designer strategies
Effective Gate Voltage : $V_{EG} = V_{GS} - V_T$	$K_{EG} = V_{EG2} / V_{EG1}$	
Drain current : $I_D$	$K_I = I_{D2} / I_{D1}$	
Length of the Transistor : L	$K_L = L_2 / L_1$	} <b>Geometrical Factors</b>
Width of the Transistor : W	$K_W = W_2 / W_1$	

**Figure 12 : Définition des facteurs de redimensionnement, partant d'une technologie 1, vers une technologie 2**



**Figure 13 : Schéma électrique de l'OTA**



La technologie initiale est AMS BiCMOS 0,8 $\mu$ m/5V et la migration a lieu vers les technologies AMS BiCMOS 0,35 $\mu$ m/3.5V, ST CMOS 0,25 $\mu$ m/2.5V et ST CMOS 0,12 $\mu$ m/1.2V.

Afin de mettre en évidence l'importance de l'étape 3 relative à la stratégie spécifique du concepteur, nous avons imaginé un cas où le principal objectif est de conserver le gain en transconductance et de réduire de 30% la consommation en courant, même si cela se fait au prix d'une dynamique en courant diminuée.

Les 3 étapes de la méthode s'appliquent alors ainsi :

- 1<sup>ère</sup> étape :

Conservation du gain en transconductance  $G_{OTA}$  et diminution d'1/3 de  $I_{DD}$ .

- 2<sup>ème</sup> étape :

Les facteurs de redimensionnement technologiques  $K_V$ ,  $K_{OX}$ , et  $K_\mu$  sont calculés pour la technologie cible.

- 3<sup>ème</sup> étape :

L'application de la loi quadratique du modèle de niveau 1 donne:

$$K_I = \frac{K_\mu K_{ox} K_W K_{EG}^2}{K_L}$$

Nous choisissons de conserver le niveau d'inversion de chaque transistor pour ne pas dépolariser le circuit et prenons ainsi :

$$K_{EG} = 1$$

D'autre part, les objectifs définis dans l'étape 1 nous imposent :

$$K_I = \frac{I_{D2}}{I_{D1}} = 2/3$$

La conservation du  $g_m$  s'impose également ; pour cela, si l'on se réfère à l'expression théorique de  $g_m$ , la diminution de  $I_D$  peut être compensée de 2 façons : soit en augmentant  $W$  d'un facteur  $K_I$ , soit en diminuant  $L$  du même facteur, tout en restant dans les limites technologiques. C'est la 2<sup>ème</sup> solution qui a été choisie :

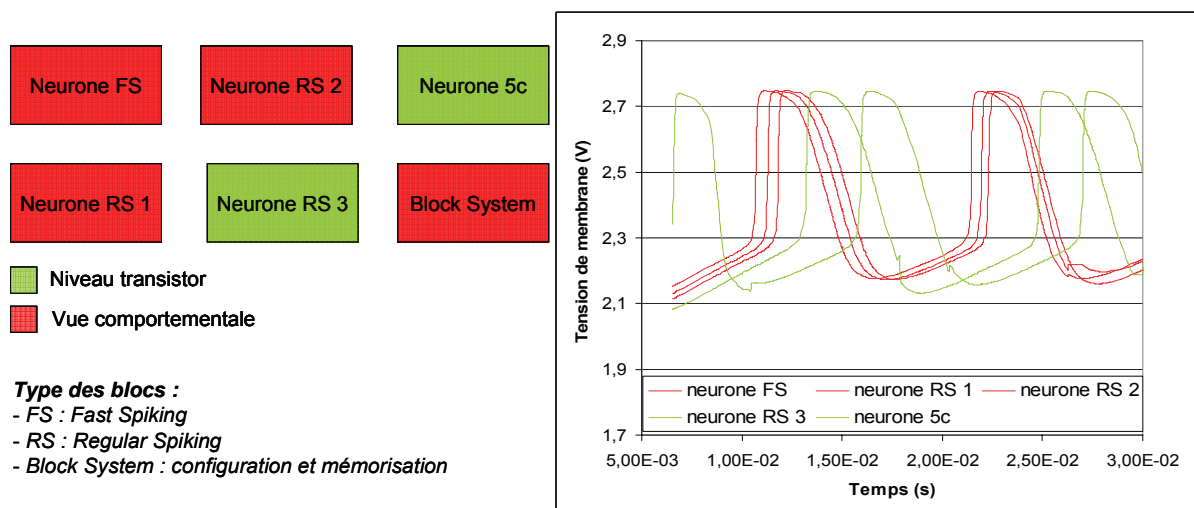
$$K_L = \frac{L_{MIN2}}{K_I L_{MIN1}}$$

Selon cette stratégie, on détermine finalement  $K_W$  par :

$$K_W = \frac{K_L K_I}{K_\mu K_{ox} K_{EG}^2}$$

### 3.4. Vérification

Cette phase intervient à la fin du processus de conception ; il s'agit de vérifier par la simulation que le système prêt à partir en fonderie est conforme aux spécifications. Les différents modèles attachés à l'IP, correspondant à différents niveaux d'abstraction, prennent ici tout leur intérêt. En effet, il est souvent impossible de valider le bon fonctionnement d'un système à partir d'une description détaillée de tous ses blocs. D'autre part, soulignons que de nombreuses erreurs de conception proviennent de défauts de connexions dans la description du système à partir de ses blocs. Une approche intéressante consiste donc à procéder à des simulations « mixtes » pour lesquelles certains blocs sont modélisés par des descriptions de haut-niveau (connectique, fonctionnelle) et d'autres modélisés par des descriptions plus bas-niveau (comportementale, niveau transistor) car ils représentent des fonctionnalités plus critiques à l'échelle du système. Cela permet un compromis entre la durée de simulation et le « taux de couverture » de la vérification.



**Figure 14 : Résultats d'une simulation multi-niveaux d'abstractions d'un ASIC configuré pour 5 neurones**

A titre d'exemple, la fig.14 montre les résultats d'une simulation réalisée sur un ASIC complet configuré pour émuler 5 neurones de différents types [BOR06]. Le critère de validation est la présence d'oscillations présentant des potentiels d'action « neuro-réalistes » sur la tension de membrane de chaque neurone.

## **4. Conclusion**

Le présent chapitre propose la synthèse d'une expérience sur conception analogique centrée autour de la notion d'IP. Initiée par des réflexions menées dans le cadre d'une Action Spécifique du GDR SoC-SiP [AS], les points abordés concernent la définition d'un IP analogique et l'organisation du flot de conception système à partir d'une base de données d'IPs. Cette méthodologie a été appliquée à la conception d'ASICs neuromimétiques de l'équipe ISN.

Ces travaux ont donné lieu à 7 communications, 2 articles et un chapitre d'ouvrage. Ces publications concernent soit la problématique de la conception analogique [TLB07], [LTL07a], [LLT07b], [LLT08], [LLS08] à base d'IPs de circuits neuromimétiques, soit la question particulière du redimensionnement des circuits CMOS [LLT06], [LTL07b], [LLT07a], [LLT07c], [LTL09].

# ACQUISITION ET CONTROLE

## DE L'ACTIVITE BIOELECTRIQUE

Contexte, méthodes et applications

### **1. Contexte local**

Ce dernier chapitre expose mon projet de recherche. Il prend appui d'une part sur l'histoire et les compétences de l'équipe ISN (Ingénierie des Systèmes Neuromorphiques) du laboratoire IMS et d'autre part sur les spécificités de mes précédentes thématiques de recherche, résumées dans les chapitres 1 à 3 du présent mémoire.

L'équipe ISN dirigée par Sylvie Renaud développe depuis 1993 des circuits intégrés et systèmes d'instrumentation dont le comportement et l'architecture imitent ceux des réseaux de neurones biologiques. Cette activité a un cadre pluridisciplinaire et a été soutenue par de nombreux projets nationaux et internationaux faisant intervenir des biologistes, neuroscientifiques, physiciens, informaticiens et médecins [ISN].

Deux types d'expérimentations illustrent les applications de ces circuits et systèmes neuromorphiques. D'une part des simulations faites à partir de ces réseaux de neurones artificiels biologiquement réalistes, pour en analyser le comportement dynamique ; ils servent alors d'outils d'investigation pour les neurosciences computationnelles<sup>1,2</sup> [LBA06], [LER07], [LBR09]. D'autre part des expérimentations hybrides où neurones artificiels communiquent en temps réel avec des neurones vivants<sup>3</sup> [LRD02], [MSC04].

La thèse de T. Lévi a été une charnière qui m'a permis de rejoindre en 2005 l'équipe ISN, à l'occasion de la restructuration des groupes de recherche, liée au projet de création du laboratoire IMS. Au cours de cette thèse, le thème de la conception analogique à base d'IPs a

---

<sup>1</sup> Les neurosciences computationnelles sont un champ de recherche des neurosciences qui s'applique à découvrir les principes de computation sous-jacents aux fonctions cérébrales. Elles s'appuient sur des modèles s'appliquant à différents niveaux de description, de la molécule au comportement.

<sup>2</sup> Projets européen FACETS (2005-2009 ; FP6-IST-FETPI- 2004-15879).

<sup>3</sup> Projet européen NEUROBIT (2002-2005 IST-2001-33564).

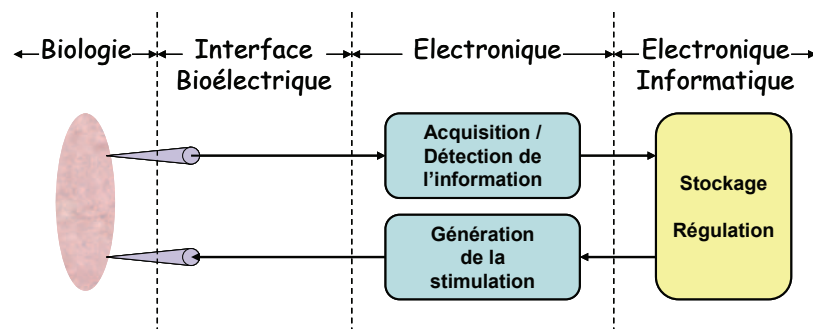
été appliqué à un ASIC neuromimétique de l'équipe ISN, et cela a fourni un cadre à ma conversion thématique<sup>4</sup>.

Depuis 2009, l'équipe ISN et l'équipe BioEM dirigée par B. Veyret se sont rassemblées autour d'un projet commun qui s'est concrétisé par la création du groupe de recherche Bioélectronique.

Au 1<sup>er</sup> janvier 2010, l'équipe ISN a prévu de se scinder en 2 équipes, correspondant aux 2 champs d'application décrits plus haut.

L'équipe AS2N<sup>5</sup> dirigée par S. Saïghi traitera de la conception de réseaux de neurones biologiquement réalistes sur silicium, en gardant le contexte applicatif des neurosciences computationnelles, afin de proposer de nouveaux paradigmes de calcul bio-inspirés.

Pour ma part je dirigerai l'équipe ELIBio<sup>6</sup>, organisée autour de l'interaction des systèmes électroniques avec le vivant, dans un but de compréhension des mécanismes biologiques ou de stimulation intelligente pour applications thérapeutiques. Cette équipe fédère plusieurs projets qui sont tous construits selon le même paradigme expérimental ; une description fonctionnelle en est faite ci-dessous.



**Figure 1 : Paradigme expérimental des projets d'ELIBio**

La majeure partie des projets est centrée autour de l'interaction avec des tissus neuronaux, mais nous verrons que le dispositif et les méthodes peuvent également s'appliquer à tout type de cellule vivante présentant une activité électrique et pouvant être excitée électriquement. Lorsqu'il s'agit d'observer et d'analyser le comportement des cellules vivantes soumises à des conditions expérimentales bien précises, la chaîne d'acquisition suffit. Mais dans la plupart des cas, il s'agit de stimuler en retour le tissu vivant de manière

<sup>4</sup> CRCT accordé sur la période 2007-2009.

<sup>5</sup> Architecture of Silicon Neural Networks.

<sup>6</sup> Systèmes ELectroniques en Interaction avec la BIOlogie.

« intelligente », c'est-à-dire en passant par un organe de régulation qui contrôle les paramètres de la stimulation en fonction de ceux de l'acquisition. C'est ce dernier point qui impose la contrainte de « temps réel » à nos systèmes. Il faut comprendre par là qu'entre l'instant où un échantillon de données est acquis et l'instant où il va produire une stimulation appropriée dans le vivant, il doit s'écouler un temps inférieur à celui qui sépare 2 échantillons successifs.

## **2. Contexte scientifique et problématique**

### **2.1. Introduction**

Le contexte scientifique général est celui des systèmes électroniques pour l'observation et le contrôle de l'activité électrique des tissus nerveux. Ces dispositifs permettent de progresser dans la compréhension des mécanismes neuro-physiologiques mais également de concevoir des solutions thérapeutiques sous la forme d'implants. On peut citer différents cas de neuro-prothèses aujourd'hui commercialisées et implantées chirurgicalement sur de nombreux patients. C'est le cas des prothèses cochléaires [GET02] utilisées pour restaurer les fonctions auditives mais également des implants de stimulation cérébrale profonde, utilisés depuis près de 20 ans en chirurgie pour traiter les symptômes de la maladie de Parkinson [BPG91], et plus récemment pour traiter d'autres affections comme les douleurs neuropathiques [CAG07] ou l'épilepsie [THF04]. Citons aussi les travaux aujourd'hui à l'étude pour proposer des prothèses rétinienne [WLH05].

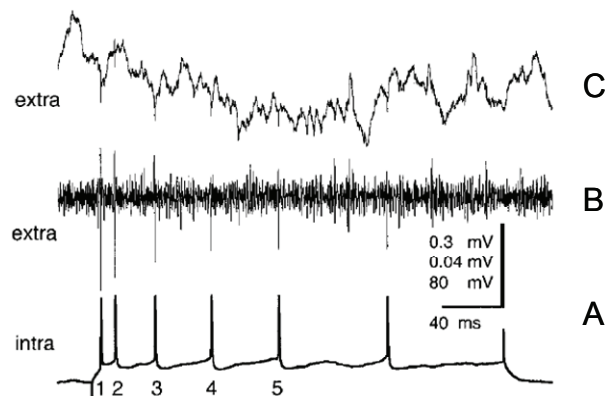
Soulignons enfin le vaste champ d'investigation des interfaces cerveau-machine ou BMI (Brain-Machine-Interface), où les signaux neuronaux sont traduits en signaux de commande pour contrôler des systèmes externes (bras robotisés, ordinateurs...) [CMM99], [NIC01], [TTS02], [VPS08].

Afin de bien cerner le contexte scientifique et le positionnement du projet de recherche exposé au paragraphe **3**, les parties suivantes sont consacrées tout d'abord à un résumé des propriétés du signal neuro-électrique (**2.2**) et aux principales configurations d'électrodes (**2.3**). Viendront ensuite des considérations sur les circuits de détection de l'information (**2.4**), la stimulation contrôlée (**2.5**) et le traitement en temps réel (**2.6**).

### **2.2. Le signal nerveux**

Les tissus nerveux sont des organisations complexes qui sont le siège du transport et du traitement des informations sensorielles ; ces informations sont intégrées pour produire une

réaction adaptée de l'organisme. Au cœur de cette chaîne de traitement se trouvent les neurones et les synapses. Le signal électrique est produit dans le soma d'un neurone, sous l'effet d'une différence de concentrations ioniques de part et d'autre de sa membrane. L'activité électrique d'un neurone se traduit alors sous la forme d'un potentiel de membrane  $V_{mem}$  oscillant et présentant des « pics » d'activité appelés *potentiels d'action* ou *spikes* en anglais (voir fig.2). L'information électrique contenue dans les spikes est ensuite véhiculée depuis le neurone considéré, jusqu'à une assemblée de neurones voisins par le biais de synapses. Les connections synaptiques déterminent l'architecture d'un réseau de neurones et sont soumises à des lois d'adaptation modifiant dynamiquement cette architecture.



**Figure 2 : Exemples de signaux neuroélectriques**

**A : enregistrement intracellulaire potentiel de membrane, B – C : enregistrements extracellulaires, B : signal complet, C : champs locaux**

Le type de signal A représenté fig.2 ne peut être observé que lors d'expériences d'électrophysiologie *intra-cellulaire*. Cette technique de mesure utilise en effet une microélectrode capable de pénétrer à l'intérieur d'un neurone. Les signaux B et C proviennent de mesures *extra-cellulaires* ; l'électrode est alors en contact avec les tissus nerveux et prélève, par couplage capacitif, l'activité électrique combinée d'un lot de neurones voisins.

L'amplitude typique du signal extra-cellulaire atteint la centaine de  $\mu\text{V}$ . On distingue deux composantes dans ce signal :

- une composante rapide sous la forme de pics, ce sont les spikes, dont la durée est de l'ordre de la ms,
- une composante lente dont l'oscillation est visible sur le signal C fig.2 ; cette composante est l'image de ce qu'on appelle les potentiels de champs locaux ou LFP (*Local Field Potential*), qui proviennent de l'activité électrique « moyenne » dans la région de l'électrode. La plupart du temps, cette composante apparaît comme un

bruit blanc mais lorsque les neurones de cette région ont une activité synchrone on observe des oscillations lentes.

Le signal **B** de la fig.2 comprend ces 2 composantes, spikes et LFP.

La technique de mesure *intra-cellulaire* délivre directement le potentiel de membrane d'un neurone, pouvant atteindre 100 mV. Les spikes, représentatifs de l'activité du neurone, sont alors facilement identifiables, ainsi que le neurone dont ils proviennent ; par contre, le placement et le maintien de l'électrode sont des tâches délicates. Avec la technique de mesure *extra-cellulaire*, le signal est davantage bruité car il combine l'activité de plusieurs neurones et l'activité de fond du milieu. La principale difficulté n'est donc plus d'ordre expérimental mais elle est reportée sur le traitement du signal mesuré, pour en extraire une information pertinente.

Les projets qui vont être décrits plus loin reposent sur l'acquisition du signal biologique extra-cellulaire.

### **2.3. Technologie de l'interface neuro-électrique**

Au cœur de ces systèmes hybrides se trouve un transducteur de l'activité électrique entre le milieu biologique et les circuits électroniques. Ce transducteur est une électrode ou un réseau d'électrodes conçus pour mesurer différents types de signaux. La mobilité entre les ions du milieu biologique est très inférieure à celle des électrons/trous du métal ou semi-conducteur, si bien que la bande passante de ces électrodes ne dépasse pas la dizaine de kHz. Le phénomène physique de transduction repose soit sur un transfert d'électrons<sup>7</sup>, soit sur un couplage capacitif, lorsque les variations de tensions sont trop faibles pour entraîner un transfert de charge significatif [PEG08]. Lorsque l'électrode est utilisée en enregistrement, c'est en général l'effet capacitif qui est mis en jeu et le signal est appliqué à l'entrée d'un amplificateur haute-impédance ; lorsqu'elle est utilisée en stimulation, l'électrode est plutôt le siège d'un transfert de charge par le biais d'un courant statique non-négligeable. Notons, que les modèles électriques de ces électrodes seront donc différents selon leur utilisation en enregistrement ou en stimulation : un modèle petit signal sera adapté à l'enregistrement tandis qu'un modèle grand signal pourra être nécessaire en stimulation.

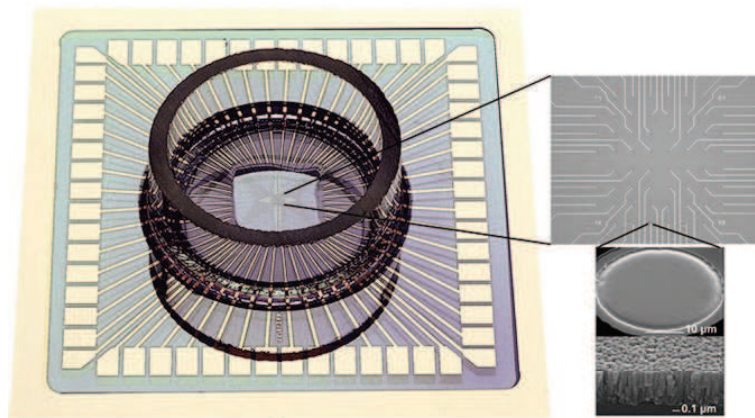
---

<sup>7</sup> Par oxydation/réduction.



La configuration des électrodes diffère selon le type d'expérience et l'application visée. On distingue ci-après les expériences *in-vitro* ou *in-vivo*.

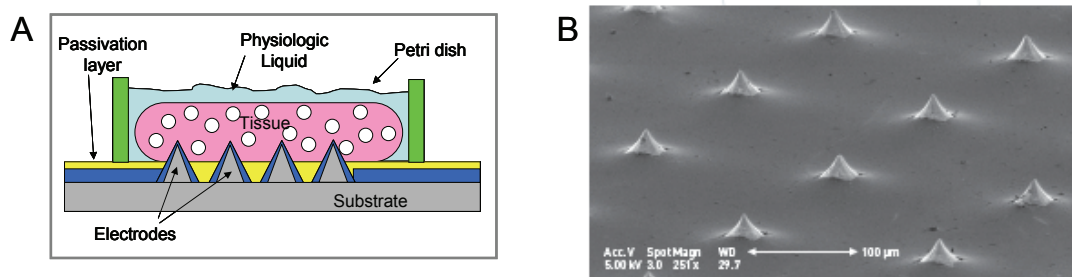
Dans le cadre d'expérimentations *in-vitro*, on utilise des MEAs (*Multi-Electrode Arrays*) introduits pour la première fois en 1972 [TSL72] et aujourd'hui commercialisés [MCS], [AYA]. Ce sont des dispositifs conçus à base de matériau biocompatible (verre, céramique, silicium) sur lequel sont déposés des sites d'enregistrement ou stimulation, reliés à des fils conducteurs et plots de connexion.



**Figure 3 : MEA commercialisé par Multichannel Systems [MCS]**

Les électrodes de ces matrices MEAs peuvent accueillir des cellules en culture ou en tranche, et permettent des expériences d'enregistrement ou de stimulation pendant plusieurs semaines voire plusieurs mois. Les cellules en question sont très souvent des neurones mais toute cellule active et excitable électriquement peut être observée par ce type d'instrumentation (cellules cardiaques ou pancréatiques par exemple [BRB10]).

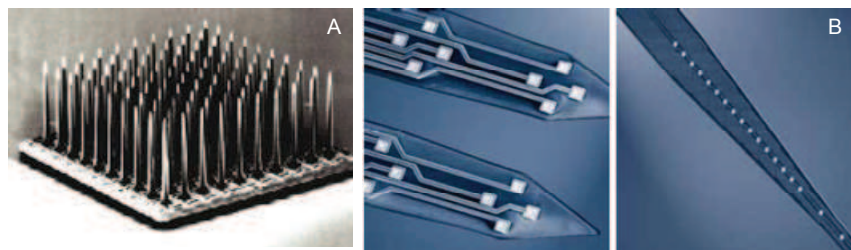
Le nombre de voies sur MEAs peut aller jusqu'à plusieurs centaines ; le diamètre d'un site d'enregistrement est de l'ordre de  $10\mu\text{m}$  et l'espace entre 2 sites de l'ordre de  $100\mu\text{m}$ . L'extrémité de l'électrode peut être planaire ou bien profilée en pointe (voir fig.4) ; le profil en pointe, plus pénétrant, est mieux adapté au cas des tissus en tranche aigüe (cerveau, moelle épinière, rétine) [HFR02].



**Figure 4 : MEA avec électrodes profilées en pointe [RPC07] (A), [AYA] (B).**

Ces MEAs sont utilisées dans différents types d'application. Un grand nombre d'applications relèvent de la recherche fondamentale, afin de comprendre les mécanismes biologiques liés à la dynamique des réseaux (évolution, auto-arrangement, mémorisation). Elles peuvent être utilisées en enregistrement [MAS02], [WPP06] ou en stimulation, dans le but d'induire des modifications plastiques dans les réseaux de neurones vivants [SHM01], [EBM03], [WMP05]. D'autres champs d'application utilisent des neurones en culture sur MEA pour contrôler différents types de processus. On peut citer les travaux de S. Potter dans le domaine de la robotique [DWB01] ou ceux de M. Chiappalone et S. Martonoia appliqués à la pharmacologie [CVT03].

Pour des expérimentations *in-vivo*, on utilise différents types d'électrodes pénétrantes, de la simple pointe métallique au réseau multi-pointes. Dans les années 80, le domaine de la fabrication de circuits intégrés a vu émerger les techniques de micro-fabrication. Ce contexte a promu le développement de réseaux d'électrodes sur substrat de silicium, lesquelles ont permis une grande avancée dans le domaine des neurosciences puisqu'elles ont ouvert la voie à l'enregistrement multi-voie de l'activité neuronale, dans des régions bien précises du cerveau [ODV95]. On peut citer par exemple le dispositif Utah Electrode Array, dont une photographie est donnée fig.5, aujourd'hui commercialisé par plusieurs sociétés [AOE]. Il s'agit d'un composant à base de silicium, comportant un réseau de 10x10 électrodes, dont le diamètre est de 80  $\mu\text{m}$  à la base, d'une longueur de 1,5 mm avec une distance entre électrodes de 400  $\mu\text{m}$ . Il existe d'autres profils d'électrodes implantables selon les applications (voir fig.5) et les zones du cerveau à atteindre.



**Figure 5 : Electrodes implantables microfabriquées**  
A – Utah Electrode Array [AOE] ; B – NeuroNexus probes [NEU]

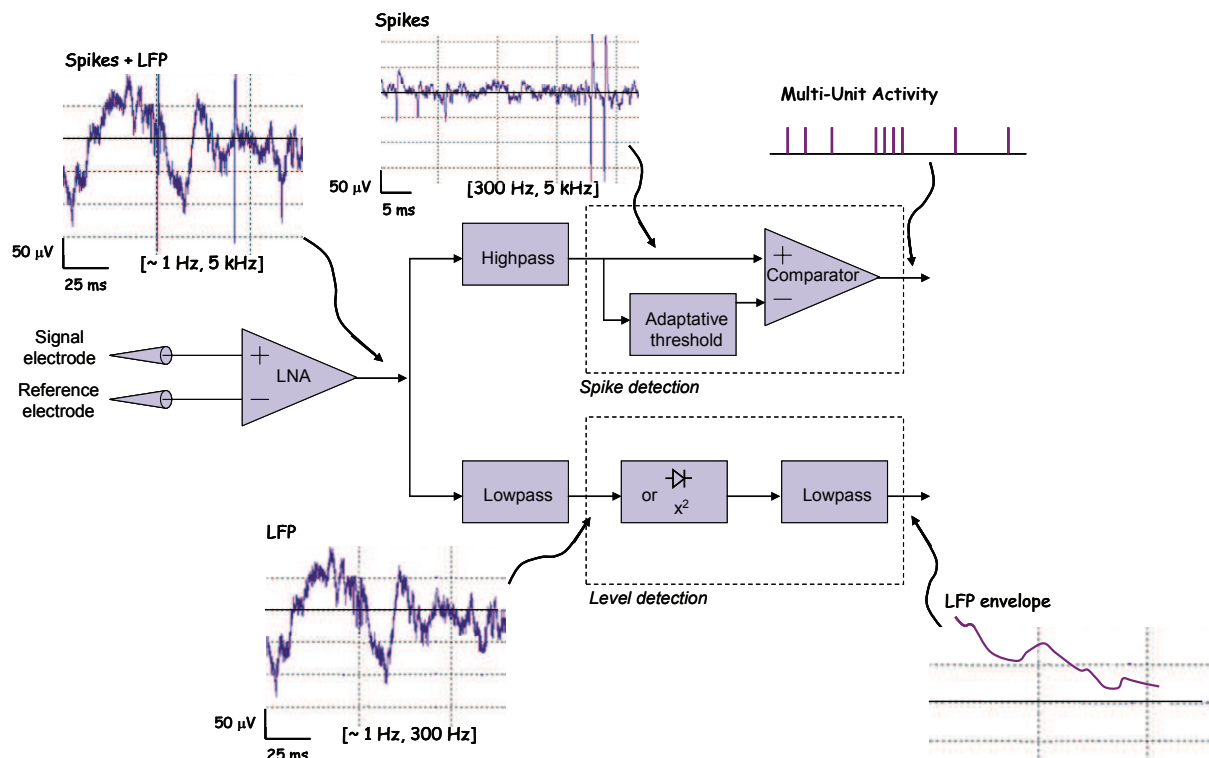
#### **2.4. Circuits de détection de l'information**

Nous nous plaçons dans le cas de l'acquisition de signal *extra-cellulaire*, que ce soit par le biais d'une simple électrode ou d'un réseau d'électrodes. Ce nombre de voies d'entrée ne

va pas affecter le type de traitement nécessaire pour extraire une information pertinente, mais il aura un impact sur les performances globales du traitement, en particulier sur la possibilité d'opérer en *temps réel* pour un traitement *on-line* ou une éventuelle rétroaction. On reviendra sur ces aspects dans le paragraphe 2.6.

La possibilité d'enregistrer l'activité électrique simultanée de plusieurs neurones grâce aux interfaces décrites précédemment a suscité un déploiement de la recherche dans le domaine des circuits intégrés dédiés au traitement des signaux neuronaux [PEG06], [BTR07], [HAR08], [PEG08], [JDW09].

La fig.6 donne un aperçu des fonctions de traitement amont du signal neuroélectrique typiquement enregistré sur une voie. Passons en revue chacun de ces blocs et précisons les besoins de conception.



**Figure 6 : Vue d'ensemble des circuits de détection amont du signal neuroélectrique**

### Amplification faible bruit

Cet étage est nécessaire à cause de la faible amplitude des signaux enregistrés en extracellulaire (quelques dizaines de  $\mu$ V). Voici les performances que l'on cherchera typiquement à atteindre [HAR08], [BTR07], [HAC03] :

- bruit ramené en entrée inférieur la dizaine de  $\mu$ V
- dynamique d'entrée suffisante pour permettre le passage d'amplitudes de spikes allant jusqu'à quelques mV

- impédance d'entrée très grande devant l'impédance de l'électrode, pour minimiser l'atténuation du signal
- bande passante adaptée au type de signal à analyser (LFP ou spikes)
- minimisation de l'offset pour ne pas saturer l'amplificateur
- minimisation de la surface et de la consommation.

Ce dernier point est particulièrement crucial pour le développement de systèmes implantés [JDW09].

### Filtrage Passe-Haut ou Passe-Bas

L'amplificateur faible bruit permet d'extraire un signal présentant 2 composantes que l'on traite indépendamment : les LFP et les spikes. Un filtrage approprié avec une frontière de quelques 100 Hz isole facilement les 2 composantes (voir fig.6).

### Détection des spikes

Le signal neuro-électrique une fois débarrassé de ses composantes basses fréquences reflète ce qu'on appelle l'activité neuronale multi-unitaire (Multi-Unit Activity, MUA) ; l'information pertinente de ce signal est contenue dans les spikes. La détection des spikes peut se faire par une technique de seuillage ou bien par reconnaissance de forme. La 1<sup>ère</sup> solution peut être intégrée sur un ASIC de traitement amont, comme cela est représenté fig.6. La technique de reconnaissance de forme met en œuvre des algorithmes plus complexes et intégrés sur support numérique (logiciel ou matériel) après conversion analogique/numérique [LEW98]. Nous ne parlerons ici que de la 1<sup>ère</sup> solution.

Comme cela est visible sur la fig.6, les spikes sont généralement des pics dont l'amplitude dépasse le bruit de fond et une technique de détection par seuil paraît alors indiquée. L'amplitude des spikes enregistrés sur les électrodes d'un MEA peut varier considérablement d'une électrode à l'autre ; en effet, cela dépend de la position du site d'enregistrement par rapport au neurone responsable de l'impulsion, mais aussi du bruit ambiant (synaptique et électronique). Le seuil de détection a donc besoin d'être réglé au cours d'une expérience. Cela peut se faire manuellement, au moyen d'un signal externe envoyé sur l'entrée de référence du comparateur, mais une solution de réglage autonome du seuil prend tout son intérêt dans le traitement de nombreuses voies d'acquisition. Un circuit adaptatif est par exemple décrit dans [HAR08].

### Classification de spikes

Cette opération consiste à faire correspondre à chaque spike le neurone qui l'a produit. Cela ne peut pas se faire à partir d'un signal d'activité multi-unitaire détectée par seuillage, car toutes les impulsions se ressemblent. Cela nécessite de trier les spikes sur la base de facteurs de forme, en partant du principe qu'à un neurone correspond une forme de spike. Plusieurs méthodes sont alors possibles comme l'analyse de la dérivée, l'analyse en composante principale ou la transformée en ondelettes [LEW98]. Ce type de traitement pose la question du compromis précision/temps, car plus l'algorithme de tri est efficace plus il est coûteux en temps de calcul. Cette question est particulièrement critique pour les applications où le signal doit être traité en ligne et davantage encore pour les systèmes bouclés en temps réel. Dans les équipements commerciaux, ce type de traitement est généralement fait par un logiciel vendu avec les microélectrodes [AYA]. Certains travaux d'actualité visent une implémentation sur circuit numérique [ESC08], [LBB09], ou sur circuit analogique [GOS08].

### Détection du niveau des LFP

Les LFP résultent en un signal très basse fréquence, témoignant d'une activité d'ensemble, dont l'origine biologique est complexe<sup>8</sup>. On peut noter la corrélation du signal LFP avec le signal issu d'un électroencéphalogramme (EEG). C'est l'activité rythmique (oscillatoire) du signal LFP et son rôle dans le traitement des informations sensori-motrices qui intéresse les chercheurs en neurosciences. Par exemple, il a été démontré que le signal LFP joue un rôle dans le contrôle volontaire des mouvements [SJA05], ce qui ouvre des perspectives dans le domaine des interfaces cerveau-machine ou encore dans l'étude de la physiologie et le traitement de la maladie de Parkinson [BWK94], [LAH02].

L'information pertinente dans le signal LFP est donc le « niveau » d'une activité oscillatoire dans une bande étroite de fréquence. Après avoir procédé au filtrage adéquat, cette information peut être extraite par un circuit détecteur de crête ou encore un circuit détecteur d'énergie tel que celui proposé par [HSS04].

Les principales fonctions d'acquisition et de traitement du signal neuro-électrique viennent d'être décrites. La frontière entre le traitement analogique et le traitement numérique est mobile. L'étage d'amplification faible bruit est souvent placé au plus près des électrodes,

---

<sup>8</sup> Il résulterait de la combinaison des potentiels synaptiques, des repolarisations du soma et de dendrites, et des activités oscillatoires de membranes des neurones.

afin de ne pas détériorer le rapport signal à bruit. La plupart des équipements commerciaux placent le convertisseur analogique/numérique juste après, le reste des fonctions étant implémenté sur logiciel. Aujourd'hui, que ce soit pour des applications de systèmes implantés *in vivo* ou pour garantir un traitement *on line* des données mesurées *in vitro*, des efforts sont faits pour implémenter ces fonctions sur circuit intégré, le cas extrême étant l'intégration combinée de l'électronique et des électrodes, grâce aux technologies MEMS/CMOS [HWK07].

## 2.5. Stimulation contrôlée

La stimulation électrique de tissus excitables comme les muscles ou les neurones est probablement la fonction prééminente dans les dispositifs biomédicaux implantés aujourd'hui, que ce soit pour rétablir une activité cardiaque rythmique (pacemaker), pour compenser un dysfonctionnement du système nerveux central (stimulation cérébrale profonde) ou pour transmettre une information sensorielle artificielle se substituant à celle d'un organe récepteur endommagé (prothèses cochléaires ou visuelles).

Si l'on se penche sur les applications à l'interface des tissus nerveux, la conception du circuit de stimulation doit être guidée par les questions suivantes, car l'efficacité de la stimulation en dépend [GHO08] :

- Quel type d'électrode ?
  - Monopolaire, bipolaire ou multi-polaire
- Quel comportement électrique de l'interface électrode/tissu ?
  - Modélisation de l'impédance [PEG08], [WEG09]
- Quelle forme pour le signal de stimulation ?
  - Impulsionnelle, à décharge exponentielle, sinusoïdale
  - Mono- ou multi-phasique, équilibrée en charge
  - Paramètres de contrôle possibles : amplitude, durée, fréquence, instant de déclenchement.

Enfin, un autre phénomène est à prendre en compte est l'artefact de stimulation. En effet, l'artefact de stimulation est un signal parasite observé au niveau l'acquisition et provenant d'un couplage avec l'électrode de stimulation. Plusieurs stratégies sont possibles pour éliminer cet artefact : on peut citer la solution de compensation par un circuit électronique [NBR08] ou celle basée sur une optimisation géométrique de la forme des électrodes.

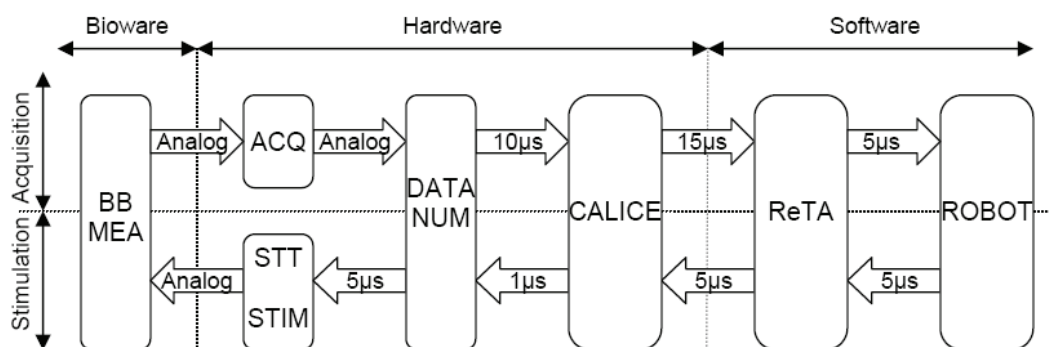
Enfin, n'oublions pas que dans nos applications la stimulation devra être contrôlée par les paramètres caractéristiques du signal d'acquisition. Il sera donc nécessaire de choisir une méthode de modulation adaptée.

## 2.6. Traitement temps réel

Dans nos applications, nous cherchons à effectuer une acquisition de signal bio-électrique et, en retour, une stimulation contrôlée du tissu vivant. Comme il a été expliqué précédemment, cela nécessite un traitement de boucle en temps réel. Il importe donc d'estimer avec précision les délais introduits par chaque étage de traitement.

Un prétraitement réalisé en analogique est favorable à cette contrainte car il n'introduit pas de délai. La bande passante du signal neuro-électrique extracellulaire étant de l'ordre de 5 kHz. La fréquence d'échantillonnage suffisante pour les analyses expérimentales (neuroscientifiques ou cliniques) va de 10 à 30 kHz. Le traitement numérique arrive dans la chaîne pour des raisons de souplesse de calcul mais des délais apparaissent alors, qu'ils soient liés au protocole de transmission des données ou aux algorithmes de calcul.

Il est évident que le nombre d'électrodes accroît les contraintes de débit de données. Notons que de nombreux travaux concernent la transmission *sans fil* des données enregistrées par des systèmes implantés in vivo, et qu'étant donné les bandes de fréquences allouées aux applications bio-médicales, cela nécessite l'utilisation d'algorithmes de compression de données [HAR08].



**Figure 7 : Délais de propagation du système hybride HyNNet [BRG07]**

La fig.7 reprend l'architecture d'un système bouclé sur MEA 60 électrodes et destiné au pilotage d'un composant artificiel de type robot. Les éléments de traitement du système sont distribués sur support matériel, analogique ou numérique et logiciel. En additionnant les délais des modules numériques on obtient une durée de 46  $\mu$ s, alors que la période d'échantillonnage est de 100  $\mu$ s. Ainsi, une durée de 64  $\mu$ s est encore disponible pour insérer dans la boucle un robot ou toute autre unité de calcul [BRG07].



En ce qui concerne le protocole de communication, le débit des données peut être réduit en utilisant des protocoles asynchrones ou évènementiels déjà appliqués dans le cadre de l'ingénierie neuromorphique [BOA00], [VMC06], [BTM09].

### 3. Projets et stratégie

Cette partie propose une revue des projets pluridisciplinaires qui sont à la base de la création de l'équipe EIBio. Je commencerai par énoncer brièvement le contexte et les objectifs de chaque projet et finirai par l'exposé de la méthode de travail préconisée et de son cadre stratégique. Je rappelle ici le principe expérimental sur lequel reposent tous les projets décrits ci-après :

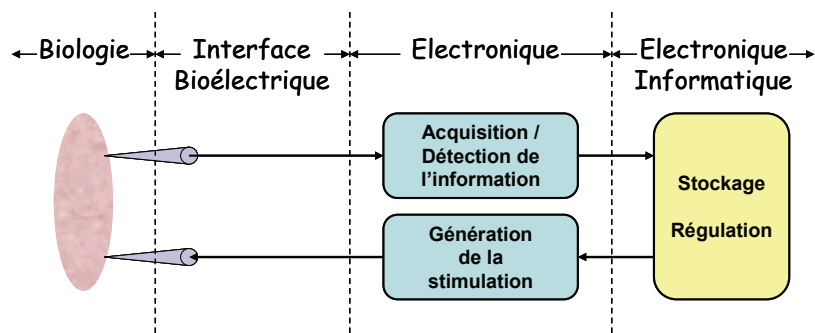


Figure 8 : Paradigme expérimental des projets d'EIBio

#### 3.1. Projet STN Oscillations

Il s'agit d'un projet financé par l'ANR sur l'appel d'offre « Maladies neurologiques et maladies psychiatriques », sur la période 2009 – 2013.

Les partenaires impliqués sont les laboratoires MAC (Mouvement, Adaptation, Cognition, Université Bordeaux 2) et IMS (Intégration du Matériau au Système, Université Bordeaux 1).

Le noyau sous-thalamique (*Sub-Thalamic Nucleus* ou *STN*, en anglais) est une région profonde du cerveau, dont l'activité oscillatoire est en relation avec le contrôle volontaire du mouvement. Par ailleurs, l'enregistrement des LFP sur des patients atteints de la maladie de Parkinson a révélé la présence d'oscillations dans le STN sur les bandes de fréquences beta (15-30 Hz) et theta (30-80 Hz). Ces résultats laissent à penser qu'il y a un lien direct entre l'activité des neurones du STN et la manifestation des symptômes moteurs liés à la maladie de Parkinson, sans qu'une démonstration en ait été fournie à l'heure actuelle. C'est pourquoi



l'objectif du projet **STN Oscillations** est d'étudier le rôle et le contrôle des oscillations du noyau sous-thalamique dans la physiopathologie de la maladie de Parkinson, afin d'en déduire une nouvelle stratégie de stimulation compensatoire et thérapeutique.

En premier lieu, il s'agit d'analyser les oscillations LFP enregistrées sur un animal (rat) sain ou bien rendu malade par injection de MPTP<sup>9</sup>.

En deuxième lieu, on étudiera l'influence d'une stimulation électrique construite à l'image des LFP « pathologiques » sur l'animal sain ou malade.

En conclusion, on apportera un 1<sup>er</sup> point de vue sur la possibilité de produire une stimulation contrôlée par les caractéristiques des LFP enregistrés, en proposant un processus de régulation adapté. A terme, le développement d'un système intégré d'acquisition et de contrôle des LFP, auto-adaptatif, fonctionnant en temps réel et in-vivo est envisagé.

Au cours de ce projet, l'IMS doit spécifier et concevoir un circuit de détection multi-bande de l'amplitude des LFP, un circuit de stimulation configurable et proposer une architecture de contre-réaction capable de compenser les oscillations « pathologiques » prélevées dans le signal LFP. Ces circuits seront conçus en électronique analogique (pour favoriser le traitement en temps réel), avec des contraintes de surface et de consommation réduites (pour favoriser l'implantation ultérieure d'un dispositif in-vivo).

Actuellement A. Zbrzeski est affectée à ce projet dans le cadre de son doctorat (allocation MESR), sous la co-direction de S. Renaud et moi-même.

Ce projet rentre dans les priorités du projet « Neurocampus » soutenu par la Région Aquitaine et dans lequel nos partenaires du MAC, A. Benazzouz et T. Boraud, sont fortement impliqués [HEI09].

### **3.2. Projet HYRENE**

Il s'agit d'un projet soumis à l'ANR sur l'appel d'offre « Programmes Blancs – Thématique STIC/DEFIS », et demandé sur la période 2010 – 2014. HYRENE est l'acronyme de HYbridation de REseaux de Neurones.

---

<sup>9</sup> Substance chimique utilisée en laboratoire pour simuler la maladie de Parkinson en vue d'en étudier la physiologie et d'expérimenter des traitements potentiels.

L'IMS apparaît comme porteur de ce projet, en partenariat avec l'ESIEE-Paris (Equipe ESYCOM, Equipe Systèmes de Communication et Microsystèmes) et le CNIC (Centre de Neurosciences Intégratives et Cognitives, Université Bordeaux 1).

Le projet HYRENE est un projet de recherche fondamentale visant à développer de nouvelles technologies de communication hybride entre réseaux de neurones vivants et réseaux de neurones artificiels électroniques. L'ambition particulière de ce projet est de parvenir à coupler un organe entier (moelle épinière de souris *in vitro*) à un réseau artificiel, avec pour objectif de rétablir une activité fonctionnelle de l'organe lorsque celui-ci a subi une lésion. A plus long terme, les concepts mis en place dans le cadre de ce projet permettront de développer des interfaces neuro-électroniques « intelligentes » à des fins de réhabilitation fonctionnelle.

Comme tout système bouclé, l'architecture envisagée comprend une partie acquisition et une partie rétroaction, pilotée par un processus de régulation qui garantit au système son autonomie de fonctionnement. La grande originalité de ce projet est que l'organe de régulation est un réseau de neurone artificiel (RNA), intrinsèquement auto-adaptatif puisqu'il est implémenté avec des règles de plasticité synaptiques [BTM08]. Il s'agit d'une proposition entièrement novatrice car elle vise non seulement la réalisation d'un système hybride vivant/électronique bouclé en temps réel, mais aussi car elle inclut dans la boucle un calculateur neuromorphique comme élément de contrôle. Ce projet est transversal au sein du groupe Bioélectronique, parce qu'il s'appuie sur les compétences des équipes ELIBio et AS2N.

Notre partenaire ESIEE a développé des technologies spécifiques de micro-fabrication de matrices de microélectrodes planaires ou tridimensionnelles et sera en charge de concevoir un réseau d'électrodes dont le profil est optimisé pour la stimulation. Le CNIC sera en charge des préparations physiologiques du tissu vivant mais a aussi un rôle important dans la définition des protocoles expérimentaux de validation du système hybride et de tout ce qui concerne l'interprétation des résultats sur les propriétés du réseau de neurones vivant. Le rôle de l'IMS est multiple ; il faudra en effet concevoir les circuits de détection de l'information neuro-électrique (tels qu'ils sont décrits fig.6) ainsi que le générateur de stimulation (qui sera commandé par les sorties du RNA), concevoir les interfaces de communications entre le RNA et les circuits de mise en forme du signal, régler la configuration et la plasticité du RNA pour qu'il joue son rôle dans le contrôle du système bouclé. L'intégration de l'ensemble des

briques de ce système hétérogène et la mise en œuvre de la communication entre ces briques correspond à un thème majeur de l'équipe EIBio.

Soulignons 2 types de retombées attendues de ce projet :

- la possibilité de concevoir un système hybride vivant/électronique à l'échelle d'un réseau entier, dans un contexte de la réhabilitation,
- la démonstration des atouts du calcul neuromorphique sur circuit intégré analogique et mixte pour concevoir de nouveaux paradigmes de traitement du signal.

En se plaçant sous un autre angle, l'idée que la réparation d'un organisme vivant lésé puisse se faire au moyen d'un circuit bio-inspiré apparaît très séduisante ...

### **3.3. Projet NeuroEM**

Il s'agit d'une action initiée à l'occasion d'un financement PEPS obtenu par S. Saighi et B. Veyret pour poser les bases d'une instrumentation permettant d'exposer un réseau de neurones in-vitro à des signaux de téléphonie mobile de type GSM. Ce thème a une importance particulière dans le groupe Bioélectronique par son caractère fédérateur et transversal, car il réunit aujourd'hui les compétences de toutes les équipes du groupe.

De nombreux travaux concernant les effets des ondes RF sur la santé ont conclu à l'absence d'effet biologique à des niveaux d'exposition conformes aux normes. Le seul mécanisme connu des RF est l'effet thermique sur les tissus, mais des résultats expérimentaux montrent que l'interaction avec l'activité électrique des réseaux de neurones constitue une hypothèse complémentaire.

Ce contexte décline une nouvelle fois le paradigme expérimental de la fig.8. Des cultures de neurones de cortex de rat sont réalisées sur MEA et sont actives pendant plusieurs semaines. Le dispositif développé dans le cadre du PEPS permet d'exposer le MEA à des signaux GSM 1,8 GHz et d'acquérir l'activité électrique du réseau de neurones exposé.

A partir de ce dispositif expérimental, il s'agit aujourd'hui de :

- traiter le signal issu des neurones,
- étudier la corrélation entre l'exposition aux RF et les modifications électrophysiologiques,
- conclure sur l'existence, la nature et les mécanismes des effets observés, en particulier le rôle de la modulation du signal RF.

Ce travail fait l'objet d'un sujet de thèse pour lequel une allocation MESR a été demandée pour la rentrée 2010. Un expert en neurosciences interviendra dans ce projet, il s'agit d'A. Garenne (laboratoire MAC, Université Bordeaux 2). Plus généralement, les supports financiers envisagés pour le projet **NeuroEM** seront la Fondation Santé et Radiofréquences et le nouvel appel « communications sans fil et santé » du FP7. Enfin soulignons que ce thème de recherche est une recommandation prioritaire de l'OMS.

### **3.4. Stratégie**

Ce paragraphe concerne la façon dont vont être abordés les projets précédemment décrits et plus généralement les projets de l'équipe E $\ell$ IBio. La méthode de travail est le fruit d'une convergence de mes thématiques de recherche antérieures et des besoins liés aux domaines d'applications d'E $\ell$ IBio.

#### **3.4.1 Méthodologie de conception des systèmes hybrides vivant/électronique**

Il a déjà été démontré que tous les projets rentrent dans un même schéma d'expérimentation, tel qu'il est illustré fig.8. Les points clés de cette méthodologie seront :

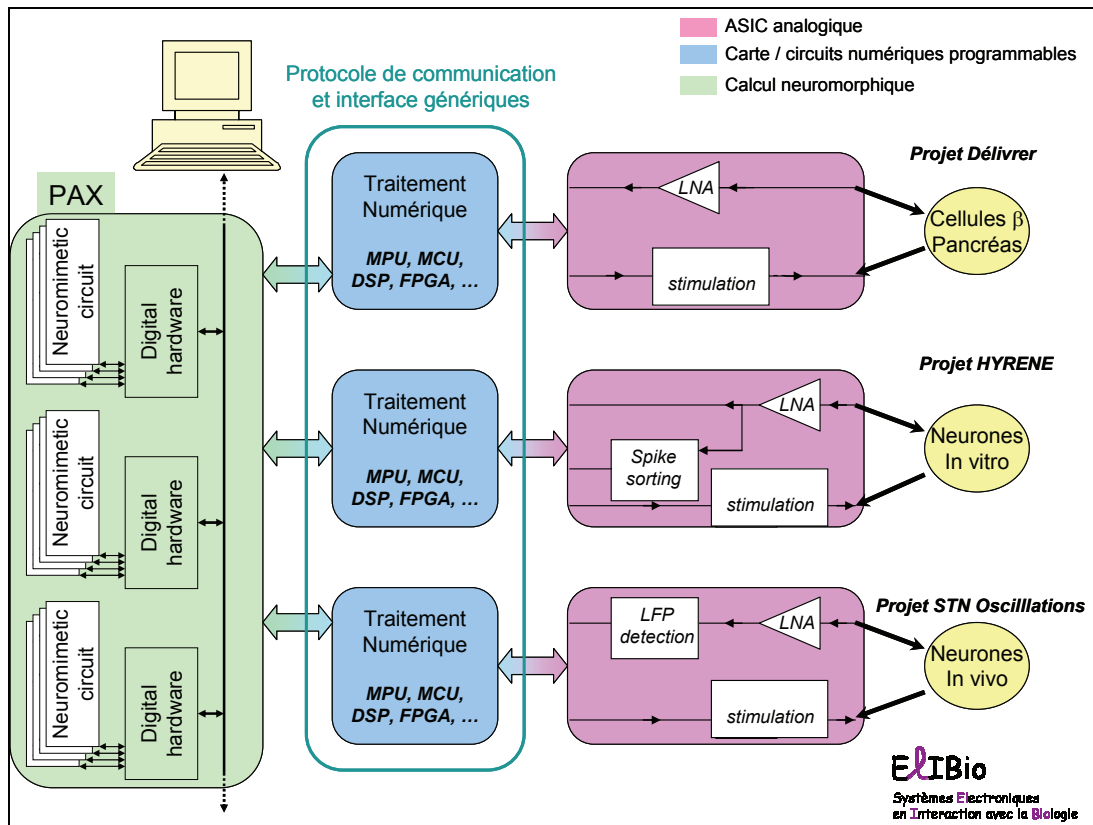
- 1) la progression vers un banc expérimental générique et modulaire,
- 2) la prise en compte de l'hétérogénéité des systèmes dès la conception,
- 3) l'utilisation de composants programmables analogiques FPAA<sup>10</sup> pour la réalisation des circuits de détection et de stimulation,
- 4) l'utilisation des circuits neuromorphiques en tant que processus de contrôle dans la boucle.

Les points 1) et 4) sont illustrés sur la figure suivante et un projet de fin d'étude d'Ingénieur CNAM amorcera le travail autour du banc d'instrumentation générique. Sur la fig.9, ce banc multi-application est décliné pour 3 projets actuellement en cours dans l'équipe et construits autour de l'interface avec des neurones in vitro (projet HYRENE), in vivo (projet STN Oscillations) ou des cellules pancréatiques (projet DELIVReR<sup>11</sup>).

---

<sup>10</sup> Field Programmable Analog Array.

<sup>11</sup> DELIVReR : Délivrance d'Insuline in Vivo Régulée et Rétrocontrôlée – Projet Région Aquitaine / FEDER, 2007-2010.



**Figure 9 : Synoptique du banc d'instrumentation générique pour le développement de systèmes hybrides vivant/électronique**

Le système PAX est un réseau de neurones biologiquement réaliste développé par l'équipe AS2N. Les fonctions qui sont à l'interface de la biologie sont réalisées sur circuits intégrés analogiques par des étudiants actuellement en doctorat dans l'équipe EIBio (G. Bontorin, A. Zbreski, A. Quotb).

Le travail demandé consiste à concevoir une interface et un protocole de communication entre la partie analogique et le réseau de neurones PAX. Différents supports pourront être envisagés pour cette tâche : microcontrôleur, circuit spécialisé type DSP ou circuit programmable FPGA, offrant ainsi des niveaux de programmation plus ou moins optimisés et adaptés à la maturité du projet considéré.

Le point 2) se focalisera sur la modélisation électrique de l'interface tissu/électrode en collaboration avec nos collègues spécialistes des technologies micro-système de l'ESIEE.

Le point 3) rentre dans le cadre d'une collaboration avec le Pr P. Hasler de Georgia Tech, Atlanta. P. Hasler est reconnu pour son expertise dans le domaine des circuits VLSI et porteur d'une technologie de conception très originale basée sur les transistors à grille

flottante et menant à des composants analogiques programmables appelés FPAA [CAS07]. Cette collaboration permettra d'une part de mettre en lien la conception analogique à base d'IPs initiée dans la thèse de T. Lévi et les circuits FPAA, mais aussi de réaliser des prototypes configurables de nos circuits interfacés avec le vivant. A. Zbrzeski a récemment fait un séjour à Atlanta, au laboratoire de P. Hasler et lui-même a le projet de venir en mai 2010 pour un séjour en tant que professeur invité à l'Université Bordeaux 1.

### 3.4.2 Pluridisciplinarité et ancrage dans les thématiques de recherche reconnues

La pluridisciplinarité est au cœur des projets d'ElIBio et la collaboration avec des chercheurs d'autres disciplines est fondamentale. Nous aurons à cœur de pérenniser ces collaborations.

#### Collaborations et ancrage régionaux :

- Interaction ondes/vivant et systèmes d'exposition : équipe de BioEM de B. Veyret, IMS, Université Bordeaux 1
- Neurophysiologie et Neurosciences :
  - Parkinson : A. Bennazouz et T. Boraud, MAC, Université Bordeaux 2
  - Acquisition MEA et interprétation : A. Garenne, MAC, Université Bordeaux 2
  - Dynamique des réseaux de neurones : B. Yvert CNIC, Université Bordeaux 1
- Chimie :
  - Bio-molécules, pharmacologie : V. Ravaine ISM - site ENSCPB, Université Bordeaux 1
- Biologie :
  - Cellules pancréatiques : J. Lang IECB, Université Bordeaux 1
- Médecine :
  - Diabète, pompes à insuline : B. Catargi Service de Diabétologie, CHU Bordeaux
- Cadre du projet « Neurocampus », Région/Université Bordeaux
- Cadre du pôle d'excellence *Neurosciences* de l'Opération Campus du PRES Université Bordeaux.

#### Collaborations et ancrage nationaux :

Le GDR SoC-SiP animé par M. Renovell constitue le principal cadre national des collaborations, en particulier ses thématiques *Circuits AMS et RF* (P. Desgreys Telecom-Paris, Y. Deval IMS) et *Systèmes Hétérogènes* (P. Garda LIP6, I. O'Connor LEOM).

Les appels à projets ANR suivis dont ceux de la thématique STIC/DEFI, car nos projets sont finalement en rupture avec les lignes technologiques identifiées, mais également ceux qui relèvent des Technologies pour la Santé et des Maladies Neuro-dégénératives.

#### Collaborations et ancrage internationaux :

Dans la continuité de l'ancrage de l'équipe ISN dans les projets européens, (projets Sensemaker, Neurobit, et FACETS), l'équipe prêtera une attention particulière à l'appel à projets du FP7 Neuro-Bio-ICT attendu pour 2010.

Hors Union Européenne, citons la collaboration avec P. Hasler de Georgia Tech, Atlanta, dont le cadre est expliqué plus haut.

#### **3.4.3 Transfert et politique de valorisation**

La plupart de nos projets sont associés à des retombées à plus ou moins long terme dans le domaine de la santé publique. Certains pourraient aboutir à des solutions thérapeutiques nouvelles voire des circuits intégrés implantables.

Un dossier de demande de brevet dans le domaine des Biotechnologies a été déposé récemment ; issu du projet DELIVRER, il porte sur la mesure continue, directe et à long terme, de l'activité électrique de cellules beta-pancréatiques, avec traitement et extraction sur circuit intégré. Cette invention pourrait déboucher à moyen terme sur la réalisation d'un système intégré et embarqué de mesure des besoins en insuline. Dans la continuité de ce projet, notre équipe déposera, avec nos partenaires de l'IECB et le soutien de Aquitaine Valo, un projet ANR dans l'appel d'offres Emergence-TEC en mars 2010 ; il s'agit d'un programme ANR spécifiquement orienté vers les projets de technologies pour la santé à fort potentiel de valorisation, pour lesquels les premières phases de validation des hypothèses ont déjà été réalisées.

Une telle politique de valorisation sera menée avec un intérêt constant au sein de l'équipe EIBio. Nous l'envisageons également pour le développement du stimulateur "intelligent" pour le traitement symptomatique de la maladie de Parkinson, issu de notre collaboration avec le laboratoire MAC, dans le cadre du projet STN Oscillations.

## **4. Conclusion**

Ainsi s'achève l'exposé du contexte et du contenu de mon projet de recherche. Les mots-clés qui le caractérisent sont :

- projet scientifique pluridisciplinaire,
- système bouclé sur interface vivant/électronique,
- conception de systèmes hétérogènes,
- circuits intégrés de traitement du signal bio-électrique,
- le calcul neuromorphique comme intelligence des systèmes,
- compréhension des mécanismes neuro-physiologiques,
- innovation thérapeutique.

A l'échelle du laboratoire IMS, ce projet se concrétise par la création de l'équipe E $\ell$ IBio qui signifie « Systèmes Electronique en Interaction avec la Biologie » et qui compte aujourd'hui :

- 3 chercheurs permanents :  
S. Renaud Pr. ENSEIRB, Y. Bornat MdC ENSEIRB et moi-même
- 1 ingénieur de recherche à 50% : G. N'Kaoua IR CNRS
- 1 ingénieur contractuel : Y. Boutaib, CDD projet STN Oscillations
- 3 doctorants : G. Bontorin, A. Zbrzeski et A. Quotb.

Cette équipe, dont j'assure la direction, aura à cœur de conforter ses collaborations pluridisciplinaires, en commençant par celles qui sont ancrées dans le contexte régional autour du pôle d'excellence Neurosciences. Soulignons les retombées de nos travaux exploratoires dans le contexte sociétal de santé publique, ainsi que l'intérêt des systèmes interfacés avec le vivant pour mettre en œuvre des problématiques liées à l'intégration des systèmes hétérogènes.





## CONCLUSION

Ce mémoire d'Habilitation à Diriger des Recherches relate un épisode de 16 années de mes activités de recherche à l'Université Bordeaux 1, au sein du laboratoire IMS. Une période de 16 ans à l'échelle de l'industrie du semi-conducteur et de la loi de Moore, cela correspond à une densité d'intégration multipliée par un facteur 65000. Même si on observe aujourd'hui un ralentissement de cette croissance exponentielle, en 16 ans les finesses de gravure des transistors sont passées du micron à la dizaine de nanomètres ; en 1993 Intel sortait son premier Pentium avec 3 millions de transistors, alors que ses processeurs en comptent près d'1 milliard aujourd'hui. En 16 ans, la miniaturisation des puces et la maîtrise des procédés technologiques a permis une réduction régulière du coût lié à fabrication des circuits intégrés. Par contre, la tâche de conception de puces fonctionnelles devient très complexe et provoque finalement un retard de mise sur le marché (*Time-To-Market*) qui a un coût économique important.

A quoi tient la complexité de la conception ? Elle n'est pas uniquement liée au nombre de transistors sur une puce ; par exemple, l'effort de conception pour passer d'une mémoire 256 kOctets à une mémoire 256 MOctets n'est pas énorme, en raison de la répétitivité des blocs, en tout cas il n'est pas multiplié par 1000. En réalité, les principales causes de la complexité de la conception sont plutôt :

- la co-conception des parties numériques et analogiques des circuits mixtes,
- le nombre croissant d'ingénieurs impliqués dans la conception d'un système sur puce, en raison de la nature très variée des composants à assembler,
- la rapide évolution des technologies de fabrication qui oblige les concepteurs à ajuster ou reconcevoir leurs bibliothèques de circuits, pour faire face à la variation des paramètres technologiques.

L'objectif des mes travaux de recherche a été de proposer des solutions pour contribuer à résoudre ces difficultés.

La question des modèles et de leur génération est fondamentale. Il est apparu que la **modélisation comportementale** était une première solution pour gérer la complexité de la conception analogique, surtout si elle s'insère dans le cadre d'une **conception hiérarchique** des systèmes intégrés. La modélisation d'un bloc déjà conçu permet d'identifier les

paramètres clés de sa conception et de les relier au comportement électrique du bloc ; elle doit également tenir compte de besoins différents selon l'étape du flot de conception considérée. Ce sont des points qui ont été abordés au **chapitre 1**, sous un angle méthodologique. Ce chapitre met ainsi en avant des **méthodes générales** et se voulant **systematiques**, pour la **génération de modèles de circuits analogiques et mixtes**, considérés à tous les niveaux hiérarchiques de la conception d'un système. Un concept intéressant de **modèle multi-abstraction** y est souligné.

Le **chapitre 2** s'est quant à lui concentré sur un point de modélisation bien précis et incontournable dans le contexte des systèmes radiofréquence : il s'agit de la **modélisation du bruit** en général, du **bruit de phase** en particulier, et ce, dans le cadre d'une simulation événementielle adaptée aux **circuits à temps discrets**. Une technique a été exposée pour générer des **sources de bruit génériques, paramétrables**, permettant de couvrir les principaux gabarits de densité spectrale de puissance rencontrés dans le domaine RF. Un logiciel a été développé pour la **génération automatique** de ces modèles.

Le **chapitre 3** revient sur la nécessité de concevoir des circuits en vue de leur **réutilisation** dans différents projets de conception et donc de trouver un format et un ensemble de représentations capables d'encapsuler les caractéristiques du circuit et la connaissance liée à son processus de conception. Un paradigme de **conception analogique à base d'IPs** y est développé et appliqué à la **conception de circuits neuromimétiques**.

Le **chapitre 4** tient lieu de **projet de recherche**. Il se concrétise par la création de l'équipe E<sub>l</sub>IBio, que je dirigerai à partir de Janvier 2010, et organisée autour de l'**interaction des systèmes électroniques avec le vivant**. Ainsi ce chapitre situe la problématique de l'**acquisition et du contrôle de l'activité bioélectrique**, dans le but de développer une méthodologie de conception de **systèmes hybrides vivant/électronique**, pour l'acquisition et le contrôle **en temps réel** de l'activité bioélectrique. Ce projet se situe à la convergence de l'histoire de l'équipe ISN et de mes compétences en méthodes de conception analogique et mixte. Il s'agit d'un **projet fortement pluridisciplinaire** mêlant des chercheurs en électronique, biologie, neurosciences, chimie et médecine ; il s'inscrit localement dans le contexte du **pôle d'excellence Neurosciences** du PRES Université Bordeaux et se décline au travers de nombreux axes d'applications, ancrés dans la politique de financement de la recherche nationale et internationale. Les retombées attendues correspondent à deux types d'enjeu : d'une part la **compréhension des mécanismes physiologiques**, voire

physiopathologiques, et l'exploration de nouvelles **solutions thérapeutiques bioélectriques intégrées**, d'autre part la problématique de l'**intégration des systèmes hétérogènes**, qui relève d'une priorité des **STIC** (Sciences et Technologies de l'Information et de la Communication).



## REFERENCES BIBLIOGRAPHIQUES

- [ABS00] S. Abiteboul, P. Buneman, and D. Suciu, *Data on the web from relations to semistructured data and XML*, Morgan Kaufmann Publishers, 2000.
- [AOE] Alpha-Omega-Engineering, <http://www.alphaomega-eng.com>.
- [AS] N. Lewis, P. Desgreys, M. M. Louerat, Y. Hervé, and I.O'Connor, "AS SoC-AMS - SP3 : Méthodes et Outils pour la conception des SoC-AMS - Rapport final",
- [ASA02] R. Acosta, F. Silveira, and P. Aguirre, "Experiences on Analog Circuit Technology Migration and Reuse", Proceedings of the *Symposium on Integrated Circuits and Systems Design*, 2002, pp. 169-174.
- [AYA] Ayanda-Biosystems, <http://www.ayanda-biosys.com/index.html>.
- [BLB08] G. Bontorin, C. Lopez, Y. Bornat, N. Lewis, S. Renaud, A. Garenne, M. Chanaud, and G. LeMasson, "A real-time setup for multisite signal recording and processing in living neural networks", Proceedings of the *IEEE International Symposium on Circuits and Systems, ISCAS'08 Seattle*, 2008, pp. 2953-2956.
- [BOA00] K. A. Boahen, "Point-to-point connectivity between neuromorphic chips using address-events," *IEEE Transaction on Circuits and Systems II*, vol. 47, pp. 416-434, May 2000.
- [BOC00] R. Bowen and K. Coar, *Apache server*, Campus Press, 2000.
- [BOR06] Y. Bornat, "Réseaux de neurones sur silicium : une approche mixte analogique/numérique, pour l'étude des phénomènes d'adaptation, d'apprentissage et de plasticité", Thèse en Electronique, Université Bordeaux 1, 2006.
- [BOR98] C. Borchers, "Symbolic behavioral model generation of non-linear analog circuits," *IEEE Transactions on circuits and systems - II : Analog and digital signal processing*, vol. 45, n°. 10, pp. 1362-1371, Oct. 1998.
- [BPC74] G. R. Boyle, D. O. Pederson, B. M. Cohn, and J. E. Solomon, "Macromodeling of integrated circuit operational amplifiers," *IEEE Journal of Solid-State Circuits*, vol. 9, n°. 6, pp. 353 - 364, Dec. 1974.
- [BPG91] A. L. Benabid, P. Pollak, C. Gervason, D. Hoffmann, D. M. Gao, M. Hommel, J. E. Perret, and J. DeRougemont, "Long-term suppression of tremor by chronic stimulation of the ventral intermediate thalamic nucleus," *Lancet*, vol. 337, pp. 403-406, 1991.
- [BRB10] Y. Bornat, M. Raoux, Y. Boutaib, F. O. Morin, G. Charpentier, J. Lang, and S. Renaud, "Detection of electrical activity of pancreatic beta-cells using micro-electrode arrays", Proceedings of the *IEEE International Symposium on Electronic Design, Test & Applications, DELTA'10*, Ho Chi Minh City, 2010.
- [BRG07] G. Bontorin, S. Renaud, A. Garenne, L. Alvado, G. LeMasson, and J. Tomas, "A Real-Time Closed-Loop Setup for Hybrid Neural Networks", Proceedings of the *IEEE International Conference of the Engineering in Medicine and Biology Society, EMBS'07*, Lyon, 2007.
- [BTM08] B. Belhadj, J. Tomas, O. Malot, G. N'Kaoua, Y. Bornat, and S. Renaud, "FPGA-based Architecture for Real-time Synaptic Plasticity Computation", Proceedings of the *International Conference on Electronics, Circuits and Systems, ICECS'08*, Malta, 2008.
- [BTM09] B. Belhadj, J. Tomas, O. Malot, Y. Bornat, G. N'Kaoua, and S. Renaud, "Token-Passing Communication Protocol in Hardware based Real-time Spiking Neural Networks", Proceedings of the *International Conference on Bioengineered and Bioinspired Systems*, 2009.
- [BTR07] G. Bontorin, J. Tomas, and S. Renaud, "Low noise and low cost neural amplifiers", Proceedings of the *IEEE International Conference on Electronics, Circuits and Systems (ICECS 2007)*, Marrakech, 2007.

- [BWK94] H. Bergman, T. Wichmann, B. Karmon, and M. R. DeLong, "The primate subthalamic nucleus. II. Neuronal activity in the MPTP model of parkinsonism," *Journal of Neurophysiology*, vol. 72, pp. 507-520, 1994.
- [CADRF] Cadence, "RF block simulation", <http://www.cadence.com/products/rf/>.
- [CAG07] G. Cruccu, T. Z. Aziz, L. Garcia-Larrea, P. Hansson, T. S. Jensen, and J. P. Lefaucheur, "EFNS guidelines on neurostimulation therapy for neuropathic pain," *European Journal of Neurology*, vol. 14, pp. 952-970, 2007.
- [CAS07] R. Chawla, F. Adil, G. Serrano, and P. E. Hasler, "Programmable Gm-C Filters Using Floating-Gate Operational Transconductance Amplifiers," *IEEE Transaction on Circuits and Systems I :Regular papers*, vol. 54, n<sup>o</sup>. 3, pp. 481-491, Mar. 2007.
- [CFG06] R. Castro-Lopez, F. V. Fernandez, O. Guerra-Vinuesa, and A. Rodriguez-Vasquez, *Reuse-based methodologies and tools in the design of analog and mixed-signal integrated circuits*, Springer, 2006.
- [CMM99] J. K. Chapin, K. A. Moxon, R. S. Markowitz, and M. A. Nicolelis, "Real-time control of a robot arm using simultaneously recorded neurons in the motor cortex," *Nature Neuroscience*, vol. 2, pp. 664-670, 1999.
- [CTR06] I. O'Connor, F. Tissafi-Drissi, G. Revy, and F. Gaffiot, "UML/XML-based approach to hierarchical AMS synthesis", in *Applications of specification & design languages for SoCs - Best of FDL'05*, A. Vachoux, Ed.: Kluwer Academic Publishers, 2006.
- [CVT03] M. Chiappalonea, A. Vatoa, M. Tedescoa, M. Marcolib, F. Davidec, and S. Martinoia, "Networks of neurons coupled to microelectrode arrays: a neuronal sensory system for pharmacological applications," *Biosensors and Bioelectronics*, vol. 18, n<sup>o</sup>. 5-6, pp. 627-634, May 2003.
- [DEV94] Y. Deval, "Conception de circuits intégrés analogiques - Elaboration d'une technique originale d'optimisation basée sur les plans d'expériences", Thèse en Electronique, Université Bordeaux 1, 1994.
- [DLS96] A. Demir, E. Liu, and A. Sangiovanni-Vincentelli, "Time-domain non-Monte-Carlo noise simulation for nonlinear dynamic circuits with arbitrary excitations," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 15, n<sup>o</sup>. 5, pp. 493-505, May 1996.
- [DWB01] T. B. DeMarse, D. A. Wagenaar, A. W. Blau, and S. M. Potter, "The neurally controlled animat : biological brains acting with simulated bodies," *Autonomous Robots*, vol. 11, n<sup>o</sup>. 3, November 2001.
- [EBM03] D. Eytan, N. Brenner, and S. Marom, "Selective Adaptation in Networks of Cortical Neurons," *Journal of Neuroscience*, vol. 23, pp. 9349-9356, 2003.
- [ELD10] G. Eynard, N. Lewis, D. Dallet, and B. LeGal, "Efficient phase noise modelling of a PLL-based frequency synthesizer", Proceedings of the *International Symposium on Communications, Control and Signal Processing, ISCCSP'10*, submitted, 2010.
- [ESC08] R. Escola, "Traitement de signaux neuronaux sur système embarqué", Thèse en Electronique, INSA Lyon, 2008.
- [FAK02] A. Fakhfakh, "Contribution à la modélisation comportementale des circuits radio-fréquence", Thèse en électronique, Université Bordeaux 1, 2002.
- [FLL02] A. Fakhakh, H. Levi, N. Lewis, and Y. Danto, "Behavioral Modeling of Analogue and Mixed Integrated Systems with VHDL-AMS for RF Applications", Proceedings of the *SBCCI02, Symposium on Integrated Circuits and Systems Design*, Porto Alegre, Brésil, 2002.
- [FMB01] A. Fakhfakh, N. Milet-Lewis, J.-B. Bégueret, and H. Lévi, "VHDL-AMS Model of a Synchronous Oscillator including Phase Noise", Proceedings of the *FDL'01, Forum on Design Languages*, Lyon, France, 2001.
- [FMB02] A. Fakhfakh, N. Milet-Lewis, J.-B. Bégueret, and H. Lévi, "VHDL-AMS Model of a Synchronous Oscillator including Phase Noise", in *System on chip design languages: extended papers : best of FDL'01 and HDLCon'01*, A. Mignotte, E. Villar, and L. Horobin, Eds.: Kluwer Academic Publishers, 2002.
- [FMD01] A. Fakhfakh, N. Milet-Lewis, Y. Deval, and H. Lévi, "Study and behavioral simulation of phase noise and jitter in oscillators", Proceedings of the *2001 IEEE International Symposium on Circuits and Systems (ISCAS 2001)*, Sydney, Australie, 2001.

- [FML00] A. Fakhfakh, N. Milet-Lewis, H. Levi, and T. Zimmer, "A systematic approach for the behavioural modelling of analogue integrated systems", Proceedings of the *ACIDCA'2000, International Conference on Artificial and Computational Intelligence For Decision, Control and Automation In Engineering and Industrial Applications*, Monastir, Tunisia, 2000.
- [GET02] J. Georgiou and C. Toumazou, "A micropower cochlear prosthesis system", Proceedings of the *International Conference on Digital Signal Processing, DSP'02*, 2002, pp. 343- 348.
- [GHL01] P. R. Gray, P. J. Hurst, S. H. Lewis, and R. G. Meyer, *Analysis and design of analog integrated circuits*, 4th ed. ed., John Wiley & sons, INC., 2001.
- [GHO08] M. Ghovanloo, "Intergrated circuits for neural interfacing : neural stimulation", in *VLSI Circuits for Biomedical Applications*, K. Iniewski, Ed.: Artech House, 2008, pp. 191-199.
- [GIE02] G. Gielen, "Design and modeling tools for analog/RF circuits in mixed-signal integrated systems on chip", Proceedings of the *Forum on Design Languages, FDL'02*, Marseille, France, 2002.
- [GIS91] G. Gielen and W. Sansen, *Symbolic Analysis for Automated Design of Analog Integrated Circuits*. Amsterdam, Kluwer Academic Publisher, 1991.
- [GOS08] B. Gosselin and M. Sawan, "An Ultra Low-Power CMOS Action Potential Detector", Proceedings of the *IEEE International Symposium on Circuits and Systems, ISCAS'08*, Seattle, 2008.
- [GSC02] C. Galup-Montoro, M. C. Schneider, and R. Coitinho, "Resizing Rules for MOS Analog-Design Reuse," *IEEE Design and Test of Computers*, vol. 19, pp. 50-58, March 2002.
- [GU05] Q. Gu, *RF system design of transceivers for wireless communications*, Springer, 2005.
- [GUG01] M. Guglielmi, "Synthese de signaux en 1/f avec controle de la precision spectrale", in *GRETSI, Groupe d'Etudes du Traitement du Signal et des Images*, <http://documents.irevues.inist.fr/handle/2042/13217>.
- [GUN99] K. S. Guillory and R. A. Normann, "A 100-channel system for real time detection and storage of extracellular spike waveforms," *Journal of Neuroscience Methods*, vol. 91, n°. 1-2, pp. 21-29, 15 September 1999.
- [HAC03] R. R. Harrison and C. Charles, "A low-power, low-noise CMOS amplifier for neural recording applications," *IEEE J. Solid-State Circuits*, vol. 38, n°. 6, June 2003.
- [HAR08] R. R. Harrison, "The design of integrated circuits to observe brain activity," *Proceedings of the IEEE*, vol. 96, n°. 7, pp. 1203-1216, July 2008.
- [HEI09] F. Heimburger, "Parkinson n'est plus une fatalité", in *Chercheurs d'Aquitaine - Zoom sur les Neurosciences*. vol. 1, 2009, pp. 6-7.
- [HFR02] M. O. Heuschkel, M. Fejtl, M. Raggenbass, D. Bertrand, and P. Renaud, "A three-dimensional multi-electrode array for multi-site stimulation and recording in acute brain slices," *Journal of Neuroscience Methods* vol. 114, pp. 135–148, 2002.
- [HOH52] A. L. Hodgkin and A. F. Huxley, "A quantitative description of ion currents and its applications to conduction and excitation in nerve membranes," *Journal of Physiology*, vol. 117, pp. 500-544, 1952.
- [HSS04] R. R. Harrison, G. Santhanam, and V. V. Krishna, "Local Field Potential Measurement with Low-Power Analog Integrated Circuit", Proceedings of the *IEEE International Conference of the Eng. Med. Biol. Soc., EMBS'04*, San Francisco, 2004, pp. 4067-4070.
- [HWK07] R. R. Harrison, P. T. Watkins, R. J. Kier, R. O. Lovejoy, D. J. Black, B. Greger, and F. Solzbacher, "A low-power integrated circuit for a wireless 100-electrode neural recording system," *IEEE Journal of Solid-State Circuits*, vol. 42, pp. 123-133, Jan. 2007.
- [ISK08] R. Iskander, "Connaissance et synthèse en vue de la conception et la réutilisation de circuits analogiques intégrés", Thèse en Informatique, Université Paris VI, 2008.
- [ISN] ISN, "Engineering of Neuromorphic Systems Web page", <http://neuromorphic.ims-bordeaux.fr/>.
- [JDW09] T. Jochum, T. Denison, and P. Wolf, "Integrated circuit amplifiers for multi-electrode intracortical recording," *Journal of Neural Engineering*, vol. 6, January 2009.
- [KAS95] N. J. Kasdin, "Discrete simulation of colored noise and stochastic processes and 1/fa power law noise generation," *Proceedings of the IEEE*, vol. 83, n°. 5, pp. 802-827, May 1995.



- [KUN03] K. Kundert, "Introduction to RF simulation and its application", in *The Designer's Guide community*, Version 2, April 2003, <http://www.designers-guide.org/Analysis/>.
- [KUN06] K. Kundert, "Predicting the phase noise and jitter of PLL-based frequency synthesizers", in *The Designer's Guide community*, Version 4g, August 2006, <http://www.designers-guide.org/Analysis/>.
- [LAF98] A. Laflaquière, "Neurones artificiels sur Silicium : conception analogique et construction de réseaux hybrides", Thèse en électronique, Université Bordeaux 1, 1998.
- [LAH02] R. Levy, P. Ashby, W. D. Hutchison, A. E. Lang, A. M. Lozano, and J. O. Dostrovsky, "Dependence of subthalamic nucleus oscillations on movement and dopamine in Parkinson's disease," *Brain*, vol. 125, pp. 1196-1209, 2002.
- [LBA06] N. Lewis, Y. Bornat, L. Alvado, C. Lopez, A. Daouzli, T. Levi, J. Tomas, S. Saighi, and S. Renaud, "PAX : un outil logiciel / matériel d'investigation pour les neurosciences computationnelles", Proceedings of the *NeuroComp'06, Conférence française de Neurosciences Computationnelles*, Pont-à-Mousson, 2006.
- [LBB09] T. Levi, J. F. Beche, S. Bonnet, and R. Guillemaud, "IP-based Toolbox for Digital Signal Processing Reuse: Application to Real-time Spike Sorting", in *Design & Re-use, D&R Industry Articles*, 2009.
- [LBR09] N. Lewis, Y. Bornat, and S. Renaud, "Spiking Neural Network Hardware implementation : from single neurons to networks", Proceedings of the *NeuroComp'09, Conférence française de Neurosciences Computationnelles*, Bordeaux, 2009.
- [LEE02] D. C. Lee, "Analysis of jitter in phase-locked loops," *IEEE Transactions on Circuits And Systems-II: Analog and digital signal processing*, vol. 49, n°. 11, pp. 704-711, November 2002.
- [LEG05] B. LeGal, "Contribution à la prise en compte des contraintes des applications TDSI dans la synthèse de haut niveau", Thèse en Electronique, Université de Bretagne Sud, 2005.
- [LER07] N. Lewis and S. Renaud, "Spiking neural networks "in silico" : from single neurons to large scale networks", Proceedings of the *Systems, Signal and Devices Conference, SSD'07*, Hammamet, 2007.
- [LEV07] T. Levi, "Méthodologie de développement d'une bibliothèque d'IP-AMS en vue de la conception automatisée de systèmes sur puce analogiques et mixtes : application à l'ingénierie neuromorphique", Thèse en électronique, Université Bordeaux 1, 2007.
- [LEV59] E. C. Levy, "Complex-Curve fitting," *IRE Transactions on Automatic Control*, pp. 37-43, Mai 1959.
- [LEW04] L. Lewis, "Synthèse automatique de bruit via Scilab", ENSEEIHT - Toulouse, Rapport de stage, Juillet 2004.
- [LEW97] N. Milet-Lewis, "Contribution a la modélisation comportementale des circuits analogiques", Thèse en électronique, Université Bordeaux 1, 1997.
- [LEW98] M. S. Lewickiy, "A review of methods for spike sorting: the detection and classification of neural action potentials," *Network : Comput. Neural Syst.*, vol. 9, n°. 4, pp. R53-R78, 1998.
- [LLS08] T. Levi, N. Lewis, S. Saïghi, J. Tomas, Y. Bornat, and S. Renaud, "Neuromimetic Integrated Circuits", in *VLSI Circuits for Biomedical Applications*, K. Iniewski, Ed.: Artech House, 2008, pp. 241-262.
- [LLT06] T. Levi, N. Lewis, J. Tomas, and P. Fouillat, "Scaling Rules for MOS Analog Design Reuse", Proceedings of the *Mixed Design of Integrated Circuits and Systems MIXDES 2006*, Gdynia, Pologne, 2006.
- [LLT07a] T. Levi, N. Lewis, J. Tomas, and P. Fouillat, "IP-based design for analogue ASICs : a case study", in *Design & Re-use, D&R Industry Articles*, 2007.
- [LLT07b] T. Levi, N. Lewis, J. Tomas, and P. Fouillat, "IP-Based Library for Analog Design Reuse", Proceedings of the *Colloque du GDR SoC-SIP*, Paris, 2007.
- [LLT07c] T. Levi, N. Lewis, J. Tomas, and P. Fouillat, "Scaling Guidelines for CMOS Linear Analog Design", Proceedings of the *IEEE Ph.D. Research In Microelectronics and Electronics Conference, PRIME'07*, Bordeaux, 2007, pp. 209-212.
- [LLT08] T. Levi, N. Lewis, J. Tomas, and P. Fouillat, "IP-based methodology for analog design flow: application on neuromorphic engineering", Proceedings of the *IEEE NEWCAS-TAISA Conference, NEWCAS-TAISA'08*, Montréal, 2008, pp. 343-346.

- [LML05] N. Lewis, G. Monnerie, L. Lewis, J. Sabatier, and P. Melchior, "Génération de modèles de bruit en  $1/f^N$  pour la simulation des systèmes électroniques", Proceedings of the *Journées de l'action thématique "Les systèmes à dérivées non entières"*, SDNE'05, Bordeaux, 2005.
- [LML06] N. Lewis, G. Monnerie, L. Lewis, J. Sabatier, and P. Melchior, "Automatic procedure generating noise models for discrete-time applications", Proceedings of the *IEEE International Symposium on Circuits and Systems, ISCAS'06*, Kos Island, Greece, 2006.
- [LMS04] N. Lewis, G. Monnerie, J. Sabatier, P. Melchior, M. Robbe, and H. Levi, "Using fractional differentiation for modelling of  $1/f$  noise - Application to discrete-time noise sources in VHDL-AMS", Proceedings of the *IEEE International Symposium on Industrial Electronics, ISIE'04*, Ajaccio, France, 2004.
- [LRD02] G. LeMasson, S. Renaud, D. Debay, and T. Bal, "Feedback inhibition controls spike transfer in hybrid thalamic circuits," *Nature*, vol. 417, n°. 6891, pp. 854-858, 2002.
- [LTL07a] T. Levi, J. Tomas, N. Lewis, and P. Fouillat, "IP-based design reuse for analogue systems: a case study", Proceedings of the *SPIE VLSI Circuits and Systems III*, Maspalomas, Gran Canaria, 2007.
- [LTL07b] T. Levi, J. Tomas, N. Lewis, and P. Fouillat, "Resizing methodology for CMOS analog circuit", Proceedings of the *SPIE VLSI Circuits and Systems III* Maspalomas, Gran Canaria, 2007.
- [LTL09] T. Levi, J. Tomas, N. Lewis, and P. Fouillat, "CMOS Resizing Methodology for Analog Circuits," *IEEE Design and Test of Computers*, vol. 26, n°. 1, pp. 78-87, Feb. 2009.
- [MAS02] S. Marom and G. Shahaf, "Development, learning and memory in large random networks of cortical neurons: Lessons beyond anatomy," *Q. Rev. Biophys.*, vol. 35, pp. 63-87, 2002.
- [MCL99] N. Milet-Lewis, J.-J. Charlot, H. Levi, T. Zimmer, A. Laflaquière, and J.-B. Duluc, "Using VHDL-AMS to build a specific analogue behavioural model library", Proceedings of the *FDL'99, 2nd Forum on Design Languages*, E.N.S. Lyon, FRANCE, 1999.
- [MCS] MultiChannelSystems, <http://www.multichannelsystems.com>.
- [MCZ99] N. Milet-Lewis, J.-J. Charlot, T. Zimmer, H. Levi, and A. Laflaquière, "Comparing Four Analog HDL Modelings and Simulations of a Complex System", Proceedings of the *BMAS'99, IEEE/ACM International Workshop on Behavioral Modeling and Simulation*, Orlando, USA, 1999.
- [MED09] S. Medhat-Nazmi, "Génération automatique de modèles de performances pour circuits intégrés analogiques", Université Pierre et Marie Curie, M2 ACSI, Rapport de stage de fin d'études, Juin 2009.
- [MIS08] J. Michel and F. Schwartz, "Analogue circuit sizing method using interval analysis", Proceedings of the *2008 Joint International IEEE Northeast Workshop on Circuits and Systems and TAISA Conference, NEWCAS-TAISA*, Montreal, 2008, pp. 331-334.
- [MIZ99] N. Milet-Lewis and T. Zimmer, "Initiation à la conception analogique hiérarchique par modélisation comportementale", Proceedings of the *CETSI-EEA'99, Colloque sur l'Enseignement des Technologies et des Sciences de l'Information et des Systèmes*, Université Montpellier II, 1999.
- [MLD03] G. Monnerie, N. Lewis, D. Dallet, H. Levi, and M. Robbe, "Modelling of transient noise sources with VHDL-AMS and normative spectral interpretation", Proceedings of the *Forum on Design Languages, FDL'03*, Frankfurt, 2003.
- [MLL04] G. Monnerie, H. Levi, N. Lewis, and P. Fouillat, "Behavioral modeling of noise in discrete time systems with VHDL-AMS application to a sigma-delta", Proceedings of the *IEEE International Conference on Industrial Technology, ICIT'04*, Hammamet, 2004.
- [MMF01] N. Milet-Lewis, G. Monnerie, A. Fakhfakh, D. Geoffroy, Y. Hervé, H. Lévi, and J.-J. Charlot, "A VHDL-AMS library of RF blocks models", Proceedings of the *BMAS'01, IEEE International Workshop on Behavioural Modelling and Simulation*, Santa-Rosa, USA, 2001.
- [MON01] G. Monnerie, "Modélisation VHDL-AMS d'un synthétiseur de fréquence fractionnaire", ENSEIRB - Université Bordeaux 1, Rapport de stage de Master 2 Microélectronique, Juin 2009.
- [MON05] G. Monnerie, "Etude et modélisation de sources de bruit dans des structures à temps discret", Thèse en électronique, Université Bordeaux 1, 2005.

- [MSC04] S. Martinoia, V. Sanguinetti, L. Cozzi, L. Berdondini, J. V. Pelt, J. Tomas, G. LeMasson, and F. Davide, "Towards an embodied in vitro electrophysiology : the NeuroBIT project," *Neurocomputing*, vol. 58-60, pp. 1065-1072, March 2004.
- [MZD94] N. Milet-Lewis, T. Zimmer, and J. P. Dom, "Macromodeling of operational amplifiers", Proceedings of the *2nd European IC-CAP User Meeting*, Horbourg, 1994.
- [MZD96] N. Milet-Lewis, T. Zimmer, and J. P. Dom, "High-Frequency Modeling of Op-Amps," *Characterization Solutions, European Edition*, HP EEsof Publisher, pp. 6-11, June 1996.
- [MZD97] N. Milet-Lewis, T. Zimmer, J. B. Duluc, and A. Meresse, "Génération automatique d'un macromodèle multi-niveau d'amplificateur opérationnel", Proceedings of the *Colloque CAO de circuits intégrés et systèmes*, Grenoble, 1997.
- [NBR05] R. Narasimha, S. P. Bandi, R. M. Rao, and P. R. Mukund, "1/f noise synthesis model in discrete-time for circuit simulation," *IEEE Transactions on Circuits And Systems-I: Regular papers*, vol. 52, n°. 6, pp. 1104-1114, June 2005.
- [NBR08] Y. Nama, E. A. Brownb, J. D. Rossc, R. A. Blumb, B. C. Wheelera, and S. P. DeWeerth, "A retrofitted neural recording system with a novel stimulation IC to monitor early neural responses from a stimulating electrode," *Journal of Neuroscience Methods*, vol. 178, n°. 1, pp. 99-102, 30 March 2009.
- [NEU] NeuroNexus-Technologies, <http://www.neuronexustech.com/>.
- [NGU06] P. NguyenTuong, "Définition et implantation d'un langage de conception de composants analogiques réutilisables", Thèse en Informatique, Université Paris VI, 2006.
- [NIC01] M. A. Nicolelis, "Actions from thoughts," *Nature* vol. 409, pp. 403-407, 2001.
- [ODV95] A. L. Owens, T. J. Denison, H. Versnel, M. Rebbert, M. Peckerar, and S. A. Shamma, "Multi-electrode array for measuring evoked potentials from surface of ferret primary auditory cortex," *Journal of neuroscience methods*, vol. 58, n°. 1-2, pp. 209-220, 1995.
- [OLM00] A. Oustaloup, F. Levron, B. Mathieu, and F. Nanot, "Frequency band complex non integer differentiator : characterization and synthesis," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 47(1), 25-40, 2000., vol. 47, n°. 1, pp. 25 - 39, Jan 2000.
- [OUS95] A. Oustaloup, *La dérivation non entière : théorie, synthèse et applications dans les sciences pour l'ingénieur*, Editions Hermès, 1995.
- [PEG06] Y. Perelman and R. Ginosar, "Analog frontend for multichannel neuronal recording system with spike and LFP separation," *Journal of Neuroscience Methods*, vol. 153, pp. 21-26, 2006.
- [PEG08] Y. Perelman and R. Ginosar, *The NeuroProcessor - An Integrated Interface to Biological Neural Networks*, Springer, 2008.
- [PEPS] J. Michel, M. M. Louërat, N. Lewis, and I. O'Connor, "PEPS : Unified IC Sizing Approach", Rapport final, Janvier 2009.
- [PHK00] J. Phillips and K. Kundert, "An introduction to cyclostationary noise", in *The Designer's Guide community*, Version 1a, May 2000, <http://www.designers-guide.org/Theory/>.
- [PMR03] J. Park, K. Muhammad, and K. Roy, "Efficient generation of 1/f noise using a multi-rate filter bank", Proceedings of the *IEEE Custom Integrated Circuits Conference*, San Jose, USA, 2003.
- [RIG06] P. Rigaux, *Pratique de MySQL et Php*, O'Reilly, 2006.
- [RPC07] L. Rousseau, V. Perrais, G. Charvet, R. Guillemaud, G. Lissorgues, P. Meyrand, and B. Yvert, "New high-density 3D MEAs associated with integrated electronics system (BioMEA™)", Proceedings of the *Proceedings of the MicroTAS'2007 Conference*, Paris, 2007.
- [SAI04] S. Saighi, "Circuits et systèmes de modélisation analogique de réseaux de neurones biologiques : application au développement d'outils pour les neurosciences computationnelles", Thèse en Electronique, Université Bordeaux 1, 2004.
- [SCI] SCILAB, "version 4.1.2", <http://www.scilab.org/>.
- [SCK06] A. Savio, L. Colalongo, and Z. M. Kovacs-Vajna, "Automatic scaling rules procedures for analog design reuse," *IEEE Transactions on Circuits and Systems*, vol. 53, pp. 2539-2547, Dec. 2006.

- [SFB05] R. B. Staszewski, C. Fernando, and P. T. Balsara, "Event-driven simulation and modeling of phase noise of an RF oscillator," *IEEE Transactions on Circuits And Systems—I: Regular papers*, vol. 52, n°. 4, pp. 723-733, April 2005.
- [SHM01] G. Shahaf and S. Marom, "Learning in Networks of Cortical Neurons," *Journal of Neuroscience*, vol. 21, pp. 8782-8788, 2001.
- [SHV95] C. J. R. Shi and A. Vachoux, "VHDL-A design objectives and rationale", in *Current issues in electronic modeling*. vol. 2, K. A. Publishers, Ed., 1995, pp. 1-30.
- [SJA05] H. Scherberger, M. R. Jarvis, and R. A. Andersen, "Cortical Local Field Potential Encodes Movement Intentions in the Posterior Parietal Cortex," *Neuron*, vol. 46, n°. 2, pp. 347-354, 21 April 2005.
- [SWM06] R. Saleh, S. Wilton, S. Mirabbasi, A. Hu, M. Greenstreet, G. Lemieux, P. P. Pande, C. Grecu, and A. Ivanov, "System-on-Chip: Reuse and Integration," *Proceedings of the IEEE*, vol. 94, n°. 6, pp. 1050 - 1069, June 2006.
- [TEK99] M. Terrovitis and K. Kundert, "Device noise simulation of  $\Delta\Sigma$  modulators", in *The Designer's Guide community*, Version 1, August 1999, <http://www.designers-guide.org/Analysis/>.
- [THF04] W. H. Theodore and R. S. Fisher, "Brain stimulation for epilepsy," *Lancet Neurology*, vol. 3, n°. 2, pp. 111-118, 2004.
- [TLB07] T. Levi, N. Lewis, Y. Bornat, A. Destexhe, and J. Tomas, "An IP-based library for the design of analog Hodgkin Huxley neurons", *Proceedings of the Systems, Signal and Devices Conference, SSD'07*, Hammamet, 2007.
- [TOG04] F. Tissafi-Drissi, I. O'Connor, and F. Gaffiot, "RUNE: Platform for automated design of integrated multi-domain systems. Application to high-speed CMOS photoreceiver front-ends", *Proceedings of the IEEE Conference on Design Automation and Test in Europe, DATE'04*, Paris, 2004, pp. 16-21.
- [TSL72] C. A. Thomas, P. A. Springer, G. E. Loeb, Y. Berwald-Netter, and L. M. Okun, "A miniature microelectrode array to monitor the bioelectric activity of cultured cells," *Exp. Cell Res.*, vol. 74, pp. 61-6, 1972.
- [TTS02] D. M. Taylor, S. I. Tillery, and A. B. Schwartz, "Direct cortical control of 3D neuroprosthetic devices," *Science* vol. 296, pp. 1829-1832, 2002.
- [VAC02] A. Vachoux, "Méthodes et outils pour la modélisation de SoC-AMS", exposé pour AS SoC-AMS, CNRS, 13/11/02, Paris.
- [VMC06] R. J. Vogelstein, U. Mallik, and G. Cauwenberghs, "Beyond address-eventcommunication", in *Institute of Neuromorphic Engineering (INE) - The Neuromorphic Engineer*, <http://www.ine-news.org/>.
- [VPS08] M. Velliste, S. Perel, M. C. Spalding, A. S. Whitford, and A. B. Schwartz, "Cortical control of a prosthetic arm for self-feeding," *Nature* vol. 453, pp. 1098-1101, 2008.
- [VSI01] VSIA, "Analog/Mixed-Signal VSI Extension Specification Version 2.2", <http://vsi.org/docs/AMS-122-26Feb01.pdf>.
- [VSIA] "Virtual Socket Interface Alliance", <http://www.vsi.org>.
- [WEG09] X. F. Wei and W. M. Grill, "Impedance characteristics of deep brain stimulation electrodes in vitro and in vivo," *Journal of Neural Engineering*, vol. 6, 9 July 2009.
- [WLH05] J. D. Weiland, W. Liu, and M. S. Humayun, "Retinal Prosthesis," *Annual Review of Biomedical Engineering*, vol. 7, pp. 361-401, August 2005.
- [WMP05] D. A. Wagenaar, R. Madhavan, J. Pine, and S. M. Potter, "Controlling Bursting in Cortical Cultures with Closed-Loop Multi-Electrode Stimulation," *Journal of Neuroscience*, pp. 680-688, 2005.
- [WPP06] D. A. Wagenaar, J. Pine, and S. M. Potter, "Searching for plasticity in dissociated cortical cultures on multi-electrode arrays," *Journal of Negative Results in BioMedicine*, 26 Oct. 2006.
- [ZMF99] T. Zimmer, N. Milet-Lewis, A. Fakhfakh, B. Ardouin, H. Levi, J. B. Duluc, and P. Fouillat, "Hierarchical Analogue Design and Behavioural Modelling", *Proceedings of the IEEE MSE '99, Computer Society International Conference on Microelectronic Systems Education*, Arlington, US, 1999.



# LISTE DES PUBLICATIONS DE L'AUTEUR

## 8 articles :

- [P1] N. Milet-Lewis, T. Zimmer, and J.P. Dom, *High-Frequency Modeling of Op-Amps*, Characterization Solutions, European Edition, June 1996, HP EEsof Publisher: p. 6-11.
- [P2] T.Z. J.B. Duluc, N. Milet-Lewis, J.P. Dom and N. Vol. 36, 1996, *A wafer level reliability method for short-loop processing*, Microelectronics and Reliability, 1996. **36**(11-12): p. 1859-1862.
- [P3] T. Zimmer, J.B. Duluc, and N. Milet-Lewis, *Method for BJT transit time evaluation*, Electronics Letters, 1998. **34**(20).
- [P4] A. Fakhfakh, N. Milet-Lewis, J.-B. Bégueret, and H. Lévi, *VHDL-AMS Model of a Synchronous Oscillator including Phase Noise*, in *System on chip design languages: extended papers : best of FDL'01 and HDLCon'01*, A. Mignotte, E. Villar, and L. Horobin, Editors. 2002, Kluwer Academic Publishers.
- [P5] B. Mongellaz, F. Marc, N. Milet-Lewis, and Y. Danto, *Contribution to ageing simulation of complex analogue circuit using VHDL-AMS behavioural modelling*, Microelectronics Reliability, 2002. **42**(9-11): p. 1353-1358.
- [P6] T. Levi, N. Lewis, J. Tomas, and P. Fouillat, *IP-based design for analogue ASICs : a case study*, in *Design & Re-use, D&R Industry Articles*. 2007.
- [P7] N. Lewis, M. Billaud, D. Geoffroy, P. Cazenave, and T. Zimmer, *A distance measurement platform dedicated to electrical engineering*, IEEE Transactions on Learning Technologies, 2009. **2**.
- [P8] T. Levi, J. Tomas, N. Lewis, and P. Fouillat, *CMOS Resizing Methodology for Analog Circuits*, IEEE Design and Test of Computers, 2009. **26**(1): p. 78-87.

## 1 chapitre de livre :

- [L1] T. Levi, N. Lewis, S. Saïghi, J. Tomas, Y. Bornat, and S. Renaud, *Neuromimetic Integrated Circuits*, in *VLSI Circuits for Biomedical Applications*, K. Iniewski, Editor. 2008, Artech House. p. 241-262.

## 41 communications dont 1 en soumission :

- [C1] N. Milet-Lewis, T. Zimmer, and J.P. Dom. *Macromodeling of operational amplifiers*, in *2nd European IC-CAP User Meeting*. 1994, Horbourg.
- [C2] T. Zimmer, J.B. Duluc, N. Milet-Lewis, and J.P. Dom. *A new WLR method based on model parameter analysis*, in *IEEE Integrated Reliability Workshop*. 1995, Stanford Sierra Camp.
- [C3] T. Zimmer, S. Limtouch, P. LeHalpère, J.B. Duluc, N. Milet-Lewis, J.P. Dom, P. Roy, and F. Leger. *Process reliability assessment based on SPICE parameter analysis*, in *European Symposium on Reliability of Electron Devices, Failure Physics and Analysis, ESREF'95*. 1995, Arcachon.
- [C4] T. Zimmer, J. Berkner, B. Branciard, N. Milet-Lewis, J.B. Duluc, and J.P. Dom. *Method for Determining the Effective Base Resistance of Bipolar Transistors*, in *IEEE Bipolar / BiCMOS Circuits and Technology Meeting, BCTM'96*. 1996, Minneapolis.
- [C5] J.B. Duluc, T. Zimmer, N. Milet-Lewis, and J.P. Dom. *Determination of critical process steps for enhanced yield improvement*, in *SPIE, Microelectronic Manufacturing Yield, Reliability, and Failure Analysis II*. 1996, Austin.
- [C6] N. Milet-Lewis, T. Zimmer, J.B. Duluc, and A. Meresse. *Génération automatique d'un macromodèle multi-niveau d'amplificateur opérationnel*, in *Colloque CAO de circuits intégrés et systèmes*. 1997, Grenoble.
- [C7] T. Zimmer, J.B. Duluc, N.M. -Lewis, and J.P. Dom. *Advanced statistical tools for worst case parameter analysis*, in *European IC-CAP User Meeting*. 1997, Berlin.
- [C8] J.B. Duluc, T. Zimmer, N.M.-. Lewis, and J.P. Dom. *Worst case parameter generation methodology for analogue and digital circuit simulation*, in *IEEE International Conference on Microelectronics, ICM'95*. 1997, Jakarta.

- [C9] N. Milet-Lewis, J.-J. Charlot, H. Levi, T. Zimmer, A. Laflaquière, and J.-B. Duluc. *Using VHDL-AMS to build a specific analogue behavioural model library*, in *FDL'99, 2nd Forum on Design Languages*. 1999, E.N.S. Lyon, FRANCE.
- [C10] N. Milet-Lewis, J.-J. Charlot, T. Zimmer, H. Levi, and A. Laflaquière. *Comparing Four Analog HDL Modelings and Simulations of a Complex System*, in *BMAS'99, IEEE/ACM International Workshop on Behavioral Modeling and Simulation*. 1999, Orlando, USA.
- [C11] N. Milet-Lewis and T. Zimmer. *Initiation à la conception analogique hiérarchique par modélisation comportementale*, in *CETIS-EEA'99, Colloque sur l'Enseignement des Technologies et des Sciences de l'Information et des Systèmes*. 1999, Université Montpellier II.
- [C12] T. Zimmer, N. Milet-Lewis, A. Fakhfakh, B. Ardouin, H. Levi, J.B. Duluc, and P. Fouillat. *Hierarchical Analogue Design and Behavioural Modelling*, in *IEEE MSE '99, Computer Society International Conference on Microelectronic Systems Education*. 1999, Arlington, US.
- [C13] T. Zimmer, J.B. Duluc, B. Ardouin, N. Milet-Lewis, F. Marc, and P. Fouillat. *Comparison of three approaches for SPICE parameter determination*, in *European IC-CAP User Group Meeting*. 1999, Marseille.
- [C14] A. Fakhfakh, N. Milet-Lewis, H. Levi, and T. Zimmer. *A systematic approach for the behavioural modelling of analogue integrated systems*, in *ACIDCA'2000, International Conference on Artificial and Computational Intelligence For Decision, Control and Automation In Engineering and Industrial Applications*. 2000, Monastir, Tunisia.
- [C15] J.-F. Charlot, J.K. Seon, N. Milet-Lewis, H. Lévi, and J.-J. Charlot. *A VHDL-AMS approach for analysing thermo-electrical effects in EEPROMs*, in *International Conference on Mixed Design of Integrated Circuits and Systems, MIXDES'00*. 2000, Gdynia.
- [C16] J.-J. Charlot, J. Oudinot, N. Milet-Lewis, T. Zimmer, and H. Lévi. *Speaking VHDL-AMS, a newly standardized description language, for teaching mixed technology and mixed signal systems*, in *International Conference on Information Technology Based Higher Education and Training, ITHET'00*. 2000, Istanbul.
- [C17] N. Milet-Lewis, H. Lévi, T. Zimmer, and J.-L. Battaglia. *VHDL-AMS : un outil pour la simulation des systèmes électroniques et multi-technologiques*, in *Sixièmes Journées Pédagogiques du CNFM*. 2000, Saint-Malo.
- [C18] A. Fakhfakh, N. Milet-Lewis, J.-B. Bégueret, and H. Lévi. *VHDL-AMS Model of a Synchronous Oscillator including Phase Noise*, in *FDL'01, Forum on Design Languages*. 2001, Lyon, France.
- [C19] A. Fakhfakh, N. Milet-Lewis, Y. Deval, and H. Lévi. *Study and behavioral simulation of phase noise and jitter in oscillators*, in *2001 IEEE International Symposium on Circuits and Systems (ISCAS 2001)*. 2001, Sydney, Australie.
- [C20] N. Milet-Lewis, G. Monnerie, A. Fakhfakh, D. Geoffroy, Y. Hervé, H. Lévi, and J.-J. Charlot. *A VHDL-AMS library of RF blocks models*, in *BMAS'01, IEEE International Workshop on Behavioural Modelling and Simulation*. 2001, Santa-Rosa, USA.
- [C21] J.-J. Charlot, N. Milet-Lewis, T. Zimmer, and H. Lévi. *VHDL-AMS for mixed technology and mixed signal, an overview*, in *The 9th Mediterranean Conference on Control and Automation, MED'01*. 2001.
- [C22] A. Fakhfakh, H. Levi, N. Lewis, and Y. Danto. *Behavioral Modeling of Analogue and Mixed Integrated Systems with VHDL-AMS for RF Applications*, in *SBCCI02, Symposium on Integrated Circuits and Systems Design*. 2002, Porto Alegre, Brésil.
- [C23] N. Milet-Lewis, S. Snaidero, Y. Herve, G. Monnerie, and D. Geoffroy. *Behavioural library development : validation and qualification*, in *Forum on Design Languages, FDL'02*. 2002, Marseille.
- [C24] G. Monnerie, N. Lewis, D. Dallet, H. Levi, and M. Robbe. *Modelling of transient noise sources with VHDL-AMS and normative spectral interpretation*, in *Forum on Design Languages, FDL'03*. 2003, Frankfurt.
- [C25] N. Lewis, G. Monnerie, J. Sabatier, P. Melchior, M. Robbe, and H. Levi. *Using fractional differentiation for modelling of 1/f noise - Application to discrete-time noise sources in VHDL-AMS*, in *IEEE International Symposium on Industrial Electronics, ISIE'04*. 2004, Ajaccio, France.
- [C26] G. Monnerie, H. Levi, N. Lewis, and P. Fouillat. *Behavioral modeling of noise in discrete time systems with VHDL-AMS application to a sigma-delta*, in *IEEE International Conference on Industrial Technology, ICIT'04*. 2004, Hammamet.

- [C27] N. Lewis, G. Monnerie, L. Lewis, J. Sabatier, and P. Melchior. *Génération de modèles de bruit en  $1/f^N$  pour la simulation des systèmes électroniques*, in *Journées de l'action thématique " Les systèmes à dérivées non entières "*, SDNE'05. 2005, Bordeaux.
- [C28] T. Levi, N. Lewis, J. Tomas, and P. Fouillat. *Scaling Rules for MOS Analog Design Reuse*, in *Mixed Design of Integrated Circuits and Systems MIXDES 2006*. 2006, Gdynia, Pologne.
- [C29] N. Lewis, Y. Bornat, L. Alvado, C. Lopez, A. Daouzli, T. Levi, J. Tomas, S. Saighi, and S. Renaud. *PAX : un outil logiciel / matériel d'investigation pour les neurosciences computationnelles*, in *NeuroComp'06, Conférence française de Neurosciences Computationnelles*. 2006, Pont-à-Mousson.
- [C30] N. Lewis, G. Monnerie, L. Lewis, J. Sabatier, and P. Melchior. *Automatic procedure generating noise models for discrete-time applications*, in *IEEE International Symposium on Circuits and Systems, ISCAS'06*. 2006, Kos Island, Greece.
- [C31] T. Levi, N. Lewis, Y. Bornat, A. Destexhe, and J. Tomas. *An IP-based library for the design of analog Hodgkin Huxley neurons*, in *Systems, Signal and Devices Conference, SSD'07*. 2007, Hammamet.
- [C32] T. Levi, N. Lewis, J. Tomas, and P. Fouillat. *IP-Based Library for Analog Design Reuse*, in *Colloque du GDR SoC-SIP*. 2007, Paris.
- [C33] T. Levi, N. Lewis, J. Tomas, and P. Fouillat. *Scaling Guidelines for CMOS Linear Analog Design*, in *IEEE Ph.D. Research In Microelectronics and Electronics Conference, PRIME'07*. 2007, Bordeaux.
- [C34] T. Levi, J. Tomas, N. Lewis, and P. Fouillat. *IP-based design reuse for analogue systems: a case study*, in *SPIE VLSI Circuits and Systems III*. 2007, Maspalomas, Gran Canaria.
- [C35] T. Levi, J. Tomas, N. Lewis, and P. Fouillat. *Resizing methodology for CMOS analog circuit*, in *SPIE VLSI Circuits and Systems III 2007*, Maspalomas, Gran Canaria.
- [C36] N. Lewis and S. Renaud. *Spiking neural networks "in silico" : from single neurons to large scale networks*, in *Systems, Signal and Devices Conference, SSD'07*. 2007, Hammamet.
- [C37] G. Bontorin, C. Lopez, Y. Bornat, N. Lewis, S. Renaud, A. Garenne, M. Chanaud, and G. LeMasson. *A real-time setup for multisite signal recording and processing in living neural networks*, in *IEEE International Symposium on Circuits and Systems, ISCAS'08 2008*, Seattle.
- [C38] T. Levi, N. Lewis, J. Tomas, and P. Fouillat. *IP-based methodology for analog design flow: application on neuromorphic engineering*, in *IEEE NEWCAS-TAISA Conference, NEWCAS-TAISA'08*. 2008, Montréal.
- [C39] N. Lewis, T. Zimmer, M. Billaud, and D. Geoffroy. *An E-Learning Experience in Practical Electronics: Technological and Pedagogical Aspects*, in *5th Congress of Scientific Research Outlook in the Arab World*. 2008, Fez.
- [C40] H. Mnif, S. Sahnoun, F. Hdiji, D. Geoffroy, M. Billaud, N. Lewis, and T. Zimmer. *Electronic Practical Course via the Web in a Virtual Laboratory*, in *International Conference on Remote Engineering & Virtual instrumentation, REV'08*. 2008, Düsseldorf.
- [C41] G. Eynard, N. Lewis, D. Dallet, and B. LeGal. *Efficient phase noise modelling of a PLL-based frequency synthesizer*, in *International Symposium on Communications, Control and Signal Processing, ISCCSP'10*. 2010, submitted.

#### **1 communication invitée :**

- [CI1] N. Lewis, Y. Bornat, and S. Renaud. *Spiking Neural Network Hardware implementation : from single neurons to networks*, in *NeuroComp'09, Conférence française de Neurosciences Computationnelles*. 2009, Bordeaux.