



HAL
open science

Méthologie de développement d'une bibliothèque d'IP-AMS en vue de la conception automatisée de systèmes sur puces analogiques et mixtes: application à l'ingénierie neuromorphique

Timothée Levi

► **To cite this version:**

Timothée Levi. Méthologie de développement d'une bibliothèque d'IP-AMS en vue de la conception automatisée de systèmes sur puces analogiques et mixtes: application à l'ingénierie neuromorphique. Micro et nanotechnologies/Microélectronique. Université Sciences et Technologies - Bordeaux I, 2007. Français. NNT: . tel-00288469

HAL Id: tel-00288469

<https://theses.hal.science/tel-00288469>

Submitted on 17 Jun 2008

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

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

N° d'ordre : 3480

THÈSE

PRESENTEE A

L'UNIVERSITÉ BORDEAUX I

ECOLE DOCTORALE DES SCIENCES PHYSIQUES ET DE L'INGENIEUR

Par Timothée LEVI

POUR OBTENIR LE GRADE DE

DOCTEUR

SPÉCIALITÉ : Electronique

**METHODOLOGIE DE DEVELOPPEMENT D'UNE BIBLIOTHEQUE
D'IP-AMS EN VUE DE LA CONCEPTION AUTOMATISEE DE
SYSTEMES SUR PUCES ANALOGIQUES ET MIXTES :
APPLICATION A L'INGENIERIE NEUROMORPHIQUE**

Soutenue le : **4 décembre 2007**

Après avis de :

M. I. O'CONNOR	Professeur des Universités	Ecole Centrale Lyon, LEOM	Rapporteur
M. F.V. FERNANDEZ	Professeur des Universités	Université Séville, IMSE	Rapporteur

Devant la commission d'examen formée de :

Mme I. DUFOUR	Professeur des Universités	Université Bordeaux 1, IMS	Présidente
M. P. FOUILLAT	Professeur des Universités	ENSEIRB, IMS	Examineur
Mme M. - M. LOUËRAT	Chargée de Recherche CNRS	Université P. & M. Curie, LIP6	Examineur
Mme N. LEWIS	Maître de Conférences	Université Bordeaux 1, IMS	Examineur
M. J. MICHEL	Maître de Conférences	Université L. Pasteur, IneSS	Examineur
M. J. TOMAS	Maître de Conférences	Université Bordeaux 1, IMS	Examineur

A Jessie,

REMERCIEMENTS

Je remercie tout d'abord le professeur André TOUBOUL pour mon intégration depuis mon master recherche et pendant la réalisation de ces travaux au sein du laboratoire IMS (anciennement IXL), récemment dirigé par le professeur Pascal FOUILLAT.

Je remercie ce dernier de m'avoir offert ce sujet de recherche et de m'avoir soutenu en tant que directeur de thèse.

Mes remerciements vont également aux professeurs I. O'CONNOR et F.V. FERNANDEZ pour leurs lectures attentives et critiques du mémoire, en tant que rapporteurs de cette thèse.

Je tiens à remercier le professeur Isabelle DUFOUR d'avoir accepté de présider ce jury de thèse.

Je remercie M.M. LOUERAT et J. MICHEL pour l'intérêt qu'ils portent à ces travaux en acceptant de faire partie du jury de thèse.

Je remercie mes co-encadrants pour m'avoir fait confiance et soutenu tout au long de ces travaux mais également pour m'avoir initié à la recherche. Qu'ils trouvent ici, l'expression de ma profonde gratitude.

Ainsi, je remercie chaleureusement Noëlle LEWIS pour nos nombreuses discussions enrichissantes, sa disponibilité et son amitié.

Je remercie vivement Jean TOMAS pour ses conseils, son expérience et sa bonne humeur.

Je remercie le professeur Sylvie RENAUD pour la confiance qu'elle m'a accordée depuis mes études à l'ENSEIRB, en me proposant le stage à l'université d'Heidelberg et mon sujet de master recherche et également pour mon intégration dans l'équipe ISN (Ingénierie des Systèmes Neuromorphiques).

Remerciements

Un grand merci à toute l'équipe ISN, travailler avec vous est un grand plaisir, tant par votre amitié que par les nombreux goûters. Je tiens à remercier particulièrement Sylvain SAIGHI pour ses conseils.

Je souhaite aussi remercier l'ensemble du personnel administratif et technique des différents services du laboratoire IMS pour les nombreuses aides qu'ils ont pu me rendre durant mon quotidien de doctorant.

Merci aussi à tous les membres du laboratoire, qu'ils soient doctorants ou professeurs.

Je remercie ma famille, en particulier mes parents pour leurs soutiens et ma sœur pour ses talents de traductrice.

Je remercie infiniment Romain pour les nombreux cafés et les parties endiablées, Mathieu pour ses relectures et ses discussions footballistiques et Jean-Marie pour sa disponibilité.

Enfin, je tiens tout particulièrement à exprimer ma profonde reconnaissance à Jessie pour son réconfort et sa tendresse même pendant les moments difficiles.

TABLE DES MATIERES

INTRODUCTION GENERALE.....1

Chapitre 1 La conception de circuits analogique et mixte

I	INTRODUCTION.....	6
I.1	HISTORIQUE ET PERSPECTIVES DE LA CONCEPTION ELECTRONIQUE	6
I.2	UNE COMPLEXITE GRANDISSANTE	9
I.3	DEVELOPPEMENT DES METHODOLOGIES DE REUTILISATION	9
II	FLOT DE CONCEPTION AMS	10
II.1	EVOLUTION DU FLOT DE CONCEPTION NUMERIQUE	11
II.1.1	<i>Description du flot numérique.....</i>	<i>11</i>
II.1.2	<i>Outils existant pour supporter le flot de conception des systèmes numériques</i>	<i>13</i>
II.1.3	<i>IP numérique.....</i>	<i>15</i>
II.2	EVOLUTION DU FLOT DE CONCEPTION ANALOGIQUE ET MIXTE	17
II.2.1	<i>Description du flot analogique et mixte</i>	<i>17</i>
II.2.2	<i>Outils existant pour supporter le flot de conception des systèmes AMS.....</i>	<i>20</i>
II.2.3	<i>Synthèse analogique.....</i>	<i>22</i>
II.2.4	<i>Conception analogique à base d'IPs : principe et difficultés.....</i>	<i>23</i>
III	REUTILISATION DE LA CONCEPTION ANALOGIQUE : ETAT DE L'ART	27
III.1	TRAVAUX D'EQUIPES FRANÇAISES	27
III.1.1	<i>Equipe ASIM du LIP6, Université Pierre et Marie Curie Paris VI.....</i>	<i>27</i>
III.1.2	<i>Equipe LEOM de l'Ecole Centrale Lyon.....</i>	<i>28</i>
III.2	TRAVAUX D'EQUIPES EUROPEENNES.....	30
III.2.1	<i>Equipe de l'Université Polytechnique de Madrid</i>	<i>30</i>
III.2.2	<i>Equipe IMSE de l'Université de Seville</i>	<i>31</i>
III.2.3	<i>Equipe de Katholieke Universiteit Leuven (KUL).....</i>	<i>32</i>
III.3	AUTRES TRAVAUX	33
III.3.1	<i>Stockage des connaissances du concepteur</i>	<i>33</i>
III.3.2	<i>Dimensionnement et layout automatiques.....</i>	<i>33</i>
III.3.3	<i>Point de vue sur l'IP analogique réutilisable.....</i>	<i>35</i>
III.3.4	<i>L'utilisation du langage XML</i>	<i>37</i>
III.4	SYNTHESE	37
IV	CONCLUSION.....	39

Chapitre 2

Conception de Neurones sur Silicium

I	INTRODUCTION.....	42
II	MODELE DE NEURONE IMPLEMENTE.....	43
II.1	FORMALISME D’HODGKIN HUXLEY	43
II.2	EQUATIONS IMPLEMENTEES	45
II.3	EQUIVALENCE ENTRE LES PARAMETRES BIOLOGIQUES ET ELECTRIQUES	45
III	CONCEPTION D’UN ASIC NEUROMIMETIQUE	46
III.1	LE LOGICIEL NEURON	46
III.2	METHODOLOGIE DE CONCEPTION DE NEURONES SUR SILICIUM.....	48
III.3	HIERARCHIE DE L’ASIC.....	50
III.4	PRESENTATION DES DIFFERENTES CELLULES	51
III.4.1	<i>La fonction sigmoïde</i>	51
III.4.2	<i>La fonction cinétique</i>	52
III.4.3	<i>La fonction puissance</i>	54
III.4.4	<i>La fonction sortie</i>	55
III.5	UN EXEMPLE D’ASIC NEUROMIMETIQUE : LE CIRCUIT GALWAY	56
IV	DISCUSSION SUR LE CHOIX DU DOMAINE D’APPLICATION	57
V	CONCLUSION.....	58

Chapitre 3

Bibliothèque d’IPs Analogiques : Développement et Exploration

I	INTRODUCTION.....	60
II	DEFINITION ET CONTENU D’UN IP ANALOGIQUE.....	61
II.1	DEFINITION GENERALE	61
II.2	ANALYSE DES VUES MODELISEES EN VERILOGA.....	64
II.3	APPLICATION SUR UN BLOC ELEMENTAIRE.....	66
II.3.1	<i>Vue symbole</i>	67
II.3.2	<i>Vue connectique</i>	68
II.3.3	<i>Vue fonctionnelle</i>	69
II.3.4	<i>Vue comportementale</i>	70
II.3.5	<i>Vue schéma</i>	71
II.3.6	<i>Vue layout</i>	72
II.3.7	<i>Vue documentation</i>	72
III	CARACTERISATION D’UN IP ANALOGIQUE.....	73
III.1	CAS GENERAL	73
III.2	APPLICATION	74
III.2.1	<i>Détermination de la résistance d’entrée et de la résistance de sortie</i>	74
III.2.2	<i>Détermination des spécifications</i>	75
III.2.3	<i>Détermination de la robustesse de la cellule</i>	78
IV	CREATION DE LA BASE DE DONNEES	80
IV.1	OBJECTIFS.....	80
IV.2	MISE EN OEUVRE.....	80

V	EXPLORATION DE LA BASE DE DONNEES.....	83
V.1	PRINCIPE ET OUTILS DE L'EXPLORATION.....	83
V.2	MISE EN OEUVRE.....	85
V.2.1	<i>Entrée des spécifications par le client.....</i>	<i>86</i>
V.2.2	<i>Choix du concepteur, création des requêtes et exploration de la base de données.....</i>	<i>89</i>
V.2.3	<i>Récapitulatif et résultats.....</i>	<i>91</i>
VI	AIDE A LA RECONCEPTION EN CAS D'ECHEC.....	93
VI.1	IDENTIFICATION DU PROBLEME.....	93
VI.2	PROBLEME CONCERNANT LES SPECIFICATIONS DEMANDEES.....	93
VI.3	PROBLEME DU A LA TECHNOLOGIE OU A LA TENSION D'ALIMENTATION.....	94
VII	CONCLUSION.....	94

Chapitre 4

Conception automatisée d'un ASIC neuromorphique : Exemples d'application

I	INTRODUCTION.....	96
II	EXEMPLE 1 : CONCEPTION AUTOMATISEE DE L'ASIC GALWAY.....	96
II.1	SPECIFICATIONS.....	97
II.2	EXPLORATION DE LA BASE DE DONNEES.....	102
II.3	VERIFICATION.....	107
II.3.1	<i>Vérification électrique du système complet.....</i>	<i>107</i>
II.3.2	<i>Vérification par rapport au simulateur Neuron.....</i>	<i>109</i>
III	EXEMPLE 2 : IDENTIFICATION D'UN ECHEC ET CONCEPTION D'UN IP MANQUANT.....	110
III.1	SPECIFICATIONS.....	110
III.2	EXPLORATION ET IDENTIFICATION DU PROBLEME.....	112
III.3	RESOLUTION DU PROBLEME.....	113
III.4	AJOUT D'UN NOUVEL IP DANS LA BASE DE DONNEES.....	115
III.5	VERIFICATION.....	116
IV	EXEMPLE 3 : AJOUT D'UNE NOUVELLE FONCTION NEUROMORPHIQUE.....	117
IV.1	DIFFERENTES ETAPES.....	117
IV.2	CREATION ET VALIDATION DU MODELE FONCTIONNEL.....	118
IV.3	VALIDATION DE LA NOUVELLE FONCTION DANS LE SYSTEME ET CONCLUSION.....	119
V	CONCLUSION.....	122

Chapitre 5

Le redimensionnement : un atout majeur dans la réutilisation

I	INTRODUCTION.....	124
II	ETAT DE L'ART.....	124
II.1	VUE D'ENSEMBLE DES DIFFERENTS TRAVAUX CONCERNANT LE REDIMENSIONNEMENT	124
II.2	METHODOLOGIE DE C. GALUP-MONTORO ET M.C. SCHNEIDER [GALU 2002]	125
II.3	ETUDE D'ACOSTA, SILVEIRA ET AGUIRRE [ACOS 2002]	128
II.4	TRAVAIL DE SAVIO, COLALONGO ET KOVACS-VAJNA [SAVI 2004][SAVI 2006].....	128
II.5	REALISATIONS INDUSTRIELLES	130
<i>II.5.1</i>	<i>Neocell et Neocircuit.....</i>	<i>130</i>
<i>II.5.2</i>	<i>Barcelona Design.....</i>	<i>130</i>
III	DESCRIPTION DE LA METHODOLOGIE PROPOSEE	130
III.1	PLAN DE LA METHODOLOGIE.....	130
III.2	DEFINITION DES FACTEURS DE REDIMENSIONNEMENT [LEVI 2006A].....	132
III.3	ANALYSE DE LA 3EME ETAPE DE LA METHODOLOGIE.....	132
III.4	DISCUSSION SUR LE CHAMP D'APPLICATION DE LA METHODOLOGIE	133
IV	APPLICATION SUR DES CIRCUITS LINEAIRES.....	135
IV.1	L'OTA.....	135
IV.2	PREMIERE APPLICATION DE LA METHODOLOGIE: CONSERVATION DU GAIN EN TRANSCONDUCTANCE.....	135
<i>IV.2.1</i>	<i>Redimensionnement.....</i>	<i>135</i>
<i>IV.2.2</i>	<i>Résultats</i>	<i>137</i>
<i>IV.2.3</i>	<i>Evaluation de la robustesse.....</i>	<i>139</i>
IV.3	DEUXIEME APPLICATION : CONSERVATION DU GAIN EN TRANSCONDUCTANCE ET REDUCTION DE LA CONSUMMATION EN PUISSANCE	141
<i>IV.3.1</i>	<i>Redimensionnement.....</i>	<i>141</i>
<i>IV.3.2</i>	<i>Résultats</i>	<i>142</i>
IV.4	CONCLUSION CONCERNANT LES EXEMPLES LINEAIRES	143
V	APPLICATION SUR DES CIRCUITS NON-LINEAIRES	144
V.1	L'OSCILLATEUR EN ANNEAU	144
V.2	L'INVERSEUR CMOS	145
V.3	REDIMENSIONNEMENT	145
V.4	RESULTATS DES SIMULATIONS	146
V.5	EVALUATION DE LA ROBUSTESSE.....	148
VI	CONCLUSION ET PERSPECTIVES	150
	CONCLUSION GENERALE ET PERPESPECTIVES.....	151
	PUBLICATIONS DE L'AUTEUR.....	155
	BIBLIOGRAPHIE.....	157
	ANNEXES.....	169
	Annexe 1 : Anatomie et Physiologie d'un neurone.....	171
	Annexe 2 : Manuel d'utilisation de la bibliothèque d'IPs et de son exploration.....	179
	Annexe 3 : Modèle fonctionnel de la cinétique à 2 constantes de temps.....	203

TABLE DES FIGURES

Figure 1.1	Loi de Moore [GIME 2001]	7
Figure 1.2	Flot de conception numérique à base d'IPs [LEGA 2005]	12
Figure 1.3	Flot de conception SystemC [POLT 2007]	15
Figure 1.4	Flot de conception d'un circuit mixte [DESG 2004]	20
Figure 1.5	Illustration du flot de conception à base d'IPs	24
Figure 1.6	Flot de conception des Soft IP et Hard IP [O'CON 2005a]	29
Figure 1.7	Flot de conception AMS automatisé [ODA 2002]	35
Figure 2.1	Illustration de la collaboration entre les biologistes et les microélectroniciens	43
Figure 2.2	Modélisation du neurone	44
Figure 2.3	Capture d'écran du logiciel Neuron	47
Figure 2.4	Méthodologie de conception et de validation d'un circuit intégré neuromimétique	48
Figure 2.5	Allure des potentiels d'action sous Neuron d'un neurone exciteur	48
Figure 2.6	Neurone inhibiteur avec le logiciel Neuron	49
Figure 2.7	Neurone inhibiteur d'un ASIC simulé par Cadence	49
Figure 2.8	Description et correspondance de la hiérarchie	50
Figure 2.9	Implémentation électronique des fonctions mathématiques	51
Figure 2.10	Synopsis d'un courant ionique	51
Figure 2.11	Synopsis de la fonction sigmoïdale	52
Figure 2.12	Schéma synoptique de la fonction cinétique	53
Figure 2.13	Schéma bloc de la fonction cinétique	53
Figure 2.14	Schéma électrique de la fonction puissance m^3h	54
Figure 2.15	Synopsis de l'étage de sortie d'une conductance	56
Figure 2.16	Routage final du circuit Galway	56
Figure 2.17	Activités de trois neurones (A, B, C) du circuit Galway	57
Figure 3.1	Hiérarchie d'un ASIC neuromorphique	63
Figure 3.2	Représentation du contenu d'une fonction	64
Figure 3.3	Schéma complet de la fonction Sigmoïde	66
Figure 3.4	Symbole de la cellule sigmoïde	67
Figure 3.5	Schéma au niveau transistor de la cellule sigmoïde	71
Figure 3.6	Layout de l'ip1_sigmoid_act	72
Figure 3.7	Résistance de sortie de la cellule sigmoïde	74
Figure 3.8	Réponse DC de la cellule sigmoïde selon plusieurs valeurs de V_{offset}	76
Figure 3.9	Réponse DC de la cellule sigmoïde selon plusieurs valeurs de K_{pente}	76
Figure 3.10	Evaluation de l'erreur quadratique moyenne selon la valeur V_{offset} utilisée	77
Figure 3.11	Simulation MonteCarlo de la cellule sigmoïde	78
Figure 3.12	Figure 3.11. Simulation MonteCarlo de la cellule sigmoïde avec pour valeurs d'entrée les milieux des intervalles	79
Figure 3.13	Simulation MonteCarlo de la cellule sigmoïde avec pour valeurs d'entrée les bornes supérieures des intervalles	79

Table des figures

Figure 3.14	Simulation MonteCarlo de la cellule sigmoïde avec pour valeurs d'entrée les bornes inférieures des intervalles	79
Figure 3.15	Présentation de la hiérarchie d'un ASIC neuromorphique	80
Figure 3.16	Exploration de la base de données	83
Figure 3.17	Illustration de la démarche de résultats	86
Figure 3.18	Diagramme UML des Use Case de la conception du système Neuromorphique	87
Figure 3.19	Diagramme UML d'activités des choix du client	88
Figure 3.20	Capture d'écran du formulaire d'entrée des spécifications	89
Figure 3.21	Diagramme UML d'activité de toutes les étapes du formulaire	89
Figure 3.22	Diagramme UML d'activités de la création des requêtes	90
Figure 3.23	Diagramme UML d'activités des résultats	91
Figure 3.24	Diagramme UML d'activités de la création des fichiers XML et HTML	92
Figure 3.25	Tableaux récapitulatif de l'exploration	93
Figure 4.1	Présentation de la hiérarchie d'un ASIC neuromorphique	96
Figure 4.2	Différentes étapes de la conception du système à partir d'une bibliothèque d'IPs	97
Figure 4.3	Capture d'écran des choix des conductances	99
Figure 4.4	Capture d'écran de la saisie des paramètres des conductances	100
Figure 4.5	Capture d'écran des choix du concepteur	101
Figure 4.6	Capture d'écran des spécifications électriques	101
Figure 4.7	Capture d'écran des diagnostics de l'exploration	103
Figure 4.8	Table ipsystem_links	104
Figure 4.9	Architecture entre un Macro-bloc neurone et ces blocs génériques	104
Figure 4.10	Vue d'ensemble du circuit final	105
Figure 4.11	Récapitulatif et liens des IPs choisis	107
Figure 4.12	Comparaison entre le modèle comportemental et le modèle niveau transistor du neurone FS	108
Figure 4.13	Simulation mixte des cinq neurones	108
Figure 4.14	Configuration de la simulation mixte	109
Figure 4.15	Comparaison des courbes Fréquences/Stimulation données par le logiciel Neuron et par l'ASIC conçu. A) Simulations par Neuron. B) Simulations de l'ASIC	110
Figure 4.16	Diagnostics de l'exploration	112
Figure 4.17	Récapitulatif de l'exploration et raisons de l'échec	113
Figure 4.18	Comparaison entre les différents IPs et le modèle fonctionnel	114
Figure 4.19	Capture d'écran de l'interface graphique permettant l'ajout d'un nouvel IP	115
Figure 4.20	Diagnostics de la nouvelle exploration	116
Figure 4.21	Simulation du modèle fonctionnel de la nouvelle fonction cinétique	118
Figure 4.22	Comparaison entre les courants Sodium avec des cinétiques différentes	119
Figure 4.23	Comparaison de l'activité électrique de deux neurones possédant des cinétiques différentes	120
Figure 4.24	Comparaison des activités sous le logiciel Neuron	120
Figure 5.1	Organigramme de la procédure d'optimisation pour la réponse en fréquence	129
Figure 5.2	Topologie de l'OTA	135
Figure 5.3	Comportement DC des différents OTAs	137

Table des figures

Figure 5.4	Schéma de la simulation	139
Figure 5.5	Simulations MonteCarlo du gain en tension DC de l'OTA en technologie 0,8 μm et 5 V (à gauche) et en 0,35 μm et 5 V (à droite)	140
Figure 5.6	Simulations MonteCarlo du gain en tension DC de l'OTA en technologie 0,8 μm et 5 V (à gauche) et en 0,35 μm et 5 V (à droite)	140
Figure 5.7	Simulations MonteCarlo du gain en tension DC de l'OTA optimisé en technologie 0,12 μm et 1,2 V	140
Figure 5.8	Comportement DC des différents OTAs	142
Figure 5.9	Un oscillateur en anneau	144
Figure 5.10	Caractéristiques de transfert d'un inverseur CMOS	145
Figure 5.11	Comportement transitoire des oscillateurs en anneau	148
Figure 5.12	Simulation MonteCarlo de la fréquence d'oscillation de l'oscillateur pour la technologie 0,8 μm avec 5 V (à gauche) et en technologie 0,35 μm avec 3,3 V (à droite)	149
Figure 5.13	Simulation MonteCarlo de la fréquence d'oscillation de l'oscillateur en anneau pour la technologie 0.12 μm avec 1.2 V et en utilisant la fonction d'optimisation	149

LISTE DES TABLEAUX

Tableau 1.1	2006 : Bilan et estimation des besoins en conception de circuits et systèmes	9
Tableau 1.2	Classification des IPs numériques	16
Tableau 1.3	Différents outils du flot de conception [DESG 2004]	21
Tableau 1.4	Synthèse des travaux	38
Tableau 2.1	Coefficients appliqués aux paramètres du modèle pour l'intégration	46
Tableau 3.1	Différentes vues incluses dans un IP	62
Tableau 3.2	Explication de la vue connectique	68
Tableau 3.3	Explication de la vue fonctionnelle	69
Tableau 3.4	Explication de la vue comportementale	70
Tableau 3.5	Intervalles de valeurs des entrées	72
Tableau 3.6	Intervalles de validité des entrées du bloc sigmoïde	77
Tableau 4.1	Spécifications du neurone FS	98
Tableau 4.2	Spécifications du neurone RS	98
Tableau 4.3	Spécifications du neurone à 5 conductances	98
Tableau 4.4	Spécifications du neurone FS	111
Tableau 4.5	Spécifications du neurone RS	111
Tableau 5.1	Règles de redimensionnement selon la stratégie	126
Tableau 5.2	Redimensionnement d'un OTA	127
Tableau 5.3	Définition des facteurs de redimensionnement	132
Tableau 5.4	Valeurs des facteurs de redimensionnement technologiques	135
Tableau 5.5	Récapitulatif des dimensions des transistors	136
Tableau 5.6	Résultats des simulations	137
Tableau 5.7	Comparaison entre les tailles des transistors	138
Tableau 5.8	Comparaison des performances avec ou sans la boucle d'optimisation	138
Tableau 5.9	Tailles des transistors	141
Tableau 5.10	Tailles des transistors	142
Tableau 5.11	Résultats des simulations	146
Tableau 5.12	Nouvelles tailles des transistors	147
Tableau 5.13	Fréquence d'oscillation	148

INTRODUCTION GENERALE

Depuis la naissance du premier circuit intégré dans les années soixante, le domaine de la microélectronique n'a cessé de se développer. Le perfectionnement constant des procédés de fabrication, des méthodologies de conception et des logiciels de CAO (Conception Assistée par Ordinateur) a conduit aux densités d'intégration que l'on connaît aujourd'hui. Un cycle complet de fabrication a pour objectif l'obtention d'un système conforme aux spécifications initiales, au moindre coût et en un temps le plus court possible.

Pour répondre à ces exigences, la réutilisation de circuits déjà conçus paraît aujourd'hui indispensable. Elle existe déjà dans la conception numérique et est basée sur la notion d'IP (*Intellectual Property*). Plus généralement, les systèmes numériques bénéficient aujourd'hui d'outils et méthodes standardisés qui ont permis l'automatisation complète du flot de conception. Bien que la nature des signaux et circuits soit très différente, le flot de conception analogique a intérêt à s'inspirer du flot de conception numérique.

Selon le thème *Design* de l'International Technology Roadmap for Semiconductors [ITRS 2006], les méthodologies de réutilisation pour les circuits analogiques sont en pleine expansion et paraissent incontournables pour les conceptions futures. Par exemple, la maturité des méthodes et outils de modélisation AMS (*Analog and Mixed Signal*) devrait atteindre 73 % du niveau de ceux de la conception numérique en 2013. Ainsi le développement d'un flot de conception analogique basée sur la réutilisation est essentiel et nécessite l'extension et l'adaptation de l'IP au domaine analogique.

Les travaux présentés dans ce mémoire apportent une contribution à l'automatisation du flot de conception analogique et mixte. En ayant pour principal objectif de favoriser la réutilisation de blocs déjà conçus, les questions abordées couvrent la notion d'IP analogique, l'organisation de ces IPs dans une base de données intégrée au flot de conception et le redimensionnement de circuits CMOS lors d'une migration technologique. Nos travaux sont articulés en cinq chapitres.

Le premier chapitre est une description de la conception actuelle de circuits analogiques et mixtes. Il propose une analyse de chaque flot de conception et passe en revue les outils associés. Un état de l'art des développements récents est exposé ; on y voit comment chaque équipe envisage et applique la notion de réutilisation en conception analogique. Finalement, un flot de conception *idéal* est proposé, qui s'appuie sur l'usage d'une bibliothèque d'IPs analogiques.

Le chapitre deux s'intéresse au domaine d'application du système choisi comme démonstrateur. Il n'est pas à proprement parler un élément de notre méthodologie mais s'avère indispensable pour éclairer le lecteur sur les spécificités de notre cas d'étude. Ce chapitre traite ainsi de la conception d'un système neuromimétique, c'est-à-dire reproduisant l'activité électrique de neurones biologiques. Les différentes étapes de la conception d'un tel système sont présentées. On y décrit les modèles implémentés, la hiérarchie et les différents blocs constitutifs d'un ASIC neuromimétique. Enfin, une discussion sur le choix de cette application a lieu, en lien avec notre objectif de flot de conception optimisé par la réutilisation.

Le chapitre trois propose une méthodologie de conception de circuits analogiques et mixtes. Cette méthodologie est basée sur la définition et la caractérisation d'un IP analogique. Ces IPs sont regroupés et intégrés dans une base de données. Cette bibliothèque d'IPs sera le cœur d'une plate-forme de conception automatisée de systèmes neuromimétiques. La mise en place de cette bibliothèque et le principe de son exploration sont décrits dans ce chapitre.

Le chapitre quatre présente trois exemples d'utilisation de la bibliothèque d'IPs, correspondant chacun à un projet de conception plausible. Ces exemples sont appliqués au démonstrateur décrit au chapitre deux. La première application décrit la conception automatisée d'un ASIC neuromorphique s'appuyant sur l'exploration de la bibliothèque. Le deuxième traite d'un cas d'échec partiel de l'exploration automatique. Il montre comment il est possible de reconcevoir rapidement un nouvel IP, grâce aux informations attachées à l'IP aux spécifications les plus proches, présent dans la bibliothèque. Le troisième exemple concerne l'ajout d'une nouvelle fonction neuromorphique et l'enrichissement de la bibliothèque d'IPs.

Le chapitre cinq présente une méthodologie de redimensionnement de circuits analogiques CMOS lors d'une migration technologique. Ce cas peut se rattacher à une autre possibilité d'échec de l'exploration de la bibliothèque d'IPs : un IP aux spécifications convenables a été trouvé mais il n'est pas dans la technologie voulue. La conservation des performances du circuit, lors d'un passage d'une technologie à une autre, est l'objectif principal du redimensionnement. Après un état de l'art sur cette question, une méthodologie est proposée pour calculer les nouvelles valeurs des paramètres géométriques (W : la largeur et L : la longueur) de chaque transistor MOS. Elle est appliquée à des circuits analogiques linéaires et non-linéaires.

Ce mémoire s'achève par une discussion sur les perspectives d'application de ces travaux à court et moyen terme.

CHAPITRE 1 :

LA CONCEPTION DE CIRCUITS ANALOGIQUES ET MIXTES

SOMMAIRE

I	INTRODUCTION.....	6
I.1	HISTORIQUE ET PERSPECTIVES DE LA CONCEPTION ELECTRONIQUE	6
I.2	UNE COMPLEXITE GRANDISSANTE	7
I.3	DEVELOPPEMENT DES METHODOLOGIES DE REUTILISATION	8
II	FLOT DE CONCEPTION AMS	10
II.1	EVOLUTION DU FLOT DE CONCEPTION NUMERIQUE.....	11
II.1.1	<i>Description du flot numérique.....</i>	<i>11</i>
II.1.2	<i>Outils existant pour supporter le flot de conception des systèmes numériques</i>	<i>13</i>
II.1.3	<i>IP numérique.....</i>	<i>15</i>
II.2	EVOLUTION DU FLOT DE CONCEPTION ANALOGIQUE ET MIXTE	17
II.2.1	<i>Description du flot analogique et mixte</i>	<i>17</i>
II.2.2	<i>Outils existant pour supporter le flot de conception des systèmes AMS.....</i>	<i>20</i>
II.2.3	<i>Synthèse analogique.....</i>	<i>22</i>
II.2.4	<i>Conception analogique à base d'IPs : principe et difficultés.....</i>	<i>23</i>
III	REUTILISATION DE LA CONCEPTION ANALOGIQUE : ETAT DE L'ART	27
III.1	TRAVAUX D'EQUIPES FRANÇAISES	27
III.1.1	<i>Equipe ASIM du LIP6, Université Pierre et Marie Curie Paris VI.....</i>	<i>27</i>
III.1.2	<i>Equipe LEOM de l'Ecole Centrale Lyon.....</i>	<i>28</i>
III.2	TRAVAUX D'EQUIPES EUROPEENNES.....	30
III.2.1	<i>Equipe de l'Université Polytechnique de Madrid</i>	<i>30</i>
III.2.2	<i>Equipe IMSE de l'Université de Seville</i>	<i>31</i>
III.2.3	<i>Equipe de Katholieke Universiteit Leuven (KUL).....</i>	<i>32</i>
III.3	AUTRES TRAVAUX	33
III.3.1	<i>Stockage des connaissances du concepteur</i>	<i>33</i>
III.3.2	<i>Dimensionnement et layout automatiques.....</i>	<i>33</i>
III.3.3	<i>Point de vue sur l'IP analogique réutilisable.....</i>	<i>35</i>
III.3.4	<i>L'utilisation du langage XML</i>	<i>37</i>
III.4	SYNTHESE	37
IV	CONCLUSION.....	39

I INTRODUCTION

La conception de circuits intégrés doit actuellement relever le défi de l'augmentation de la densité d'intégration, de l'évolution des procédés technologiques et de la croissance de la complexité. Ce défi ne pourra être relevé que grâce à l'adéquation des outils et méthodes de conception.

Les systèmes numériques bénéficient aujourd'hui d'outils et méthodes standardisés qui ont permis l'automatisation complète du flot de conception. Il est donc légitime de s'inspirer de ce modèle pour la conception de systèmes analogiques ou mixtes (analogiques et numériques). Cependant la criticité des blocs analogiques freine l'évolution de ce flot de conception AMS (*Analog and Mixed-Signal*). La réutilisation de blocs déjà conçus a donné naissance au concept d'IP (*Intellectual Property*) qui doit permettre de réduire considérablement le temps et le coût de conception des systèmes intégrés.

Si l'IP numérique est aujourd'hui bien connu et largement utilisé, la question de l'IP analogique et mixte, IP-AMS, reste ouverte : comment peut-on définir un IP-AMS réutilisable ? La condition de réutilisation en analogique n'est pas évidente. En effet en analogique le meilleur circuit pour une fonction n'est que rarement le meilleur circuit pour une autre fonction.

Nous décrirons dans une première partie le flot de conception analogique et mixte en étudiant préalablement le flot de conception numérique. Puis nous développerons l'état de l'art concernant la réutilisation de la conception analogique. Enfin nous ferons une synthèse pour identifier les différents problèmes et ainsi proposer des solutions dans les chapitres suivants.

1.1 Historique et perspectives de la conception électronique

L'évolution de l'industrie de l'électronique est très rapide ce qui implique le développement de nouvelles méthodologie pour permettre de suivre cette croissance. La loi de Moore décrit bien cette évolution.

La Loi de Moore a été exprimée en 1965 par Gordon Moore [MOOR 1965], ingénieur de Fairchild Semiconductor, un des deux fondateurs d'Intel. Il expliquait que la complexité des semiconducteurs doublait tous les dix-huit mois à coût constant depuis 1959, date de leur invention. Cette augmentation exponentielle fut rapidement nommée Loi de Moore ou, compte tenu de l'ajustement ultérieur, Première loi de Moore.

En 1975, Moore [MOOR 1975] réévalua sa prédiction en posant que le nombre de transistors des microprocesseurs (et non plus de simples circuits intégrés moins complexes car formés de composants indépendants) sur une puce de silicium double tous les deux ans. Bien qu'il ne s'agisse pas d'une vraie loi physique, cette prédiction s'est révélée étonnamment exacte. Entre 1971 et 2001, la densité des transistors a doublé chaque 1,96 année. En conséquence, les machines électroniques sont devenues de moins en moins coûteuses et de plus en plus puissantes.

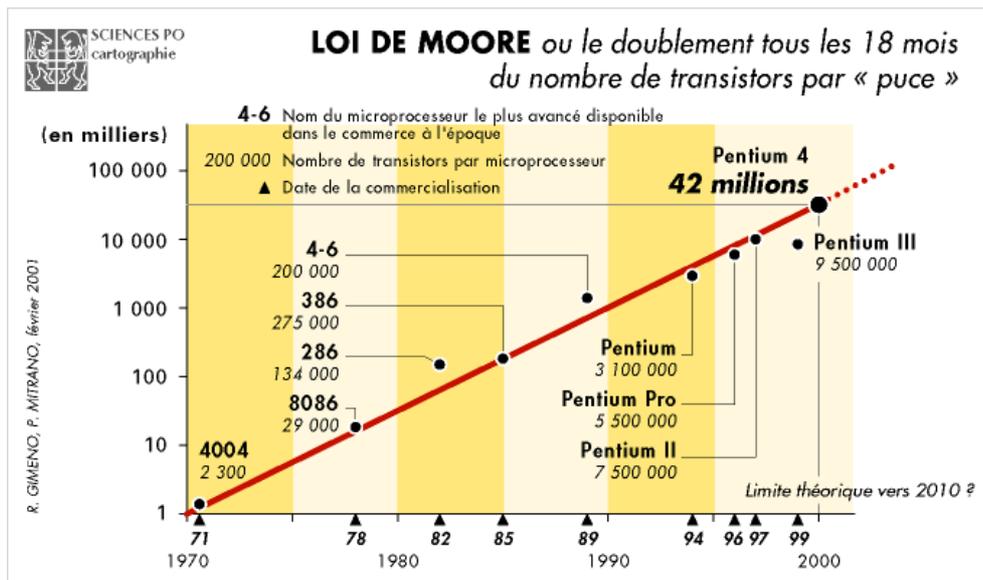


Figure 1.1. Loi de Moore [GIME 2001]

La deuxième loi (figure 1.1) est à peu près vérifiée depuis 1973, mais depuis 2004, elle souffre d'un petit ralentissement dû notamment à des difficultés de dissipation thermique, qui empêche une montée en fréquence en dépit de la taille plus faible des composants.

Cette évolution rapide de la densité d'intégration va de pair avec une augmentation de la complexité de conception des circuits.

1.2 Une complexité grandissante

Le terme de complexité de conception peut signifier plusieurs choses. Le nombre de transistors, la surface, le nombre de personnes contribuant à la conception ou même la pluridisciplinarité peuvent être des facteurs pour évaluer la complexité.

Si nous prenons comme facteur principal le nombre de transistors, nous pouvons nous retrouver avec des incohérences. Prenons comme exemple, la différence entre le temps de conception pour la création d'un mémoire 256-kBytes et d'une mémoire 256-MBytes. Cette différence est minime puisqu'il y a beaucoup d'éléments similaires. Ainsi quantifier la

complexité par le nombre de transistors peut ne pas être pertinent [REIN 2002]. De même, les SoC (*System on Chip*) et plus particulièrement les AMS-SoC possèdent de nombreux éléments numériques et un nombre beaucoup plus réduit d'éléments analogiques et mixtes qui requièrent plus d'effort de conception. Ainsi suivant le nombre d'éléments dans chaque catégorie, la complexité fluctue.

Suivant [GIEL 2000] et [ITRS 2001], les principaux facteurs impliquant une complexité croissante sont :

- L'augmentation du nombre de transistors par circuit
- L'usage de nouvelles architectures de circuits et de nouveaux algorithmes
- L'agrandissement de l'équipe de conception
- L'utilisation de technologies récentes

Bien sûr, le temps de conception augmentera si la complexité du circuit augmente, ce qui pénalise le temps de mise sur le marché du produit final. Ainsi la constante adaptation des méthodes et outils de conception est indispensable pour diminuer le temps de conception.

Parmi les solutions susceptibles de réduire le temps de conception, nous nous intéresserons plus particulièrement à ce qui se rapporte à la *réutilisation*.

1.3 Développement des méthodologies de réutilisation

Selon le thème *Design* de l'International Technology Roadmap for Semiconductors [ITRS 2006], les méthodologies de réutilisation sont en pleine expansion et paraissent inévitables pour les conceptions futures. Comme nous pouvons le constater par le tableau suivant, le pourcentage de blocs réutilisables est en augmentation, de même que les outils de synthèse. Mais nous pouvons constater que même si la synthèse analogique est de plus en plus présente et que des méthodologies de modélisation sont de plus en plus utilisées, cette évolution reste faible et des améliorations sont à apporter pour atteindre le niveau de la conception numérique.

TABLEAU 1.1. 2006 : BILAN ET ESTIMATION DES BESOINS EN CONCEPTION DE CIRCUITS ET SYSTEMES

ANNEE DE PRODUCTION	2005	2006	2007	2008	2009	2010	2011	2012	2013
DRAM ½ DISTANCE (NM)	80	70	65	57	50	45	40	36	32
% DE BLOCS REUTILISES DANS UN SoC	32 %	33 %	35 %	36 %	38 %	40 %	41 %	42 %	44 %
% DE PLATEFORMES DE CONCEPTION AUTOMATIQUE	3 %	6 %	10 %	25 %	35 %	50 %	57 %	64 %	75 %
AUTOMATISATION DE LA CONCEPTION ANALOGIQUE	12 %	14 %	17 %	17 %	24 %	24 %	27 %	30 %	32 %
MATURITE DES METHODES ET OUTILS DE MODELISATION AMS	53 %	55 %	58 %	60 %	62 %	65 %	67 %	70 %	73 %

La moitié de la distance entre les cellules d'une mémoire dynamique RAM est l'étalon pour mesurer l'évolution d'une technologie.

Les pourcentages des lignes *Automatisation de la conception analogique* et *Maturité des méthodes et outils de modélisation AMS*, sont calculés par rapport à l'avancée de ces techniques en comparaison avec celles du numériques. Par exemple, si le degré d'automatisation en analogique devient le même (au niveau des performances et de l'expérience) que celui du numérique, alors ce pourcentage sera de 100 %.

Toutes les cases grisées indiquent que pour le moment aucune solution pour atteindre ce niveau n'a été trouvée.

[ITRS 2006] insiste également sur la nécessité de développer des outils adéquats de vérification car cette étape du flot de conception s'avère la plus importante et consommatrice de temps et cela à chaque niveau de la hiérarchie de conception.

Les progrès technologiques pour la conception électronique dépendent du développement d'outils pertinents qui permettent une vérification des performances et permettant une aide à la réutilisation.

Nous exposerons dans un premier temps l'évolution du flot de conception numérique, puis l'évolution du flot analogique et mixte. Ensuite nous décrirons et discuterons de l'état de l'art de la réutilisation AMS.

II FLOT DE CONCEPTION AMS

La conception de systèmes AMS dépend de l'évolution du flot numérique d'une part et du flot analogique d'autre part. Le flot numérique étant évolué et à un stade d'automatisation avancé, les verrous se retrouvent essentiellement sur l'analogique et sa multitude de contraintes.

De plus, l'environnement du système mixte ajoute des contraintes sur la conception. Plusieurs cohabitations de procédés ou de contraintes peuvent être envisagées pour la conception de systèmes analogiques et mixtes :

- La conception combinée des parties analogiques et logiques
- Les circuits multi-technologiques pour la réalisation de capteurs où cohabitent des blocs physiques (mécanique, chimique, etc.) et des blocs de contrôle électroniques (analogique, numérique).
- La conception combinée de la partie logicielle et matérielle d'un système numérique.
- La prise en compte des contraintes environnementales (température, rayonnement) et leur impact sur la fiabilité.

Chacune de ces mixités nécessite la mise en œuvre de nouveaux outils et de nouvelles méthodologies de conception.

Nous proposons ici dans un premier temps de décrire l'évolution des flots de conception numérique puis analogique. Nous accorderons une attention spéciale à la notion d'IP, principal support de la *réutilisation*. Ensuite nous décrirons les différentes stratégies en cours de plusieurs groupes de recherche pour améliorer le flot de conception analogique.

II.1 Evolution du flot de conception numérique

II.1.1 Description du flot numérique

Actuellement le flot de conception numérique est très développé et automatisé. Premièrement, la profusion d'activités dans le domaine de la conception numérique a eu pour résultat la définition de niveaux d'abstraction clairs et bien définis (comportemental, fonctionnel, RTL : transfert de registres, porte, transistor). Ces abstractions sont des approximations qui fonctionnent, car les interconnexions numériques à tous les niveaux (sauf celui du transistor) représentent des tensions qui pilotent des nœuds à haute impédance, et les blocs du circuit sont, de ce fait, pratiquement indépendants les uns des autres. Deuxièmement des outils de conception automatisée existent et aident le développement et l'évolution du flot de conception numérique, notamment le langage VHDL qui a ouvert la voie à la synthèse automatique.

La synthèse logique (ou numérique) permet à partir d'une description de haut niveau, de concevoir de manière automatique le circuit numérique d'un projet en tenant compte de contraintes comme la vitesse, la surface ou la consommation. Les ressources de base sont des cellules précaractérisées dans la conception d'ASIC (*Application Specific Integrated Circuits*), des structures précalculées dans la conception des FPGA (*Field Programmable Gate Array*). A l'heure actuelle, on peut décrire avec un haut niveau d'abstraction un système numérique de quelques millions de portes grâce à des descriptions HDL (*Hardware Description Language*), tabulaires ou graphiques [LEGA 2005] (schémas, graphes de fluence, réseaux de Pétri).

La figure ci-dessous décrit le flot de conception numérique basé sur la réutilisation de blocs logiciels ou matériels préconçus (IP).

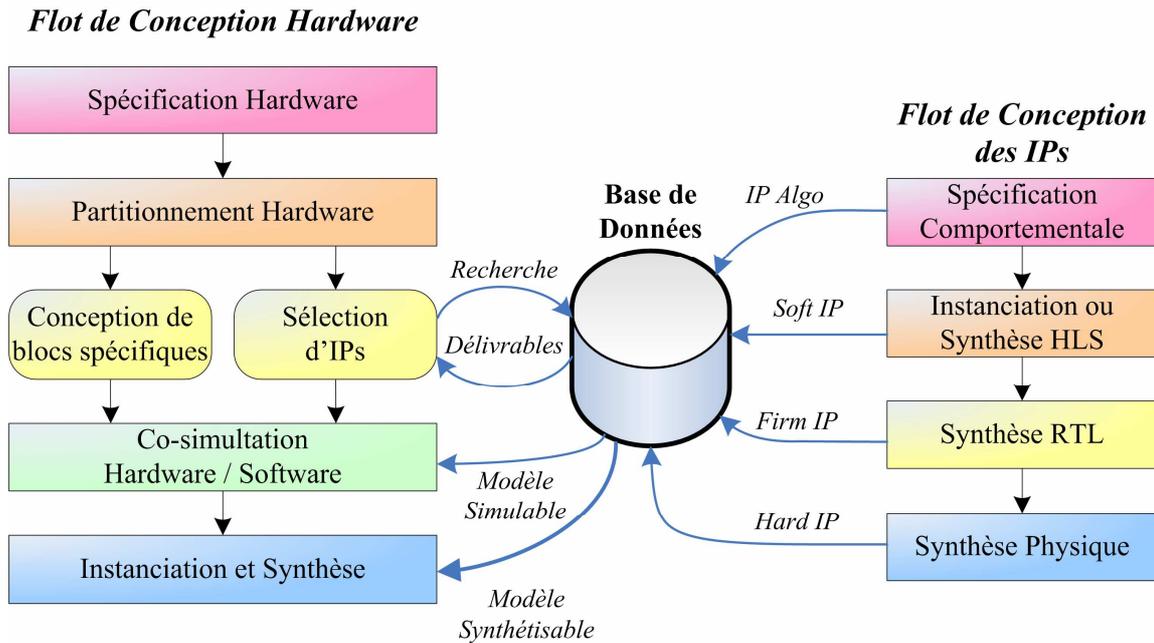


Figure 1.2. Flot de conception numérique à base d'IPs [LEGA 2005]

Cette figure décrit le flot de conception Hardware basé sur la réutilisation et l'assemblage de composants préconçus (IP). L'intégration est validée par un ensemble de simulations. Les IPs ont leur propre flot de conception qui se décompose en plusieurs étapes. Tout d'abord la définition des spécifications, puis la synthèse HLS (*High Level Synthesis*) et la synthèse RTL (*Register Transfer Level*) de ces spécifications avant la synthèse physique. La synthèse HLS est la transformation d'une description de type algorithmique en une description de type transfert entre registres (RTL). Le résultat de cette synthèse est donc une interconnexion de mémoires, de chemins de données et d'automates de contrôle dont le comportement est synthétisable par les outils de plus bas niveau. La synthèse RTL quant à elle, est la transformation d'une description RTL en une interconnexion d'éléments matériels, tout en respectant des contraintes d'optimisation. On définit ainsi les trois types d'IPs numériques : Soft, Firm et Hard.

Un exemple typique de synthèse numérique consiste à spécifier les contraintes temporelles sur les signaux d'horloge, d'entrée et de sortie, et ensuite à trouver une solution qui optimise la vitesse, la puissance ou la surface (où la vitesse est liée implicitement à la puissance). Cette synthèse s'effectue alors automatiquement, grâce notamment à des langages de descriptions adaptés.

II.1.2 Outils existant pour supporter le flot de conception des systèmes numériques

Nous présentons ici deux langages de description utilisés pour la conception de systèmes numériques :

- VHDL [VHDL 1997]
- SystemC [SYST 2003]

II.1.2.1 VHDL

Le VHDL (*Very High speed integrated circuit hardware Description Language*) est un langage de description du matériel destiné à décrire le comportement et/ou l'architecture d'un système électronique numérique. L'intérêt d'une telle description réside dans son caractère exécutable : une spécification décrite en VHDL peut être vérifiée par simulation, avant que la conception détaillée ne soit terminée. En outre, les outils CAO (Conception Assistée par Ordinateur) de synthèse logique permettent de passer directement d'une description fonctionnelle en VHDL à un schéma en portes logiques.

Le but d'un tel langage de description matériel (HDL) est de faciliter le développement d'un circuit numérique en fournissant une méthode rigoureuse de description du fonctionnement et de l'architecture du circuit désirée. L'idée est de ne pas avoir à réaliser un composant réel, en utilisant à la place des outils de développement permettant de vérifier le fonctionnement attendu. Ce langage permet en effet d'utiliser des simulateurs, dont le rôle est de tester le fonctionnement décrit par le concepteur.

Cette description matérielle peut être synthétisée pour obtenir un composant réalisant les fonctions désirées, à l'aide d'éléments logiques concrets (portes logiques, bascules ou registres). Ceux-ci seront implémentés, selon la technologie utilisée, soit directement en portes (dans le cas d'un ASIC), ou en se basant sur les éléments programmables des FPGA.

Le VHDL ayant une double fonction (simulation et synthèse), une partie seulement du VHDL est synthétisable, l'autre existant uniquement pour faciliter la simulation (écriture de modèles comportementaux et des modèles de tests). Pour obtenir du VHDL synthétisable et portable, il est donc nécessaire de se limiter à des constructions simples, dont la transcription en portes et bascules est simple à réaliser. La norme 1076.6 a été initiée pour tenter de définir un sous-ensemble de VHDL « de synthèse ».

Le langage Verilog HDL [VERI 1999], est très similaire au VHDL. Il existe de fait une quasi équivalence entre les deux langages, d'où l'existence de nombreux scripts de traduction de l'un vers l'autre. Le langage VHDL est maintenant le langage de description matérielle

majoritairement utilisé par les entreprises européennes alors que Verilog est souvent préféré de l'autre côté de l'Atlantique.

Cependant le VHDL ne permet pas de faire une conception mixte (*Hardware / Software*) parce qu'il est orienté hardware, de plus le VHDL ne possède que quelques niveaux de conception, il ne peut donc pas gérer la complexité des grandes architectures efficacement.

II.1.2.2 SystemC

SystemC est, comme Verilog et VHDL, un langage de description de matériel. C'est le fruit de contributions de plusieurs sociétés (Synopsys, Frontier Design, CoWare). Les spécifications de SystemC ont été étendues en 2001 à la modélisation de systèmes abstraits (de très haut niveau, avant partitionnement matériel / logiciel).

Ce langage est une extension du langage C++ pour la description des systèmes électroniques. Il représente une bibliothèque enrichie avec laquelle on peut réaliser la modélisation et la simulation des systèmes électroniques. L'utilisation du langage SystemC est basée sur les concepts qu'il peut fournir et l'ensemble des APIs (*Application Programming Interface* : Interface de Programmation) facilitant la modélisation de tels systèmes.

Ces concepts sont décrits ci-dessous :

- La hiérarchie : afin d'aborder la complexité des systèmes et de permettre une modélisation simple et claire, SystemC nous offre le concept de la hiérarchie.
- La concurrence : SystemC nous offre grâce à son simulateur intégré la possibilité d'exécuter en parallèle plusieurs modules (tâches, ou composants).
- La communication : pour que les différents modules puissent communiquer, SystemC propose une modélisation de la communication permettant l'échange de données ou des signaux entre les différents modules.

L'un des inconvénients majeurs des flots habituels vient de la multitude des langages mis en oeuvre (algorithmique, UTF *Unicode Transformation Format*, HDL, etc.). Toutes ces transcriptions manuelles introduisent des erreurs. L'objectif principal de SystemC est de maintenir un seul et même langage d'un bout à l'autre du flot de conception.

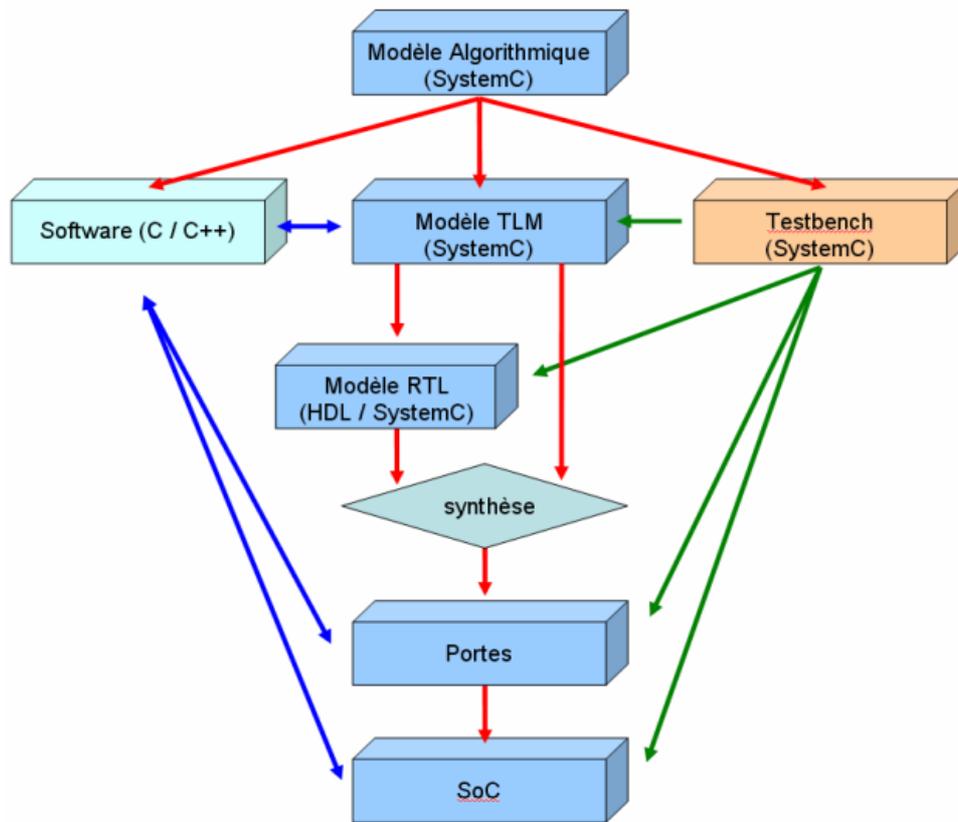


Figure 1.3. Flot de conception SystemC [POLT 2007]

A chaque étape du flot décrit par la figure ci-dessus, c'est-à-dire au fur et à mesure que l'architecture du circuit est raffinée et devient de plus en plus précise, le même environnement de test (écrit en SystemC) peut être utilisé, garantissant la fonctionnalité du système (*Hardware / Software*). Ce n'est qu'à la toute fin du flot (gravure des portes dans le silicium) qu'on traduit, de façon automatique, le SystemC en portes, en passant éventuellement par un autre langage HDL.

II.1.3 IP numérique

La base de la synthèse numérique est l'utilisation d'IPs qui permettent la réutilisation de circuits numériques et ainsi diminuent le temps de conception des systèmes.

Selon Virtual Socket Interface Alliance (VSIA) [VSIA 2001a], [VSIA 2001b] et le guide de méthodologies de réutilisation numérique [KEAT 1999], les blocs IP peuvent être classés dans ces différentes catégories :

- **Soft IP.** Les blocs réutilisables sont donnés sous forme de fichiers synthétisables HDL (Hardware Description Language). Ils ont l'avantage d'être flexible mais ils sont moins performants et n'ont pas une surface optimisée.

- **Hard IP.** Ces blocs sont optimisés en terme de performance, de puissance et de surface mais sont dépendant d'une technologie. Par exemple, ce sont des netlists avec des layouts spécifiques à une technologie. Ces IPs sont plus performants mais ne sont pas flexibles. Le redimensionnement de ce type d'IP n'est pas faisable, et une recherche intensive se poursuit sur ce sujet [KEAT 1999].
- **Firm IP.** Ce sont des blocs réutilisables qui ont été optimisés au niveau de la structure et de la topologie pour de meilleures performances. Ils ont été créés en utilisant une bibliothèque de technologies génériques, ce qui leur donnent une meilleure flexibilité par rapport aux Hard IPs. Les Firm IPs offrent un compromis entre les Hard IPs et les Soft IPs. En effet, ils sont plus flexible que les Hard IPs et possèdent de meilleures performances et une meilleure surface que les Soft IPs.

Le tableau suivant 1.2 résume les performances de ces différents IPs.

TABLEAU 1.2. CLASSIFICATION DES IPs NUMÉRIQUES

IP	Performance	Flexibilité	Migration des données
Soft	Moyenne	Très Grande	Illimitée
Firm	Grande	Grande	Moyenne
Hard	Très grande	Aucune	Très difficile

Concernant toujours les IPs numériques, Pierre Bricaud [BRIC 1998] définit les différentes méthodologies de réutilisation en numérique, tout comme [GOER 1999]. Ces méthodologies de conception reposent sur la réutilisation de blocs logiciels ou matériels préconçus (IP), et sur l'élévation du niveau de description des applications.

Le flot de conception numérique peut se décliner en trois familles [CAI 2003] :

- La synthèse à partir d'une spécification système fonctionnelle (*Synthesis from specification*). Les spécifications décrivent le fonctionnement du système mais ne donnent pas d'informations sur l'implémentation.
- La conception s'appuyant sur l'utilisation de plate-forme (*Platform-Based Design*). Ces plate-formes sont des combinaisons spécifiques de composants dédiés à certaines applications. Ils peuvent contenir des IPs.
- La conception basée sur la réutilisation de composants (*Component-Based Design* ou *IP assembly*). Le système est conçu à partir d'IPs.

Les méthodologies *Platform-Based Design* et *Component-Based Design* vont nous aider à développer nos propres méthodologies pour la conception analogique.

Comme nous venons de le voir, le flot de conception numérique est très avancé et largement automatisé contrairement au flot analogique et mixte que nous décrivons dans le prochain paragraphe.

II.2 Evolution du flot de conception analogique et mixte

Le CNRS organise des actions spécifiques (AS) de perspectives concernant plusieurs thèmes. En particulier, nous nous intéressons à l'AS SoC-AMS créée en juillet 2002 par P. Garda et P. Fouillat. Ses objectifs sont de faire l'état de l'art du domaine des SoC-AMS aux niveaux académique et industriel, mais aussi de déterminer les verrous dans ce domaine et de proposer des actions de recherche. Le thème qui nous intéresse *Méthodes et outils pour la conception*, regroupe les synthèses de P. Desgreys, M.M. Louërat, Y. Hervé, N. Lewis et I. O'Connor. Quatre sous-thèmes ont été dégagés pour cette recherche : Flot de conception AMS, Modélisation comportementale, Bibliothèque d'IP-AMS et Synthèse analogique. Nous reprendrons une partie des conclusions du workshop final 2004 en rapport avec la problématique de cette thèse.

Une description détaillée du flot de conception est effectuée [DESG 2004] permettant de passer en revue les outils logiciels sur lesquels repose la conception actuelle des systèmes analogiques et mixtes, mais aussi de mettre en évidence les lacunes de ce flot et de fixer les objectifs généraux. La modélisation comportementale est analysée [LEWI 2004] et trouve sa place dans le flot de conception car elle constitue un premier pas vers la gestion de la complexité. Les bibliothèques d'IP-AMS sont étudiées par Y. Hervé [HERV 2004] et apparaissent comme le pivot autour duquel devrait s'articuler le flot de conception AMS idéal. Concernant la synthèse automatique [O'CON 2004], elle est étudiée par Ian O'Connor dont on décrira le groupe de recherche.

II.2.1 Description du flot analogique et mixte

La conception analogique est décomposable en deux grandes phases qui sont la conception électrique du circuit au niveau schématique et la conception physique du circuit au niveau dessin des masques (layout).

II.2.1.1 Conception Electrique

La conception électrique s'effectue idéalement selon une approche Top-Down et se décompose en trois étapes :

- Description du système et de sa hiérarchie
- Définition et liens hiérarchiques de chaque bloc fonctionnel
- Dimensionnement des cellules

La première étape est la description du système et de sa hiérarchie, qui consiste en la définition des entrées, des sorties et de leurs relations. Cette définition peut être effectuée par un modèle écrit dans un langage de haut niveau (par exemple, le langage C). Puis, on choisit l'architecture du système qui permet une décomposition du système en blocs fonctionnels. Ces blocs possèdent des interconnexions hiérarchiques et des spécifications résultant du système complet. L'automatisation de cette étape n'est possible que pour les systèmes dont l'architecture est fixée. Des difficultés s'ajoutent pour les systèmes mixtes, en effet, il est indispensable de réfléchir à la répartition des tâches entre le traitement analogique et numérique du signal. Enfin, une vérification fonctionnelle de l'architecture est effectuée.

La deuxième étape est la définition de chaque bloc. Ceci permet d'identifier chaque bloc par sa fonction et donc de le distinguer de son environnement. Pour la partie numérique, le développement de ces blocs (mémoires, filtres, DSP, etc.) sera facilité par des outils de synthèse automatiques existants. Pour la partie analogique, étant donné que des outils de synthèse n'existent pas encore, ces blocs (PLL, convertisseurs A/N et N/A, etc.) seront validés par une simulation comportementale en utilisant des langages dédiés aux circuits mixtes comme le VHDL-AMS [VHDL 1999] ou le Verilog-AMS [VERI 2004].

La troisième étape est le dimensionnement des cellules pour la partie analogique. Les cellules sont définies comme étant des fonctions de base, avec une complexité réduite permettant de concevoir des blocs comme des amplificateurs ou des oscillateurs. Le dimensionnement de ces cellules consiste en la détermination des tailles des composants (transistors, capacités, résistances) de chaque cellule pour atteindre les performances souhaitées. Les méthodes de dimensionnement sont classées en deux catégories : celles utilisant le savoir-faire du concepteur et celles utilisant des fonctions d'optimisation. Les cellules dimensionnées sont ensuite validées par des simulations électriques à partir de netlists et d'un calculateur de type SPICE. Pour le cas particulier de blocs mixtes, les outils de simulations pour les circuits mixtes sont alors utilisés.

II.2.1.2 Conception Physique

La création du layout s'effectue en trois étapes :

- Génération des masques des cellules. On utilise la netlist dimensionnée de la cellule et les informations additionnelles sur les éléments parasites, l'appariement et les performances attendues. Généralement ce travail est effectué manuellement mais certaines équipes l'automatisent partiellement selon différentes méthodes d'optimisation.
- Création des masques au niveau des blocs fonctionnels. Quelques outils dédiés à des applications bien définies ont été développés pour automatiser cette étape [JANG 2003].
- Placement et routage des différents blocs qui composent le système pour obtenir les masques du système complet.

Une vérification des performances au niveau électrique pour les cellules, au niveau comportemental pour les blocs et au niveau fonctionnel pour le système est effectuée après chaque étape, selon une approche Bottom-Up.

Etant donné la complexité du système initial, il est indispensable de le décomposer en sous-systèmes. Les notions de hiérarchisation d'un système et de blocs sont essentielles.

La figure 1.4 résume la conception d'un système analogique et mixte. Les méthodologies Top-Down et Bottom-Up sont représentées.

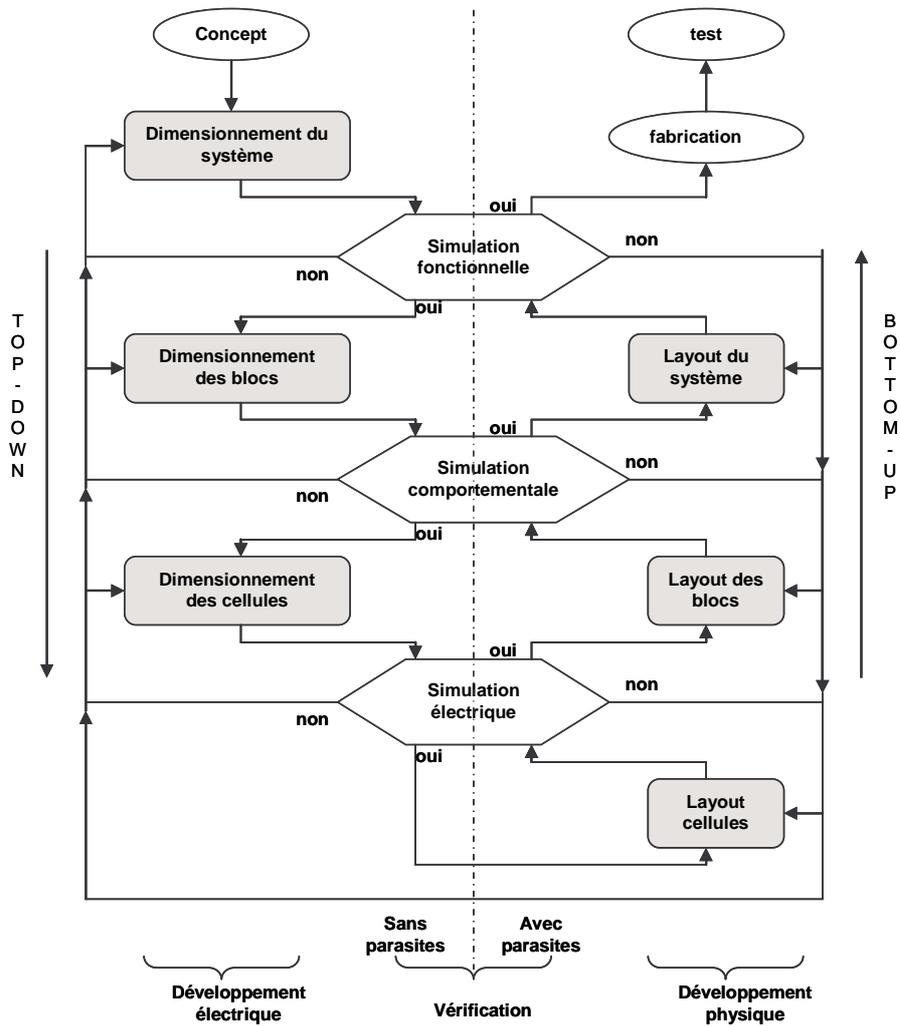


Figure 1.4. Flot de conception d'un circuit mixte [DESG 2004]

II.2.2 Outils existant pour supporter le flot de conception des systèmes AMS

Il s'agit d'outils de spécification, de simulateurs « système », de simulateurs « circuit », d'outils de layout, d'outils d'optimisation, des interconnexions entre ces outils, ...

TABLEAU 1.3. DIFFERENTS OUTILS DU FLOT DE CONCEPTION [DESG 2004]

Type d'Outils	Exemples d'outils	Degré d'utilisation de 1 à 5	Automatique / Manuel	Automatisation possible ou à améliorer
Outils de Spécification	MATLAB, C			
Simulateurs Systèmes	MATLAB SIMULINK	5		
	Agilent PTOLEMY pour les systèmes RF (Tuner Satellite ou Cable)	5		
	Simulateurs basés sur VHDL-AMS : ADVanceMS	3		
	Simulateurs basés sur VerilogAMS : AMSdesigner	0		
Simulateurs Circuits	Simulateurs électriques classiques de type SPICE (ELDO, SPECTRE, ADS...) avec analyses DC, AC, TRAN, RF, NOISE et simulations statistiques de type Monte Carlo	5		
	Simulateurs SPICE accélérés basés sur des Look-up table et adaptés à la simulation de mémoires, de circuits digitaux (HSIM, Nanosim)	5		
	Simulateurs VHDL-AMS ou VerilogAMS pour pouvoir mélanger différents niveaux de description (VHDL-AMS + SPICE + VHDL)	3		
Outils de Layout	Editeur de polygone (Cadence, Mentor)	5	M	
	Outils Full-Custom CADENCE basés sur Virtuoso-XL (Virtuoso Custom Placer et Virtuoso Cutom Router)	3	M+A	X
	NeoCell de Cadence/Neolinear : générateurs de structures complexes, placement et routage automatique	1	M+A	X
Outils d'optimisation	Recherche d'une topologie : BarcelonaDesign	1	A	
	Recherche des paramètres de composants à partir d'une topologie donnée : NeoCircuit	2	M+A	X
Outils de modélisation comportementale de blocs analogiques	Optimisation des paramètres des équations d'un modèle Génération de polynômes à l'aide de régressions Génération de tables de données	2	M	X
	Analyse Symbolique : SYMBA	0	M	X
Outils de caractérisation et de génération de datasheet	Applicables à la caractérisation d' IP et au « calibrage » de modèles Antrim ACV, Mentor EldoDesignToolBox (EDTB)	2	M+A	X
Outils de vérification	Mentor Calibre	5	A	
Outils d'extraction	Synopsys Arcadia and StarRC XT	5	A	
Outils d'analyse des bruits du substrat	Simplex SubstrateStorm	2		

Ce tableau présente les différents outils utilisés aux différentes étapes du flot de conception Top-Down puis Bottom-Up. Il décrit aussi les degrés d'automatisation de ces étapes. On peut d'ailleurs remarquer qu'il existe très peu d'outils entièrement automatisés ce qui souligne la différence avec les outils numériques.

Intéressons-nous maintenant à la synthèse analogique très en retard par rapport à la synthèse numérique.

II.2.3 Synthèse analogique

La synthèse analogique [O'CON 2004] pour les circuits intégrés est un domaine de recherche qui existe depuis une vingtaine d'années. Elle représente un défi scientifique crucial et correspond à un réel besoin de l'industrie semi-conducteur pour accélérer le cycle de conception, réutiliser les techniques de conception, faciliter la migration entre technologies et améliorer la robustesse des systèmes. La synthèse analogique comporte essentiellement deux étapes : la sélection de topologie, et l'étape de dimensionnement. Les défis de la synthèse analogique résident dans le nombre de degrés de liberté et le nombre de compromis complexes.

Les outils de synthèse numérique possèdent une avance considérable par rapport à ceux de la synthèse analogique et mixte. Même si en moyenne la partie analogique dans un circuit représente seulement 10 % de la surface, son temps de conception est le plus important. Comment expliquer cette différence ?

Tout d'abord, la conception analogique ne possède pas une hiérarchie fixe étant donné qu'un bloc fonctionnel analogique peut être utilisé dans plusieurs niveaux de hiérarchie. De plus, les blocs analogiques sont très sensibles aux variations des impédances d'entrée et sortie des blocs associés. Nous pouvons aussi constater que les spécifications d'un bloc analogique sont très nombreuses (gain, bande-passante, marge de phase, etc.) et interdépendantes contrairement à celles des blocs numériques. L'espace de conception est donc bien plus complexe pour les blocs analogiques. Les variables de conception sont également différentes : la conception numérique utilise des bibliothèques de cellules standards afin de réaliser la fonction désirée alors que la conception analogique choisit la topologie et ajuste les tailles de chaque transistor. Les variations des paramètres technologiques et de l'environnement jouent aussi un grand rôle dans la difficulté de la synthèse analogique. De légères différences dans le procédé de fabrication, la polarisation, la température ou les composants parasites du layout peuvent grandement influencer les performances du circuit. Cette sensibilité est moins marquée dans les circuits numériques grâce à la dynamique importante des signaux et l'insensibilité au bruit.

Une grande partie de la conception de circuits analogiques et mixtes se fait en utilisant une approche spécifique, où le dimensionnement des composants s'effectue individuellement, et les performances sont optimisées. Ceci donne au concepteur une flexibilité maximale dans sa recherche du circuit optimal satisfaisant à toutes les spécifications.

Deux techniques de synthèse existent :

- Une technique basée sur l'optimisation utilise des méthodes numériques (par exemple : les outils ANACONDA, OPASYN et ANANAS)
- Une technique basée sur les connaissances utilise des procédures explicites de conception (par exemple : les outils PROSAIC, BLADES et IDAC)

Quels sont les avantages et les défauts de chacune de ces méthodes ? La première approche (optimisation) permet la conception sans connaissances particulières, mais nécessite un temps CPU (Central Process Unit) considérable pour la mener à bien. Cet inconvénient majeur est réduit par les méthodes à base de connaissances (procédure). Cependant la génération de procédures explicites de conception est une tâche de longue durée et la dépendance en une topologie est importante. Plus récemment, et en ne considérant pas les deux solutions comme mutuellement exclusives, des outils sont apparus qui mixent les approches à base d'optimisation et à base de connaissances afin de tirer les avantages des deux : des outils industriels (Antrim-MSS, NeoLinear-NeoCell) et académiques (ANACONDA, VASE). Nous proposerons une méthodologie de redimensionnement avec optimisation dans le chapitre 5.

Comme nous avons défini auparavant l'IP numérique, nous allons maintenant présenter la conception à base d'IPs analogiques même si à l'heure actuelle il n'existe pas de définition standard pour l'IP.

II.2.4 Conception analogique à base d'IPs : principe et difficultés

Nous présentons dans cette partie la conception analogique à base d'IPs. Ce flot de conception nous intéresse particulièrement et nos méthodologies traitées ultérieurement se rapporteront à ce principe de flot.

II.2.4.1 Principe de la conception de système à base d'IPs

Cette conception particulière existant en numérique, permet une meilleure réutilisation. Son principe est d'utiliser et de réutiliser des blocs déjà conçus, les IPs.

La figure 1.5 représente les trois étapes essentielles de ce flot.

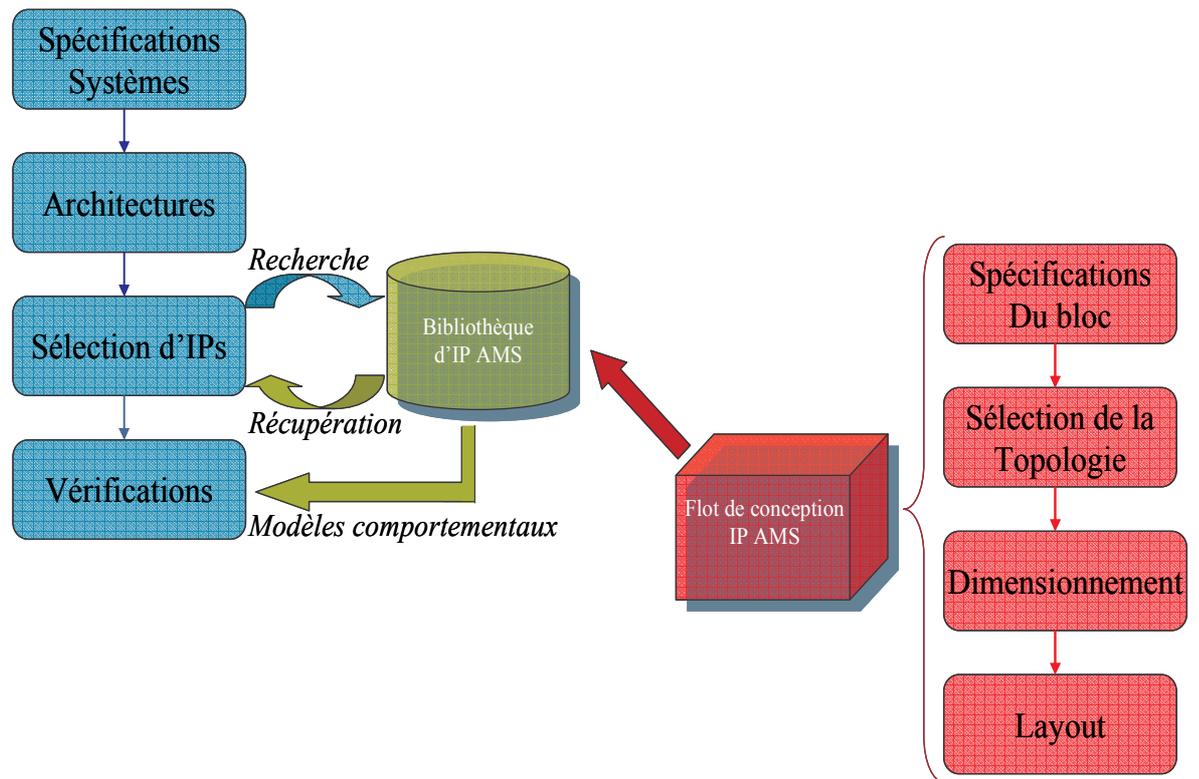
Flot de conception système

Figure 1.5. Illustration du flot de conception à base d'IPs

La première étape est le flot de conception d'IP-AMS. Elle concerne la génération de blocs analogiques et mixtes tels que l'OTA par exemple. Les différentes étapes de la création de nouveaux IPs sont les spécifications du bloc, la sélection de la topologie, le dimensionnement (ou le redimensionnement) et la génération du layout. Cette première étape permet de peupler la bibliothèque d'IPs.

La deuxième étape est la création d'une base de données, qui ici se nomme bibliothèque d'IP-AMS. Tous les IPs conçus lors de la première étape, sont ensuite caractérisés et modélisés. La hiérarchie du système est définie de même que tous les liens entre les différentes fonctions du système. Les IPs sont ensuite classés par leur fonction et leur niveau hiérarchique.

La troisième étape est le flot de conception système dont une des étapes est l'exploration de cette bibliothèque d'IPs pour concevoir le système. A partir des spécifications du système, des requêtes sont créées pour explorer cette base de données et les IPs sont choisis selon leur capacité à répondre au mieux à ces spécifications. Le système est ainsi conçu à partir de ces IPs. Une vérification de la solution proposée est alors effectuée en utilisant les modèles comportementaux des IPs.

Notre travail s'appuie sur ce flot de conception car nous présenterons dans les chapitres 3 et 5, nos méthodologies pour ces trois étapes.

Cependant toute cette méthodologie dépend de la définition et du contenu d'un IP en analogique.

II.2.4.2 Difficultés relatives à l'IP analogique

La propriété intellectuelle (IP) analogique peut-elle être exploitée de la même façon que l'IP numérique ?

Comme nous l'avons introduit précédemment, il existe un besoin fort en solutions pour l'automatisation de la conception analogique. D'autre part, le nombre d'ingénieurs possédant des compétences en conception analogique augmente moins vite que le nombre de CI (Circuits Intégrés) utilisant des composants analogiques, RF et signal mixte. L'IP analogique pourrait-il contribuer à résoudre ce problème ?

Dans un SoC, les blocs analogiques sont sensibles à la disposition, à l'harmonisation des dispositifs, à la symétrie, aux gradients thermiques, aux variations de signal, aux niveaux zéro et à la proximité de circuits de commutation. Avec un nombre aussi important de paramètres sensibles, l'utilisateur ou le fournisseur d'IP a la responsabilité de garantir le bon fonctionnement du bloc circuit dans un SoC. La direction la plus sûre est d'exploiter les propriétés uniquement suivant le même processus et pour les mêmes applications spécifiques.

La notion d'IP analogique a fait grand débat [OHR 2002], [KOCH 2003]. De très nombreux articles ont été écrits spécialement dans EE Times concernant le futur, les possibilités et les rêves concernant l'IP analogique et l'automatisation du flot de conception [SANC 2001], [NAJI 2003], [OHR 2003], [GOER 2003], [FULL 2003], [MCGR 2005], [SING 2005], [GOER 2006] et [VAND 2003].

Bien sûr, une définition claire d'un IP analogique est le point de départ d'un flot de conception automatique en analogique. Les définitions des IPs numériques ne peuvent être appliquées en analogique. Les raisons sont :

- Généralement un bloc analogique est conçu pour une application particulière, et il est très difficile d'utiliser ce bloc pour une autre application : des blocs génériques sont difficile à concevoir.
- Le redimensionnement d'un circuit analogique est complexe. En effet, trop de paramètres sont influents pour un circuit analogique. Changer de technologie

implique en général une grande variation de performance du circuit [GILB 2002], [DESS 2001].

- L'utilisation d'un bloc analogique sans connaître ses limites et son contexte est difficile. Sans un guide du bloc avec une caractérisation précise la réutilisation n'est pas possible [HAMM 2004].
- La réutilisation d'un circuit analogique dans un environnement physique différent peut affecter ses performances.
- Les effets parasites du layout sur les performances est important, le fait de changer de spécifications ou de technologies changent les valeurs de ces effets et ainsi changent les performances du circuit.
- La hiérarchie, du système au composant, n'est pas clairement identifiée en analogique. Par conséquent la synthèse devient très difficile.

Lors du rapport final de l'AS SoC-AMS du sous-thème *Bibliothèque d'IPs*, Y. Hervé propose différentes pistes concernant la définition et le contenu d'un IP. Ces IPs doivent contenir des modèles de haut niveau (comportementaux) pour l'étude architecturale et de différents niveaux de description pour le processus de conception. Lors de la création d'un IP, il est important de le caractériser en terme de réutilisation pour paver l'espace des spécifications en vue de l'exploration de la bibliothèque d'IPs. Enfin le format de la bibliothèque doit pouvoir faire interagir différentes sources, différents moyens de description. Une méthode de méta-description peut permettre cette approche, nous pouvons citer ici des langages prometteurs comme UML et XML. Toutes ces conclusions nous ont permis d'éclaircir notre projet et de concevoir des méthodes et outils d'exploration appropriés au flot de conception analogique.

Nous allons maintenant décrire les travaux de différentes équipes concernant la définition d'un IP analogique et l'amélioration du flot de conception analogique basée sur la réutilisation.

III REUTILISATION DE LA CONCEPTION ANALOGIQUE : ETAT DE L'ART

Nous présentons dans cette partie les travaux des différentes équipes qu'elles soient françaises ou étrangères. Puis nous effectuerons une synthèse de ces travaux pour une meilleure compréhension.

III.1 Travaux d'équipes françaises

III.1.1 Equipe ASIM du LIP6, Université Pierre et Marie Curie Paris VI

Le laboratoire LIP6 et plus particulièrement l'équipe ASIM de Marie Minerve Louërat de l'Université Paris VI, développent une plateforme de conception basée sur la réutilisation. Cette plateforme est développée depuis plusieurs années et elle est le fruit d'une expérience mûrie au cours de plusieurs thèses [NGUY 2006]. Un langage a été créé, dont les bases proviennent du langage C++, qui permet de décrire des générateurs de fonctions analogiques prenant comme entrées les spécifications et la technologie. Ce langage se nomme CAIRO+.

Le langage CAIRO+ est un support formel permettant d'implémenter une méthodologie de conception pensée en 4 étapes. La 1^{ère} phase (*CREATE*) correspond à la définition du schéma électrique générique non-dimensionné ; la 2^{ème} (*DSES*) procède au dimensionnement du circuit sous contraintes fonctionnelles et pour une technologie cible. S'ensuivent les 2 dernières étapes de placement (*PLACE*) et routage (*ROUTE*), tandis qu'une boucle de retour permet de mesurer l'influence des capacités parasites du routage et d'ajuster en conséquence le dimensionnement. En premier lieu, le concepteur est invité à décrire hiérarchiquement son circuit ou module, pouvant lui-même contenir des sous-modules jusqu'aux dispositifs élémentaires, tout en donnant une indication sur le placement relatif des sous-modules. C'est dans l'étape dite *DSES* (*Design Space Exploration and Sizing*) que se concentre l'essentiel de l'expertise du concepteur ; l'effort de formalisme est particulièrement délicat et pourtant crucial pour le succès de l'outil. Le langage CAIRO+ réutilise l'expérience d'un projet antérieur, CAIRO, ayant fait l'objet de la thèse de M. Dessouky, soutenue en 2001 [DESS 2001], et portant principalement sur la génération du dessin des masques.

Il permet aussi de créer une architecture complexe en utilisant des générateurs de fonctions élémentaires, aidant ainsi le flot de conception analogique et mixte. Le papier [DESS 2000] présente une méthodologie de conception d'un modulateur $\Sigma\Delta$ du troisième ordre, en partant du haut niveau jusqu'au layout (méthodologie Top Down). Ce modulateur a été conçu sous deux technologies différentes pour démontrer la flexibilité de cette méthodologie pour les circuits AMS. De plus, les connaissances acquises pour la conception du modulateur $\Sigma\Delta$ du

troisième ordre ont été réutilisées avec succès pour la conception d'un modulateur $\Sigma\Delta$ du quatrième ordre. La synthèse d'un IP amplificateur analogique CMOS avec CAIRO+ a aussi été effectuée [ISKA 2005].

L'équipe ASIM continue actuellement ses travaux sur la plateforme CAIRO+. Toute la conception est basée sur la notion de générateur de fonction, cette fonction ensuite appelée dans un système, ressortira un circuit dimensionné, et son layout associé. Ce type de générateur se rapproche d'avantage du *Soft IP* numérique ; il permet de créer, selon les paramètres d'entrée, un *Hard IP* dans une technologie donnée. L'expérience du concepteur est, en grande partie, concentrée dans la formulation des règles de dimensionnement.

En résumé, l'équipe ASIM propose un outil efficace pour automatiser le flot de conception d'IP-AMS.

III.1.2 Equipe LEOM de l'Ecole Centrale Lyon

L'équipe du groupe Conception du LEOM de l'Ecole Centrale de Lyon dirigée par Ian O'Connor étudie particulièrement les méthodes de conception de circuits hétérogènes [O'CON 2005a]. En utilisant le concept d'IP-AMS, le travail s'effectue au niveau architectural, en manipulant les blocs et non les composants élémentaires. Cette équipe a défini une structure informatique qui formalise le contenu de l'IP-AMS, et a développé une plateforme de cosynthèse (RuneII) permettant la construction de ces blocs et leur paramétrage par exploration architecturale. Cette plateforme ouvre la voie à la création et la population de bibliothèques IP-AMS (modèles et méthodes de conception).

La figure 1.6 représente le flot de conception des *Soft* et *Hard IP*. Les spécifications sont données au niveau le plus haut. Le choix de la topologie et la création de modèles comportementaux sont ensuite effectués. La topologie est alors dimensionnée et le layout associé est créé ainsi que le modèle extrait. Enfin une aide à la documentation permet la création d'une documentation (*datasheet*) en vue de la réutilisation.

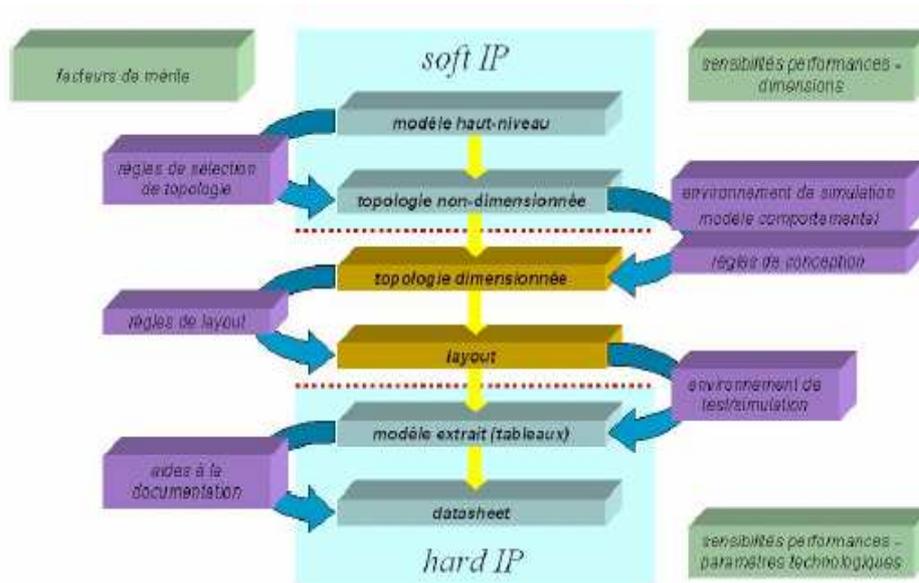


Figure 1.6. Flot de conception des *Soft IP* et *Hard IP* [O'CON 2005a]

Une définition d'un IP-AMS est donnée [O'CON 2005b] dont le contenu peut se résumer ainsi :

- Définition de la fonction : classe de fonctions (équivalent VHDL : entity, behavioral architecture)
- Critère de performances : spécification et évaluation du bloc IP (équivalent VHDL : generic)
- Terminaux : entrées/sorties (équivalent VHDL : terminal)
- Structure : structure interne de l'IP (équivalent VHDL : structural, architecture)
- Variables : liste des variables utilisées pour la conception ou les algorithmes d'optimisation (équivalent VHDL : subset of generic map)
- Paramètres physiques : liste des paramètres physiques associés aux composants internes (équivalent VHDL : generic map)
- Méthode d'évaluation : code définissant le fonctionnement de l'IP, transformation des paramètres physiques en critères de performance (équivalent VHDL : (partly) process or procedure)
- Méthode d'extraction des paramètres : code définissant la méthode d'extraction de critères de performances à partir des résultats de simulations

- Méthode de synthèse : code définissant comment synthétiser l'IP et transformer les critères de performances en valeurs des variables du design
- Méthode de distribution des contraintes [MALA 1999] : code définissant comment transformer les paramètres de l'IP en spécifications à un niveau hiérarchique plus bas

La plateforme RuneII permet la création de blocs IP, et décrit aussi des méthodes génériques de dimensionnement. La méthodologie associée est composée d'une exploration descendante de l'espace de conception, puis d'un dimensionnement physique exploitant directement les outils d'évaluation, et enfin de la vérification ascendante en utilisant l'extraction de paramètres de modèles. Du côté de l'évaluation, la plateforme est capable, par son interface générique d'évaluation, d'employer la simulation multi-domaine comportementale afin d'optimiser les structures hétérogènes. Cette plateforme utilise les langages UML, Java et XML qui sert uniquement au stockage des informations des IPs.

Ainsi cette équipe a développé des méthodologies et des outils relatifs au flot de conception analogique dans son ensemble : synthèse de circuits, utilisation d'une bibliothèque d'IPs, conception de systèmes.

III.2 Travaux d'équipes européennes

III.2.1 Equipe de l'Université Polytechnique de Madrid

L'Université Polytechnique de Madrid effectue également des recherches sur les méthodologies de conception analogique et mixte, illustrées notamment par le projet européen TOOLIP démarré en 2003 [MART 2003] [SEEP 2002]. Ce projet tente d'élaborer des méthodes et outils permettant de réduire le temps de conception et la complexité du système, de faciliter la simulation et les tests ainsi que la réutilisation.

Deux thèmes principaux peuvent être dégagés : l'optimisation du flot de conception système et la définition d'un IP analogique.

Pour améliorer la réutilisation pour le flot de conception analogique et mixte, cette équipe, en collaboration avec l'IMSE de l'Université de Séville et du laboratoire FZI de Karlsruhe avance quelques propositions [MART 2001] :

- Utilisation d'une méthodologie Top Down, le système doit être décrit avec des modèles comportementaux des composants au niveau du système.

- Une librairie de modèles des composants doit être disponible et cela à chaque niveau. Un modèle au niveau système et des modèles au niveau bloc possédant des propriétés comme la non linéarité etc.
- Une vérification bottom-up. Des informations sont rajoutées au niveau comportemental pour une meilleure prise en compte des caractéristiques de l'implémentation physique.
- Création de nouveaux outils et plateformes pour faire fonctionner ces méthodologies de réutilisation.

Pour l'application de ces propositions, un IP et son contenu minimal doit être défini.

Une définition de cet IP [GINE 2002] est donnée par A.J. Ginés en collaboration avec l'IMSE. L'IP contient un modèle comportemental en plus du layout et du schéma (netlist) pour diminuer le coût et le temps de vérification. Ils définissent ainsi un *Hard IP* si l'on reprend la terminologie numérique. Des recherches sur la création de bibliothèque de modèles comportementaux pour les circuits AMS ont commencé. Ils utilisent un modèle comportemental générique lors de la phase de conception Top-Down. De même ils ajoutent des paramètres dans ce modèle, tels les effets non idéaux, lors de la phase de vérification Bottom-Up.

Cette équipe a donc présenté des méthodologies de conception de système analogique ainsi qu'une définition d'IP.

III.2.2 Equipe IMSE de l'Université de Seville

Le laboratoire IMSE de l'Université de Séville effectue des recherches sur les méthodologies de réutilisation et sur le développement d'outils pour la conception de circuits analogiques et mixtes. R. Castro-Lopez [CAST 2000], [CAST 2002], [CAST 2006] a effectué plusieurs travaux sur ce sujet. Il y décrit une méthodologie de conception de circuits analogiques basée sur la réutilisation en donnant des définitions claires de blocs analogiques réutilisables.

Ces blocs analogiques sont définis en trois vues : une vue comportementale, une vue structurelle et une vue layout. Les deux premières vues sont utilisées pour la conception Top-Down et pour la vérification Bottom-Up, et la troisième est utile pendant la synthèse physique. Cette définition de bloc analogique est comparable à un *Firm IP* analogique en reprenant la terminologie numérique.

Une méthodologie de conception à base de blocs réutilisables a été développée. Ce flot facilite l'implémentation de blocs AMS réutilisables ce qui réduit le temps de conception. La hiérarchie du système est définie en 4 niveaux : transistor, cellule, module et système. Les blocs analogiques réutilisables appartiennent au niveau cellule. Une approche Bottom-Up est alors utilisée pour concevoir les niveaux hiérarchiques supérieurs.

Un outil de génération automatique de layout a aussi été développé. Il se décompose en deux étapes. La première, appelée étape géométrique, détermine la surface du layout et la taille des composants en utilisant des boucles d'optimisation. Cela consiste en la détermination des paramètres du layout tels que sa surface soit optimisée. La deuxième étape, appelée ajout de parasites, consiste à ajouter tous les effets parasites dépendants du placement des composants afin de maintenir les performances souhaitées de la cellule.

L'équipe de l'IMSE a notamment traité comme exemple d'applications, des amplificateurs opérationnels pour les blocs réutilisables au niveau cellule. Un système, IQ digital-to-analog, a été développé en utilisant les méthodologies et les outils créés.

En résumé, cette équipe a développé des méthodologies et des outils de réutilisation relatifs au flot de conception analogique et mixte dans son ensemble : synthèse de circuits, utilisation de blocs réutilisables, conception de systèmes.

III.2.3 Equipe de Katholieke Universiteit Leuven (KUL)

L'équipe de G. Gielen [GIEL 2000] de Katholieke Universiteit Leuven (KUL) propose des outils de conception :

- l'outil générique AMGIE [VAND 2001]; il permet un dimensionnement automatique, et annonce des performances élevées (du cahier des charges au layout en 20 minutes).
- CYCLONE [DERA 2000], un générateur de selfs pour oscillateurs. Le layout est conçu automatiquement. Cet outil utilise les modèles au niveau transistor avec un algorithme d'optimisation.
- DAISY [FRAN 2000], un synthétiseur de modulateurs sigma delta : sélection de topologie, exploration architecturale suivie d'optimisation à base de simulation

L'équipe de G. Gielen a donc développé plusieurs outils concernant la synthèse automatique de circuits du domaine RF (Radio Fréquence) mais aussi concernant la synthèse de systèmes spécifiques (modulateur $\Sigma\Delta$).

III.3 Autres travaux

D'autres articles ont été recensés car ils relatent des travaux ponctuels concernant notre problématique. Ces différents travaux sont classés par thèmes.

III.3.1 Stockage des connaissances du concepteur

L'article [LIU 2003] décrit un système qui permet de stocker les connaissances du concepteur comme une part des caractéristiques du circuit, ce qui permet une réutilisation plus facile. Le logiciel STAR (Schematic Tool for Analog Reuse) permet de connaître les paramètres du circuit mais aussi les contraintes imposées par l'environnement et la technologie. Ce logiciel aide aussi l'utilisateur dans le choix des composants pendant le design. Il permet en outre d'inscrire des informations pendant la conception pour aider le futur réutilisateur à mieux comprendre le circuit. Il récupère des informations et permet de rajouter les contraintes en prévision d'une conception future. L'utilisation des commentaires se fait en trois étapes : spécification de la simulation et de ses conditions, analyse des résultats de la simulation et attribution d'un nom à chaque résultat pour lui permettre d'être utilisé par d'autres routines. Ce travail met en évidence la nécessité de trouver un formalisme pour documenter un bloc en cours de conception et en vue de sa réutilisation.

III.3.2 Dimensionnement et layout automatiques

Nous présentons dans ce paragraphe quelques travaux importants sur la synthèse automatique de circuits (dimensionnement + layout). Des travaux de Mohamed Dessouky sont présentés de même qu'un article sur l'utilisation d'outils commerciaux tels que NeoCircuit® et NeoCell®.

Mohamed Dessouky a effectué sa thèse au LIP6-ASIM avec Marie-Minerve Louërat, a proposé plusieurs méthodes de dimensionnement et de génération de layout. Ci-dessous sont exposés quelques résultats de ses recherches. Les papiers [HAMM 2004a] et [HAMM 2004b] proposent un dimensionnement automatique lors d'une migration technologique. A partir du circuit original, les principales caractéristiques sont extraites et le but est de reproduire les mêmes performances du circuit original en redimensionnant la topologie. Ces papiers décrivent tout d'abord l'état de l'art du dimensionnement :

- le dimensionnement manuel demande beaucoup de temps
- La division et l'optimisation : le système est divisé en sous blocs et chaque bloc est dimensionné. Cependant ceci implique de très longues simulations.

- Template-based reuse : Cela est basé sur l'étude en détails des topologies, les équations non linéaires qui gouvernent le modèle sont déterminées. Ensuite en utilisant des logiciels d'optimisation, les transistors sont dimensionnés. Dès que la nouvelle topologie est prête, la migration est très rapide. Néanmoins pour de nouveaux circuits, la description de l'architecture s'effectue en plusieurs semaines. De plus cette méthodologie ne fonctionne que si l'on utilise toujours les mêmes blocs.

Puis, une méthodologie est proposée et qui fonctionne en quatre étapes :

- Extraction des principales caractéristiques du circuit original, puis reconnaissance d'une hiérarchie.
- Ecriture des dimensions, mesures, symétries, etc. associés aux différents blocs.
- Préservation de ces informations avec le moteur du logiciel dans le dimensionnement du nouveau design.
- Création d'une netlist de migration

Cette méthodologie a été appliquée sur des modulateurs $\Sigma\Delta$ du troisième ordre lors d'une migration technologie (0.35 μm à 0.16 μm).

Le deuxième article [ODA 2002] décrit une méthodologie de dimensionnement automatique pour les SoC AMS (figure 1.7) qui a été développée sur la base des outils Neolinear's NeoCircuit® et NeoCell® qui seront présentés dans le chapitre cinq de même que tous les travaux de redimensionnement. Ce dimensionnement est basé sur des algorithmes d'optimisation et sur de nombreuses simulations. Cette méthode est utilisée sur un SAR (*Successive Approximation Register*, Registre à Approximations Successives) ADC et ses performances sont comparées avec le circuit initial. L'avantage de cette méthodologie est sa flexibilité quant aux changements de technologies. En effet elle permet la réutilisation en récupérant des informations des concepteurs (test-benches, simulations, etc.).

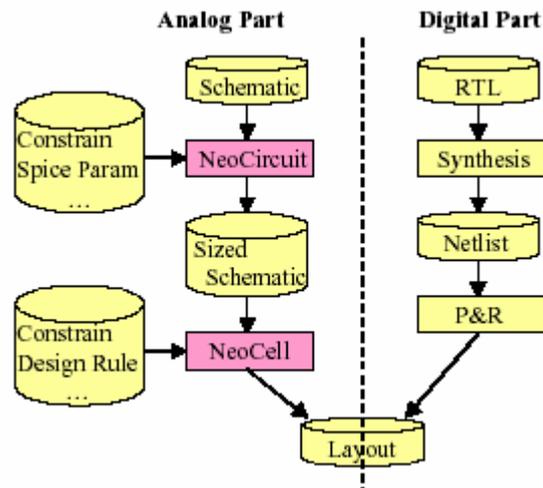


Figure 1.7. Flot de conception AMS automatisé [ODA 2002]

L'utilisation de NeoCircuit (qui dimensionne un circuit à partir du schéma et des paramètres de contraintes Spice) et de NeoCell, pour créer le layout à partir des règles de conception rend possible la réutilisation.

III.3.3 Point de vue sur l'IP analogique réutilisable

Nous présentons ici deux travaux sur la définition d'un IP analogique. Le premier article classe les IPs par classes et fonctions et en utilisant la méthode de transformation des contraintes pour remonter les spécifications. Le deuxième présente une définition d'un IP analogique en se basant sur la définition d'un IP numérique.

Le papier [LI 2003] présente une discussion sur la conception et l'application d'IP-AMS réutilisables. Le point principal de cette discussion concerne la classification que l'on peut définir pour les rendre réutilisable. Les IPs sont classés par leur application et par leur fonction (amplificateurs, filtres, opérateurs, oscillateurs...). Les contraintes de réutilisation (consommation de puissance, pas d'ajustement d'architecture etc.) doivent être définies ainsi que des classes de circuits avec leurs caractéristiques.

L'utilisation de la méthode de transformation des contraintes [MALA 1999] permet de déplacer les spécifications du circuit en plusieurs sous tâches. Elle s'applique à la phase Top-Down. A la fin de la synthèse, à un niveau hiérarchique donné, un bloc IP est défini par des variables physiques dont certaines sont des paramètres d'un IP d'un niveau hiérarchique plus bas. Pour continuer la synthèse, l'IP qui est plus bas dans la hiérarchie devient alors un bloc IP qu'il faut concevoir et il est alors nécessaire de transformer les paramètres de ce bloc en spécifications.

Une étude sur la possibilité de réutilisation d'un IP est donnée, et les caractéristiques de réutilisation sont :

- Possibilité d'ajustement d'architecture
- Possibilité de changement de tension d'alimentation et de consommation en puissance
- Possibilité d'implantation dans un SoC

Ces trois points définissent le niveau de réutilisation d'un bloc et permettent ainsi de connaître à l'avance les IPs les plus flexibles et permettant la meilleure réutilisation possible.

L'article [HAMO 2003] propose une transposition partielle du numérique de la définition d'un IP analogique. Les IPs numériques sont classés par catégories : Soft, Firm ou Hard, la différence étant la proportion de circuit synthétisé jusqu'au layout et donc du degré de flexibilité de ces circuits. Ici, la définition de l'IP analogique est donnée en tant que Firm IP c'est-à-dire qu'elle contient une partie software (vue comportementale) et une partie hardware (vue layout). La définition d'un Firm IP analogique et de son contenu est donnée, nous pouvons le résumer en 5 points :

- Une vue schématique (au niveau transistor)
- Une vue comportementale (VerilogA ou VHDL-AMS)
- Une vue analytique (modèle plus détaillé du circuit) destinée à des simulations spécifiques
- Une vue physique (Layout) reliée à la vue schématique
- Des Testbenchs (Simulations)

Une application expérimentale est donnée sur la conception d'une PLL et de son schéma associé. Une vue comportementale du circuit écrite en VerilogA est définie. Une optimisation est ensuite effectuée pour obtenir les meilleures performances. Une fois le dimensionnement final du circuit connu, la vue layout est créée. Cette vue contient les informations sur le placement des transistors. Le layout final n'est pas donné.

Cette définition d'IP peut être considéré comme un *Firm IP* et non pas comme un *Hard IP* car la vue layout final n'est pas incluse. Seules des informations contenant les coordonnées des transistors sont données.

III.3.4 L'utilisation du langage XML

Le langage XML (*Extensible Markup Language* ou langage de balisage extensible) est un standard du World Wide Web Consortium qui sert de base pour créer des langages balisés spécialisés; c'est un « méta langage ». Il est suffisamment général pour que les langages basés sur XML puissent être utilisés pour décrire toute sorte de données et de textes. Il s'agit donc partiellement d'un format de données.

Plusieurs groupes de recherche utilisent le langage XML comme langage de conception pour la mise en œuvre d'une base de donnée d'IPs [KARA 2000], [ROSE 2004] ou même l'équipe de Ian O'Connor comme nous l'avons vu précédemment.

Le travail [MONT 2004] présente une méthodologie de conception de systèmes SoC à partir de spécifications écrites en langage XML et d'IPs décrits eux aussi en XML. Cela inclut un logiciel permettant de générer le système entier à partir de spécifications de départ. Cette méthodologie a été validée sur un prototype contenant un FPGA et un processeur ARM. Le langage XML est utilisé pour prendre en compte les spécifications du design, il permet aussi la communication entre IPs. Dans ce travail, deux sortes de spécifications sont définies : celles des IPs, et celles du système. Pour les IPs, les spécifications sont décomposées en deux parties, celles *Hardware*s (entrées-sorties, bus, adresse, etc.) et celles *Software*s (bibliothèques, nom de la structure et du fichier écrit en langage C). Pour le système, les spécifications du système et les références qui relient le système aux IPs sont entrées. Enfin, un programme en Java relie le système et l'IP.

L'utilisation du langage XML semble une voie à explorer pour la création d'une plateforme de conception analogique et mixte. Ainsi nous présenterons notre travail sur ce domaine dans le chapitre trois.

III.4 Synthèse

Les différents travaux présentés apportent des contributions variées, avec des points communs et des particularités. Nous allons récapituler et regrouper les différentes solutions envisagées par ces équipes dans le tableau 1.4.

Nous avons décomposé tous ces travaux en trois catégories (figure 1.6) :

- Flot de conception système (conception du système en utilisant des IPs, exploration architecturale)
- Conception d'une bibliothèque d'IPs analogiques et mixtes (définition d'un IP, gestion d'une bibliothèque)
- Flot de conception d'IP-AMS (génération automatique de circuits ou de fonctions analogiques, dimensionnement et création du layout)

TABLEAU 1.4. SYNTHÈSE DES TRAVAUX

Laboratoires	Flot de conception Système	Bibliothèque d'IP-AMS	Flot de conception d'IP-AMS
ASIM-LIP6		oui/non	oui
LEOM	oui	oui	oui
Université de Madrid	oui	oui	
IMSE	oui	oui	oui
KUL	oui	oui/non	oui

Le laboratoire ASIM-LIP6 et Katholieke Universiteit Leuven (KUL) ont développé une plateforme de conception à base de générateurs de fonction analogiques. Ces générateurs sont au départ des vues logicielles permettant le dimensionnement d'un circuit et la création de son layout associé. Les autres équipes définissent généralement l'IP analogique comme étant un *Hard* ou un *Firm* IP descendant au niveau transistor et layout. Pour démarquer cette approche des autres équipes, nous avons mis oui/non dans la case bibliothèque d'IPs.

Nous venons de voir les différentes équipes dont les travaux portent sur le flot de conception analogique et mixte, et sur les IP-AMS. Comme nous pouvons le constater, ce flot est bien moins automatisé que celui du numérique. Et de nombreuses réponses doivent être apportées concernant d'une part la définition d'un IP-AMS et d'autre part l'évolution du flot de conception mixte.

IV CONCLUSION

Ce chapitre fait le point sur l'évolution de la conception analogique et mixte. Etant donné l'automatisation du flot numérique, des méthodologies améliorant le flot analogique sont indispensables à développer. Un état de l'art des travaux des différentes équipes travaillant sur cette problématique a été réalisé.

De cette analyse, un flot de conception idéal à base d'IPs est ressorti. Nous l'utiliserons comme référence pour notre approche méthodologique. Les difficultés de définition d'un IP analogique ont aussi été mise en avant, nous allons donc dans les prochains chapitres donner une définition simple mais précise de cet entité.

Le prochain chapitre décrit brièvement notre champ d'application, la conception de neurones sur silicium. Il apporte des précisions sur la hiérarchie et la composition de notre cas d'étude.

CHAPITRE 2 :

CONCEPTION DE NEURONES SUR SILICIUM

SOMMAIRE

I	INTRODUCTION.....	42
II	MODELE DE NEURONE IMPLEMENTE.....	43
<i>II.1</i>	<i>FORMALISME D'HODGKIN HUXLEY</i>	<i>43</i>
<i>II.2</i>	<i>EQUATIONS IMPLEMENTEES</i>	<i>45</i>
<i>II.3</i>	<i>EQUIVALENCE ENTRE LES PARAMETRES BIOLOGIQUES ET ELECTRIQUES</i>	<i>45</i>
III	CONCEPTION D'UN ASIC NEUROMIMETIQUE	46
<i>III.1</i>	<i>LE LOGICIEL NEURON</i>	<i>46</i>
<i>III.2</i>	<i>METHODOLOGIE DE CONCEPTION DE NEURONES SUR SILICIUM.....</i>	<i>48</i>
<i>III.3</i>	<i>HIERARCHIE DE L'ASIC.....</i>	<i>50</i>
<i>III.4</i>	<i>PRESENTATION DES DIFFERENTES CELLULES</i>	<i>51</i>
<i>III.4.1</i>	<i>La fonction sigmoïde.....</i>	<i>51</i>
<i>III.4.2</i>	<i>La fonction cinétique.....</i>	<i>52</i>
<i>III.4.3</i>	<i>La fonction puissance.....</i>	<i>54</i>
<i>III.4.4</i>	<i>La fonction sortie</i>	<i>55</i>
<i>III.5</i>	<i>UN EXEMPLE D'ASIC NEUROMIMETIQUE : LE CIRCUIT GALWAY</i>	<i>56</i>
IV	DISCUSSION SUR LE CHOIX DU DOMAINE D'APPLICATION	57
V	CONCLUSION.....	58

I INTRODUCTION

L'objectif de notre travail est de développer une méthodologie de conception pour le flot analogique et mixte. Pour l'illustrer, nous utiliserons comme cas d'application un circuit de notre équipe du laboratoire IMS, l'équipe Ingénierie des Systèmes Neuromorphiques (ISN). Cette équipe travaille sur des circuits neuromimétiques, c'est-à-dire qui reproduisent l'activité électrique de neurones biologiques.

L'ingénierie des circuits intégrés neuromorphiques est un domaine de recherche où la microélectronique rencontre la biologie. Le lien entre ces deux domaines est, entre autres, opéré par les neurosciences computationnelles. Cette science a pour but la compréhension d'une partie de l'activité du cerveau grâce à la modélisation et la simulation de neurones impulsionnels et même de réseaux de neurones impulsionnels.

Les neurones impulsionnels diffèrent totalement des neurones formels car leurs modèles sont biologiquement réalistes. Différents paramètres peuvent être modélisés : de la physiologie du neurone à la plasticité d'un réseau de neurones. La présentation de l'anatomie et de la physiologie d'un neurone est donnée en annexe 1.

Une des solutions pour reproduire les modèles biologiques est de créer un système adéquat de simulation qui permet d'implémenter ces modèles ; c'est le rôle de l'ingénierie neuromorphique [BORN 2005], [SAIG 2006]. Les points importants de ces systèmes sont la reconfigurabilité, et leur aptitude à fonctionner en temps réel, propriété indispensable lorsqu'ils sont liés à une application orientée vers la physiologie.

Les circuits intégrés (ASICs) analogiques et mixtes conçus dans le cadre de ces recherches modélisent l'activité électrique des neurones biologiques en temps réel et continu [RENA 2004]. Ces circuits reprennent la modélisation issue du formalisme d'Hodgkin et Huxley que nous présenterons ultérieurement. Les modèles implantés dans les premiers circuits ont dû subir des hypothèses simplificatrices en regard du formalisme retenu [DUP 1998]. Les circuits suivants se rapprochent un peu plus de la modélisation retenue ; ils incluent des canaux ioniques dépendants au calcium [LAFL 1998]. Plus tard, l'approche des circuits s'est adaptée pour leur mise en réseaux en adjoignant des synapses [DOUE 2000]. La réalisation d'une bibliothèque de fonctions électroniques [SAIG 2004] permettant la conception de ces circuits a ensuite permis l'intégration de neurones paramétrables. Ces circuits sont destinés à l'étude des réseaux de neurones biologiquement réalistes, et autorisent les connexions en temps réel avec

des cellules vivantes. Ce sont ces ASICs, et plus particulièrement le circuit Galway développé dans le cadre du projet européen FACETS qui nous serviront de cas d'étude pour notre méthodologie de conception analogique.

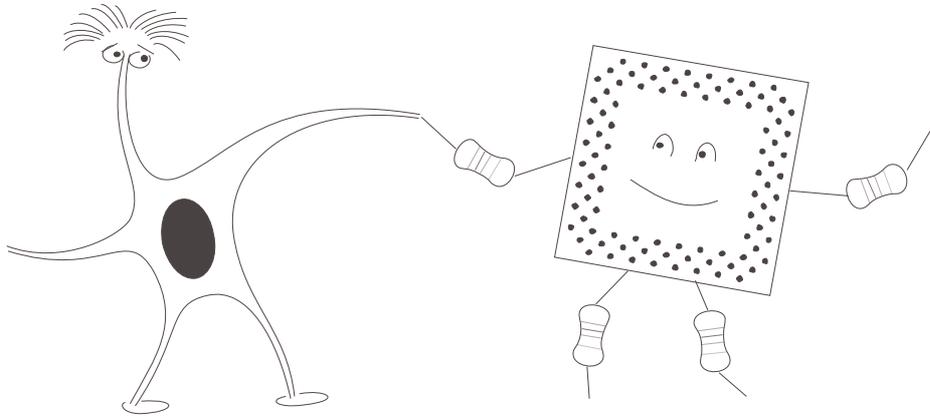


Figure 2.1. Illustration de la collaboration entre les biologistes et les microélectroniciens

II MODELE DE NEURONE IMPLEMENTE

II.1 Formalisme d'Hodgkin Huxley

Le formalisme d'Hodgkin-Huxley est une étape très importante dans la modélisation d'un neurone.

En 1952, Alan Lloyd Hodgkin et Andrew Fielding Huxley ont fourni à la communauté scientifique quatre publications dans un même volume de la revue *Journal of Physiology* [HODG 1952a], [HODG 1952b], [HODG 1952c] et [HODG 1952d] suivies d'une cinquième qui sortira la même année [HODG 1952e]. Cette dernière, souvent citée comme référence, fait la synthèse des quatre premières et propose un formalisme mathématique pour décrire l'activité électrique du potentiel de membrane des cellules nerveuses. Ces travaux, complétés par John Carew Eccles sur les mécanismes synaptiques, ont été une telle avancée pour les neurosciences qu'ils ont permis à chacun de ces trois scientifiques de partager le Prix Nobel de Médecine en 1963.

Le formalisme qui a été élaboré propose un circuit électrique ayant le même comportement électrique que la membrane. Les différentes branches qui constituent ce circuit électrique sont la capacité de membrane C_{MEM} qui sépare les milieux extra et intracellulaire, les générateurs de courants sodium (Na) et potassium (K) qui possèdent des conductances voltage dépendantes et enfin un canal de fuite à conductance constante. Ce formalisme est très largement utilisé dans le domaine des neurosciences car il est généralisable à tout type de cellule nerveuse.

Pour la conception de neurones sur Silicium, nous utilisons le formalisme d'Hodgkin-Huxley simplifié à base de conductances.

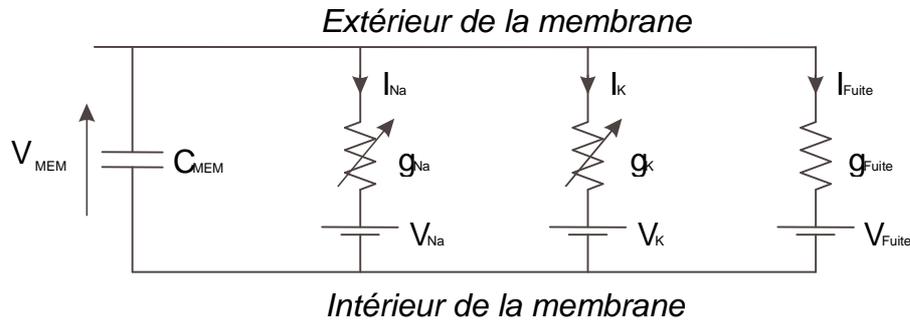


Figure 2.2. Modélisation du neurone

Ce schéma électrique nous permet d'établir l'équation suivante :

$$C_{mem} \cdot \frac{dV_{mem}}{dt} + I_{Na} + I_K + I_{fuite} = 0 \quad (2.1)$$

Avec cette notation tous les courants sont rentrants dans les différentes branches constituant le circuit électrique. V_{mem} et C_{mem} représentent respectivement le potentiel et la capacité de la membrane.

L'équation des courants ioniques se traduit sous la forme :

$$I_{ion} = g_{max} m^p h^q (V_{mem} - V_{equi}) \quad (2.2)$$

V_{equi} est le potentiel propre à chaque ion, il traduit la différence de concentration de cet ion de part et d'autre de la membrane, g_{max} est la valeur maximale de la conductance. Quant à m et h , ils sont respectivement les fonctions d'activation et d'inactivation qui se traduisent par les équations suivantes :

$$m_{\infty}(V_{mem}) = \frac{1}{1 + \exp\left(\pm \frac{V_{mem} - V_{offset}}{V_{slope}}\right)} \quad (2.3)$$

$$\tau(V_{mem}) \cdot \frac{dm}{dt} = m_{\infty}(V_{mem}) - m \quad (2.4)$$

$$\tau(V_{mem}) = K \quad \text{avec } K \text{ une constante} \quad (2.5)$$

Ces équations proviennent du modèle simplifié à base de conductances, en effet le modèle original d'Hodgkin-Huxley possède une cinétique variable.

La catégorie du neurone dépend du nombre et du type de ces courants ioniques. Par exemple, un neurone Fast Spiking (FS) possède trois courants ioniques : le courant Sodium, le courant Potassium et un courant de Fuite.

II.2 Equations implémentées

Nous présentons maintenant les équations spécifiques des principaux courants ioniques.

Equation de la conductance de fuite :

Le courant de fuite est voltage dépendant, alors :

$$I_{\text{fuite}} = g_{\text{fuite}} \cdot (V_{\text{mem}} - V_{\text{fuite}}) \quad (2.6)$$

Equation de la conductance sodium :

La conductance sodique est pondérée par deux variables m et h qui prennent en compte l'activation et l'inactivation du canal sodique. Sa modélisation est la suivante :

$$I_{\text{Na}} = g_{\text{Na}} \cdot m^3 \cdot h \cdot (V_{\text{mem}} - V_{\text{Na}}) \quad (2.7)$$

Equation de la conductance potassium :

La conductance potassique est, elle aussi, pondérée par une variable n, par contre elle ne comprend qu'un terme d'activation :

$$I_{\text{K}} = g_{\text{K}} \cdot n^4 \cdot (V_{\text{mem}} - V_{\text{K}}) \quad (2.8)$$

Equation de la conductance modulante :

La conductance potassique est, elle aussi, pondérée par une variable n, par contre elle ne comprend qu'un terme d'activation :

$$I_{\text{M}} = g_{\text{M}} \cdot m^k \cdot h \cdot (V_{\text{mem}} - V_{\text{M}}) \text{ avec } k = [1 ; 2] \quad (2.9)$$

Pour la conception de nos circuits neuromimétiques, nous utiliserons ces quatre courants ioniques.

II.3 Equivalence entre les paramètres biologiques et électriques

A partir des spécifications biologiques, nous devons créer un ASIC permettant de reproduire les activités électriques des neurones.

Le cahier des charges qui est fourni aux électroniciens est élaboré par des biologistes. Ceci pose un problème quant à la valeur donnée des paramètres. En effet, en électronique si la valeur demandée des tensions ou des courants est trop faible, son intégration sera très difficile

à effectuer. Le concepteur électronique choisit alors des facteurs de conversion permettant l'implémentation des modèles sur silicium. Ainsi les courants, les tensions et les conductances biologiques sont convertis en valeurs électroniques. Les valeurs électroniques des courants et des tensions sont généralement plus élevées en électronique permettant une meilleure dynamique du système. Comme nous travaillons en temps réel, le facteur de conversion du temps est égal à un.

Le tableau ci-dessous présente un exemple de conversion pour la conception d'un ASIC neuromimétique : Trieste.

TABLEAU 2.1. COEFFICIENTS APPLIQUES AUX PARAMETRES DU MODELE POUR L'INTEGRATION

	Biologique	Electronique
Courants	1 nA	100 nA
Tensions	100 mV	1 V
Conductances	1 mS	10 mS
Capacités	1 nF	10 nF

III CONCEPTION D'UN ASIC NEUROMIMETIQUE

Afin de comprendre notre cas d'étude, nous décrivons dans cette partie la méthodologie de conception d'un ASIC neuromimétique, tout en définissant sa hiérarchie et les différentes cellules qui le compose. Une brève description du logiciel Neuron est effectuée.

III.1 Le logiciel Neuron

Ce logiciel a été créé à l'université de Yale (Etats-Unis).

Il permet de simuler l'activité électrique dans les neurones, permettant même de simuler les différents canaux du neurone ainsi que les échanges au niveau de la synapse etc.

Il permet de reproduire l'activité des neurones à partir d'équations mathématiques du modèle : c'est un simulateur numérique.

Voici ici une capture d'écran du logiciel Neuron [CARN 2006] visualisant une reconstitution 3D d'un neurone pyramidal.

Nous apercevons sur cette capture d'écran, les diverses possibilités du logiciel Neuron. Des graphiques sont à disposition pour observer l'activité électrique du neurone et même les courants ioniques. Les paramètres des modèles de ces neurones peuvent être modifiés par l'utilisateur.

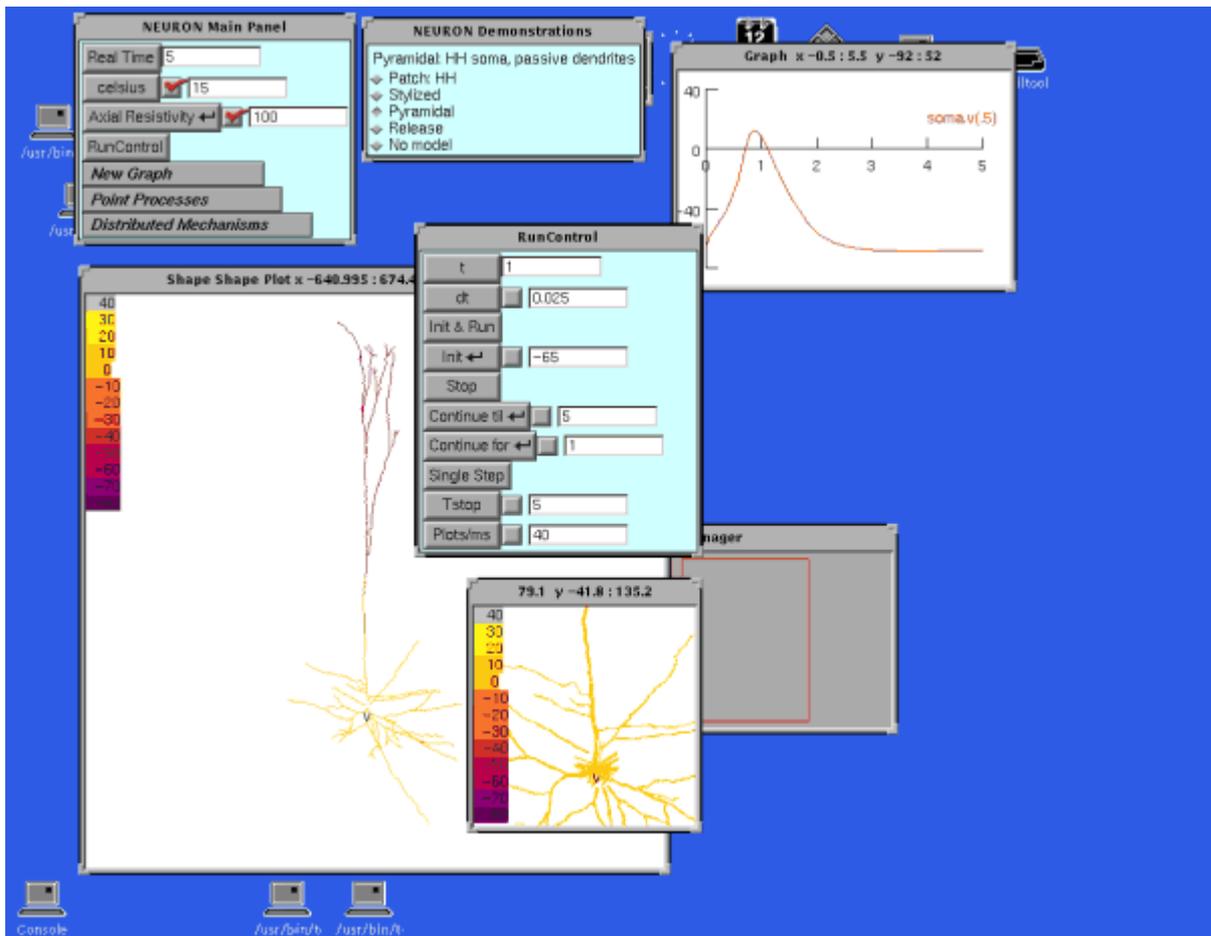


Figure 2.3. Capture d'écran du logiciel Neuron

Ce logiciel permet de connaître la fréquence des oscillations des neurones suivant les valeurs des paramètres biologiques entrés. Notre but étant de reproduire l'activité électrique des neurones, nous pouvons ainsi vérifier nos résultats avec ce logiciel.

III.2 Méthodologie de conception de neurones sur silicium

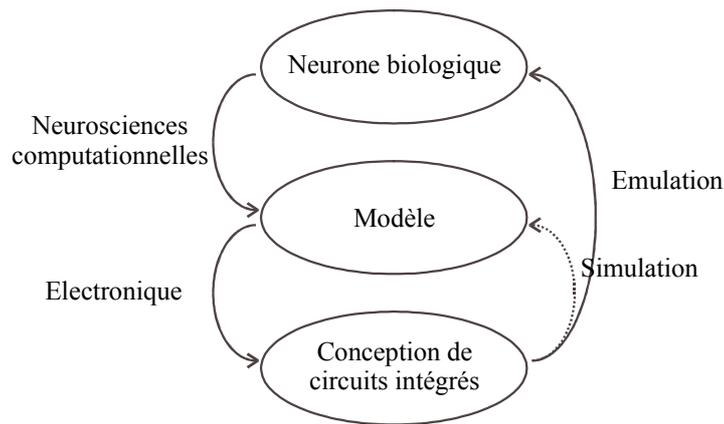


Figure 2.4. Méthodologie de conception et de validation d'un circuit intégré neuromimétique

La figure ci-dessus présente les différentes étapes lors de la conception et de la validation d'un ASIC neuromimétique.

A partir de l'étude des neurones biologiques, les neurosciences computationnelles développent des modèles de neurones caractérisés par leurs paramètres biologiques. Ces modèles sont ensuite entrés dans le logiciel Neuron, pour une vérification de la forme et de la fréquence des potentiels d'action de chaque neurone.

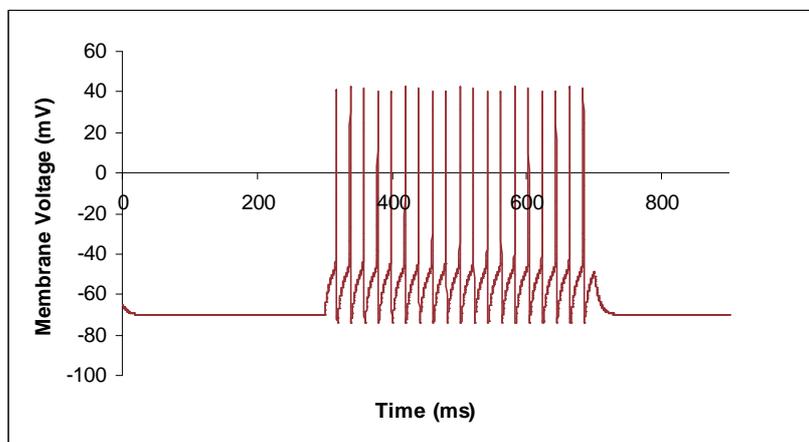


Figure 2.5. Allure des potentiels d'action sous Neuron d'un neurone exciteur

Les valeurs des paramètres de ce modèle sont données en annexe 1.

Le cahier des charges biologique nous donne le nombre de neurones à implémenter, le nombre de courants ioniques qui composent chaque neurone et enfin tous les paramètres des courants ioniques à implémenter. Le travail des électroniciens est de concevoir un ASIC

correspondant à ces spécifications, en effectuant quelques conversions entre les valeurs biologiques et électroniques.

Un travail de validation de l'ASIC est alors à faire. Pour cela, des vérifications entre les activités des neurones données par l'ASIC et celles données par le logiciel Neuron sont effectuées. Ci-dessous, la courbe d'un neurone donnée par le logiciel Neuron et celle donnée par notre ASIC.

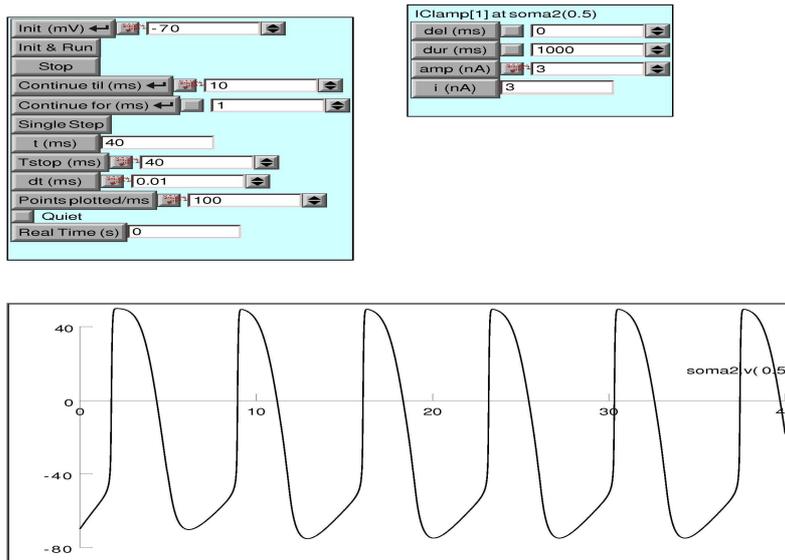


Figure 2.6. Neurone inhibiteur avec le logiciel neuron

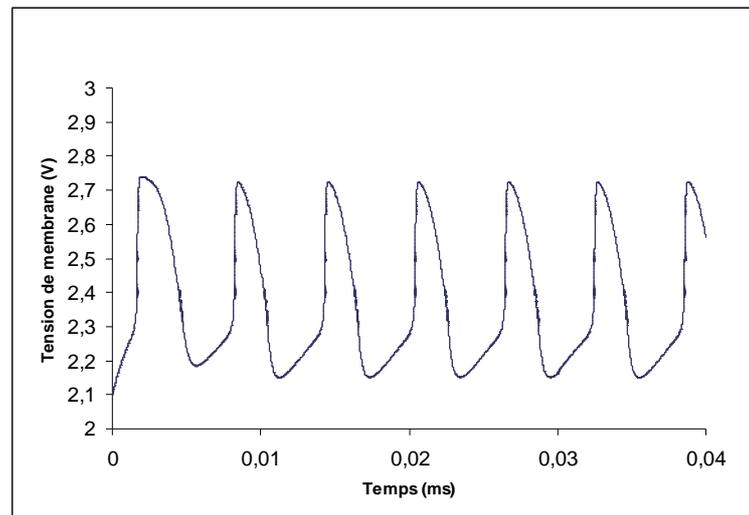


Figure 2.7. Neurone inhibiteur d'un ASIC simulé par Cadence

Enfin, après avoir validé l'ASIC, toutes les expériences menées sur celui-ci permettent de modifier et d'améliorer les modèles de neurones biologiques.

III.3 Hiérarchie de l'ASIC

La hiérarchie de cet ASIC est bien définie. Le réseau possède plusieurs neurones (*macro-blocs* au niveau électrique) qui sont décomposés en courants ioniques (*blocs*).

Chaque bloc *courant ionique* implémente l'équation suivante :

$$I_{ion} = g_{max} m^p h^q (V_{mem} - V_{equi}) \quad (2.10)$$

Cette équation se décompose en plusieurs *fonctions mathématiques* (puissance, cinétique etc.). Chacune de ces fonctions correspond à une *cellule* électrique particulière.

La figure ci-dessous décrit la correspondance et la hiérarchie du système.

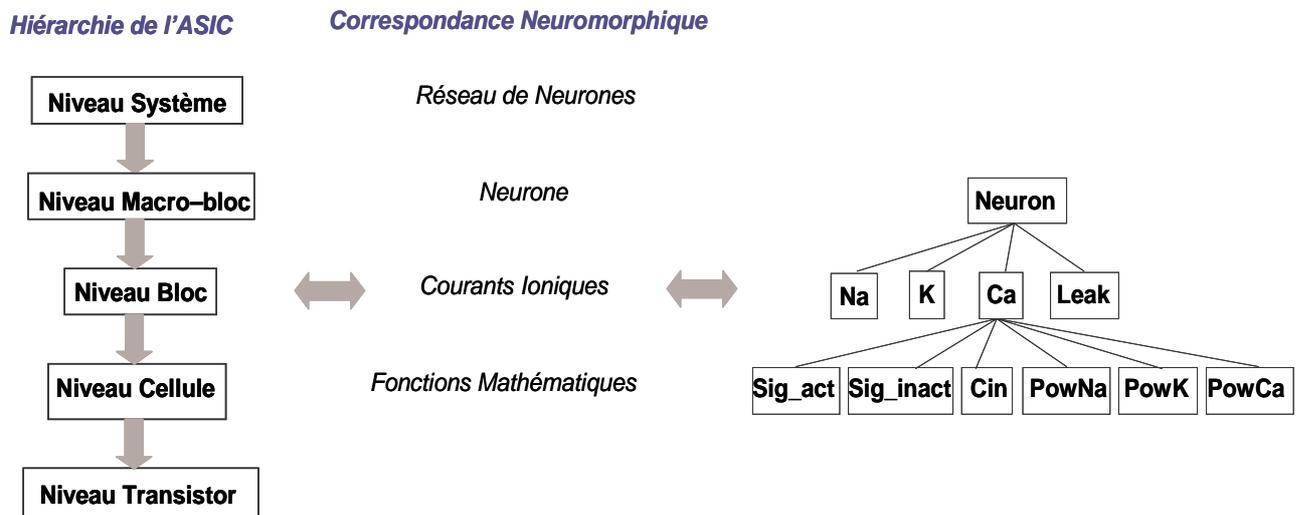


Figure 2.8. Description et correspondance de la hiérarchie

La figure suivante présente à partir de l'équation 2.10, les solutions pour concevoir ces cellules en électronique. Nous nous apercevons que l'implémentation électronique de toutes les équations est possible en utilisant des blocs électroniques bien connus comme les paires différentielles ou des OTAs.

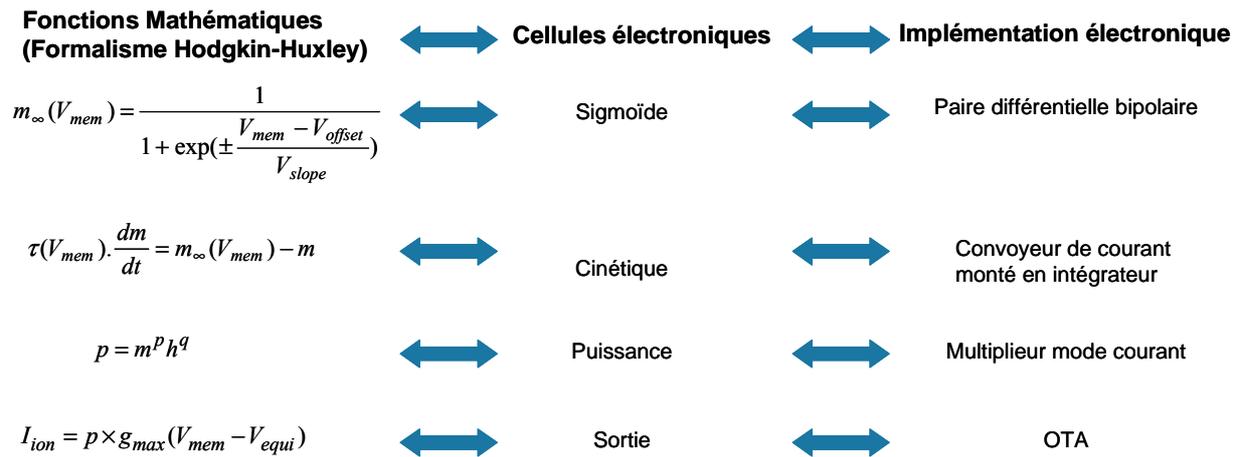


Figure 2.9. Implémentation électronique des fonctions mathématiques

Présentons maintenant toutes les cellules qui composent ce circuit.

III.4 Présentation des différentes cellules

Suivant l'équation 2.10, un courant ionique est décomposé en quatre cellules comme la figure 2.10 le décrit.

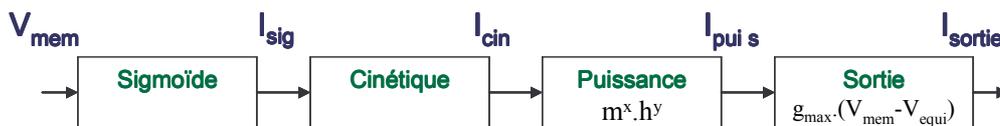


Figure 2.10. Synopsis d'un courant ionique

Nous avons vu que les courants ioniques étaient décrits par des fonctions mathématiques élémentaires (figure 2.9) :

- Une fonction sigmoïde d'activation et d'inactivation,
- Une fonction cinétique
- Une fonction puissance
- Une fonction sortie

Décrivons maintenant plus spécifiquement chaque cellule électronique [SAIG 2004].

III.4.1 La fonction sigmoïde

La constitution de la fonction sigmoïde fait appel à deux montages, à savoir un générateur de courant commandé par une tension pour la création des courants I_{PENTE} et la paire différentielle bipolaire avec étage de prédistorsion.

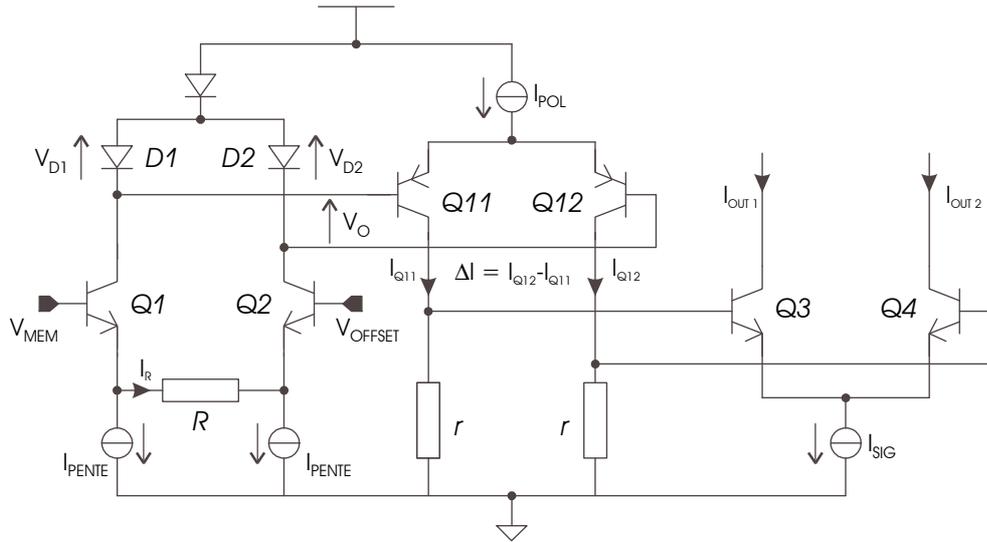


Figure 2.11. Synopsis de la fonction sigmoïdale

La différence de courant ΔI amène, à travers les résistances r , une différence de tensions à l'entrée de la paire différentielle composée par les transistors Q3 et Q4.

En utilisant les résultats connus des calculs basés autour d'une paire différentielle bipolaire, nous obtenons :

$$\begin{cases} I_{OUT1} = \frac{I_{SIG}}{1 + \exp\left(+\frac{r \cdot I_{POL}}{R \cdot U_T} \cdot \frac{V_{MEM} - V_{OFFSET}}{I_{PENDE}}\right)} \\ I_{OUT2} = \frac{I_{SIG}}{1 + \exp\left(-\frac{r \cdot I_{POL}}{R \cdot U_T} \cdot \frac{V_{MEM} - V_{OFFSET}}{I_{PENDE}}\right)} \end{cases} \quad (2.11)$$

En identifiant l'équation 2.11 à l'équation 2.3, nous obtenons le terme d'activation avec le courant I_{OUT1} et le terme d'inactivation avec le courant I_{OUT2} . Ce terme d'activation, ou d'inactivation, est normé entre 0 et 1 dans le modèle biologique; il devient dans le circuit borné entre 0 et I_{SIG} .

III.4.2 La fonction cinétique

La sigmoïde précédemment réalisée est le second membre d'une équation différentielle du premier ordre.

$$\tau_m \frac{dm(V_{MEM})}{dt} = m_\infty(V_{MEM}) - m(V_{MEM}) \text{ où } m_\infty(V_{MEM}) \text{ est obtenue par la fonction}$$

sigmoïde.

Une des solutions pour implémenter électroniquement cette fonction mathématique est présentée par la figure ci-dessous.

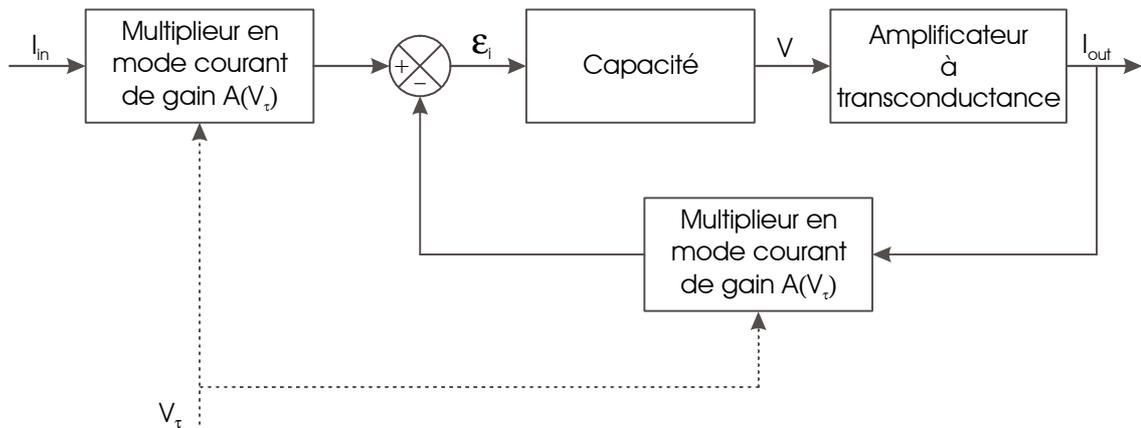


Figure 2.12. Schéma synoptique de la fonction cinétique

Afin de déterminer un schéma bloc associé, nous posons respectivement les gains $A(V_\tau)$ et B pour les multiplieurs en mode courant et pour l'amplificateur à transconductance. Rappelons que la fonction de transfert tension-courant d'une capacité C est $\frac{1}{Cp}$.

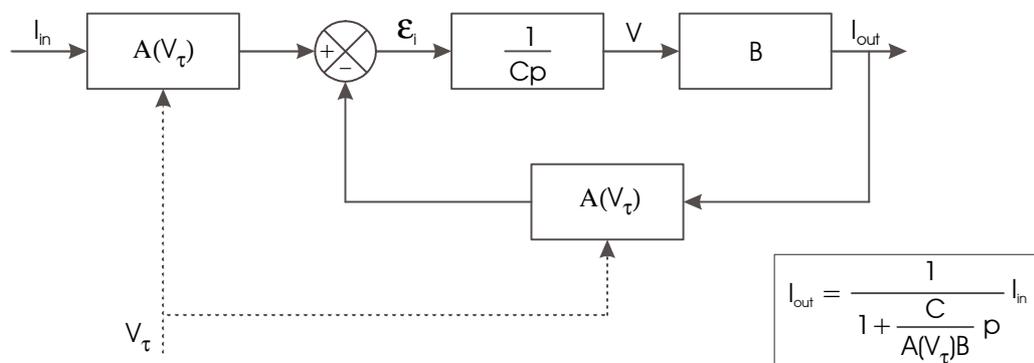


Figure 2.13. Schéma bloc de la fonction cinétique

Ce schéma bloc nous amène à l'équation suivante :

$$I_{out} = \frac{B}{Cp} (A(V_\tau) \cdot I_{in} - A(V_\tau) \cdot I_{out}) \tag{2.12}$$

$$\frac{C}{A(V_\tau) \cdot B} \cdot p \cdot I_{out} = I_{in} - I_{out} \tag{2.13}$$

L'équation 2.13 nous permet d'identifier, après passage dans le domaine temporel, la cinétique réglable τ_m :

En choisissant des transistors possédant la même surface d'émetteur, travailler sur les densités de courant des transistors revient à travailler sur les courants eux-mêmes. Nous obtenons alors :

$$I_{C_{Qh}} \cdot I_{C_{Qm}} \cdot I_{Dm1} \cdot I_{Dm2} = I_{C_{Qout}} \cdot I_{D2} \cdot I_{D1} \cdot I_{C_{Q1}} \quad (2.16)$$

Nous pouvons établir que :

- $I_{C_{Qm}} = I_{Dm1} = I_{Dm2} = I_m$
- $I_{C_{Qh}} = I_h$
- $I_{C_{Q1}} = I_{D1} = I_{D2} = I_{pol}$
- $I_{C_{Qout}} = I_{out}$

Nous aboutissons ainsi à la relation suivante :

$$I_{out} = \frac{I_m^3 \cdot I_h}{I_{pol}^3} \quad (2.17)$$

La variation des termes d'activation et d'inactivation, compris entre 0 et 1, est représentée par la variation d'un courant compris entre 0 et 20 μA . Les termes d'activation et d'inactivation, bornés entre 0 et 1, entraînent un terme puissance, qu'il soit m^3h , m^2h ou encore n^4 , lui aussi borné entre 0 et 1. Ce bloc nous donne les équations suivantes :

$$I_{out}(m^2h) = \frac{I_m^2 \cdot I_h}{I_{pol}^2} \quad I_{out}(m^3h) = \frac{I_m^3 \cdot I_h}{I_{pol}^3} \quad \text{et} \quad I_{out}(n^4) = \frac{I_n^4}{I_{pol}^3}$$

Si nous concevons le circuit électrique avec $I_{pol} = 20 \mu A$, nous obtenons alors un courant I_{out} borné entre 0 et 20 μA , tout comme les courants représentant les termes d'activation et d'inactivation. Ce choix de conception nous offre l'avantage de n'avoir à concevoir qu'un seul bloc réalisant la fonction de sortie ; car quelle que soit sa conductance d'appartenance son signal d'entrée aura toujours les mêmes caractéristiques d'amplitude.

III.4.4 La fonction sortie

Nous utilisons le montage avec la paire différentielle bipolaire avec étage de prédistorion car il a déjà fourni les résultats attendus dans le cas de la fonction sigmoïde. Ceci a amené le montage présenté à la figure 2.15.

Nous pouvons écrire :

$$I_S = I_2 - I_1 = \frac{I_{\text{PUISS}} \cdot g_{\text{MAX}}}{I_{\text{POL}}} \cdot \frac{(V_{\text{MEM}} - V_{\text{EQUI}})}{R} \quad (2.18)$$

Ce montage répond correctement à la fonction souhaitée.

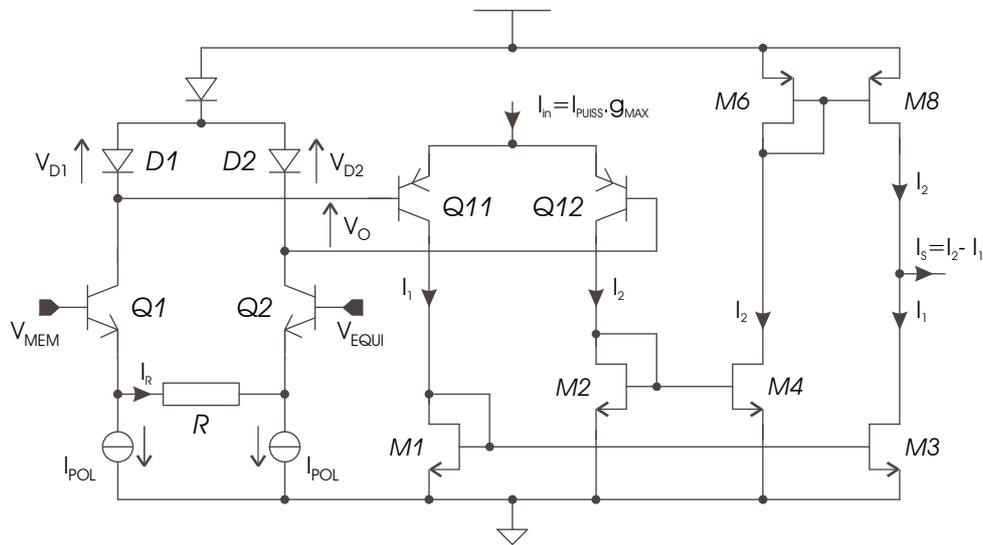


Figure 2.15. Synopsis de l'étage de sortie d'une conductance

Après avoir décrit tous ces blocs, nous pouvons étudier un des circuits neuromimétiques conçus.

III.5 Un exemple d'ASIC neuromimétique : le circuit Galway

Le circuit Galway est conçu pour être flexible et configurable. Il est réalisé en technologie AMS SiGe 0.35 µm et il occupe une surface de 10.5 mm² pour environ 50 000 composants.

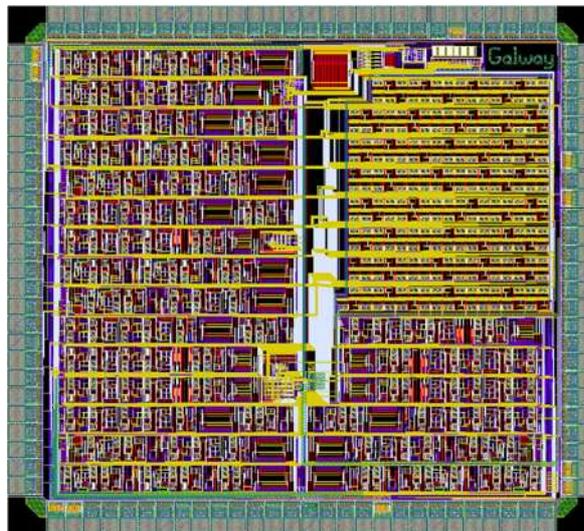


Figure 2.16. Routage final du circuit Galway

Cet ASIC contient 5 neurones artificiels. Le premier neurone contient 3 conductances (Sodium, Potassium et Fuite), il peut simuler un neurone inhibiteur (FS). Les suivants ont une quatrième conductance pour l'ajout d'un courant calcium ou modulant. Le cinquième neurone a sa quatrième conductance doublée pour avoir à la fois un courant calcium et un courant modulant.

Chaque neurone possède également trois multisynapses. Il est ainsi possible de modéliser trois types différents de connexions.

Voici une présentation d'activités de potentiels de membranes données par Galway.

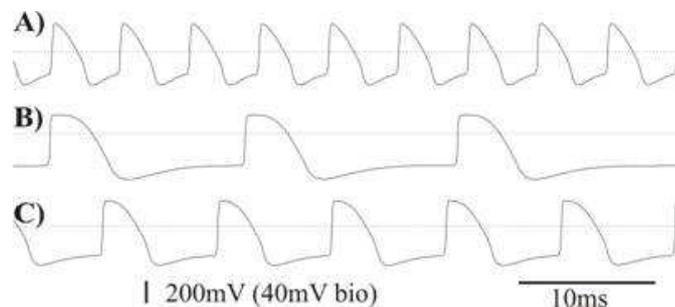


Figure 2.17. Activités de trois neurones (A, B, C) du circuit Galway

IV DISCUSSION SUR LE CHOIX DU DOMAINE D'APPLICATION

Ainsi c'est dans ce contexte d'Ingénierie Neuromorphique qu'ont été développés ces travaux de thèse et l'ASIC Galway servira concrètement de démonstrateur dans le chapitre quatre. Le choix de ce domaine d'application a été motivé par différentes raisons.

D'une part, l'équipe Ingénierie des Systèmes Neuromorphiques (ISN) possède une expérience longue de 14 ans sur la conception de Neurones sur Silicium. Au fil des 20 ASICs conçus, les blocs et fonctions élémentaires réutilisables ainsi que la hiérarchie d'un système sont désormais bien définis (figure 2.8). Cela représente un atout important pour la mise en place d'un flot de conception automatisé à base d'IPs. D'autre part, l'optimisation de la conception des ASICs, et plus largement des Systèmes Neuromorphiques, est un objectif correspondant aux besoins réels de l'équipe. En effet, étant impliquée dans des projets européens en Neurosciences Computationnelles, cette équipe répond à des projets de conception sans cesse renouvelés. Enfin, si les fonctions élémentaires à implémenter sont clairement identifiées, la migration technologique d'un bloc élémentaire déjà conçu restera toujours un problème d'actualité, en raison de l'obsolescence des technologies. Voilà encore une question rattachée à la réutilisation de la conception au sens large.

V CONCLUSION

Dans ce chapitre, nous avons brièvement décrit une partie de l'ingénierie neuromorphique et les principales étapes de l'intégration de neurones électriques. Ceci nous a permis d'approfondir ce sujet spécifique. Par la suite nous avons procédé à une description de notre cas d'application - un ASIC neuromimétique - en précisant les différents blocs. Enfin une explication a été donnée quant au choix de ce cas d'étude et aux avantages que l'on retire de ce circuit spécifique.

Comme nous l'avons vu dans ce chapitre, ce cas d'étude correspond parfaitement pour notre méthodologie de conception pour le flot analogique et mixte. Nous allons maintenant la décrire dans le chapitre 3, puis l'appliquer sur ce démonstrateur dans le chapitre 4.

CHAPITRE 3 :**BIBLIOTHEQUE D'IPs ANALOGIQUES :
DEVELOPPEMENT ET EXPLORATION****SOMMAIRE**

I	INTRODUCTION.....	60
II	DEFINITION ET CONTENU D'UN IP ANALOGIQUE.....	61
II.1	DEFINITION GENERALE	61
II.2	ANALYSE DES VUES MODELISEES EN VERILOGA.....	64
II.3	APPLICATION SUR UN BLOC ELEMENTAIRE.....	66
II.3.1	<i>Vue symbole</i>	67
II.3.2	<i>Vue connectique</i>	68
II.3.3	<i>Vue fonctionnelle</i>	69
II.3.4	<i>Vue comportementale</i>	70
II.3.5	<i>Vue schéma</i>	71
II.3.6	<i>Vue layout</i>	72
II.3.7	<i>Vue documentation</i>	72
III	CARACTERISATION D'UN IP ANALOGIQUE.....	73
III.1	CAS GENERAL	73
III.2	APPLICATION	74
III.2.1	<i>Détermination de la résistance d'entrée et de la résistance de sortie</i>	74
III.2.2	<i>Détermination des spécifications</i>	75
III.2.3	<i>Détermination de la robustesse de la cellule</i>	78
IV	CREATION DE LA BASE DE DONNEES	80
IV.1	OBJECTIFS.....	80
IV.2	MISE EN OEUVRE.....	80
V	EXPLORATION DE LA BASE DE DONNEES.....	83
V.1	PRINCIPE ET OUTILS DE L'EXPLORATION	83
V.2	MISE EN OEUVRE.....	85
V.2.1	<i>Entrée des spécifications par le client</i>	86
V.2.2	<i>Choix du concepteur, création des requêtes et exploration de la base de données</i>	89
V.2.3	<i>Récapitulatif et résultats</i>	91
VI	AIDE A LA RECONCEPTION EN CAS D'ECHEC	93
VI.1	IDENTIFICATION DU PROBLEME	93
VI.2	PROBLEME CONCERNANT LES SPECIFICATIONS DEMANDEES	93
VI.3	PROBLEME DU A LA TECHNOLOGIE OU A LA TENSION D'ALIMENTATION	94
VII	CONCLUSION.....	94

I INTRODUCTION

Ce chapitre va décrire les différentes méthodologies pour le développement de bibliothèques d'IP-AMS pour le flot de conception analogique et mixte. A partir de notre cas d'étude, le système neuromorphique, nous présentons nos définitions mais aussi une application pour chaque étape de notre méthodologie de conception.

Les différents objectifs de ce chapitre sont de définir un IP-AMS et son contenu, puis de créer une base de données d'IPs et d'étudier son exploration à partir d'un cahier des charges. Enfin, une présentation des résultats de cette exploration et des différentes manières d'aider la réutilisation sera fournie.

La définition d'un IP-AMS et de son contenu est le socle de notre méthodologie, c'est la première étape de notre méthodologie. Définir cet IP permettra la création d'une bibliothèque qui servira de base de données. Ces composants auront plusieurs niveaux de modélisation (vues).

La base de données regroupe une quantité d'IPs réutilisables de niveaux hiérarchiques différents. Plusieurs tables écrites en langage SQL définissent la morphologie de cette base. La création d'une base de données est la 2^{ème} étape dans notre méthodologie et elle sera essentielle tout comme la définition d'un IP.

Le système à concevoir doit répondre à la demande de nos collègues des sciences neurocomputationnelles, ils nous fournissent alors les spécifications biologiques. A partir de celles-ci, des conversions sont effectuées et des requêtes sont créées. Une exploration de la base de données a lieu, les IPs correspondant au cahier des charges sont choisis, un diagnostic est alors donné.

A la fin de l'exploration de la base de donnée, se produit un bilan : soit l'exploration a permis de sélectionner un système adéquat aux spécifications soit un échec a eu lieu. C'est à ce moment là que plusieurs pistes sont présentées pour la réutilisation. Nous définirons une norme de réutilisation ce qui permettra de reconnaître les blocs permettant une meilleure réutilisation. De même une méthodologie de redimensionnement (chapitre 5) sera décrite pour que les IP-AMS puisse être plus facilement migrés dans d'autres technologies.

II DEFINITION ET CONTENU D'UN IP ANALOGIQUE

II.1 Définition générale

Si l'on se réfère à l'état de l'art dressé au chapitre 1, notre conception de l'IP analogique est du type *Hard IP* dans le sens numérique c'est-à-dire qu'il correspond à un bloc déjà conçu au niveau transistor et, a priori, optimisé pour une technologie.

Quelles sont les principales propriétés attendues d'un IP analogique ? Premièrement, il doit donner une caractérisation précise du bloc déjà conçu. Deuxièmement, il doit être décrit à différents niveaux d'abstraction, utiles pour l'exploration Top-Down et la vérification Bottom-Up du système. De manière générale, il doit contenir un maximum d'informations propices à la réutilisation du bloc tel quel, mais aussi facilitant une re-conception du bloc si nécessaire.

Notre IP possède 7 vues donnant des informations électriques résumées par le tableau 3.1.

Hormis la vue appelée *documentation*, toutes les vues de l'IP possèdent le même symbole et les mêmes terminaux. Ceci permettra d'identifier facilement la vue adéquate à chaque niveau hiérarchique du système.

Nous avons privilégié VerilogA [VERI 1997] comme langage de modélisation pour les vues *connectique*, *fonctionnelle* et *comportementale*. En effet, comme nous utilisons l'environnement Cadence pour les simulations, il est préférable de choisir VerilogA plutôt que VHDL-AMS.

Ces vues sont utiles pour toutes les simulations multi-niveaux en particulier dans la phase de vérification du flot de conception (Bottom-Up). La vue *connectique* vérifie la connexion des tensions d'alimentation et du courant de polarisation. La vue *fonctionnelle* décrit les équations idéales représentant la fonction à implémenter. La vue *comportementale* est plus riche, elle décrit le circuit avec des équations reproduisant le comportement électrique du circuit.

Le Tableau 3.1 rassemble les différentes vues incluses dans l'IP.

TABLEAU 3.1. DIFFERENTES VUES INCLUSES DANS UN IP

Nom des vues :	Description / Rôle	Format de description
Symbole	visualise la fonction	Graphique
Connectique	vérifie les connexions entre les différents blocs	VerilogA
Fonctionnelle	modélise le comportement électrique idéal	VerilogA
Comportementale	modélise le comportement électrique non idéal, extrait du schéma	VerilogA
Schéma	présente le schéma au niveau transistor	Graphique / netlist
Layout	dessin des masques dans la technologie cible	Graphique / GDS-II
Documentation	voir ci-dessous...	Texte

La vue *documentation* est essentielle pour la future exploration de la base de données à partir de spécifications définies au niveau système. Cette vue est donc un point capital de notre méthodologie de réutilisation, car elle contient des informations concernant la conception et le potentiel de réutilisation du bloc IP.

Cette vue contient les informations suivantes :

- La technologie
- La tension d'alimentation
- Les différents terminaux
- Les spécifications et procédures de caractérisation associées
- Les différentes relations entre les paramètres des spécifications (gain, dynamique...) et ceux de conception (courant de polarisation, résistances...)
- La surface du layout
- Le coût du circuit (déterminé à partir de la technologie et de la surface)

Nous rappelons ici la hiérarchie d'un ASIC (figure 3.1).

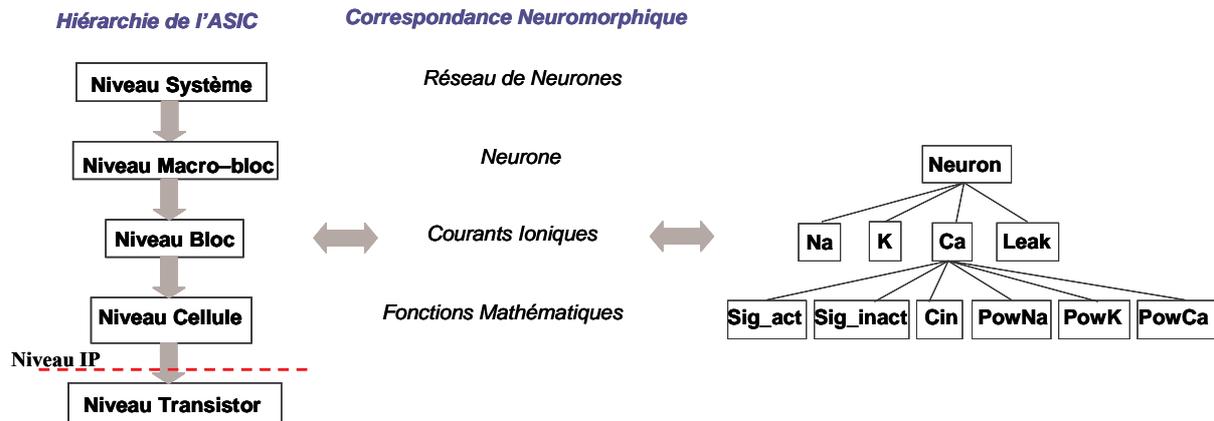


Figure 3.1. Hiérarchie d'un ASIC neuromorphique

Les IPs dont le contenu est défini précédemment, correspondent au niveau *Cellule* de la hiérarchie d'un ASIC. Les niveaux *Bloc* et *Macro-bloc* sont des descriptions d'architecture composées de ces *Cellules*.

Dans notre cas, les spécifications d'un IP sont essentiellement les dynamiques d'entrée. Ces domaines de variations sont les intervalles de valeurs tels que la fonction circuit est garantie. Ces informations proviennent des simulations des IPs (Niveau *Cellule*) et se propagent ensuite au niveau *Bloc* et *Macro-bloc*.

Comme le montre la figure 3.2, à une fonction mathématique au sens neuromorphique (par exemple la fonction Sigmoidale, voir chapitre 2), peut correspondre plusieurs IPs. Ainsi une vue générique supplémentaire (VerilogA), indépendante de l'IP, modélise cette fonction mathématique. Nous retrouvons également sur la figure 3.2, les 7 vues décrites précédemment.

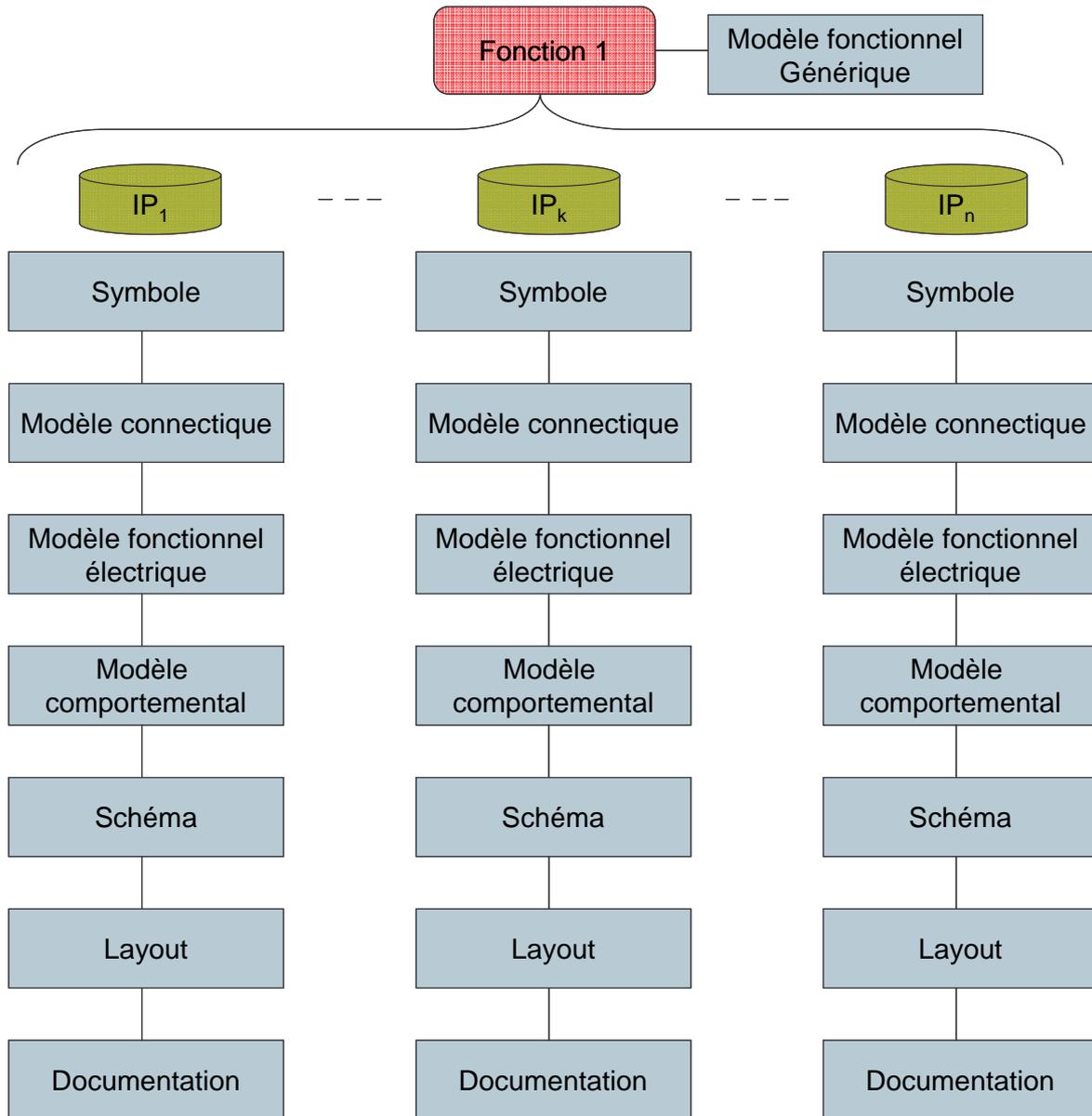


Figure 3.2. Représentation du contenu d'une fonction

II.2 Analyse des vues modélisées en VerilogA

Nous avons conçu tous ces modèles suivant une structure standard que nous expliquons ici.

Mais une question se pose dans un premier temps : quelle est l'utilité de ces modèles ?

Ces modèles sont utiles dans les deux phases du flot de conception : la phase Top-Down de synthèse et celle Bottom-Up de vérification.

Lors de la phase Bottom-Up, ces trois niveaux de modélisation permettent de vérifier le fonctionnement du système par une simulation multi-niveaux. Elle consiste à analyser le fonctionnement en détail d'un bloc avec la vue comportementale (ou au niveau transistor) tout en décrivant les autres blocs à des niveaux d'abstraction plus hauts (vue fonctionnelle voire

connectique). Cette approche permet de réaliser des simulations rapides, et même d'en effectuer certaines qui auraient été impossibles au niveau transistor.

Lors de la phase de synthèse Top-Down, le modèle fonctionnel est utile à la validation d'architecture du système. Egalement pendant la conception d'un nouveau bloc au niveau transistor, le concepteur peut s'aider de ce modèle pour créer le circuit et la topologie appropriée visant à reproduire la fonction voulue.

Structure du modèle connectique

Le modèle connectique permet de tester le courant de polarisation.

Le courant de polarisation appliqué sur un bloc est comparé au courant de polarisation nominal du système. Ce test permet ainsi de s'assurer de l'homogénéité du courant appliqué sur tous les blocs mais aussi de vérifier si aucun court-circuit (ou circuit ouvert) malencontreux n'est présent.

Les équations de ces tests sont simples, il suffit de regarder si la valeur du courant de polarisation appartient à un intervalle. Celui-ci est constitué d'une borne inférieure égale à la valeur nominale (courant de polarisation) moins ε et d'une borne supérieure égale à la valeur nominale plus ε . Pour notre application, ε est égal à 8 % de la valeur nominale.

Nous pourrions facilement effectuer aussi un test similaire pour la vérification de la tension d'alimentation.

Structure des modèles fonctionnel et comportemental

Ces modèles peuvent se décomposer en trois parties :

- Le bloc d'entrée modélise les caractéristiques d'entrée du circuit, classiquement les résistances d'entrées.
- Le bloc de transfert constitue le cœur du circuit. Il comprend les équations de la fonction demandée.
- Le bloc de sortie modélise les caractéristiques de sortie du circuit, classiquement les résistances de sorties.

On peut souligner la différence entre le modèle comportemental et le modèle fonctionnel. Le modèle fonctionnel comprend les résistances d'entrées et de sorties idéales de même que l'équation idéale de la fonction. Alors que le modèle comportemental est établi à partir du

schéma au niveau transistor de la fonction. De nombreux ajustements des équations idéales et des paramètres ont été effectués pour modéliser au mieux le comportement du circuit.

En général, le modèle fonctionnel est utile pour la vérification du fonctionnement du circuit idéal tandis que le modèle comportemental permet la vérification des performances du circuit en incluant ses imperfections.

II.3 Application sur un bloc élémentaire

Pour une meilleure compréhension, nous donnons ici un exemple de notre définition d'IP sur la cellule sigmoïde décrite dans le chapitre précédent.

Voici l'équation idéale de cette fonction mathématique qui permet de calculer la valeur asymptotique que doit atteindre l'activation d'un courant ionique :

$$m_{\infty}(V_{mem}) = \frac{1}{1 + e^{\frac{-(V_{mem} - V_{offset})}{V_{pente}}}} \quad (3.1)$$

Dans la pratique, la variable m_{∞} est représentée par un courant normalisé et est comprise entre 0 et 20 μA c'est-à-dire à deux fois le courant de polarisation.

L'équation devient alors :

$$I_{m_{\infty}} = \frac{2I_{10\mu}}{1 + e^{\frac{-(V_{mem} - V_{offset})}{V_{pente}}}} \quad (3.2)$$

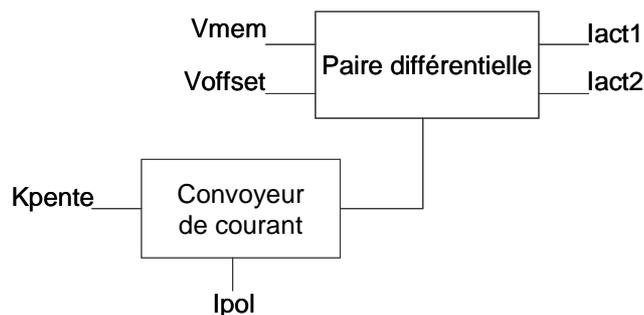


Figure 3.3. Schéma complet de la fonction Sigmoïde

Etant donné le schéma (figure 3.3) de notre cellule sigmoïde, l'entrée K_{pente} est transformée en un courant I_{pente} par le biais d'un convoyeur de courant. Ce courant permet de moduler la pente de la sigmoïde ce qui correspond à V_{pente} . Ainsi l'entrée K_{pente} correspond à une fonction de V_{pente} ainsi $V_{pente} = f(K_{pente})$.

Pour le modèle fonctionnel, nous avons cette relation :

$$V_{pente} = \frac{(V_{cc} - K_{pente})RU_T}{10rR_{conv}I_{h_10u}} = \frac{V_{cc} - K_{pente}}{C_1} \text{ avec } C_1=26 \quad (3.3)$$

Et pour le modèle comportemental, nous avons cette équation :

$$V_{pente} = C_2 \frac{(V_{cc} - K_{pente})RU_T}{10rR_{conv}I_{h_10u}} \quad (3.4)$$

C_2 est une constante qui représente les imperfections du circuit

II.3.1 Vue symbole

Voici le symbole utilisé pour cet IP, il permet d'identifier rapidement toutes les entrées sorties de ce bloc. Ce symbole est le même pour toutes les vues utilisées.

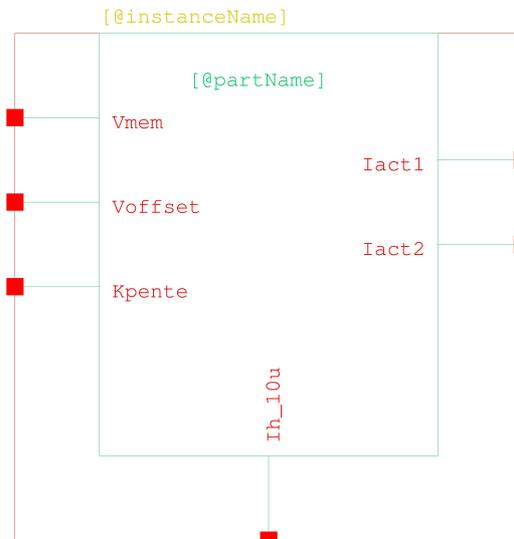


Figure 3.4. Symbole de la cellule sigmoïde

Les ports d'entrée de cette cellule sont V_{mem} , V_{offset} , I_{h_10u} et K_{pente} . L'entrée K_{pente} est une fonction de V_{pente} de l'équation 3.2. L'entrée I_{h_10u} correspond au courant de polarisation qui est égal à $10 \mu A$.

Les ports de sortie sont $Iact1$ et $Iact2$. Ces deux sorties sont identiques.

II.3.2 Vue connectique

Nous effectuons un test sur le courant de polarisation. Si sa valeur n'est pas comprise dans l'intervalle $[0,92 I_{pol} ; 1,08 I_{pol}]$ avec I_{pol} la valeur nominale, nous mettons les sorties (I_{act1} , I_{act2}) à 0. Ainsi nous pouvons détecter pour chaque cellule si le test est positif, en vérifiant la sortie à chaque étage.

TABLEAU 3.2. EXPLICATION DE LA VUE CONNECTIQUE

<pre>// VerilogA for Facets_IP, activation, veriloga_connec `include "constants.vams" `include "disciplines.vams"</pre>	
<pre>module activation(Iact1, Iact2, Ih_10u, Vmem, Voffset, Kpente); output Iact1; electrical Iact1; output Iact2; electrical Iact2; input Ih_10u; electrical Ih_10u; input Vmem; electrical Vmem; input Voffset; electrical Voffset; input Kpente; electrical Kpente; parameter real Vcc = 5 ;</pre>	<p>// Interface de déclaration du module //</p> <p>// Déclaration des entrées/sorties</p> <p>Nous déclarons les ports : input, output, ou inout (pour les deux sens), et nous définissons leur type : electrical.</p> <p>// Variable interne</p>
<pre>analog begin if(I(Ih_10u) > 9.2u) begin if (I(Ih_10u) < 10.8u) begin I(Iact1) <+ 10u; I(Iact2) <+ 10u; end end else begin I(Iact1) <+ 0; I(Iact2) <+ 0; end end endmodule</pre>	<p>// Interface de description du module //</p> <p>// Test du courant de polarisation</p> <p>Si le test est positif, la sortie de cette cellule sera égale à 10 μA, sinon elle sera égale à 0.</p>

II.3.3 Vue fonctionnelle

Voici la vue fonctionnelle écrite en langage VerilogA.

TABLEAU 3.3. EXPLICATION DE LA VUE FONCTIONNELLE

<pre>// VerilogA for Facets_IP, activation, veriloga_fonc `include "constants.vams" `include "disciplines.vams"</pre>	
<pre>module activation(Iact1, Iact2, Ih_10u, Vmem, Voffset, Kpente); output Iact1; electrical Iact1; output Iact2; electrical Iact2; input Ih_10u; electrical Ih_10u; input Vmem; electrical Vmem; input Voffset; electrical Voffset; input Kpente; electrical Kpente; parameter real Vcc = 5 ;</pre>	<p><i>// Interface de déclaration du module //</i></p> <p><i>// Déclaration des entrées/sorties</i></p> <p>Nous déclarons les ports : input, output, ou inout (pour les deux sens), et nous définissons leur type : electrical.</p> <p><i>// Variable interne</i></p>
<pre>analog begin I(Iact1) <+ -2*I(Ih_10u)/(1+exp(- (V(Vmem)-V(Voffset))/((Vcc - V(Kpente))/26+0.001f))); I(Iact2) <+ I(Iact1); end endmodule</pre>	<p><i>// Interface de description du module //</i></p> <p><i>// Equation de la sigmoïde</i></p> <p>Nous rentrons ici l'équation de du modèle fonctionnel (équation 3.3).</p>

II.3.4 Vue comportementale

Voici la vue fonctionnelle écrite en langage VerilogA.

Cette cellule est composée d'un convoyeur de courant, de miroirs de courant, d'un AOP et d'une paire différentielle. Pour chacun de ces blocs, nous caractérisons la variation engendrée et nous l'incluons dans ce modèle comportemental. La caractérisation s'effectue en comparant l'allure statique de la sortie en courant du modèle fonctionnel avec celui au niveau transistor.

TABLEAU 3.4. EXPLICATION DE LA VUE COMPORTEMENTALE

<pre>// VerilogA for Facets_IP, activation, veriloga_comp `include "constants.vams" `include "disciplines.vams"</pre>	
<pre>module activation(Iact1, Iact2, Ih_10u, Vmem, Voffset, Kpente); output Iact1; electrical Iact1; output Iact2; electrical Iact2; input Ih_10u; electrical Ih_10u; input Vmem; electrical Vmem; input Voffset; electrical Voffset; input Kpente; electrical Kpente; real Ih_diff, Ih_100u, Ih_20u, pente1, pente2, conv, Iact0; parameter real Rout=3000; parameter real Vcc=5; parameter real UT=25m; parameter real Rconv=14994.7; parameter real R=60000; parameter real r=30000; parameter real alpha_un=0.007; parameter real alpha_deux=0.017; parameter real alpha_trois=-0.0425; parameter real alpha_quatre=0.015; parameter real alpha_cinq=-0.017;</pre>	<p>// Interface de déclaration du module //</p> <p>// Déclaration des entrées/sorties</p> <p>Nous déclarons les ports : input, output, ou inout (pour les deux sens), et nous définissons leur type : electrical.</p> <p>// Variables intermédiaires</p> <p>// Variables internes</p> <p>Les paramètres (alpha_un, alpha_deux, etc.) permettent de corriger les imperfections dues au schéma et que nous détaillons ci-dessous.</p>
<pre>analog begin Ih_20u = 2*I(Ih_10u)*(1+alpha_deux); Ih_100u=10*I(Ih_10u)*(1+alpha_trois) ; pente1 = conv*(1+alpha_un);</pre>	<p>// Interface de description du module //</p> <p>// Miroir de courant imparfait</p> <p>Ces trois équations permettent de corriger les erreurs de recopie dues aux miroirs de courant.</p>

<pre> pen2 = pen1*(1+alpha_quatre); Ih_diff = Ih_20u*(1+alpha_cinq); conv = (Vcc-V(Kpen2))/Rconv; Iact0 = -Ih_diff/(1+exp(- (r*Ih_100u*(V(Vmem)- V(Voffset)))/(R*UT*(pen2+1f)))); I(Iact1) <+ Iact0 + V(Iact1)/Rout; I(Iact2) <+ I(Iact1); end endmodule </pre>	<p>// Défaut de l'AOP du convoyeur Cette équation permet la correction due à l'erreur de gain de l'AOP du convoyeur de courant.</p> <p>// Défaut de la paire différentielle Cette équation permet de quantifier la variation due à la paire différentielle.</p> <p>// Fonction de transfert Ces équations définissent l'équation non idéale de la fonction sigmoïde :</p> $I_{act} = \frac{2I_{h_10u}}{1 + e^{\frac{-10rR_{conv} \cdot I_{h_10u} \cdot (V_{mem} - V_{offset})}{R \cdot U_T \cdot (V_{cc} - K_{pen2})}}}$ <p>// Résistance de sortie Cette équation permet de définir la résistance de sortie de cette cellule.</p>
--	--

II.3.5 Vue schéma

Comme nous l'avons présenté dans le chapitre précédent, la cellule sigmoïde est construite avec des paires différentielles.

Pour faciliter la compréhension du schéma, nous avons ci-dessous un schéma simplifié. Les générateurs de courant I_{pen2} et I_{sig} sont en fait respectivement obtenus par les entrées K_{pen2} via un convoyeur de courant et un courant de polarisation I_{pol} (voir figure 3.3).

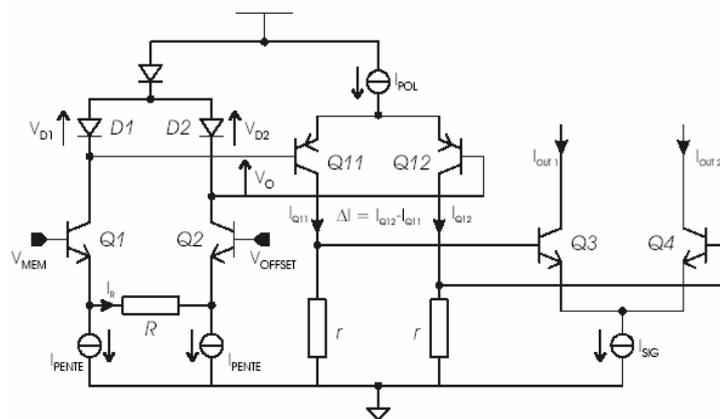


Figure 3.5. Schéma au niveau transistor de la cellule sigmoïde

II.3.6 Vue layout

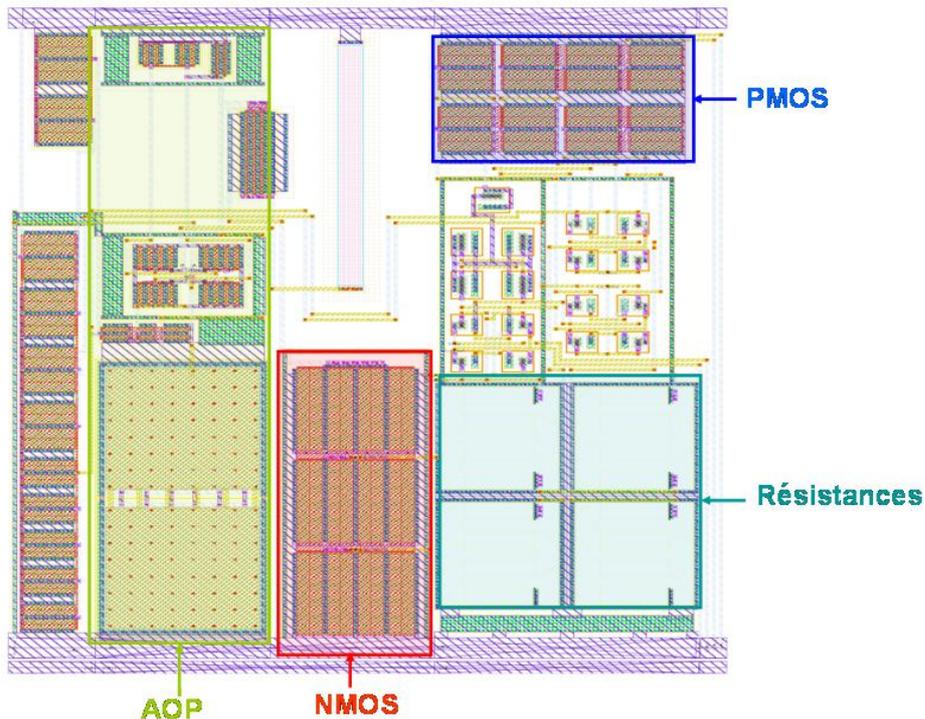


Figure 3.6. Layout de l'ip1_sigmoid_act

La figure 3.6 décrit le layout associé à la fonction sigmoïde, il est réalisé en technologie AMS BiCMOS 0.35 μm avec une tension d'alimentation à 5 V. La hauteur et la largeur ont été normalisées afin de faciliter des modifications ultérieures. Sa surface est de 23125 μm^2 .

II.3.7 Vue documentation

Technologie : AMS BiCMOS 0.35 μm

Tension d'alimentation : 5 V

Entrées : V_{mem} , V_{offset} , K_{pente} , I_{h_10u}

Sorties : I_{act1} , I_{act2}

Spécifications :

TABLEAU 3.5. INTERVALLES DE VALEURS DES ENTREES

Entrées	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

Relations entre l'équation idéale et les vues fonctionnelles et la vue comportementale :

L'équation idéale de la fonction sigmoïde est :

$$I_{act} = \frac{2I_{h_10u}}{1 + e^{\frac{-(V_{mem} - V_{offset})}{V_{pente}}}} \quad (3.5)$$

L'équation du modèle comportemental est :

$$I_{act} = \frac{2I_{h_10u}}{1 + e^{\frac{-10rR_{conv} \cdot I_{h_10u} \cdot (V_{mem} - V_{offset})}{R \cdot U_T \cdot (V_{cc} - K_{pente})}}} \quad (3.6)$$

Donc la relation entre ces deux équations vues est :

$$K_{pente} = V_{cc} - \frac{10rR_{conv} I_{h_10u}}{R U_T} V_{pente} \quad (3.7)$$

Surface du layout : 23125 μm^2

III CARACTERISATION D'UN IP ANALOGIQUE**III.1 Cas général**

La caractérisation de l'IP sert principalement à l'étude des performances de la cellule. Une étude précise est effectuée pour déterminer les intervalles dans lesquels la fonction mathématique étudiée est conservée.

Les performances sont caractérisées en effectuant des simulations DC, petits signaux (AC) et transitoire de la cellule au niveau transistor. La robustesse, quant à elle, est déterminée par des simulations MonteCarlo.

Cette étape est très importante car elle permet de bien caractériser l'IP, et tous ces intervalles de performances serviront lors de l'exploration de la base de données à partir des spécifications du cahier des charges.

Cette caractérisation s'effectue en plusieurs étapes :

- Détermination de la résistance d'entrée et de la résistance de sortie pour compléter le modèle comportemental
- Détermination des spécifications

- Détermination de la robustesse du bloc.

III.2 Application

Nous décrivons dans cette partie la caractérisation effectuée sur la fonction mathématique sigmoïde.

III.2.1 Détermination de la résistance d'entrée et de la résistance de sortie

Les entrées, de type tension, se trouvent toutes sur des grilles de transistor MOS, les courants d'entrée sont donc nuls. La résistance d'entrée de la cellule sigmoïde est donc infinie.

Pour déterminer la résistance de sortie, nous effectuons des simulations sous l'environnement Cadence.

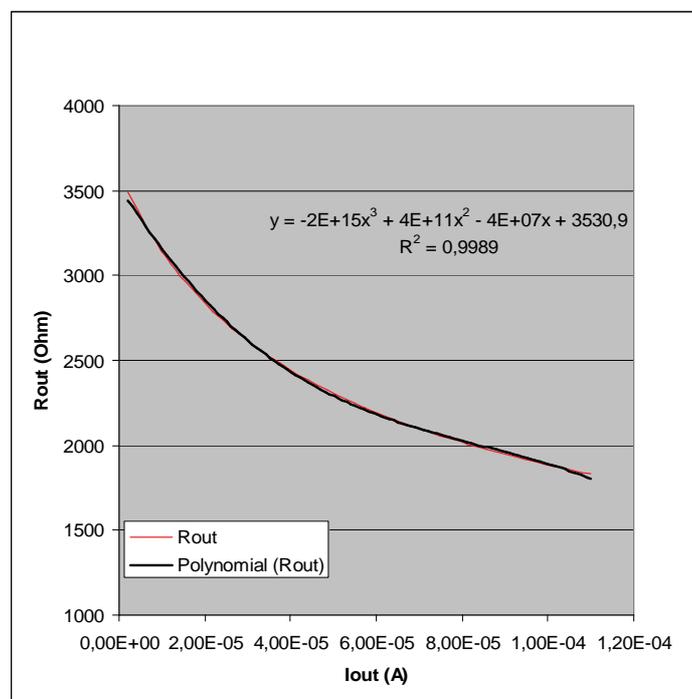


Figure 3.7. Résistance de sortie de la cellule sigmoïde

Ainsi une évaluation de la résistance de sortie est obtenue, et nous utilisons une régression polynomiale pour obtenir la valeur de cette résistance en fonction du courant de sortie I_{out} .

La cellule sigmoïde présente, sur le domaine de variation de I_{out} , une résistance non-linéaire dont on ne choisit ici de ne modéliser que la valeur moyenne afin de ne pas surcharger les vues comportementales. La simulation du système entier permettra de valider cette approximation.

III.2.2 Détermination des spécifications

Pour cette fonction, les spécifications correspondent aux dynamiques d'entrée du circuit. La performance que nous voulons conserver est la réponse statique de la sigmoïde en fonction de la tension d'entrée V_{mem} .

Les entrées étant les tensions V_{mem} , K_{pente} et V_{offset} , nous déterminons leurs intervalles de validité, en jouant sur les valeurs de ces tensions afin de conserver la réponse en courant DC.

Pour déterminer ces intervalles de validité, nous effectuons plusieurs simulations en deux étapes. Ces simulations sont ensuite comparées avec celles données par la vue fonctionnelle idéale. L'erreur quadratique entre ces deux courbes est déterminée, elle servira pour définir les intervalles de validité.

La première étape comprend trois simulations :

- Réponse statique en courant de la cellule avec les tensions V_{mem} , K_{pente} fixées à $V_{\text{dd}}/2$ et la tension V_{offset} qui varie.
- Réponse statique en courant de la cellule avec les tensions V_{offset} , K_{pente} fixées à $V_{\text{dd}}/2$ et la tension V_{mem} qui varie.
- Réponse statique en courant de la cellule avec les tensions V_{mem} , V_{offset} fixées à $V_{\text{dd}}/2$ et la tension K_{pente} qui varie.

Après cette première étape, les domaines de validité de ces entrées se dessinent. Il reste maintenant à les rendre les plus précis possibles. Pour cela, la deuxième étape étudie les pires cas en effectuant des simulations utilisant les bornes inférieures et supérieures des intervalles définies lors de l'étape 1.

La deuxième étape s'articule ainsi :

- Trois simulations de la réponse statique en courant de la cellule avec deux des tensions fixées à la borne inférieure de leur domaine de validité, et en faisant varier l'autre tension.
- Trois simulations de la réponse statique en courant de la cellule avec deux des tensions fixées à la borne supérieure de leur domaine de validité, et en faisant varier l'autre tension.

- Finalement, redéfinition des domaines de validité de chacune de ces entrées.

La figure suivante représente les réponses du circuit en jouant sur la tension V_{offset} .

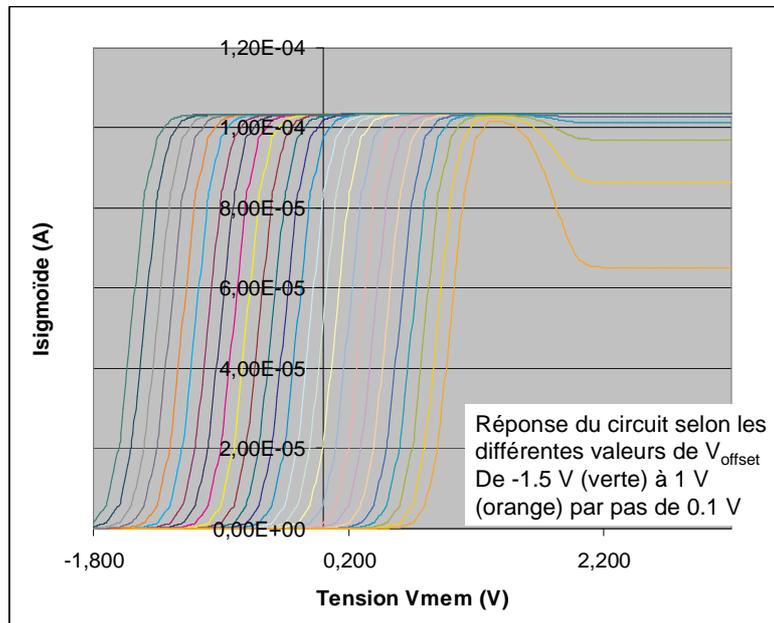


Figure 3.8. Réponse DC de la cellule sigmoïde selon plusieurs valeurs de V_{offset}

Concernant la figure 3.8, Nous avons volontairement choisi d'afficher qu'une partie des courbes (25) alors que plus d'une centaine de simulations ont été effectuées.

Ces simulations et leurs résultats ont pu être traités par un programme en langage Perl qui permet de récupérer toutes les données et les intégrer dans un fichier Excel. La figure suivante représente l'évolution de la sortie en courant de la cellule sigmoïde en effectuant une variation de la valeur K_{pente} .

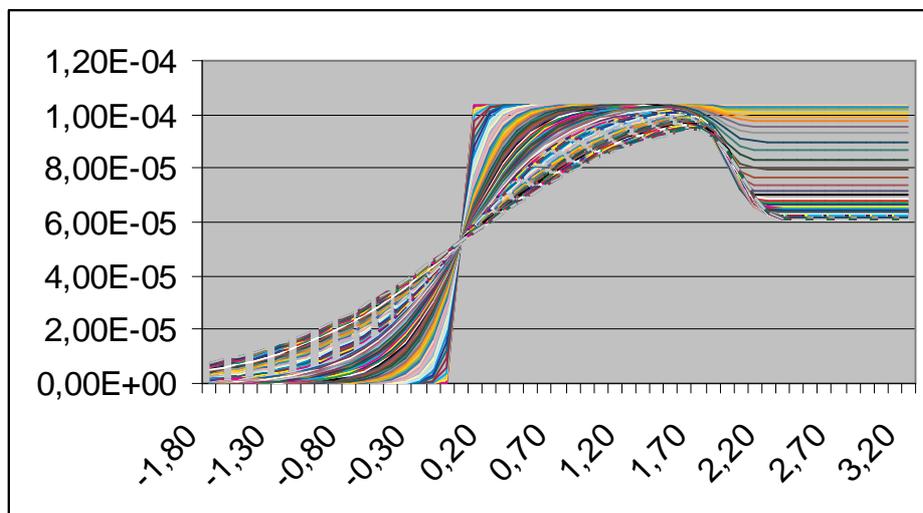
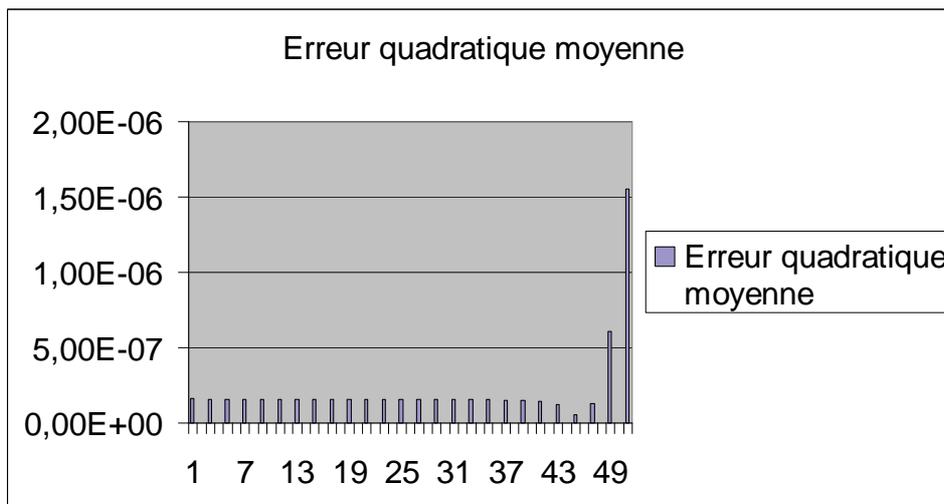


Figure 3.9. Réponse DC de la cellule sigmoïde selon plusieurs valeurs de K_{pente}

Pour déterminer les valeurs les plus précises du domaine de validité, nous utilisons l'erreur quadratique moyenne entre la fonction idéale et celle mesurée pour caractériser le circuit. Cette erreur ne doit pas dépasser 1 %. L'intervalle de validité est ainsi défini.

Voici un histogramme des résultats (figure 3.10), avec pour ordonnée l'erreur quadratique moyenne et en abscisse les différents numéros des simulations correspondant à des valeurs V_{offset} différentes.

Figure 3.10. Evaluation de l'erreur quadratique moyenne selon la valeur V_{offset} utilisée

Après traitement des résultats de cette histogramme, nous pouvons constater que la courbe numéro 50 (avec une valeur de $V_{\text{offset}} = 1$ V) possède une erreur quadratique de plus de 1 %.

Avec toutes ces données, nous pouvons définir les intervalles des entrées de la cellule sigmoïde.

TABLEAU 3.6. INTERVALLES DE VALIDITE DES ENTREES DU BLOC SIGMOÏDE

Entrées	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

III.2.3 Détermination de la robustesse de la cellule.

Pour déterminer la robustesse d'une cellule, nous utilisons les résultats des simulations MonteCarlo. Celles-ci étudient le comportement DC en courant de la cellule.

Deux tests de robustesse sont possibles. Premièrement, nous comparons les résultats obtenus avec la caractéristique idéale, au moyen de l'erreur quadratique moyenne qui ne devra pas dépasser 5 %. Nous pouvons facilement déterminer quels sont les cas défailants et ainsi réduire le domaine de validité. La figure 3.10 donne les différentes caractéristiques, il suffit alors de comparer chacune de ces courbes avec celle idéale.

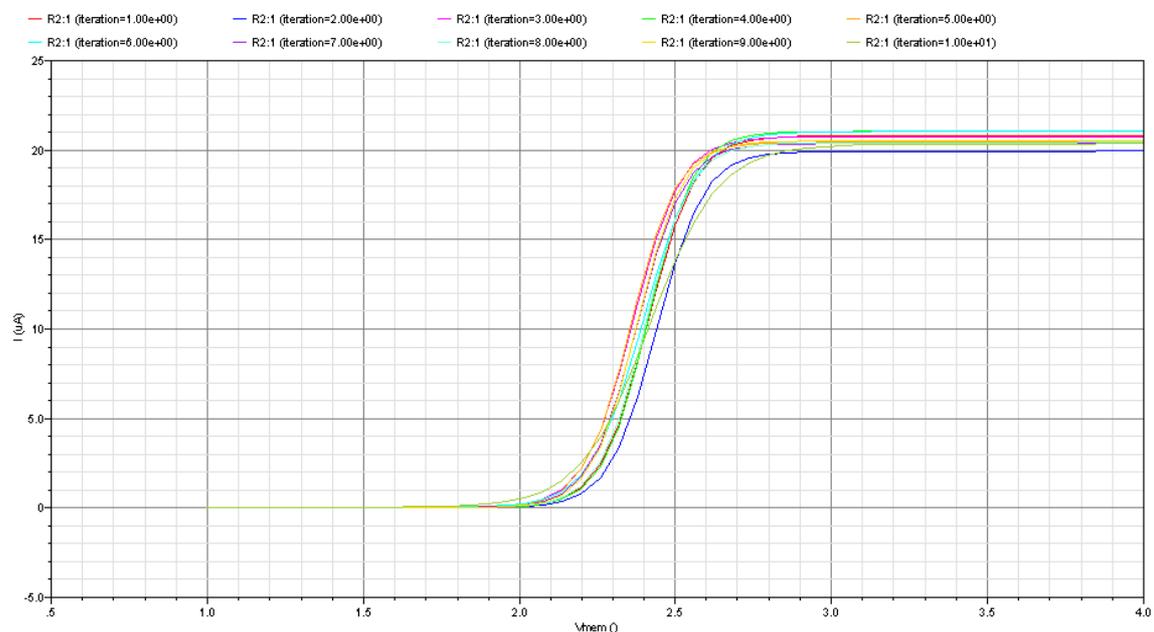


Figure 3.11. Simulation MonteCarlo de la cellule sigmoïde

Le deuxième test permet de vérifier si l'écart type n'est pas trop élevé. Ainsi pour chacun des histogrammes MonteCarlo, nous calculons le nombre de cas n'étant pas compris dans l'intervalle suivant $[0.95 \text{ Valeur moyenne} ; 1.05 \text{ Valeur moyenne}]$. Si ce nombre est trop élevé (plus de 5% par rapport aux nombres de simulations), nous réduisons le domaine de validité.

Tout d'abord nous effectuons une simulation MonteCarlo en utilisant, comme valeur des entrées, le milieu des intervalles déterminés.

Puis deux autres simulations sont faites, une avec comme valeurs la borne supérieure de l'intervalle et l'autre avec la borne inférieure.

Les figures suivantes, donnent ces trois simulations pour la cellule sigmoïde.

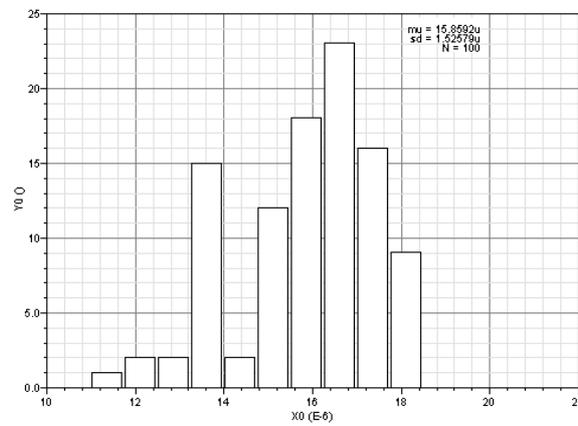


Figure 3.12. Simulation MonteCarlo de la cellule sigmoïde avec pour valeurs d'entrée les milieux des intervalles

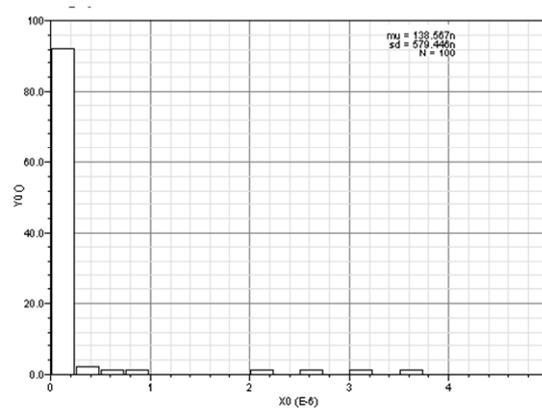


Figure 3.13. Simulation MonteCarlo de la cellule sigmoïde avec pour valeurs d'entrée les bornes supérieures des intervalles

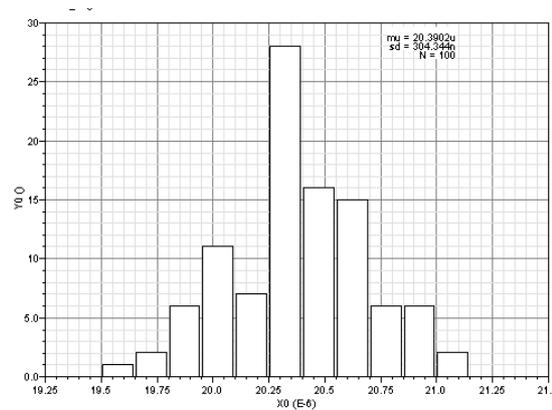


Figure 3.14. Simulation MonteCarlo de la cellule sigmoïde avec pour valeurs d'entrée les bornes inférieures des intervalles

Etant donné que les résultats MonteCarlo sont positifs, les domaines de validité que nous avons définis auparavant sont conservés.

IV CREATION DE LA BASE DE DONNEES

IV.1 Objectifs

Après avoir défini et caractérisé tous les IPs, ils sont regroupés dans une bibliothèque ayant une structure d'une base de données.

Rappelons que le système étudié possède différents niveaux hiérarchiques dans sa description (figure 3.14). L'IP élémentaire dont le contenu est défini précédemment correspond au niveau *Cellule*. Au niveau *Bloc* et *Macro-bloc*, des IPs ont également été définis, contenant une description structurelle à partir des IPs élémentaires, ainsi que les spécifications résultantes.

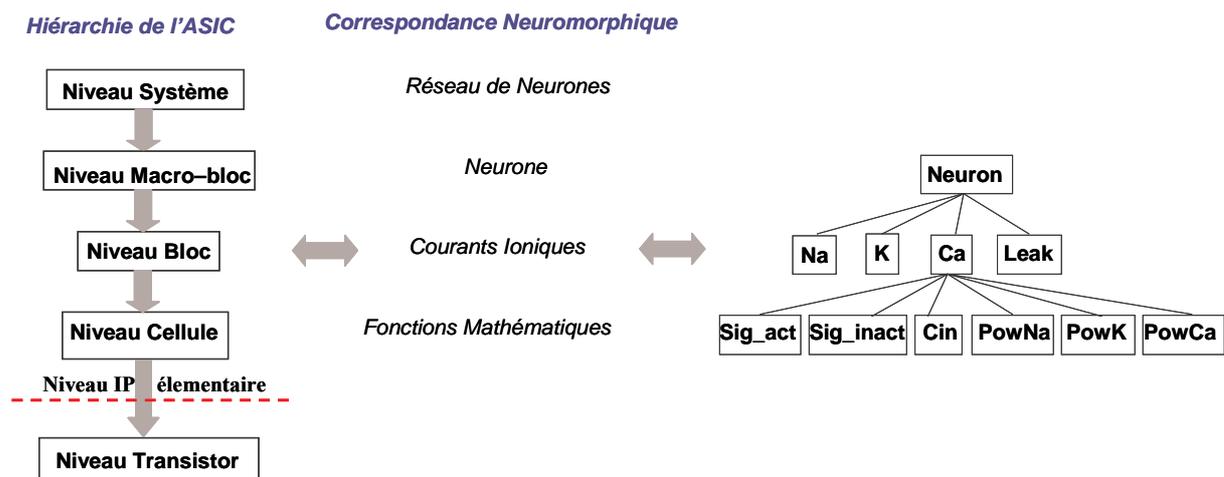


Figure 3.15. Présentation de la hiérarchie d'un ASIC neuromorphique

L'objectif principal de cette base de données est de permettre une exploration aisée et automatique des solutions correspondantes aux spécifications du système. La mise en place des liens précis et clairs entre les IPs et entre les différents niveaux de hiérarchie est essentielle afin de permettre une exploration logique de cette base de donnée.

Nous allons maintenant mettre en oeuvre ces objectifs pour définir notre base de données.

IV.2 Mise en oeuvre

Les outils utilisés pour la création, l'organisation et l'exploration de la bibliothèque d'IPs sont Php [RIGA 2006], SQL [DUBO 2005], HTML et XML [ABIT 2000].

Les langages PHP et HTML serviront pour la création et la gestion d'un formulaire de saisie des spécifications. Le langage SQL est utilisé pour la création des requêtes et la création des tables de la base de données. Tandis que le langage XML servira d'intermédiaire entre le résultat des requêtes et la vue finale sous HTML.

En ce qui concerne la structure de la base de données, plusieurs tables SQL ont été définies, elles seront décrites plus tard. La base de données est hébergée sur un serveur Apache [BOWE 2000].

SQL (*Structured Query Language*, soit Langage de requêtes structuré) est à la fois un langage de définition de données (LDD, ou en anglais DDL *Data Definition Language*), un langage de manipulation de données (LMD, ou en anglais DML, *Data Manipulation Language*), et un langage de contrôle de données (LCD, ou en anglais DCL, *Data Control Language*), pour les bases de données relationnelles.

- SQL s'avère être un langage de définition de données (LDD) dans la mesure où il permet la création de tables dans une base de données relationnelles, ainsi que de les modifier ou de les supprimer.
- SQL est un langage de manipulation de données (LMD), c'est à dire qu'il permet de sélectionner, insérer, modifier ou supprimer des données dans une table d'une base de données relationnelles.
- SQL est un langage de protection d'accès. Il est possible avec SQL de définir des permissions au niveau des utilisateurs d'une base de données. On parle de DCL (Data Control Language).

Voici la présentation des différentes tables définies dans la base de données.

Il existe 10 tables :

- Définition de chaque IP : 1 table
- Lien père-fils (lien hiérarchique entre le niveau n et le niveau n-1) entre les fonctions neuromorphiques (figure 3.14) : 1 table
- Lien père-fils entre les IPs (figure 3.14) : 3 tables
- Nombre d'IPs pour une fonction donnée : 1 table

- Valeurs des spécifications (dynamique des entrées) des IPs pour chaque niveau de hiérarchie existant (*Cellule*, *Bloc*, *Macro-bloc*) : 3 tables
- Connexions entre les différents *Macro-blocs* constituant le système : 1 table

Nous allons maintenant décrire plus en détail chaque catégorie de table.

Définition de chaque IP

La table *définition d'IP* regroupe tous les IPs existants avec leur contenu. Tous les liens vers les différentes vues telles que toutes les vues de modélisation, les vues symbole, layout, schéma de même que la technologie, la tension d'alimentation et la surface du layout sont données. Le niveau hiérarchique de l'IP et les différentes entrées/sorties sont aussi définis.

Lien père-fils entre les fonctions neuromorphiques

La table *lien père-fils fonction* présente les différents liens hiérarchiques pour une fonction donnée. C'est-à-dire qu'elle présente tous les *blocs* (ou respectivement *cellules*) du niveau hiérarchique inférieur qui sont utilisées pour concevoir ce *macro-bloc* (ou respectivement *bloc*). Il existe trois tables, une pour chaque niveau hiérarchique existant (*Cellule*, *Bloc*, *Macro-bloc*). Ceci est illustré à la figure 3.14, pour un neurone ainsi que pour un courant ionique calcium (Ca).

Liens père-fils entre les IPs

Les tables *liens père-fils IP* sont semblables aux tables *liens père-fils fonction* mais celles-ci définissent les liens hiérarchiques entre les IPs, sachant que pour une fonction neuromorphique donnée, il peut exister plusieurs réalisations (ou IPs) possibles.

Nombre d'IPs pour une fonction donnée

La table *nombre d'IPs* donne le nombre d'IPs accessibles pour une fonction neuromorphique donnée.

Valeurs des spécifications

Ces tables possèdent toutes les valeurs des spécifications (intervalles des domaines de validité) pour chaque IP. Elles sont très importantes car l'exploration de la base de données est conditionnée par ces valeurs. Trois tables sont définies, une pour chaque niveau hiérarchique (domaines de validité des fonctions mathématiques, domaines de validité des courants ioniques et domaines de validité des neurones).

Connexions entre les différents *Macro-blocs*

La table *connexions* contient les ports d'entrées/sorties de chaque *Macro-bloc*. La table *connexions* sert à la création de la netlist du système final trouvé par l'exploration (voir le prochain paragraphe).

V EXPLORATION DE LA BASE DE DONNEES

Après cette description de la base de données, nous pouvons passer à l'étape suivante, à savoir son exploration.

V.1 Principe et outils de l'exploration

A partir des spécifications du système, des requêtes sont créées. Ce sont celles-ci qui vont explorer la base de données pour déterminer et récupérer les IPs correspondant au mieux à ce cahier des charges. Cette exploration s'effectue par une approche Top-Down en recherchant les IPs de plus haut niveau, du niveau *Macro-bloc* au niveau *Cellule*, et en utilisant les valeurs des spécifications de chaque IP comme critère de recherche. Si la recherche est négative au niveau hiérarchique le plus haut, on descend alors au niveau inférieur pour effectuer une nouvelle phase de recherche.

Dans le cas où une recherche aboutirait à plusieurs solutions, nous choisissons celle qui possède les spécifications les plus larges.

Pour faciliter la compréhension de cette démarche, nous illustrons cette approche par la figure suivante.

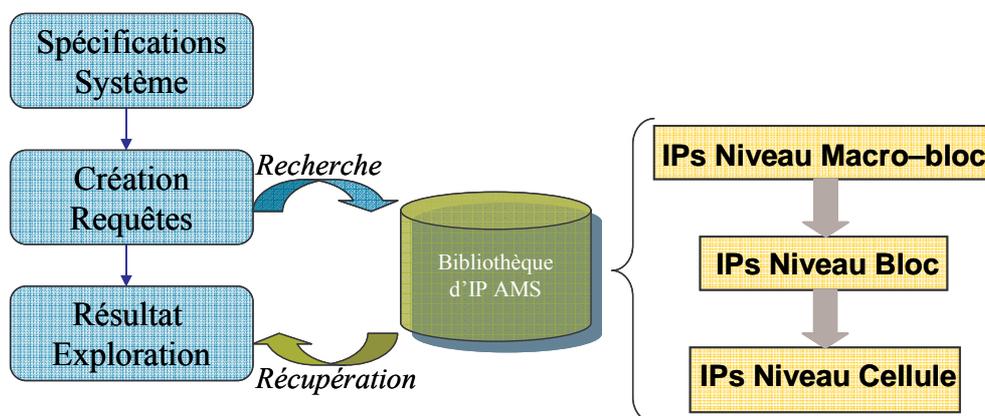


Figure 3.16. Exploration de la base de données

Les langages PHP et HTML serviront pour la création et la gestion d'un formulaire de saisie des spécifications. Le langage SQL est utilisé pour la création des requêtes. Tandis que le langage XML servira d'intermédiaire entre le résultat des requêtes et la vue finale sous HTML.

PHP est un langage interprété (un langage de script) qui est exécuté du côté serveur (comme les scripts CGI, ASP, ...) et non du côté client (comme un script écrit en Javascript ou une applet Java qui s'exécute sur l'ordinateur...). La syntaxe du langage provient de celles du langage C, du Perl et de Java. Ses principaux atouts sont :

- Une grande communauté de développeurs partageant des centaines de milliers d'exemples de script PHP ;
- La gratuité et la disponibilité du code source (PHP est distribué sous licence GNU GPL) ;
- La simplicité d'écriture de scripts ;
- La possibilité d'inclure le script PHP au sein d'une page HTML (contrairement aux scripts CGI, pour lesquels il faut écrire des lignes de code afin d'afficher chaque ligne en langage HTML) ;
- La simplicité d'interfaçage avec des bases de données (de nombreux Système de Gestion de Base de Données SGBD sont supportés, mais le plus utilisé avec ce langage est MySQL, un SGBD gratuit disponible sur de nombreuses plateformes : Unix, Linux, Windows, MacOS X, Solaris, etc...) ;
- L'intégration au sein de nombreux serveurs web (Apache, Microsoft IIS, etc.).

XML (*Extensible Markup Language* ou langage de balisage extensible) est un standard du World Wide Web Consortium qui sert de base pour créer des langages balisés spécialisés; il s'agit d'un « méta langage ». Il est suffisamment général pour que les langages basés sur XML, appelés aussi dialectes XML, puissent être utilisés pour décrire toute sorte de données et de textes. Il s'agit donc partiellement d'un format de données.

Nous utiliserons XML et son fichier de style associé XSL, pour une meilleure organisation et présentation des résultats de l'exploration de la base de données.

Le résultat de l'exploration est donné par un diagnostic évaluant le pourcentage de réutilisation pour le système, à savoir la proportion d'IPs trouvés dans la base de données pour l'ensemble des fonctions neuromorphiques décrivant le système.

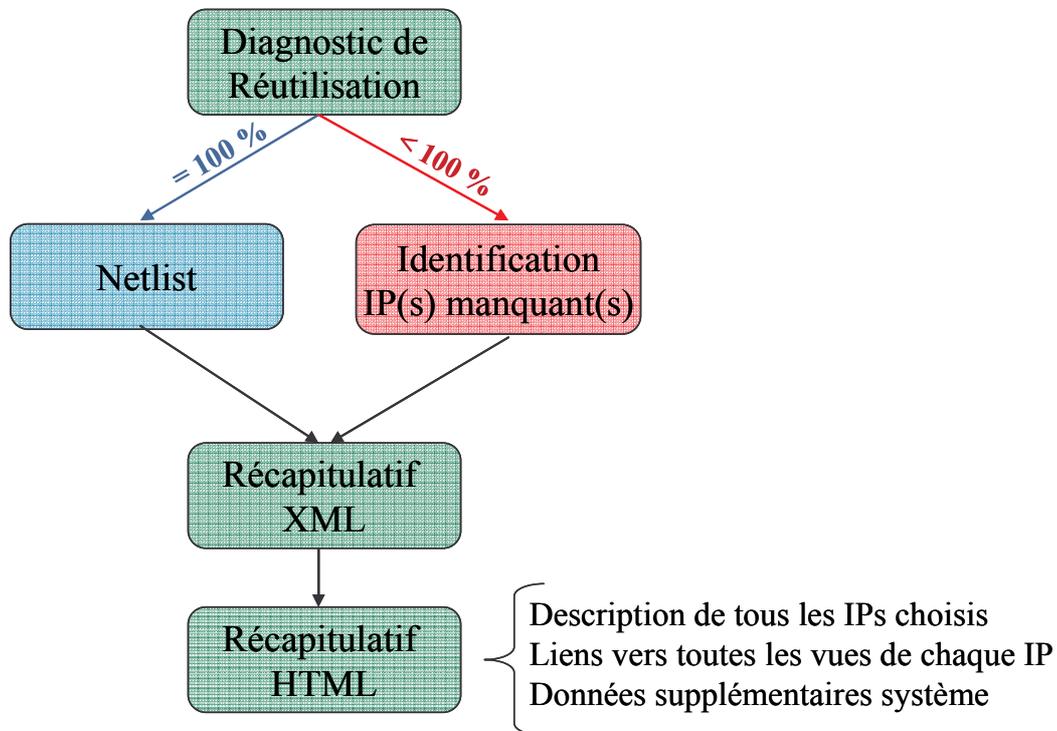


Figure 3.17. Illustration de la démarche de résultats

Si le pourcentage est de 100 %, une netlist est créée rassemblant tous les *Macro-blocs* pour concevoir le système.

Sinon d'une part la liste de tous les IPs correspondant aux spécifications initiales est fournie, et d'autre part le(s) IP(s) manquant(s) sont identifiés et localisés dans la hiérarchie. Une aide à la conception de ces IPs manquants est alors donnée. Cette aide sera décrite dans un des paragraphes suivants.

Un récapitulatif est alors créé, il décrit tous les IPs choisis avec la possibilité de voir chacune de ces vues. Des données supplémentaires sont affichées concernant le circuit comme la tension d'alimentation, la technologie utilisée et même les valeurs des composants passifs à ajouter pour la conception du système final.

V.2 Mise en oeuvre

L'exploration de la base de données peut se décomposer en plusieurs étapes.

Ci-dessous est donné le diagramme UML de la conception d'un réseau de neurones à partir des spécifications d'un client. Toutes ces étapes vont être décrites dans les paragraphes suivants.

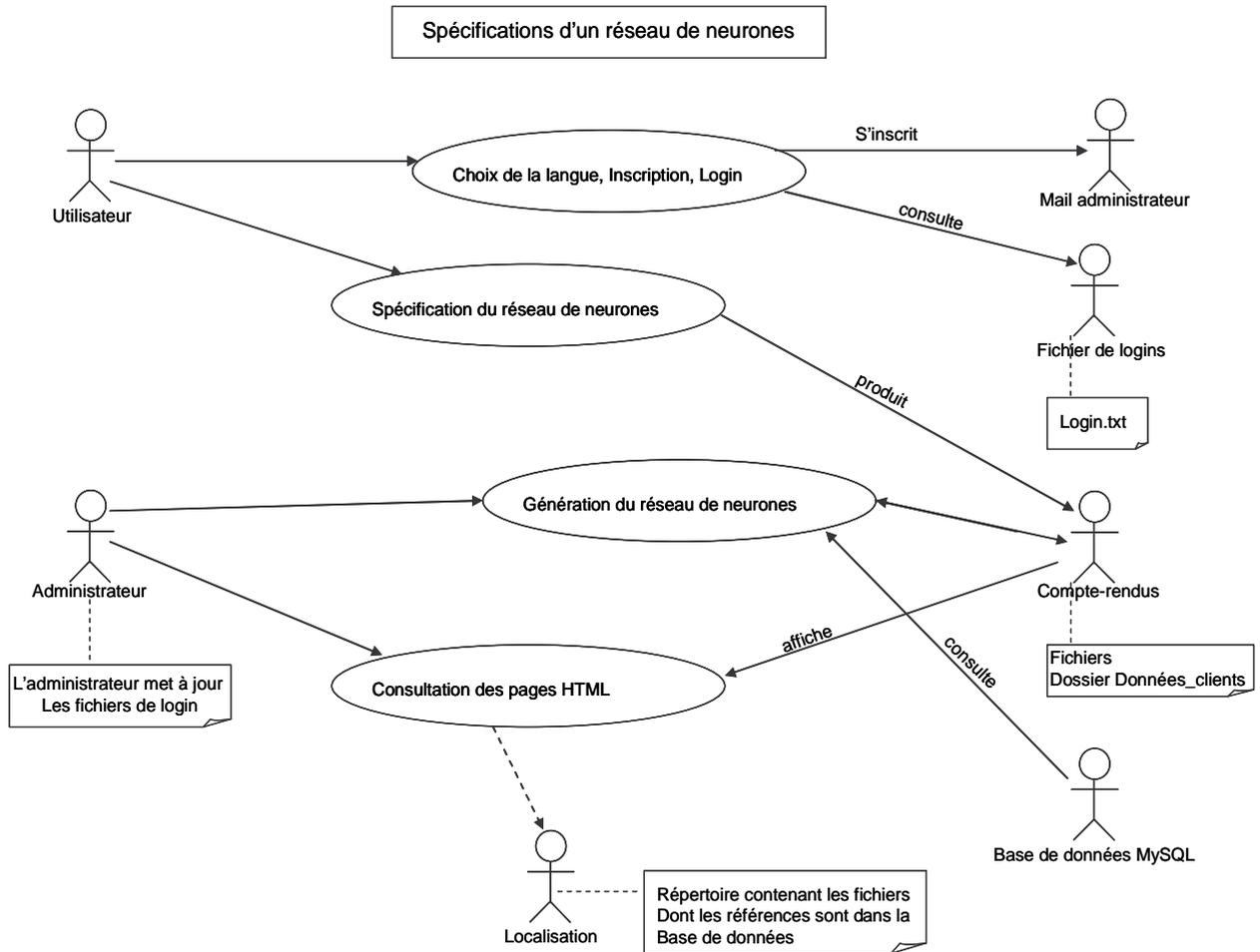


Figure 3.18. Diagramme UML des Use Case de la conception du système neuromorphique

La figure 3.18 représente tous les acteurs, symbolisés par une forme humanoïde, qui interagissent avec le système. Ces acteurs sont des entités externes (physique ou logicielle)

Nous présentons tout d'abord l'entrée des spécifications par le client et les différentes options qu'il peut choisir. Puis nous détaillerons la création des requêtes et leurs explorations. Enfin nous décrivons les récapitulatifs et les résultats.

V.2.1 Entrée des spécifications par le client

Le client saisit dans le formulaire PHP, le cahier des charges de son système neuromorphique. Il peut choisir le nombre de neurones de son réseau mais aussi toutes les conductances pour chacun des neurones. Afin d'être le plus précis possible, tous les paramètres des conductances sont rentrés ainsi que la capacité de membrane et la surface du neurone,

selon le formalisme d'Hodgkin-Huxley que nous avons présenté précédemment dans le chapitre 2.

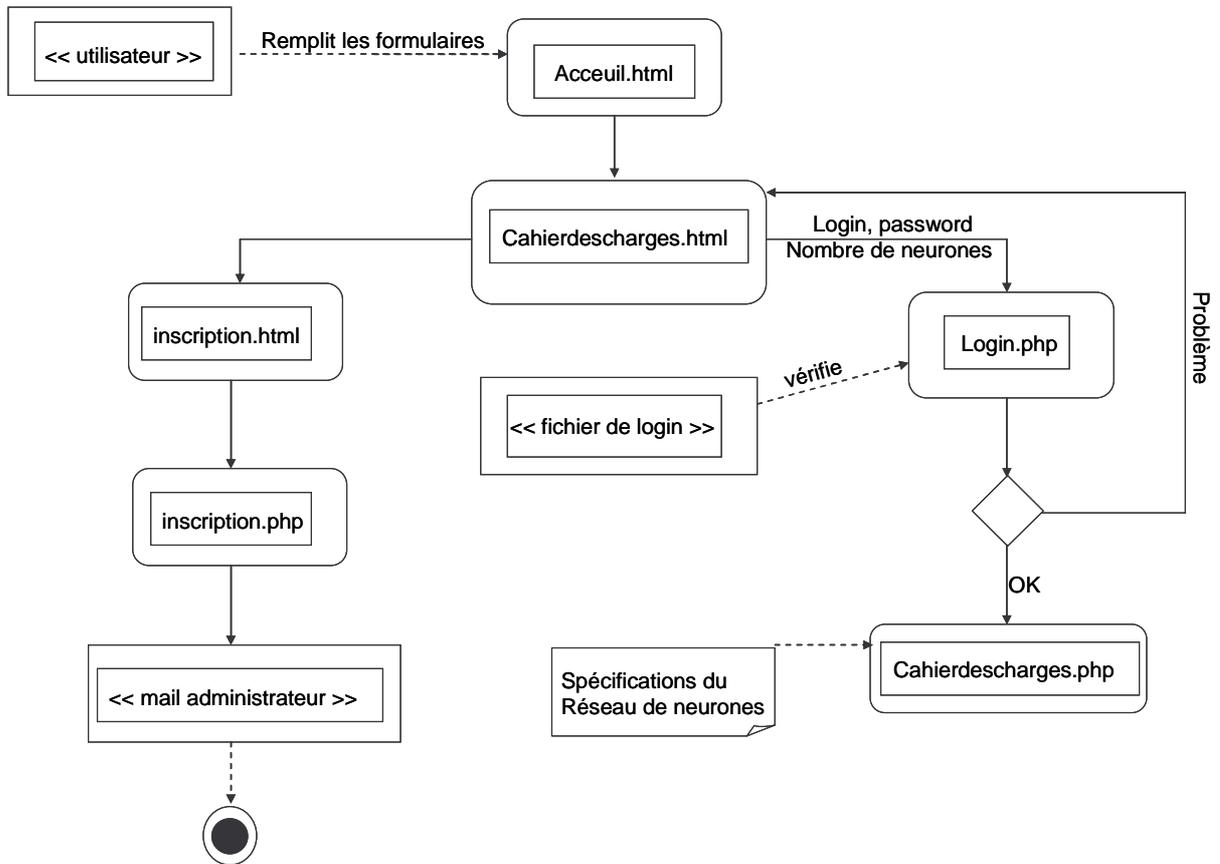


Figure 3.19. Diagramme UML d'activités des choix du client

La figure 3.19 décrit les différentes étapes que le client doit effectuer. Pour chaque étape, un fichier PHP est utilisé.

Pour proposer son cahier des charges au concepteur, le client doit préalablement s'être inscrit. Un mail est alors envoyé à l'administrateur pour demander l'accès à l'entrée des spécifications.

Une fois inscrit, le client peut rentrer son cahier des charges. Ci-après, une capture d'écran de l'entrée du choix des conductances pour chaque neurone.

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 nF/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 3.20. Capture d'écran du formulaire d'entrée des spécifications

Le formulaire possède une interface en anglais et en français. Nous n'avons choisi d'afficher ici que des captures d'écran en anglais.

La figure suivante 3.21 décrit les différentes étapes de la saisie des spécifications avec pour sortie un fichier de cahier des charges qui résume toutes les spécifications et qui est indexé au nom du client et à l'horaire de saisie.

Le client rentre les différents courants ioniques associés à chaque neurone (cahierdescharges.php), puis il rentre les valeurs des paramètres de ces courants (courant.php). Le compte-rendu est alors affiché (parametres.php) et ces spécifications sont enregistrées dans le répertoire donnees_clients.

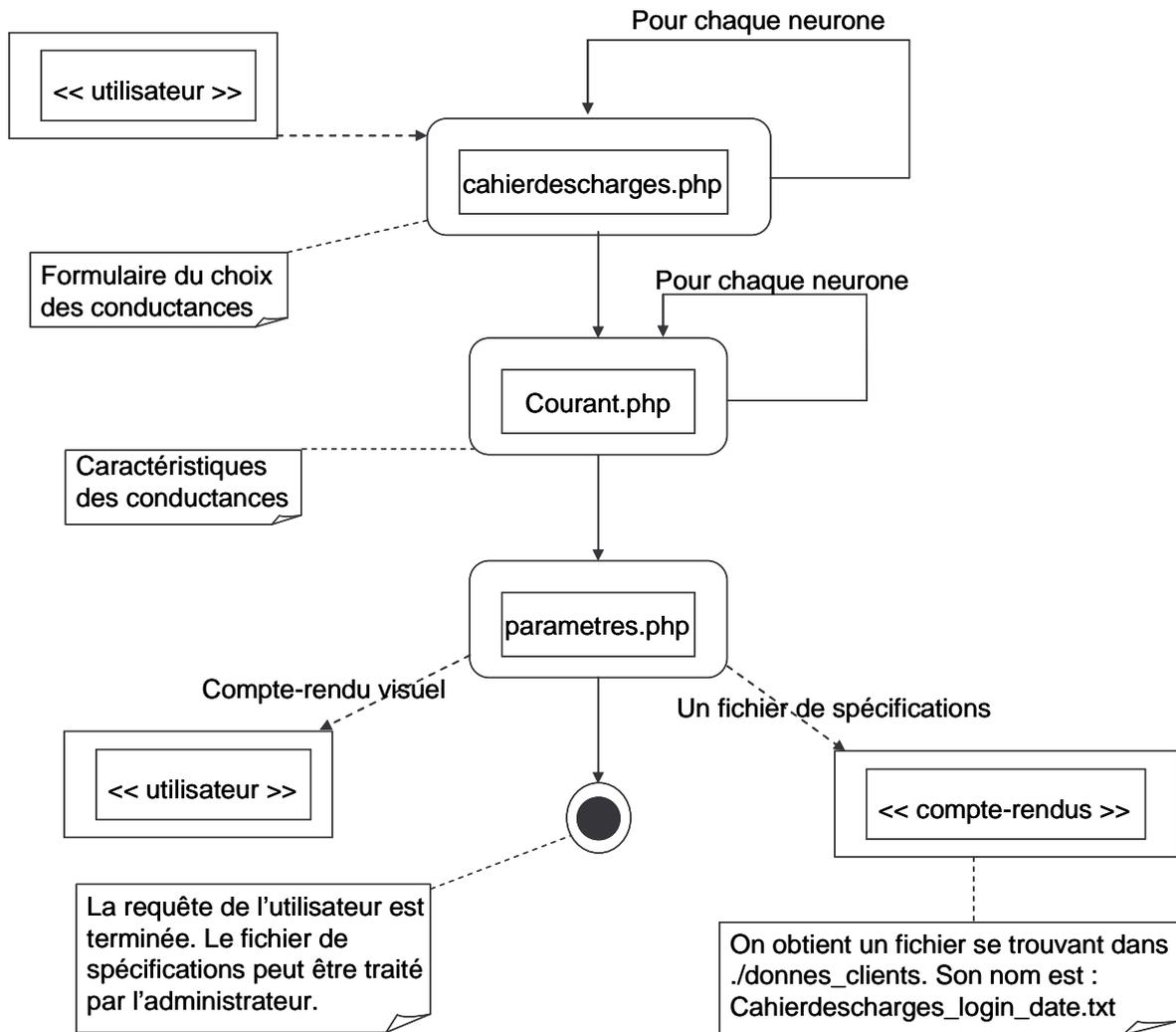


Figure 3.21. Diagramme UML d'activité de toutes les étapes du formulaire

V.2.2 Choix du concepteur, création des requêtes et exploration de la base de données

Une fois que les spécifications ont été remplies par le client, le concepteur les reçoit. Il choisit alors les différents facteurs de conversion (voir chapitre 2) entre les valeurs biologiques et les valeurs électriques mais aussi la technologie utilisée et la tension d'alimentation voulue.

Les requêtes sont alors créées et l'exploration de la base de données commence.

La figure 3.22 représente les différentes étapes entre les spécifications du client et la création des requêtes.

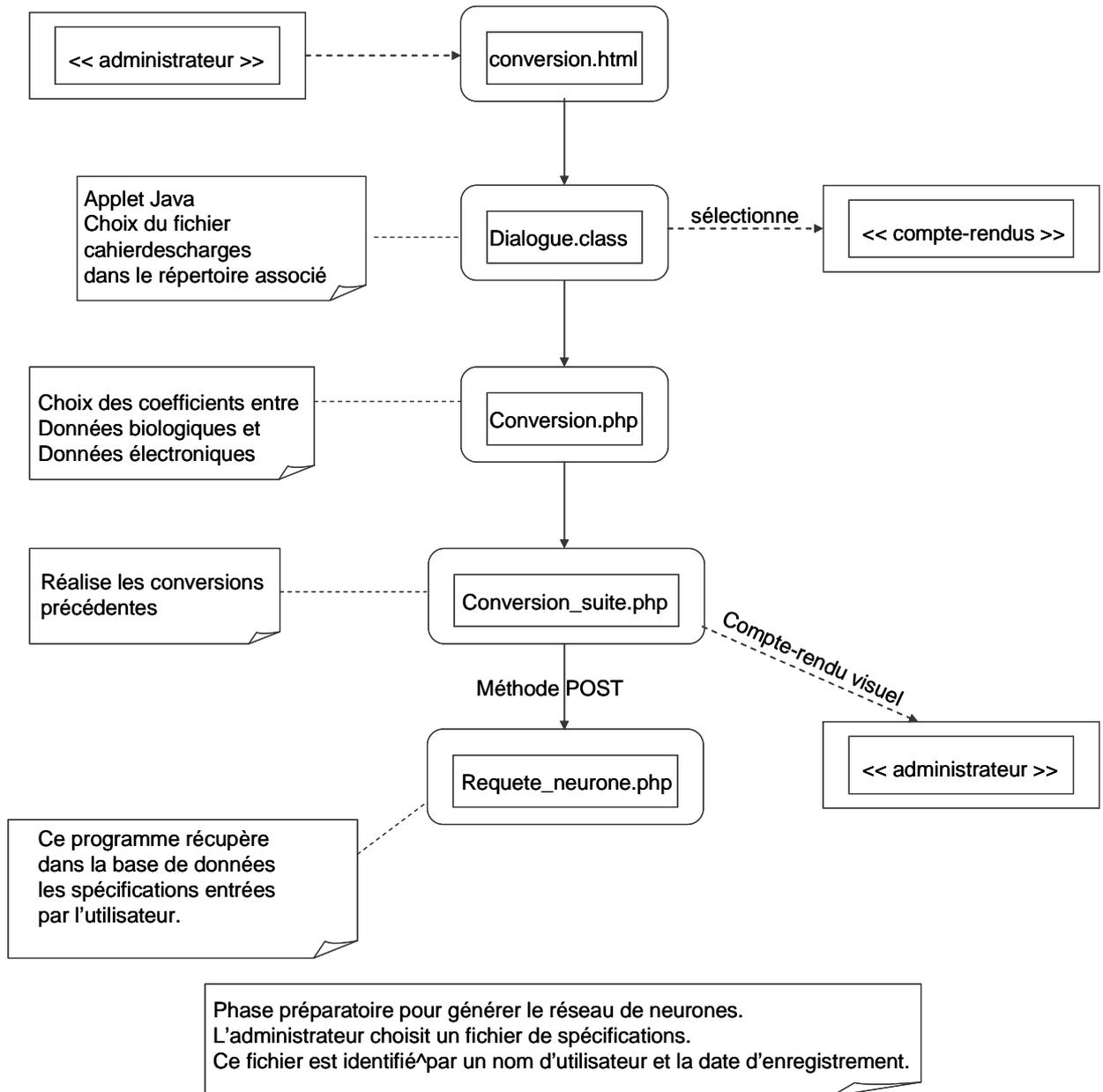


Figure 3.22. Diagramme UML d'activités de la création des requêtes

L'exploration s'effectue par une méthodologie Top-Down (pour notre cas d'étude, du neurone aux fonctions mathématiques en passant par les courants ioniques). Le critère le plus important dans le choix des IPs est le domaine de validité. Ainsi tous les IPs sont triés et sélectionnés par leur technologie, leur tension d'alimentation, puis leur domaine de validité.

V.2.3 Récapitulatif et résultats

Quand l'exploration de la base de données est terminée, un récapitulatif de celle-ci est donné. Ceci permet au concepteur de visualiser les résultats. Des fichiers sont créés pour résumer les différents résultats et s'il existe un échec lors de l'exploration, il est alors signalé et identifié.

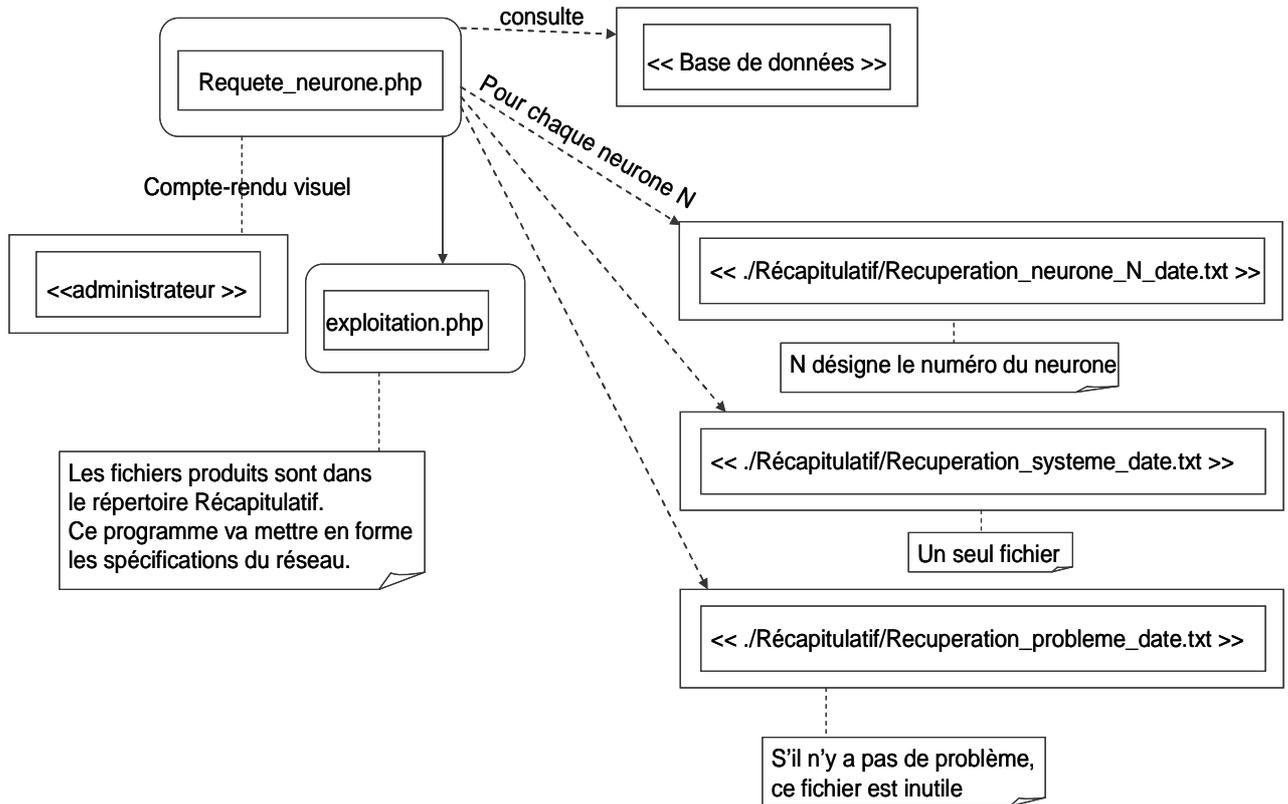


Figure 3.23. Diagramme UML d'activités des résultats

Pour chaque neurone, un fichier donnant les IPs sélectionnés pour le concevoir est engendré de même qu'un autre comportant les IPs de base pour concevoir un système. Un fichier précisant les échecs de l'exploration (s'ils existent) est créé.

Suivant le résultat de l'exploration, deux orientations existent. Si le diagnostic de réutilisation est de 100 % (figure 3.17), une netlist est alors créée contenant l'architecture complète du système ainsi obtenu. Sinon une identification précise du ou des échecs est donnée.

Une dernière étape existe, ce sont les comptes-rendus des résultats avec tous les liens des différentes vues des IPs sélectionnés pour concevoir le système. Ceux-ci sont créés via un fichier XML qui est ensuite transformé en fichier HTML pour une meilleure visualisation. Le

concepteur peut ainsi connaître les IPs sélectionnés et avoir accès aux différentes vues de ces IPs.

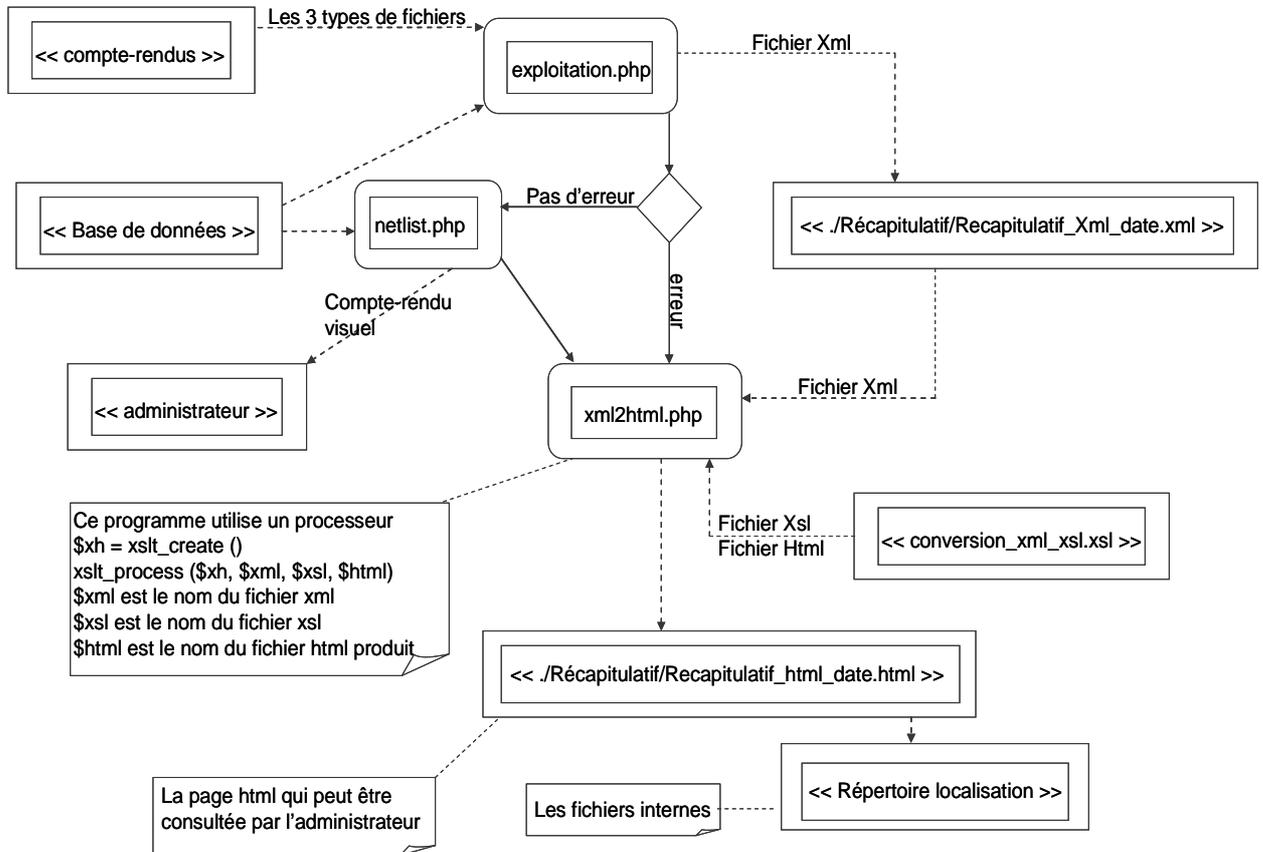


Figure 3.24. Diagramme UML d'activités de la création des fichiers XML et HTML

La figure 3.23 présente les différentes étapes concernant l'exploitation des résultats de l'exploration. Un processeur transformant les fichiers XML et XSL en extension HTML est utilisé. Ainsi à partir de tous les fichiers créés lors de l'exploration, un fichier HTML, récapitulant toutes ces informations, est développé et permet au concepteur de visualiser le système à concevoir.

Dans le dernier fichier récapitulatif, tous les IPs choisis sont donnés mais aussi les différentes capacités externes à implémenter et la surface totale du système. Une vérification est alors nécessaire pour confirmer ce résultat en utilisant les vues comportementales de ces IPs.

VI AIDE A LA RECONCEPTION EN CAS D'ECHEC

Si le diagnostic de l'exploration n'est pas de 100 %, une aide est alors donnée pour permettre au concepteur de concevoir les IPs manquants. Nous présentons ici les différentes voies à explorer pour le concepteur.

VI.1 Identification du problème

Lorsque l'exploration est terminée, des tableaux sont formés pour visualiser s'il existe les échecs. Le récapitulatif de l'exploration identifie clairement le(s) échec(s) en donnant leur(s) nature(s), et l'emplacement hiérarchique des fonctions pour lesquelles aucun IP n'a pu être trouvé.

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

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

Figure 3.25. Tableaux récapitulatif de l'exploration

Nous présentons maintenant les deux cas de figures pouvant provoquer un échec de l'exploration, c'est-à-dire un problème concernant les spécifications demandées ou alors un problème concernant la technologie et/ou la tension d'alimentation choisie.

VI.2 Problème concernant les spécifications demandées

Par exemple, la figure 3.25 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 de cette fonction neuromorphique sigmoïde pour atteindre les spécifications demandées.

Dans ce cas, il faut déterminer la différence entre les spécifications voulues et celles des IPs existants. Suivant l'importance de cette différence, l'IP le plus proche de ces spécifications pourra être modifié avec une variation mineure de ces paramètres de conception (résistances, capacités, courant de polarisation, etc.). Il sera alors possible de le réutiliser pour concevoir facilement un nouvel IP correspondant aux spécifications. Mais si la différence est trop grande, une nouvelle topologie devra être développée et un nouvel IP devra être conçu.

Un exemple concret du traitement de ce type d'échec sera développé dans le chapitre 4.

VI.3 Problème dû à la technologie ou à la tension d'alimentation

Si l'identification du problème concerne la technologie et/ou la tension d'alimentation, une solution est le redimensionnement de l'IP dans la technologie souhaitée avec une conservation des spécifications. Dans le chapitre 5, nous décrivons une méthodologie de redimensionnement lors d'une migration technologique ou lors d'un changement de tension d'alimentation. Cette méthodologie peut alors être appliquée pour concevoir ce nouvel IP.

VII CONCLUSION

Ce chapitre détaille le développement et l'exploration de notre bibliothèque d'IPs analogiques.

Lors des chapitres précédents, nous avons mis en évidence l'intérêt d'un flot de conception analogique basé sur une bibliothèque d'IPs. Ainsi nous avons présenté notre définition d'un IP analogique et de son contenu. Une caractérisation précise de chaque cellule de notre système a été développée. Par la suite, ces différents IPs sont regroupés dans une base de données dont la structure facilite son exploration future. Puis nous avons décrit le principe et les outils de cette exploration. Toutes les étapes de la conception automatisée d'un système à partir de spécifications initiales ont été détaillées. En cas d'échec(s) de l'exploration, le problème est clairement identifié et des aides à la reconception sont proposées.

Afin de compléter ce chapitre, le manuel d'utilisation de notre base de données est accessible en annexe 2. Pour quantifier le travail concernant la création de l'interface du formulaire et de l'exploration de la base de données, le nombre de lignes écrites en PHP, SQL, XML et HTML est d'environ 13500.

Le chapitre suivant utilise notre bibliothèque d'IPs pour concevoir de manière automatisée des systèmes neuromorphiques à partir de trois projets de conception.

CHAPITRE 4 :

CONCEPTION AUTOMATISEE D'UN ASIC NEUROMORPHIQUE : EXEMPLES D'APPLICATION

SOMMAIRE

I	INTRODUCTION.....	96
II	EXEMPLE 1 : CONCEPTION AUTOMATISEE DE L'ASIC GALWAY.....	96
II.1	SPECIFICATIONS.....	97
II.2	EXPLORATION DE LA BASE DE DONNEES	102
II.3	VERIFICATION.....	107
II.3.1	<i>Vérification électrique du système complet.....</i>	<i>107</i>
II.3.2	<i>Vérification par rapport au simulateur Neuron.....</i>	<i>109</i>
III	EXEMPLE 2 : IDENTIFICATION D'UN ECHEC ET CONCEPTION D'UN IP MANQUANT	110
III.1	SPECIFICATIONS.....	110
III.2	EXPLORATION ET IDENTIFICATION DU PROBLEME	112
III.3	RESOLUTION DU PROBLEME	113
III.4	AJOUT D'UN NOUVEL IP DANS LA BASE DE DONNEES	115
III.5	VERIFICATION.....	116
IV	EXEMPLE 3 : AJOUT D'UNE NOUVELLE FONCTION NEUROMORPHIQUE	117
IV.1	DIFFERENTES ETAPES.....	117
IV.2	CREATION ET VALIDATION DU MODELE FONCTIONNEL	118
IV.3	VALIDATION DE LA NOUVELLE FONCTION DANS LE SYSTEME ET CONCLUSION	119
V	CONCLUSION.....	122

I INTRODUCTION

Ce chapitre présente différentes applications utilisant la bibliothèque d'IPs que nous avons définie et créée. Le circuit utilisé pour l'application est un système neuromorphique défini dans le chapitre 2. La méthodologie et les outils utilisés, de la notion IP analogique à la conception d'une base de données, sont ceux décrits dans le chapitre 3.

Nous allons dans ce chapitre, effectuer une validation progressive de notre méthodologie selon trois exemples de projet de conception réaliste.

Tout d'abord, nous commençons ce chapitre par la conception automatisée d'un système neuromorphique (Galway). Nous présentons les diverses étapes de sa conception, depuis ses spécifications, en passant par l'exploration jusqu'à la vérification du système.

Le deuxième exemple permettra d'expliquer l'aide apportée au concepteur pour la résolution d'un problème lors d'un échec de l'exploration ainsi que l'ajout d'un nouvel IP.

Le troisième exemple décrira l'ajout d'une nouvelle fonction neuromorphique afin d'enrichir la base de données et de tester de nouvelles modélisations de neurones.

II EXEMPLE 1 : CONCEPTION AUTOMATISEE DE L'ASIC GALWAY

L'ASIC Galway a été décrit lors du chapitre 2. Il a déjà fait l'objet d'une conception *traditionnelle* (Bottom-Up et manuel) concrétisée par un prototype revenue de fonderie en juin 2006. Nous présentons ici sa conception automatisée à partir d'IPs sur la base des mêmes spécifications.

Nous rappelons la correspondance entre la hiérarchie d'un ASIC classique et celle d'un ASIC neuromimétique (Galway).

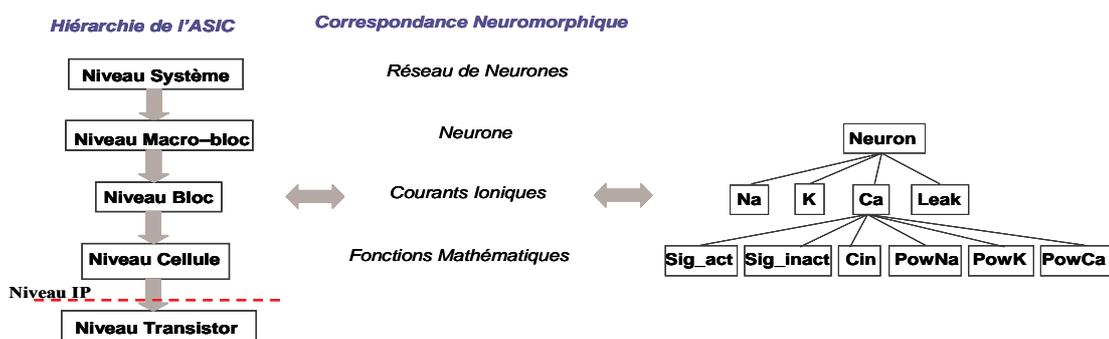


Figure 4.1. Présentation de la hiérarchie d'un ASIC neuromorphique

La figure 4.2 ci-dessous montre les différentes étapes de cette conception.

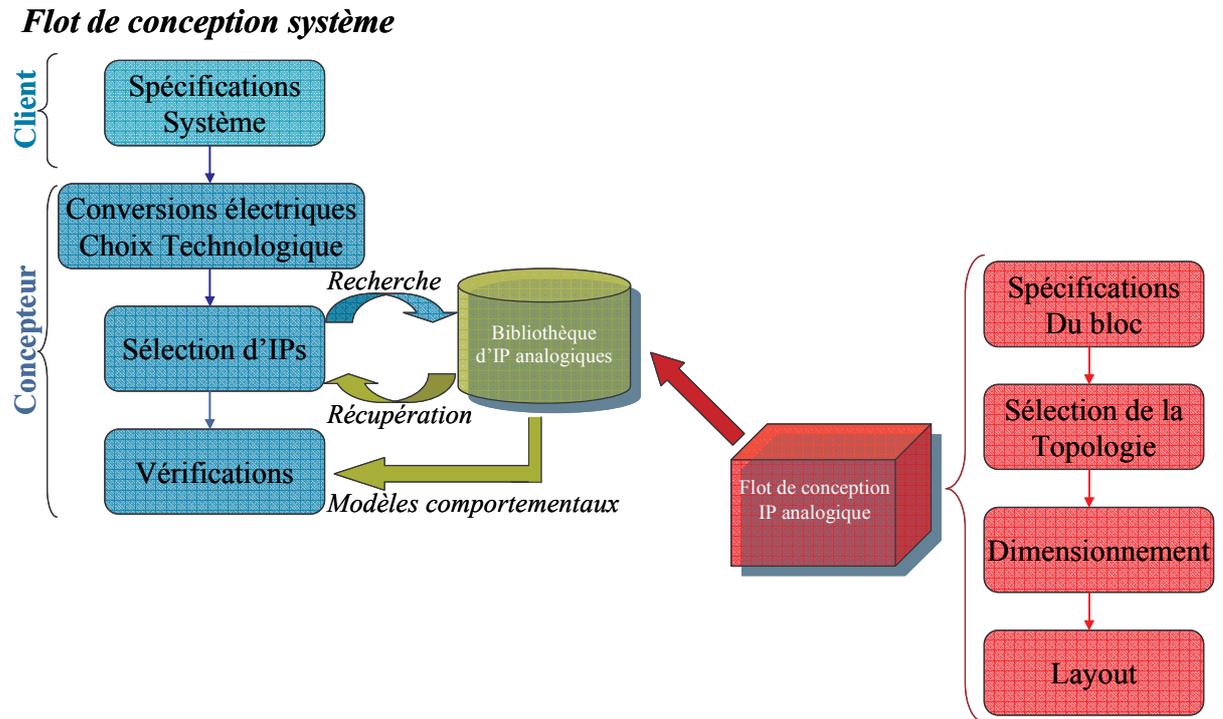


Figure 4.2. Différentes étapes de la conception du système à partir d'une bibliothèque d'IPs

Nous allons tout d'abord entrer les spécifications du système. Celles-ci vont se transformer en requêtes pour l'exploration de la bibliothèque d'IPs. La sélection d'IPs convenables pourra alors être faite, et une vérification comportementale de l'architecture proposée sera effectuée pour la valider.

II.1 Spécifications

La première étape de la conception de ce système est l'entrée des spécifications. Nous donnons tout d'abord les spécifications biologiques.

Cet ASIC doit contenir 5 neurones artificiels. Le premier neurone contient 3 conductances (Sodium, Potassium et Fuite), il peut simuler un neurone inhibiteur (FS). Les suivants ont une quatrième conductance pour l'ajout d'un courant Calcium ou Modulant ; ils peuvent simuler un neurone excitateur (RS). Le cinquième neurone (5c) a sa quatrième conductance doublée pour avoir à la fois un courant Calcium et un courant Modulant. Le neurone FS possède trois multisynapses et les neurones RS et 5c (possédant cinq conductances) en ont deux.

Les valeurs des paramètres fournies par nos collègues des neurosciences pour chaque neurone sont les suivantes:

TABLEAU 4.1. SPÉCIFICATIONS DU NEURONE FS

	Voffset_act	Vpente_act	Voffset_inact	Vpente_inact	Tau_act	Tau_inact	Vequi	gmax
Sodium	-37 mV	7,32 mV	-42 mV	-4,6 mV	0,03 ms	0,25 ms	50 mV	1,1 μ S
Potassium	-37 mV	11,38 mV			3 ms		-100 mV	2,2 μ S
Fuite							50 mV	0,33 μ S
Synapse					5 ms		52 mV	0,1 μ S

TABLEAU 4.2. SPÉCIFICATIONS DU NEURONE RS

	Voffset_act	Vpente_act	Voffset_inact	Vpente_inact	Tau_act	Tau_inact	Vequi	gmax
Sodium	-37 mV	7,32 mV	-42 mV	-4,6 mV	0,03 ms	0,25 ms	50 mV	1,1 μ S
Potassium	-37 mV	11,38 mV			3 ms		-100 mV	2,2 μ S
Modulant	-35 mV	11,4 mV	-35 mV	-10 mV	8 ms	8 ms	-100 mV	0,1 μ S
Fuite							50 mV	0,22 μ S
Synapse					4 ms		10 mV	0,1 μ S

TABLEAU 4.3. SPÉCIFICATIONS DU NEURONE A 5 CONDUCTANCES

	Voffset_act	Vpente_act	Voffset_inact	Vpente_inact	Tau_act	Tau_inact	Vequi	gmax
Sodium	-37 mV	7,32 mV	-42 mV	-4,6 mV	0,03 ms	0,25 ms	50 mV	1,1 μ S
Potassium	-37 mV	11,38 mV			3 ms		-100 mV	2,2 μ S
Calcium	-35 mV	11,4 mV	-35 mV	-10 mV	8 ms	8 ms	-100 mV	0,1 μ S
Modulant	-35 mV	11,4 mV	-35 mV	-10 mV	8 ms	8 ms	-100 mV	0,1 μ S
Fuite							50 mV	0,22 μ S
Synapse					4 ms		10 mV	0,1 μ S

Le client entre ses spécifications dans le formulaire qui se décompose en trois étapes. Tout d'abord la saisie du nombre de neurones voulus. Ensuite, le client définit les différents courants ioniques et le nombre de multi-synapses contenus dans chaque neurone. La surface de chaque neurone et sa capacité de membrane sont aussi à donner. Enfin, les paramètres de chaque courant ionique et de chaque synapse sont entrés.

Ci-dessous quelques captures d'écran de cette étape. Comme Galway est inscrit dans un projet européen, ces captures sont en anglais bien qu'il existe aussi une version en français.

Summary of the system specifications of levi

You already have an account

You want to have 5 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 ?

Neuron 2

Cmem_bio in uF/cm2

Area of neuron 2 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 ?

Neuron 3

Figure 4.3. Capture d'écran des choix des conductances

Characterization of ionic currents

Please give the characteristic of the ionic currents for each neuron

Voltages are in mV, time constants in seconds and conductances in μS

Neuron 1

Fill in the parameters of the multisynapse 1

Time constant

Vequi

gmax

Fill in the parameters of the multisynapse 2

Time constant

Vequi

gmax

Give the parameters of this current Sodium of the neuron 1

Voffset activation

Vslope inactivation

Time constant of activation

Time constant of inactivation

Vequi

gmax

Give the parameters of this current Potassium of the neuron 1

Voffset

Vslope

Time constant of activation

Vequi

gmax

Give the parameters of this current Fuite of the neuron 1

Vequi

gmax

Neuron 2

Figure 4.4. Capture d'écran de la saisie des paramètres des conductances

Les spécifications biologiques étant rentrées, c'est maintenant au tour du concepteur de commencer l'exploration de la bibliothèque. A partir de ce cahier des charges, les spécifications biologiques sont automatiquement transformées en spécifications électriques.

Pour cela, le concepteur choisit les différents facteurs de conversion ainsi que la technologie et la tension d'alimentation voulue (figure 4.5).

Parameters Conversion

Please give the conversion parameters

Voltages are in mV, time constants in seconds and conductances in uS

Choose the technology

Choose the supply voltage

Choose the Cmem conversion factor : KCmem

Choose the voltage conversion factor : Kv

Choose the time conversion factor : Kt

Figure 4.5. Capture d'écran des choix du concepteur

Le facteur de conversion du temps est égal à 1 car nous voulons travailler en temps réel. Celui de la tension est de 5 pour augmenter la dynamique et ainsi faciliter l'intégration de neurones sur Silicium. Enfin le facteur pour la capacité de membrane est égal à 0.044 car la capacité utilisée pour le circuit intégré est de 5 nF et la valeur biologique est de 0.22 nF ($0.22/5=0.44$). Tous ces facteurs déterminent alors ceux correspondant aux courants (K_I) et aux conductances (K_G).

Electrical Parameters

Cmem value for neuron 1 is 0.000968 uF/cm2

Neurons number is 5

Neuron 1 has 2 multisynapses

Parameters of the multisynapse 1 are :

Tau value is : 0.005

Vequi value is : 2.760

gmax value is : 2.273

Parameters of the multisynapse 2 are :

Tau value is : 0.030

Vequi value is : 2.735

gmax value is : 3.182

Neuron 1 has as ionic currents Sodium Potassium Fuite

Parameters of Sodium ionic current of neuron 1 are :

Voffset activation value is : 2.315

Voffset inactivation value is : 2.290

Vslope activation value is : 3.901

Vslope inactivation value is : 4.309

Tau activation value is : 0.000030

Tau inactivation value is : 0.000250

Vequi value is : 2.750

gmax value is : 25.000

Figure 4.6. Capture d'écran des spécifications électriques

Maintenant que les spécifications électriques existent, l'exploration de la base de données peut alors débuter.

II.2 Exploration de la base de données

Nous venons d'effectuer les deux premières étapes du flot de conception de notre système (figure 4.1), à savoir, les spécifications du système, les conversions électriques et le choix de la technologie. Nous passons maintenant à la prochaine étape, c'est-à-dire l'exploration de la base de données pour établir une sélection d'IPs afin de concevoir notre système.

A partir des spécifications biologiques rentrées et transformées en spécifications électroniques, les requêtes pour l'exploration de la base de données sont définies. Cinq requêtes ont été créées, une pour chaque neurone.

L'exploration s'effectue selon une méthodologie Top-Down en utilisant comme critère principal la dynamique d'entrée des IPs. En effet, rappelons que les spécifications biologiques correspondent à des valeurs typiques de paramètres de réglage de modèles de neurones (V_{equi} , g_{max} , V_{offset} , etc.) et que ces paramètres sont les grandeurs d'entrées des *cellules* IPs.

Voici les différentes étapes qui résument les résultats de l'exploration.

Tout d'abord, un récapitulatif des informations concernant la technologie et la tension d'alimentation est donné. Ensuite des tableaux sont créés représentant le diagnostic de l'exploration pour chaque neurone. La couleur verte signifie qu'un IP correspondant aux spécifications a été trouvé tandis que la couleur rouge caractérise un échec. Les cases laissées en blanc correspondent à des cas non applicables (par exemple la synapse ne contient pas de fonction d'activation, cette case est alors blanche).

Exploration Results

Data base connection: successfull
 Number of neurons: 5
 Technology: AMS BiCMOS 0.35 um
 Supply voltage: 5 V

Request of fs_neuron
 The choosen IP is ip1_fs_neuron . ip1_fs_neuron corresponds with the technology and with the supply voltage.

FS neuron exists and the choosen IP is ip1_fs_neuron

Diagnostic of neuron 1

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

■ ■ ■

Diagnostic of neuron 2

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

■ ■ ■

Diagnostic of neuron 3

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

■ ■ ■

Diagnostic of neuron 4

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

■ ■ ■

Diagnostic of neuron 5

	Activation	Inactivation	activation kinetic	inactivation kinetic	Power	Output Level
Sodium	█	█	█	█	█	█
Potassium	█		█		█	█
Leak						█
Slow conductance 1	█	█	█	█	█	█
Slow conductance 2	█	█	█	█	█	█
Synapse 1			█			█
Synapse 2			█			█
Synapse 3			█			█

Figure 4.7. Capture d'écran des diagnostics de l'exploration

Les diagnostics montrent que l'exploration de la base de données a permis de trouver 100 % du système à concevoir. Les IPs sélectionnés correspondent au cahier des charges demandé.

Une fois les IPs au niveau *Macro-bloc* sélectionnés, la table `ipsystem_links` (figure 4.8) permet de relier ces IPs avec l'IP final du neurone (sur la figure, `ip1_neuron_fs_final`) qui est composé du *Macro-bloc* correspondant aux spécifications (sur la figure, `ip1_fs_neuron`) et de deux IPs dépendants de celui-ci (sur la figure `ip1_fs_param` et `ip1_nim_topo_fs`) qui décrivent la partie numérique (entrées du circuit qui permettent d'agir sur les interrupteurs qui connectent les différents courants ioniques pour un neurone, stockage et rafraîchissement des paramètres analogiques).

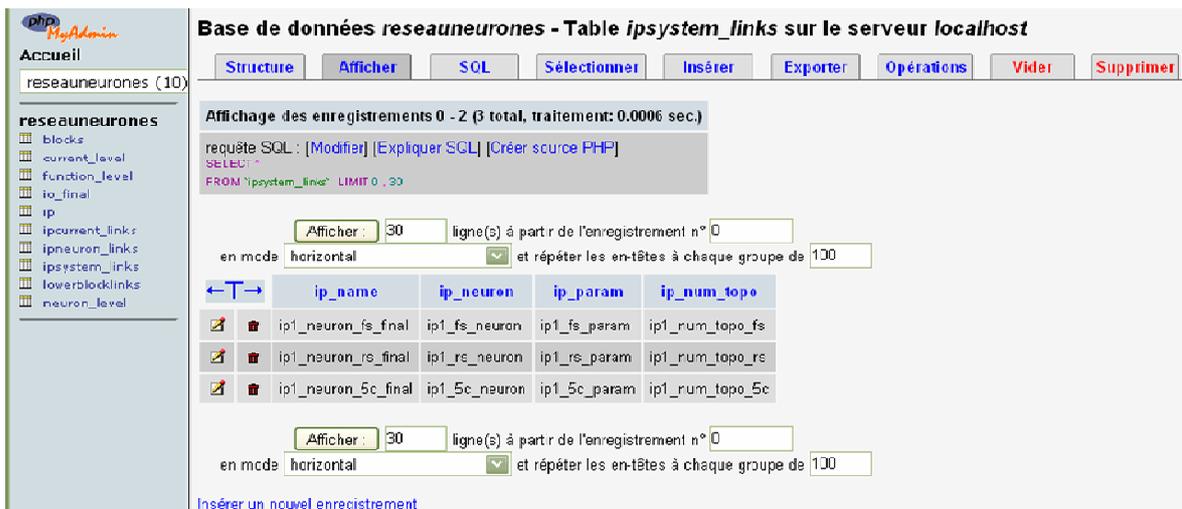


Figure 4.8. Table `ipsystem_links`

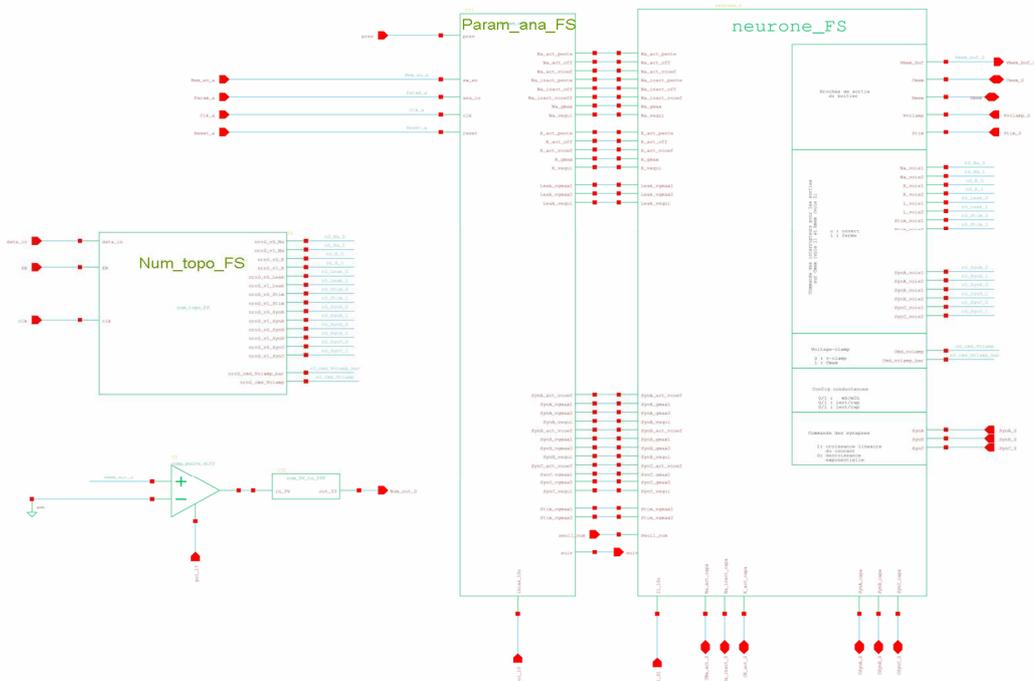


Figure 4.9. Architecture entre un *Macro-bloc* neurone et ces blocs génériques

La figure 4.9 présente l'architecture d'un neurone *final*.

Une netlist est alors créée pour permettre la connexion entre les différents neurones et ainsi concevoir le système final. Ce circuit final contient alors les différents neurones mais aussi un IP *Block_system* qui regroupe toutes les alimentations nécessaires pour le fonctionnement du circuit (tension d'alimentation, courant de polarisation, seuil numérique, etc.) au fonctionnement du système final. Cet IP est ajouté à chaque conception d'ASIC neuromimétique.

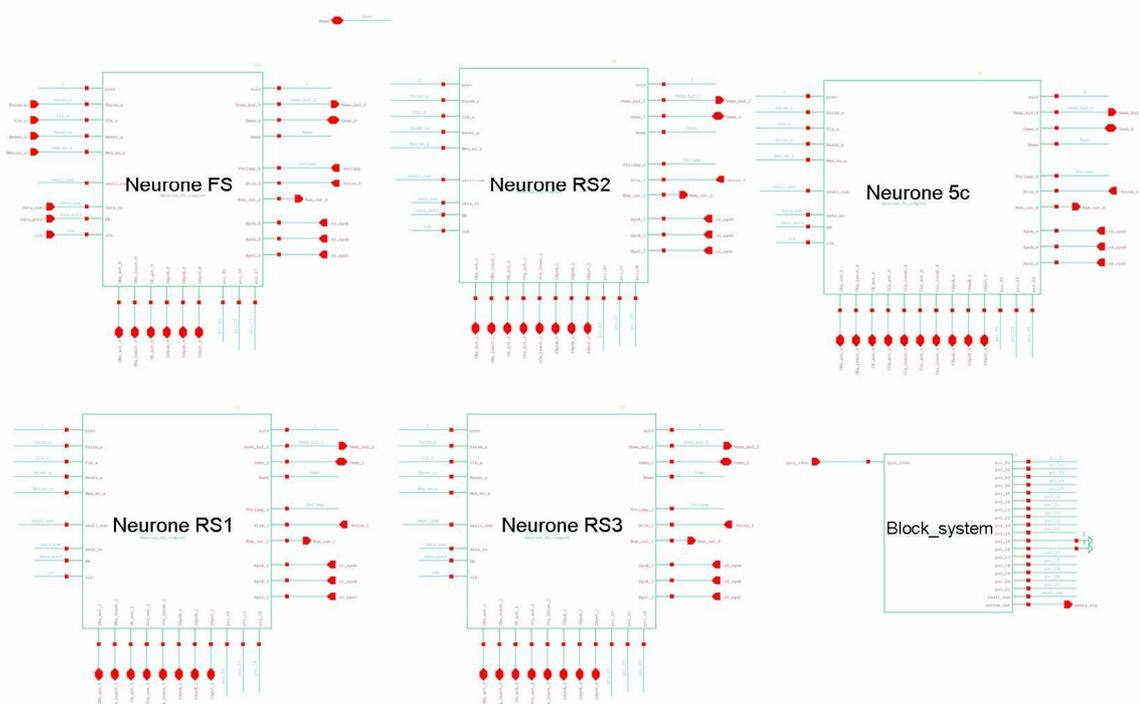


Figure 4.10. Vue d'ensemble du circuit final

Enfin, sous un format HTML, tous les IPs choisis avec leur contenu sont donnés et décrits ainsi que les valeurs des composants passifs (ici les capacités externes) qui sont calculées automatiquement. Par exemple, pour un neurone de type FS, 5 capacités sont nécessaires.

**** Neuron recovery **** : ip1_neuron_fs_final

Level	0
values	0
symbol	Facets_IP/Heurone_FS_complet/symbol/symbol.cdb
connectical	pamina/neuron_fs_final/connectical.txt
functional_model	pamina/neuron_fs_final/functional.txt
behavioural_model	pamina/neuron_fs_final/behavioural.txt
schematic	pamina/neuron_fs_final/schematic/sch.cdb
layout	pamina/neuron_fs_final/layout/layout.cdb
technology	AMS BICMOS 0.35 um
supply_voltage	5
Nb_inputs	23
Nb_outputs	5
layout_area	887265
ip_cost	0
datasheet	ip1_neuron_fs_final/datasheet.pdf

act_Na capacitance

Valeur de la Capacite = 4.995004995005E-009

inact_Na capacitance

Valeur de la Capacite = 4.1625041625042E-008

act_K capacitance

Valeur de la Capacite = 4.995004995005E-007

synapse capacitance 1

Valeur de la Capacite = 1.2025012025012E-006

synapse capacitance 2

Valeur de la Capacite = 7.2150072150072E-006

[Return to the Top](#)

[Return to the Neuron](#)

**** Neuron recovery **** : ip1_neuron_rs_final

Figure 4.11. Récapitulatif et liens des IPs choisis

La figure 4.11 permet d'obtenir des informations sur les valeurs des capacités à implémenter, ainsi que l'ensemble des IPs récupérés pour concevoir un neurone. Des liens vers toutes les vues de cet IP sont disponibles.

Des simulations sont maintenant nécessaires pour la vérification du système, la dernière étape de ce flot de conception.

II.3 Vérification

II.3.1 Vérification électrique du système complet

Maintenant que le choix des IPs est effectué, des simulations doivent être faites pour la vérification du bon fonctionnement du système dans son ensemble, éventuellement dans plusieurs configurations.

Ces vérifications se font en 3 étapes :

- Vérification de la connectique
- Vérification comportementale
- Vérification mixte

La vérification de la connectique est effectuée en utilisant les vues connectiques des différents IPs composant le système. On peut ainsi s'assurer de la bonne connexion des blocs mais aussi vérifier les valeurs du courant de polarisation (on pourrait aussi vérifier la tension d'alimentation). Rappelons que nous pouvons contrôler le courant de polarisation en regardant la sortie de chaque cellule : si cette sortie est à 0 alors le test est négatif et si la sortie est à I_{pol} alors le test est positif. Cette vérification a été faite et aucun problème de connectique n'a été détecté.

La vérification comportementale permet d'analyser le fonctionnement du système en un temps de simulation moindre qu'avec des modèles au niveau transistor. Pour exemple, la simulation d'un neurone FS au niveau transistor prend 14 min 56 s pour 60 ms de simulation alors qu'avec le niveau comportemental, cette simulation ne prend que 2 min 17 s.

Ci-dessous, nous présentons des simulations du système complet. La première figure (figure 4.12) représente un neurone FS au niveau transistor et son modèle comportemental.

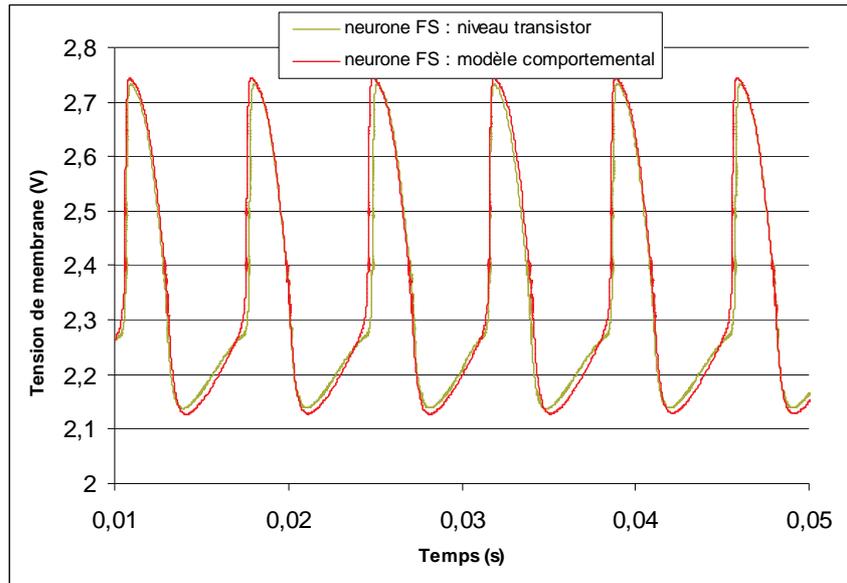


Figure 4.12. Comparaison entre le modèle comportemental et le modèle niveau transistor du neurone FS

Des simulations mixtes sont ensuite effectuées, celles-ci comprenant différentes modélisations (niveau transistor et vue comportementale). Ces simulations permettent de vérifier le comportement du système en se focalisant sur un bloc critique. Des simulations du système uniquement au niveau transistor ne peuvent être réalisées (temps de simulation trop grand et taille du fichier trop importante). L'intérêt et l'utilité des modèles VerilogA à différents niveaux d'abstraction sont ainsi prouvés.

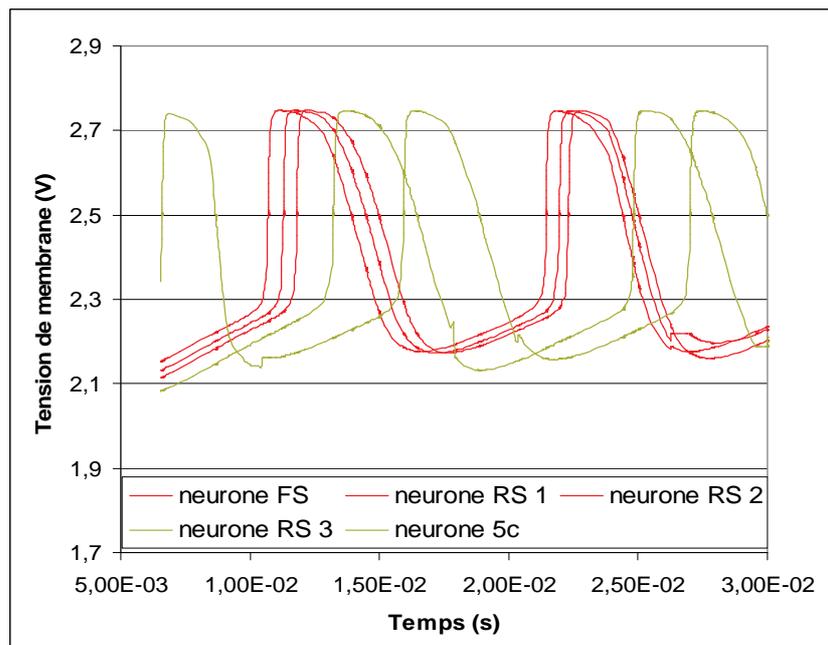


Figure 4.13. Simulation mixte des cinq neurones

La figure 4.13 représente une simulation transitoire du système avec les cinq neurones. Cette simulation est une simulation mixte utilisant des modèles comportementaux et des modèles au niveau transistor pour vérifier des blocs en particulier (ceux au niveau transistor).

La figure 4.14 représente la configuration de cette simulation.

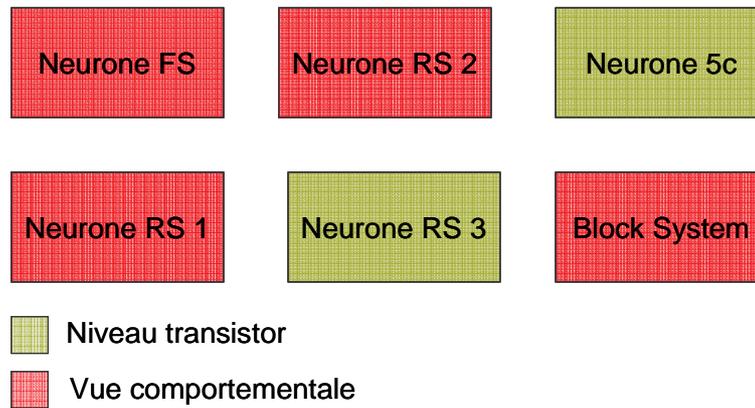


Figure 4.14. Configuration de la simulation mixte

Celle-ci permet la vérification d'une oscillation de chacun des neurones du système. Cette oscillation étant présente, nous vérifions maintenant les performances du système créé.

II.3.2 Vérification par rapport au simulateur Neuron

Pour déterminer les performances de l'ASIC, nous effectuons une analyse F(I) : calcul de la fréquence des potentiels d'actions de chaque neurone en fonction du courant de stimulation injecté en entrée.

En comparant cette analyse avec celle effectuée sous le logiciel Neuron (logiciel de référence pour les neurosciences voir chapitre 2) avec les modèles et les paramètres implémentés dans l'ASIC, nous pouvons conclure sur la validité de l'ASIC. La figure suivante représente la comparaison pour chacun des 5 neurone (FS : *Fast Spiking*, RS : *Regular Spiking*) de cette analyse.

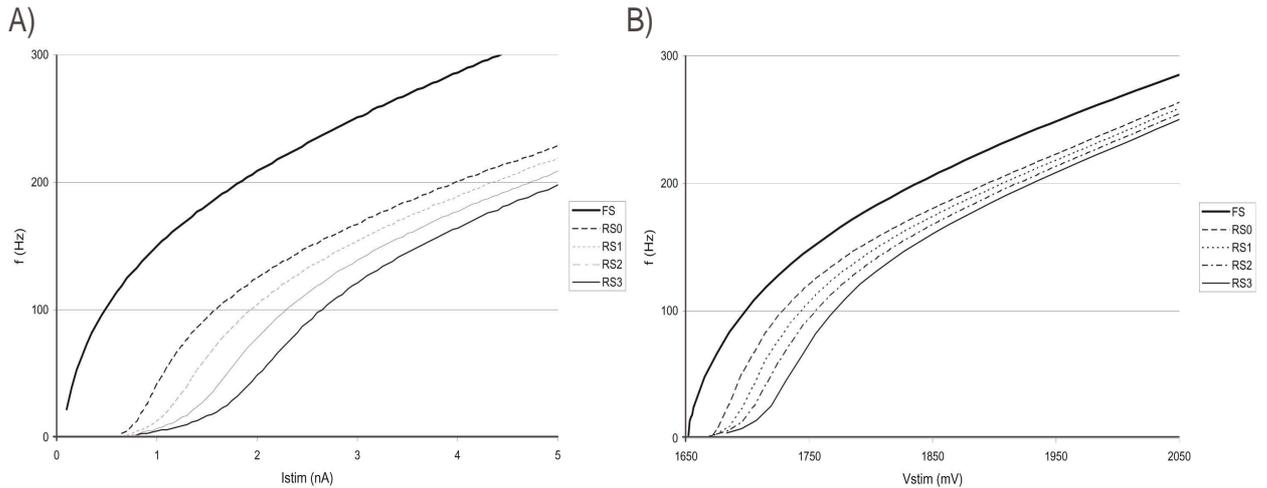


Figure 4.15. Comparaison des courbes Fréquences/Stimulation données par le logiciel Neuron et par l'ASIC conçu.

A) Simulations par Neuron.

B) Simulations de l'ASIC.

Nous pouvons, à la lumière de ces comparaisons, conclure sur le fait que ce système conçu de manière automatisée fonctionne convenablement. Avant le départ en fonderie, il reste néanmoins à placer les *Macro-blocs* layout issus de la bibliothèque et à les router manuellement.

III EXEMPLE 2 : IDENTIFICATION D'UN ECHEC ET CONCEPTION D'UN IP MANQUANT

Pour décrire toutes les possibilités de notre base de données et de son exploration, nous avons choisi un autre jeu de spécifications plausibles mais qui aboutit à un échec partiel de la recherche automatique. Ainsi nous verrons comment est identifié le problème et comment le résoudre.

III.1 Spécifications

Nous choisissons de concevoir un réseau de 2 neurones, l'un possédant 3 courants ioniques (neurone de type FS) et l'autre possédant 4 courants ioniques (neurone de type RS).

Voici les spécifications des paramètres qui composent ces neurones.

Les spécifications pour chaque neurone sont détaillées dans les tableaux 4.4 et 4.5.

TABLEAU 4.4. SPÉCIFICATIONS DU NEURONE FS

	Voffset_act	Vpente_act	Voffset_inact	Vpente_inact	Tau_act	Tau_inact	Vequi	gmax
Sodium	-39 mV	8,55 mV	-47 mV	-23 mV	0,03 ms	0,25 ms	40 mV	0,7 μ S
Potassium	-27 mV	12,6 mV			3 ms		-80 mV	1,4 μ S
Fuite							50 mV	0,5 μ S
Synapse					4 ms		60 mV	0,3 μ S

TABLEAU 4.5. SPÉCIFICATIONS DU NEURONE RS

	Voffset_act	Vpente_act	Voffset_inact	Vpente_inact	Tau_act	Tau_inact	Vequi	gmax
Sodium	-39 mV	8,55 mV	-47 mV	-6,6 mV	0,03 ms	0,25 ms	40 mV	0,7 μ S
Potassium	-27 mV	12,6 mV			3 ms		-80 mV	1,4 μ S
Modulant	-55 mV	10,4 mV	-38 mV	-20 mV	6 ms	6 ms	-100 mV	0,1 μ S
Fuite							50 mV	0,33 μ S
Synapse					6 ms		10 mV	0,1 μ S

Toutes ces spécifications biologiques sont transformées en spécifications électroniques. A partir de celles-ci, les requêtes sont créées.

La valeur de la pente pour l'inactivation du courant Sodium est passée de -4.6 mV à -23 mV. L'échec de l'exploration se situera à ce niveau et nous allons montrer comment elle est repérée et comment la résoudre.

III.2 Exploration et identification du problème

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

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

Figure 4.16. Diagnostics de l'exploration

Sur la figure 4.16 donnant les diagnostics de l'exploration, nous pouvons voir qu'il y a un échec lors de l'exploration de la base de données concernant l'inactivation du courant sodium du neurone 1. Concrètement, cela signifie qu'il n'existe pas dans la bibliothèque, d'IP *Sigmoïde* ayant une dynamique d'entrée balayant la spécification requise, dans la technologie et avec la tension d'alimentation préconisées par le concepteur. Trois questions peuvent alors se poser :

- L'IP existe-t-il mais dans une autre technologie ?
- L'IP existe-t-il mais avec une autre tension d'alimentation ?
- Si l'IP n'existe pas, sur quelle(s) spécification(s) a eu lieu l'échec ?

Nous pouvons répondre à ces questions car un récapitulatif de l'exploration est donnée et identifie clairement et précisément toutes les causes d'un échec. La figure 4.14 signale qu'il s'agit bien d'un échec sur la valeur de la pente d'inactivation.

Exploration Results

Data base connection: successfull
 Number of neurons: 2
 Technology: AMS BiCMOS 0.35 um
 Supply voltage: 5 V

Request of fs_neuron
 The choosen IP is ip1_fs_neuron . ip1_fs_neuron corresponds with the technology and with the supply voltage.

FS neuron does not exist

Issue with Sodium current
 The choosen IP is ip1_na . ip1_na corresponds with the technology and with the supply voltage.
 Ionic current research
 Sodium current does not exist
 Inactivation sigmoid issue
 The choosen IP is ip1_sigmoid_inact .
 research of Elementary function level
 Issue with Vslope inactivation value

Diagnostic of neuron 1

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

Figure 4.17. Récapitulatif de l'exploration et raisons de l'échec

La figure 4.17 montre que l'erreur ne se situe pas au niveau de la technologie et de la tension d'alimentation (*the choosen IP...corresponds with the technology and with the supply voltage*) mais qu'il y a un échec lors de l'inactivation de la sigmoïde du courant sodium au niveau de la valeur de V_{pente} (*Issue with Sodium current, Inactivation sigmoid issue, Issue with V_{slope} inactivation value*).

Nous venons donc d'identifier le problème, nous allons maintenant le résoudre.

III.3 Résolution du problème

L'échec est dû à la valeur de la pente d'inactivation qui est de -23 mV. Les IPs inclus dans la base de données ont des domaines de validité pouvant atteindre -21 mV. Comme la différence est faible, nous allons sélectionner l'IP qui possède le domaine de validité le plus proche et nous allons étudier sa vue *documentation*.

Voici l'équation du courant d'inactivation de la vue comportementale de l'IP sigmoïde :

$$I_{inact} = \frac{2 \cdot I_{h_10u}}{1 + e^{\frac{+10r \cdot I_{h_10u} \cdot (V_{mem} - V_{offset_h})}{R \cdot U_T \cdot I_{pente}}} \quad (4.1)$$

$$\text{Avec } I_{\text{pente}} = \frac{V_{cc} - K_{\text{pente}}}{R_{\text{conv}}} \quad (4.2)$$

$$\text{D'où } V_{\text{pente}} = \frac{(V_{cc} - K_{\text{pente}})RU_T}{10rR_{\text{conv}}I_{h_10u}} \quad (4.3)$$

En jouant sur les valeurs des résistances R, r ou R_{conv} , nous pouvons modifier la dynamique d'entrée V_{pente} . Ainsi nous créerons un nouvel IP dont le domaine de validité est translaté vers les valeurs voulues de la pente.

En changeant la valeur de la résistance R de 30 k Ω à 35 k Ω , la borne inférieure du domaine de validité de la pente de notre IP est passée de -21 mV à -26 mV.

Ainsi un nouvel IP peut être conçu à partir de ce schéma ; il faut lui associer les autres vues attendues (connectique, fonctionnel, etc.). Ce travail peut être fait en copiant le contenu de l'IP le plus proche au niveau des spécifications et en lui appliquant la modification clairement identifiée et localisée. La durée de cette étape est d'environ 1 heure pour un échec mineur.

La figure 4.18 compare ce nouvel IP avec l'ancien IP. Nous pouvons voir une simulation des deux IPs au niveau transistor et celle du modèle fonctionnel permettant la vérification de ce changement.

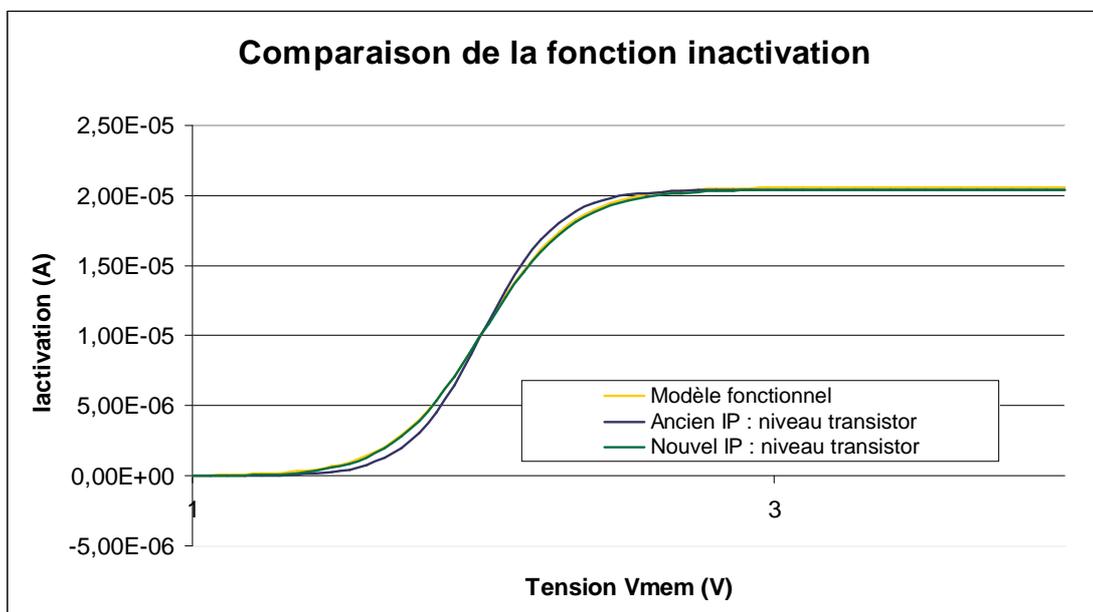


Figure 4.18. Comparaison entre les différents Ips et le modèle fonctionnel

Nous venons donc de créer un nouvel IP. Il nous reste maintenant à l'ajouter dans la base de données.

III.4 Ajout d'un nouvel IP dans la base de données

L'ajout de l'IP s'effectue aisément par le biais de l'interface de notre base de données. Comme nous venons de concevoir un IP appartenant au niveau hiérarchique *fonction mathématique* au sens neuromorphique, cet IP doit être ajouté aux différentes tables de la base de données. Le lien hiérarchique doit être défini, ses propriétés technologiques et ses spécifications aussi. Le temps moyen pour effectuer un ajout d'IP est inférieur à 5 minutes.

La figure suivante décrit une capture d'écran pour l'ajout d'un IP dans une table.

The screenshot shows a web-based database management interface. On the left, there is a navigation menu with a tree view under 'reseaneurones' containing items like 'blocks', 'current_level', 'function_level', 'io_final', 'ip', 'ipcurrent_links', 'ipneuron_links', 'ipsystem_links', 'lowerblocklinks', and 'neuron_level'. The main area displays a table of records with the following columns: 'name_ip', 'Voffset_act_inf', 'Voffset_act_sup', 'Voffset_inact_inf', 'Voffset_inact_sup', 'Vslope_act_inf', and 'Vslope_act_sup'. The table contains 7 rows of data. Below the table, there are controls for displaying records, including a search bar, a dropdown menu for 'horizontal', and a 'Version imprimable' link.

	name_ip	Voffset_act_inf	Voffset_act_sup	Voffset_inact_inf	Voffset_inact_sup	Vslope_act_inf	Vslope_act_sup
	ip1_sigmoid_act	1.6	3.2	0	0	1.7	4.7
	ip1_sigmoid_inact	0	0	1.6	3.2	0	0
	ip1_output_level_kna	0	0	0	0	0	0
	ip1_kinetic_gene	0	0	0	0	0	0
	ip2_kinetic_exp	0	0	0	0	0	0
	ip2_output_level_ca	0	0	0	0	0	0
	ip3_output_level_leakzyb	0	0	0	0	0	0

Figure 4.19. Capture d'écran de l'interface graphique permettant l'ajout d'un nouvel IP

Une autre possibilité existe concernant l'ajout d'IPs. Dans le cas d'un ajout d'un grand nombre d'IPs, nous n'effectuons pas la même démarche. Nous créons un fichier texte des informations relatives à tous les IPs. Ensuite notre base de données permet l'insertion de ces informations via le fichier texte. Ainsi toutes les données du fichier texte sont instantanément incluses dans la base.

L'annexe 2 (Manuel d'utilisation) commentera plus en détails cette étape.

L'IP conçu venant d'être ajouté à la base de données, nous pouvons maintenant relancer une nouvelle exploration.

III.5 Vérification

Maintenant qu'un nouvel IP correspondant aux spécifications attendues a été créé puis intégré dans la base de données, nous pouvons relancer notre exploration et concevoir ce système.

Voici les diagnostics de cette exploration.

Diagnostic of neuron 1

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

Diagnostic of neuron 2

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

Figure 4.20. Diagnostics de la nouvelle exploration

La figure 4.20 montre que les tableaux de diagnostics sont tous valides, l'exploration est donc positive à 100 %. L'échec dû à la valeur de la pente de la sigmoïde d'inactivation a été corrigé. Une netlist est alors créée et tous les IPs composant ce système peuvent être extraits de la bibliothèque.

Nous venons donc de valider ce système grâce à l'implémentation d'un nouvel IP correspondant aux spécifications requises. Nous allons maintenant montrer la flexibilité de la base de données avec l'ajout d'une nouvelle fonction n'ayant jamais été intégrée dans les ASICs des générations précédentes.

IV EXEMPLE 3 : AJOUT D'UNE NOUVELLE FONCTION NEUROMORPHIQUE

Nous proposons dans cette partie de démontrer la flexibilité de notre base de données. En effet, cette base peut être modifiée suivant les nouvelles orientations et modélisations des neurosciences computationnelles. Nous décrivons ici les différentes étapes à effectuer pour ajouter une nouvelle fonction. Nous montrerons ainsi comment cette base de données peut servir de plateforme d'expérimentation pour de nouvelles idées de conception et ainsi nous concluons quant à la pertinence de cette nouvelle fonction.

IV.1 Différentes étapes

Quelques étapes sont nécessaires pour l'ajout d'une fonction :

- 1ère étape : Création du modèle fonctionnel associé à la fonction mathématique désirée
- 2ème étape : Simulation du modèle fonctionnel de la fonction
- 3ème étape : Simulation du système complet avec le modèle fonctionnel de la nouvelle fonction
- 4ème étape : Conception du circuit et du modèle comportemental associé de la fonction
- 5ème étape : Ajout de l'IP correspondant à la nouvelle fonction dans la base de données

Les deux premières étapes servent à la création et à la validation du modèle fonctionnel de la fonction à implémenter. La troisième permet la validation par rapport au système. C'est à cette étape que le concepteur peut abandonner l'idée de cette nouvelle fonction si les résultats ne sont pas concluants. Enfin les deux dernières étapes concernent l'intégration de cette fonction sous la forme d'un nouvel IP dans la base de données.

Nous proposons d'illustrer cette méthodologie par un exemple.

Une manière d'approcher au mieux le comportement physiologique d'un neurone consiste à proposer un modèle de cinétique à deux constantes de temps alors que les anciens modèles possédaient une cinétique constante. Nous allons donc vérifier l'intérêt de cette nouvelle orientation, puis nous montrerons l'implémentation de cette nouvelle fonction cinétique à 2 constantes de temps.

IV.2 Création et validation du modèle fonctionnel

La première étape est la création du modèle fonctionnel associé à cette nouvelle fonction cinétique.

Pour exemple, voici la fonction cinétique que doit réaliser l'inactivation du courant sodium :

$$\text{Tau}_{\text{inact_Na}} = 3 \text{ ms si } V_{\text{mem}} > 0$$

$$\text{Tau}_{\text{inact_Na}} = 0.25 \text{ ms si } V_{\text{mem}} < 0$$

Alors que la fonction cinétique existante pour cette fonction était :

$$\text{Tau}_{\text{inact_Na}} = 0.25 \text{ ms quelque soit } V_{\text{mem}}$$

Nous avons alors conçu ce nouveau modèle fonctionnel de cinétique en VerilogA, son contenu est disponible en annexe. Voici une simulation de cette fonction (figure 4.21) qui vérifie le changement de constante de temps selon la valeur de la tension de membrane V_{mem} . La constante de temps de montée est au début à 3 ms puis elle passe à 0.25 ms car la tension de membrane vient d'être négative.

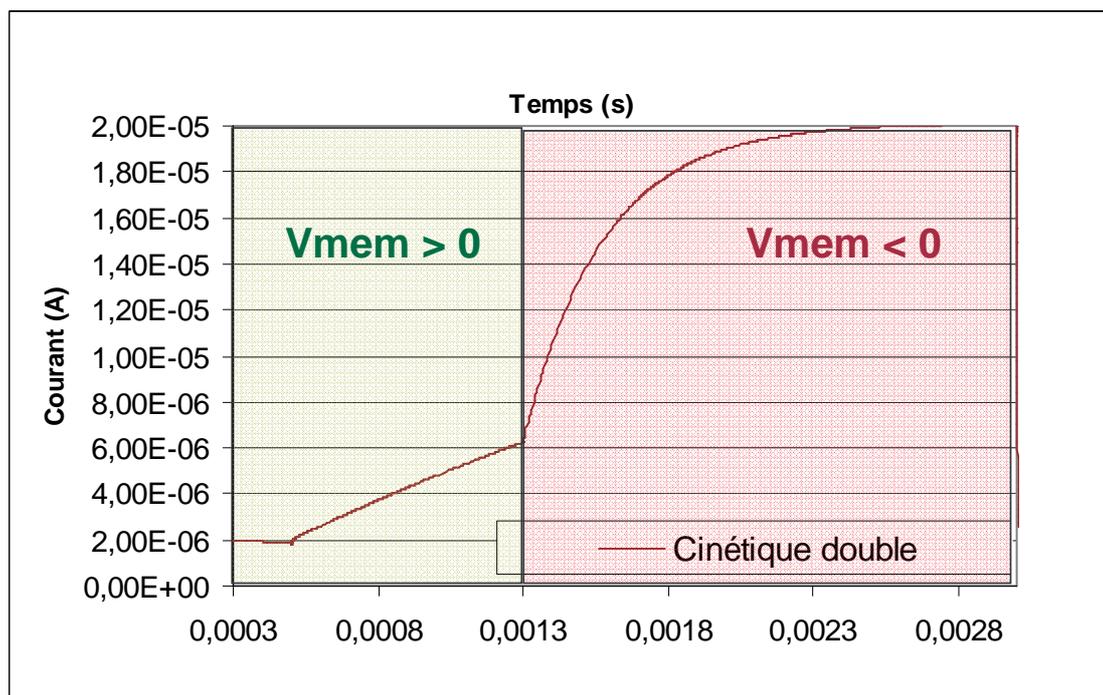


Figure 4.21. Simulation du modèle fonctionnel de la nouvelle fonction cinétique

Nous pouvons maintenant vérifier le comportement du système avec cette nouvelle fonction.

IV.3 Validation de la nouvelle fonction dans le système et conclusion

Nous allons maintenant vérifier le fonctionnement de notre modèle en l'incluant dans le système complet. Nous effectuons alors des comparaisons entre le fonctionnement du système avec une cinétique constante et avec une cinétique double.

Nous avons ajouté une fonction cinétique double pour le courant Sodium. Voici la différence entre ces deux courants Sodium.

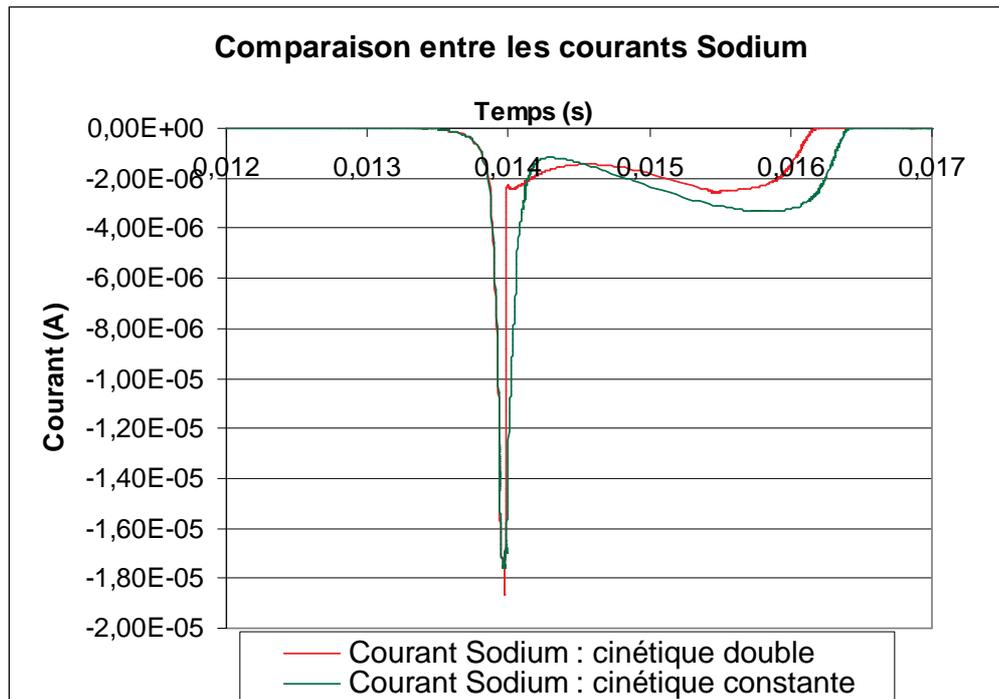


Figure 4.22. Comparaison entre les courants Sodium avec des cinétiques différentes

Cette figure décrit la différence entre un courant sodium possédant une cinétique constante et un autre avec une cinétique double. La réponse transitoire diffère légèrement entre ces deux courants. Pour permettre d'apprécier plus précisément les conséquences de ce changement de cinétique, nous effectuons des simulations au niveau du neurone. Ceci permettra d'évaluer le changement de réponse du neurone au niveau de la fréquence et de la forme des potentiels d'action.

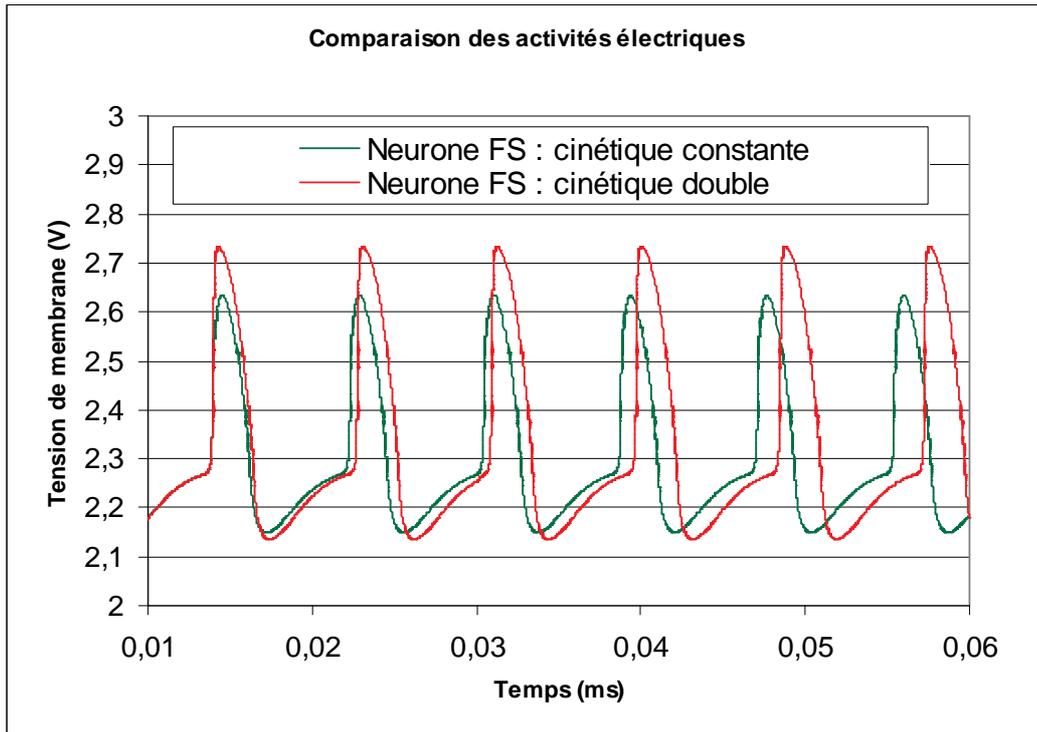


Figure 4.23. Comparaison de l'activité électrique de deux neurones possédant des cinétiques différentes

Nous apercevons sur cette figure 4.23 que le fait de modifier la cinétique diminue la fréquence d'oscillation du neurone et modifie la morphologie du potentiel d'action. Nous allons vérifier ces informations en effectuant les mêmes simulations sous le logiciel Neuron.

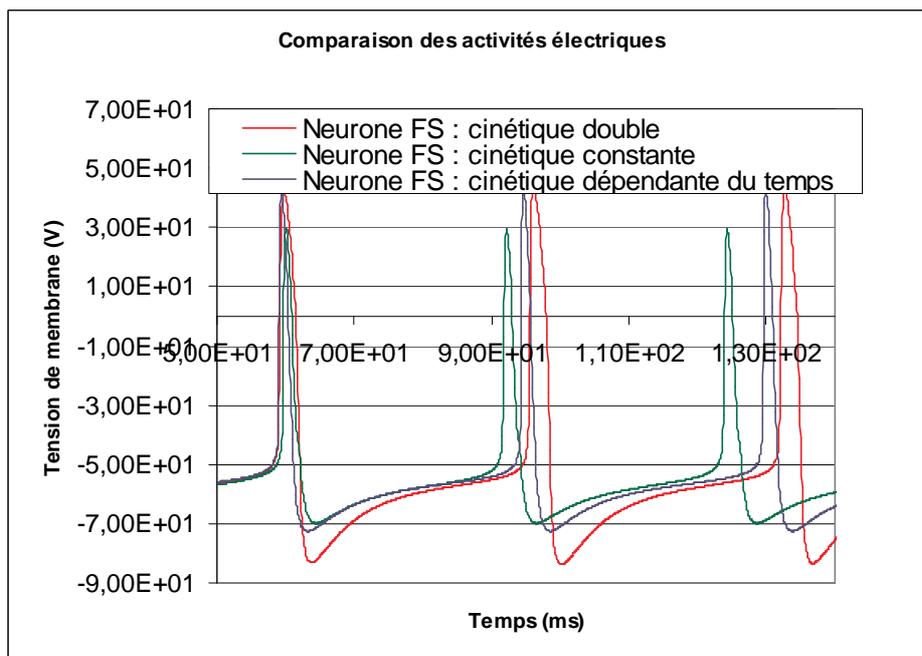


Figure 4.24. Comparaison des activités sous le logiciel Neuron

Cette figure 4.24 décrit les mêmes résultats que précédemment, c'est-à-dire une diminution de la fréquence pour le neurone possédant une cinétique double et une modification de la morphologie (le potentiel d'action a une amplitude plus grande). Nous avons aussi ajouté la courbe d'un neurone possédant une cinétique variable ce qui est le plus ressemblant par rapport au modèle biologique. Nous pouvons alors vérifier quelle cinétique (constante ou double) est la plus proche de modèle biologique ; l'orientation vers une cinétique double semble la plus pertinente.

Le modèle fonctionnel de cette cinétique existe, il convient maintenant de créer le circuit associé puis de la caractériser. Son implémentation sera alors possible dans la base de données.

Ces simulations ont permis de déterminer l'importance du choix entre une cinétique constante et une cinétique variable. Comme l'intérêt de nos circuits réside dans le fait de la conception de neurones biologiquement réalistes, cette nouvelle orientation a un sens et devrait se concrétiser par la conception d'un nouvel IP et de son ajout dans la base de données. Il sera utile aux générations futures d'ASICs neuromorphiques.

Nous venons donc de montrer la flexibilité de la base de données et de prouver que cette méthodologie permet aussi de tester et de vérifier la pertinence de nouveaux choix de conception.

V CONCLUSION

Nous venons de présenter les différentes applications de notre méthodologie de conception à partir d'une bibliothèque d'IPs.

La première application, concernant la conception du système neuromorphique Galway, a permis de présenter les diverses étapes de cette conception automatisée : des spécifications, en passant par l'exploration jusqu'à la vérification du système.

La deuxième application a montré l'aide apportée au concepteur pour la résolution d'un problème lors d'un échec de l'exploration. Au travers d'un exemple, nous avons montré les différentes étapes de l'ajout d'un nouvel IP et ainsi résoudre l'échec présenté.

La troisième application a servi à démontrer la flexibilité de cette base de données. En effet en ajoutant une nouvelle fonction, la base de données démontre qu'elle peut être enrichie mais aussi modifiée assez facilement ce qui a pour effet de prolonger sa durée de vie et son utilité. Cette méthodologie permet alors de vérifier les nouveaux modèles proposés avant de commencer la conception du système.

Toutes ces applications nous ont permis de valider notre méthodologie. En effet, pour chaque exemple, le résultat est positif. Le temps de conception d'un ASIC neuromimétique est fortement réduit. L'aide à la réutilisation pour le concepteur a été effectué et a permis la conception d'un nouvel IP. Enfin, une nouvelle orientation de modélisation a été testée et a été validée, un nouvel ASIC neuromorphique possédant cette modélisation sera bientôt envoyé en fonderie.

Nous allons maintenant présenter une méthodologie de redimensionnement pour les circuits CMOS lors d'un échec de la technologie et/ou de la tension d'alimentation. Cette méthode permet alors au concepteur, une aide pour la réutilisation de circuits déjà conçus, en utilisant un ancien IP et en effectuant une migration technologique.

CHAPITRE 5 :

LE REDIMENSIONNEMENT :

UN ATOUT MAJEUR DANS LA REUTILISATION

SOMMAIRE

I	INTRODUCTION.....	124
II	ETAT DE L'ART.....	124
II.1	VUE D'ENSEMBLE DES DIFFERENTS TRAVAUX CONCERNANT LE REDIMENSIONNEMENT	124
II.2	METHODOLOGIE DE C. GALUP-MONTORO ET M.C. SCHNEIDER [GALU 2002]	125
II.3	ETUDE D'ACOSTA, SILVEIRA ET AGUIRRE [ACOS 2002]	128
II.4	TRAVAIL DE SAVIO, COLALONGO ET KOVACS-VAJNA [SAVI 2004][SAVI 2006].....	128
II.5	REALISATIONS INDUSTRIELLES	130
II.5.1	<i>Neocell et Neocircuit.....</i>	<i>130</i>
II.5.2	<i>Barcelona Design.....</i>	<i>130</i>
III	DESCRIPTION DE LA METHODOLOGIE PROPOSEE	131
III.1	PLAN DE LA METHODOLOGIE.....	131
III.2	DEFINITION DES FACTEURS DE REDIMENSIONNEMENT [LEVI 2006A].....	132
III.3	ANALYSE DE LA 3EME ETAPE DE LA METHODOLOGIE	132
III.4	DISCUSSION SUR LE CHAMP D'APPLICATION DE LA METHODOLOGIE	133
IV	APPLICATION SUR DES CIRCUITS LINEAIRES.....	135
IV.1	L'OTA.....	135
IV.2	PREMIERE APPLICATION DE LA METHODOLOGIE : CONSERVATION DU GAIN EN TRANSCONDUCTANCE 135	
IV.2.1	<i>Redimensionnement.....</i>	<i>135</i>
IV.2.2	<i>Résultats</i>	<i>137</i>
IV.2.3	<i>Evaluation de la robustesse.....</i>	<i>139</i>
IV.3	DEUXIEME APPLICATION : CONSERVATION DU GAIN EN TRANSCONDUCTANCE ET REDUCTION DE LA CONSOMMATION EN PUISSANCE	141
IV.3.1	<i>Redimensionnement.....</i>	<i>141</i>
IV.3.2	<i>Résultats</i>	<i>142</i>
IV.4	CONCLUSION CONCERNANT LES EXEMPLES LINEAIRES	143
V	APPLICATION SUR DES CIRCUITS NON-LINEAIRES	144
V.1	L'OSCILLATEUR EN ANNEAU	144
V.2	L'INVERSEUR CMOS.....	145
V.3	REDIMENSIONNEMENT	145
V.4	RESULTATS DES SIMULATIONS	146
V.5	EVALUATION DE LA ROBUSTESSE.....	148
VI	CONCLUSION ET PERSPECTIVES	150

I INTRODUCTION

L'objectif d'un redimensionnement dans le contexte des IPs est de conserver les caractéristiques du circuit originel lors d'un passage d'une technologie à une autre. Le défi consiste aussi à optimiser simultanément la surface et la consommation.

Le redimensionnement entre dans le cadre de la réutilisation de l'expérience de conception. Il s'agit ici de faire face à une migration technologique en réutilisant une topologie de circuit déjà dimensionnée pour une technologie initiale.

Nous proposons dans ce chapitre une méthodologie de redimensionnement dédiée aux circuits CMOS. Le résultat du redimensionnement est donc attendu sous la forme de nouvelles valeurs des paramètres géométriques W , la largeur, et L , la longueur, de chaque transistor MOS.

Nous exposerons dans un premier temps un état de l'art des méthodologies existantes avec leurs avantages et leurs inconvénients, avant d'explicitier notre propre approche de redimensionnement. Nous l'appliquerons enfin sur des circuits linéaires (Operational Transconductance Amplifier : OTAs) et non-linéaires (oscillateurs en anneau).

II ETAT DE L'ART

II.1 Vue d'ensemble des différents travaux concernant le redimensionnement

Ces dernières années, de nombreux travaux de recherche ont été consacrés à la réutilisation de blocs A/M (analogique et mixte), incluant le redimensionnement. Les travaux de C. Galup-Montoro et M.C. Schneider [GALU 2000], [GALU 2001], [GALU 2002], proposent une méthode analytique et plusieurs stratégies selon le type de circuit utilisé. Le modèle de transistor MOS utilisé est le modèle ACM [CUNH 1998] et la méthode est testée sur des OTAs.

Cette méthode a été reprise dans deux directions de recherche. Les travaux [ACOS 2002] visent à conserver un nombre plus important de figures de mérite dans la procédure de redimensionnement et apportent des résultats expérimentaux. Les travaux [SAVI 2004], [SAVI 2006] ajoutent une procédure de réglage basée sur une boucle d'optimisation.

Le principal défaut de ces études est l'augmentation de la surface de la puce lors du redimensionnement des OTAs, alors que la migration s'effectue toujours vers une technologie plus fine.

Certaines sociétés de CAO [NEOL] [BARC] ont récemment commencé à offrir des outils et des services destinés à la réutilisation et à la migration technologique. [NEOL] et [FUNA 2000] utilisent également des approches basées sur des boucles d'optimisation et l'accumulation de simulations SPICE pour modifier les paramètres des circuits afin d'obtenir les performances voulues.

Tous ces travaux valident leurs résultats à partir d'un OTA. Par souci de comparaison, nous utiliserons aussi comme circuit d'application un OTA possédant des performances similaires à ceux utilisés dans ces différentes études. Le but d'un redimensionnement étant également de diminuer la surface de la puce, il convient de poursuivre les recherches dans cette direction pour éviter les problèmes apparus dans les travaux décrits.

Les paragraphes suivants exposent de manière plus détaillée les différents développements proposés dans la littérature.

II.2 Methodologie de C. Galup-Montoro et M.C. Schneider [GALU 2002]

La méthodologie de C. Galup-Montoro et M.C. Schneider repose sur des règles de redimensionnement provenant du modèle ACM (Advanced Compact Model) du transistor MOSFET. Ce modèle reprend les bases du modèle EKV [VITT 1990] mais n'utilise pas les courbes d'interpolation pour relier les régimes de faible et de forte inversion. L'équation du transistor NMOS en saturation pour le modèle ACM est :

$$I_D = I_S i_f \quad (5.1)$$

Avec I_D le courant de drain, I_S le courant de saturation.

$$I_S = \frac{\mu_n n C_{ox} W U_T^2}{2L} \quad (5.2)$$

Avec C_{ox} la capacité de l'oxyde, U_T le potentiel thermique, W la largeur du transistor, L la longueur du transistor, μ_n la mobilité des électrons et n le facteur de pente sous le seuil.

i_f est un courant normalisé dépendant du niveau d'inversion du transistor qui vérifie (5.3).

$$\frac{(V_G - V_{T0})}{n} - V_S = U_T (\sqrt{1 + i_f} - 2 + \ln(\sqrt{1 + i_f} - 1)) \quad (5.3)$$

Où V_G et V_S sont les tensions de grille et de source, et V_{T0} la tension de seuil.

A partir des équations (5.1), (5.2) et (5.3), le rapport de la transconductance g_m et du courant de drain est donné par :

$$\frac{g_m}{I_D} = \frac{1}{n.U_T} \frac{2}{(\sqrt{1+i_f} + 1)} \quad (5.4)$$

En envisageant le passage d'une technologie 1 à une technologie 2, cinq facteurs de redimensionnement sont définis :

$$K_V = \frac{V_{DD1}}{V_{DD2}} \quad K_L = \frac{L_{MIN1}}{L_{MIN2}} \quad K_\mu = \frac{\mu_1}{\mu_2}$$

$$K_{COX} = \frac{C_{OX2}}{C_{OX1}} \quad K_E = \frac{V_{E2}}{V_{E1}}$$

Avec V_{DD} la tension d'alimentation, L_{MIN} la longueur minimale du transistor dans la technologie, V_E la tension d'Early par unité de longueur. Remarquons que dans le cadre d'une migration vers une technologie plus fine, ces facteurs sont tous supérieurs à 1.

Les équations de redimensionnement sont déterminées de façon à conserver le gain en tension DC, le produit gain bande, la marge de phase et le rapport signal sur bruit maximum d'un OTA.

Deux stratégies de redimensionnement sont analysées dans ce travail : celle de la conservation du niveau d'inversion (conservation de i_f alors que tous les courants et les transconductances sont divisés par K_v^2) et celle de la proportionnalité aux longueurs minimales (où l'on diminue la longueur du transistor par K_L). Voici le tableau représentant les règles de redimensionnement des deux stratégies.

TABLEAU 5.1. REGLES DE REDIMENSIONNEMENT SELON LA STRATEGIE

Paramètres	Conservation du niveau d'inversion (1)	Proportionnalité des longueurs minimales (2)	
L	$K_{\mu}^{-0.5}$	K_L^{-1}	
W	$K_v^2 K_{ox}^{-1} K_{\mu}^{-0.5}$	$K_v^2 K_{ox}^{-1} K_L$	
		WI	SI
I_D	K_v^2	K_v^2	$K_{\mu} K_v^2 K_L^{-2}$
i_f	1	$K_{\mu} K_L^{-2}$	$K_{\mu}^2 K_L^{-4}$
Gain DC	K_E	$K_E K_L^{-1}$	$K_E K_L K_{\mu}^{-1}$
Surface	$K_v^2 K_{ox}^{-1} K_{\mu}^{-1}$	$K_v^2 K_{ox}^{-1}$	$K_v^2 K_{ox}^{-1}$
Consommation	K_v	K_v	$K_{\mu} K_L^{-2} K_v$

$$\text{Avec } \sqrt{1+i_{f2}} - 1 = \frac{K_{\mu}}{K_L^2} (\sqrt{1+i_{f1}} - 1).$$

WI correspond au régime de faible inversion (Weak Inversion) et **SI** celui de forte inversion (Strong Inversion).

A partir de cette méthodologie et en utilisant les deux stratégies, un OTA est redimensionné à partir d'une technologie $3 \mu\text{m} / 5 \text{V}$ vers une technologie plus fine : $0.35 \mu\text{m} / 3.3 \text{V}$.

Le tableau 5.2 décrit les résultats de la méthodologie appliquée sur un OTA Miller. Nous remarquons que la surface de l'OTA redimensionné augmente de 3 à 4 fois par rapport à l'OTA initial.

TABLEAU 5.2. REDIMENSIONNEMENT D'UN OTA

Technologie	Gain en tension (dB)	Marge de Phase (Degré)	GB (MHz)	Courant total (μA)	Surface
$3 \mu\text{m} / 5 \text{V}$	73	65	1	27	A
$0.35 \mu\text{m} / 3.3 \text{V}$ Stratégie (1)	77	63	0.91	60	$\approx 4 \text{ A}$
$0.35 \mu\text{m} / 3.3 \text{V}$ Stratégie (2)	74	60	0.90	14	$\approx 3 \text{ A}$

Même si elle permet d'obtenir des résultats satisfaisants pour les deux stratégies, la méthode n'est pas sans inconvénients. En effet il ne s'agit que de simulations, et pour certaines technologies à faibles longueurs minimales, les figures de mérites ne sont pas conservées durant le redimensionnement. Mais le principal défaut est l'augmentation de la surface de la puce lors d'une diminution de la tension d'alimentation. En effet si $K_v^2 K_{ox}^{-1} > 1$ alors la surface de la puce redimensionnée sera plus grande que celle de la puce initiale.

II.3 Etude d'Acosta, Silveira et Aguirre [ACOS 2002]

Cette étude propose une analyse plus détaillée de la méthodologie précédente, notamment par le biais de paramètres de performance qui n'étaient pas inclus dans le travail précédent [GALU 2002]. L'idée est d'étudier l'évolution de plusieurs paramètres de performance dont le Slew Rate d'un montage source commune et la fréquence du pôle d'un miroir de courant. En utilisant les équations de redimensionnement précédentes, la détermination de ces deux caractéristiques est possible.

Un OTA a alors été redimensionné, simulé et fabriqué sur silicium en technologie 0.8 μm . La technologie de départ est à 2.4 μm et celles d'arrivée sont à 0.8 μm et 0.35 μm . Les tests sur silicium prouvent la validité de la méthode. Les résultats montrent alors que l'on peut prévoir le Slew Rate et la réponse en fréquence. Concernant le Slew Rate, les résultats s'avèrent décevants, car il n'est pas conservé, contrairement aux objectifs affichés. Les résultats concernant la réponse en fréquence sont satisfaisants car cette méthodologie la préserve.

L'apport de ce travail est d'avoir analysé l'évolution de quelques figures de mérite et ainsi de décrire quelles sont celles qui sont conservées par la méthodologie de départ. La possibilité d'utiliser une procédure de réglage basée sur des simulations SPICE a aussi été étudiée, permettant ainsi de réduire les inconvénients de la méthodologie d'origine (augmentation de la surface et non-conservation de certaines figures de mérite). La procédure utilise le courant de polarisation comme paramètre d'ajustement.

II.4 Travail de Savio, Colalongo et Kovacs-Vajna [SAVI 2004][SAVI 2006]

Là encore la méthodologie de [GALU 2002] est reprise comme point de départ puis deux procédures d'optimisation sont ajoutées : une pour les circuits linéaires (réponse en fréquence) et une pour les circuits non-linéaires (réponse transitoire). En effet, la méthodologie d'origine s'avère moins adaptée aux technologies récentes, pour lesquelles les effets parasites ne sont pas pris en compte. Une optimisation est donc nécessaire pour vraiment conserver les figures de mérite lors de la migration technologique.

Les procédures d'optimisation sont basées sur des simulations SPICE et l'algorithme de Levenberg-Marquardt [PRES 2002]; elles permettent d'ajuster des paramètres du circuit pour conserver la réponse en fréquence ou transitoire du circuit de base.

L'optimisation pour la réponse en fréquence joue sur les longueurs et les largeurs de chaque transistor pour conserver les transconductances g_m et g_{ds} des transistors. Des simulations SPICE sont effectuées autour du point de fonctionnement pour déterminer les transconductances.

La figure 5.1 décrit cette approche d'optimisation. A partir du circuit original et de ses tailles de transistors (W_1, L_1) et de ses performances (g_{m1}, g_{ds1}), un premier redimensionnement est effectué en utilisant la méthodologie de Galup-Montoro. On obtient alors de nouvelles tailles (W_2, L_2) et de nouvelles performances (g_{m2}, g_{ds2}). Ensuite la procédure d'optimisation permet d'obtenir les tailles finales des transistors (W_3, L_3) du circuit redimensionné qui possède les mêmes performances (g_{m1}, g_{ds1}) en fréquence que le circuit original.

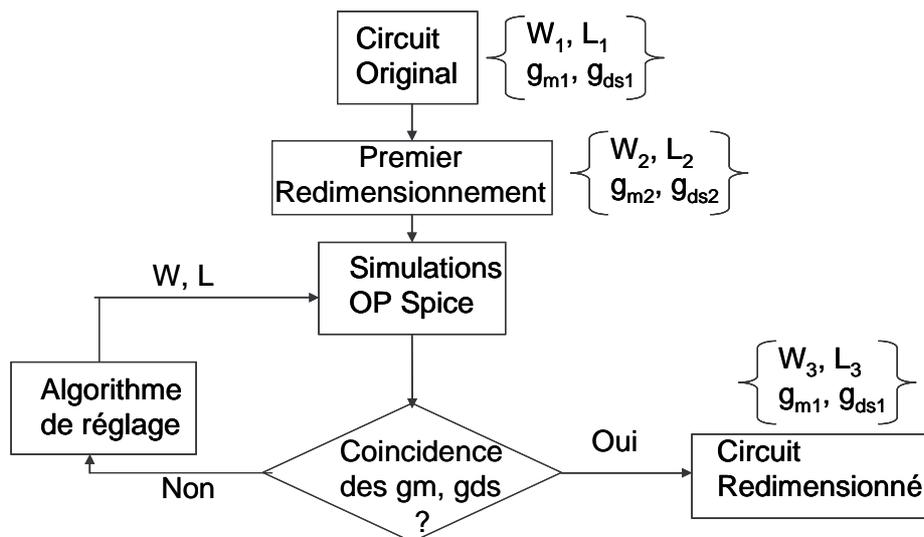


Figure 5.1. Organigramme de la procédure d'optimisation pour la réponse en fréquence

L'optimisation pour la réponse transitoire joue sur les longueurs et les largeurs de chaque transistor pour faire coïncider les réponses transitoires du circuit original et de celui d'arrivée.

Un OTA a été redimensionné et les deux procédures ont été effectuées pour démontrer leur validité.

Ainsi cette méthode améliore la précédente car elle permet véritablement de conserver les figures de mérite et elle diminue la consommation. Cependant un inconvénient subsiste concernant la surface. En effet si pour [GALU 2002] la surface de l'OTA redimensionné

doublait, elle quadruple avec cette méthode. La diminution de la surface de la puce étant à notre avis essentielle, nous proposerons dans la prochaine partie une autre approche.

II.5 Réalisations industrielles

II.5.1 Neocell et Neocircuit

Les logiciels Neocell et Neocircuit sont apparus en 1998 sous l'égide de la société Neo Linear achetée par Cadence. Ces outils sont intégrés à la chaîne analogique Virtuoso. Virtuoso Neocircuit est dédié au dimensionnement électrique et à l'optimisation, tandis que Virtuoso Neocell se consacre au placement / routage. Tous deux proposent des bibliothèques de cellules.

Le flot de conception se compose en quatre étapes :

- Le concepteur choisit sa topologie et la saisit via l'interface graphique de Virtuoso. Le schéma non dimensionné peut être partiellement dimensionné.
- A l'aide de l'interface graphique, le concepteur précise les contraintes simples comme les rapports de dimensionnement.
- Les figures de mérite (produit gain-bande, marge de phase, etc.) sont précisées avec leur domaines de variations possibles. Les objectifs sont aussi définis.
- Virtuoso se charge d'optimiser le circuit par algorithme de recuit simulé en faisant appel à un simulateur externe. Le domaine de conception est ainsi exploré.

Même si le concepteur a la possibilité de saisir un certain nombre de contraintes électriques, il ne lui est pas possible d'intervenir sur la méthode d'optimisation. L'appel à un simulateur électrique extérieur, de même que la nature des algorithmes d'optimisation, augmente le temps de conception.

II.5.2 Barcelona Design

L'entreprise Barcelona Design proposait une chaîne de conception essentiellement orientée vers les aspects systèmes et circuits. Le flot de conception se compose en trois étapes :

- Le schéma non dimensionné est saisi à l'aide d'un éditeur
- Le comportement du circuit est saisi par le concepteur sous forme d'équations analytiques.
- Un logiciel se charge de l'optimisation électrique par programmation géométrique, les contraintes ayant été préalablement imposées.

Barcelona Design proposait un ensemble de générateurs faciles à utiliser, mais la description analytique fine des éléments du circuit était impossible à réaliser faute de langage disponible et les générateurs étaient liés à un procédé technologique. Barcelona Design a procédé à la cessation de ses activités en mars 2005 ce qui met l'accent sur la difficulté de proposer des méthodologies de redimensionnement générales.

Ces outils industriels proposent le dimensionnement d'un circuit ad initio et non le redimensionnement à partir d'un circuit existant face à une migration technologique. Dans notre premier chapitre, le laboratoire LIP6 de Paris a été présenté, et plusieurs recherches sont menées pour le dimensionnement automatique d'un circuit : création du langage CAIRO+. Quant à notre approche, nous nous situons dans le redimensionnement et nous utilisons les valeurs initiales des transistors pour déterminer les nouvelles tailles des transistors.

III DESCRIPTION DE LA METHODOLOGIE PROPOSEE

III.1 Plan de la méthodologie

La méthodologie de redimensionnement que nous proposons peut être ramenée à trois étapes :

1^{ère} étape : Définition des figures de mérite à conserver

2^{ème} étape : Calcul des facteurs de redimensionnement technologiques

3^{ème} étape : Définition de la stratégie et calcul des facteurs de redimensionnement électriques et géométriques

La première étape pose les objectifs du redimensionnement. La deuxième est une simple application numérique qui dépend de la technologie ciblée. La troisième étape rassemble l'ensemble des 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.

Cette dernière étape dépend du modèle choisi de transistor MOS. Ici, le niveau 1 du transistor MOS a été utilisé et donne des estimations tout à fait pertinentes. Bien sûr, il est possible de choisir un modèle plus complexe comme celui du BSIM3V3 pour déterminer les facteurs de redimensionnement et ainsi calculer les nouvelles tailles des transistors.

III.2 Définition des facteurs de redimensionnement [LEVI 2006a]

Les facteurs de redimensionnement sont définis par le tableau 5.2, suivant une migration d'une technologie 1 à une technologie 2. Les trois premiers facteurs sont calculés à partir des paramètres technologiques. Le but étant de déterminer les facteurs géométriques K_L et K_W , qui sont eux-mêmes liés aux facteurs électriques.

TABLEAU 5.3. DEFINITION DES FACTEURS DE REDIMENSIONNEMENT

Paramètres	Facteurs de redimensionnement
Tension d'alimentation : V_{DD}	$K_V = V_{DD2} / V_{DD1}$
Capacité d'oxyde : C_{OX}	$K_{OX} = C_{OX2} / C_{OX1}$
Mobilité : μ_0	$K_\mu = \mu_{02} / \mu_{01}$
Tension effective de grille : $V_{EG} = V_{GS} - V_T$	$K_{EG} = V_{EG2} / V_{EG1}$
Courant de polarisation : I_{pol}	$K_I = I_{pol2} / I_{pol1}$
Longueur du transistor : L	$K_L = L_2 / L_1$
Largeur du transistor : W	$K_W = W_2 / W_1$

Nous remarquons que les valeurs des facteurs dépendent du type de transistors utilisés (NMOS ou PMOS).

Trois différents types de facteurs sont définis, les facteurs de redimensionnement technologiques, électriques et géométriques.

Pour cette méthodologie, nous avons défini le facteur de redimensionnement K_{EG} qui correspond à la tension effective de grille ($V_{EG} = V_{GS} - V_T$). Ce facteur a été choisi car il permet de conserver la différence entre la tension de seuil et la tension grille-source. En effet, généralement lors d'une migration technologique, la longueur minimale du transistor diminue de même que la tension de seuil V_T et la tension d'alimentation. En conservant un V_{GS} constant, le risque est le changement de régime du transistor. Maintenir une différence $V_{GS} - V_T$ constante est ainsi plus sensé pour le bon fonctionnement du circuit et la conservation de ses performances.

III.3 Analyse de la 3ème étape de la méthodologie

La 3^{ème} étape de la méthodologie est celle qui requiert le plus d'expertise. En effet, c'est à ce moment que nous décidons de la stratégie de redimensionnement et du choix des équations qui seront utilisées pour déterminer les facteurs géométriques.

Nous présentons ci-dessous les principales équations, provenant du niveau 1 du modèle de transistor MOS, qui seront utilisées dans les prochains paragraphes :

-le courant de drain en régime de saturation

$$I_D = \frac{\mu_0 C_{ox} W}{2L} V_{EG}^2 \quad (5.5)$$

-la transconductance g_m

$$g_m = \left[\frac{\partial i_D}{\partial v_{GS}} \right]_{V_{GS}=const}^{I_D=const} = \frac{\mu_0 C_{ox} W}{L} V_{EG} = \sqrt{\frac{2\mu_0 C_{ox} W I_D}{L}} \quad (5.6)$$

-la résistance drain-source moyenne lors d'une commutation $[0, V_{DD}]$

$$R_N = \frac{V_{DD}}{\frac{\mu_{0N} C_{OXN} W}{2} \frac{V_{DD} - V_{TN}}{L}} \quad (5.7)$$

$$R_P = \frac{V_{DD}}{\frac{\mu_{0P} C_{OXP} W}{2} \frac{V_{DD} + V_{TP}}{L}} \quad (5.8)$$

Ces équations donnent une estimation de la résistance entre le drain et la source du MOSFET pendant une commutation [DEMA 1996].

-la capacité de grille [BAKE 1998]

$$C_G \approx W L C_{OX} \quad (5.9)$$

III.4 Discussion sur le champ d'application de la méthodologie

Il est possible d'appréhender dès maintenant les limites de cette méthodologie. Pour les technologies les plus récentes qui possèdent une faible longueur minimale, les effets parasites ne sont pas inclus dans les équations et donc l'erreur d'estimation des performances n'est parfois pas négligeable. Pour répondre à ce problème, deux solutions émergent.

Dans un premier temps, la méthodologie peut être affinée par une procédure d'optimisation, comme cela a été présenté dans certains travaux de l'état de l'art. Dans les exemples qui vont suivre, la fonction d'optimisation de l'environnement Cadence [CADE] sera utilisée. Elle donnera de meilleurs résultats pour les technologies récentes et réduira ainsi la variation de performance due aux effets parasites. Cet outil itératif permet une optimisation des

performances en ajustant des variables choisies par le concepteur, en l'occurrence les tailles des transistors. Une légère variations des longueurs et largeurs des transistors est obtenue, comme cela est explicité plus loin.

Dans un deuxième temps, le modèle de transistor MOS utilisé peut être amélioré. En utilisant les équations du modèle BSIM3V3, la méthodologie donnera de meilleurs résultats avec pour seule différence une complexité accrue des calculs pour la détermination des facteurs de redimensionnement. Notons toutefois que l'usage du modèle ACM (proche du modèle EKV) par les équipes précitées, ne dispense pas en définitive d'une procédure d'optimisation.

Cette méthodologie est appliquée sur :

- des circuits linéaires : OTAs où tous les transistors sont en régime de saturation
- des circuits non-linéaires : oscillateurs en anneau.

La migration technologique se fera d'une technologie AMS BiCMOS 0,8 μm /5 V vers une technologie AMS BiCMOS 0,35 μm /3,3 V, ST CMOS 0,25 μm /2,5 V et enfin ST CMOS 0,12 μm /1,2 V.

IV APPLICATION SUR DES CIRCUITS LINEAIRES

IV.1 L'OTA

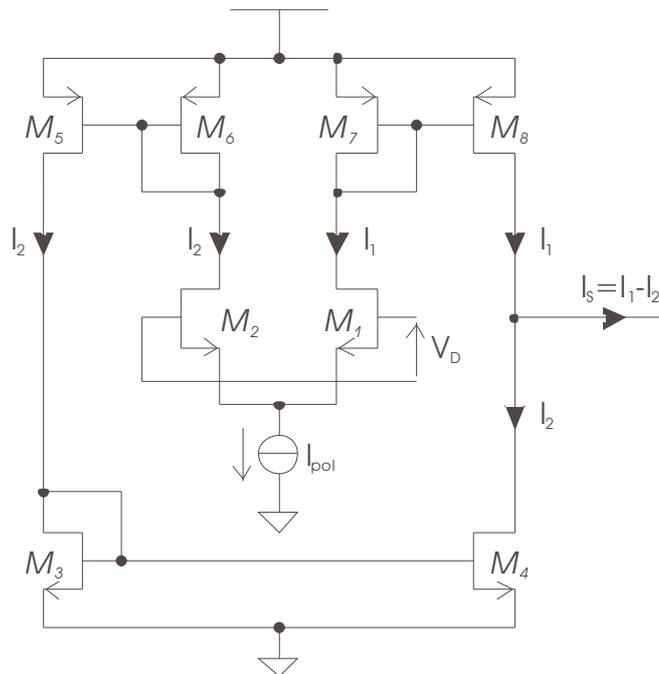


Figure 5.2. Topologie de l'OTA

La figure 5.3 représente une topologie classique d'OTA. Les tailles initiales des transistors sont données par le tableau 5.5.

IV.2 Première application de la méthodologie : conservation du gain en transconductance

IV.2.1 Redimensionnement

1^{ère} étape : la figure de mérite à conserver est le gain en transconductance de l'OTA.

2^{ème} étape : K_V , K_{OX} , K_μ sont déduits à partir des informations technologiques.

TABEAU 5.4. VALEURS DES FACTEURS DE REDIMENSIONNEMENT TECHNOLOGIQUES

Technologie :	Facteurs de redimensionnement technologiques	PMOS	NMOS
CMOS 0.8 μm 5 V	K_μ	0,833	0,937
	K_{OX}	1,063	1,067
	K_V	0,660	0,660

3^{ème} étape : le choix le plus simple pour conserver le gain en transconductance est de maintenir le courant de drain, alors à partir de (5.5) et du tableau 5.3 :

$$\frac{\mu_{01}C_{OX1}W_1}{2L_1}V_{EG1}^2 = \frac{\mu_{02}C_{OX2}W_2}{2L_2}V_{EG2}^2 \quad (5.10)$$

$$W_2 = \frac{\mu_{01}C_{OX1}W_1 2L_2}{\mu_{02}C_{OX2} 2L_1} \frac{V_{EG1}^2}{V_{EG2}^2} \quad (5.11)$$

$$W_2 = \frac{K_L}{K_\mu K_{ox} K_{EG}^2} W_1 \quad (5.12)$$

$$K_W = \frac{K_L}{K_\mu K_{OX} K_{EG}^2} \quad (5.13)$$

A ce niveau, K_L et K_{EG} sont inconnus. D'autres choix stratégiques doivent être faits afin d'avoir deux équations supplémentaires.

Nous décidons de conserver la tension effective de grille V_{EG} ($V_{EG1}=V_{EG2}$), vérifiant par la suite si cette décision est compatible avec une diminution de la tension d'alimentation V_{DD} . De plus, nous réduisons la longueur des transistors L par un facteur correspondant à la décroissance de la longueur minimale de la technologie. Ainsi K_L est égal à L_{MIN2} / L_{MIN1} .

Nous pouvons à présent calculer K_W et nous connaissons ainsi toutes les valeurs des facteurs de redimensionnement à partir desquelles on déduira les nouvelles tailles des transistors (Tableau 5.5).

TABLEAU 5.5. RECAPITULATIF DES DIMENSIONS DES TRANSISTORS

Transistors W/L	0,8 μm 5 V	0,35 μm 5 V	0,25 μm 2,5 V	0,12 μm 1,2 V
M ₁ (μm)	20/10	8,75/4,4	3,6/3,15	0,8/1,5
M ₂ (μm)	20/10	8,75/4,4	3,6/3,15	0,8/1,5
M ₃ (μm)	40/10	17,5/4,4	7,25/3,15	1,55/1,5
M ₄ (μm)	40/10	17,5/4,4	7,25/3,15	1,55/1,5
M ₅ (μm)	40/10	19,75/4,4	6,45/3,15	2,45/1,5
M ₆ (μm)	40/10	19,75/4,4	6,45/3,15	2,45/1,5
M ₇ (μm)	40/10	19,75/4,4	6,45/3,15	2,45/1,5
M ₈ (μm)	40/10	19,75/4,4	6,43/3,15	2,45/1,5

IV.2.2 Résultats

Pour évaluer les performances DC de la méthodologie, nous comparons le gain en transconductance DC : G_0 (correspondant à la pente dans la région linéaire de la caractéristique DC, figure 5.3).

Pour obtenir le gain en tension basse fréquence, le produit gain-bande passante et la marge de phase, une capacité de charge a été ajoutée en sortie de l'OTA. Sa valeur est de 50 pF.

Toutes les simulations ont été menées avec Spectre dans l'environnement Cadence et avec le modèle du transistor BSIM3V3, dans lequel la modulation de la longueur du canal et les effets du canal court sont inclus.

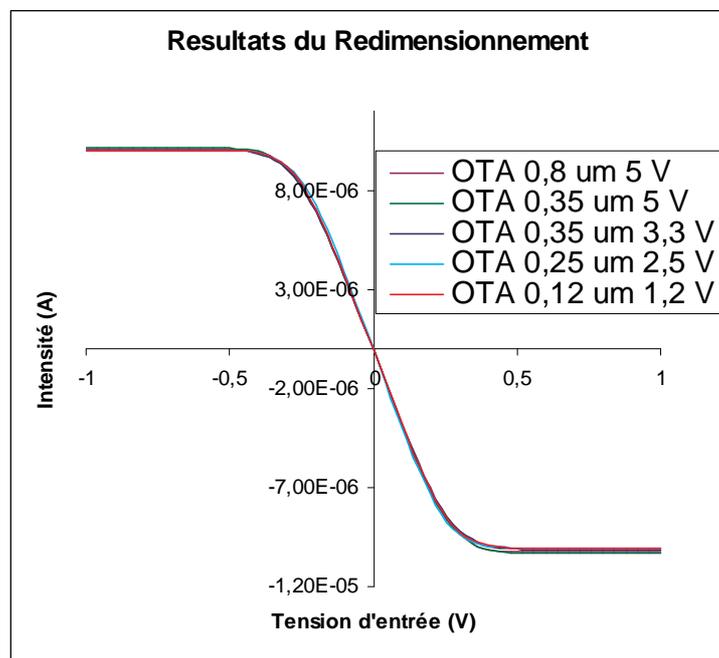


Figure 5.3. Comportement DC des différents OTAs

TABLEAU 5.6. RESULTATS DES SIMULATIONS

Technologie (μm)	0,8	0,35	0,35	0,25	0,12
Tension d'alimentation (V)	5	5	3,3	2,5	1,2
G_0 ($\mu\text{A}\cdot\text{V}^{-1}$)	38	39	38	40	38
Gain en tension DC (dB)	54	54	54	54	49
Produit Gain-Bande passante (MHz)	6.5	6.5	6.5	6.5	6.1
Marge de Phase ($^\circ$)	88	89	89	89	89
Consommation en puissance (μW)	302	302	198	150	72
Surface (μm^2)	2800	580	580	150	22

Le tableau 5.6 décrit les résultats des simulations DC et AC menées sur l'OTA. L'objectif étant la conservation du gain en transconductance G_0 , les résultats peuvent être considérés comme satisfaisants. De plus la surface décroît largement (80 % entre la technologie 0.8 μm et la technologie 0.35 μm , 75 % entre la technologie 0.35 μm et la technologie 0.25 μm et 85 % entre la technologie 0.25 μm et la technologie 0.12 μm). La consommation en puissance décroît d'un facteur K_V dépendant de la baisse de la tension d'alimentation.

Toutefois nous observons une légère décroissance du gain en tension et du produit gain-bande passante pour la technologie 0.12 μm . Pour y remédier, nous utilisons la fonction d'optimisation d'Analog Artist de l'environnement Cadence qui nous permettra de calculer les nouvelles tailles des transistors (Tableau 5.7). Les variables d'ajustement de cette optimisation sont les W et L des transistors afin de conserver le gain en tension DC. Les performances de cet OTA sont données dans le tableau 5.8.

TABLEAU 5.7. COMPARAISON ENTRE LES TAILLES DES TRANSISTORS

Transistors W/L	0.12 μm 1.2 V	0.12 μm avec l'optimisation 1.2 V
M ₁ (μm)	0,8/1,5	1,05/2,65
M ₂ (μm)	0,8/1,5	1,05/2,65
M ₃ (μm)	1,55/1,5	2,65/1,1
M ₄ (μm)	1,55/1,5	2,65/1,1
M ₅ (μm)	2,45/1,5	2,75/1,35
M ₆ (μm)	2,45/1,5	2,75/1,35
M ₇ (μm)	2,45/1,5	2,75/1,35
M ₈ (μm)	2,45/1,5	2,75/1,35

TABLEAU 5.8. COMPARAISON DES PERFORMANCES AVEC OU SANS LA BOUCLE D'OPTIMISATION

Technologie (μm)	0.12	0.12 avec l'optimisation
Tension d'alimentation (V)	1.2	1.2
G_0 ($\mu\text{A}\cdot\text{V}^{-1}$)	38	38
Gain en tension DC (dB)	49	54
Produit Gain-Bande passante (MHz)	6,1	6,5
Marge de Phase ($^\circ$)	89	89
Consommation en puissance (μW)	72	72
Surface (μm^2)	22	29

IV.2.3 Evaluation de la robustesse

L'analyse de la robustesse lors d'un redimensionnement est très importante. En effet, si le circuit redimensionné possède les mêmes performances que celui initial mais que sa robustesse n'est pas conservée, la moindre variation lors de la fabrication aura pour conséquence un changement de ces performances.

L'analyse de MonteCarlo [PELG 1989] consiste à définir aléatoirement la valeur des composants (ou plus exactement les paramètres des modèles correspondants) à l'intérieur de leur gamme de tolérance et ensuite à évaluer l'indice (ou les indices) de performance concerné. L'opération est réalisée un grand nombre de fois, et les caractéristiques numériques statistiques sont enfin calculées. Si les corrélations entre paramètres ne sont pas prises en compte, cette technique conduit à un gonflement de l'espace de dispersion. Cette simulation permet de connaître la robustesse du circuit par rapport à une variation d'un ou plusieurs de ses paramètres. Il est possible de faire varier de façon uniforme les paramètres (d'un wafer à l'autre) grâce à la simulation MonteCarlo « avec Process », mais aussi de les faire varier indépendamment pour chaque composant (sur le même wafer), il s'agit de la simulation MonteCarlo « avec Mismatch ».

Les figures 5.5, 5.6 et 5.7 décrivent les résultats des simulations MonteCarlo avec Mismatch et Process du gain de l'OTA en boucle fermée. Le circuit simulé est un amplificateur en tension basé sur l'OTA avec pour boucle un réseau de résistances. La valeur du gain en tension DC initial est de 61 dB. Nous ne donnons pas l'histogramme des simulations pour la technologie 0.25 μm car les modèles MonteCarlo des transistors ne sont pas disponibles pour cette technologie.

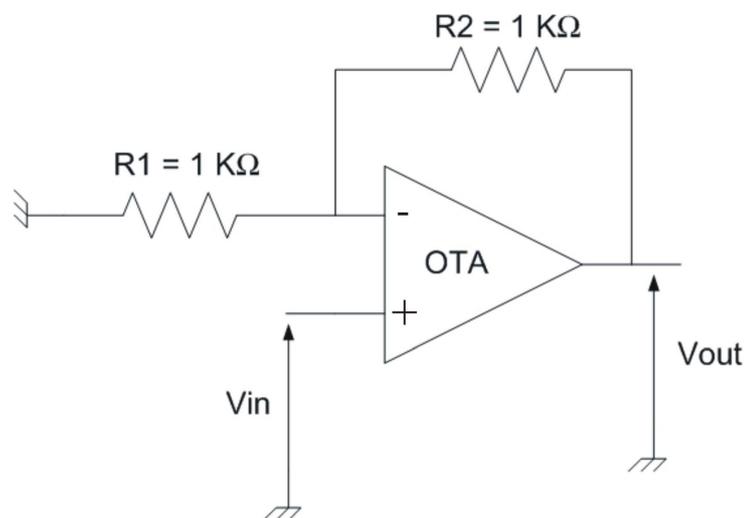


Figure 5.4. Schéma de la simulation

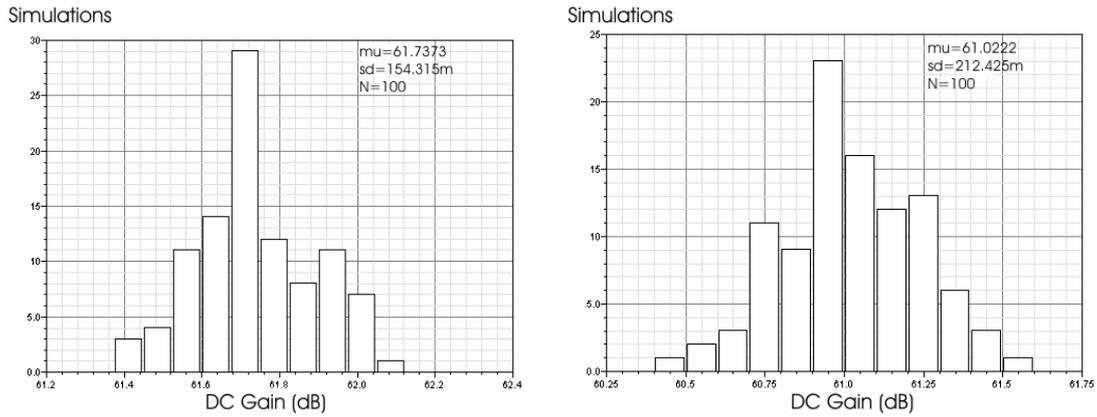


Figure 5.5. Simulations MonteCarlo du gain en tension DC de l'OTA en technologie 0,8 μm et 5 V (à gauche) et en technologie 0,35 μm et 5 V (à droite)

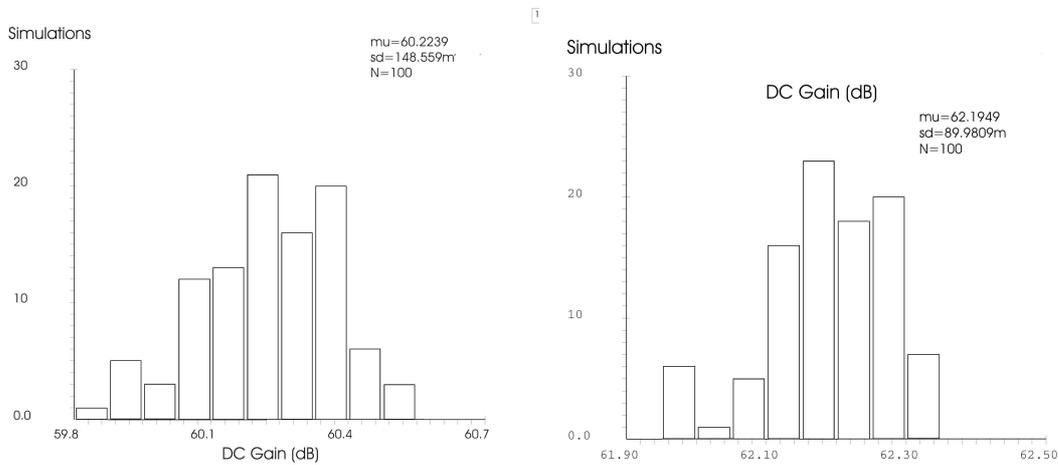


Figure 5.6. Simulations MonteCarlo du gain en tension DC de l'OTA en technologie 0,35 μm et 3,3 V (à gauche) et en technologie 0,12 μm et 1,2 V (à droite)

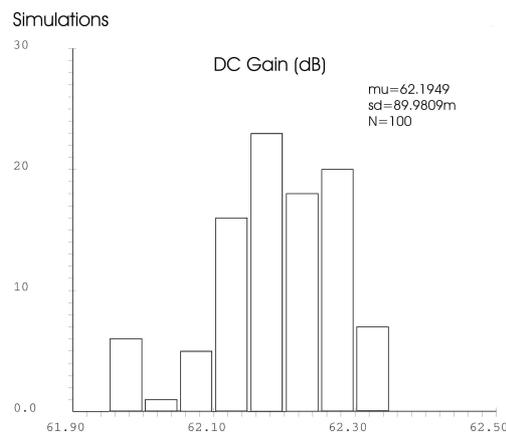


Figure 5.7. Simulations MonteCarlo du gain en tension DC de l'OTA optimisé en technologie 0,12 μm et 1,2 V

Nous avons vérifié que le réseau de résistances ne perturbait pas le résultat de la robustesse du circuit. En effet lors d'une variation du gain en transconductance, G_0 , le gain en tension DC varie de la même manière.

Pour les technologies $0.8 \mu\text{m}$ 5 V, $0.35 \mu\text{m}$ 3.3 V et $0.12 \mu\text{m}$ 1.2 V, toutes les valeurs du gain DC sont incluses entre 98 et 102 % de la valeur initiale. Nous pouvons donc affirmer que la robustesse du circuit est conservée avec cette méthodologie.

Pour conclure avec cet exemple d'application, les figures du comportement DC de l'OTA (5.3), et les figures des simulations MonteCarlo (5.5, 5.6, 5.7) et les tableaux 5.5 et 5.7 prouvent que cette méthodologie conserve les performances de l'OTA aussi bien en DC, AC ainsi que la robustesse. Surtout, elle atteint les objectifs fixés en termes de conservation du gain en transconductance et affiche une importante réduction de la surface. L'outil d'optimisation est utilisé pour l'amélioration des résultats AC pour certaines technologies.

IV.3 Deuxième application : conservation du gain en transconductance et réduction de la consommation en puissance

IV.3.1 Redimensionnement

Pour cet exemple, l'OTA utilisé est le même que précédemment (figure 5.2). La seule différence réside dans le choix de la stratégie (3^{ème} étape). En effet cette fois ci, le concepteur veut aussi réduire la puissance consommée de son circuit même si la dynamique en courant de l'OTA baisse un peu.

1^{ère} étape : la figure de mérite à conserver reste le gain en transconductance de l'OTA avec une contrainte de réduction de la consommation en puissance

2^{ème} étape : K_V , K_{OX} , K_μ sont déduits à partir des informations technologiques.

3^{ème} étape : Nous voulons conserver le gain en transconductance g_m de l'OTA. Pour diminuer la consommation, nous choisissons de réduire le courant de polarisation I_{pol} d'un facteur K_I . La valeur de K_I dépend des spécifications et des objectifs que nous voulons atteindre. Dans cette application, nous avons pris $2/3$ comme valeur de K_I .

Comme le décrit l'équation (5.6), pour réduire la consommation en puissance du circuit d'un facteur K , nous devons multiplier la longueur et la largeur des transistors par ce facteur mais aussi diviser le courant de polarisation par ce même facteur.

$$I_{D2} = K_I I_{D1} \tag{5.14}$$

$$K_I^2 = \frac{K_\mu K_{ox} K_{EG}^2 K_W}{K_L} \tag{5.15}$$

$$K_L = \frac{L_{MIN2}}{K_I L_{MIN1}} \tag{5.16}$$

$$K_W = \frac{K_L K_I}{K_\mu K_{ox} K_{EG}^2} \tag{5.17}$$

TABLEAU 5.9. TAILLES DES TRANSISTORS

Transistors W/L	0.8 μm 5 V	0.35 μm 3.3 V	0.25 μm 2.5 V	0.12 μm 1.2 V
M ₁ (μm)	10/2	9,85/1,3	5,6/0,85	2,5/0,55
M ₂ (μm)	10/2	9,85/1,3	5,6/0,85	2,5/0,55
M ₃ (μm)	10/2	9,85/1,3	5,6/0,85	2,5/0,55
M ₄ (μm)	10/2	9,85/1,3	5,6/0,85	2,5/0,55
M ₅ (μm)	10/2	11,1/1,3	4,95/0,85	3,9/0,55
M ₆ (μm)	10/2	11,1/1,3	4,95/0,85	3,9/0,55
M ₇ (μm)	10/2	11,1/1,3	4,95/0,85	3,9/0,55
M ₈ (μm)	10/2	11,1/1,3	4,95/0,85	3,9/0,55
I _{POL}	45 μA	30 μA	20 μA	13 μA

IV.3.2 Résultats

Les résultats de la méthodologie sur cet exemple sont donnés par la figure 5.8 et le tableau 5.10.

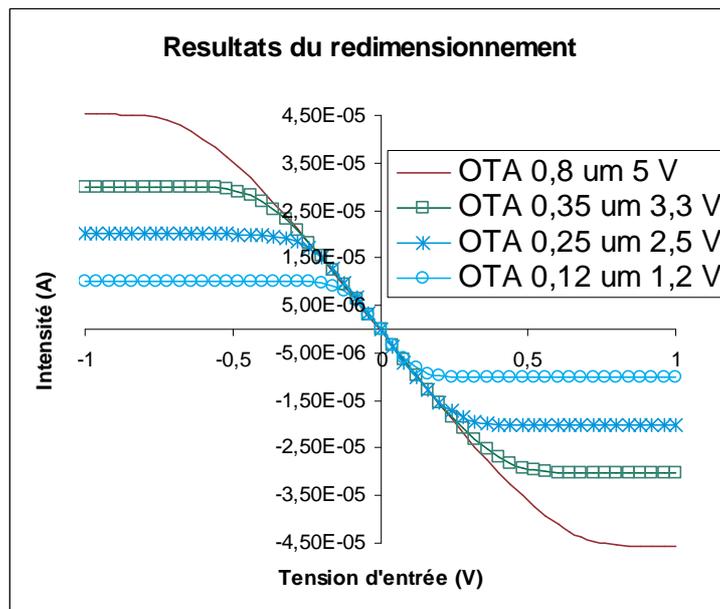


Figure 5.8. Comportement DC des différents OTAs

TABLEAU 5.10. TAILLES DES TRANSISTORS

Transistors W/L	0,8 μm 5 V	0,35 μm 3,3 V	0,25 μm 2,5 V	0,12 μm 1,2 V
M ₁ (μm)	10/2	9,85/1,3	5,6/0,85	2,5/0,55
M ₂ (μm)	10/2	9,85/1,3	5,6/0,85	2,5/0,55
M ₃ (μm)	10/2	9,85/1,3	5,6/0,85	2,5/0,55
M ₄ (μm)	10/2	9,85/1,3	5,6/0,85	2,5/0,55
M ₅ (μm)	10/2	11,1/1,3	4,95/0,85	3,9/0,55
M ₆ (μm)	10/2	11,1/1,3	4,95/0,85	3,9/0,55
M ₇ (μm)	10/2	11,1/1,3	4,95/0,85	3,9/0,55
M ₈ (μm)	10/2	11,1/1,3	4,95/0,85	3,9/0,55
I _{POL}	45 μA	30 μA	20 μA	13 μA

TABLEAU 5.11. RÉSULTATS DES SIMULATIONS

Technologie (μm)	0,8	0,35	0,25	0,12
Tension d'alimentation (V)	5	3,3	2,5	1,2
G₀ ($\mu\text{A.V}^{-1}$)	78	80	81	80
Consommation en puissance (μW)	459	200	100	32
Surface (μm^2)	160	109	36	14

Les résultats des simulations apparaissant dans la figure 5.8 et le tableau 5.10 montrent que cette méthodologie conserve la transconductance (G_0), tout en réduisant le coût du circuit avec la diminution de surface (32 % entre la technologie 0.8 μm et la technologie 0.35 μm , 67 % entre la technologie 0.35 μm et la technologie 0.25 μm et 62 % entre la technologie 0.25 μm et celle en 0.12 μm) et en entraînant une baisse de la consommation en puissance (57 % entre la technologie 0.8 μm et celle en 0.35 μm , 50 % entre la technologie 0.35 μm et la technologie 0.25 μm et 68 % entre la technologie 0.25 μm et celle en 0.12 μm).

Nous n'avons pas estimé nécessaire d'illustrer les résultats des simulations MonteCarlo de cet exemple car les conclusions de ces simulations sont les mêmes que pour l'exemple précédent.

IV.4 Conclusion concernant les exemples linéaires

Premièrement nous avons appliqué la méthodologie sur un OTA avec pour objectif de conserver le gain en transconductance. Puis nous avons changé la stratégie de redimensionnement afin d'y ajouter une réduction de la consommation en puissance.

Les résultats étant très satisfaisants, nous remarquons que ces deux exemples démontrent la flexibilité de la méthodologie. Dans les deux exemples, le gain en transconductance est conservé et la surface est réduite quelles que soient les technologies CMOS utilisées.

V APPLICATION SUR DES CIRCUITS NON-LINEAIRES

La méthodologie de redimensionnement est appliquée sur un circuit non-linéaire, un oscillateur en anneau conçu avec des inverseurs. Un inverseur CMOS est un bloc basique pour la conception de circuits.

V.1 L'oscillateur en anneau

Un oscillateur en anneau est construit avec un nombre impair d'inverseurs qui forment une boucle fermée. Tous les inverseurs sont identiques. La principale caractéristique de l'oscillateur en anneau est sa fréquence d'oscillation.

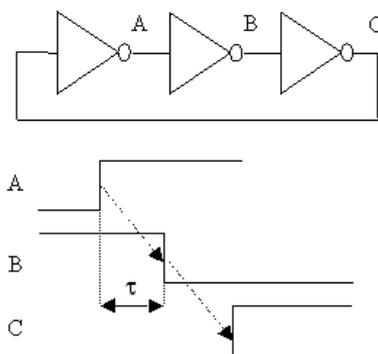


Figure 5.9. Un oscillateur en anneau

1^{ère} étape : la figure de mérite à conserver est la fréquence d'oscillation. Elle est définie par [SEDR 1991].

$$f_{osc} = \frac{1}{2n(t_{PHL} + t_{PLH})} \quad (5.18)$$

Avec $t_{PHL} + t_{PLH} = (R_n + R_p)C_{tot}$

2^{ème} étape : K_V , K_{OX} , K_μ sont déduits à partir des informations technologiques.

3^{ème} étape : Pour maintenir la fréquence d'oscillation, nous avons choisi de conserver les résistances et les capacités associées à l'inverseur lors d'une migration technologique.

V.2 L'inverseur CMOS

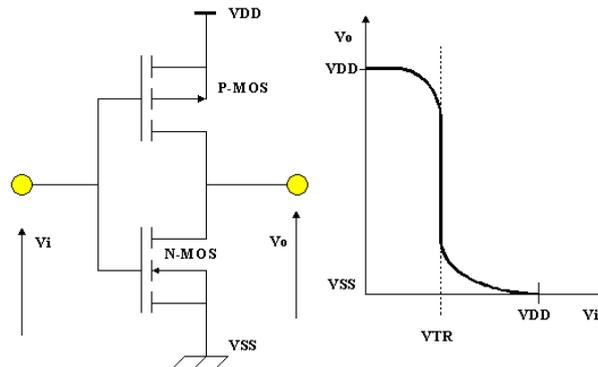


Figure 5.10. Caractéristiques de transfert d'un inverseur CMOS

Pour notre oscillateur en anneau, nous utilisons des inverseurs CMOS à taille minimale. Ce qui signifie que les largeurs et les longueurs du PMOS et du NMOS de l'inverseur sont identiques ($W_P=W_N$ et $L_P=L_N$).

Capacités de l'inverseur CMOS :

La capacité d'entrée de l'inverseur est donnée par la relation :

$$C_{IN} = C_{GN} + C_{GP} \approx (W_N L_N + W_P L_P) C_{OX} \quad (5.19)$$

La capacité de grille du NMOS est appelée C_{GN} et celle du PMOS C_{GP} .

La capacité de sortie de l'inverseur est négligeable par rapport à la capacité d'entrée. Si bien que la capacité équivalente d'un nœud de l'oscillateur vaut $C_{IN} + C_{OUT} \approx C_{IN}$. Par conséquent, la capacité totale de l'oscillateur en anneau avec n inverseurs est égale à nC_{IN} .

$$C_{TOT} = n C_{IN} = n (W_N L_N C_{OXN} + W_P L_P C_{OXP}) \quad (5.20)$$

Résistances de l'inverseur CMOS :

La résistance drain-source d'un transistor MOS est donnée par (5.7) et (5.8).

V.3 Redimensionnement

Notre stratégie consiste à conserver les résistances et les capacités équivalentes de chaque transistor, dans le but de maintenir la fréquence d'oscillation.

Conservation des capacités :

$$\text{NMOS: } W_{N2} L_{N2} C_{OXN2} = W_{N1} L_{N1} C_{OXN1} \quad (5.21)$$

$$\text{PMOS: } W_{P2} L_{P2} C_{OXP2} = W_{P1} L_{P1} C_{OXP1} \quad (5.22)$$

En utilisant (5.21) et (5.22), nous obtenons :

$$W_2 L_2 = \frac{W_1 L_1}{K_{ox}} \quad (5.23)$$

$$\text{Soit } K_W K_L = \frac{1}{K_{ox}} \quad (5.24)$$

Conservation des résistances:

$R_{N2} = R_{N1}$ et $R_{P2} = R_{P1}$ alors

$$\frac{L_2}{W_2} = \frac{L_1}{W_1} \frac{K_{ox} K_\mu}{K_V} \frac{(V_{DD2} - V_{T2})^2}{(V_{DD1} - V_{T1})^2} \quad (5.25)$$

Avec (5.24) et (5.25), on obtient cette relation :

$$\frac{K_L}{K_W} = \frac{K_{ox} K_\mu}{K_V} \frac{(V_{DD2} - V_{T2})^2}{(V_{DD1} - V_{T1})^2} \quad (5.26)$$

Ainsi tous les facteurs de redimensionnement sont déterminés, et notamment les principaux, à savoir la largeur du transistor K_W et sa longueur K_L .

$$K_W = \frac{1}{K_{ox}} \sqrt{\frac{K_V}{K_\mu}} \frac{(V_{DD1} - V_{T1})}{(V_{DD2} - V_{T2})} \quad (5.27)$$

$$K_L = \sqrt{\frac{K_\mu}{K_V}} \frac{(V_{DD2} - V_{T2})}{(V_{DD1} - V_{T1})} \quad (5.28)$$

V.4 Résultats des simulations

Plusieurs tests ont été effectués sur des oscillateurs en anneau possédant des caractéristiques différentes et un nombre d'inverseurs différent lors d'une migration technologique. Deux essais sur des oscillateurs ayant la même topologie mais des tailles de transistors différentes ont été effectués ainsi qu'un test sur des oscillateurs ne comptant pas le même nombre d'inverseurs. Comme pour l'OTA, la technologie de départ est la AMS BiCMOS 0.8 μm et celles d'arrivée sont la AMS BiCMOS 0.35 μm , la ST CMOS 0.25 μm et la ST CMOS 0.12 μm .

Concernant la technologie ST CMOS 0.12 μm , nous observons que les équations (5.7), (5.8) et (5.20) ne sont pas valides pour cette technologie. En effet le canal du transistor dans cette technologie avec 1.2 V pour tension d'alimentation n'est pas complètement fermé et nous

avons alors des effets de canal court [LEE 2003]. Par conséquent, le transistor est plus souvent dans un régime linéaire que dans un régime de saturation et les nouvelles équations sont alors égales à :

$$C_{TOT} = nC_{IN} = n\frac{2}{3}(W_N L_N C_{OXN} + W_P L_P C_{OXP}) \quad (5.29)$$

$$R_N = \frac{V_{DD}}{\frac{2\mu_{0N} C_{OXN}}{3} \frac{W}{L} (V_{DD} - V_{TN})^2} \quad (5.30)$$

$$R_P = \frac{V_{DD}}{\frac{2\mu_{0P} C_{OXP}}{3} \frac{W}{L} (V_{DD} + V_{TP})^2} \quad (5.31)$$

Toutes les applications effectuées valident la méthodologie. Nous ne présentons ici qu'une des applications : un oscillateur en anneau possédant 5 inverseurs.

Détermination de la fréquence d'oscillation :

En utilisant (5.7), (5.8), (5.18) et (5.20) nous déterminons la fréquence théorique d'oscillation.

Les tailles initiales du NMOS et du PMOS dans la technologie 0.8 μm avec 5 V de tension d'alimentation sont :

$$L_{IN} = L_{IP} = 2 \mu\text{m} \text{ et } W_{IN} = W_{IP} = 50 \mu\text{m}.$$

Avec les facteurs de redimensionnement (5.27) et (5.28), nous calculons les nouvelles tailles des transistors pour les différentes technologies (0.35, 0.25 et 0.12 μm).

TABLEAU 5.12. NOUVELLES TAILLES DES TRANSISTORS

Taille	0,35 μm 5 V	0,35 μm 3,3 V	0,25 μm 2,5 V	0,12 μm 1,2 V	0,12 μm avec l'optimisation 1,2 V
L_N (μm)	1,95	1,25	1,00	1,00	1,00
L_P (μm)	1,75	1,40	1,05	0,73	0,70
W_N (μm)	47,85	75,45	34,10	21,95	24,60
W_P (μm)	53,70	64,30	32,15	29,9	21,00

Les fréquences théoriques et expérimentales d'oscillation sont données par le tableau 5.13. Les valeurs théoriques sont calculées à partir des facteurs de redimensionnement tandis que les valeurs expérimentales sont issues des simulations transitoires.

La figure 5.11 décrit les oscillations de quatre oscillateurs en anneau, le premier en technologie CMOS 0.8 μm , le second en technologie CMOS 0.35 μm , le troisième en technologie CMOS 0.25 μm , et le dernier en technologie CMOS 0.12 μm .

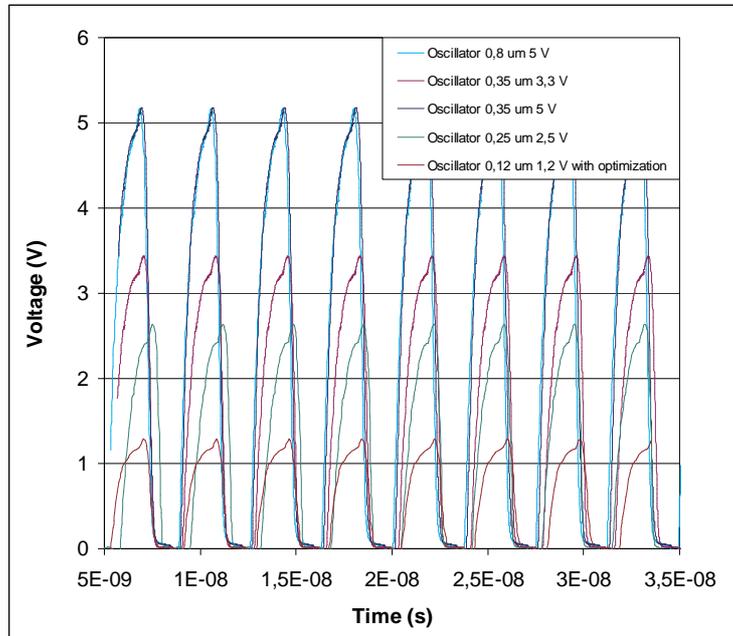


Figure 5.11. Comportement transitoire des oscillateurs en anneau

TABLEAU 5.13. FREQUENCE D'OSCILLATION

L_{MIN} (μm)	0,8	0,35	0,35	0,25	0,12	0,12 avec l'optimisation
V_{DD} (V)	5	5	3,3	2,5	1,2	1,2
f_{osc} théorique (MHz)	267	268	267	268	267	242
f_{osc} expérimentale (MHz)	267	268	266	270	293	268
Surface (μm^2)	1000	936	930	339	229	196

Le tableau 5.13 nous indique que les fréquences d'oscillation des trois oscillateurs en anneau restent identiques.

V.5 Evaluation de la robustesse

Les prochaines figures 5.12 et 5.13 donnent le résultat des oscillateurs aux simulations MonteCarlo avec Process et Mismatch. Comme pour l'OTA, les modèles MonteCarlo pour la technologie 0.25 μm ne sont pas disponibles.

Concernant ces simulations, en examinant le rapport longueur des transistors par rapport à la longueur minimale de la technologie, nous pouvons prévoir le comportement du circuit. En effet comme ce rapport est d'environ 3 pour la technologie $0.8 \mu\text{m}$, de 5 pour la technologie $0.35 \mu\text{m}$ et de 8 pour la technologie $0.12 \mu\text{m}$ nous nous attendons à ce que le circuit devienne plus robuste par rapport à celui d'origine.

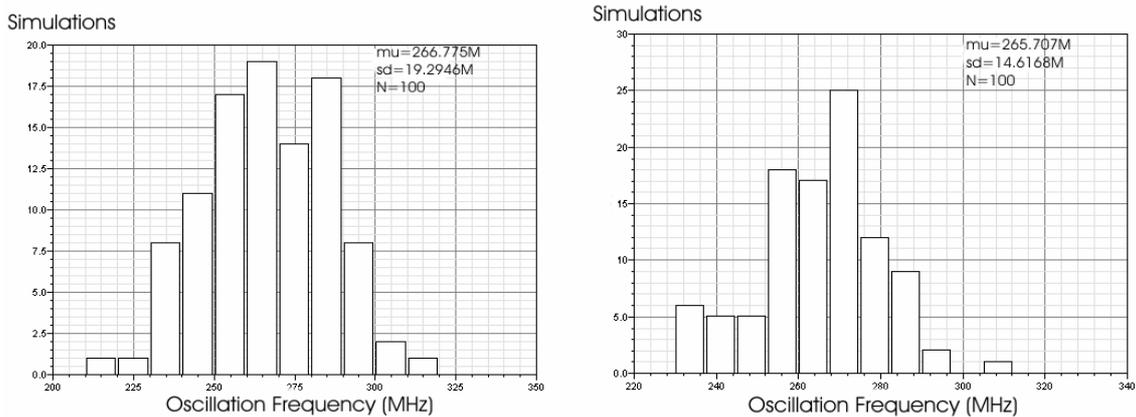


Figure 5.12. Simulation MonteCarlo de la fréquence d'oscillation de l'oscillateur pour la technologie $0,8 \mu\text{m}$ avec 5 V (à gauche) et en technologie $0,35 \mu\text{m}$ avec $3,3 \text{ V}$ (à droite)

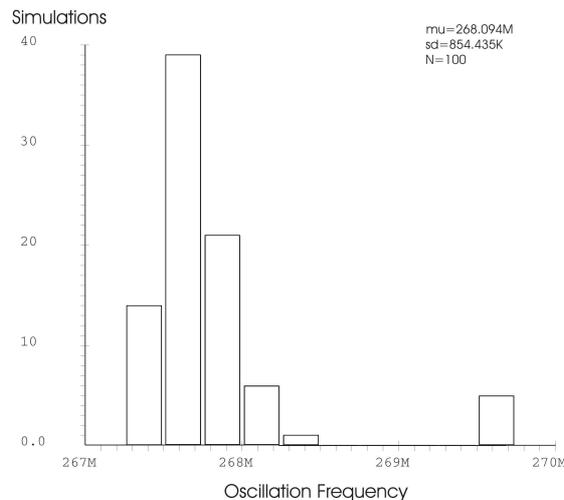


Figure 5.13. Simulation MonteCarlo de la fréquence d'oscillation de l'oscillateur en anneau pour la technologie $0.12 \mu\text{m}$ avec 1.2 V et en utilisant la fonction d'optimisation

La valeur moyenne de la fréquence d'oscillation est conservée de même que l'écart type pendant la migration technologique.

Pour les technologies $0.8 \mu\text{m}$ 5 V , $0.35 \mu\text{m}$ 3.3 V et $0.12 \mu\text{m}$ 1.2 V , toutes les valeurs des fréquences d'oscillation sont incluses entre 96 et 104 % de la valeur initiale (267 MHz). Nous pouvons donc affirmer que la robustesse du circuit est conservée avec cette méthodologie en remarquant qu'elle est même améliorée par rapport au circuit initial comme nous l'avons dit dans le paragraphe précédent.

De plus le coût du circuit est réduit lors de ces migrations technologiques car la surface diminue (7 % entre la technologie 0.8 μm et la technologie 0.35 μm , 64 % entre la technologie 0.35 μm et la technologie 0.25 μm et 42 % entre la technologie 0.25 μm et la technologie 0.12 μm).

Néanmoins nous remarquons que pour la technologie 0.12 μm , la méthodologie montre ses limites : pour conserver les performances du circuit, l'outil d'optimisation est nécessaire et les formules issues du modèle de niveau 1 adaptées.

VI CONCLUSION ET PERSPECTIVES

Dans ce chapitre, nous avons défini une méthodologie de redimensionnement pour la réutilisation de circuits analogiques. L'objectif de ce travail étant de conserver les performances d'un circuit lors d'une migration technologique, nous pouvons affirmer que cette méthodologie est valide. Nous avons appliqué ce travail sur des circuits linéaires (OTAs) ce qui a permis une comparaison avec d'autres travaux sur le redimensionnement, mais nous avons aussi proposé des applications non-linéaires (oscillateurs en anneau) en guise d'extension, ce que prouve la généralisation de la méthode. Les différentes topologies d'origine sont en technologie CMOS 0.8 μm , avec une tension d'alimentation de 5 V. Elles ont ensuite été redimensionnées en technologie CMOS 0.35 μm , 0.25 μm et 0.12 μm avec respectivement 3.3, 2.5 et 1.2 V pour tension d'alimentation.

Cette méthodologie peut se décomposer en trois étapes et identifie clairement quels paramètres sont déduits des informations technologiques et ceux qui sont déterminés selon les différentes stratégies.

Tous les résultats proviennent de simulations effectuées sous l'environnement Cadence et les conclusions de celles-ci montrent que la méthodologie conserve les performances DC, AC et transitoires de même que la robustesse.

CONCLUSION GENERALE ET PERSPECTIVES

Les travaux présentés dans ce mémoire apportent une contribution à l'automatisation du flot de conception analogique et mixte.

L'état de l'art dressé au chapitre 1 a permis de mettre en évidence les lacunes du flot de conception analogique par rapport au flot de conception numérique. De réels besoins en outils et méthodologies de conception sont apparus pour augmenter le degré d'automatisation de la conception analogique. L'idée maîtresse de ces travaux est la réutilisation de blocs déjà conçus, idée traduite par la notion d'IP. Le chapitre 1 propose une revue des approches de différentes équipes vis-à-vis de cette problématique. Par la suite, les points qui seront étudiés dans ces travaux sont : la définition du contenu d'un IP analogique, le développement d'une base de données d'IPs, le principe d'exploration de cette base de données dans un flot de conception système et enfin une méthodologie de redimensionnement d'un IP dans le cadre d'une migration technologique.

Les différentes questions abordées dans cette thèse ont été appliquées dans un contexte particulier : il s'agit de circuits intégrés neuromimétiques dont l'utilisation première est de constituer le coeur de calcul de simulateurs dédiés dans le domaine des Neurosciences Computationnelles. Ce contexte et les spécificités de ces circuits sont explicités dans le chapitre 2.

Parmi les besoins énoncés en amont, le chapitre 3 propose une réponse à la question centrale de l'IP analogique et de son intégration au sein d'une base de données. Celui-ci possède sept vues qui permettent d'utiliser différents niveaux de modélisation et contient des informations caractéristiques de cet IP et relatives aux méthodes de conception. Une procédure de caractérisation des IPs neuromimétiques est également détaillée dans ce chapitre. La structure de la base de données regroupant ces IPs a été conçue de manière à faciliter son exploration. Cette bibliothèque d'IPs permet la conception automatisée d'ASICs neuromimétiques.

Le chapitre 4 présente trois exemples d'applications des méthodes et outils développés auparavant. Ils montrent comment la bibliothèque d'IPs peut être utilisée pour automatiser la conception de circuits et systèmes neuromimétiques. La première application concerne la conception de l'ASIC « Galway » et permet de présenter les différentes étapes de cette conception automatisée : des spécifications, en passant par l'exploration de la bibliothèque jusqu'à la vérification du système obtenu. La deuxième application traite le cas d'un échec dans l'exploration de la bibliothèque d'IPs et montre comment les informations relatives à un IP de la bibliothèque peuvent aider à la reconception rapidement un nouvel IP, par ajustement

des variables de conception. La troisième application prend l'exemple d'un nouveau projet de conception, visant à développer *ab initio* une fonction non encore envisagée dans la bibliothèque. L'ajout d'une nouvelle fonction permet d'enrichir et de pérenniser la base de données ; les étapes préliminaires de cette conception sont ici exposées. Cette étude préliminaire permet de conclure favorablement sur la pertinence d'introduire cette nouvelle fonction dans les futurs ASICs à concevoir.

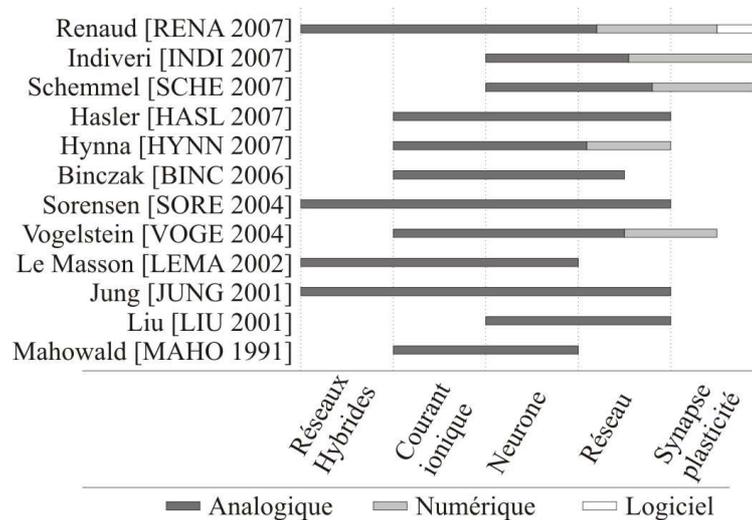
Le chapitre 5 concerne le redimensionnement de circuits CMOS. Ce cas peut correspondre en effet à un échec partiel de l'exploration de la base de données : un IP existe, avec des spécifications convenables, mais il n'est pas dans la technologie souhaitée. Une migration technologique est nécessaire, son objectif est de conserver les performances de l'IP existant lors de la migration. Une fois exposé un état de l'art de cette question spécifique, une méthodologie de redimensionnement est exposée et appliquée avec succès sur des circuits linéaires (OTAs) et non-linéaires (oscillateurs en anneau). Les exemples envisagés s'appuient sur des technologies CMOS 0.35 μm , 0.25 μm et 0.12 μm ayant respectivement 3.3, 2.5 et 1.2 V pour tension d'alimentation.

A la suite des travaux de cette thèse, diverses perspectives sont envisagées.

Pour reprendre le dernier point, une méthode de redimensionnement de circuits bipolaires est à l'étude. La procédure est similaire à celle relative aux circuits CMOS. Ce travail permettra, à moyen terme, de développer une méthodologie générale pour les circuits BiCMOS largement utilisés dans notre domaine d'application.

Par ailleurs, ces travaux ont défini une plateforme de conception à base d'IPs. Elle servira lors des prochains projets de conceptions de l'équipe ISN du laboratoire IMS. Le gain de temps obtenu grâce à la recherche automatique dans la bibliothèque d'IPs permettra au concepteur de porter éventuellement son attention sur la conception de fonctions nouvelles. Une perspective à court terme est l'envoi en fonderie d'un ASIC possédant entre autre une nouvelle fonction neuromorphique de cinétique à deux constantes de temps, celle étudiée dans le chapitre 4.

Enfin, un objectif à plus long terme est d'élargir l'automatisation de la conception de l'ASIC, qui intègre les neurones et synapses, vers le système complet, matériel et logiciel, qui intègre un réseau de neurones et la gestion des règles de plasticité qui le régissent. La figure ci-dessous illustre la problématique de l'Ingénierie Neuromorphique et situe les travaux de l'équipe ISN (Renaud [RENA 2007]) dans l'état de l'art.



Généralement, les parties matérielles analogiques intègrent les niveaux bas dans la modélisation de réseaux de neurones biologiques (niveau courant ioniques, neurones, synapses) tandis que les parties matérielles numériques et logicielles sont le support de la connectivité et de la plasticité du réseau de neurones. Selon la précision du modèle choisi pour représenter l'activité d'un neurone, la réalisation de réseaux de neurones hybrides (artificiel / vivant) est possible ou non.

L'objectif recherché est donc d'automatiser la conception de système neuromorphique complet, matériel et logiciel, sachant que la conception du cœur analogique (ASIC) repose maintenant sur la plateforme exposée dans de mémoire. Une autre thèse est en cours à ce sujet : Bilel Belhadj travaille en effet sur l'intégration de réseaux de neurones complexes avec fonctions de plasticité reconfigurables.

PUBLICATIONS DE L'AUTEUR

Publications dans des revues internationales avec comité de lecture

T. LEVI, J. TOMAS, N. LEWIS, P. FOUILLAT, "A CMOS Resizing Methodology for Analog Circuits: linear and non-linear applications", *IEEE Journal Design and Test of Computer*, 2007, accepté

T. LEVI, N. LEWIS, J. TOMAS, P. FOUILLAT, "IP-based design for analogue ASICs: A case study", *Design&Reuse Articles*, 7 Mai 2007

T. LEVI, N. LEWIS, J. TOMAS, S. SAIGHI, S. RENAUD, Y. BORNAT, L. ALVADO, "Neuromimetic Integrated Circuits", VLSI Circuits for Biomedical Applications, Iniewski Editor, sous presse

T. LEVI, N. LEWIS, J. TOMAS, P. FOUILLAT, "IP-based design for analogue ASICs: A case study", Top stories of the month, einfochips Dashboard, Mai 2007, Volume IV, Issue V, www.einfochips.com/newsletters/may_07.html

Publications dans des conférences internationales avec comité de lecture et actes

T. LEVI, N. LEWIS, J. TOMAS, P. FOUILLAT, "Scaling Guidelines for CMOS Linear Analog Design", IEEE PRIME 2007, Bordeaux, France, 2-5 Juillet 2007

T. LEVI, J. TOMAS, N. LEWIS, P. FOUILLAT, "Resizing methodology for CMOS analog circuit", *SPIE Microtechnologies for the New Millenium 2007*, Maspalomas, Gran Canaria, Espagne, 2-4 Mai 2007

T. LEVI, J. TOMAS, N. LEWIS, P. FOUILLAT, "IP-based design reuse for analogue systems: a case study", *SPIE Microtechnologies for the New Millenium 2007*, Maspalomas, Gran Canaria, Espagne, 2-4 Mai 2007

T. LEVI, N. LEWIS, Y. BORNAT, A. DESTEXHE, J. TOMAS, "An IP-based library for the design of analog Hodgkin Huxley neurons", *Systems, Signal and Devices Conference SSD 2007*, Hammamet, Tunisie, 19-22 Mars 2007

T. LEVI, N. LEWIS, J. TOMAS, P. FOUILLAT, "IP-based design for analogue ASICs: A case study", *IP-based SoC Design Conference 2006, IP-SoC 2006*, Grenoble, France, 6-7 Décembre 2006, pp. 135-139

T. LEVI, N. LEWIS, J. TOMAS, P. FOUILLAT, "Scaling Rules for MOS Analog Design Reuse", *Mixed Design of Integrated Circuits and Systems MIXDES 2006*, Gdynia, Pologne, 22-24 Juin 2006, pp 378-382

J. TOMAS, Y. BORNAT, S. SAIGHI, T. LEVI, S. RENAUD, "Design of a modular and mixed neuromimetic ASIC", *ICECS 2006, 13th IEEE International Conference on Electronics, Circuits and Systems*, Nice, France, 10-13 Décembre 2006, pp. 946-949

Publications dans des conférences nationales avec comité de lecture et actes

T. LEVI, N. LEWIS, J. TOMAS, P. FOUILLAT, "IP-Based Library for Analog Design Reuse", GDR SoC-SIP, Paris, France, 13-15 Juin 2007

T. LEVI, "Implantation sur Silicium en Technologie CMOS d'un Modèle de Neurones", *Journées Nationales du Réseau Doctoral de Microélectronique, JNRDM 2005*, Paris, France, 10-12 Mai 2005

N. LEWIS, Y. BORNAT, L. ALVADO, C. LOPEZ, A. DAOUZLI, T. LEVI, J. TOMAS, S. SAIGHI, S. RENAUD, "PAX : un outil logiciel / matériel d'investigation pour les neurosciences computationnelles", *NeuroComp, 1ère conférence française de Neurosciences Computationnelles*, Pont-à-Mousson, 23-24 octobre 2006, pp 171-17

BIBLIOGRAPHIE

- [ABIT 2000] S. Abiteboul, P. Buneman, D. Suciu, Data on the web from relations to semistructured data and XML, Morgan Kaufmann Publishers, 2000
- [ACOS 2002] R. Acosta, F. Silveira, P. Aguirre, *Experiences on Analog Circuit Technology Migration and Reuse*, Proceedings Integrated Circuits and Systems Design, Septembre 2002 pp 169-174
- [ALVA 2003] L. Alvado, Neurones artificiels sur Silicium : évolution vers les réseaux, Thèse, Université Bordeaux 1, 2003
- [BAKE 1998] R. J. Baker, H. W. Li, D. E. Boyce, CMOS Circuit design, layout, and simulation, IEEE Press series on Microelectronic Systems, 1998
- [BARC] <http://www.barcelonadesign.com>
- [BINC 2006] S. Binczak, S., Jacquir, J. Bilbault, V.B. Kazantsev, V.I. Nekorkin, *Experimental study of electrical FitzHugh–Nagumo neurons with modified excitability*, Neural Networks, pp. 684-693, 2006
- [BORN 2005] Y. Bornat, J. Tomas, S. Saïghi, S. Renaud, *BiCMOS Analog Integrated Circuits for Embedded Spiking Neural Networks*, Proceedings DCIS 2005, Lisbonne, Portugal, Novembre 2005
- [BORN 2006] 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, Université Bordeaux 1, Décembre 2006
- [BOWE 2000] R. Bowen, K. Coar, Apache server, Campus Press, 2000
- [BRIC 1998] P.J. Bricaud, *IP Reuse creation for system-on-a-chip design*, Mentor graphics corporation, Sophia Antipolis, 1998
- [CADE] <http://www.cadence.com>
- [CAI 2003] L. Cai, D. Gadjski, *Transaction Level Modeling in System Level Design*, CECS Technical Report 03-10, 2003

- [CARN 2006] N.T. Carnevale, M.L. Hines, The NEURON Book, Cambridge, UK: Cambridge University Press, 2006
- [CAST 2000] R. Castro-Lopez, F.V. Fernandez, A. Rodriguez-Vasquez, *Reusability methodology for mixed-signal IC designs at layout and schematic levels*, Proceedings of Design of Circuits and Integrated Systems DCIS 2000, pp. 492-497, Montpellier, France, 21-24 Novembre 2000
- [CAST 2002] R. Castro-Lopez, F.V. Fernandez, F. Medeiro, A. Rodriguez-Vasquez, *Generation of technology-independent retargetable analog blocks*, Analog Integrated Circuits and Signal Processing, Vol. 33, pp. 57-70, Kluwer Academic Publishers, Novembre 2002
- [CAST 2006] R. Castro-Lopez, F.V. Fernandez, O. Guerra-Vinuesa, A. Rodriguez-Vasquez, Reuse-Based Methodologies and Tools in the Design of Analog and Mixed-Signal Integrated Circuits, Springer Publishers, 2006
- [CUNH 1998] A. Cunha, M. Schneider, C. Galup-Montoro, *An MOS Transistor Model for Analog Circuit Design*, IEEE Journal of Solid-State Circuits, Vol. 33, No 10, pp. 1510-1519, Octobre 1998
- [DEMA 1996] T. DeMassa, Z. Ciccone, Digital Integrated Circuits, John Wiley and Sons, 1996
- [DERA 2000] C. De Ranter, B. De Muer, G. Van der Plas, P. Vancorenland, M. Steyaert, G. Gielen, W. Sansen, *CYCLONE : automated design and layout of RF LC-oscillators*, Proceedings 37^{ème} IEEE Design Automation Conference, pp. 11-14, Los Angeles, Etats-Unis, 2000
- [DESG 2004] P. Desgreys, *Flot de conception AMS*, Workshop final AS SoC-AMS, 15-16 Janvier 2004
- [DESS 2000] M. Dessouky, A. Kaiser, M-M. Louërat, A. Greiner, *Analog Design for Reuse – Case Study : Very Low-voltage $\Delta\Sigma$ Modulator*, IEEE Design Automation and Test in Europe. Conference 2000, pp. 353–360, Paris, France, 27-30 Mars 2000
- [DESS 2001] M. Dessouky, Conception en vue de la Réutilisation de Circuits Analogiques. Application : Modulateur Delta-Sigma à très Faible Tension, Thèse, Université Pierre et Marie Curie, Paris VI, Janvier 2001

Bibliographie

- [DOUE 2000] V. Douence, Circuits et systèmes de modélisation analogique de neurones biologiques, Thèse, Université Bordeaux 1, 2000
- [DUBO 2005] P. Dubois, MySQL, Sams Developer's Library, 2005
- [DUPE 1998] D. Dupéyron, Contribution à l'intégration sur silicium de modèles analogiques de neurones biologiques, Thèse, Université Bordeaux 1, 1998
- [FRAN 2000] K. Francken, P. Vancorenland, G. Gielen, *DAISY: a simulation-based high-level synthesis tool for $\Delta\Sigma$ modulators*, Proceedings ICCAD 2000, pp. 188-192, San José, Etats-Unis, 2000
- [FULL 2003] Brian Fuller, *Il est toujours possible de tenir la promesse de la technologie IP*, EETIMES 10/06/2003
- [FUNA 2000] S. Funaba, A. Kitagawa, T. Tsukada, G. Yokomizo, *A Fast and Accurate Method of Redesigning Analog Subcircuits for Technology Scaling*, Analog Integrated Circuits and Signal, Vol. 25, pp. 299-307, Décembre 2000
- [GALU 2000] C. Galup-Montoro, M.C. Schneider, *Resizing rules for the reuse of MOS analog design*, Proceedings SBCCI 2000 : XIII Symposium on Integrated Circuits and Systems Design, pp. 89-93, Manaus, Brésil, Septembre 2000
- [GALU 2001] C. Galup-Montoro; M.C Schneider, R Coitinho, *Scaling rules allow the reuse of MOS analog design*, Microelectronic Systems Education 2001, pp. 8-9, Las Vegas, USA, 17-18 Juin 2001
- [GALU 2002] C. Galup-Montoro, M.C. Schneider, R. Coitinho, *Resizing Rules for MOS Analog-Design Reuse*, IEEE Design and Test of Computer, Vol. 19 pp. 50-58, Mars 2002
- [GIEL 2000] G. Gielen, R.A. Rutenbar, *Computer-aided design of analog and mixed-signal integrated circuits*, Proceedings IEEE, Vol. 88, pp. 1825-1854, Décembre 2000
- [GILB 2002] B. Gilbert, Design for manufacture, Trade-offs in analog circuit design. The designer's companion, Kluwer Academic Publishers, 2002

- [GIME 2001] R. Gimeno, P. Mitrano, Science Po Cartographie, Février 2001
- [GINE 2002] A.J. Ginés, E. Peralias, A. Rueda, N. Martinez, R. Seepold, *A Mixed-Signal Design Reuse Methodology Based on Parametric Behavioural Models with Non-Ideal Effects*, Proceedings DATE 2002, Paris, France, pp. 310-314, Mars 2002
- [GOER 1999] R. Goering, *A series on the Reuse Methodology Manual*, 5 articles published in EE Times, 1999
- [GOER 2003] R. Goering, *Un consortium promet de simplifier la réutilisation de l'IP dans les systèmes sur puce*, EETIMES 27/03/2003
- [GOER 2006] R. Goering, *True Circuits rolls out 65-nm analog IP*, EETIMES 12/06/2006
- [GUAR 2002] C. Guardiani, P. McNamara, L. Daldoss, S. Zanella, S. Saxena, S. Liu; W. Xiang, *Analog IP testing: diagnosis and optimization*, Proceedings Design, Automation and Test in Europe Conference and Exhibition 2002, pp. 192-196, Paris, France, 4-8 Mars 2002
- [HAMM 2004a] S. Hammouda, M. Dessouky, M. Tawfik, W. Badawy, *Analog IP Migration Using Design Knowledge Extraction*, International Conference Computer Aided Design ICCAD 2004, San Jose, USA, 7-11 Novembre, 2004
- [HAMM 2004b] S. Hammouda, M. Dessouky, M. Tawfik, W. Badawy, *A Fully Automated Approach for Analog Circuit Reuse*, Proceedings System on chip for real time applications 2004, pp 237-240, Banff, Canada, 19-21 Juillet 2004
- [HAMO 2003] Hamour, M.; Saleh, R.; Mirabbasi, S.; Ivanov, *Analog IP design flow for SoC applications*, Proceedings of the 2003 International Symposium ISCAS, Vol 4, pp. 676-679, Bangkok, Thailand, 25-28 May 2003
- [HASL 2007] P. Hasler, S. Kozoil, E. Farquhar, A. Basu, *Transistor Channel Dendrites implementing HMM classifiers*, Proceedings of IEEE International Symposium on Circuits And Systems, pp. 3359-3362, New-Orleans, USA, 2007

Bibliographie

- [HERV 2004] Y. Hervé, *Bibliothèques d'IP-AMS*, Workshop final AS SoC-AMS, 15-16 Janvier 2004
- [HODG 1952a] A. L. Hodgkin, A. F. Huxley, B. Katz, *Measurements of current-voltage relations in the membrane of the giant axon of loligo*, Journal of Physiology, Vol. 116, pp. 424-448, 1952
- [HODG 1952b] A. L. Hodgkin, A. F. Huxley, *Currents carried by sodium and potassium ions through the membrane of the giant axon of loligo*, Journal of Physiology, Vol. 116, pp. 449-472, 1952
- [HODG 1952c] A. L. Hodgkin, A. F. Huxley, *The components of membrane conductance in the giant axon of loligo*, Journal of Physiology, Vol. 116, pp. 473-496, 1952
- [HODG 1952d] A. L. Hodgkin, A. F. Huxley, *The dual effect of membrane potential on sodium conductance in the giant axon of loligo*, Journal of Physiology, Vol. 116, pp. 497-506, 1952
- [HODG 1952e] A. L. Hodgkin, A. F. Huxley, *A quantitative description of membrane current and its application to conduction and excitation in nerve*, Journal of Physiology, Vol. 117, pp. 500-544, 1952
- [INDI 2007] G. Indiveri, S. Fusi, *Spike-based learning in VLSI networks of integrate-and-fire neurons*, Proceedings of IEEE International Symposium on Circuits And Systems, pp. 3371-3374, New-Orleans, USA, 2007
- [ISKA 2005] R. Iskander, L. De Lamarre, P. Nguyen Tuong, M-M. Louërat, A. Kaiser, *Synthèse d'un IP amplificateur analogique CMOS avec CAIRO+*, 6^{ème} Colloque sur le Traitement Analogique de l'Information du Signal et ses Applications TAISA 2005, pp. 69-72, Marseille, France, Octobre 2005
- [ITRS 2001] International Technology Roadmap for Semiconductors, Edition 2001
Available : <http://www.itrs.net/Links/2001ITRS/Home.htm>
- [ITRS 2006] International Technology Roadmap for Semiconductors, Edition 2006
Available : <http://www.itrs.net/Links/2006Update/Home.htm>

- [JANG 2003] N. Jangkrajarn, S. Bhattacharya, R. Hartono, C.-J. R. Shi, *IPRAIL – Intellectual Property Reuse-based Analog IC layout Automation*, Integration the VLSI Journal, Vol. 36, pp. 237-262, Novembre 2003
- [JUNG 2001] R. Jung, E.J. Brauer, J.J. Abbas, *Real-time interaction between a neuromorphic electronic circuit and the spinal cord*, IEEE Transactions on Neural Systems and Rehabilitation Engineering, pp. 319–326, 2001
- [KARA 2000] T. Karayiannis, J. Mades, T. Schneider, A. Windisch, W. Ecker, *Using XML for representation and visualization of elaborated VHDL-AMS models*, Proceedings of VHDL International Users Forum Fall Workshop 2000, pp. 83-87, Orlando, USA, 18-20 Octobre 2000
- [KEAT 1999] M. Keating, P. Bricaud, Reuse methodology manual, 2nd Edition, Kluwer Academic Publishers, 1999
- [KOCH 2003] R.J. Koch, F. Dielacher, *Analog IP – stairway to SoC heaven ?*, Proceedings IEEE Solid-State Circuits Conference, pp. 1-2, 2003
- [LAFL 1998] A. Laflaquière, Neurones artificiels sur silicium : conception analogique et construction de réseaux hybrides, Thèse, Université Bordeaux 1, 1998
- [LEE 2003] T. H. Lee, The Design of CMOS Radio-Frequency Integrated Circuits, Second Edition, Cambridge University Press, 2003
- [LERD 2002] R. Lerdorf, K. Tatroe, Programming Php, O'Reilly publisher, 2002
- [LEGA 2005] B. Le Gal, Contribution à la prise en compte des contraintes des applications TDSI dans la synthèse de haut niveau, Thèse, Université de Bretagne Sud, Lorient, Décembre 2005
- [LEMA 2002] G. LeMasson, S. Renaud, D. Debay, *Feedback inhibition controls spike transfer in hybrid thalamic circuits*. Nature, n°4178, pp. 854-858, 2002
- [LEVI 2006a] T. Levi, N. Lewis, J. Tomas, P. Fouillat, *Scaling Rules for MOS analog design reuse*, Proceedings IEEE MIXDES 06, pp. 378-382, Gdynia, Poland, Mai 2006
- [LEWI 2004] N. Lewis, *Modélisation comportementale*, Workshop final AS SoC-AMS, 15-16 Janvier 2004

- [LI 2003] Z. Li, L. Luo, J. Yuan, *A Study on Analog IP Blocks for Mixed-Signal SoC*, Proceedings ASIC 2003, Vol. 1 pp 564-567, Octobre 2003
- [LIU 2003] D. Liu, S. Sidiropoulos, M. Horowitz, *A Framework for designing Reusable Analog Circuits*, Proceedings ICCAD 2003 pp 375-380, San Jose, USA, 9-13 Novembre 2003
- [LIU 2001] S. Liu, R. Douglas, *Temporal coding in a silicon network of integrate-and-fire neurons*, IEEE Transactions on Neural Networks, pp. 1305-1314, 2001
- [MALA 1999] E. Malavasi, E. Charbon, *Constraint transformation for IC physical design*, IEEE Transactions on Semiconductor Manufacturing, Vol. 12, pp. 386-395, Novembre 1999
- [MART 2001] N. Martinez, E. Peralías, A. Acosta, A. Rueda, *Analog/Mixed-Signal IP modelling for design reuse*, Proceedings DATE Conference 2001, Munich, Allemagne, Mars 2001
- [MART 2002] N. Martínez , R. Seepold, *Virtual Component Reuse and Qualification for Digital and Analogue Design*, System-on-Chip Methodologies & Design Languages, Kluwer Academic Publishers, pp. 307-316, 2001
- [MART 2003] N. Martinez, R. Seepold, A. Vörg, *Definition of IP exchange formats*, Rapport Technique ToolIP FZI-T1.2-Q4/02, Janvier 2003
- [MAHO 1991] M. Mahowald, R. Douglas, *A silicon neuron*, Nature, n°354, pp. 515-518, 1991
- [MCGR 2005] D. McGrath, *British company's new tools promising analog reuse in days*, EETIMES 22/04/2005
- [MONT 2004] M. Montón, O. Font, J. Joven, P. Garcia, L. Terés, J. Carrabina, *XML specification and tools for automatic SoC generation*, Proceedings Design of Circuits and Integrated Systems DCIS 2004, Bordeaux, France, 24-26 Novembre 2004
- [MOOR 1965] G. E. Moore, *Cramming more components onto integrated circuits*, Electronics, Vol. 38, Avril 1965

Bibliographie

- [MOOR 1975] G. E. Moore, *Progress in digital integrated circuit*, IEEE Int. Electron Devices Meeting Technology Digest, pp. 11, Décembre 1975
- [NAJI 2003] T. Najibi, *Enabling analog-IP reuse: relating requirements to reality*, Cadence Design Systems, D&R Industry Articles, 2003
- [NEOL] <http://www.neoliner.com>
- [NGUY 2006] P. Nguyen Tuong, Définition et implantation d'un langage de conception de composants analogiques réutilisables, Thèse, Université Pierre et Marie Curie, Paris VI, Juin 2006
- [O'CON 2004] I. O'Connor, *Synthèse analogique*, Workshop final AS SoC-AMS, 15-16 Janvier 2004
- [O'CON 2005a] I. O'Connor, Systèmes sur puce physiquement hétérogènes: Méthodologies de conception et systèmes optiques d'interconnexion, Habilitation à diriger les recherches, Ecole Centrale de Lyon, France, 2005
- [O'CON 2005b] I. O'Connor, F. Tissafi-Drissi, G. Révy, F. Gaffiot, *UML/XML-based approach to hierarchical AMS synthesis*, Proceedings FDL 2005, Lausanne, Suisse, 27-30 Septembre 2005
- [ODA 2002] K. Oda, L.A. Prado, A.J. Gadiant, *A new methodology for Analog/Mixed-Signal (AMS) SoC design that enables AMS design reuse and achieves full-custom*, Proceedings Electronics Design Processes EDP 2002, Monterey, USA ,21-23 Avril 2002
- [OHR 2002] S. Ohr, L. Marchant, *PANEL : analog intellectual property : now ? or never ?*, Proceedings ACM/IEEE Design Automation Conference, pp. 181-182, 2002
- [OHR 2003] S. Ohr, *Intérêt grandissant pour l'IP analogique*, EETIMES 04/03/2003
- [PECH 2005] F. Pêcheux, C. Lallement, A. Vachoux, *VHDL-AMS and Verilog-AMS as Alternative Hardware Description Languages for Efficient Modeling of Multidiscipline Systems*, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol. 24, Février 2005

- [PELG 1989] M.J.M. Pelgrom, A.C. Duinmaijer, A.P.G. Welbers, *Matching properties of MOS transistors*, IEEE Journal of Solid-State Circuit, Vol. 24, pp. 1433-1440, Octobre 1989
- [POLT 2007] A. Polti, *TP System C INF 227*, Telecom Paris, 2007
- [PRES 2002] W. Press, S. Teukolsky, W. Vetterling, B. Flannery, Numerical Recipes in C, 2nd Edition Cambridge University Press, 2002
- [REIN 2002] M. Reinhardt, Automatic layout modification, Kluwer Academic Publishers, 2002
- [RENA 2004] S. Renaud, G. Le Masson, L. Alvado, S. Saïghi, J. Tomas, *A neural simulation system based on biologically-realistic electronic neurons*, Journal of Information Science, Vol. 161, pp. 57-69, 2004
- [RENA 2007] S. Renaud, J. Tomas, Y. Bornat, A. Daouzli, S. Saïghi, *Neuromimetic ICs with analog cores: an alternative for designing spiking neural networks*, Proceedings of IEEE International Symposium on Circuits And Systems, pp. 3355-3358, New-Orleans, USA, 2007
- [RIGA 2006] P. Rigaux, Pratique de MySQL et Php, O'Reilly, 2006
- [ROSE 2004] W. Rosenstiel, *IP and design reuse*, Integration the VLSI Journal, Vol. 37, pp. 190-191, 2004
- [SANC 2001] R. Sanchez, *Extraction method verifies IP functions*, EETIMES 26/03/2001
- [SAIG 2004] S. Saïghi, Circuits et systèmes de modélisation analogique de réseaux de neurones biologiques : application au développement d'outils pour les neurosciences computationnelles, Thèse, Université Bordeaux 1, 2004
- [SAIG 2006] S. Saïghi, Y. Bornat, J. Tomas, S. Renaud, *Neuromimetic ICs and System for Parameters Extraction in Biological Neuron Models*, Proceedings ISCAS 2006, pp. 4207-4210, Ile de Kos, Grèce, Mai 2006
- [SAVI 2004] A. Savio, L. Colalongo, Zs. M. Kovacs-Vajna, Proceedings, *Scaling rules and parameter tuning procedure for analog design reuse in technology migration*, Proceedings ISCAS 2004 Vol. 5 pp. 117-120, Vancouver, Canada, Mai 2004

- [SAVI 2006] A. Savio, L. Colalongo, Zs. M. Kovacs-Vajna, *Automatic Scaling rules procedures for analog design reuse*, IEEE Transactions on Circuits and Systems, Vol. 53 pp.2539-2547, Décembre 2006
- [SCHE 2007] J. Schemmel, D. Brüderle, K. Meier, B. Ostendorf, *Modeling Synaptic Plasticity within Networks of Highly Accelerated I&F Neurons*, Proceedings of IEEE International Symposium on Circuits And Systems, pp. 3367-3370, New-Orleans, USA, 2007
- [SEDR 1991] A. S Sedra, K. C. Smith, Microelectronic Circuits Third Edition, International Edition, Saunders College Publishing, 1991
- [SEEP 2002] R. Seepold, N. Martínez, A. Vörg, W. Rosenstiel, M. Radetzki, P. Neumann, J. Haase, *A Qualification Platform for Design Reuse*, Proceedings 2002 International Symposium on Quality of Electronic Design (ISQED 2002), San Jose, USA, 18-20 Mars 2002
- [SEEV 1988] E. Seevicnk, *Analysis and Synthesis of Translinear Integrated Circuits*, Edition Elsevier, Amsterdam, 1988
- [SING 2005] R. Singh, *Analog IP re-use: concerns for “digitally-oriented” SoC designers*, EETIMES 19/12/2005
- [SORE 2004] M. Sorensen, S. DeWeerth, G. Cymbalyuk, R.L. Calabrese, *Using a hybrid neural system to reveal regulation of neuronal network activity by an intrinsic current*, Journal of Neurosciences, pp. 5427-5438, 2004
- [SYST 2003] SystemC 2.0.1 Language Reference Manual, Open SystemC Initiative, 2003
- [POLT 2007] A. Polti, Cours de SystemC, Telecom Paris, 2007
- [VAND 2001] G. Van der Plas, G. Debyser, F. Leyn, K. Lampaert, J. Vandenbussche, G.G.E. Gielen, W. Sansen, P. Veselinovic, D. Leenarts, *AMGIE-A synthesis environment for CMOS analog integrated circuits*, Transactions on IEEE Computer-Aided Design of Integrated Circuits and Systems, Vol. 20, pp. 1037-1058, Septembre 2001
- [VAND 2003] J. Vandenbussche, G. Gielen, M. Steyaert, Systematic design of analog IP blocks, Kluwer Academic Publisher, 2003

Bibliographie

- [VERI 1997] VerilogA, Reference Manual, Cadence, 1997
- [VERI 1999] Verilog, Reference Manual, Synopsys, 1999
- [VERI 2004] Verilog-AMS, Reference Manual, Synopsys, 1999
- [VHDL 1997] VHDL, Reference Manual, Mentor Graphics, 1997
- [VHDL 1999] VHDL-AMS, Reference Manual, Mentor Graphics, 1999
- [VITT 1990] E.A. Vittoz, *Future of Analog in the VLSI Environment*, Proceedings IEEE ISCAS 1990 : International Symposium on Circuits and Systems, pp. 1372-1375, Genève, Suisse, Mai 1990
- [VOGE 2004] R.J. Vogelstein, U. Malik, G. Cauwenberghs, *Silicon spike-based synaptic array and address-event transceiver*, Proceedings of ISCAS'04, pp. 385-388, 2004
- [VSIA 2001a] *Model taxonomy version 2.1 (SLD 2 2.1)*, VSI AllianceTM, 2001
- [VSIA 2001b] *Analog/Mixed-Signal VSI Extension Specification Version 2.2*, VSI AllianceTM, 2001

ANNEXES

ANNEXE 1 : ANATOMIE ET PHYSIOLOGIE DU NEURONE.....171

ANNEXE 2 : MANUEL D'UTILISATION DE LA BIBLIOTHEQUE
D'IPs ET DE SON EXPLORATION.....179

ANNEXE 3 : MODELE FONCTIONNEL DE LA CINETIQUE
A 2 CONSTANTES DE TEMPS.....203

ANNEXE 1

ANATOMIE ET PHYSIOLOGIE DU NEURONE

SOMMAIRE

I	ANATOMIE DU NEURONE.....	172
	<i>I.1.1</i> Le soma	173
	<i>I.1.2</i> Les dendrites	173
	<i>I.1.3</i> L'axone.....	173
	<i>I.1.4</i> Les synapses	173
	<i>I.1.5</i> La membrane plasmique	173
II	PHYSIOLOGIE DU NEURONE.....	174
	<i>II.1.1</i> Le potentiel de repos	174
	<i>II.1.2</i> Le potentiel d'action.....	174
III	MODELE D'UN NEURONE EXCITATEUR.....	175
	<i>III.1</i> PARAMÈTRES DU COURANT SODIUM.....	176
	<i>III.2</i> PARAMÈTRES DU COURANT POTASSIUM	176
	<i>III.3</i> PARAMÈTRES DU COURANT CALCIUM.....	176
	<i>III.4</i> PARAMETRES DU COURANT FUIITE	177
IV	CONCLUSION.....	177
V	BIBLIOGRAPHIE.....	178

I ANATOMIE DU NEURONE

Il existe une grande diversité de neurones en fonction de l'espèce animale, de leur localisation au sein du système nerveux ou de leur fonction. Nous pouvons voir sur la figure 2.2 quelques architectures de neurones.

Cependant il existe une base commune à ces architectures : chaque neurone possède un corps cellulaire appelé soma, des dendrites, un axone et des synapses.

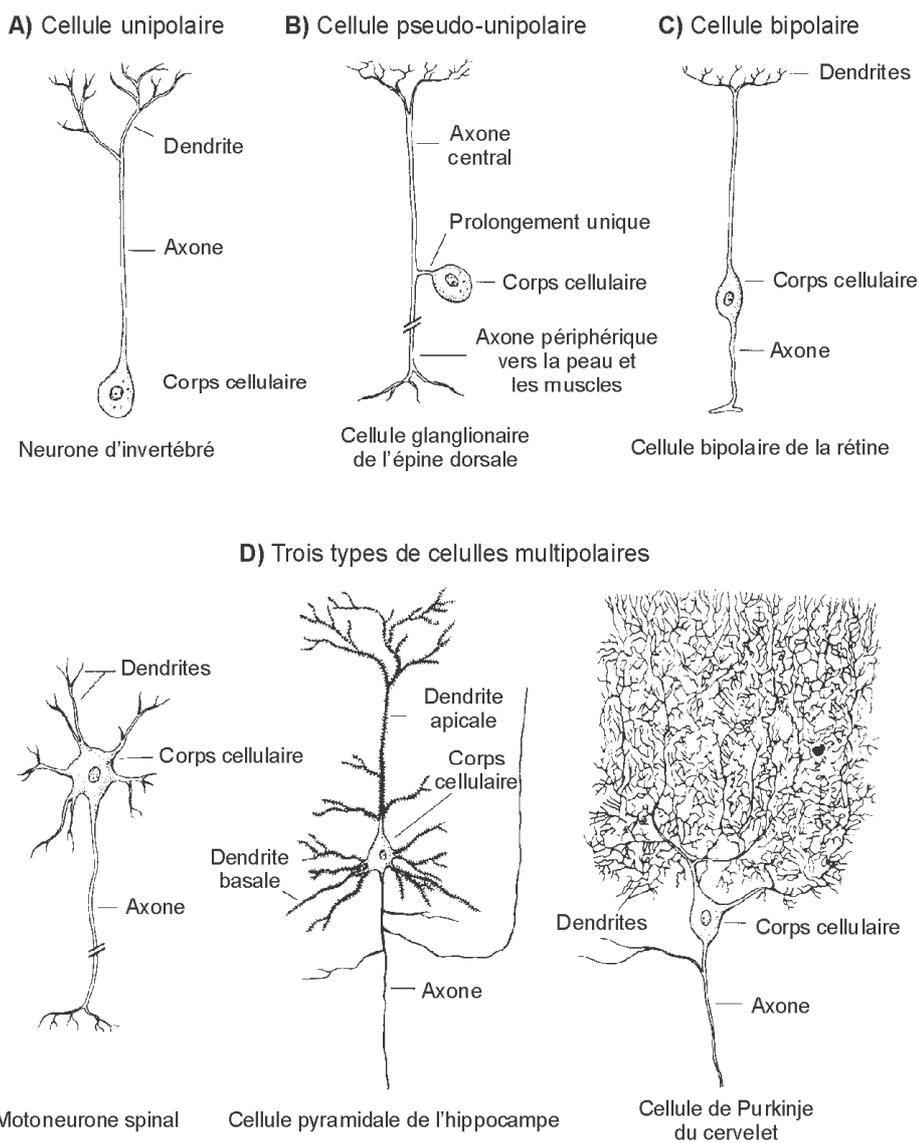


Figure 1. Différents neurones d'après [KAND 1991]

I.1.1 Le soma

La forme du corps cellulaire est variable suivant le type de neurone ; elle peut être pyramidale, ovoïde ou granulaire. Pour le corps humain, le diamètre du soma est de l'ordre de 20 μm .

I.1.2 Les dendrites

Ce sont les prolongements du neurone au niveau du soma. Ces ramifications permettent les connexions synaptiques qui assurent la continuité de la propagation de l'information entre les différents neurones. Les dendrites constituent la surface principale de réception de l'information.

I.1.3 L'axone

L'axone est unique en règle générale. Il peut être myélinisé (substance constituée principalement de lipides). Sa longueur est variable, elle peut aller de quelques microns à plusieurs centimètres. Il se décompose en deux parties ; le segment initial où naît le potentiel d'action et la partie terminale où se situe le dialogue permanent avec ses effecteurs afin d'interagir avec eux.

I.1.4 Les synapses

Ce sont les points de transmissions de l'information. Il existe deux types de synapses : les synapses électriques et les synapses chimiques.

I.1.5 La membrane plasmique

La membrane plasmique neuronale délimite le contour de la cellule. Elle sert de barrière entre le milieu intracellulaire et le milieu extracellulaire. Elle se compose d'une bicouche lipidique traversée par plusieurs types de protéines.

Les milieux exocellulaires et endocellulaires se différencient par leur composition et leur concentration en espèces ioniques.

- Le milieu exocellulaire est riche en ions Na^+ et Cl^- mais pauvre en ions K^+ .
- Le milieu endocellulaire est pauvre en ions Na^+ et Cl^- mais riche en ions K^+ .

Après avoir décomposé le neurone, nous présentons maintenant sa physiologie.

II PHYSIOLOGIE DU NEURONE

La physiologie du neurone peut être décomposé en deux étapes, le potentiel de repos et le potentiel d'action.

II.1.1 Le potentiel de repos

Lorsque la membrane n'est pas soumise à une excitation électrique, le système constitué de la membrane, du milieu extracellulaire et du milieu intracellulaire atteint un équilibre.

Nous pouvons alors mesurer ce potentiel, appelé potentiel de repos, il est de l'ordre de -65 mV.

II.1.2 Le potentiel d'action

Les canaux ioniques ont une activité dépendante de la différence de potentiel entre les milieux intra et extracellulaire.

La propriété de ces canaux donne naissance à un phénomène électrique se propageant le long de l'axone : le potentiel d'action.

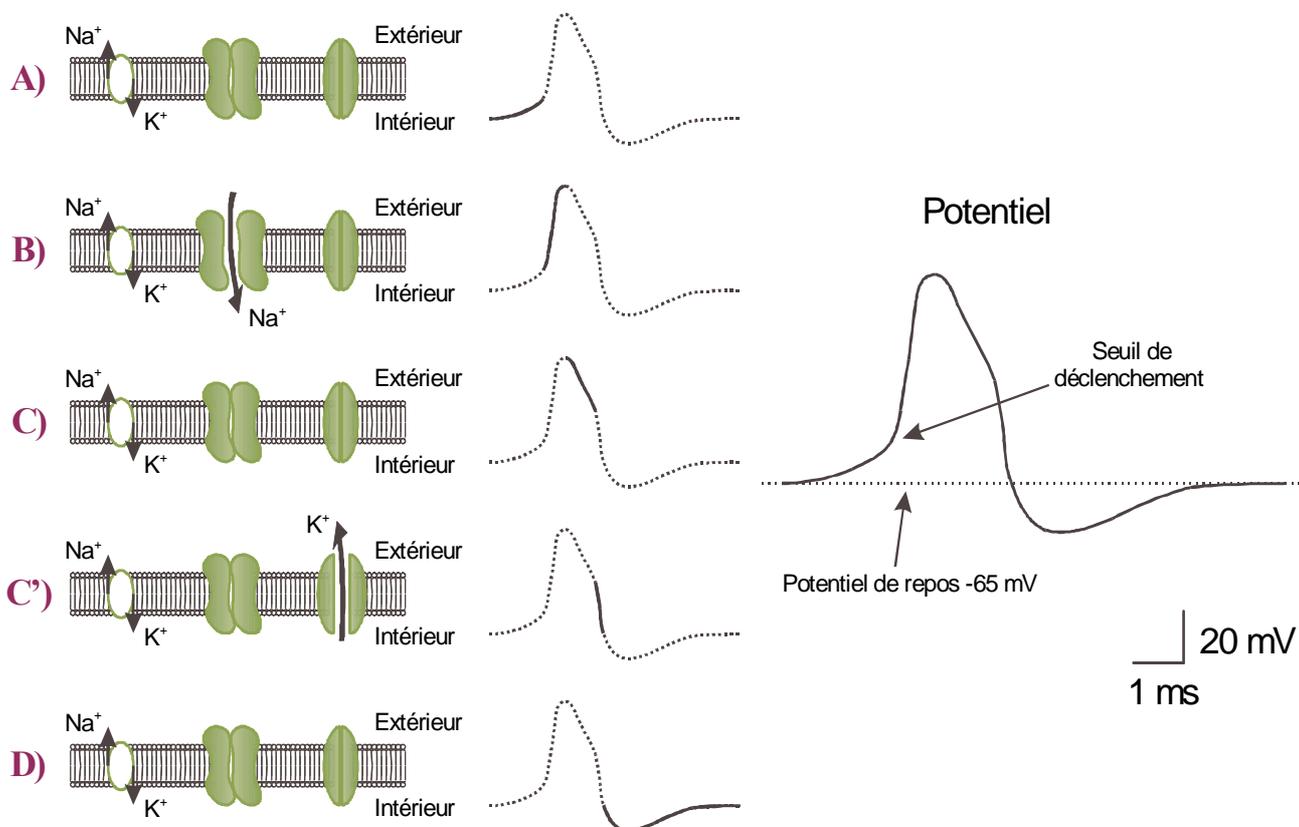


Figure 2. Description du potentiel d'action

Suite à une stimulation électrique, la membrane commence par se dépolariser (A).

Si le phénomène dure, la dépolarisation déclenche l'ouverture rapide des canaux Na^+ . L'entrée des ions Na^+ continue de dépolariser la membrane même après l'arrêt de la stimulation conduisant la membrane hors de son état d'équilibre (B).

Le potentiel ne croît cependant pas jusqu'au potentiel d'équilibre du sodium, car très vite deux processus entrent en action. Le premier est l'inactivation des canaux sodium, ce qui permet au courant de fuite de commencer à repolariser la membrane (C). Toujours grâce au potentiel atteint, mais avec après un certain délai, l'ouverture des canaux K^+ accélère le phénomène de repolarisation (C').

On retrouve également l'inertie de changement d'état des canaux potassiques à leur fermeture ce qui provoque une hyperpolarisation transitoire de la membrane (D).

III MODELE D'UN NEURONE EXCITATEUR

Nous donnons dans ce paragraphe, les différents paramètres du modèle d'un neurone exciteur présenté dans le chapitre 2 à la figure 2.5. Ce neurone possède quatre courants ioniques (Na, K, Ca, Leak).

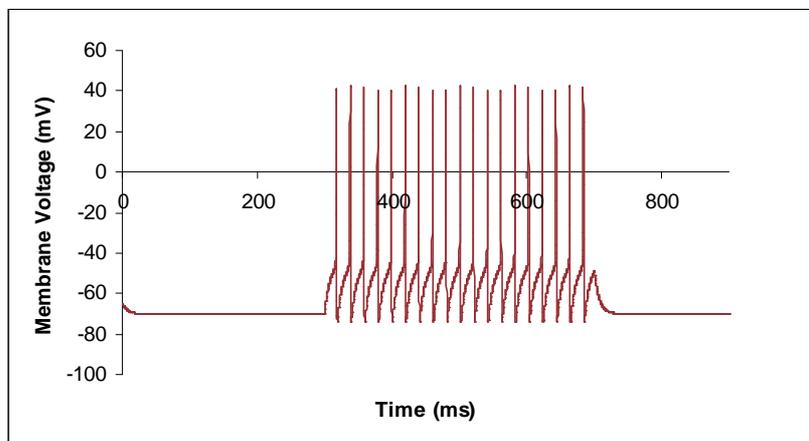


Figure 3. Activité électrique du neurone exciteur

Ce neurone possède une capacité de membrane de $1 \mu\text{F} \cdot \text{cm}^{-2}$ et une surface de 0.00022 cm^2 .

Nous rappelons l'équation générale d'un courant ionique : $I_{ion} = g_{max} m^p h^q (V_{mem} - V_{equi})$

III.1 Paramètres du courant Sodium

$$I_{Na} = g_{Na} * m^3 * h * (V - V_{equiNa})$$

$$g_{Na} = 0.05 \text{ S/cm}^2$$

$$V_{equiNa} = 50 \text{ mV}$$

TABLEAU 1. VALEURS DES PARAMÈTRES D'ACTIVATION ET D'INACTIVATION

Paramètres	m	h
V_{offset}	-29 mV	-33 mV
V_{slope}	6.61 mV	-4 mV
tau	0.08 ms	3 ms pour $V_{mem} < -20 \text{ mV}$ 0.8 ms pour $V_{mem} > -20 \text{ mV}$

III.2 Paramètres du courant Potassium

$$I_K = g_K * n^4 * (V - V_{equiK})$$

$$g_K = 0.005 \text{ S/cm}^2$$

$$V_{equiK} = -100 \text{ mV}$$

n :

$$V_{offset} = -31 \text{ mV}$$

$$V_{slope} = 8.5 \text{ mV}$$

$$\text{tau} = 1.2 \text{ ms}$$

III.3 Paramètres du courant Calcium

$$I_{Ca} = g_{Ca} * m^2 * h * (V - V_{equiCa})$$

$$g_{Ca} = 0.001 \text{ S/cm}^2$$

$$V_{equiCa} = 120 \text{ mV}$$

TABLEAU 2. VALEURS DES PARAMETRES D'ACTIVATION ET D'INACTIVATION

Paramètres	m	h
V_{offset}	-57 mV	-81 mV
V_{slope}	6.2 mV	-4 mV
tau	8 ms	16.5 ms

III.4 Paramètres du courant Fuite

$$I_L = g_{\text{Fuite}} * (V - V_{\text{equiFuite}})$$

$$g_{\text{Fuite}} = 0.001 \text{ S/cm}^2$$

$$V_{\text{equiFuite}} = 120 \text{ mV}$$

IV CONCLUSION

Nous venons de présenter l'anatomie et la physiologie du neurone. Plusieurs travaux concernant les débuts de l'ingénierie neuromorphique, sont très intéressants et importants pour leurs avancées, nous les donnons ici dans la bibliographie.

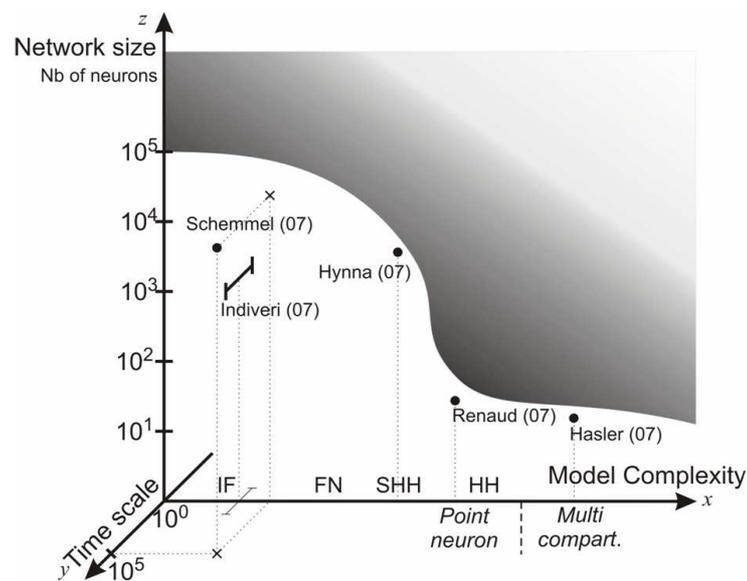


Figure 4. Etat de l'art des ASICs neuromorphiques

Cette figure représente l'état de l'art dans la conception d'ASICs neuromorphiques. Nous la présentons pour permettre au lecteur d'identifier les différentes équipes qui travaillent sur ce sujet et ainsi approfondir ce domaine s'il le souhaite.

V BIBLIOGRAPHIE

- [CALA 1995] R. L. Calabrese, *Half-Center Oscillators Underlying Rhythmic Movements*, pp. 444-447 dans The Handbook of Brain Theory and Neural Networks de M. A. Arbib, MIT Press, Cambridge
- [DOUG 1995] R. Douglas, M. Mahowald, C. Mead, Neuromorphic Analogue VLSI, Oxford University Press, New York, 1995
- [HAHN 2000] R. Hahnloser, R. Sarkeshpar, M. Mahowald, R. Douglas, S. Seung, *Digital selection and analogue coexist in a cortex-inspired silicon circuit*, Letters to Nature, Vol. 405, pp. 947-951, juin 2000 [KAND 1991] E. R. Kandel, J. H. Schwartz, T. M. Jessell, Principles of neural science, Troisième édition, Prentice Hall International, Londres, 1991
- [LIU 2004] S-C Liu, R. Douglas, *Temporal coding in a silicon network of integrate and fire neurons*, IEEE Transactions of Neural Networks, Vol. 15, pp. 1305-1314, 2004
- [MAHO 1991] M. Mahowald, R. Douglas, *A silicon neuron*, Nature, Vol 354, pp. 515-518, décembre 1991
- [PATE 1997] G. N. Patel, S.P. DeWeerth, *Analogue VLSI Morris-Lecar neuron*, Electronics Letters, Vol. 33, pp. 997-998, 5 juin 1997
- [SIMO 2004] M. F. Simoni, G. S. Cymbalyuk, M. E. Sorensen, R. L. Calabrese, S. P. DeWeerth, *A Multiconductance Silicon Neuron With Biologically Matched Dynamics*, IEEE Transactions on Biomedical Engineering, Vol. 51, pp. 342-354, février 2004

ANNEXE 2

MANUEL D'UTILISATION DE LA BIBLIOTHEQUE D'IPs ET DE SON EXPLORATION

SOMMAIRE

I	INTRODUCTION.....	180
II	DESCRIPTION DE LA BASE DE DONNEES.....	180
II.1	DEFINITION DE CHAQUE IP OU IP	182
II.2	LIEN PERE-FILS ENTRE LES FONCTIONS NEUROMORPHIQUES OU LOWERBLOCKLINKS	183
II.3	LIENS PERE-FILS ENTRE LES IPs OU IPCURRENT_LINKS, IPSYSTEM_LINKS, IPNEURON_LINKS	184
II.4	NOMBRE D'IPs POUR UNE FONCTION DONNEE OU BLOCKS.....	185
II.5	VALEUR DES SPECIFICATIONS OU CURRENT_LEVEL, FUNCTION_LEVEL, NEURON_LEVEL	186
II.6	CONNEXIONS ENTRE LES DIFFERENTS MACRO-BLOCS OU IO_FINAL.....	187
III	AJOUT D'IP.....	188
IV	PRESENTATION DE L'EXPLORATION	189
IV.1	ENTREE DES SPECIFICATIONS PAR LE CLIENT	192
IV.1.1	<i>Choix du concepteur, création des requêtes et exploration de la base de données.....</i>	<i>195</i>
IV.1.2	<i>Récapitulatif et résultats.....</i>	<i>197</i>
IV.2	DESCRIPTION DES FICHIERS UTILISES	199
IV.3	DESCRIPTION DES LANGAGES UTILISES	200
IV.3.1	<i>Langage PhP.....</i>	<i>200</i>
IV.3.2	<i>Langage SQL.....</i>	<i>201</i>
IV.3.3	<i>Langage XML.....</i>	<i>201</i>
V	CONCLUSION.....	202

I INTRODUCTION

Ce manuel est une aide pour la compréhension et l'utilisation de la bibliothèque. Il présente les mécanismes et le contenu de la base de données d'IPs. Les différents fichiers créés sont décrits et l'exploration de la base est présentée.

II DESCRIPTION DE LA BASE DE DONNEES

Rappelons que le système étudié possède différents niveaux hiérarchiques dans sa description (figure 1). L'IP élémentaire qui possède 7 vues correspond au niveau *Cellule*. Au niveau *Bloc* et *Macro-bloc*, des IPs ont également été définis, contenant une description structurelle à partir des IPs élémentaires, ainsi que les spécifications résultantes.

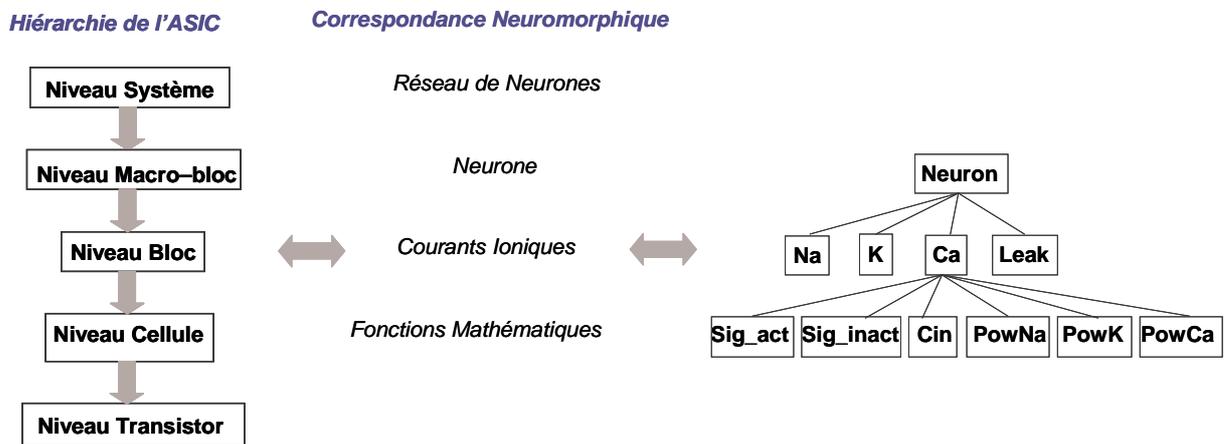


Figure 1. Présentation de la hiérarchie de notre cas d'étude

La base de donnée a été créée en utilisant le langage SQL. Plusieurs tables ont été définies et des programmes en HTML, XML, et PhP ont été créés pour exploiter au mieux ces données.

Voici la présentation des différentes tables définies pour la création de cette base de données.

Il existe 10 tables :

- Définition de chaque IP : 1 table, *Ip*
- Lien père-fils entre les fonctions neuromorphiques (figure 3.14) : 1 table, *lowerblocklinks*
- Lien père-fils entre les IPs (figure 3.14) : 3 tables, *ipcurrent_links*, *ipneuron_links*, *ipsystem_links*
- Nombre d'IPs pour une fonction donnée : 1 table, *blocks*
- Valeurs des spécifications (dynamique des entrées) des IPs pour chaque niveau de hiérarchie existant (*Cellule*, *Bloc*, *Macro-bloc*) : 3 tables, *function_level*, *current_level*, *neuron_level*
- Connexions entre les différents *Macro-blocs* constituant le système : 1 table, *io_final*

Base de données *reseauneurones* sur le serveur *localhost*

Structure SQL Exporter Rechercher Requête Supprimer

Table	Action	Enregistrements	Type	Taille
<input type="checkbox"/> blocks	[Icons]	33	MyISAM	3,2 Ko
<input type="checkbox"/> current_level	[Icons]	5	MyISAM	2,3 Ko
<input type="checkbox"/> function_level	[Icons]	7	MyISAM	2,3 Ko
<input type="checkbox"/> io_final	[Icons]	3	MyISAM	1,7 Ko
<input type="checkbox"/> ip	[Icons]	40	MyISAM	15,3 Ko
<input type="checkbox"/> ipcurrent_links	[Icons]	7	MyISAM	1,8 Ko
<input type="checkbox"/> ipneuron_links	[Icons]	3	MyISAM	1,4 Ko
<input type="checkbox"/> ipsystem_links	[Icons]	3	MyISAM	1,2 Ko
<input type="checkbox"/> lowerblocklinks	[Icons]	54	MyISAM	3,1 Ko
<input type="checkbox"/> neuron_level	[Icons]	3	MyISAM	2,9 Ko
10 table(s)	Somme	158	--	35,2 Ko

Tout cacher / Tout décocher Pour la sélection : [v]

- [Version imprimable](#)
- [Dictionnaire de données](#)
- Créer une nouvelle table sur la base *reseauneurones* :
Nom :

Figure 2. Différentes tables de la base de données

Nous allons maintenant présenter plus en détail chaque catégorie de table.

II.1 Définition de chaque IP ou Ip

La table *définition d'IP* regroupe tous les IPs existants avec leur contenu. Tous les liens vers les différentes vues telles que toutes les vues de modélisation, les vues symbole, layout, schéma de même que la technologie, la tension d'alimentation et la surface du layout sont données. Le niveau hiérarchique de l'IP et les différentes entrées/sorties sont aussi définis.

Le champ *level* définit le niveau hiérarchique de l'IP, 1 pour une fonction mathématique, 2 pour un courant ionique et 3 pour un neurone.

Le champ *Values* explique si l'IP possède des spécifications à rentrer dans les tables *current_level*, *function_level* ou *neuron_level*.

Base de données reseauneurones - Table ip sur le serveur localhost

Chang	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 3. Table Ip

II.2 Lien père-fils entre les fonctions neuromorphiques ou lowerblocklinks

La table *lien père-fils fonction* présente les différents liens hiérarchiques pour une fonction donnée. C'est-à-dire qu'elle présente tous les *blocs* (ou respectivement *cellules*) du niveau hiérarchique inférieur qui sont utilisées pour concevoir ce *macro-bloc* (ou respectivement *bloc*). Il existe trois tables, une pour chaque niveau hiérarchique existant (*Cellule*, *Bloc*, *Macro-bloc*). Ceci est illustré à la figure 1, pour un neurone ainsi que pour un courant ionique calcium (Ca).

		← T →	father	son
			5c_neuron	buffer
			5c_neuron	ca
			5c_neuron	k
			5c_neuron	leak
			5c_neuron	na
			5c_neuron	pol_neuron
			5c_neuron	stim
			5c_neuron	synapse
			buffer	aop_buffer
			buffer	switch
			ca	kinetic
			ca	output_level
			ca	pol_function
			ca	pow_ca
			ca	sigmoid_act
			ca	sigmoid_inact
			ca	switch
			fs_neuron	buffer

Figure 4. Table *lowerblocklinks*

II.3 Liens père-fils entre les IPs ou *ipcurrent_links*, *ipsystem_links*, *ipneuron_links*

Les tables *liens père-fils IP* sont semblables aux tables *liens père-fils fonction* mais celles-ci définissent les liens hiérarchiques entre les IPs, sachant que pour une fonction neuromorphique donnée, il peut exister plusieurs réalisations (ou IPs) possibles.

base de données reseaneurones - Table ipneuron_links sur le serveur localhost

Structure Afficher SQL Sélectionner Insérer Exporter Opérations Vider Supprimer

reseaneurones (10)

Affichage des enregistrements 0 - 2 (3 total, traitement: 0.0006 sec.)

requête SQL : [Modifier] [Expliquer SQL] [Créer source PHP]
 SELECT *
 FROM 'ipneuron_links' LIMIT 0 , 30

Afficher : 30 ligne(s) à partir de l'enregistrement n° 0
 en mode horizontal et répéter les en-têtes à chaque groupe de 100

	ipneuron_name	na	k	leak	ca	5c	synapse1	synapse2	synapse3	synapse4	stim	buffer	pol
	ip1_fs_neuron	ip1_na	ip1_k	ip1_leak			ip1_synapse	ip1_synapse	ip1_synapse	ip1_synapse	ip1_stim	ip1_buffer	ip1_pol_neuron
	ip1_rs_neuron	ip1_na	ip1_k	ip1_leak	ip1_ca		ip1_synapse	ip1_synapse	ip1_synapse	ip1_synapse	ip1_stim	ip1_buffer	ip1_pol_neuron
	ip1_5c_neuron	ip1_na	ip1_k	ip1_leak	ip1_ca	ip1_ca	ip1_synapse	ip1_synapse	ip1_synapse	ip1_synapse	ip1_stim	ip1_buffer	ip1_pol_neuron

Afficher : 30 ligne(s) à partir de l'enregistrement n° 0
 en mode horizontal et répéter les en-têtes à chaque groupe de 100

[sérer un nouvel enregistrement](#)
[ersion imprimable](#)

[Fenêtre SQL](#)

Figure 5. Table *ipneuron_links*

II.4 Nombre d'IPs pour une fonction donnée ou blocks

La table *nombre d'IPs* donne le nombre d'IPs accessibles pour une fonction neuromorphique donnée.

The screenshot shows a web application interface with a sidebar on the left and a table in the main content area. The sidebar contains a navigation menu with the following items: Accueil, reseaneurones (10), and a list of sub-items under 'reseaneurones': blocks, current_level, function_level, io_final, ip, ipcurrent_links, ipneuron_links, ipsystem_links, lowerblocklinks, and neuron_level. The table in the main content area has the following structure:

	name_block	ip1	ip2	ip3	ip4	ip5
✂	block_system	ip1_block_system	NULL	NULL	NULL	NULL
✂	fs_neuron_final	ip1_fs_neuron_final	NULL	NULL	NULL	NULL
✂	rs_neuron_final	ip1_rs_neuron_final	NULL	NULL	NULL	NULL
✂	5c_neuron_final	ip1_5c_neuron_final	NULL	NULL	NULL	NULL
✂	fs_param	ip1_fs_param	NULL	NULL	NULL	NULL
✂	rs_param	ip1_rs_param	NULL	NULL	NULL	NULL
✂	5c_param	ip1_5c_param	NULL	NULL	NULL	NULL
✂	fs_neuron	ip1_fs_neuron	NULL	NULL	NULL	NULL
✂	rs_neuron	ip1_rs_neuron	NULL	NULL	NULL	NULL
✂	5c_neuron	ip1_5c_neuron	NULL	NULL	NULL	NULL
✂	num_topo_fs	ip1_num_topo_fs	NULL	NULL	NULL	NULL
✂	num_topo_rs	ip1_num_topo_rs	NULL	NULL	NULL	NULL
✂	num_topo_5c	ip1_num_topo_5c	NULL	NULL	NULL	NULL
✂	pol_neuron	ip1_pol_neuron	NULL	NULL	NULL	NULL
✂	na	ip1_na	NULL	NULL	NULL	NULL
✂	k	ip1_k	NULL	NULL	NULL	NULL
✂	ca	ip1_ca	NULL	NULL	NULL	NULL
✂	leak	ip1_leak	NULL	NULL	NULL	NULL

Figure 6. Table *blocks*

II.5 Valeur des spécifications ou *current_level*, *function_level*, *neuron_level*

Ces tables possèdent toutes les valeurs des spécifications (intervalles des domaines de validité) pour chaque IP. Elles sont très importantes car l'exploration de la base de données est conditionnée par ces valeurs. Trois tables sont définies, une pour chaque niveau hiérarchique (domaines de validité des fonctions mathématiques, domaines de validité des courants ioniques et domaines de validité des neurones).

Base de données *reseaneurones* - Table *function_level* 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(50)		Non			
<input type="checkbox"/> Voffset_act_inf	float	Oui	NULL			
<input type="checkbox"/> Voffset_act_sup	float	Oui	NULL			
<input type="checkbox"/> Voffset_inact_inf	float	Oui	NULL			
<input type="checkbox"/> Voffset_inact_sup	float	Oui	NULL			
<input type="checkbox"/> Vslope_act_inf	float	Oui	NULL			
<input type="checkbox"/> Vslope_act_sup	float	Oui	NULL			
<input type="checkbox"/> Vslope_inact_inf	float	Oui	NULL			
<input type="checkbox"/> Vslope_inact_sup	float	Oui	NULL			
<input type="checkbox"/> Vcoef_tau_inf	float	Oui	NULL			
<input type="checkbox"/> Vcoef_tau_sup	float	Oui	NULL			
<input type="checkbox"/> Vequi_inf	float	Oui	NULL			
<input type="checkbox"/> Vequi_sup	float	Oui	NULL			
<input type="checkbox"/> gmax_inf	float	Oui	NULL			
<input type="checkbox"/> gmax_sup	float	Oui	NULL			

↑ Tout cocher / Tout décocher Pour la sélection :

Figure 7. Table *function_level*

II.6 Connexions entre les différents Macro-blocs ou *io_final*

La table *connexions* contient les ports d'entrées/sorties de chaque *Macro-bloc*. La table *connexions* sert à la création de la netlist du système final trouvé par l'exploration.

Base de données *reseaneurones* - Table *io_final* sur le serveur *localhost*

Structure Afficher SQL Sélectionner Insérer

Affichage des enregistrements 0 - 2 (3 total, traitement: 0.0010 sec.)

requête SQL : [Modifier] [Expliquer SQL] [Créer source PHP]
 SELECT *
 FROM *io_final* LIMIT 0, 30

Afficher : 30 ligne(s) à partir de l'enregistrement n° 0
 en mode horizontal et répéter les en-têtes à chaque groupe de 100

	ip_name	I01	I02	I03	I04	I05	I06	I07	I08	I09	I010	I011	I012	I013	I014	I015	
		ip1_neuron_fs_final	param_a	Clk_a	Reset_a	Mem_en_a	seuil_num	data_in	EN	clk	prev	suiv	Vmem_buf	Cmem	Rmem	Vvclamp	Stim
		ip1_neuron_rs_final	param_a	Clk_a	Reset_a	Mem_en_a	seuil_num	data_in	EN	clk	prev	suiv	Vmem_buf	Cmem	Rmem	Vvclamp	Stim
		ip1_neuron_sc_final	param_a	Clk_a	Reset_a	Mem_en_a	seuil_num	data_in	EN	clk	prev	suiv	Vmem_buf	Cmem	Rmem	Vvclamp	Stim

Afficher : 30 ligne(s) à partir de l'enregistrement n° 0
 en mode horizontal et répéter les en-têtes à chaque groupe de 100

[Insérer un nouvel enregistrement](#)
[Version imprimable](#)

Exporter

Fenêtre SQL

Figure 8. Table *io_final*

III AJOUT D'IP

L'ajout de l'IP s'effectue aisément par le biais de l'interface de notre base de données. L'IP doit être ajouté aux différentes tables de la base de données. Le lien hiérarchique doit être défini, ses propriétés technologiques et son domaine de validité aussi. Le temps moyen pour effectuer un ajout d'IP est inférieur à 5 minutes.

Prenons l'exemple de l'ajout d'un IP d'une fonction sigmoïde d'activation.

Il faut tout d'abord l'insérer dans la table *Ip* et rentrer les informations suivantes :

- le nom de cet IP (*name*) et son niveau hiérarchique (*level*, pour une fonction mathématique c'est le niveau 1, pour les courants ioniques et respectivement les neurones, les niveaux sont 2 et respectivement 3)
- donner le chemin d'accès de toutes les vues (connectique, schéma, etc.) de cet IP
- la technologie et la tension d'alimentation
- Le nombre d'entrées et de sorties
- La surface du layout et son éventuel prix
- Renseigner si cet IP possède des spécifications précises à intégrer dans les tables des spécifications (0 pour non et 1 pour oui, comme cette fonction possède des spécifications, nous mettons 1)

Ensuite il faut ajouter dans la table *blocks*, le nom de cet IP dans la ligne sigmoïde d'activation (*sigmoid_act*).

Enfin, dans la table *function_level*, nous rentrons les spécifications de cet IP, à savoir ses intervalles de dynamique d'entrée.

Une autre possibilité existe concernant l'ajout d'IPs. Dans le cas d'un ajout d'un grand nombre d'IPs, nous n'effectuons pas la même démarche. Nous créons un fichier texte contenant les informations relatives à tous les IPs. Puis nous l'insérons dans la base de données. Ainsi toutes les données du fichier texte sont instantanément incluses dans la base.

Annexe 2 : Manuel d'utilisation de la bibliothèque d'IPs et de son exploration

The screenshot shows the phpMyAdmin interface for the 'reseaneurones' database. The 'Import' tab is active, showing options for loading data from a text file into the 'ip' table. The interface includes a sidebar with a tree view of the database structure, a top navigation bar with buttons like 'Structure', 'Afficher', 'SQL', etc., and a main form area with various options for importing data from a text file. The form includes fields for file location, replacement options, field delimiters, and column names, along with an 'Exécuter' button.

Figure 9. Ajout d'Ips à partir d'un fichier texte

Nous venons de décrire l'ajout d'un IP à la base de données, nous pouvons maintenant présenter l'exploration de la bibliothèque d'IPs.

IV PRESENTATION DE L'EXPLORATION

Après avoir décrit la base de données, nous présentons les différentes étapes de l'exploration et les fichiers créés. Tous les fichiers permettant l'exploration sont écrits en langages PHP, SQL et HTML.

A partir des spécifications du système, des requêtes sont créées. Ce sont celles-ci qui vont explorer la base de données pour déterminer et récupérer les IPs correspond au mieux à ce cahier des charges. Cette exploration s'effectue par une approche Top-Down en recherchant les IPs de plus haut niveau, du niveau *Macro-bloc* au niveau *Cellule*, et en utilisant les valeurs des spécifications de chaque IP comme critère de recherche. Si la recherche est négative au niveau hiérarchique le plus haut, on descend alors au niveau inférieur pour effectuer une nouvelle phase de recherche.

Dans le cas où une recherche aboutirait à plusieurs solutions, nous choisissons celle qui possède les spécifications les plus larges.

Pour faciliter la compréhension de cette démarche, nous illustrons cette approche par la figure suivante.

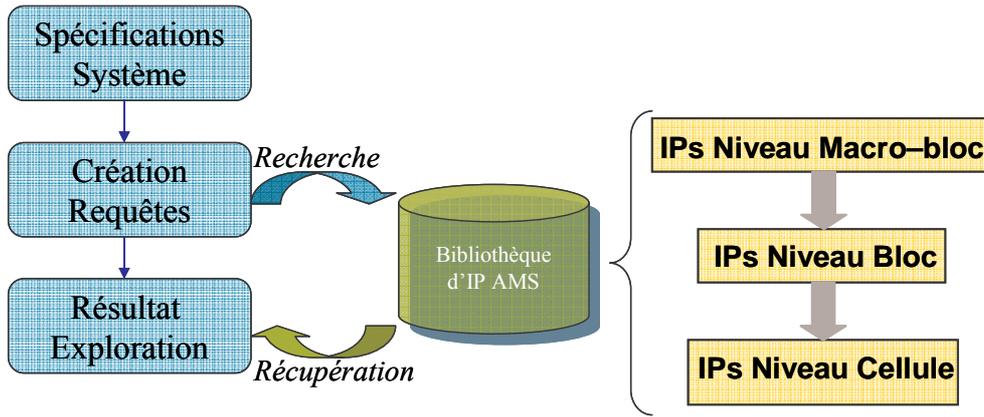


Figure 10. Exploration de la base de données

Le résultat de l'exploration est donné par un diagnostic évaluant le pourcentage de réutilisation.

Si le pourcentage est de 100 %, une netlist est créée rassemblant tous les *Macro-blocs* pour concevoir le système.

Sinon d'une part la liste de tous les IPs correspondants aux spécifications initiales est fournie, et d'autre part le(s) IP(s) manquant(s) sont identifiés et localisés dans la hiérarchie. Une aide à la conception de ces IPs manquants est alors donnée. Cette aide sera décrite dans un des paragraphes suivants.

Un récapitulatif est alors créé, il décrit tous les IPs choisis avec la possibilité de voir chacune de ces vues. Des données supplémentaires sont affichées concernant le circuit comme la tension d'alimentation, la technologie utilisée et même les valeurs des composants passifs à ajouter pour la conception du système final.

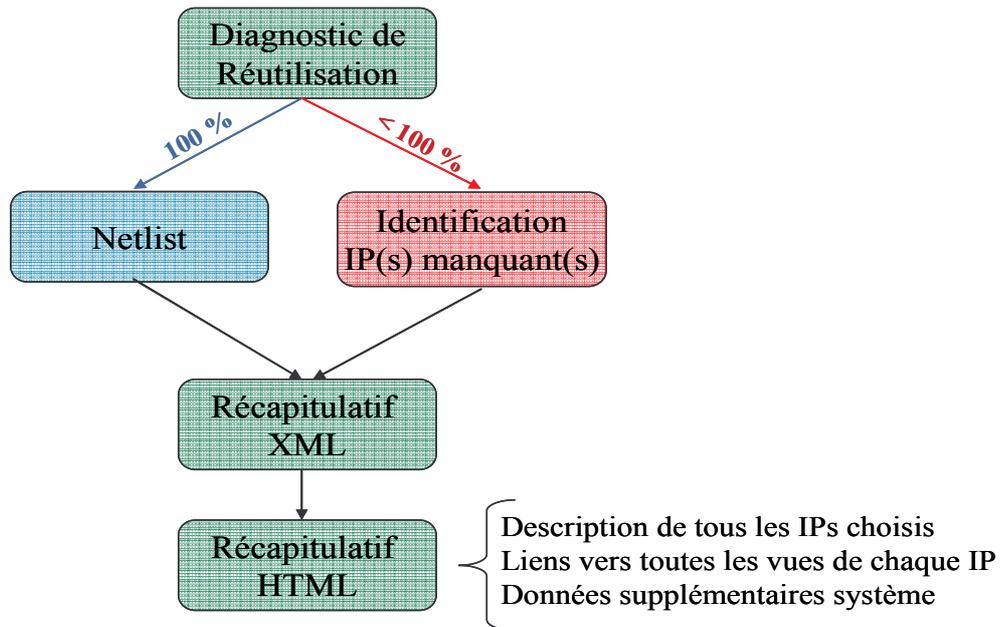


Figure 11. Illustration de la démarche de résultats

L'exploration de la base de données peut se décomposer en plusieurs étapes.

Ci-dessous est donné le diagramme UML de la conception d'un réseau de neurones à partir des spécifications d'un client. Toutes ces étapes vont être décrites dans les paragraphes suivants.

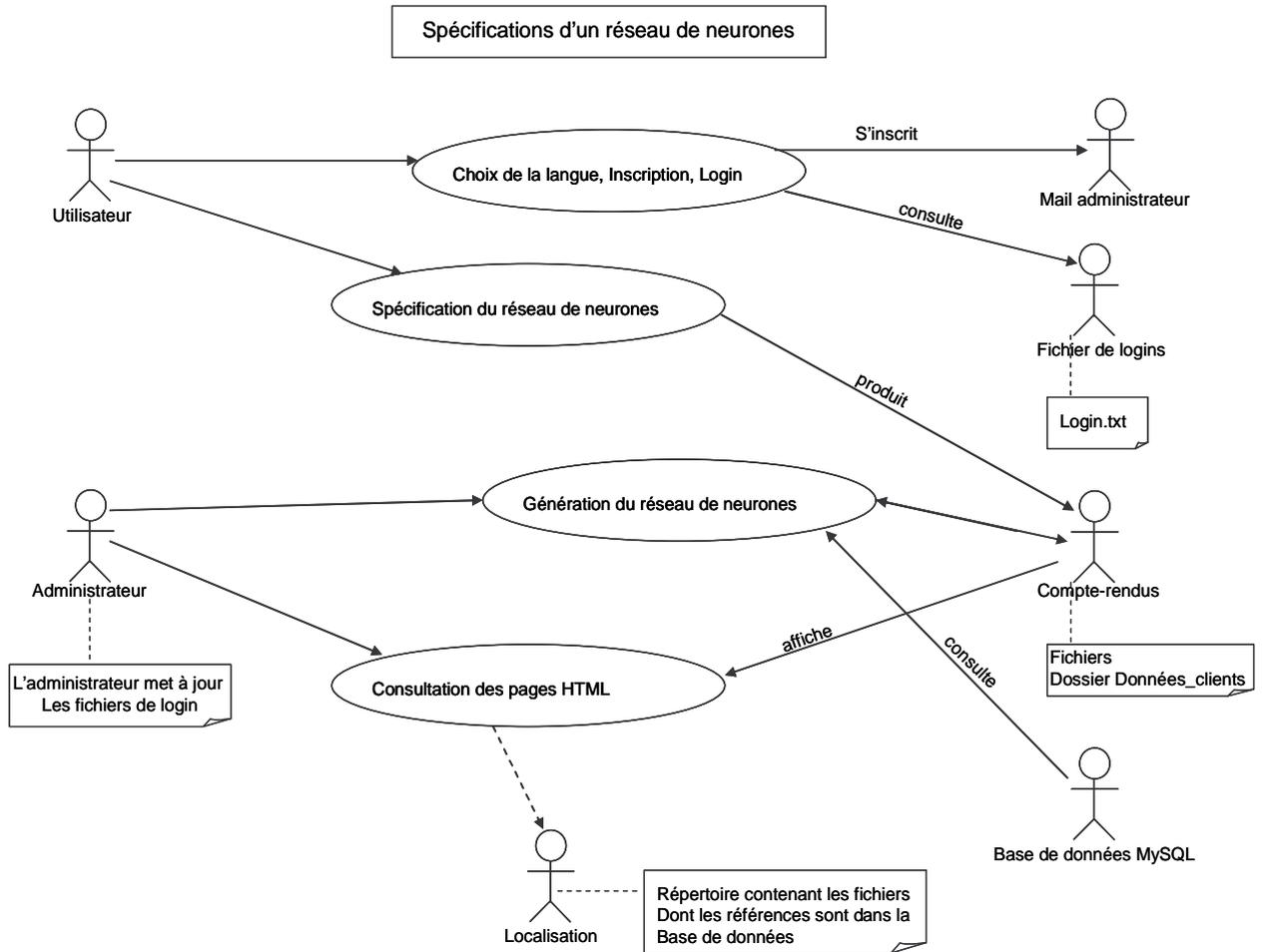


Figure 12. Diagramme UML de la conception du système neuromorphique

La figure 12 représente tous les acteurs, symbolisés par une forme humanoïde, qui interagissent avec le système. Ces acteurs sont des entités externes (physique ou logicielle)

Nous présentons tout d'abord l'entrée des spécifications par le client et les différentes options qu'il possède. Puis nous détaillerons la création des requêtes et leurs explorations. Enfin nous décrivons les récapitulatifs et les résultats.

IV.1 Entrée des spécifications par le client

Le client saisit dans le formulaire PHP, le cahier des charges de son système neuromorphique. Il peut choisir le nombre de neurones de son réseau mais aussi toutes les conductances pour chacun des neurones. Afin d'être le plus précis possible, tous les paramètres des conductances sont rentrés ainsi que la capacité de membrane et la surface du neurone, selon le formalisme d'Hodgkin-Huxley.

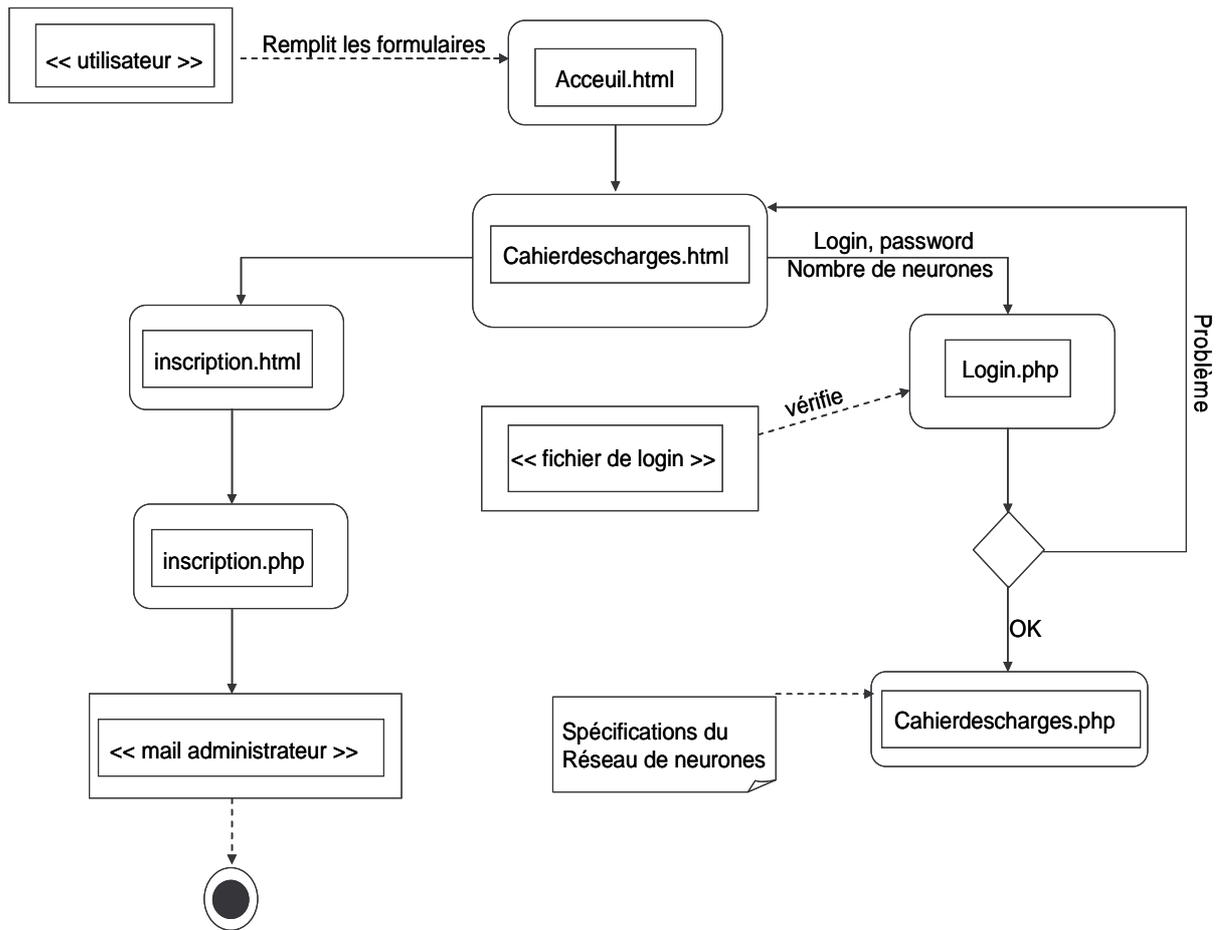


Figure 13. Diagramme UML des choix du client

La figure 3.18 décrit les différentes étapes que le client doit effectuer. Pour chaque étape, un fichier PHP est utilisé.

Pour proposer son cahier des charges au concepteur, le client doit préalablement s'être inscrit. Un mail est alors envoyé à l'administrateur pour demander l'accès à l'entrée des spécifications.

Une fois inscrit, le client peut rentrer son cahier des charges. Ci-après, une capture d'écran de l'entrée du choix des conductances pour chaque neurone.

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 14. Capture d'écran du formulaire d'entrée des spécifications

La figure suivante 3.18 décrit les différentes étapes de la saisie des spécifications avec pour sortie un fichier de cahier des charges qui résume toutes les spécifications et qui est indexé au nom du client et à l'heure de saisie.

Le client rentre les différents courants ioniques associés à chaque neurone (cahierdescharges.php), puis il rentre les valeurs des paramètres de ces courants (courant.php). Le compte-rendu est alors affiché (parametres.php) et ces spécifications sont enregistrées dans le répertoire donnees_clients.

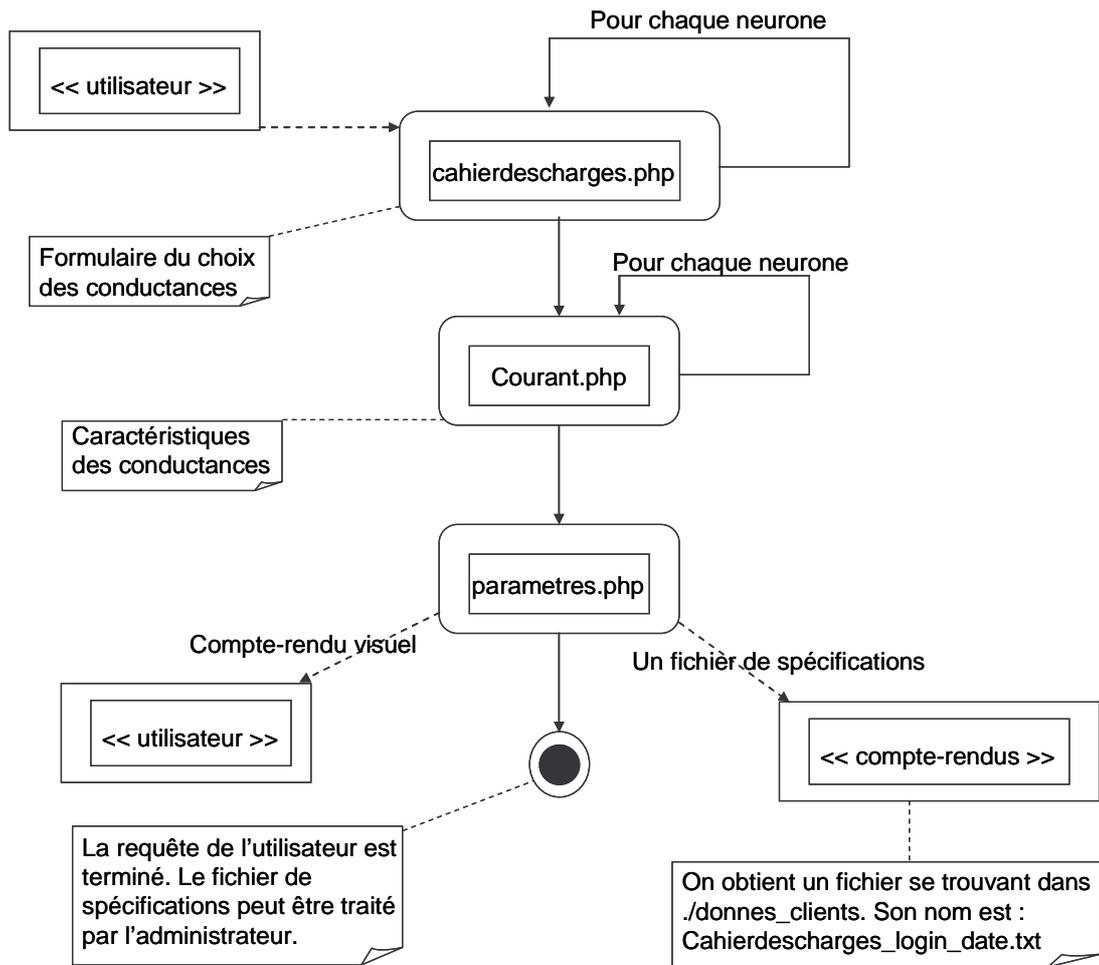


Figure 15. Diagramme UML de toutes les étapes du formulaire

IV.1.1 Choix du concepteur, création des requêtes et exploration de la base de données

Une fois que les spécifications ont été remplies par le client, le concepteur les reçoit. Il choisit alors les différents facteurs de conversion entre les valeurs biologiques et les valeurs électriques mais aussi la technologie utilisée et la tension d'alimentation voulue.

Les requêtes sont alors créées et l'exploration de la base de données commence.

La figure suivante représente les différentes étapes entre les spécifications du client et la création des requêtes.

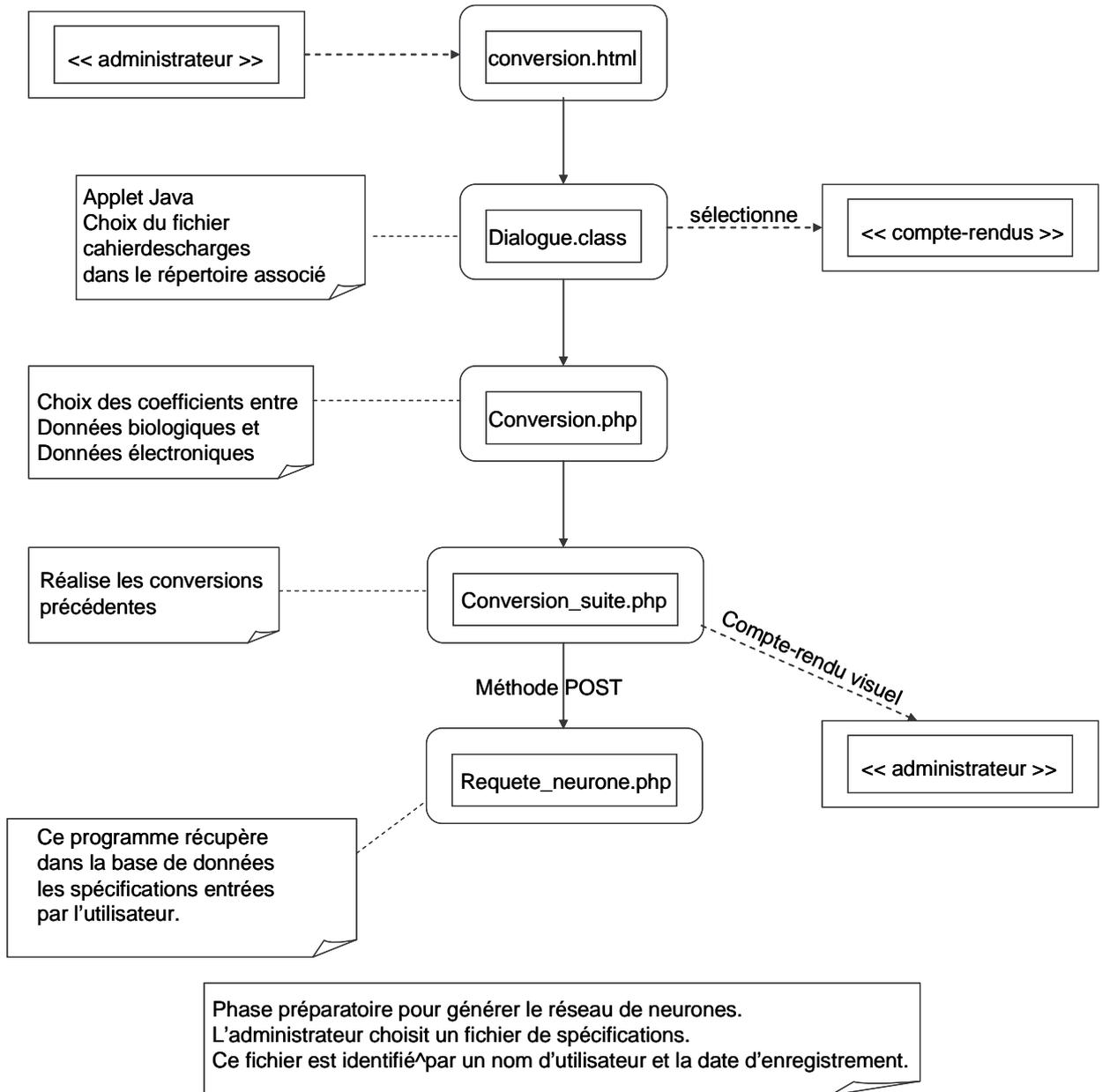


Figure 16. Diagramme UML de la création des requêtes

L'exploration s'effectue par une méthodologie Top-Down (pour notre cas d'étude, du neurone aux fonctions mathématiques en passant par les courants ioniques). Le critère le plus important dans le choix des IPs est le domaine de validité. Ainsi tous les IPs sont triés et sélectionnés par leur technologie, leur tension d'alimentation, puis leur domaine de validité.

IV.1.2 Récapitulatif et résultats

Quand l'exploration de la base de données est terminée, un récapitulatif de celle-ci est donné. Ceci permet au concepteur de visualiser les résultats. Des fichiers sont créés pour résumer les différents résultats et s'il existe un échec lors de l'exploration, il est alors signalé et identifié.

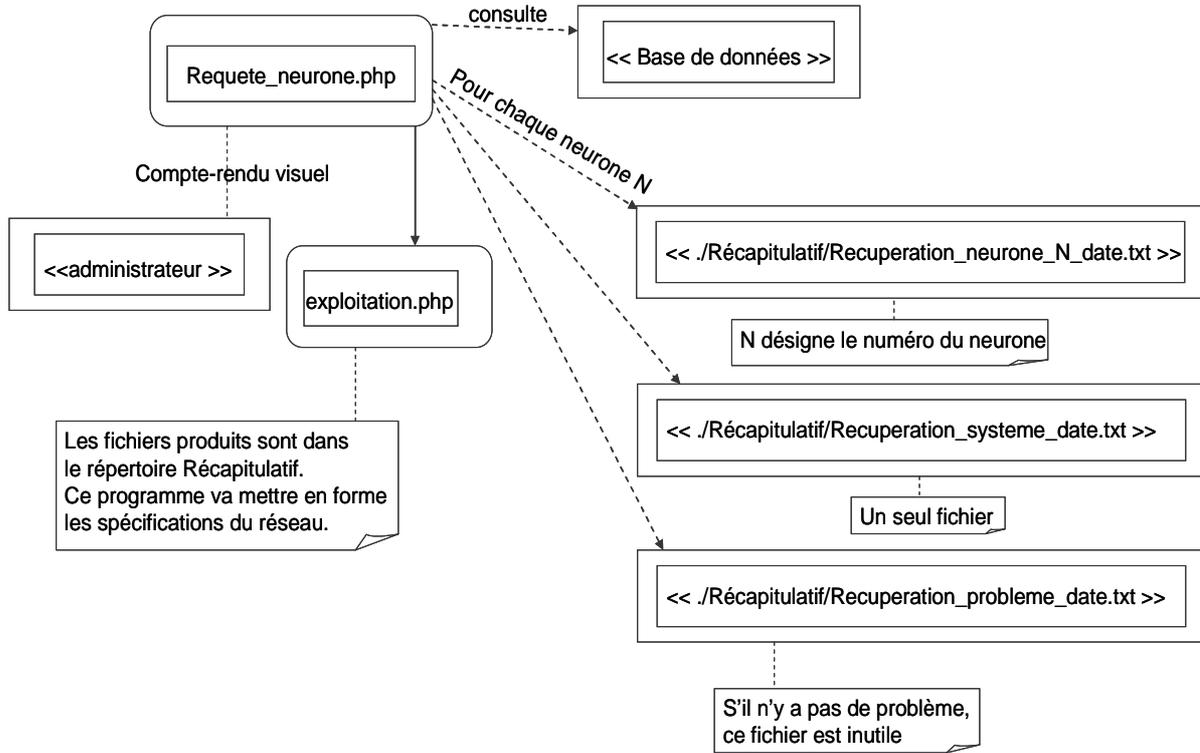


Figure 17. Diagramme UML des résultats

Pour chaque neurone, un fichier donnant les IPs sélectionnés pour le concevoir est engendré de même qu'un autre comportant les IPs de base pour concevoir un système. Un fichier précisant les échecs de l'exploration (s'ils existent) est créé.

Suivant le résultat de l'exploration, deux orientations existent. Si le diagnostic de réutilisation est de 100 % (figure 11), une netlist est alors créée contenant l'architecture complète du système ainsi obtenu. Sinon une identification précise du ou des échecs est donnée.

Une dernière étape existe, ce sont les comptes-rendus des résultats avec tous les liens des différentes vues des IPs sélectionnés pour concevoir le système. Ceux-ci sont créés via un fichier XML qui est ensuite transformé en fichier HTML pour une meilleure visualisation. Le concepteur peut ainsi connaître les IPs sélectionnés et avoir accès aux différentes vues de ces IPs.

Annexe 2 : Manuel d'utilisation de la bibliothèque d'IPs et de son exploration

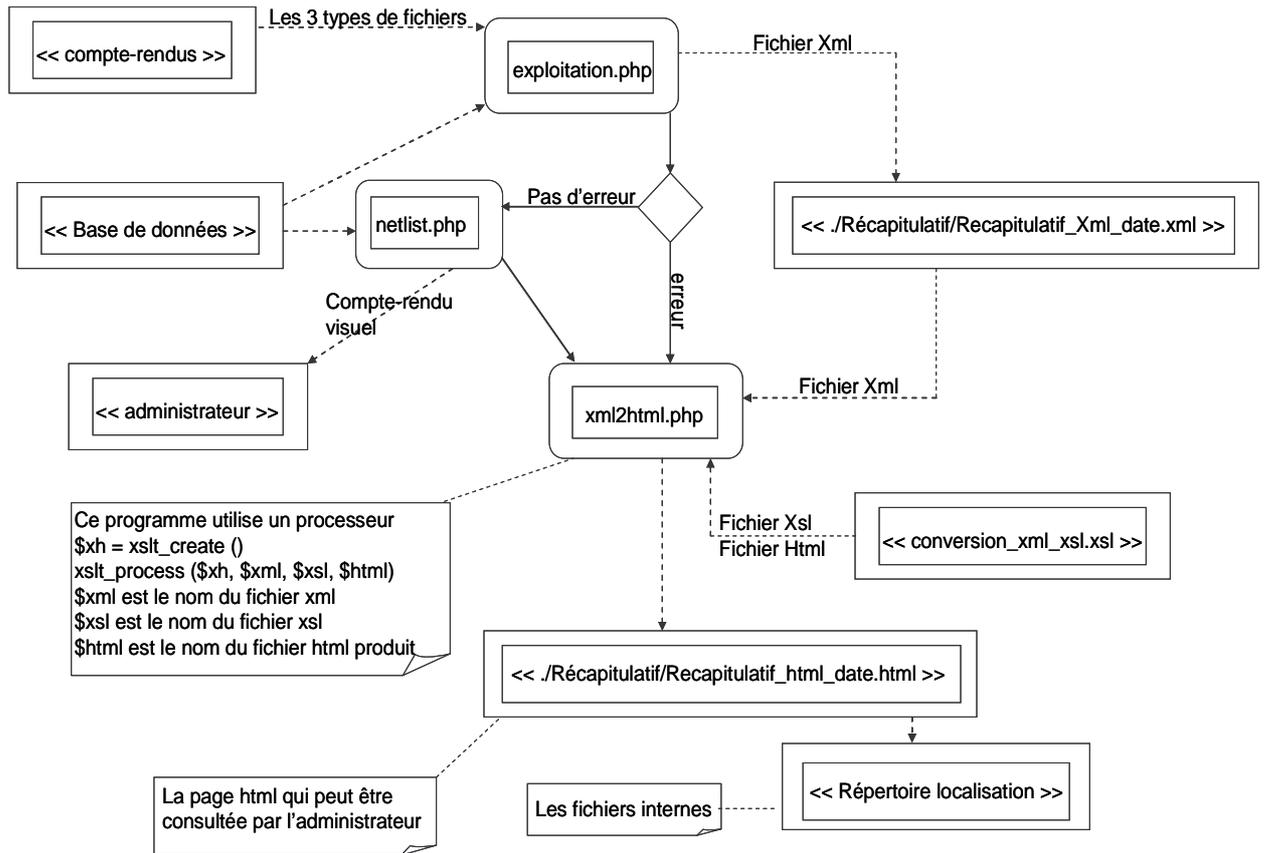


Figure 18. Diagramme UML de la création des fichiers XML et HTML

La figure 18 présente les différentes étapes concernant l'exploitation des résultats de l'exploration. Un processeur transformant les fichiers XML et XSL en extension HTML est utilisé. Ainsi à partir de tous les fichiers créés lors de l'exploration, un fichier HTML, récapitulant toutes ces informations, est développé et permet au concepteur de visualiser le système à concevoir.

Dans le dernier fichier récapitulatif, tous les IPs choisis sont donnés mais aussi les différentes capacités externes à implémenter et la surface totale du système. Une vérification est alors nécessaire pour confirmer ce résultat en utilisant les vues comportementales de ces IPs.

IV.2 Description des fichiers utilisés

Nous décrivons ici tous les fichiers utilisés.

- **Acceuil.html** : ce fichier permet de choisir la langue (anglais, français) pour entrer les spécifications
- **Inscription.php** : ce fichier permet d'envoyer un email à l'administrateur avec ses informations pour pouvoir s'inscrire et ainsi avoir le droit de compléter le formulaire des spécifications
- **Login.php** et **login.txt** : le fichier **login.php** effectue un test de reconnaissance pour autoriser ou non la poursuite du formulaire. Tous les logins autorisés sont dans le fichier **login.txt**
- **Cahierdescharges.html** : ce fichier permet de rentrer le nombre de neurones voulus ainsi que le nom et le mot de passe du client
- **Cahierdescharges.php** : le client rentre ici le nombre et le type de courant ionique pour chaque neurone ainsi que la capacité et la surface du neurone
- **Courant.php** : le client remplit tous les paramètres des courants ioniques
- **Parametres.php** : ce fichier permet de donner un récapitulatif des spécifications pour le client, il crée aussi un fichier texte comportant toutes les spécifications. Ce fichier teste est indexé par le nom du client et la date.
- **Conversion.html** : ce fichier permet au concepteur de choisir le fichier texte de spécifications
- **Conversion.php** : le concepteur choisit ici les facteurs de conversion pour passer des spécifications biologiques aux spécifications électriques
- **Conversion_suite.php** : ce fichier donne toutes les spécifications électriques en utilisant les facteurs de conversion entrés précédemment
- **Requete_neurone.php** : ce fichier effectue l'exploration de la base de données à partir des spécifications électriques. Il donne un récapitulatif de l'exploration avec des tableaux de diagnostics. Il crée plusieurs fichiers indexés décrivant les résultats, donnant les IPs choisis et s'ils existent les échecs de l'exploration.

Annexe 2 : Manuel d'utilisation de la bibliothèque d'IPs et de son exploration

- `Exploitation.php` : ce fichier permet de créer un fichier xml récapitulant tous les résultats de l'exploration et en déterminant les IPs choisis pour répondre aux spécifications.
- `Verifie_dtd` et `erreur.txt` : le fichier `verifie_dtd` permet de vérifier le format du fichier xml créé. S'il existe une erreur, elle est alors donnée dans le fichier `erreur.txt`
- `Netlist.php` : ce fichier crée les connections entre les IPs systèmes si l'exploration est validée complètement.
- `Conversion_xml_xsl.xsl` et `Xml2html.php` : ces fichiers créent à partir du fichier xml le fichier html final permettant un récapitulatif complet et des liens avec les IPs choisis.

IV.3 Description des langages utilisés

Les outils utilisés pour la création, l'organisation et l'exploration de la bibliothèque d'IPs sont easyPhP, MySQL, HTML et XML.

Les langages PhP et HTML serviront pour la création et la gestion d'un formulaire de saisie des spécifications. Le langage SQL est utilisé pour la création de la base de donnée et des requêtes. Tandis que le langage XML sert d'intermédiaire entre le résultat des requêtes et la vue finale sous HTML.

IV.3.1 Langage PhP

PhP est un langage interprété (un langage de script) qui est exécuté du côté serveur (comme les scripts CGI, ASP, ...) et non du côté client (comme un script écrit en Javascript ou une applet Java qui s'exécute sur l'ordinateur...). La syntaxe du langage provient de celles du langage C, du Perl et de Java. Ses principaux atouts sont :

- Une grande communauté de développeurs partageant des centaines de milliers d'exemples de script PHP ;
- La gratuité et la disponibilité du code source (PHP est distribué sous licence GNU GPL) ;
- La simplicité d'écriture de scripts ;
- La possibilité d'inclure le script PHP au sein d'une page HTML (contrairement aux scripts CGI, pour lesquels il faut écrire des lignes de code afin d'afficher chaque ligne en langage HTML) ;

- La simplicité d'interfaçage avec des bases de données (de nombreux Système de Gestion de Base de Données SGBD sont supportés, mais le plus utilisé avec ce langage est MySQL, un SGBD gratuit disponible sur de nombreuses plateformes : Unix, Linux, Windows, MacOS X, Solaris, etc...) ;
- L'intégration au sein de nombreux serveurs web (Apache, Microsoft IIS, etc.).

IV.3.2 Langage SQL

SQL (*Structured Query Language*, soit Langage de requêtes structuré) est à la fois un langage de définition de données (LDD, ou en anglais DDL Data Definition Language), un langage de manipulation de données (LMD, ou en anglais DML, Data Manipulation Language), et un langage de contrôle de données (LCD, ou en anglais DCL, Data Control Language), pour les bases de données relationnelles.

SQL s'avère être un langage de définition de données (LDD) dans la mesure où il permet la création de tables dans une base de données relationnelles, ainsi que de les modifier ou de les supprimer.

SQL est un langage de manipulation de données (LMD), c'est à dire qu'il permet de sélectionner, insérer, modifier ou supprimer des données dans une table d'une base de données relationnelle.

SQL est un langage de protection d'accès. Il est possible avec SQL de définir des permissions au niveau des utilisateurs d'une base de données. On parle de DCL (Data Control Language).

IV.3.3 Langage XML

XML (*Extensible Markup Language* ou langage de balisage extensible) est un standard du World Wide Web Consortium qui sert de base pour créer des langages balisés spécialisés; il s'agit d'un « méta langage ». Il est suffisamment général pour que les langages basés sur XML, appelés aussi dialectes XML, puissent être utilisés pour décrire toute sorte de données et de textes. Il s'agit donc partiellement d'un format de données.

Nous utiliserons XML et son fichier de style associé XSL, pour une meilleure organisation et présentation des résultats de l'exploration de la base de données.

V CONCLUSION

Ce manuel a décrit la conception de la base de données avec la description de toutes les tables conçues. L'ajout d'un élément de cette bibliothèque a été présenté. Les différentes étapes de l'exploration et la présentation de tous les fichiers PHP et HTML ont été effectuées.

ANNEXE 3

MODELE FONCTIONNEL DE LA CINETIQUE A 2 CONSTANTES DE TEMPS

Annexe 3 : Modèle fonctionnel de la cinétique à 2 constantes de temps

```
analog begin

    @(initial_step ) begin
        I(Capa1) <+ I(Capa2);
        V(Capa1) <+ I(Iin)/g_ota+2.5;
        I(Capa2) <+ 0;
        V(Capa2) <+ I(Iin)/g_ota+2.5;
        V(Ih_10u) <+ 2.5;
        I(Ih_10u) <+ 10u;
    end;

    // Fonction de transfert
    V(Iin) <+ I(Iin)*Rin+2.5;
    V(Ih_10u) <+ I(Ih_10u)*Rin+2.5;

    Iconv = (5-V(Vcoef))/Rconv;
    Imult1 = I(Iin)*Icoef/(I(Ih_10u)+1f);
    Imult2 = I(Idiff)*Icoef/(I(Ih_10u)+1f);

    if(V(Vmem) > 2.5 ) begin
        I(Capa1) <+ -Imult12-Imult22;
        I(Idiff) <+ -(V(Capa1)-2.5)*g_ota;
    end else begin
        I(Capa2) <+ -Imult12-Imult22;
        I(Idiff) <+ -(V(Capa2)-2.5)*g_ota;
    end

end

endmodule
```

Résumé

Les travaux de cette thèse apportent une contribution à l'automatisation du flot de conception analogique et mixte, en termes de méthodologies de réutilisation. Des méthodologies de développement et d'exploration de bibliothèques d'IPs (*Intellectual Property*) analogiques sont développées : définition et caractérisation d'un IP analogique, création et exploration d'une base de données d'IPs, aide à la réutilisation destinée au concepteur. Le circuit utilisé pour l'application de ces méthodologies est un système neuromimétique c'est-à-dire qu'il reproduit l'activité électrique de neurones biologiques. Ces applications montrent à travers trois exemples, l'efficacité et la souplesse de notre méthodologie. Ces travaux proposent également une méthodologie de redimensionnement de circuits analogiques CMOS lors d'une migration technologique.

Mots-clés

Flot de conception analogique et mixte, méthodologies de réutilisation, IP-AMS, bibliothèque d'IPs, ingénierie neuromorphique, circuits intégrés mixtes, modélisation comportementale, redimensionnement, circuits intégrés neuromimétiques, neurones artificiels.

Title

Development methodology of an IP-AMS library for automatic analog and mixed system design: applied to neuromorphic engineering

Abstract

The main objective of this dissertation is the improvement of the analogue and mixed design flow, using reuse methodologies. Development and exploration methodologies for the IP-based library (*Intellectual Property*) are developed: definition and characterization of an analogue IP, creation and exploration of the IP-based library, reuse help for the designer. The demonstrator is a neuromimetic integrated circuit which fits the electrical activity of biological neurons. Those applications describe, with three examples, the efficiency and the flexibility of this methodology. A resizing methodology of CMOS analogue designs during a technological migration is also developed in this work.

Key words

Analogue and mixed design flow, reuse methodologies, IP-AMS, IP-based library, neuromorphic engineering, mixed integrated circuits, behavioural modelling, resizing, neuromimetic integrated circuits, artificial neurons.