HAL
open science

Contribution à la réalisation électronique de réseaux de neurones formels : intégration analogique d'une Machine De Boltzmann

Eric Belhaire

▶ **To cite this version:**

Eric Belhaire. Contribution à la réalisation électronique de réseaux de neurones formels : intégration analogique d'une Machine De Boltzmann. domain_stic.othe. Université Paris Sud - Paris XI, 1992. Français. NNT : . tel-00008989

HAL Id: tel-00008989

https://theses.hal.science/tel-00008989v1

Submitted on 11 Apr 2005

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.

ORSAY
n° d'ordre :

UNIVERSITE DE PARIS-SUD

CENTRE D'ORSAY

THESE

Présentée

Pour obtenir

Le grade de Docteur en Science

de L'Université Paris XI Orsay

Par

Eric BELHAIRE

SUJET : Contribution à la réalisation électronique de Réseaux de Neurones Formels : Intégration Analogique d'une MACHINE DE BOLTZMANN.

Soutenue le 6 Février 1992 devant la Commission d'examen

MM.	E. VITTOZ	Président
	R. AZENCOTT	
	F. DEVOS	
	P. GARDA	
	P. PERETTO	Rapporteur
	P. SENN	Rapporteur

Les travaux présentés ici ont été effectués à l'Institut d'Electronique Fondamentale de l'Université de Paris Sud, Orsay. Ils se sont déroulés au sein de l'opération "Machine Neuronale pour la Vision" du département Architecture et Conception des Circuits Intégrés et Systèmes (AXIS) de ce laboratoire. Ils ont bénéficié du support de la DRET (convention de recherche n° 87/292) du GCIS (opération "Machine de Boltzmann" et du PRC-ANM (projet RA).

Je tiens à remercier Madame Suzanne LAVAL de m'avoir accueilli dans son laboratoire. J'aimerais remercier Patrick GARDA, qui a assuré la responsabilité scientifique de ce travail de thèse. Il m'a guidé quotidiennement et j'ai très souvent profité de ses idées et de ses discussions. Je le remercie aussi pour sa très grande disponibilité et sa clairvoyance.

Je remercie Francis DEVOS, mon directeur de thèse, pour m'avoir accueilli dans son département de recherche.

J'aimerai exprimer ma gratitude aux membres du jury ; je suis honoré qu'ils aient accepté d'en faire partie. Je suis reconnaissant à Monsieur le Professeur Eric VITTOZ d'avoir accepté la présidence de mon jury. Ses nombreuses publications sont pour moi une grande source de connaissance. Je remercie Monsieur le Professeur Pierre PERETTO et Monsieur le Professeur Patrice SENN d'avoir accepté la "lourde charge" de d'être rapporteur de mon travail. Je remercie Monsieur le Professeur Robert AZENCOTT pour les nombreuses discussions qui ont largement contribué à la définition de ce travail.

Quotidiennement, le support d'une équipe est déterminant et je tiens à remercier tous les autres membres du département AXIS (ex GME) que j'ai eu plaisir à côtoyer pendant ces années. Je remercie tout particulièrement ceux qui m'ont apporté une aide précieuse : Claude, Christophe, Hubert, Kurosh, Vincent, Yiming…

Vous trouverez dans ce manuscrit des travaux nés d'une collaboration des plus agréables avec l'Institut d'Optique Théorique et Appliquée et je remercie tout particulièrement Jean-Claude RODIER et Philippe LALANNE avec lesquels j'ai plaisir à travailler.

Je dois surtout beaucoup à ma famille. J'ai longtemps cherché comment je pourrais remercier Maryanne et mes parents en quelques lignes, et je cherche encore ! Qu'ils considère ce travail comme un peu le leur. Ce manuscrit leur est dédié.

INTRODUCTION

RESEAUX DE NEURONES FORMELS ET TRAITEMENT DE L'INFORMATION ANALOGIQUE.

CIRCUITS NUMERIQUES ET CIRCUITS ANALOGIQUES EN 1991.

Depuis l'apparition des technologies MOS (Metal Oxide Semiconductor) modernes, on utilise de plus en plus les techniques numériques plutôt que les techniques analogiques pour réaliser un algorithme sous forme de circuit intégré. Pourquoi les technologies VLSI (Very Large Scale Integration) imposent-elles le recours massif aux techniques numériques ? On peut trouver plusieurs raisons à cet état de fait :

- La première raison est d'ordre technique.

Dans les systèmes de calcul analogique, on représente l'information par une grandeur physique (tension, courant ou charge) et on utilise les propriétés physiques des transistors pour implanter les fonctions à réaliser. Cependant, la caractéristique de transfert d'un transistor est non linéaire, mal définie, et l'information est noyée dans du bruit. Ces systèmes ont alors une précision limitée. Ils servent à la réalisation de fonctions relativement simples parce qu'on ne peut pas avoir des chaînes de traitements longues et complexes qui fonctionnent correctement [Tsi87]. Les chaînes de traitement analogiques ont en plus le défaut d'être difficilement programmables et d'être tout au plus paramétrables. On peut, toutefois, espérer distinguer 256 valeurs de tension différentes entre les deux alimentations d'un circuit.

Dans les systèmes numériques, par contre, on décide de coder l'information, non plus par des valeurs de tension, mais par des intervalles de tension. Ces intervalles sont, de plus, très éloignés l'un de l'autre ("1"∅tension proche de Vdd, "0"∅tension proche de la masse). Les unités de traitement sont alors des éléments fortement non-linéaires, les portes logiques, qui sont chargés de discriminer les différents intervalles de codage et d'assurer le transfert de l'information de l'un à

l'autre. Elles sont conçues de manière à respecter différentes contraintes (marge de bruit, sortance,…), afin d'assurer que le signal ne subisse pas de dégradation et soit même régénéré par les unités de traitement.

Ainsi, dans les systèmes analogiques, le signal se dégrade au fur et à mesure qu'on cherche à le traiter. Alors qu'en se limitant à un bit d'information sur la gamme de tension disponible, la régénération du signal par les unités de traitement permet d'obtenir un nombre total d'unités et des distances parcourues par le signal virtuellement illimités.

- La deuxième raison est d'ordre pratique.

Elle est une conséquence directe de la première. Les technologies VLSI permettent de réaliser sur un même substrat en Silicium énormément de transistors MOS (plus d'un million), et les techniques numériques apportent une solution relativement simple pour exploiter la formidable puissance de traitement ainsi disponible. L'utilisation d'une information symbolique (binaire) a permis de développer des simulateurs indépendant de la technologie utilisée et des logiciels de CAO (Conception Assistée par Ordinateur) très complets. Outre la vérification des contraintes déjà énoncées pour les portes logiques telle que la sortance, ces logiciels peuvent à partir d'une bibliothèque de portes ou de cellules élémentaires générer directement le dessin des masques de fabrication et tout cela avec une intervention humaine très limitée. Le travail du concepteur d'ASIC consiste alors "simplement" à décrire son circuit sous la forme d'une schématique ou d'une description fonctionnelle en respectant des régles, là encore, clairement énoncées [Nai90]. Les circuits numériques ont de plus l'avantage d'être programmables et la mise au point d'un système en est simplifiée.

En conclusion, les techniques numériques permettent de réaliser, avec une grande sûreté de conception, des systèmes programmables complexes et d'une grande souplesse d'utilisation.

A ce jour, on peut schématiquement dire que les techniques analogiques sont surtout utilisés pour remplir les taches d'acquisition et de traitement du signal alors que les techniques numériques le sont plutôt pour remplir les taches de traitement de l'information [Dev91]. Dans une chaîne de traitement classique, nous trouverons donc en suivant le parcours d'un signal : tout d'abord des capteurs, des transducteurs, des unités analogiques de pré-traitement et d'interface avec les circuits numériques (amplification, filtrage, conversion analogique-numérique...),

puis les circuits de traitement numérique de l'information, et enfin si cela est nécessaire un convertisseur numérique-analogique [Tsi85], [Can86], [Tsi87], [Dev91].

La frontière entre le domaine du traitement du signal et celui du traitement de l'information n'est pas clairement définie. On peut donc parfois avoir le choix entre les deux techniques d'implantation, et l'utilisation des filtres à capacités commutées ou des filtres à temps continu est souvent opposée à celle des filtres numériques. Chaque technique a ses avantages, ses inconvénients, ses champs d'application, mais aussi ses partisans et ses détracteurs, ...

Il faut toutefois signaler que les circuits analogiques sont encore largement utilisés dans des domaines spécifiques où les technologies actuelles n'ont pas encore permis leur remplacement par des circuits numériques comme par exemple pour les signaux à très haute fréquence[1], mais ils comportent alors assez peu de transistors ou d'éléments d'amplification.

Cette situation peut paraître irrémédiable, et on peut penser que les circuits analogiques ne seront bientôt plus utilisés que pour faire des interfaces avec les circuits numérique. Mais…

LES ARCHITECTURES NEURONALES

Cependant, il y a eu ces dernières années une résurgence d'intérêt pour les réseaux de neurones formels (RNF) et ce concept nous permet d'entrevoir un bouleversement dans les architectures des systèmes de traitement de l'information. Les RNF apportent, en effet, une alternative aux architectures de type Von-Neumann [Darpa88] et certains affirment qu'ils seront utilisés pour la résolution des problèmes que l'intelligence artificielle (IA) classique ne sait pas résoudre. Il faut bien dire que d'autres jugent qu'il est naïf de croire cela, et que l'on met trop d'espoir dans les réseaux de neurones formels.

De toute manière, même si ces systèmes n'ont pas toutes les capacités et toute la puissance qu'on leur accorde aujourd'hui, il est clair qu'ils ont des propriétés qui méritent d'être explorées par une réalisation physique. En effet, les systèmes biologiques savent résoudre aisément des problèmes complexes que les ordinateurs conventionnels ont beaucoup de mal à traiter. Ainsi, un oiseau a des capacités de traitement d'image très supérieures aux ordinateurs modernes les plus puissants.

---

[1]Pour les signaux à très large bande et à très haute fréquence, les circuits analogiques sont utilisés pour la transmission : modulation/démodulation, multiplexage, amplification.

On peut dire que les RNF se distinguent des autres formes de calcul par deux caractéristiques principales :

• Ils sont adaptatifs : ils ne sont pas programmés, mais ils sont entrainés à partir de données. Ainsi beaucoup pensent que cela soulagera le programmeur d'une charge de travail importante. De plus, les RNF sont réputés s'améliorer avec l'expérience ; plus ils ont appris de données, plus leur réponse est exacte [Darpa88].

• Les réseaux de neurones sont naturellement massivement parallèles. Ceci suggère qu'ils sauront prendre une décision en un temps très court et avec une certaine tolérance aux fautes.

Dans les RNF, la puissance de traitement n'est plus obtenue en enchainant quelques unités complexes traitant le signal séquentiellement, mais en interconnectant fortement un très grand nombre d'unités très simples et fonctionnant en parallèle. Le résultat du traitement émane alors d'un comportement collectif du réseau plutôt que d'un comportement individuel des unités de calcul.

Dans les RNF, les chemins de calcul sont très court et les neurones peuvent être vus comme des éléments de décision régénérant en permanence le signal. De plus, l'apprentissage des RNF pourrait apporter une solution efficace au problème que pose la compensation des imperfections de l'analogique.

La question se pose alors de savoir si les Réseaux de Neurones Formels permettront de profiter, dans la réalisation des architectures de traitement de l'information, à la fois de la puissance de calcul inhérente aux systèmes analogiques et de la grande capacité d'intégration des technologies VLSI.

Le but de cette thèse est de tenter d'apporter un élément de réponse à cette vaste question en étudiant la réalisation analogique d'un RNF particulièrement intéressant : La Machine de Boltzmann.

Dans le chapitre I, je présenterai successivement le concept des réseaux de neurones et je parlerai plus précisément de l'algorithme de la Machine de Boltzmann.

Dans le chapitre II, je ferai l'état de l'art des réalisations de Machine de Boltzmann, ainsi que des techniques analogiques utilisées pour les réalisations de RNF.

Dans le chapitre III, je présenterai l'architecture de la machine, telle que nous l'avons définie.

Dans le chapitre IV, je présenterai ensuite la conception et la simulation des cellules et des circuits que j'ai réalisés.

Enfin, le chapitre V sera consacré à la description des mesures que j'ai pu effectuer sur ces circuits.

CHAPITRE I

PRESENTATION DU PROBLEME

I.1- RESEAUX DE NEURONES FORMELS

I.1.1- Qu'est qu'un Réseau de Neurones Formels ?

a- Historique

Au début des années 40, le concept des Réseaux de Neurones Formels (RNF) est né des premières recherches de construction de machines dont le comportement soit semblable au cerveau. L'idée originale de ce concept était que de telles machines doivent être construites en interconnectant des éléments de base étant des abstractions simples du comportement des cellules nerveuses. La première discussion à ce sujet est à l'actif de McCulloch et Pitts en 1943 [McC43] dans une publication traitant de réseaux "neuro-logiques". Parmi les publications importantes de cette époque, il faut aussi citer l'ouvrage de Hebb, "The organization of the Behavior" en 1949 [Heb49], qui suivait une perspective d'élaboration d'une théorie de la psychologie. Des travaux de cette époque on peut retenir, tout d'abord, un modèle de neurone appelé le neurone de McCulloch et Pitts, mais aussi la loi de Hebb qui énonce l'idée qu'un neurone peut apprendre à supporter l'activité des autres neurones de son voisinage.

Dans les années 50, l'ère de la cybernétique s'ouvrit par la prospection de machines qui, avec un fonctionnement interne semblable à celui d'un cerveau, sont capables de remplir certaines tâches fixées. Ensuite, les gens ont rapidement recherché des machines susceptibles d'apprendre. L'apprentissage consistait essentiellement à renforcer les actions qui se sont déroulées avec succès. Le concept

de programmation a vu le jour à cette période. Ce nouveau concept a permis d'importants progrès en intelligence artificielle notamment grâce à l'utilisation de modèles basés sur un calcul en série d'expressions symboliques au début des années 60. Des batailles intellectuelles commencèrent alors à voir le jour entre calcul parallèle et calcul série, apprentissage et programmation, émergence et description analytique de la connaissance.

Il n'y eu plus alors de fait marquant dans le domaine des réseaux de neurones jusqu'aux années 60 où Rosenblatt développa le Perceptron [Ros62]. Ce modèle fut le premier RNF réellement opérationnel et il y eu dans la communauté scientifique un grand engouement pour lui. On a vu aussi au cours de ces années la réalisation de la machine Adaline ("ADAptative LINear Element") [Wid60] où des poids étaient implantés avec des résistances. C'est de ces travaux qu'est né l'algorithme d'apprentissage LMS (Least Mean Square) [Coh85] qui est largement utilisé pour le traitement du signal adaptatif. A cette époque, de nombreux chercheurs travaillaient sur l'apprentissage du Perceptron.

Cependant en 1969, Minsky et Papert ont publié leur célèbre livre "Perceptron". Celui-ci a montré que ce modèle n'avait pas d'aptitude pour apprendre tous les types de connaissances mais seulement des problèmes linéairement séparables [Min69], [Min88]. L'apprentissage de problèmes d'ordre supérieur à 1 (le "ou-exclusif" est par exemple un problème d'ordre 2) nécessitait le recours à des réseaux de topologie différente (perceptron multicouche), mais, les algorithmes d'apprentissage utilisés ne fonctionnaient plus correctement dans ce cas. Le nombre de chercheur travaillant sur le sujet a alors largement diminué jusqu'aux années 80.

Le récent regain d'intérêt pour les réseaux de neurones est principalement dû au développement de nouveaux algorithmes d'apprentissage.

Tout d'abord, Hopfield a publié son modèle en 1982 [Hop82], et il l'a étudié par analogie avec des modèles physiques de particules en interaction (les verres de spin). Il a ainsi introduit l'idée d'énergie du réseau de neurone et fait des optimisations sous contrainte par recherche de son minimum. Les réseaux de Hopfield peuvent être utilisés comme mémoire associative [Mez90].

Ensuite la Machine de Boltzmann fut développée par Hinton, Sejnowski et Ackley [Hin84], [Hin86] en ajoutant un aspect stochastique à la recherche du minimum d'énergie.

Enfin, une extension de l'algorithme LMS a conduit au développement de l'algorithme de "rétropropagation de gradient" qui fonctionne sur un Perceptron multicouche [Par85], [Rum86].

b- Description d'un modèle simplifié de RNF

Le neurone biologique a bien évidemment beaucoup influencé le développement du neurone formel, et on utilise souvent dans les deux cas les mêmes termes pour décrire des choses relativement différentes.

Le neurone biologique recoit des signaux en provenance des neurones voisins en certains points de contact spécialisés : les synapses. Le modèle le plus simple est celui d'un neurone passant d'un état repos à un état excité si la somme des signaux en provenance des autres neurones excède un certain seuil. L'effet d'un neurone i sur un neurone j dépend de la nature et de la quantité de neurotransmetteurs chimiques libérés par la synapse : il existe des synapes excitatrices et des synapses inhibitrices. On pourra trouver une description détaillée du neurone biologique dans [Arb87] et [Darpa88].

On appelle *neurone formel* des unités de traitement très simples, généralement combinatoires et non linéaires. Les neurones sont reliés entre eux par un très grand nombre d'interconnexions, appelées *synapses* et caractérisées par leur *poids ou coefficient synaptique*. L'influence d'un neurone sur un autre est pondérée par ce poids. La sortie d'un neurone *(son état)* dépend de la somme de toutes ces influences selon une fonction non linéaire.

Ainsi, toute la mémoire du réseau est contenue dans ces synapses, le neurone étant uniquement un élément combinatoire. Par exemple, le neurone de McCulloch et Pitts est binaire et peut prendre deux états (0,1), selon la loi suivante :

$$s_i = \begin{cases} 1 \text{ si } \sum_{i=1}^{n} w_{ij}s_j \geq \vartheta_i \\ 0 \text{ si } \sum_{i=1}^{n} w_{ij}s_j < \vartheta_i \end{cases} \qquad \text{où} \begin{cases} s_j \text{ est l'état du neurone } U_j \\ w_{ij} \text{ le poids synaptique entre } U_i \text{ et } U_j \\ \vartheta_i \text{ est le seuil du neurone } U_i \end{cases}$$

Sur la Figure I.1, j'ai représenté un réseau simple de 5 neurones où chacun est connecté à tous les autres. Les deux représentations de cette figure sont identiques ; cependant, la description matricielle permet d'envisager simplement une réalisation physique, comme on le verra dans les chapitres suivants.

Sur le réseau de la Figure I.1, chaque neurone est connecté à tous les autres. Cette architecture de réseau est la plus naturelle pour l'étude des mémoires associatives ; elle est relativement différente de l'architecture en couche (cf. Fig. I.2), plus naturelle pour l'étude des associations arbitraire entre une entrée (par exemple

sensorielle) et une sortie (par exemple motrice). L'architecture en couche sera plus largement étudiée dans le cadre de la machine de Boltzmann.

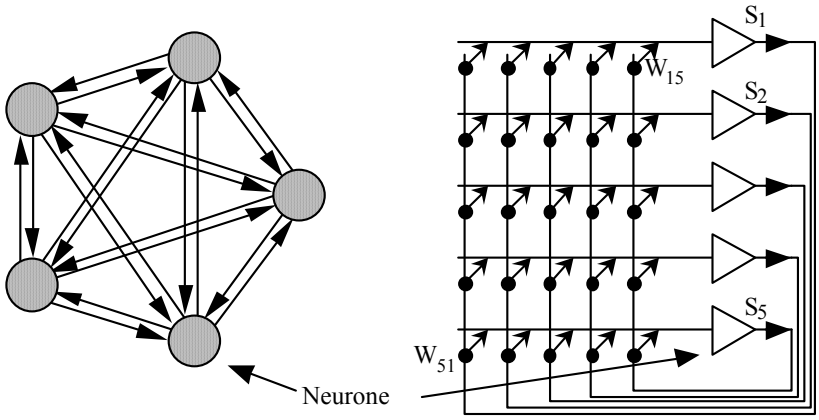


Figure I.1 : Réseau de Neurone Formel

I.1.2- Intérêt des Réseaux de Neurones Formels ?

De nombreux chercheurs, issus de différents domaines (biologie, mathématique, physique, optique, électronique, robotique…) travaillent actuellement sur les applications où les algorithmes des RNF. Les caractéristiques qui expliquent un tel engouement sont, à l'instar du cerveau humain, le parallélisme, la capacité d'adaptation, la mémoire distribuée, et l'émergence de la connaissance.

Le parallélisme apporte une plus grande vitesse de calcul, et présente donc un intérêt évident pour le concepteur d'ordinateurs, éternellement à la recherche de puissance.

La capacité d'apprentissage et d'adaptation est sûrement une propriété, qui aurait besoin d'être mieux formulée. Il est vrai qu'un Réseau de Neurone Formel n'a pas à être programmé et qu'il est doté de capacités d'apprentissage, mais il ne sait de toute façon apprendre que des données qu'il est capable d'apprendre. Ainsi l'architecture choisie est d'une très grande importance et il n'est pas encore sûr que cette propriété simplifiera la vie du programmeur.

La mémoire est distribuée, c'est à dire qu'elle émane d'un comportement collectif du réseau. Un élément de mémoire n'est pas physiquement localisable comme dans la mémoire d'un ordinateur conventionnel, mais il est réparti sur tout le réseau. Un altération d'une partie du réseau ne se transforme pas systématiquement en perte d'une trace mémoire, et le réseau présente donc une certaine tolérance aux défauts. Un autre avantage est de permettre une évolution dynamique du réseau vers un état mémorisé dans le cadre d'un environnement de données floues.

L'émergence de la connaissance évite, enfin, de formuler précisement le problème et cette propriété intéresse les concepteurs de systèmes experts.

Le champ d'application est donc très vaste et regroupe, la classification de patterns, la reconnaissance de la parole, la vision des machines, la robotique, le traitement de signal, les problèmes d'optimisation et de calcul.

Quoi qu'il en soit, après de nombreuses années d'étude, les réseaux de neurones se sont avérés bien adaptés à la résolution des problèmes de reconnaissance de formes et de classification. Dans ce domaine, ils apporteront sûrement une alternative efficace aux ordinateurs classiques, et nous avons étudié plus particulièrement les RNF pouvant remplir ces tâches.

I.2- LA MACHINE DE BOLTZMANN

I.2.1- Introduction

Parmi les différents modèles de Réseaux de Neurones Formels existant, nous avons choisi celui de la Machine de Boltzmann. Il permet, en effet, des taux de reconnaissance particulièrement élevés lors d'une utilisation en reconnaissance des formes ou en classification de patterns. De plus, son comportement asymptotique se prête à une modélisation mathématique très utile.

Les applications de la machine de Boltzmann, outre la classification de patterns résistante au bruit, peuvent être :

- le diagnostic d'alarme. Elle a en effet de très bonnes performances de reconnaissance, mais cette application ne peut être envisagée que si des machines suffisamment rapides sont réalisées pour que ces algorithmes tournent en temps réel.
- la fusion multicapteur,
- la vision artificielle telles que l'extraction de chaîne de contours, l'identification de courbes, la segmentation d'images ou l'étiquetage en relaxation.

Je vais maintenant présenter les principales caractéristiques de la Machine de Boltzmann (§ 2.2). Je présenterai ensuite une étude publiée par Kohonen sur les performances comparées de plusieurs schémas de réseaux de neurones formels (§ 2.3). Je présenterai enfin les algorithmes de relaxation stochastique (§ 2.4) et d'apprentissage de la Machine de Boltzmann (§ 2.5). Un paragraphe particulier apportera plus de détail sur l'utilisation de la température (§ 2.6). Enfin, je donnerai des exemples d'utilisation des Machines de Boltzmann (§ 2.7).

I.2.2- Caractéristiques

Le modèle de Machine de Boltzmann que nous allons considérer ici comprend des neurones à états binaires {0, 1}. Il se distingue de la plupart des autres modèles de Réseaux de Neurones Formels (RNF) par le fait que cette machine évolue par relaxation stochastique. La différence est notable avec le neurone de McCulloch et Pitts décrit au paragraphe § I.1.1, puisque ici l'entrée du neurone ne détermine pas si son état est "1" ou "0", mais qu'elle fixe la probabilité que celui-ci vaille "1" ou "0".

Dans ce modèle, les neurones (ou unités) sont classés en trois groupes :

- les neurones d'entrée,
- les neurones cachés,
- les neurones de sortie.

L'état des neurones d'entrée est imposé par les stimuli extérieurs, tandis que la sortie du réseau est représentée par l'état des neurones de sortie. Les neurones cachés sont reliés à des neurones des deux autres groupes ou à d'autres neurones cachés, mais leur état est entièrement inconnu de l'observateur et il évolue librement.

L'entrée et la sortie sont donc des vecteurs binaires de dimension différente et la machine peut être facilement utilisée pour réaliser des tâches de classification ou de reconnaissance des vecteurs d'entrée.

Le schéma de connexions entre ces neurones peut être arbitraire. Cependant, le schéma de connexions "dit en couche" est l'architecture la plus utilisée dans les différentes applications de la Machine de Boltzmann. Le schéma en couche est illustré par la figure I.2, elle représente un schéma à une seule couche cachée, une couche d'entrée composée des neurones d'entrée et une couche de sortie composée des neurones de sortie. Dans un schéma en couche, il n'existe pas d'interconnexions synaptiques directes entre les neurones de sortie et ceux d'entrée. Il peut y avoir plusieurs couches cachées et il peut exister des interconnexions synaptiques entre neurones d'une même couche.

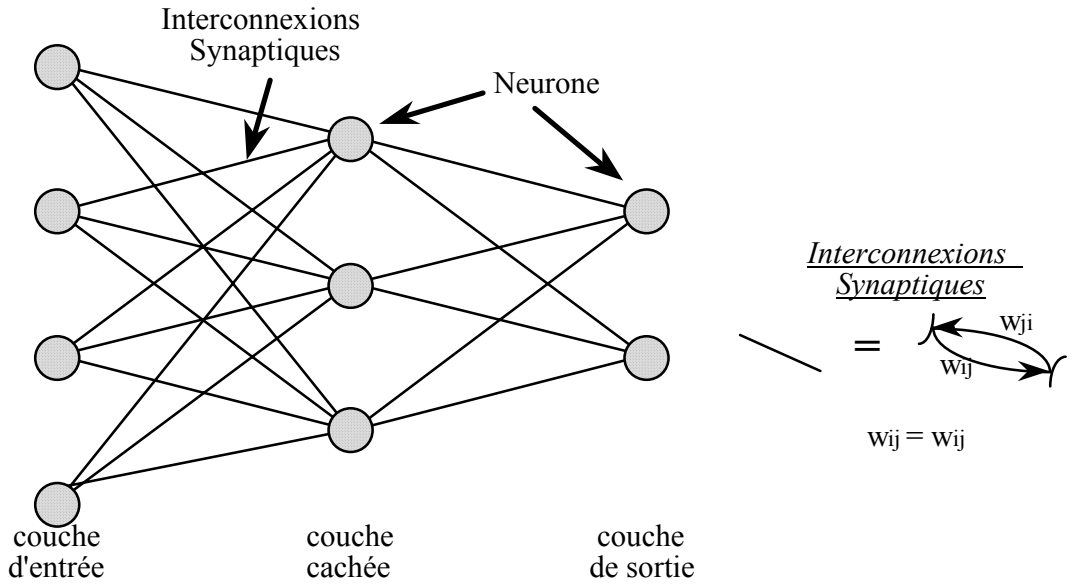
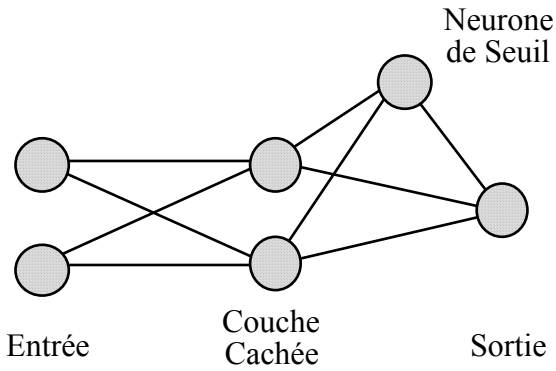


Figure I.2 : Réseau de Neurone Formel multi-couche.

Sur la figure I.3, j'ai représenté un exemple d'utilisation de ce type d'architecture pour résoudre le célèbre problème du "xor" ("ou-exclusif"). C'est ce type de problèmes (d'ordre supérieur à 1) mis à jour par Minsky et Papert [Min69], qui avait remis en cause l'utilisation des réseaux de neurones tels que les perceptrons jusqu'aux récents progrès faits dans les algorithmes d'apprentissages (voir § I.1.1).

Grâce à ses neurones cachés, la machine se construit une représentation interne apte à résoudre le problème posé lors de l'apprentissage. Celui-ci consiste alors à montrer plusieurs fois à la machine les différents couples entrée-sortie définis dans le tableau.

On voit que dans cet exemple simple tous les vecteurs d'entrées possibles sont présentés à la machine et sont appris par celle-ci.



XOR

Entrée	Couche Cachée	Sortie
0 0	?	0
0 1		1
1 0		1
1 1		0

Figure I.3 : Exemple du XOR

I.2.3- Performances comparées de Reconnaissances des formes

En 1989, Teuvo Kohonen a étudié sur des benchmarks de reconnaissance statistique de patterns, les performances de trois types de RNF. Il les a comparés entre eux et par rapport à la limite théorique du problème traité [Koh89]. Ces trois types de réseaux sont : le perceptron multicouche avec rétropropagation de gradient (RP), la machine de Boltzmann (MB) et le modèle "LVQ" (Linear Vector Quantization).

Les réseaux de neurones sont souvent basés sur un théorie déterministe de la décision, mais Kohonen a fait ce benchmark parce qu'il pense que les problèmes réels de classification de pattern doivent être statistiques. La théorie de la décision statistique utilisée est basée sur l'évaluation d'un coût moyen formulé en terme d'expression de Bayes pour les probabilités conditionnelles.

Le problème choisi est celui de la discrimination de deux classes C_1 et C_2. Les distributions des échantillons qui appartiennent à chaque classe sont choisies gaussiennes. Il y a une forte intersection des "distributions de classe", elle sont fortement non-linéaire sur les bords et les vecteurs d'entrée ont une dimension importante. La limite théorique de reconnaissance peut être mathématiquement déterminée sur ce problème artificiel, et il a été étudié pour des dimensions de 2 à 8. La classe C_1 est choisie avec une moyenne nulle et un écart type de 1 dans toutes les dimensions, la classe C_2 est choisie avec une moyenne égale à (x_1, 0, 0…) et un écart type de 2 dans toutes les dimensions.

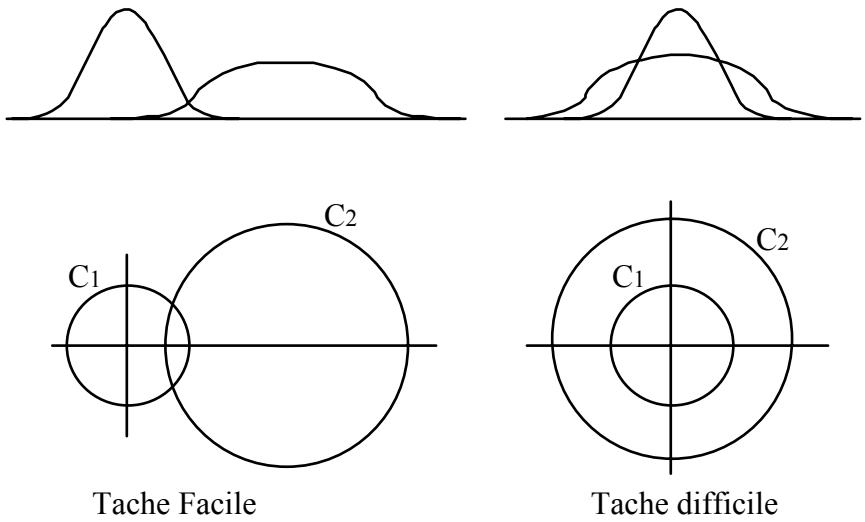


Figure I.5 : les deux première dimensions des distributions de C_1 et C_2.

Deux types de tâches ont alors été considérées : une tâche dite "facile" où x_1 vaut 2,32, et une tâche "difficile" où x_1 vaut 0. Les distributions des classes C_1 et C_2

ont été représentées sur la figure I.5 telles qu'elles seraient dans un problème de dimension 1, et telles qu'elles sont représentées dans [Koh89] dans le cas de la dimension 2. Les cercles représentent les écarts type des différentes classes.

La table I.2 décrit les résultats obtenus par les 3 types de Réseaux. Lorsque la dimension augmente, les performances de reconnaissance du réseau avec rétropropagation de gradient augmentent mais ils s'éloignent de la limite théorique : les résultats obtenus par cet algorithme sont les moins bons et il semble le moins utilisable pratiquement de ce point de vue. La Machine de Boltzmann a des performances qui sont très proches de la limite théorique de bonnes réponses : ce sont les meilleures observées. Le modèle LVQ donne des résultats intermédiaires.

Le temps d'apprentissage sur un ordinateur Masscomp MC 5600 était de : 5 heures pour la Machine de Boltzmann, de 1 heure pour la Rétropropagation de gradient, de 20 minutes pour le modèle LVQ.

Tâche	dim.	Lim. Th.	RP	MB	LVQ
X_1=2,32	2	16,4	16,4	16,5	17
	3	13,7	14	14	14,6
	4	11,6	12,5	11,7	13,1
	5	9,8	11	10,2	12,2
	6	8,4	10,8	8,7	10,7
	7	7,2	9,72	8,2	10,1
	8	6,2	11,3	6,7	10,0
X_1=0	2	26,4	26,3	26,5	26,5
	3	21,4	21,5	21,6	21,8
	4	17,6	19,4	18	18,8
	5	14,8	19,5	15,2	16,9
	6	12,4	20,7	12,7	15,3
	7	10,6	16,7	11	14,5
	8	9	18,9	9,4	13,4

Table I.2 : Pourcentages d'erreur sur des données artificielle, d'après [Koh89].

En conclusion, la machine de Boltzmann offre les meilleures performances de reconnaissance, mais présente le défaut d'être très lente à mettre en œuvre sur un ordinateur classique, même très puissant.

I.2.4- Relaxation stochastique

Le modèle de la machine de Boltzmann a été tout d'abord imaginé par Hinton et Sejnowski en 1984, [Hin84], [Hin86]. Dans leur modèle, un seul neurone change d'état à la fois dans un ordre d'examen fixé (séquentiel ou aléatoire) de telle sorte que tous les neurones aient la possibilité de changer d'état à chaque itération. Ce modèle est appelé "Machine de Boltzmann Séquentielle (ou Machine de Boltzmann Asynchrone)". Plus récemment, en 1989, Robert Azencott a étudié un nouveau modèle appelé "Machine de Boltzmann Synchrone", [Aze90a], [Aze90b]. Dans son modèle tous les neurones changent d'état simultanément à chaque itération.

Si la sortie du neurone U_i est reliée à l'entrée du neurone U_j par une connexion synaptique, la force de cette connexion est définie par son poids W_{ij}. Dans la machine de Boltzmann, les synapses sont symétriques, c'est-à-dire que le poids reliant le neurone U_i au neurone U_j est égal au poids reliant le neurone U_j au neurone U_i. Ces poids peuvent être positifs (excitation) ou négatifs (inhibition) et ils sont, de plus, généralement bornés en amplitude. On dit que deux neurones sont voisins s'il existe une connexion synaptique entre eux.

L'influence entre neurones voisins se fait au travers d'une relation de voisinage linéaire. L'influence totale sur un neurone donné des neurones de son voisinage est appelée "la contribution du réseau". Elle est égale à la somme des états des neurones du voisinage pondérée par la valeur des poids synaptiques.

La machine de Boltzmann est à temps discret et l'état du neurone à l'instant (n+1) est calculé en fonction de la contribution du réseau à l'instant n.

Notons :

- $(U_i)_{i\square\{1...N\}}$, l'ensemble des neurones,
- X_i^n , l'état du neurone U_i à l'instant n,
- W_{ij}, le poids synaptique reliant les neurones U_i et U_j,
- ϑ_i le seuil du neurone U_i.

La contribution du réseau V_i^n sur le neurone U_i est égale, à l'instant n, à :

$$V_i^n = \sum_{j\neq i} W_{ij}.X_j^n - \vartheta_i \qquad (I.1)$$

Cependant on peut s'affranchir du terme ϑ_i dans l'équation (I.1) ci-dessus en supposant que chaque neurone est connecté à un neurone de seuil U_s dont l'état est constamment X_s=1. Le poids de cette connexion est alors :

$$W_{is} = -\vartheta_i \tag{I.2}$$

Ceci conduit à l'expression suivante de V_i^n :

$$V_i^n = \sum_{j \neq i} W_{ij}.X_j^n \tag{I.3}$$

Rappelons ici que les influences entre neurones étant symétriques, on a :

$$w_{ij} = w_{ji} \tag{I.4}$$

Considérons un ensemble de N neurones. Chaque configuration Ω du réseau peut être caractérisée par un vecteur d'état, dont les coordonnées sont binaires et représentent les états des N neurones du réseau.

Etant donné une configuration Ω à l'instant n, la probabilité que l'état du neurone U_i soit nulle à l'instant n+1 est :

$$P\left(X_i^{n+1}=0\&\Omega\right) = \frac{1}{\left[1 + \exp(V_i^n/T)\right]} \tag{I.5}$$

Le contrôle de la quantité d'aléa du réseau se fait par l'intermédiaire du paramètre strictement positif T, appelé température.

Si la température T devient faible, la machine tend vers un comportement déterministe, et l'équation (I.5) est proche de :

$$P\left(X_i^{n+1}=0\right) = \begin{cases} 1 \text{ si } V_i^n > 0 \\ 0 \text{ si } V_i^n < 0 \end{cases} \tag{I.6}$$

La dénomination de température pour le paramètre T vient de l'analogie que l'on peut faire entre le réseau de neurone et un système physique de particules. En effet la règle de décision du nouvel état du neurone de l'équation (I.5) est la même que pour une particule à 2 états énergétiques. Un système de telles particules en contact avec un thermostat T convergera éventuellement vers un équilibre thermique, et la probabilité de trouver le système dans une configuration globale donnée obéira à une distribution de Boltzmann, d'où le nom donné à ce RNF. Par analogie, un réseau de neurones évoluant selon cette règle de décision convergera vers un "équilibre thermique" tel que toute configuration Ω ait une probabilité P(Ω) de la forme :

$$P(\Omega) = \frac{1}{Z_t} \exp\left(-\frac{E(\Omega)}{T}\right) \tag{I.7}$$

Z_t est le terme de normalisation de cette distribution de probabilité :

$$Z_t = \sum_{\Omega} \exp\left(-\frac{E(\Omega)}{T}\right) \tag{I.8}$$

Dans le cas du modèle de la Machine de Boltzmann Séquentielle, $E(\Omega)$ est assimilable à l'énergie totale de la configuration Ω et son expression est alors [Hin85] :

$$E(\Omega) = E_{SEQ}(\Omega) = -\sum_{i,j} W_{ij}.X_i.X_j \tag{I.9}$$

Par extension, $E(\Omega)$ est aussi appelé énergie totale de la configuration Ω dans le cas du modèle de la Machine de Boltzmann Synchrone, mais son expression est alors plus compliquée [Aze90a] :

$$E(\Omega) = E_{SYN}(\Omega) = -T.\sum_{i} \log\left[1 + \exp\frac{V_i(\Omega)}{T}\right] \tag{I.10}$$

Dans le cas synchrone, à très haute température T les différents neurones tendent à être complètement indépendants les uns des autres et aucun apprentissage n'est possible.

Dans ces deux modèles, durant la phase de relaxation, le système rejoint en théorie le minimum global en énergie puisque la règle (I.5) lui permet de quitter un minimum local en sautant occasionnellement dans une configuration à plus haute énergie.

I.2.5- Apprentissage dans la Machine de Boltzmann

a- Introduction

La machine de Boltzmann est surtout utilisée comme classifieur. On donne sur une base d'exemples une fonction d'association entre un motif d'entrée et une sortie désirée du réseau. L'apprentissage de la Machine de Boltzmann va donc consister à

présenter à la machine les différents couples (motif-étiquette) de la base d'exemples et à ajuster les poids pour que l'état des neurones de sortie correspondent après stabilisation à l'étiquette désirée.

Cet apprentissage est donc supervisé et la phase d'apprentissage est distincte de la phase de reconnaissance. La phase de reconnaissance consiste en une seule relaxation stochastique de la machine décrite ci-dessus, alors que l'apprentissage est décomposé en un grand nombre de ces relaxations. En réalité lors des phases de reconnaissance, on doit retrouver l'étiquette en sortie du réseau même si le motif d'entrée est peu différent de celui appris. L'apprentissage consiste à apprendre à la machine des règles d'association sur des exemples et généralement tous les patterns d'entrée possibles ne sont pas appris contrairement à ce que l'on a vu au paragraphe (§ I.2.2) dans l'exemple simple du "XOR".

Je vais décrire tout d'abord l'apprentissage utilisé dans le modèle Séquentiel de Hinton et Sejnowski [Hin85], je présenterai ensuite la régle d'apprentissage dérivée par Azencott [Aze90a] dans le cas de la Machine de Boltzmann Synchrone.

b- Apprentissage dans le modèle Asynchrone

Pour chaque exemple de la base de donnée, on laisse évoluer la machine dans deux phases de relaxation distinctes. Dans ces deux phases, le motif d'entrée à apprendre est imposé sur les neurones d'entrée. Dans la phase forcée, l'étiquette associée à ce motif est imposée sur les neurones de sortie. Cette phase est aussi appelée phase "plus". Dans la phase libre (ou phase moins), l'état des neurones de sortie n'est plus imposé. Les neurones cachés, mais aussi les neurones de sortie dans la phase libre, évoluent librement dans une relaxation dynamique.

La différence de comportement du réseau entre ces deux phases conduira à une modification des poids synaptiques. La règle d'apprentissage de la Machine de Boltzmann est étonnement simple et locale. Chaque synapse observe, lorsque la machine a atteint son équilibre statistique et pendant les deux phases de relaxation d'apprentissage, l'état des neurones U_i et U_j qu'elle relie. Elle calcule alors la probabilité P_{ij} que les neurones U_i et U_j soient allumés ensemble dans une même séquence de relaxation des états des neurones.

Appelons P_{ij}^+ la probabilité P_{ij} calculée en phase forcée et P_{ij}^- celle calculée en phase libre.

Les probabilités P_{ij}^+ et P_{ij}^- peuvent être estimées en comptant les *cooccurrences* des états actifs des neurones U_i et U_j pendant M séquences de relaxation lorsque l'équilibre statistique est atteint, soit en calculant :

$$P_{ij} = \frac{1}{M} \sum_{n=m}^{m+M} X_i^n X_j^n \qquad (I.11)$$

Les poids W_{ij} sont mis à jour après l'apprentissage d'un ensemble d'exemples ("batch") en fonction de la moyenne des P_{ij} et en accord avec la règle :

$$\Delta w_{ij} = \gamma.\left(P_{ij}^{+} - P_{ij}^{-}\right) \qquad (I.12)$$

γ est un paramètre de faible amplitude, afin d'assurer la convergence de l'apprentissage. Il décroît vers 0 quand le nombre de mise à jour des poids augmente (on affine ainsi l'apprentissage).

Cet algorithme (I.12) a été conçu de manière à minimiser une mesure théorique appelée distance de Kullbach. Il assure une descente dans le sens du gradient de la surface représentant cette mesure en fonction des poids [Hin84]. Cette mesure relate les différences entre ce que l'on veut et ce que l'on a à entrées fixées, c'est à dire les différences entre les probabilités des états dans les phases forcée et libre.

Lors de la mise en œuvre de cet algorithme, l'équation (I.12) est souvent remplacée par l'équation (I.13) ci-dessous :

$$\begin{cases} \Delta w_{ij} = \varepsilon.\mathrm{sign}\left(P_{ij}^{+} - P_{ij}^{-}\right) \text{si } \left|P_{ij}^{+} - P_{ij}^{-}\right| > \partial \\ \Delta w_{ij} = 0 \qquad \text{sinon} \end{cases} \qquad (I.13)$$

le paramètre ε joue le rôle du paramètre γ : il est de faible amplitude et il décroit au cours de l'apprentissage. Le paramètre ∂, par contre, peut être fixe.

c- Apprentissage dans le modèle Synchrone

R. Azencott est alors parvenu à dériver une nouvelle loi d'apprentissage autorisant un changement simultané des états des neurones lors des phases de relaxation.

Les neurones tirant dans ce cas leur état au sort simultanément, les lois d'apprentissage sont différentes de l'apprentissage séquentiel. Même si la loi de mise à jour des poids (I.12) reste inchangée dans l'apprentissage synchrone, les probabilités P_{ij}^{+} et P_{ij}^{-} sont maintenant égales aux probabilités que les neurones U_i et U_j s'allument dans des <u>successions immédiates</u>, respectivement dans les phases forcée et libre.

Chaque probabilité P_{ij}^{+} et P_{ij}^{-} peut être estimée en comptant, lorsque l'équilibre statistique est atteint, les *occurrences successives* des états actifs des neurones U_i et U_j pendant M séquences de relaxation. La loi (I.11) devient :

$$P_{ij} = \frac{1}{2.M} \sum_{n=m}^{m+M} \left(X_i^{n-1} X_j^{n} + X_j^{n-1} X_i^{n} \right) \qquad \text{(I.14)}$$

I.2.6- Température T du réseau

Revenons un instant sur le paramètre T de température. En effet, son ajustement est un point critique dans l'utilisation de la machine de Boltzmann et son mode d'utilisation diffère selon les auteurs. Hinton et Sejnowski effectuent un "recuit simulé", c'est à dire qu'ils font décroître lentement la température vers zéro au cours de la relaxation. Au contraire Azencott effectue la relaxation à température constante.

Si la température ne tend pas vers 0, les neurones de sortie n'atteignent jamais un état constant par contre la machine tend vers une stabilisation statistique. Elle n'est pas *figée* en fin de reconnaissance et si on observe la machine pendant suffisamment longtemps, on pourra adjoindre une probabilité d'apparition à chaque vecteur de sortie possible. Une étiquette sera dans ce contexte considérée comme étant "la" sortie si elle a une très grande probabilité d'apparition. Ceci présente l'avantage d'offrir une information plus riche sur la qualité du motif reconnu.

La température est par contre ajustée au problème à traiter lors de l'apprentissage et elle est très souvent variable au cours de celui-ci [Aze90b].

I.2.7- Exemples d'Utilisation : Reconnaissance à partir de contour

Je vais présenter ici un exemple d'application de la Machine de Boltzmann, publié par R. Azencott dans [Aze90c]. On pourra voir sur cet exemple d'application quels sont les types et les tailles des réseaux nécessaires à la réalisation d'une application.

Cet exemple consistait en la reconnaissance de contours de bateaux et en leur classification en 4 catégories définies a priori.

Les images de bateaux originales étaient des images infra-rouge digitalisées sur 512x512 pixels en 64 niveaux de gris. Les contours ont été extraits par des algorithmes standards, et ils ont été soumis à des déformations et à un bruitage

artificiel afin d'augmenter le nombre total d'exemples disponibles jusqu'à 275, parmi lesquels 25 ont servi pour le contrôle et 250 pour l'apprentissage.

Les contours après lissage ont été découpés en 50 arcs d'égale longueur et chaque arc a été décrit par trois paramètres : son orientation moyenne $\overline{\theta}$, sa courbure moyenne $\overline{\rho}$ et l'erreur locale de lissage $\overline{\varepsilon}$.

Ces 150 données $\left(\overline{\theta_i}, \overline{\rho_i}, \overline{\varepsilon_i}\right)_{i=\{1\ldots50\}}$ sont utilisées pour activer trois couches d'entrées de 50 neurones binaires utilisant un seuillage adaptatif. On trouve ensuite 5 couches cachées de 50 neurones dont 3 couches sont dites "d'analyse locale" et 2 "d'évaluation jointe". Enfin, on trouve 2 neurones en sortie pour différencier les quatres classes.

Le nombre de synapses est faible sur cet exemple (environ 2000) et il y a 402 neurones reliés à une couche de codage de 150 neurones. L'apprentissage synchrone a été conclu en 500 itérations de changement de poids et le temps de calcul était de plus de 2 heures sur une Connection Machine.

Les taux de reconnaissance sont de 97% sur l'ensemble d'apprentissage (250 exemples) et de 95% sur l'ensemble total (275 exemples). Ceci montre que les Machines de Boltzmann Synchrone forment une famille de RNF intéressante et qu'elles ont de bonnes aptitudes à résoudre des problèmes de vision de bas et moyen niveau.

Sur cet exemple il a pu être vérifié que la machine est très peu sensible à un bruit venant se mélanger à un pattern d'entrée. On dit qu'elle est très résistante au bruit.

I.3- CONCLUSIONS

J'ai montré dans le paragraphe (§ I.2.3) l'intérêt des machines de Boltzmann pour la résolution de problème de reconnaissance des formes en faisant sa comparaison avec d'autres modèles. Dans le paragraphe (§ I.2.7), j'ai présenté un exemple d'utilisation de la machine de Boltzmann Synchrone pour remplir des tâches de vision de bas et de moyen niveau. Ce dernier modèle est beaucoup plus rapide que celui de Hinton et Sejnowski, lorsqu'il est mis en œuvre sur un calculateur massivement parallèle, avec un gain en vitesse de l'ordre du nombre de neurones.

Nous voyons ainsi qu'une Machine de Boltzmann utile pour des tâches d'intelligence artificielle significatives requiert plusieurs milliers de neurones ayant chacun plusieurs centaines de synapses, l'exemple du (§ I.2.7) restant encore une petite application.

Malgré son très grand intérêt théorique, ce modèle a été assez peu expérimenté, l'apprentissage étant trop long à mettre en œuvre sur un ordinateur classique pour des applications pratiques, (cf. page 98 de [Darpa88]).

Il faut aussi noter que la machine de Boltzmann présente des caractéristiques très intéressantes si on envisage la réalisation d'une machine neuronale spécialisée massivement parallèle :

- Tout d'abord, les poids synaptiques sont mis à jour en parallèle à partir de données locales (cf. eqs I.12, I.11, I.14).
- De plus, les travaux d'Azencott permettent maintenant d'envisager une implantation parallèle réellement efficace de la relaxation où tous les neurones choisissent leur état en même temps.
- L'état des neurones est binaire. Même si on choisit de profiter des capacités d'intégration apportée par un recours aux techniques analogique, la réalisation d'une carte d'émulation pour un ordinateur hôte peut être facilitée en utilisant des techniques et interfaces standards.

Pour toutes ces raisons, et afin d'apporter un élément de réponse à la question posée à la fin de l'introduction, nous avons choisi de réaliser des circuits analogiques implantant l'algorithme de la Machine de Boltzmann Synchrone.

CHAPITRE II

REALISATIONS ANALOGIQUES DE RESEAUX DE NEURONES : ETAT DE L'ART

II.1- REALISATIONS DE RESEAUX DE NEURONES FORMELS

De nombreuses réalisations de RNF analogiques ont déjà été envisagées, pour des applications très variées telles que la vision, l'audition, la reconnaissance de caractères… Plusieurs livres ont déjà été consacrés à l'implantation analogique de RNF, et on peut se référer, en particulier, à [Mea89a] ou [Mea89b] pour plus de détails sur ces réalisations.

Toutes ces réalisations peuvent être caractérisée par le fait qu'elles suivent l'une ou l'autre des trois approches suivantes :

- La construction du système est directement inspirée d'observations biologiques sans que l'on recherche à décrire son fonctionnement par un modèle mathématique, [Mea89b].
- La construction du système est inspirée d'un modèle mathématique existant, mais celui-ci est modifié afin qu'il s'adapte mieux aux spécificités de l'implantation massivement parallèle, voir par exemple [Als87a].
- La construction du système cherche à respecter scrupuleusement un modèle mathématique afin que sa stabilité et sa convergence soient garanties.

La réalisation présentée dans cette thèse rentre à la fois dans la deuxième catégorie et la troisième catégorie. En effet, la Machine de Boltzmann synchrone a été développée parce que la machine asynchrone du modèle de Hinton, Sejnowski et Ackley [Hin84] n'était pas adaptée à une implantation massivement parallèle, et ce système a appartenu un temps à la deuxième catégorie. Cependant, ce modèle reste

encore très long à simuler et une modification supplémentaire du modèle est pour nous difficile. Ainsi, nous chercherons à le respecter scrupuleusement en considérant notre système dans la troisième catégorie.

Je présenterai tout d'abord les autres réalisations de machine de Boltzmann actuellement envisagées par d'autres laboratoires (§ II.2). Puis je présenterai les solutions qui ont été décrites par ailleurs pour le calcul de la contribution d'un réseau sur un neurone (§ II.3).

II.2- MACHINE DE BOLTZMANN

II.2.1- Laboratoires Bellcore (USA)

La réalisation la plus avancée de la machine de Boltzmann est celle de l'équipe d'Alspector des laboratoires Bellcore aux Etats-Unis, [Als87a], [Als87b], [Als89]. L'algorithme choisi est celui de Hinton et Sejnowski. La machine est vue comme une machine déterministe faisant un recuit simulé. Ainsi, les cooccurrences sont comptées sur un seul bit. Le détail du neurone et de la synapse sont représentés sur la Figure II.1.

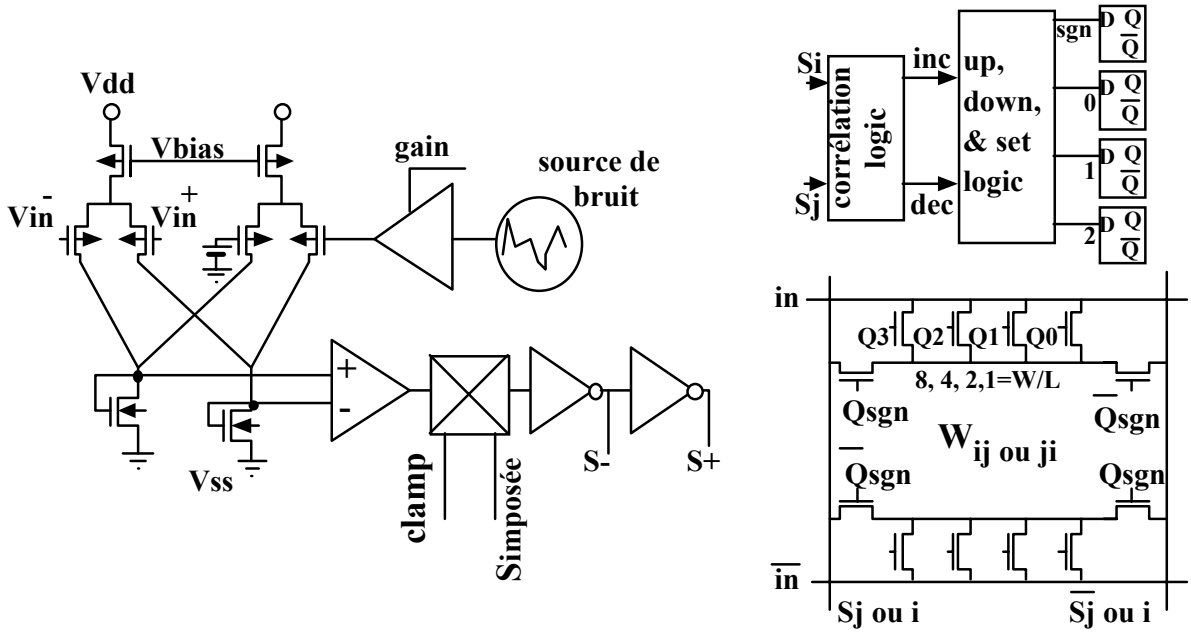


Figure II.1 : d'après [Als87a].

L'implantation est mixte Numérique/Analogique : les poids numériques sont codés sur 4 bits + 1 bit de signe, et ils sont mémorisés dans des bascules D. Le premier générateur aléatoire envisagé a utilisé l'amplification du bruit thermique dans des résistances. Cependant, devant l'instabilité de ces amplificateurs à fort gain, une seconde réalisation utilise un générateur pseudo-aléatoire pour faire l'approximation d'un bruit à distribution gaussienne, qui est lui-même l'approximation d'un bruit à fonction de répartition sigmoïdale.

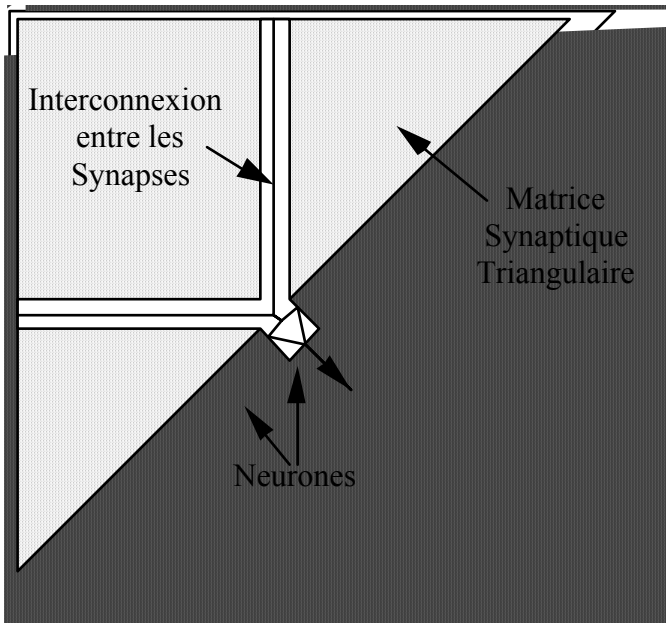


Figure II.2 : Matrice triangulaire.

L'architecture choisie est entièrement connectée, et la réalisation d'un réseau à couches est possible au prix d'un grand nombre de synapses inutilisés. Pour pouvoir avoir des poids symétriques, la matrice de connexions synaptiques est triangulaire, voir la Figure II.2.

II.2.2- Autres Réalisations de Machine de Boltzmann

Goser, de l'Université de Dortmund en Allemagne avait envisagé une architecture entièrement numérique [Kre88]. Il a renoncé à la réalisation pour des problèmes de taille, mais aussi à cause du choix du modèle de Hinton et Sejnowski qui donnait de mauvais résultats en simulation, lorsque plusieurs neurones commutent en même temps.

Le Laboratoire DEC de Paris entreprend la réalisation d'un émulateur de réseaux de Neurones qui marche pour la Machine de Boltzmann, [Sku90]. La

réalisation se fait à partir de Xilinx 3020. L'algorithme utilisé est celui de Hinton et Sejnowski modifié en ajoutant un poids w_{ii} pour permettre la commutation de plusieurs neurones en même temps (pas de justification théorique).

Un autre projet consiste en la proposition d'une architecture basée sur une technique de modulation de densité d'impulsion par une équipe de Finlande et de Oxford (GB), [Tom90].

II.3- AUTRES REALISATIONS DE RESEAUX DE NEURONES

La réalisation de machines spécifiques dédiées à la simulation et à l'expérimentation de Réseaux de Neurones Formels est un domaine de recherche très actif actuellement Je ne pourrai pas citer toutes ces réalisations et on pourra se référer à la bibliographie en fin de ce manuscrit pour en obtenir une description. Je ne vais citer que les réalisations ayant un rapport assez proche avec la thèse décrite ici.

On peut tout d'abord citer la réalisation de deux circuits spécifiques par les Laboratoires Mitsubishi au Japon [Ari91a], [Ari91b]. Ils sont basés sur une architecture proche de celle décrite par Alspector dans [Als87a] mais ils sont entièrement déterministes. L'appellation Machine de Boltzmann est donc abusive pour ce réseau qui n'est pas stochastique. Ces deux circuits ont été réalisés sur une technologie CMOS 1µm et ils utilisent des techniques de calcul analogique pour faire l'approximation du modèle mathématique. Le circuit le plus récent comprend 336 neurones et 28000 synapses avec apprentissage, il mesure 14,5∞14,5 mm^2, il consomme 3W et il utilise un support PGA à 393 broches.

Une autre réalisation analogique de RNF intéressante est celle des Laboratoires ATT aux Etats-Unis [Bos91]. Elle est plus particulièrement dédiée à des applications de reconnaissance de caractère et elle se présente sous la forme d'un circuit spécifique sans possibilités d'apprentissage interne. Ce circuit utilise le calcul analogique en interne et ses entrées/sorties sont numériques, afin de simplifier son intégration dans un système. Les calculs sont apparemment en partie multiplexés dans le temps. Il comprend 4096 synapses et quelques neurones. Il a été réalisé sur une technologie CMOS 0,9µm, il mesure 4,5∞7 mm^2 et il consomme environ 1W. Les auteurs ont mis en évidence sur leur application une trop grande imprécision des poids, qui les a conduit à réaliser la couche de sortie de façon numérique.

On peut citer en France, une autre réalisation analogique de RNF qui est à l'étude à l'INPG [Ros89]. Ce circuit est dédié à des applications de mémoires associatives, il n'a pas de capacités d'apprentissage mais les poids du réseau sont programmables.

Pour les réalisations de circuits spécifiques entièrement parallèles de Réseaux de Neurones Formels, on a souvent recours à des techniques analogiques du fait de leur plus grande densité d'intégration et de leur plus faible consommation. Une autre possibilité est d'utiliser des techniques numériques et de sacrifier un peu de parallèlisme [Per90] en ayant recours par exemple à des calculs bit-série, ou tout autre technique de multiplexage temporel.

Ainsi, l'architecture de machine MIND, construite par une équipe du Centre d'Etude Nucléaire de Grenoble, est en ce sens originale [Per90], [Gam91]. Cette machine dédiée à l'expérimentation et le développement d'une grande classe d'algorithmes d'apprentissage est construite avec des composants d'usage général. Dans la première version à 128 Neurones de cette architecture (MIND-128), les techniques de calcul analogique ont été utilisées bien que ce calcul ne se fasse pas au cœur d'un circuit intégré mais sur une plaque de circuit imprimé. Les auteurs ont utilisé des convertisseurs numérique-analogique 4 bits pour transposer les poids dans le domaine analogique. Ils ont alors montré que ceux-ci n'étaient pas assez précis pour permettre de connecter plus de 500 synapses en entrée d'un neurone. La version suivante de cette machine (MIND-1024) n'utilise plus que des techniques numériques [Gam91].

On peut citer en France d'autres travaux sur les architectures semi-parallèles numériques : ceux de l'Ecole Polytechnique et de l'ESPCI (circuit 64 neurones pour les réseaux de Hopfield [Per89], [Wei90]), ceux du LEP (circuit LNEURO [Dur89], [The90]), ceux de l'INPG (machines CRAZY [Gue89] et SMART [Bes91]) et ceux du CSI [Oua91].

II.4- ETAT DE L'ART DES SOMMATEURS

Plusieurs voies pour la réalisation du calcul de la contribution du réseau ont déjà été explorées. Les premiers circuits construits utilisaient des résistances sur couche mince qui autorisaient deux valeurs fixes et positives de poids et des états binaires de neurones [Gra86]. Ensuite, des circuits avec des poids stockés dans des mémoires digitales, soit ternaires {+1,0,-1} [Siv86], [Ver89], soit multivalués [Als87a], contrôlant des transistors MOS utilisés en résistance, ont été décrits : ils permettent d'avoir des poids binaires, programmables et signés, et des états de neurone binaires. Finalement, des circuits plus récents sont caractérisés par des poids analogiques signés et une génération de courant par amplificateur à transconductance : ils

permettent d'avoir des poids analogiques signés, et des états de neurone binaires ou multivalués [Ros89], [Ebe88], [Vit89a], [Vit89b].

La précision est le problème majeur d'une réalisation analogique de la fonction "somme de produits", et il faut particulièrement l'étudier quand beaucoup de synapses sont connectées au même neurone. Ce problème a été traité dans le cas de neurones à états ternaires par [Ver90]. Nous avons toujours cherché à prendre ce problème en compte lors de la conception du circuit de "somme du produit" présenté ici.

II.5- CONCLUSION

Cette revue de l'état de l'art nous a permis de constater que les autres réalisations de Machine de Boltzmann avaient peu d'espoir de fonctionner avec de "grands" réseaux. Nous avons donc choisi de concevoir une machine respectant au mieux le modèle mathématique afin d'assurer un fonctionnement stable et la convergence des algorithmes de reconnaissance et d'apprentissage. Nous avons de plus orienté notre travail, dès le départ du projet, vers une réalisation de grands réseaux de neurones formels par un ensemble de circuits bien adaptés.

CHAPITRE III

ARCHITECTURE

III.1- INTRODUCTION

La réalisation d'une machine de Boltzmann analogique Synchrone, décrite au chapitre I, suppose l'implantation des équations (I.3), (I.5), (I.13) et (I.14). Les deux premières décrivent la phase de reconnaissance alors que les deux dernières décrivent la phase d'apprentissage.

Ainsi, pour la réalisation d'une machine destinée à remplir une tâche bien définie où l'apprentissage n'est pas nécessaire, les équations (I.3) et (I.5) sont suffisantes. La machine est alors dite dédiée à la reconnaissance et son apprentissage est fait sur un ordinateur puissant avant sa construction. Les équations (I.3) et (I.5) seront réalisées par deux cellules que nous appelerons respectivement cellule synapse et cellule neurone. De plus, la cellule synapse mémorise le poids synaptique W_{ij}.

La réalisation d'une machine ayant en plus des capacités d'apprentissage se fait alors en complétant la cellule synapse par des moyens de modification du poids W_{ij} respectant les équations (I.13) et (I.14).

Dans ce chapitre, je présenterai d'abord les architectures des cellules neurones et synapses réalisant ces fonctions. Ensuite je décrirai comment elles sont regroupées dans des circuits comprenant neurones et synapses pour former ainsi une machine complète. Je montrerai ensuite comment un module de traitement analogique couplé à un circuit de communication numérique aboutit à un système analogique, entièrement configurable par logiciel et dédié à la reconnaissance. L'intérêt de ces choix sera illustré par des exemples.

III.2- DESCRIPTION FONCTIONNELLE DE LA CELLULE NEURONE ET DE LA CELLULE SYNAPSE

III.2.1- Premiers choix architecturaux

Dans la plupart des réalisations analogiques de RNF citées dans le chapitre III, la contribution du réseau est représentée par un courant, et le calcul de la somme de l'équation (I.3) se fait en utilisant l'infaillible loi des nœuds. Toutefois, ces schémas se distinguent principalement par le fait qu'ils sont soit simple rail, soit double rail.

Dans un schéma mono-rail, un seul fil de connexion est utilisé et le sens du courant représente alors le signe de la contribution du réseau, voir Figure (III.1).

Dans un schéma double-rail, deux fils de connexion sont utilisés et le sens du courant dans ces fils peut alors être fixe. Un des fils est utilisé pour représenter les contributions positives, l'autre pour représenter les contributions négatives.

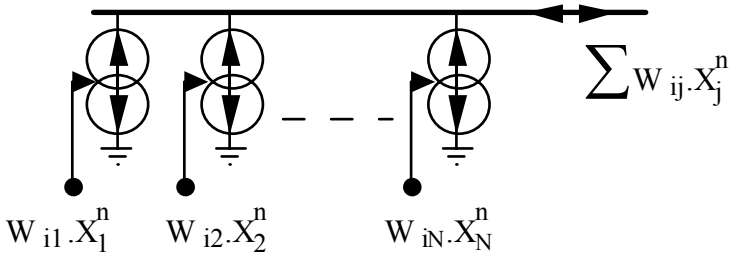


Figure III.1 : schéma mono-rail.

La symétrie des schémas double-rail permet d'obtenir une compensation d'un certain nombre de défauts du mono-rail et d'avoir des systèmes avec un meilleur rejet des bruits des alimentations. Ils sont largement utilisés pour la réalisation des systèmes analogiques où l'information peut être maintenue sous forme différentielle à l'intérieur des circuits intégrés. On trouve, par exemple, ce type de schémas dans les filtres à temps continu pour améliorer la linéarité des résistances réalisées avec des transistors MOS, et dans les filtres à capacités commutés pour diminuer la charge injectée par les interrupteurs MOS, [Tsi85].

Cependant, ils utilisent deux fois plus d'interconnexions que les schémas mono-rail et cela constitue un handicap pour la réalisation de RNF répartis sur plusieurs circuits. En effet, lorsque plusieurs circuits sont cascadés, les points critiques sont alors le nombre d'interconnections sur la plaque de circuit imprimé et le nombre de plots d'Entrée/Sortie des circuits intégrés.

Nous avons vu, dans le chapitre I, qu'une Machine de Boltzmann capable de remplir des tâches utiles d'intelligence artificielle requiert plusieurs milliers de neurones ayant chacun plusieurs centaines de synapses. On ne peut pas réaliser un tel système sur un seul circuit intégré avec les technologies CMOS disponibles aujourd'hui. Ainsi, nous avons envisagé la possibilité de répartir le calcul de la contribution du réseau sur plusieurs puces en cascade et nous avons donc choisi d'utiliser un schéma mono-rail. Nous verrons que cette solution est viable si les états de neurone sont binaires.

III.2.2- Génération de nombres aléatoires

a- Introduction

La cellule neurone calcule le nouvel état du neurone par un tirage au sort et il est donc nécessaire de construire au préalable un générateur aléatoire. La probabilité d'apparition d'un état dépend de la contribution du réseau sur le neurone (cf. eq. I.5) et ce tirage au sort doit donc être "biaisé" en conséquence.

L'équation (I.5) exprimant la probabilité de l'état d'un neurone est équivalente à l'équation (III.1) suivante :

$$P\left(X_i^{n+1}=1\right) = S_i \qquad \text{(III.1)}$$

La fonction $S_i = f(V_i)$ est appelée fonction "sigmoïde" et elle vaut :

$$S_i = \frac{1}{\left[1 + \exp(-V_i^n/T)\right]} \qquad \text{(III.2)}$$

La fonction sigmoïde et de sa dérivée sont représentées sur la Figure (III.3).

Je vais présenter deux solutions pour implanter l'équation (III.1) :

Solution A : on peut tout d'abord comparer la quantité S_i à une variable aléatoire Y de telle sorte que X_i vaille 1 si S_i est supérieur à Y. L'équation (III.1) devient alors :

$$P\left(Y<S_i\right) = S_i \qquad \text{(III.3)}$$

Cette équation traduit l'expression de la fonction de répartition de la variable aléatoire Y et Y doit donc être uniformément distribuée sur l'intervalle [0…1], comme représenté sur la Figure III.2.

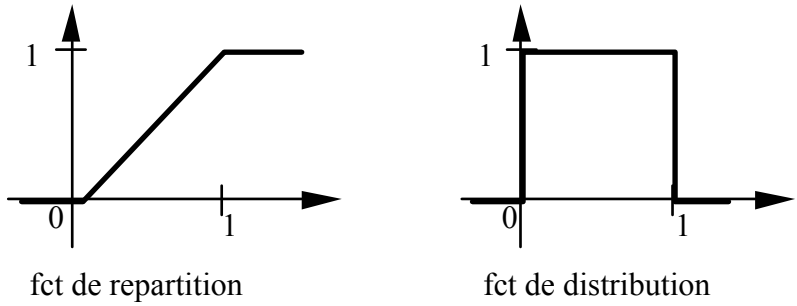


Figure III.2 : loi à distribution uniforme

Solution B : on peut aussi choisir de comparer la grandeur V_i à une variable aléatoire Y de telle sorte que X_i vaille 1 si V_i est supérieur à Y. L'équation (III.1) se transforme alors en l'expression de la fonction de répartition de Y :

$$P\left(Y<V_i\right) = \frac{1}{\left[1 + \exp(-V_i/T)\right]} \qquad \text{(III.4)}$$

Cette fonction est représentée ainsi que la fonction de distribution associée sur la Figure III.3 dans le cas où T=0,25.

Figure III.3 : fonction sigmoïdale

b- Loi uniforme à partir de nombres aléatoires binaires

Pour la solution A, il faut disposer d'une variable aléatoire à distribution uniforme dans l'intervalle [0…1]. Sa construction peut se faire, comme l'a montré Patrick Garda, par filtrage d'un flot de nombres binaires aléatoires dont les valeurs 0 et 1 sont équiprobables. La variable aléatoire Y est définie à l'instant n par :

$$Y^n = \sum_{i=0}^{\infty} a^{n-i}.2^{1-i} \qquad \text{(III.5)}$$

où a^k □ {0, 1} est un nombre binaire aléatoire à l'instant k.

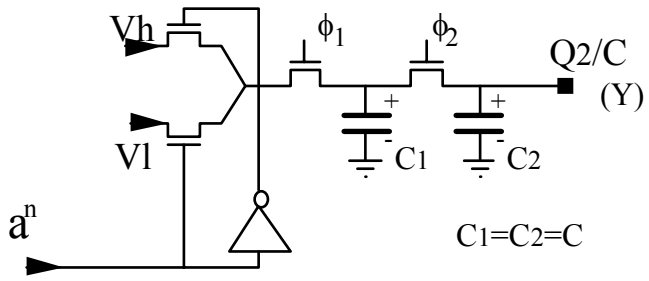


Figure III.4 : construction de la loi uniforme.

Ce filtrage peut se faire simplement en utilisant deux capacités C_1 et C_2 de même valeur C, représentées sur la Figure III.4. La tension aux bornes de la capacité C_2 représente la variable Y. Plus précisément notons :

$$Y = \frac{V_{out} - V_l}{V_h - V_l}$$

Pendant la phase ϕ_1, la capacité C_1 est préchargée à V_h si a^n vaut 1 et à V_l si a^n vaut 0. On a donc :

$$\begin{cases} Q_1/C = (1-a^n).V_l + a^n V_h \\ Q_2/C = (V_h-V_l).Y^{n-1} + V_l \end{cases}$$

Pendant la phase ϕ_2, les deux capacités sont reliées par un interrupteur et ainsi la charge Y à l'instant précédent est divisée par 2 alors que la moitié de la charge stockée dans C_1 lui est ajoutée :

$$V_{out} = \frac{Q_1}{C} = \frac{Q_2}{C} = \frac{(1-a^n).V_l + a^n V_h}{2} + \frac{(V_h-V_l).Y^{n-1} + V_l}{2}$$

d'où l'équation de récurrence $Y^n = \frac{V_{out} - V_l}{V_h - V_l} = \frac{a^n+Y^{n-1}}{2}$

de laquelle on déduit l'équation III.5 par une résolution simple.

Si l'on suppose que la relaxation se fait à une fréquence de 300 kHz, que 1000 neurones (cachés ou de sorties) changent d'état et qu'il faut 10 bits pour construire un échantillon de la loi uniforme indépendant du précédent, il faut alors un flot de 3.10^9 bits aléatoires par seconde. Le problème se ramène donc à la génération de nombre aléatoires binaires avec un grand débit et une très bonne indépendance

statistique. Nous avons étudié pour cela plusieurs solutions. Je vais présenter ici deux solutions :

- une ayant recours à un générateur aléatoire optoélectronique remplissant tout à fait ces contraintes,
- une autre ayant recours à un générateur pseudo-aléatoire à base d'automates cellulaires.

c- Génération des nombres binaires : Le speckle Optique

Le principe utilisé consiste à échantillonner l'image d'un speckle optique. Ce phénomène bien connu des opticiens est généralement un phénomène parasite apparaissant avec une lumière monochromatique cohérente. Le phénomène est ici provoqué par l'interférence d'une lumière cohérente ayant plusieurs modes de parcours dans une fibre optique multimode. On dispose donc en bout de fibre, d'une image composée de grains lumineux et de grains sombres. La fibre est agitée pour que cette image soit dynamique et constitue ainsi une image de bruit. Le schéma de principe du montage utilisé est décrit sur la Figure III.5.

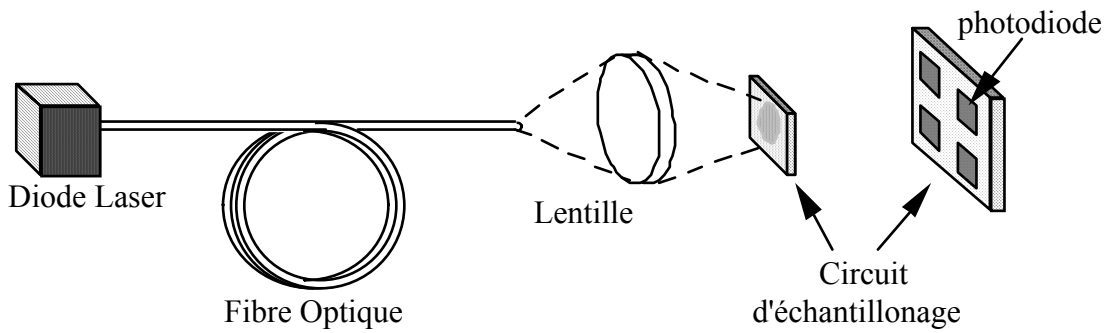


Figure III.5 : speckle optique

L'échantillonnage de l'image du speckle (son acquisition) se fait avec un circuit à photodiodes. Nous avons utilisé pour sa réalisation une technologie VLSI CMOS classique dans laquelle les photodiodes sont réalisées avec les diffusions normalement destinées à la réalisation des sources et drains des transistors. Les photodiodes sont tout d'abord préchargées en inverse, puis elles se déchargent en fonction du nombre de grains de speckle les éclairant. Il y a ainsi une double intégration de l'intensité lumineuse : tout d'abord spatiale sur la surface de la photodiode, mais aussi temporelle pendant le temps de décharge considéré. Malgré tout, la valeur de la tension aux bornes de la photodiode à l'issue d'un temps fixé

reste aléatoire et on obtient des nombres aléatoires binaires en comparant cette tension à un seuil fixe.

Le fonctionnement optique et les qualités statistiques du speckle ont été décrits dans [Lal89], alors que le circuit d'échantillonnage à photodiodes l'a été dans [Mad90]. Je ne présente ici que des résultats généraux et indispensables à l'élaboration d'un générateur aléatoire.

L'intensité moyenne déchargeant une photodiode dépend de beaucoup de paramètres : la puissance du laser, la taille relative du grain de speckle et de la photodiode, de la dynamique de l'image du speckle… La distribution de probabilité de la tension aux bornes de la photodiode à l'issue de sa décharge est représentée sur la Figure III.6 en fonction d'un paramètre appelé M qui dépend du rapport entre la taille du grain de speckle et celle de la photodiode.

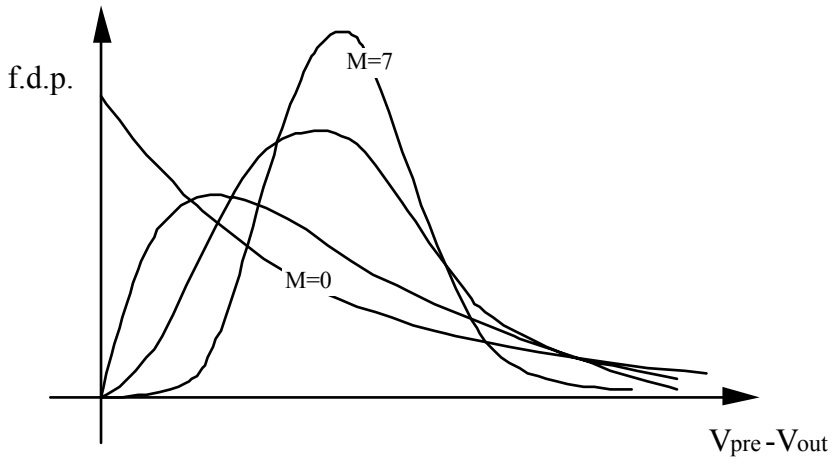


Figure III.6 : distribution des tensions V_{out} après décharge.

Dans [Lal89] et [Mad90], il a été montré que pour obtenir des nombres aléatoires binaires, utiliser une courbe proche de l'exponentielle (c'est-à-dire M proche de 0), minimise l'influence de la tension de seuil sur la répartition statistique des échantillons binaires de sortie, voir Figure III.7.

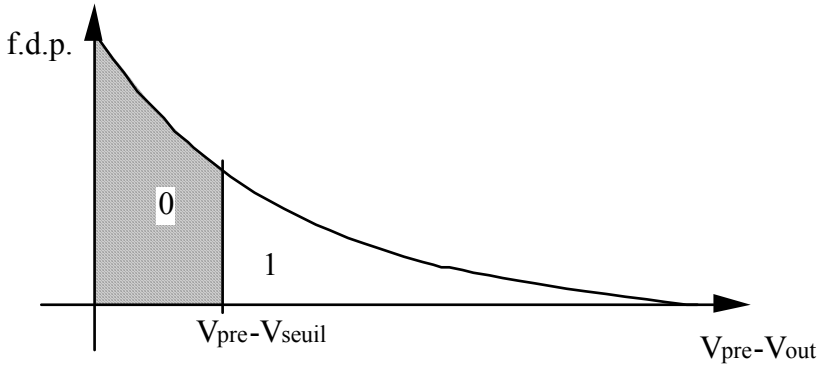


Figure III.7 : seuillage de V_{out} après décharge.

Ph. Lalanne a de plus démontré que, dans ce cas, la corrélation entre deux échantillons binaires aléatoires est inférieur à 10^{-3}, pour un débit de 10^5 échantillons par photodiode et par seconde.

d- Générateur Pseudo-Aléatoire à Automate Cellulaire.

Une alternative au speckle optique pour la génération de nombre binaires aléatoires est l'utilisation d'un générateur pseudo-aléatoire. Ceux-ci présentent l'avantage d'être réalisables sur une technologie numérique classique et d'être parfaitement reproductible et à fonctionnement sûr.

Le principe utilisé est celui de la génération d'une séquence de nombres binaires à l'aide d'un système séquentiel et déterministe. Si les séquences sont "suffisamment" longues et si deux nombres binaires successifs sont "suffisamment" indépendants statistiquement, on peut alors utiliser un tel système comme générateur aléatoire. Le terme "suffisamment" doit être explicité en fonction de l'application à laquelle ce générateur est destiné. Une séquence est "suffisamment aléatoire" pour une application dans un système particulier si les calculs que ce système fait effectivement ne sont pas assez sophistiqués pour qu'il soit capable de discerner une régularité dans la séquence [Wol86].

De nombreuses réalisations ont déjà été présentées (par registre à décalage linéaire, multiplication par a modulo n, …), et je vais décrire ici un système réalisé à l'aide d'un automate cellulaire mono-dimensionnel et présenté par Wolfram dans [Wol86]. Patrick Garda a eu l'idée d'utiliser ce générateur pseudo-aléatoire et a étudié en détail ses propriétés statistiques.

Un automate cellulaire mono-dimensionnel à état binaire consiste en une ligne de sites ayant pour valeur X_m égale à 0 ou 1. Ces valeurs sont mises à jour en

parallèle (simultanément), à des instants définis (temps discret) et en accord avec une régle fixée (Figure III.8).

Ainsi, ici, le site "m" calcule son nouvel état X_m^n à l'instant n suivant la relation :

$$X_m^n = X_{m-1}^{n-1} \; H\left(X_m^{n-1} + X_{m+1}^{n-1} \right) \qquad \text{(III.6)}$$

où "H" représente la fonction logique "ou-exclusif" et "+" la fonction "ou".
Cette règle est essentiellement non-linéaire.

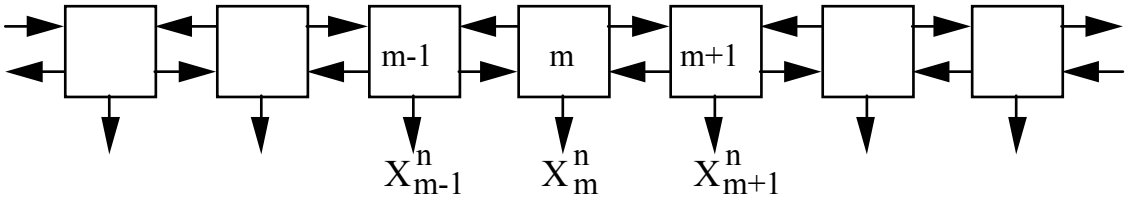


Figure III.8 : automate cellulaire mono-dimensionnel.

Le nombre de sites étant limité pratiquement à un nombre N fini, on utilise pour le calcul de l'état du site "1" la valeur du site "N" et vice versa ; l'automate est ainsi circulaire.

Pour un automate cellulaire comprenant M_t sites, la longueur des séquences dépend des conditions initiales et elle est dans tous les cas inférieure à M_t. Pour une configuration initiale où tous les sites sont à "0" sauf un seul à "1", Stephen Wolfram a constaté que l'automate évolue de manière tout à fait chaotique et qu'il ne se forme pas alors de structures régulières.

Pour notre application, nous utilisons donc un tel automate cellulaire circulaire comprenant plus de sites que de sorties binaires aléatoires. Ceci nous permet d'assurer que les filtres de la Figure III.4 seront reliés à des sites suffisamment éloignées pour délivrer des nombres statistiquement indépendants, voir Figure III.9.

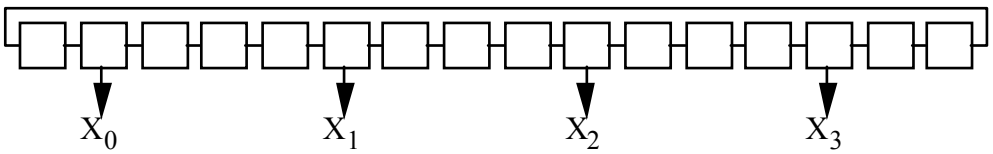


Figure III.9 : générateur pseudo-aléatoire

Cette solution présente le désavantage sur les réalisations de générateurs pseudo-aléatoires à base de registres à décalage de ne pas permettre de séquence de

longueur maximale. Cependant, elle présente l'énorme avantage de délivrer directement un grand nombre d'échantillons indépendants et d'être utilisable avec une loi d'évolution asynchrone pour une utilisation dans un réseau de neurones à temps continu. L'équation III.6 d'évolution de l'état de neurone devient alors :

$$X_m(t+\Delta t)=X_{m-1}(t)\ H\left(X_m(t) + X_{m+1}(t) \right) \qquad \text{(III.7)}$$

e- Génération de Nombre aléatoires à répartition sigmoïdale

Pour la solution B, il faut disposer d'une variable aléatoire à fonction de répartition sigmoïdale comme la variable Y de l'équation III.4. On utilise pour cela le speckle dans une configuration un peu différente de celle présentée au paragraphe (§ c) de cette section.

En effet, si on considère comme échantillon aléatoire la tension aux bornes de la photodiode après un temps de décharge constant plutôt que cette tension aprés seuillage, alors la fonction de répartition des échantillons aléatoires est proche d'une sigmoïde lorsque le paramètre M est grand (par exemple 5) , voir la publication [Lal90] reproduite en Annexe.

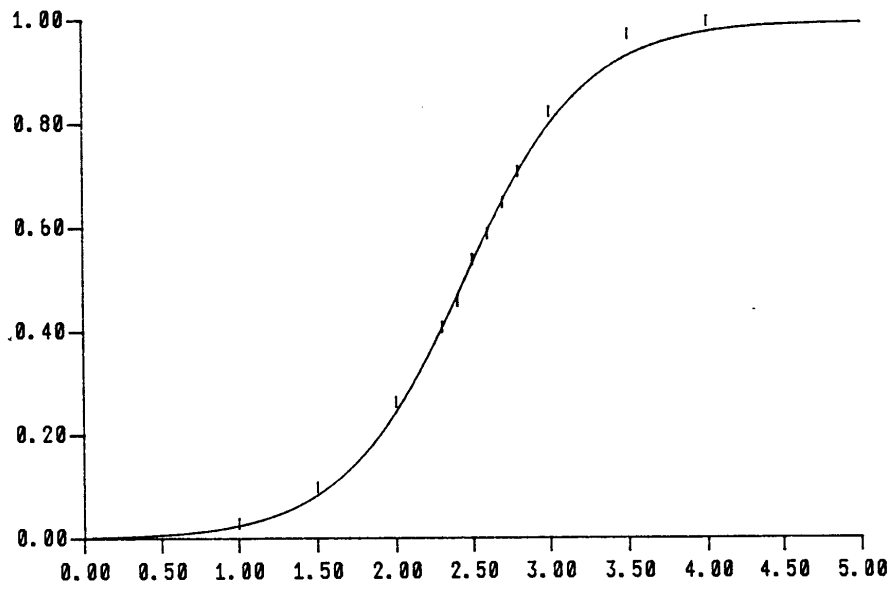


Figure III.10 : Mesures et comparaison avec une sigmoïde.

La fonction de répartition d'échantillons relevés expérimentalement ainsi que la fonction sigmoïdale qui en est la plus proche sont représentées sur la Figure III.10.

On peut observer sur cette Figure, la très grande qualité des résultats obtenus. La tension de précharge de la matrice de photodiodes est 5 Volts et la puissance du Laser a été réglée pour obtenir une tension de décharge moyenne de 2,5 Volts. La fréquence d'horloge est 10 kHz et 10000 mesures différentes ont été faites montrant une déviation standard de 1%.

En conclusion, on voit que la réalisation d'un générateur aléatoire avec une répartition des échantillons sigmoïdale sur une gamme complète de tension (de 0 à Vdd) peut aisément être réalisée par l'utilisation du speckle optique. On obtient ainsi une très grande indépendance des échantillons. Cette qualité d'indépendance est due au fait que l'origine des aléas est optique, mais aussi au fait que leur détection est totalement extérieure au circuit de traitement.

III.2.3- Cellule Neurone A

L'implantation fonctionnelle complète de la fonction neurone est représentée sur la Figure III.11.

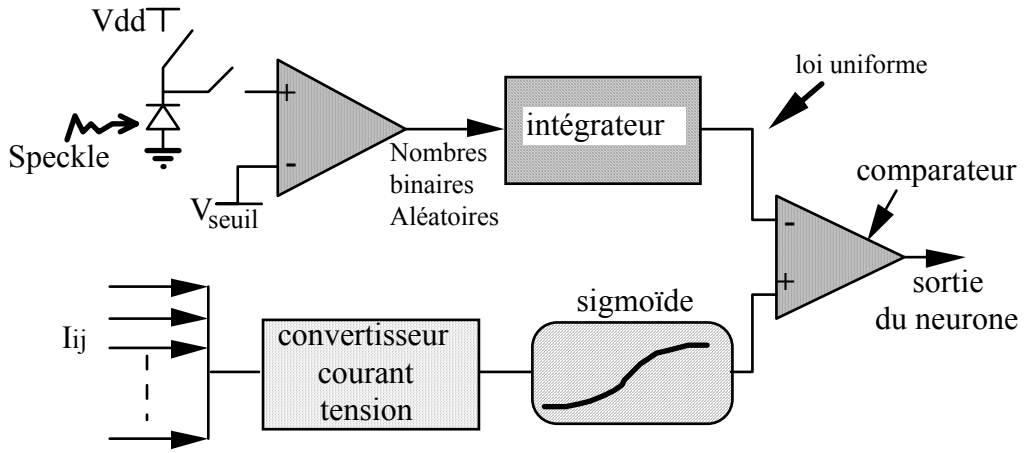


Figure III.11 : cellule neurone A

La partie génération de la loi de distribution uniforme comprend successivement : la photodiode, un comparateur de binarisation et l'intégrateur à capacités. La tension V_{seuil} doit être réglée en fonction de la puissance du laser. Elle peut être réglée depuis l'extérieur du circuit, mais si on veut compenser la dispersion des sensibilités des photodiodes et la non-uniformité de l'image du speckle optique, il faut alors concevoir un système local de détermination de la tension V_{seuil}.

Les cellules synapses délivrent des courants qui sont additionnés sur la connexion d'entrée de la cellule neurone. Ce courant total est ensuite converti en

tension par un convertisseur courant-tension. De plus, ce convertisseur maintient constant le potentiel du nœud d'entrée et sa transconductance est utilisée pour implanter la température T.

La fonction sigmoïde prend en entrée la tension issue du convertisseur représentant V_i^n /T et délivre en sortie une grandeur représentant la variable S_i (cf eq. III.2). L'état du neurone est ensuite décidé par un comparateur.

L'avantage de cette méthode est que la réalisation du neurone est indépendante du générateur aléatoire. Le circuit d'échantillonnage de l'image du speckle optique peut être différent de celui contenant les cellules neurones, et on peut avoir une connexion numérique à haut débit entre eux.

III.2.4- Cellule Neurone B

Une alternative pour l'implantation de la cellule neurone est représentée sur la Figure III.12. Elle utilise la solution B pour la génération de nombres aléatoires.

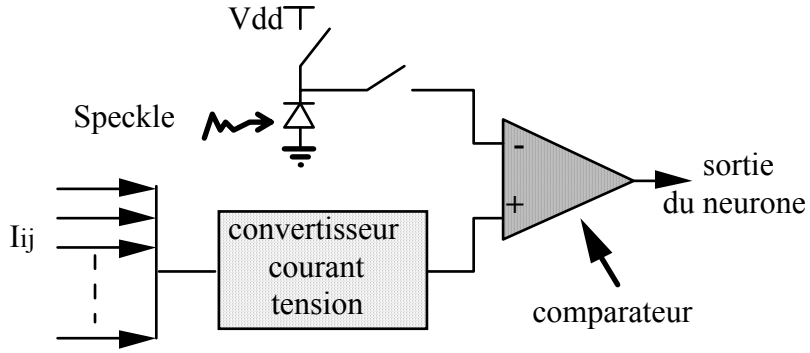


Figure III.12 : cellule neurone B

L'état du neurone est déterminé par comparaison directe entre la tension aux bornes de la photodiode aprés la décharge et la tension représentant V_i^n /T. Le convertisseur tension-courant est le même que pour la cellule neurone A et sa transconductance implante en particulier la température T.

Une alternative, non présentée ici, utilise plus avant les propriétés du speckle et évite le recours à un convertisseur courant-tension à gain variable pour implanter la température T. Celle-ci est alors obtenue par la modulation de la puissance laser et par l'adaptation automatique de la tension de précharge, en fonction de la tension de décharge moyenne de la photodiode.

Avec cette méthode, l'échantillonnage de l'image du speckle et l'implantation de la fonction neurone se font sur le même circuit. Ceci présente l'avantage de limiter

le nombre de connexions par le recours à la technologie optique, mais présente le désavantage de compliquer la conception du circuit et sa mise en place sur une plaque de circuit intégré.

III.2.5- Cellule Synapse

L'implantation de la fonction synapse nécessaire à la relaxation est représentée sur la Figure III.13. La synapse représentée ici relie la sortie du neurone U_j à l'entrée du neurone U_i et réalise l'implantation de l'équation (I.3). Elle est décomposée en deux éléments : un organe de stockage analogique du poids W_{ij} sous forme d'une tension dans une capacité et un convertisseur tension-courant prenant en entrée la valeur analogique du poids et la valeur binaire de l'état de neurone, et délivrant en sortie un courant I_{ij} tel que :

$$I_{ij} = W_{ij}.X_j \qquad \text{(III.8)}$$

La modification du poids se fait soit par un système de rafraîchissement à partir d'une donnée extérieure, soit par un bloc d'apprentissage venant modifier la charge de la capacité de stockage.

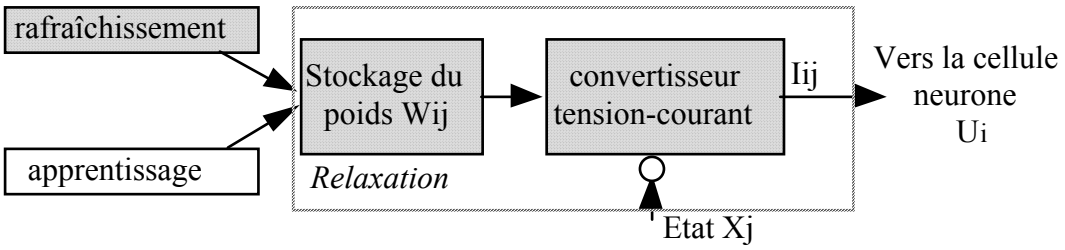


Figure III.13 : cellule synapse

La modification du poids W_{ij} se fait suivant l'équation (I.13) à partir des cooccurrences de l'équation (I.14). Ainsi les données nécessaires à l'apprentissage de W_{ij} sont les états des neurones U_i et U_j observées pendant M itérations de relaxation. Ces données sont binaires et locales.

La Figure III.14 représente le bloc d'apprentissage.

Il comprend :

- une unité de modification du poids qui apporte ou retire une charge à la capacité de stockage de poids.
  Nous appelerons ce type d'unité :*compteur analogique* .
- Un deuxième compteur analogique qui accumule des cooccurrences.

Nous appelerons cette unité : *compteur de cooccurrences*.

- Un bloc de comparaison qui détermine s'il doit y avoir incrémentation, décrémentation, ou non modification du poids, en fonction de la sortie du compteur de cooccurrences et selon l'équation (I.13).

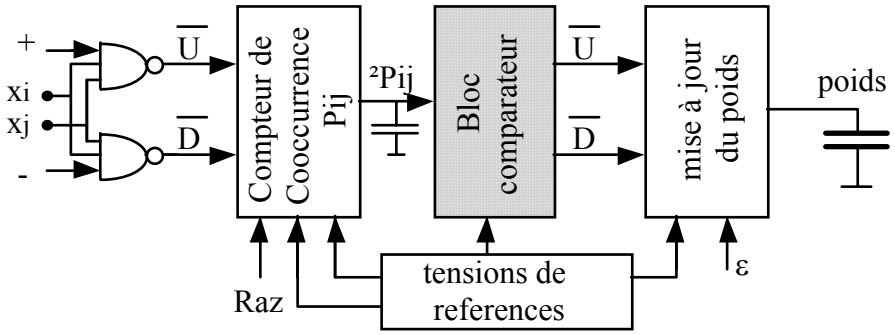


Figure III.14 : bloc d'apprentissage.

Le compteur analogique possède un signal d'entrée de comptage (U) et un signal de décomptage (D). Un troisième signal ε contrôle l'amplitude de la charge de modification du poids. Ce compteur est synchronisé par une horloge et il nécessite une demi-période de précharge avant chaque opération.

Le compteur analogique utilisé pour le calcul des cooccurrences est complété avec un signal de remise à zéro (RAZ). Il est incrémenté dans la phase d'apprentissage en régime forcé et il est décrémenté en régime libre. Dans chacun de ces régimes il faut modifier le compteur deux fois par itération de relaxation :

- Dans la première phase, X_j est égal à l'état du neurone U_j à l'instant présent et X_i à l'état du neurone U_i à l'instant précédent.
- Dans la seconde phase, X_j est égal à l'état du neurone U_j à l'instant précédent et X_i à l'état du neurone U_i à l'instant présent.

Le signal (+) passe à 1 pour incrémenter le compteur en phase forcée, alors que le signal (-) reste à 0. En phase libre, les rôles de (+) et de (-) sont inversés.

Je présenterai, dans le chapitre IV, la conception de ces cellules. Dans le chapitre V, je présenterai mes réalisations et le test des blocs dessinés en grisé sur les Figures III.13 et III.14.

III.3- UN ENSEMBLE DE CIRCUITS POUR LA MACHINE DE BOLTZMANN

III.3.1- Introduction

Nous avons choisi de réaliser une machine de Boltzmann comprenant un grand nombre de neurones et capable de remplir des tâches utiles de reconnaissances de formes. Partant du constat qu'il est impossible de construire de telles machines sur un seul circuit avec les technologies actuelles, nous avons envisagé d'interconnecter plusieurs circuits VLSI sur une plaque de circuit imprimé.

Le choix de la cellule neurone de type A décrit dans les paragraphes ci-dessus, nous apporte une grande modularité. La modularité permet de concevoir indépendamment le circuit générateur aléatoire, le circuit neurone et le circuit synapse. Elle permet aussi de remplacer le système optique de générateur aléatoire par un générateur pseudo-aléatoire électronique sans avoir à modifier la conception du neurone. Enfin, le développement de ce système optoélectronique nous apporte une expérience qui sera utile lorsque les interconnexions optiques s'imposeront dans les réalisations de réseaux de neurones.

Je vais maintenant présenter un ensemble de circuits dédiés à la réalisation de machines de Boltzmann multicouches. Ensuite, je montrerai comment il est possible, par l'introduction d'un circuit de communication et d'un réseau d'interconnexion numérique, de concevoir une machine dédiée à la reconnaissance et configurable par programme. ces travaux ont été introduits dans [Bel89], [Bel90], [Gar90].

III.3.2- Machine avec apprentissage

a- Présentation

Un des problèmes majeurs de la réalisation analogique d'une machine de Boltzmann est qu'il faut assurer la symétrie des poids ($W_{ij} = W_{ji}$). Ce problème devient critique lorsque des techniques de mémorisation analogique sont utilisées lors de l'apprentissage.

Supposons que deux cellules différentes soient utilisées pour l'apprentissage des deux poids W_{ij} et W_{ji}. L'équation (I.14) étant symétrique en i et j, ces deux cellules auront les mêmes données en entrée et l'apprentissage ne pourra donc pas compenser une erreur sur le gain ε de l'équation (I.13). Une erreur de gain aussi petite soit-elle se traduira directement par une augmentation lente de la différence entre les poids W_{ij} et W_{ji} au cours de la phase d'apprentissage, et elle entraînera donc

un mauvais fonctionnement de la Machine. Il est donc impératif que l'apprentissage des poids W_{ij} et W_{ji} se fasse sur la même cellule.

Comme je l'ai déjà montré sur la Figure I.1 du chapitre I, une structure matricielle se prête très bien à une implantation physique de RNF entièrement connecté. Cependant, nous devons adopter une structure un peu différente pour que les poids W_{ij} et W_{ji} soient physiquement confondus. La Figure III.15 représente la matrice synaptique de connexions entre deux groupes de neurones A et B.

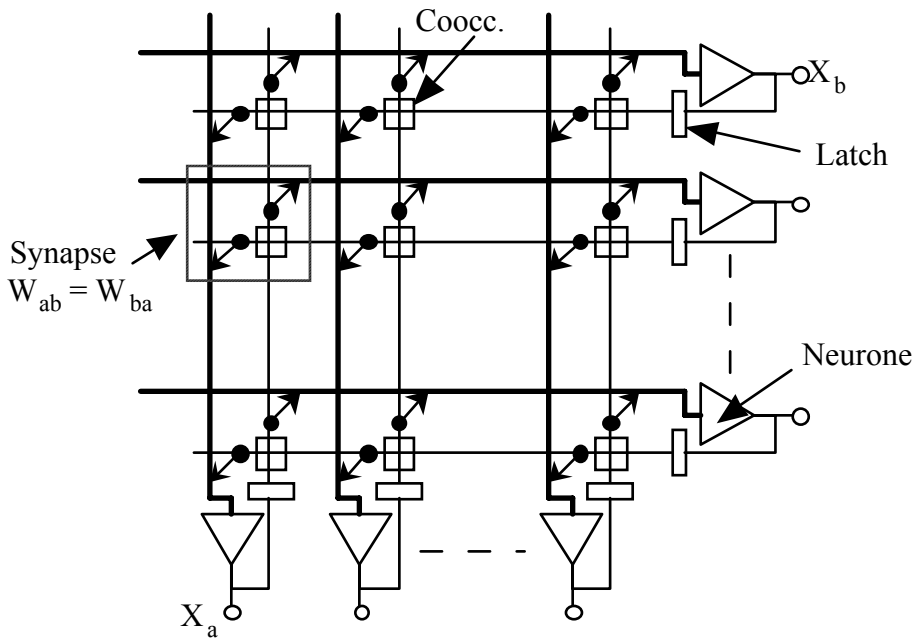


Figure III.15 : Couches A et B interconnectées

Sur cette Figure, on peut voir que tous les neurones du groupe A sont reliés à tous ceux du groupe B. A chaque neurone du groupe A (resp. du groupe B) est associé une colonne (resp. une ligne) de la matrice. La contribution du réseau sur les neurones du groupe A (resp. B) est calculée sur un fil vertical (resp. horizontal). Une cellule de latch est associée à chaque colonne (resp. à chaque ligne) de la matrice et mémorise l'état d'un neurone du groupe A (resp. B) à l'itération précédente. L'état du neurone est ensuite diffusé à la matrice en trois phases différentes. Les deux premières phases servent à l'incrément du compteur de cooccurrences (équ. I.14) et la troisième au calcul du V_i.

Appelons P_{ab}^{n} l'état du compteur de cooccurrences à l'instant n.

Lors de la première phase, les états des neurones du groupe A (resp. B) à l'itération précédente (resp. présente) sont distribués à la matrice et la synapse W_{ab} calcule :

$$P_{ab}^{n+\frac{1}{2}} = X_a^{n-1} . X_b^n + P_{ab}^n \qquad \text{(III.9)}$$

Lors de la deuxième phase, l'état des neurones du groupe A (resp. B) à l'itération présente (resp. précédente) sont distribués à la matrice et la synapse W_{ab} calcule :

$$P_{ab}^{n+1} = X_a^n . X_b^{n-1} + P_{ab}^{n+\frac{1}{2}} \qquad \text{(III.10)}$$

Lors de la troisième phase les états des neurones à l'instant présent sont distribués à la matrice et les grandeurs V_i à l'instant n+1 sont calculées suivant l'équation I.3.

Dans le schéma présenté ici, il ne peut pas exister de connexions entre neurones d'un même groupe. Ce cas est le plus courant dans les architectures à couche où les groupes A et B représentent des couches différentes. Cependant, certaines applications requièrent des couches entièrement connectées avec elle-même, et l'on doit alors pouvoir implanter des poids entre neurones d'un même groupe. Ainsi, nous pouvons prévoir un mode de configuration d'une diagonale de la matrice synaptique comme celui indiqué sur la Figure III.16.

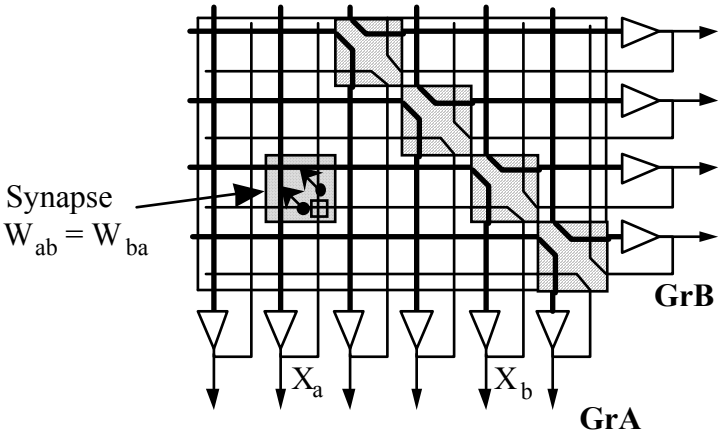


Figure III.16 : Couches entièrement connectées avec elle-même.

Dans ce cas, les deux groupes A et B sont isolés l'un de l'autre et la matrice est consacrée à des connexions à l'intérieur d'un même groupe. Ainsi, au prix d'une

augmentation assez faible de complexité, on peut implanter avec la même type de matrice synaptique des couches entièrement connectées avec elles même.

On peut remarquer aussi qu'une configuration partielle de la diagonale de la matrice permet d'avoir, sur le même circuit, des connexions des deux modes présentés ci-dessus. C'est-à-dire que l'on peut avoir des connexions entre deux couches ayant leurs propres connexions internes.

Pour pouvoir augmenter le nombre de synapses en entrée d'un neurone au-delà de ce que l'on peut obtenir sur un seul circuit, nous devons pouvoir mettre plusieurs circuits en cascade sur la plaque de circuit imprimé et obtenir ainsi une matrice plus grande. Il y a alors deux possibilités pour la construction de circuits de base :

- on peut concevoir deux circuits, l'un ne comprenant que des neurones et l'autre ne comprenant que des synapses,
- on peut concevoir un seul circuit et prévoir l'inhibition de la fonction neurone.

Dans les deux cas, les fils représentant les V_i doivent être prolongés sur la plaque de circuit imprimé. Il doit donc y avoir sur chaque circuit un plot analogique par ligne et par colonne relié au fil V_i.

Les états de neurones étant binaires, nous pouvons facilement utiliser un multiplexage temporel et un nombre réduit de plots numériques.

b- Exemple d'utilisation

Je vais présenter ici l'utilisation de la matrice sur l'exemple du réseau NETtalk décrit dans [Sej87]. Ce réseau est multicouche avec une seule couche cachée. Il est composé de 203 neurones d'entrées, 80 neurones cachés et 26 neurones de sorties. Tous les neurones cachées sont connectés à tous les neurones d'entrées et à tous les neurones de sorties. Il n'existe pas de connexions entre neurones d'une même couche. Il y a donc 18629 paramètres de poids (incluant un seuil variable par neurone).

La Figure III.17 représente ce réseau à l'aide de matrices synaptiques. On peut faire plusieurs remarques sur cet exemple :

- Les neurones d'entrée ne nécessitent pas de cellules neurones car leur état est toujours imposé par les entrées du réseau.

- Les synapses reliant les neurones d'entrée aux neurones cachés sont monodirectionnelles puisque les neurones d'entrée ne changent pas d'état.
- Il n'est pas nécessaire d'implanter de seuil pour les neurones d'entrée.
- L'implantation avec une matrice triangulaire complétement connectée introduite par Alspector est particulièrement inefficace. Elle demanderai ici une matrice 309x309 ce qui est environ 2 ordres de grandeur au dessus de ce que l'on peut faire aujourd'hui dans un seul circuit, cf [Als90].

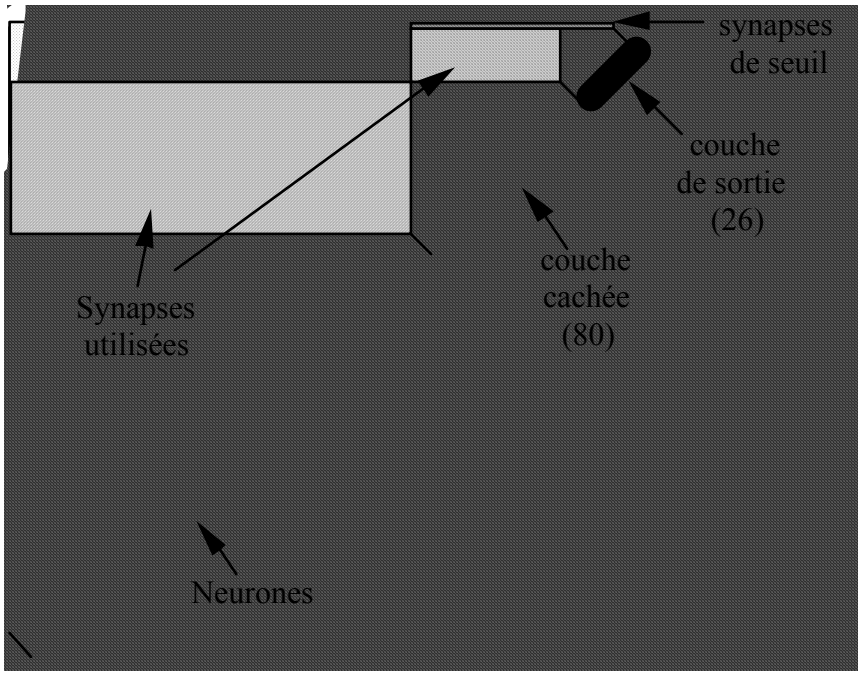


Figure III.17 : le réseau NETtalk.

c- *Circuits avec apprentissage*

Les réseaux ont souvent un nombre de neurones par couche qui diminue lorsque l'on va de l'entrée vers la sortie, comme on a pu le voir sur l'exemple ci-dessus. De ce fait la majorité des synapses est comprise dans le bloc reliant les neurones d'entrée aux neurones cachés et elles sont donc monodirectionnelles.

Afin de limiter le nombre de plots analogiques et de connexions sur la plaque de circuit imprimé, on peut concevoir deux circuits différents :

- Un circuit avec synapses monodirectionnelles et avec beaucoup d'entrées d'états de neurones. On a donc une matrice avec beaucoup plus de colonnes que de lignes.
- Un circuit avec synapses bidirectionnelles.

Cette démarche présente deux avantages :

- Elle limite l'importance des réglages de polarisation des circuits en limitant le nombre d'interconnexions analogiques entre eux.
- Elle limite la surface occupée sur la plaque de circuit imprimé par des circuits mono-dimensionnels, en limitant le nombre de plots du packaging.

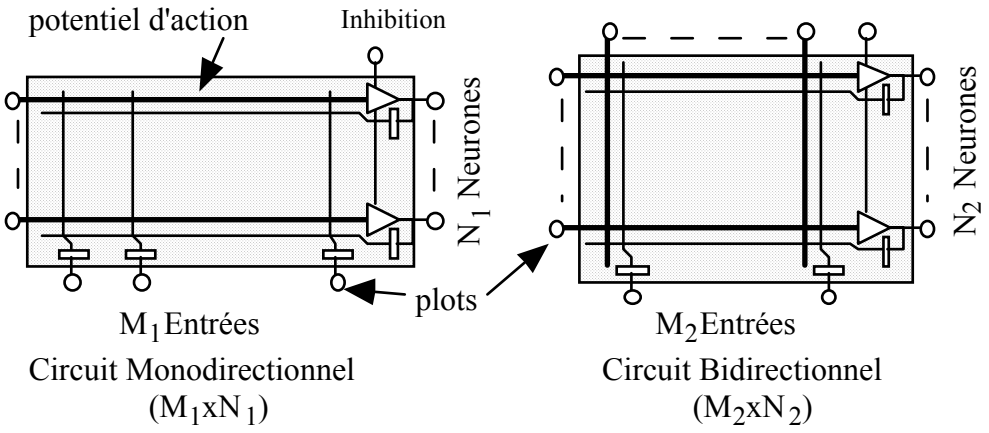


Figure III.18 : Ensemble de circuits avec apprentissage.

Supposons que les neurones associés aux lignes de la matrice sont inclus dans les circuits, mais qu'ils peuvent inhibés en cas de cascade de circuits. Supposons en outre que Z états de neurones sont multiplexés dans le temps sur le même plot.

Appelons M_1 (resp. M_2) le nombre de colonnes et N_1 (resp. N_2) le nombre de lignes du circuit mono-directionnel (resp. bidirectionnel). Ces deux circuits sont représentés sur la Figure III.18.

Le nombre de plots du circuit monodirectionnel inclut :

- N_1 plots analogiques pour les contributions V_i des N_1 lignes,
- N_1/Z plots numériques bidirectionnels pour les états de neurones des N_1 lignes (en entrée s'il y a inhibition des neurones, en sortie sinon),
- M_1/Z plots d'entrée numériques pour les M_1 états de neurones des M_1 colonnes,
- 1 entrée numérique de contrôle de l'inhibition.

Le nombre total de plots est donc, non compris les fils de contrôle:

$$P_1 = N_1 + \frac{N_1+M_1}{Z} \qquad \text{(III.11)}$$

D'après cette formule, on a intérêt à choisir M_1 grand devant N_1 pour limiter le nombre de plots. On peut par exemple choisir le paramètre M_1 suffisamment grand pour que la réalisation de la plupart des applications se fasse sans cascade de circuits pour le calcul des grandeurs V_i. Les valeurs intéressantes à ce jour sont 128, 256, 512…

Le nombre de plots du circuit bidirectionnel inclut :

- M_2+N_2 plots analogiques pour toutes les contributions V_i,
- N_2/Z plots bidirectionnels numériques pour les états de neurones des N_2 lignes (en entrée s'il y a inhibition des neurones, en sortie sinon),
- M_2/Z plots d'entrée numériques pour les états de neurones des M_2 colonnes,
- 1 entrée numérique de contrôle de l'inhibition.

Le nombre total de plots est donc, non compris les fils de contrôle:

$$P_2 = (N_2 + M_2)(1 + \frac{1}{Z}) \qquad \text{(III.12)}$$

Le nombre de plots sera la principale contrainte lors de la conception de ce circuit : la taille du circuit sera déterminée par le périmètre de l'anneau de plot (circuit "pads limited") : on a donc intérêt à minimiser N_2+M_2.

Etant donné une surface allouée aux synapses, c'est à dire un produit N_2xM_2 constant ; on minimisera N_2+M_2 lorsque N_2 et M_2 sont égaux. D'un autre coté, un nombre N_2 ne doit pas être beaucoup plus grand que le nombre de sorties de la plupart des applications pour limiter le nombre de synapses inutilisées. Les valeurs de N_2 intéressantes à ce jour sont 16, 32, 64…

Dans le cas d'un circuit ne comprenant que des synapses le nombre de plots est peu différent de celui d'un circuit incluant des neurones. Ainsi, si le circuit est "pads limited", on a intérêt à inclure les neurones et les synapses dans le même circuit et à inhiber les neurones inutilisés lors d'une cascade de plusieurs circuits. La conception de deux circuits séparés n'apporte rien dans ce cas.

d- Réalisation de NETtalk

Si on travaille par exemple avec la technologie CMOS 2,4µm que nous avons utilisé pour la réalisation de nos prototypes, on peut réaliser une matrice comprenant 2^9 Synapses.

Si on choisit Z=8 on déduit des équation (III.11) et (III.12) que :

$$\begin{cases} M_1=2^6=64 \\ N_1=2^3=8 \end{cases} \square \quad P_1 = 17 \qquad \begin{cases} M_2=2^5=32 \\ N_2=2^4=16 \end{cases} \square \quad P_2 = 54$$

Par contre, si on utilise une technologie CMOS 1,2µm largement répendues aujourd'hui, on peut réaliser une matrice comprenant 2^{11} Synapses.

Si on choisit Z=8 on déduit des équation (III.11) et (III.12) que :

$$\begin{cases} M_1=2^7=128 \\ N_1=2^4=16 \end{cases} \square \quad P_1 = 34 \qquad \begin{cases} M_2=2^6=64 \\ N_2=2^5=32 \end{cases} \square \quad P_2 = 108$$

L'exemple du réseau NETtalk présenté au paragraphe (§-b) de cette section peut alors être réalisé avec ces derniers circuits comme sur la Figure III.19. Il utilise 20 circuits mono-dimensionnels et 3 circuits bi-dimensionnels. On voit que la grande majorité des circuits sont mono-dimensionnels et qu'il est donc très important qu'ils occupent le moins de surface possible sur la plaque de circuit intégré. Les trois-quarts des fonctions neurones disponibles sont inhibées.

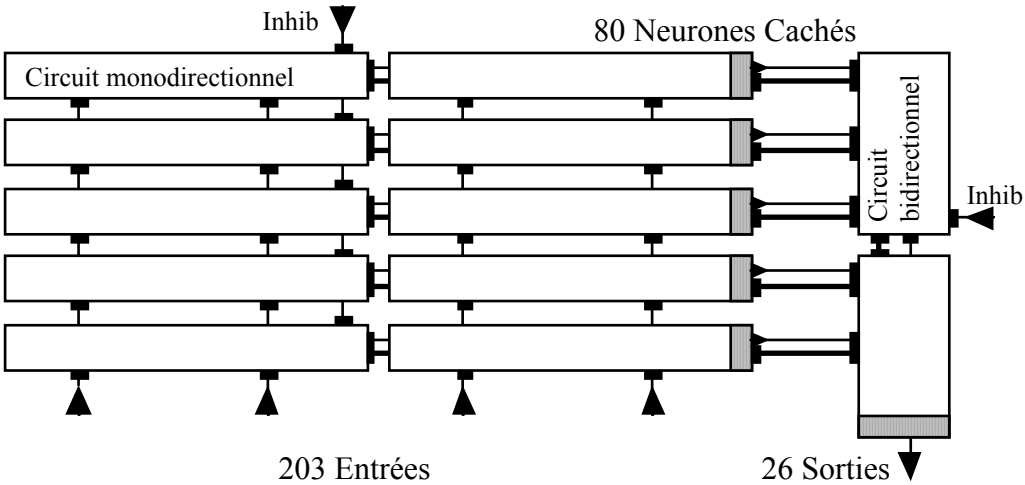


Figure III.19 : réalisation du réseau NETtalk.

III.3.3- Machine sans apprentissage

Dans le cas d'une machine dédiée à la reconnaissance, la dérive de l'apprentissage n'est plus à considérer et la robustesse de l'algorithme doit permettre d'implanter les poids W_{ij} et W_{ji} à des endroits différents. En effet, même si une légère différence existe entre ces deux poids l'algorithme de reconnaissance devrait continuer à fonctionner correctement.

Dans ce cas, on peut donc n'utiliser que des matrices synaptiques mono-directionnelles ressemblants à celles décrites ci-dessus, mais sans apprentissage, et utiliser deux synapses différentes pour réaliser les connexions bidirectionnelles. Il faut ici chercher à limiter au maximum le nombre de connexions analogiques en regroupant les synapses au plus près du neurone dont elles sont l'entrée, et donc concevoir des circuits avec beaucoup de synapses en entrée de chaque neurone.

Appelons M_3 le nombre de colonnes de la matrice, N_3 son nombre de lignes et Z le nombre d'états de neurones multiplexés dans le temps sur un seul plot.

Le calcul du nombre total de plots étant le même que dans le cas mono-dimentionnel avec apprentissage de la Figure III.18, on déduit de l'équation (III.11) que le nombre total P_3 de plots, non compris les fils de contrôle est égal à :

$$P_3 = N_3 + \frac{N_3+M_3}{Z} \qquad \text{(III.13)}$$

On peut réaliser un circuit avec M_3 suffisamment grand devant N_3 pour que la plupart des applications se construisent sans cascade de circuits pour le calcul des grandeurs V_i.

Supposons qu'une synapse sans apprentissage est 4 fois plus petite qu'une synapse avec apprentissage. Ainsi, avec la technologie 2,4µm présentée utilisée au (§-d) on peut réaliser une matrice comprenant 2^{11} Synapses.

Si on choisit Z=8 on déduit de l'équation (III.13) que :

$$\begin{cases} M_3=2^7=128 \\ N_3=2^4=16 \end{cases} \square \qquad P_3 = 34$$

La Figure III.20 représente l'architecture du circuit de base sans apprentissage. Elle présente aussi l'exemple de son utilisation pour la réalisation de l'application NETtalk avec des circuits réalisés sur la technologie 2,4µm et les valeurs de Z, M_3 et N_3 sont celles choisies ci-dessus.

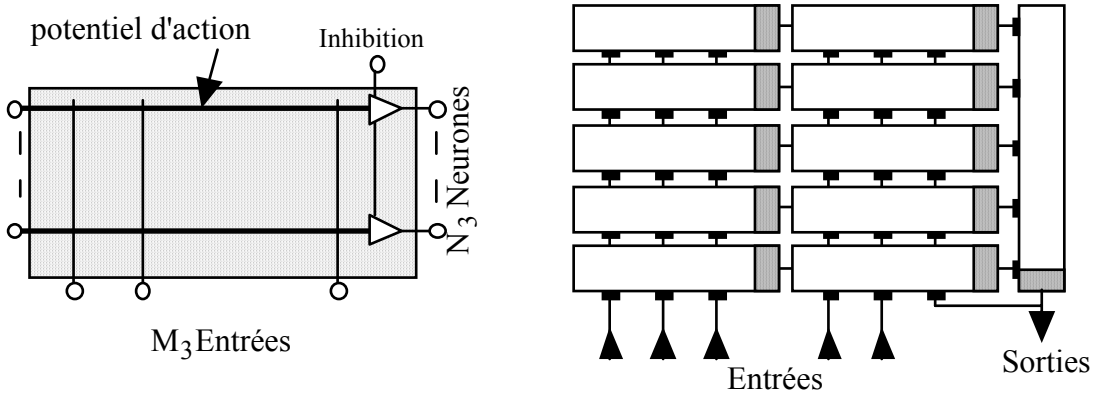


Figure III.20 : Circuits sans apprentissage + Exemple d'utilisation.

III.3.4- Système Complet sans apprentissage

Sur l'exemple de la Figure III.20, on peut voir que les interconnexions entre circuits sont uniquement numériques et qu'aucune connexion analogique n'est utilisée.

Ainsi, comme Patrick Garda l'a montré, on peut définir des modules de traitement analogiques et construire un réseau numérique pour interconnecter ces modules. Les modules de traitement peuvent être constitués d'un seul circuit, ou bien par regroupement de plusieurs circuits si l'on n'a pas accès à une technologie nous permettant de réaliser des circuits comprenant suffisamment d'entrées. Les polarisations des circuits d'un même module seront ajustées les unes par rapport aux autres pour un bon fonctionnement, chaque module restant toutefois indépendant des autres. Chaque module de traitement pourra même avoir sa propre alimentation analogique régulée ce qui limitera l'effet du bruit numérique entre modules.

Deux types d'interconnexions entre modules de traitement sont possibles :

- Ces modules peuvent être reliés par des connexions numériques au moment de la fabrication de la plaque de circuit imprimé et la Machine est alors dédiée à une application.
- On peut aussi adjoindre au module de traitement un réseau programmable d'interconnexions numérique afin de rendre la machine configurable en fonction de l'application à traiter, cf [Gar90].

Cette dernière solution est évoquée sur la Figure III.21. Le réseau d'interconnexions ne sera pas décrit en détail ici. Il est composé en adjoignant un circuit de communication, que nous appelons CIRCOM, à chaque module de traitement. Les circuits CIRCOM sont composés d'interrupteurs programmables par logiciel reliés entre eux par un bus numérique. Ils prennent aussi en compte le

chargement des états des poids dans la matrice et le rafraîchissement de ceux-ci. Pour plus de détails, on peut se référer à [Puj91a] et [Puj91b].

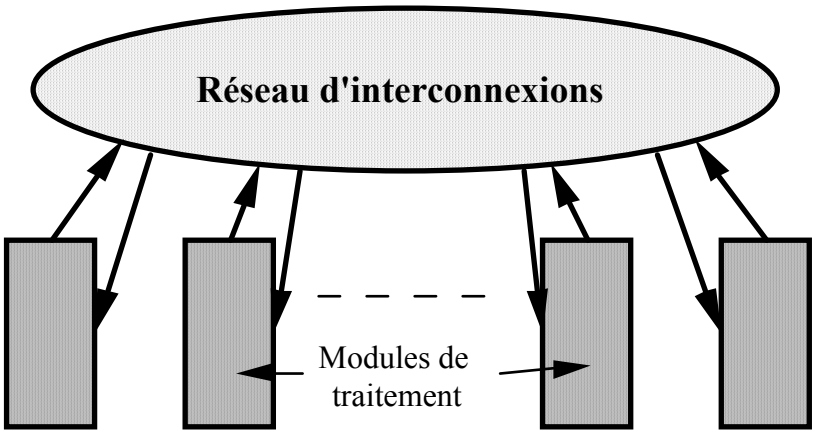


Figure III.21 : Architecture générale du système.

On peut remarquer que la démarche aboutissant à ce système a consisté à s'affranchir des problèmes de symétrie des poids en se limitant à la réalisation de problèmes de reconnaissance. La démarche aurait donc été la même si on avait considéré une machine avec apprentissage dont l'algorithme compense les erreurs de symétrie des poids. Elle sera aussi la même lorsque l'on disposera d'un algorithme avec des connexions monodirectionnelles et le système présenté sera alors un Réseau de Neurone Formel analogique et entièrement configurable par logiciel : *Un ordinateur analogique*.

III.4- CONCLUSION

J'ai exposé, dans ce chapitre, nos choix architecturaux en particulier ceux de générateur aléatoire et de sommateur mono-rail. J'ai ensuite décrit fonctionnellement les cellules neurones et synapse (I.3) et (I.5). J'ai ensuite exposé l'architecture générale d'une machine de Boltzmann avec apprentissage et j'ai défini l'architecture de 2 circuits différents pouvant être interconnectés sur une plaque de circuit imprimé pour réaliser cette machine. J'ai ensuite démontré que si on peut s'affranchir des contraintes de symétrie des poids, on peut alors profiter des états binaires de la Machine de Boltzmann pour réaliser une machine dont la configuration est entièrement programmable par logiciel.

CHAPITRE IV

CONCEPTION ET SIMULATION

IV.1- INTRODUCTION

L'architecture générale de la machine de Boltzmann est décrite dans le chapitre III à l'aide des blocs fonctionnels. Dans ce chapitre IV, je détaillerai la conception des cellules composant ces blocs.

Une fois l'architecture générale des circuits décrite, plusieurs choix de conception restent possibles. Nous avons ainsi décidé d'utiliser en grande partie les techniques analogiques et de réaliser nos circuits sur une technologie analogique CMOS d'un fondeur d'ASIC (Circuits Intégrés dédiés et en petites séries). La technologie CMOS présente l'avantage de permettre la mémorisation d'une tension sur les grilles de commande isolées des transistors. L'analogique, elle, présente l'avantage d'une puissance de calcul par unité de surface beaucoup plus grande que le numérique.

A ce stade, nous avons à choisir le point de polarisation des transistors. Nous devons faire ici un compromis entre la consommation et la taille du circuit d'une part, et la vitesse et la précision du circuit d'autre part. Cependant il ne faut pas perdre de vue que ce choix aura des répercutions sur la difficulté de conception.

LA conception d'un circuit analogique réside dans cette complexité de choix du point de polarisation.

Cependant, deux options sont possibles au début de la conception :

- Utilisation des transistors dans leur région de forte inversion,
- Utilisation des transistors dans leur région de faible inversion (appelée aussi région sous le seuil).

J'ai choisi d'utiliser les transistors en forte inversion car ils permettent :

- de faciliter la conception,
- de profiter de meilleurs simulateurs électriques,

- d'obtenir des caractéristiques de transfert de plus grande linéarité (quadratique contre exponentielle).
- de limiter la dispersion des caractéristiques électriques des dispositifs.

Je présenterai dans ce chapitre la conception du convertisseur tension-courant qui constitue le cœur de la cellule synapse présentée au paragraphe (§ III.2.5), puis celle des éléments constituant la cellule neurone A (§ III.2.3). Ensuite, je décrirai la conception des cellules nécessaires à l'apprentissage (compteurs, intégrateur,…) puis celle du circuit optoélectronique d'acquisition de l'image du speckle. Enfin, je présenterai les deux circuits de grande dimension que j'ai réalisé et qui sont les éléments de base d'une machine de Boltzmann analogique sans apprentissage.

La conception de la cellule neurone B ne sera pas détaillée ici. Cependant, ces éléments de base seront aisément conçu à partir de ceux présentés pour la réalisation de la cellule neurone A.

IV.2- CONVERTISSEUR TENSION-COURANT : CONVVI

IV.2.1- Introduction

Le convertisseur tension-courant est l'élément de base de la cellule synapse décrite sur la Figure III.13. Sa fonction de transfert est décrite par l'équation (III.8) :

$$I_{ij} = W_{ij}.X_j \qquad \text{(III.8)}$$

où I_{ij} est le courant de sortie du convertisseur, W_{ij} est une tension analogique représentant le poids et X_j une valeur binaire numérique représentant l'état d'un neurone.

J'ai cherché à "linéariser" cette fonction de transfert pour pouvoir respecter le modèle mathématique et ainsi avoir quelques garanties sur le comportement de la Machine. Il faut en particulier garantir la convergence de son apprentissage et de sa relaxation. Cependant, de petites variations par rapport au modèle sont possibles et on utilise, par exemple, couramment dans les simulations des poids à valeur saturée. L'équation (III.8) est alors remplacée par la caractéristique de transfert de la Figure (IV.1).

Ce type de caractéristique est très souvent utilisé dans les apprentissages de réseaux de neurones. Elle permet de limiter l'influence d'un neurone sur la contribution du réseau totale d'un autre neurone à une valeur maximale. La

perturbation occasionnée par le mauvais fonctionnement d'une cellule synapse se trouve ainsi limitée et ne met pas en péril le comportement global du RNF [Mea89b]. Elle confère une certaine "robustesse" à celui-ci.

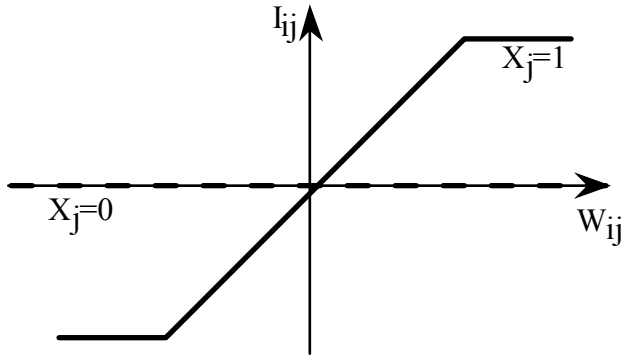


Figure IV.1 : caractéristique de transfert

IV.2.2- Description du fonctionnement

Ce convertisseur est essentiellement constitué par un amplificateur opérationnel à transconductance (OTA) "linéarisé" par dégénérescence de sources des transistors de la paire différentielle [Bel91]. L'entrée inverseuse est maintenue à une tension de référence constante, voir Figure (IV.2).

L'étage de sortie est doté de deux transistors d'inhibition N_{13} et N_{14} faisant fonction d'interrupteur et commandés par la valeur logique X_j. Ainsi, lorsque X_j est nul le courant de sortie I_{ij} est limité uniquement à des courants de fuite drain-substrat négligeables et il peut donc être considéré comme nul.

Lorsque X_j vaut "1", l'étage de sortie calcule la différence de courant (I_1-I_2) dans la paire différentielle et le courant de sortie I_{ij} est donc positif ou négatif selon le signe de W_{ij} par rapport à la tension de référence. Il est proportionnel à la différence entre la tension d'entrée et la masse.

La caractéristique de ce convertisseur se rapproche de la caractéristique idéale présentée sur la Figure (IV.1). L'utilisation qui sera faite de cet amplificateur à transconductance pour le calcul des contributions V_i d'un réseau de neurones nécessite une conception relativement différente des conceptions habituelles.

Tout d'abord, nous aurons ici un très grand nombre de ces amplificateurs dont les sorties seront interconnectées (typiquement 256). De ce fait l'encombrement de chaque convertisseur et son courant de sortie doivent être faibles. Cependant, comme cela a déjà été dit, nous avons choisi de travailler en forte inversion, et la recherche d'un compromis m'a amené à utiliser des courants de sortie compris entre

-1μA et +1μA pour chaque convertisseur. Les tensions de grilles sont alors supérieures aux tensions de seuils d'environ 300 mV.

Par exemple, le courant total consommé par un système contenant 100.000 synapses reste inférieur à 1 Ampère et il est donc tout à fait possible de le réaliser.

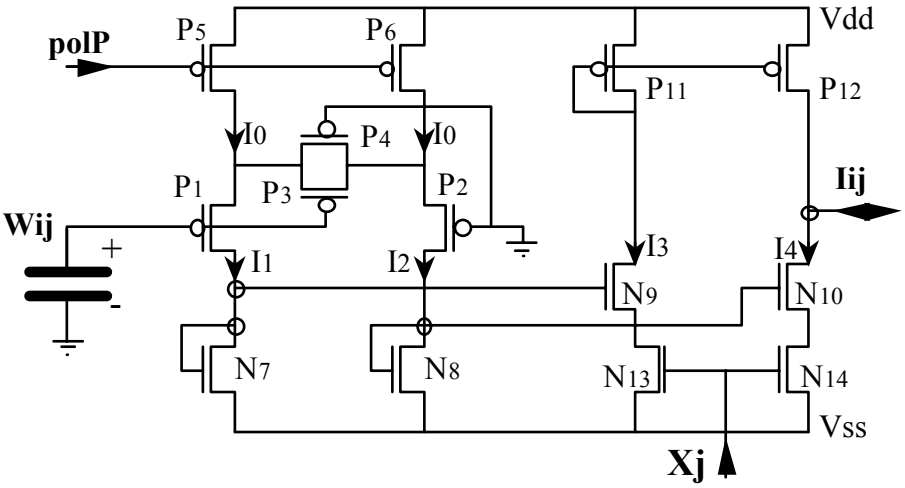


Figure IV.2 : convertisseur tension-courant

IV.2.3- Technique de "linéarisation"

La "linéarisation" de la caractéristique entrée-sortie est obtenue par une dégénérescence de la source des transistors d'entrées de la paire différentielle. Cette technique a été introduite dans [Kru88].

Pour l'étude théorique qui suit, nous supposerons que le rapport des miroirs de courant (N_7:N_9), (N_8:N_{10}) et (P_{11}:P_{12}) est égal à 1. La tension polP est telle que les transistors P_5 et P_6 sont contrôlés par une tension de 300 mV au dessous de leur tension de seuil et ils peuvent donc être considérés comme des sources de courant. Le courant de sortie est alors égal au courant différentiel et la détermination de la caractéristique statique du convertisseur se ramène à l'étude de l'association des quatres transistors P_1, P_2, P_3 et P_4.

Les résultats de cette étude ont été publiés dans [Kru88] et utilisés dans [Kru89] pour une application. J'ai moi-même repris cette étude et je suis arrivé au même résultats en utilisant le modèle quadratique du transistor MOS décrit par le Tableau IV.1. Je n'ai pas réussi à dériver une expression plus précise avec un modèle plus complet du transistor.

Je vais maintenant présenter les résultats de cette étude en précisant tout d'abord les notations utilisées :

$$ß = \mu . C_{ox} . \frac{W}{L} \qquad \text{(IV.1)}$$

μ est la mobilité des porteurs dans le canal du transistor,
C_{ox} est la capacité d'oxyde par unité de surface,
V_{T0} est la tension de seuil,
W est la largeur du transistor et L sa longueur.

Région	Conditions	Equations
Triode ou de conduction	**Erreur !**	$i_d = ß.\left[v_{gs} - V_{T0} - \frac{v_{ds}}{2}\right] .v_{ds}$
Saturation ou source de courant	**Erreur !**	$i_d = \frac{ß}{2} .\left[v_{gs} - V_{T0}\right]^2$

Tableau IV.1 : Equations Simplifiées du Transistor MOS.

Notons :

- I_{out} le courant I_1-I_2,
- V_{in} la tension d'entrée différentielle,
- g_m la transconductance ($\partial I_{out}/\partial V_{in}$) du convertisseur,
- $ß_1$ le paramètre ß des transistors P_1 et P_2,
- $ß_3$ le paramètre ß des transistors P_3 et P_4.

Pour de faibles valeurs de la tension d'entrée différentielle V_{in}, les transistors P_3 et P_4 fonctionnent dans leur zone triode et la transconductance de la paire différentielle est similaire à celle d'une paire classique avec sources couplées. Le paramètre g_m diminue donc avec $|V_{in}|$, et en utilisant les équations quadratiques simplifiées du transistor MOS nous obtenons :

$$\frac{I_{out}}{2I_0} = V.\sqrt{1-\frac{V^2}{4}} \qquad \text{(IV.2)}$$

avec

$$\begin{cases} V = \frac{V_{in}}{a} . \sqrt{\frac{ß_1}{2.I_0}} \\ a = 1 + \frac{ß_1}{4ß_3} \end{cases} \qquad \text{(IV.3)}$$

Par ailleurs, lorsque la tension d'entrée augmente (resp. diminue), le transistor P_3 (resp. P_4) entre en saturation et la transconductance g_m peut augmenter à nouveau. On peut montrer de même que l'un des transistors P_3 ou P_4 est dans sa zone de saturation pour :

$$|V| > \sqrt{\frac{1}{a^2 - a + 0,5}} \qquad \text{(IV.4)}$$

Les équations du courant I_{out} en fonction de V sont alors :

$$\frac{I_{out}}{2I_0} = \pm \frac{\left[aV.\sqrt{4a - 2} \pm \sqrt{4a - 1 - a^2V^2}\,\right]^2}{(4a - 1)^2} \qquad \text{(IV.5)}$$

La Figure IV.3 représente la caractéristique de transfert du convertisseur. Elle a été déterminée d'après les équations ci-dessus pour une valeur du paramètre "a" égale à 3. Elle est comparée à la caractéristique obtenue sans compensation de linéarité ($ß_3$ infini et paramètre "a" égal à 1). Le courant de polarisation I_0 est de 2µA et le paramètre $ß_1$ de 23,4 µA.V^{-2}.

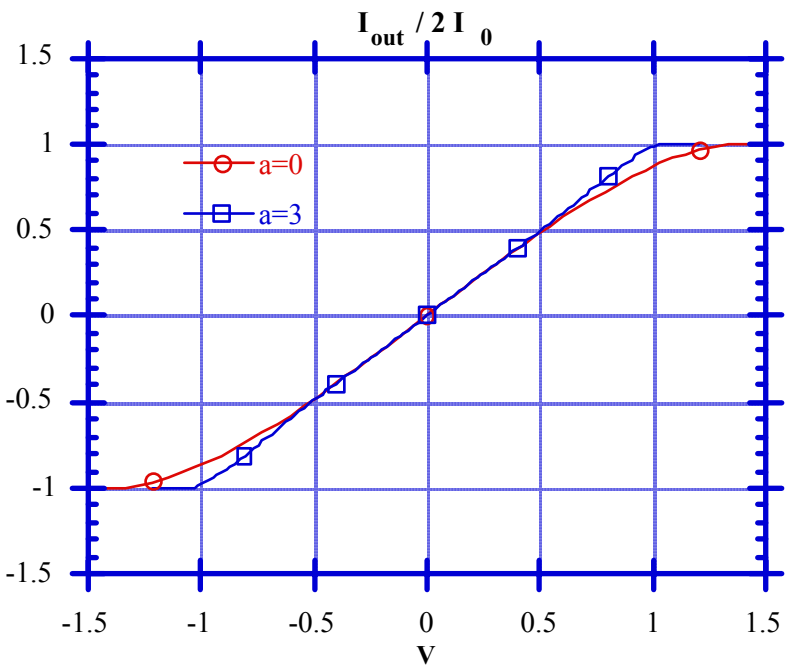


Figure IV.3 : caractéristique de transfert du convertisseur

D'après les équations ci-dessus, on peut déterminer aisément le paramètre g_m en fonction de a, ainsi que les erreurs absolue et relative entre le courant de sortie

I_{out} et la caractéristique de courant idéale. Ces différentes quantités ont été représentée sur les Figures (IV.4), (IV.5) et (IV.6) en fonction de V, pour différentes valeurs de a.

Sur la Figure (IV.5), nous voyons que l'erreur absolue Δi est inférieure à 1% de l'amplitude maximale lorsque a vaut $(1+\frac{ß_1}{4.ß_3})=3$. Sur la Figure (IV.6), nous voyons qu'elle est inférieure à 3% de l'amplitude dans ce cas a=3.

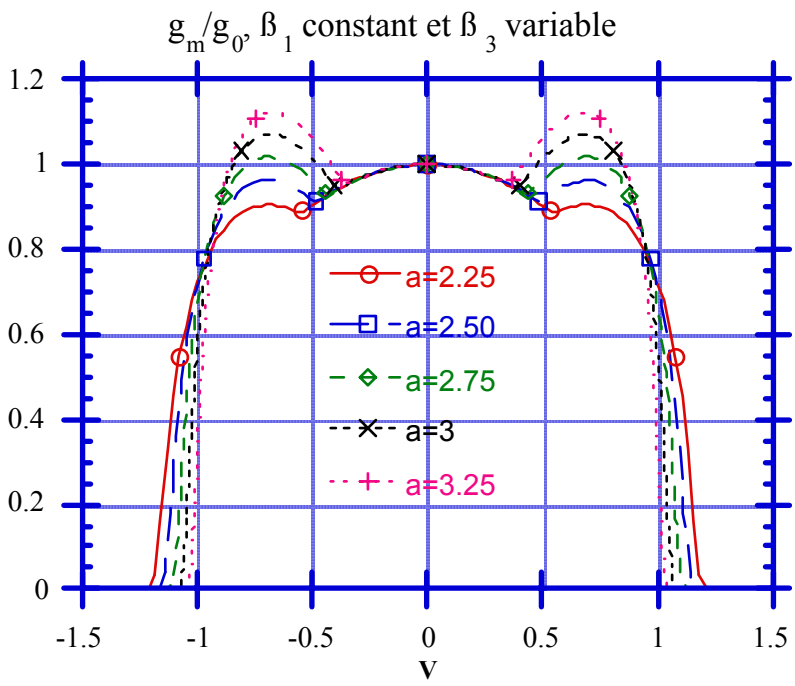


Figure IV.4 : transconductance g_m

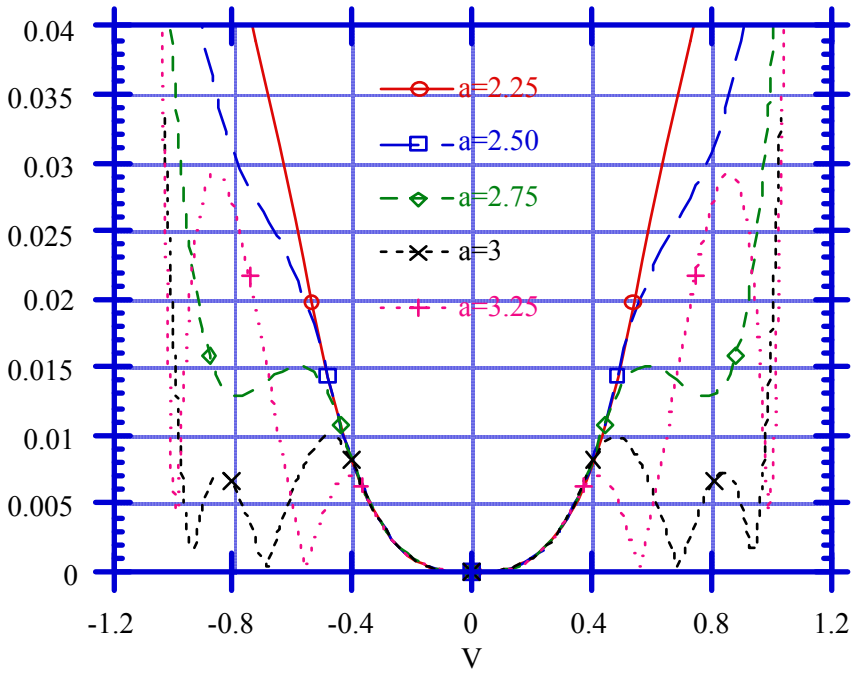


Figure IV.5 : Erreur absolue sur le courant.

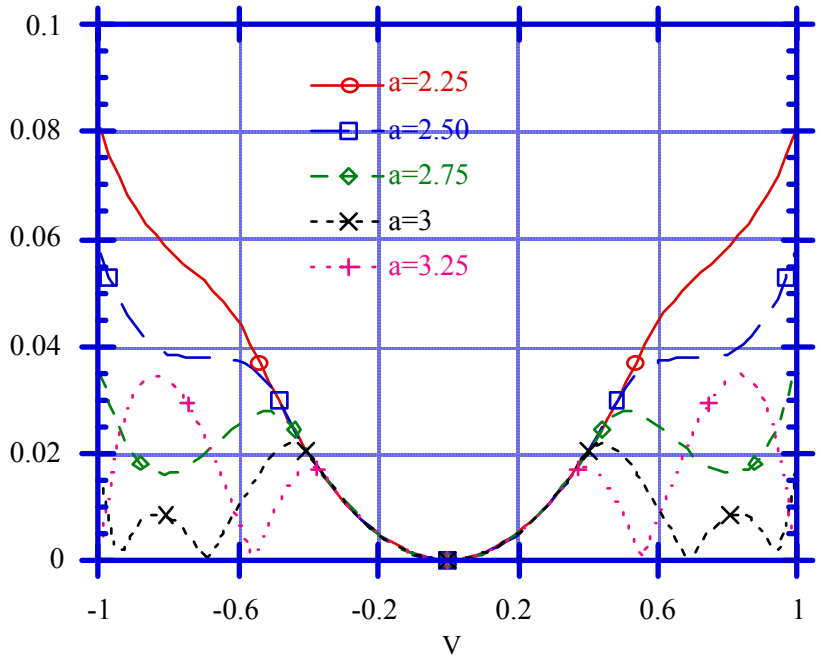


Figure IV.6 : Erreur relative sur le courant de sortie.

La valeur de a est directement liée au rapport des tailles des transistors P_1 et P_3 par l'équation (IV.3). Si ce rapport est un nombre entier, il peut être réalisé avec une grande précision en dupliquant plusieurs fois un transistor de taille élémentaire.

Le Tableau (IV.2) présente le rapport entre les tailles des transistors P_1 et P_3 pour les valeurs de a utilisées sur les Figures (IV.5) et (IV.6).

a	$\beta_1/\beta_3 = \frac{W_1/L_1}{W_3/L_3}$
2,25	5
2,50	6
2,75	7
3	8
3,25	9

Tableau IV.2 : β_1/β_3 en fonction de a.

Cependant, le modèle simplifié du transistor MOS, utilisé afin d'obtenir les équations ci-dessus, ne tient pas compte des effets du second ordre et en particulier de l'effet de modulation de la tension de seuil par la tension canal-substrat. Pourtant sa prise en compte est ici nécessaire.

En effet, j'ai choisi de réaliser la paire différentielle avec des transistors canal-p et la variation de la tension de seuil n'est pas négligeable dans ce cas, (voir la Figure IV.7). Nous travaillons avec une technologie n-well et le caisson est faiblement dopé.

J'ai malgré tout fait ce choix pour profiter de leur mobilité plus faible que celle des transistors canal-n. On obtient ainsi une plage d'entrée plus grande à courant et à taille donnés.

Cependant, dans ce cas, les imperfections du modèle théorique et la difficulté à dériver un modèle plus précis, nous obligent à confirmer le choix du paramètre "a" par une simulation électrique.

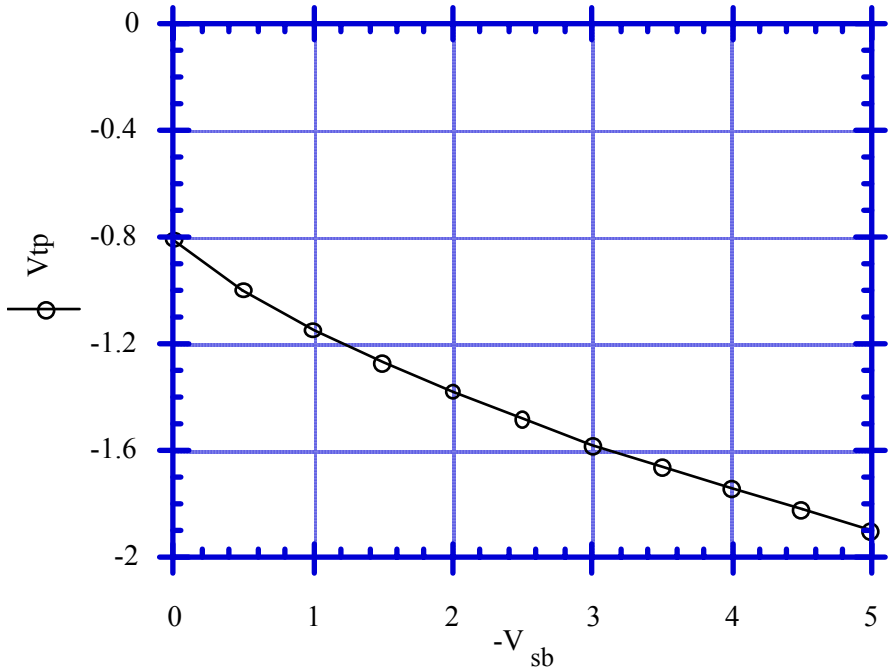


Figure IV.7 : Tension de seuil des transistors canal-p en fonction la tension canal-substrat.

IV.2.4- Simulation

Ce paragraphe présente quelques résultats de simulation électrique obtenus en utilisant le simulateur Hspice™ de Metasoftware®, le modèle 3 du transistor MOS et les paramètres fournis par le fondeur (technologie DLP 3µm par Mietec Alcatel®).

Deux courbes issues de ces simulations sont présentées ici :

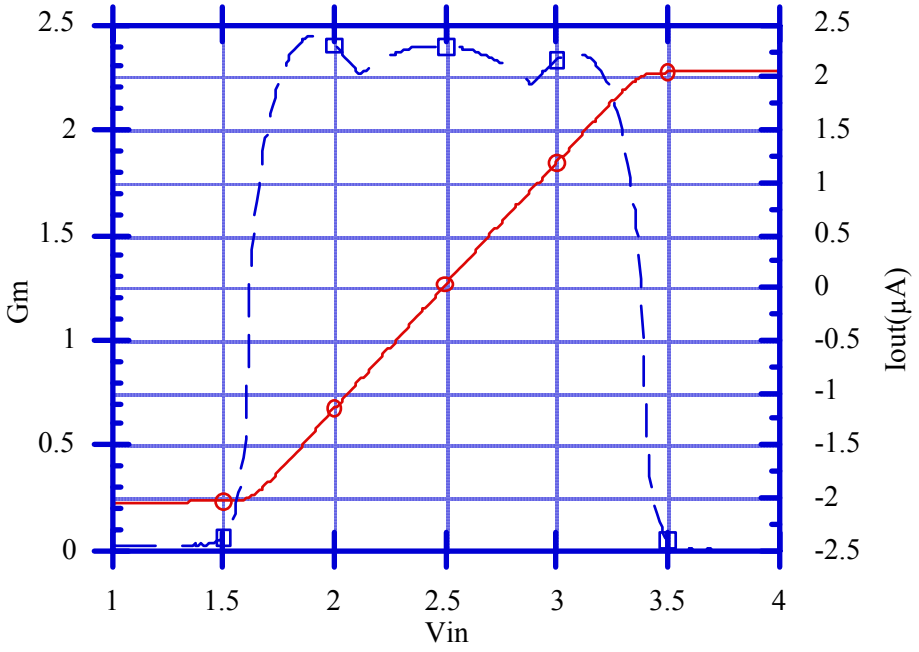


Figure IV.8 : Simulation électrique de convvi (a=2,5).

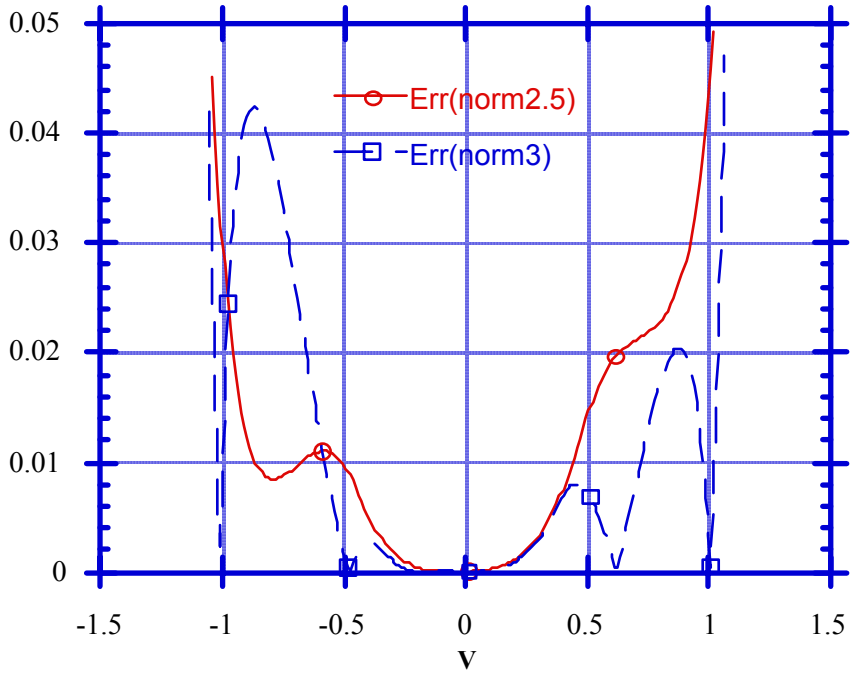


Figure IV.9 : Simulation électrique : Erreur absolue (a=2,5 ; a=3).

- Tout d'abord, la Figure (IV.8) présente la caractéristique de transfert, c'est à dire le courant de sortie I_{out} en fonction de la tension d'entrée. La valeur de a est

égale à 2,5. Le paramètre g_m est déterminé par dérivation de cette caractéristique et il est tracé sur la même Figure.

- La Figure IV.9 représente l'erreur absolue de linéarité pour les valeurs de a égales à 2,5 et 3. On peut observer sur cette Figure que l'erreur absolue est de l'ordre de 2% sur la plage de tension d'entrée [-1V, +1V] pour a=2,5.

La dissymétrie en fonction du signe de V_{in} observable sur le paramètre g_m est due à l'effet de modulation de la tension de seuil par la tension canal-substrat. En effet, la tension différentielle de mode commun est fortement variable entre une entrée différentielle de signe négatif et une entrée de signe positif puisque le potentiel de l'entrée inverseuse est maintenu à une tension de référence fixe.

Un ensemble de simulations électriques, obtenu avec les paramètres de la technologie Mietec 3µm, a montré :

- que la compensation de linéarité est trop forte dans le cas où a=3,
- et que la valeur optimale de a est plus proche de 2,5 .

IV.2.5- Conclusion

La modélisation théorique, basée sur le modèle quadratique simplifié du transistor MOS, a permis de fixer un ordre de grandeur du rapport $ß_1/ß_3$ et de déterminer une valeur idéale du rapport "a" d'environ 3. Les simulations électriques ont montré qu'une valeur du rapport "a" de 2,5 est meilleure pour la technologie Mietec 3µm.

Cependant, j'ai fait un grand nombre de simulations en utilisant les paramètres de trois technologies différentes et en utilisant des longueurs de transistors différentes, il s'est avéré que le paramètre "a" optimal était toujours de l'ordre de 2,5 ou 3.

Il est nécessaire, dans tous les cas, d'affiner l'étude théorique à l'aide de simulations prenant en compte les effets de modulation de tension de seuil et de longueur de canal.

Il faut noter que les caissons n-well sont ici tous reliés à l'alimentation afin de diminuer la taille de la cellule réalisant ce convertisseur.

Pour la réalisation de la charge active et l'extraction de la différence de courant (transistors N7, N8, N9, N10, P11, P12), j'ai adopté une structure symétrique afin de limiter l'offset du convertisseur. Les transistors de l'étage de sortie sont

suffisamment longs pour éviter un recours à un montage cascode diminuant la résistance de sortie du convertisseur.

IV.3- CONVERSION COURANT-TENSION

IV.3.1- Introduction

Le convertisseur courant-tension et la sigmoïde sont deux éléments de base de la cellule neurone A (voir Figure III.11), et leurs conceptions respectives sont étroitement liées. En effet, la gamme de tension de sortie du convertisseur doit être, non seulement adaptée à celle d'entrée de la sigmoïde, mais aussi à la gamme de température implantée dans le système, ainsi qu'au nombre de synapses connectées en entrée du convertisseur.

La gamme de température qu'il faut réaliser est en fait mal connue à ce jour et il est difficile de déceler dans les publications des mathématiciens quelle est la meilleure façon de procéder. Nous sommes donc ici en face d'un problème délicat, d'autant plus que l'on voudrait construire une machine suffisamment générale pour qu'elle puisse être utilisée sur une grande gamme d'applications.

De nos lectures et de nos discussions avec les mathématiciens de l'équipe du Professeur Azencott, nous avons toutefois pu déduire une gamme de température intéressante pour beaucoup d'applications :

> Pour 128 synapses en entrée d'un neurone, avec des poids codés sur la plage [-1, +1], une gamme de température intéressante est comprise entre 0,5 et 8.

Je vais maintenant décrire en termes de grandeurs physiques (courant, tension…) les conséquences de ces choix sur notre conception.

Supposons que :

- la Machine soit conçue pour avoir **128 poids** en entrée de chaque neurone.
- Les synapses transcodent les **poids** sous forme de courant.
- chaque poids est représenté en sortie des synapses par un courant dans l'intervalle **[-1µA, +1µA]**.

Appelons V_{pat} la tension d'entrée de la sigmoïde. Cette tension représente un codage de la valeur de V_i^n divisé par la température T (cf § III.2.3). La caractéristique de transfert de la sigmoïde a été décrite par l'équation (III.2). Compte tenu des

notations introduite ici, celle-ci peut être maintenant réécrite sous la forme de l'équation (IV.6) suivante:

$$S_i = \frac{K_{Si}}{\left[1 + \exp\left(-K_{pat}\, V_{pat}\right)\right]} - K_{S0} \qquad \text{(IV.6)}$$

Où K_{pat} est le gain du codage en tension de la grandeur V_i^n /T et s'exprime en V^{-1}. K_{pat} est déterminé par la réalisation physique de la sigmoïde.

Le type de la grandeur S_i de sortie de la sigmoïde, ainsi que son gain de codage K_{Si} et son offset K_{S0}, n'ont pas besoin d'être précisés pour l'instant.

Sachant que :

$$V_i^n = \frac{1}{I_M} \sum_j I_{ij} \qquad \text{(IV.7)}$$

On peut écrire :

$$K_{pat} \cdot V_{pat} = \frac{1}{T} \cdot \frac{1}{I_M} \cdot \sum_j I_{ij} \qquad \text{(IV.8)}$$

Où I_M est le paramètre de normalisation utilisé pour ramener les poids dans l'intervalle [-1,+1]. I_M est donc le courant maximum sur chaque synapse, et vaut ici 1µA.

Appelons R_{ct}, le rapport de transformation du convertisseur courant-tension. On a donc :

$$V_{pat} = R_{ct} \cdot \sum_j I_{ij} \qquad \text{(IV.9)}$$

Soit finalement :

$$\boxed{T = \frac{1}{R_{ct} \cdot K_{pat} \cdot I_M}} \qquad \text{(IV.10)}$$

Le codage de la tension V_{pat} peut se faire par une tension différentielle ou bien par une tension référencée par rapport à une tension fixe notée V_{ref}. La référence peut-être par exemple égale à la moitié de la tension d'alimentation.

Supposons maintenant que la Machine soit entièrement construite, et que l'on désire l'utiliser avec un nombre de synapses très inférieur à 128 en entrée des neurones : nous pouvons alors affecter plusieurs cellules synapses à chaque poids afin de maintenir une gamme de courant à peu près constante en entrée du neurone.

IV.3.2- Convertisseur simple

a- Introduction

Un convertisseur courant-tension très simple qui ne prend pas en compte la variation de la température T a un gain de conversion fixe. Sa caractéristique minimum est d'avoir une résistance d'entrée faible pour assurer aux synapses un fonctionnement à tension de sortie constante.

Un convertisseur remplissant ce cahier des charges et fonctionnant en classe AB est présenté ici.

b- Convertisseur à deux transistors

Le convertisseur courant-tension linéaire le plus simple est présenté sur la Figure IV.10. Il n'est constitué que deux transistors de taille identique avec grille et drain reliés. Son fonctionnement a été détaillé dans [Wan90a] et ses équations sont reportées ci-dessous.

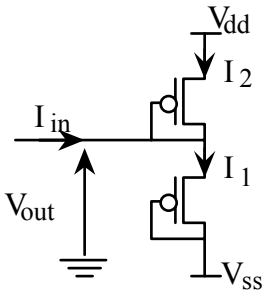


Figure IV.10 : Convertisseur courant-tension

on peut écrire :

$$\begin{cases} I_1 = ß.(V_{out}-V_{ss}-V_t)^2 \\ I_2 = ß.(V_{dd}-V_{out}-V_t)^2 \\ I_{in} = I_1 - I_2 \end{cases}$$

d'où : $$V_{out} = \frac{V_{dd} + V_{ss}}{2} + \frac{I_{in}}{2.ß.(V_{dd} - V_{ss} - 2.V_t)}$$

Les équations ci-dessus sont obtenues à partir du modèle quadratique simplifié du transistors MOS et elles supposent en particulier que la tension seuil du transistor soit constante.

Pour avoir une résistance linéaire sur une technologie n-well, il faut donc s'affranchir de l'effet de substrat (modulation de la tension de seuil par la tension canal-substrat) en utilisant des caissons-n différents pour les deux transistors. Si on utilise une technologie p-well, on adoptera la structure symétrique utilisant des transistors canal-n.

L'amplitude de la tension de sortie doit être adaptée à celle d'entrée de la sigmoïde et donc, comme on le verra, limitée à quelques centaines de mV.

Pour obtenir cette amplitude de sortie à partir d'un courant d'entrée d'une centaine de µA, il faudrait des transistors très larges (ß faible). Cependant, dans ce cas, le courant consommé serait très grand (plusieurs mA par convertisseur), ce qui n'est pas admissible pour notre application.

Il est donc nécessaire de réduire le courant d'entrée avant d'utiliser ce système de conversion.

c- Convertisseur classe AB

Le système de conversion courant-tension complet est représenté sur la Figure IV.11. Sa tension de sortie est différentielle et il est construit à partir d'un réducteur de courant et de deux convertisseurs courant-tension comme ceux décrits ci-dessus.

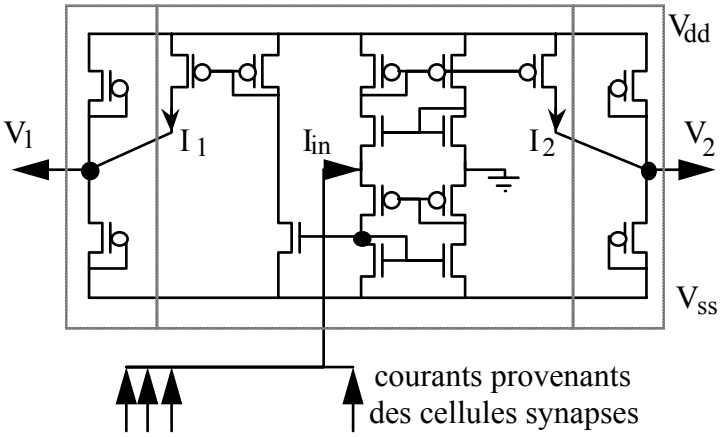


Figure IV.11 : convertisseur courant-tension simple.

Le réducteur de courant fonctionne en classe AB et il est inspiré d'un amplificateur de courant décrit dans [Wan90b]. Sa résistance d'entrée très faible permet de maintenir sa tension d'entrée constante et ainsi d'assurer un bon fonctionnement des cellules synapses.

Les deux convertisseurs courant-tension sont formés respectivement par les deux transistors les plus à gauche et par les deux transistors les plus à droite de la Figure IV.11. Le réducteur de courant étant composé des transistors se trouvant à l'intérieur du pointillé au centre de la Figure.

Figure IV.12 Simulation électrique du convertisseur simple.

Si le courant d'entrée I_{in} est nul, le courant de polarisation dans les branches du réducteur est très faible. Les courants d'entrées des deux convertisseurs, I_1 et I_2, sont alors nuls, et :

$$V_1 = V_2 = \frac{V_{dd} + V_{ss}}{2}$$

Maintenant, si le courant I_{in} devient positif, le courant dans la partie inférieure du réducteur augmente alors que celui dans la partie supérieure reste nul.

On a donc en résumé :

$$I_{in} > 0 \square \begin{cases} I_1 > 0 \quad \text{et} \quad I_2 = 0 \\ V_1 > V_2 = \dfrac{V_{dd} + V_{ss}}{2} \end{cases}$$

$$I_{in} < 0 \square \begin{cases} I_1 = 0 \quad \text{et} \quad I_2 > 0 \\ V_1 = \dfrac{V_{dd} + V_{ss}}{2} > V_2 \end{cases}$$

Une simulation électrique de ce convertisseur, effectuée avec le simulateur Hspice, est représentée sur la Figure IV.12. Elle montre la tension différentielle (V_1-V_2) en fonction du courant d'entrée.

IV.3.3- Convertisseur et Variation de Température

Un convertisseur permettant de faire varier la température est décrit sur la Figure IV.13. Son principe est classique et il utilise un amplificateur opérationnel rebouclé par une résistance variable. La valeur de cette résistance est égale au rapport de transformation R_{ct} du convertisseur d'après l'équation (IV.9). La résistance est ici réalisée avec des transistors MOS contrôlés par une tension analogique notée R_c.

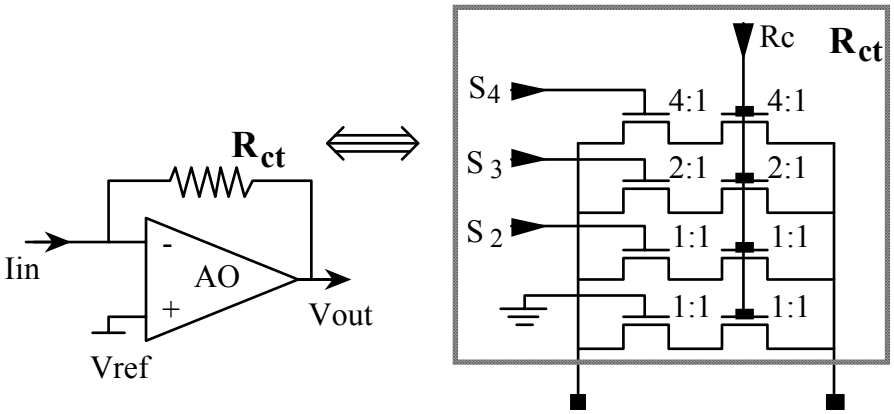


Figure IV.13 convertisseur complet.

Les différentes températures possibles sont sélectionnées, soit en faisant varier la tension R_c pour les petites variations, soit en mettant en parallèle plusieurs branches de transistors à l'aide des interrupteurs commandés par les signaux numériques S_2, S_3 et S_4.

Les tailles relatives des transistors des différentes branches sont dans un rapport de 2 et elles sont de plus en plus larges pour les branches de S_2 à S_4.

Supposons que R_c soit fixe et corresponde à un rapport R_{ct0} et à une température T_0 lorsque S_2=S_3=S_4=0, d'après l'équation (IV.10). On obtient alors les températures indiquées sur le Tableau IV.3 par la sélection des interrupteurs :

S_2	S_3	S_4	Rapport	Température
0	0	0	R_{ct0}	T_0
1	0	0	$R_{ct0}/2$	$2.T_0$
1	1	0	$R_{ct0}/4$	$4.T_0$
1	1	1	$R_{ct0}/8$	$8.T_0$

Tableau IV.3 : T en fonction de S_1,S_2 et S_3.

L'amplitude de la tension de sortie étant limitée à quelques centaine de mV, les transistors commandés fonctionnent dans leur zone de conduction et la linéarité de cette résistance variable sera sûrement suffisante. Cependant, cette linéarité peut aisément être améliorée en utilisant les techniques développées pour la réalisation des filtres à temps continu CMOS (voir par exemple [Tsi85] et [Tsi86]).

En conclusion, on dispose pour le contrôle de la température T d'un signal analogique et de plusieurs signaux numériques.

IV.4- CONCEPTION DE LA SIGMOÏDE ET DU COMPARATEUR

IV.4.1- Introduction

Dans cette section, je vais montrer, tout d'abord, que la fonction sigmoïdale peut être simplement construite par une paire différentielle à transistors bipolaires. Je présenterai ensuite comment ces transistors peuvent être réalisés sur une technologie CMOS. Puis, je montrerai comment on peut construire le bloc de comparaison. Et enfin, je présenterai comment ces différents éléments aboutissent à la conception du Neurone A complet.

IV.4.2- Sigmoïde

On a vu au paragraphe (§ IV.3.1), que l'équation de la caractéristique de la sigmoïde doit être :

$$S_i = \frac{K_{Si}}{\left[\,1 + \exp\left(-K_{pat}\,V_{pat}\right)\right]} - K_{S0} \qquad \text{(IV.6)}$$

Cette caractéristique est réalisable par l'utilisation de la fonction de transfert d'une paire différentielle de transistors bipolaires. En effet considérons la paire différentielle à transistors canal-p représentée sur la Figure IV.14.

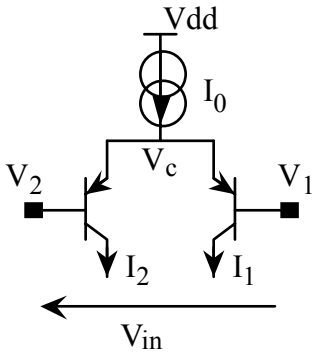


Figure IV.14 : paire différentielle

on peut écrire :

$$\begin{cases} I_1 = I_s \,.\, \exp\left(-\dfrac{V_1 - V_c}{U_T}\right) \\ I_2 = I_s \,.\, \exp\left(-\dfrac{V_2 - V_c}{U_T}\right) \\ I_0 = I_1 + I_2 \end{cases}$$

avec : $U_T = \dfrac{kT}{q} \approx 26$ mV à 300°K

I_s est une constante caractéristique du transistor.

Choisissons le courant différentiel $\Delta I = (I_1 - I_2)$ pour représenter la grandeur de sortie S_i de la sigmoïde. La grandeur S_i est comprise entre 0 et 1, alors que la différence de courant ΔI est comprise dans l'intervalle $[-I_0 , -I_0]$.

Afin de déterminer les paramètres K_{S0} et K_{S1} de l'équation (IV.6), on peut écrire :

$$\begin{cases} S_i = \Delta I = I_1 - I_2 \\ K_{S0} = I_0 \end{cases} \qquad \text{(IV.11)}$$

Et on déduit aisément de l'ensemble de ces équations que :

$$\Delta I = \frac{2\,I_0}{1 + \exp\left(-\frac{V_{in}}{U_T}\right)} - I_0 \qquad \text{(IV.12)}$$

soit donc que

$$\begin{cases} K_{S0} = 2\,I_0 \\ K_{pat} = \frac{1}{U_T} \end{cases} \qquad \text{(IV.13)}$$

IV.4.3- Transistor Bipolaire Latéral

Il a été montré qu'un transistor bipolaire, avec un collecteur qui n'est pas relié au substrat, est disponible sur les technologies CMOS [Vit83]. Ce composant est obtenu en faisant fonctionner un transistor MOS dans le mode bipolaire latéral.

Une vue en coupe d'un transistor à canal-p concentrique, réalisé sur une technologie n-well, est représentée sur la Figure IV.15. En polarisant la grille très au dessus de la tension de seuil, il y a apparition d'une région d'accumulation sous la grille et empêche le fonctionnement normal du transistor MOS. En polarisant alors correctement le caisson n-well, ainsi que la source et le drain du transistor, un mode de fonctionnement bipolaire est obtenu. Le caisson n-well constitue la base (B), la source l'émetteur (E), et le drain le collecteur (C) d'un transistor pnp latéral. Cependant, un transistor pnp parasite et vertical est aussi activé entre l'émetteur (E), la base (B) et le substrat (Sub), et son effet doit être minimisé.

Ce composant dispose donc de 5 terminaux et sa représentation symbolique est montrée sur la Figure IV.15.

Pour minimiser l'effet du transistor parasite vertical, il faut maximiser le rapport périmètre/surface de la diffusion p+ d'émetteur pour privilégier la diffusion des trous de l'émetteur vers le collecteur au dépens de la diffusion vers le substrat. Ainsi :

- l'émetteur doit être choisi de taille minimum,
- le collecteur doit entourer l'émetteur,
- la longueur de grille doit être minimum.

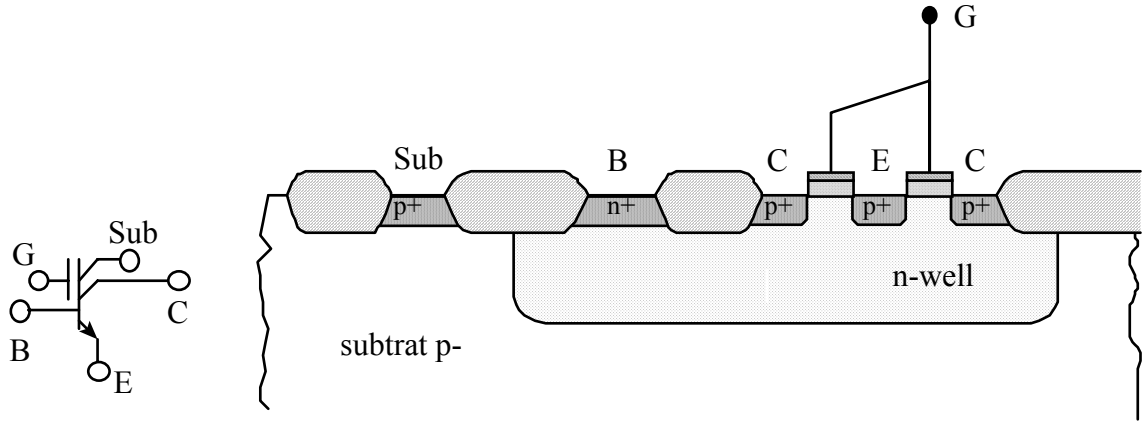


Figure IV.15 : symbole et vue en coupe d'un transistor bipolaire latéral.

Un exemple de layout d'un transistor bipolaire latéral sur la technologie Mietec 2µm est représenté sur la Figure IV.16.

Figure IV.16 : layout d'un transistor bipolaire latéral.

De la même manière, on obtient des transistors bipolaires latéraux npn sur une technologie p-well.

IV.4.4- Comparateur

Nous avons appelé comparateur l'élément de sortie de la cellule neurone A. Il doit comparer la quantité S_i de sortie de la sigmoïde à une variable aléatoire uniformément distribuée entre 0 et 1 (voir Figure III.11). Celle ci est disponible sous forme d'une tension uniformément distribuée sur un intervalle de tensions fixe [V_l, V_h]. On a vu, de plus, que la quantité S_i de sortie de la sigmoïde était un courant différentiel.

Ainsi, ce bloc de comparaison doit remplir les fonctions suivantes :

- transformer la tension représentant la variable aléatoire en un courant différentiel,
- faire une soustraction des deux courants différentiels,
- décider de l'état du neurone en fonction du signe de la soustraction.

Pour remplir la première fonction, on peut utiliser la technique de "linéarisation" présentée au paragraphe (§ IV.2.3) pour transformer la tension aléatoire en courant différentiel. Une schématique regroupant la sigmoïde et le comparateur est représentée sur la Figure IV.17. Sur cette figure, la tension

différentielle (V_1-V_2) est égale à la tension V_{pat} de l'équation (IV.6) et elle est donc obtenue à la sortie d'un convertisseur courant-tension.

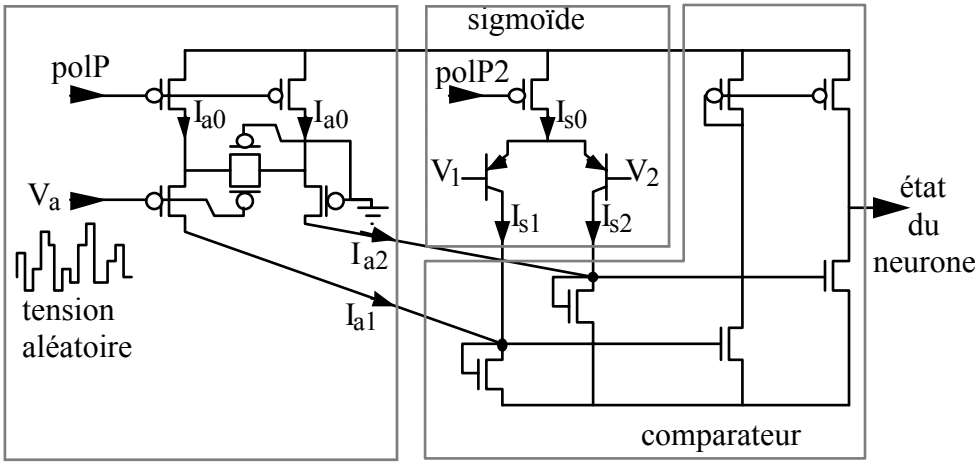


Figure IV.17 : schéma de principe du circuit neurone.

Pour un fonctionnement correct de ce bloc, il faut que :

$$\begin{cases} \text{si } V_a = V_l \text{alors} I_{a1} - I_{a2} = I_{s0} \\ \text{si } V_a = V_h \text{alors} I_{a1} - I_{a2} = -I_{s0} \end{cases}$$

Les tensions de polarisation polP et polP2 doivent donc être générées de telle sorte que ces équations soient vérifiées. En réalité, afin de rendre la génération de ces tensions indépendante de la génération de V_l et V_h, j'ai choisi d'avoir un courant de polarisation I_{a0} égal au double du courant de polarisation I_{s0}.

IV.4.5- Conception du Neurone A

La construction de la variable aléatoire à distribution uniforme à partir de nombre aléatoire binaires a été décrite dans la section (b) du paragraphe (§ III.2.2). On a vu alors qu'elle pouvait se faire par intégration d'un flot de nombres binaires (voir Figure III.4). Pour ce schéma, nous avons besoin de deux capacités C_1 et C_2 de même valeur. Le layout de cet intégrateur doit donc être très soigné et tous les éléments parasites sur une des capacités doivent donc être compensés par un élément semblable en parallèle sur l'autre capacité.

La schématique finale de la cellule neurone A est représentée sur la Figure IV.18. Afin de diminuer l'offset systématique du comparateur et d'augmenter la qualité du système et de la comparaison, des montages de type cascode sont utilisés (transistors P15, P29, P32, P33, P37, P38, P52 et P40).

Figure IV.18 : schématique de la cellule neurone A.

Des mesures sur un prototype ont montré qu'il était inutile d'appliquer sur les grilles des transistors bipolaires latéraux une tension plus grande que celle de l'alimentation "vdd". Si la grille est reliée à "vdd", on observe que la différence de potentiel entre cette grille et la base des transistors est toujours suffisante pour créer la zone d'accumulation. En effet, la gamme de variation de tension est ici relativement faible sur les entrées de la paire différentielle.

Le bloc de génération des polarisations polP et polP2, ainsi que des polarisations des montages cascodes, est présenté sur la Figure IV.19.

Figure IV.19 : schématique de la génération des polarisations

IV.5- LES ELEMENTS POUR L'APPRENTISSAGE DE LA SYNAPSE

IV.5.1- Introduction

L'architecture de la synapse complète avec apprentissage a été décrite dans le paragraphe (§ III.2.5) et notamment sur la Figure III.14. Ses éléments de base sont les compteurs analogiques et un bloc de comparateur. Je vais présenter ici le principe du compteur analogique et la conception de bloc de comparaison.

IV.5.2- Compteur Analogique

a- Définition

Un compteur analogique est une mémoire court-terme de tension qui peut être modifiée par pas élémentaires fixes ou réglables. La valeur mémorisée par le compteur analogique est stockée sous forme de charge électrostatique dans une capacité C_2. L'incrémentation et la décrémentation du compteur se font en ajoutant ou en retirant une charge élémentaire à la capacité de stockage. Cette quantité élémentaire de charge doit être parfaitement contrôlée.

b- Principe des compteurs 1 et 2

Plusieurs réalisations possibles des compteurs ont été étudiées. J'ai collaboré à l'élaboration d'un modèle des compteurs 1 et 2 présentés sur la Figure IV.20. Le principe du premier repose sur l'apport d'une charge élémentaire par mise en connexion de la capacité principale avec une petite capacité préalablement préchargée à la tension d'alimentation. Sa simplicité conduit à une caractéristique de charge fortement non-linéaire et à des modifications de la mémoire par pas fixes.

Ces défauts nous ont conduit à étudier un deuxième compteur chargé à courant constant pendant un temps constant. Ce temps est lui-même contrôlé par la décharge d'une deuxième capacité entre la tension d'alimentation et le seuil d'une porte logique. Les pas d'incrémentation et de décrémentation de ce compteur sont réglables. Son défaut majeur est que les pas de modification sont déterminés par des valeurs absolues des composants (et non des rapports) et ils sont donc très sensibles à des dispersions des paramètres technologiques. On trouvera en Annexe la publication [Mad91], relatant en détail ces deux études.

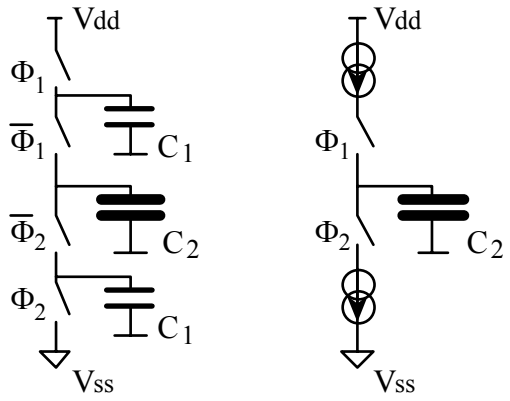


Figure IV.20 : Principe des compteurs 1 et 2.

c- Principe du compteur 3

La dispersion des caractéristiques du compteur 2 présenté ci-dessus nous a conduit a élaborer un nouveau compteur pour les prototypes de synapse avec apprentissage. J'ai collaboré à la définition de son principe que je vais décrire ici, tandis que les simulations, la réalisation et le test des prototypes ont été faits par Zhu Yiming et ils ont été présentés en détail dans [Zhu91].

J'ai représenté le principe de l'incrémentation sur la Figure IV.21 :

- Dans une première phase, une petite capacité C_1 est préchargée à une tension de précharge V_{pre}.

- Dans une seconde phase, la charge de la capacité C_1 est transférée dans un miroir de courant qui recopie une charge proportionnelle dans la capacité C_2.

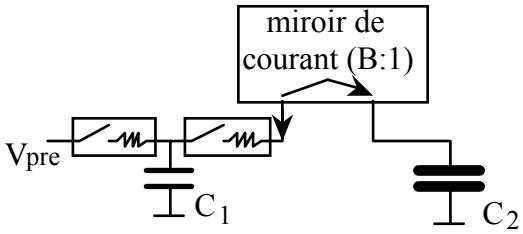


Figure IV.21 : principe d'une incrémentation du compteur 3

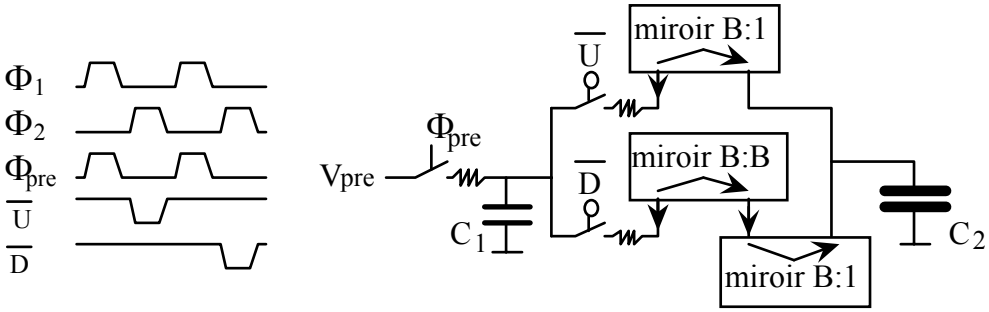


Figure IV.22 : principe du compteur.

L'avantage de ce système par rapport à ceux présentés précédemment, est d'assurer une décharge de la capacité C_1 indépendante de la tension aux bornes de la capacité C_2. On obtient donc ici un incrément d'amplitude indépendant de la valeur accumulée dans le compteur.

Pour que ce système fonctionne ainsi il faut que les transistors formant le miroir de courant ne sortent pas de leur zone de saturation. La résistance série des interrupteurs joue un rôle important car elle permet de limiter le courant instantané de décharge. De ce fait, elle limite la tension grille-source maximum des transistors formant le miroir de courant.

On veut, maintenant, compléter le compteur ci-dessus par un système de décomptage, et il faut pouvoir retirer une charge élémentaire à la capacité C_2 au lieu de l'ajouter. Nous allons chercher à utiliser la même capacité C_1 et la même tension analogique de contrôle V_{pre}. Ceci permet de limiter le nombre de tensions de contrôle extérieures, de limiter la surface de la cellule, et surtout d'assurer une meilleure coïncidence entre les valeurs de l'incrément et celle du décrément.

Le schéma de principe du compteur analogique avec incrémentation et décrémentation est représenté sur la Figure IV.22. Sa logique de commande est représentée sur la même Figure. Les interrupteurs commandés par les signaux $\overline{U}$ et

$\overline{D}$ sont des transistors à canal-p et donc à commande négative. Le séquencement de ce compteur est basé sur une horloge biphase (Φ_1,Φ_2). La précharge se fait pendant la première demi-phase Φ_1 et les signaux $\overline{U}$ et $\overline{D}$ sont synchronisés sur la deuxième demi-phase Φ_2.

IV.5.3- Comparateur

Le bloc de comparaison est décrit au paragraphe (§ III.2.5) et doit en particulier respecter l'équation (I.13). Ce bloc prend en entrée une tension variable et génère en sortie les signaux $\overline{U}$ et $\overline{D}$ utilisés par le compteur analogique décrit ci-dessus. Il utilise pour cela plusieurs tensions de références et l'horloge biphase (Φ_1,Φ_2) définie au paragraphe précédent.

La manière la plus simple pour réaliser ce bloc est de développer un comparateur simple et petit, puis d'utiliser deux de ces comparateurs pour comparer la tension d'entrée à deux tensions analogiques fixe. Ce principe est illustré sur la Figure IV.23.

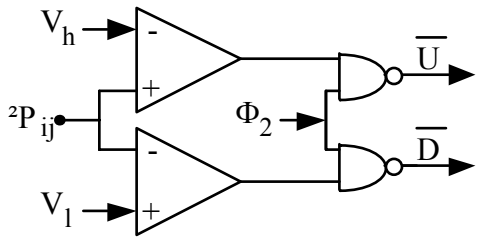


Figure IV.23 : Le bloc de comparaison.

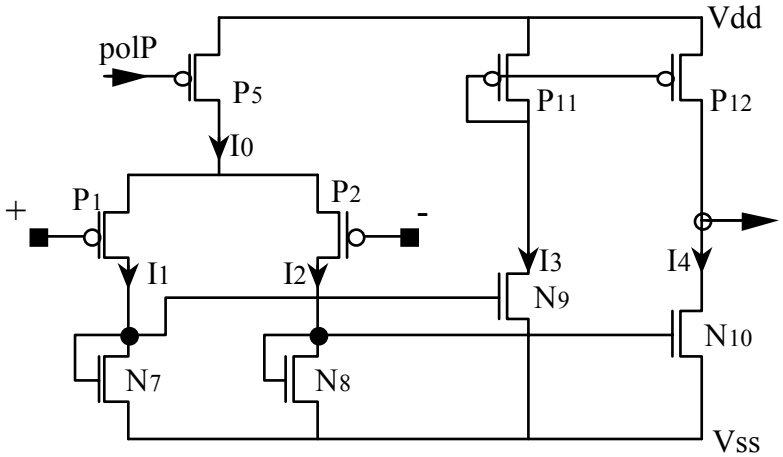


Figure IV.24 : Schématique du comparateur.

Les deux comparateurs doivent occuper une faible surface et leur layout doit être compatible avec celui de la cellule “convvi” décrite au paragraphe (§ IV.2), afin que leur contribution à l’encombrement de la synapse soit faible. J’ai ainsi choisi de réaliser un comparateur à structure symétrique et de faible surface. Sa schématique en transistors est représentée sur la Figure IV.24.

Une valeur de 10µm a été choisie pour la longueur des transistors. Ceci permet de limiter l’effet de modulation de longueur de canal et d’avoir un gain total suffisant. La réalisation de ce comparateur sur la technologie Mietec 3µm a conduit à un layout d’une surface de 6254 μm^2. La polarisation polP et les alimentations sont aussi utilisées par la cellule “convvi”, et on peut donc aisément abouter ces deux cellules. Les layouts des deux comparateurs de la Figure IV.23 sont un peu différents afin de limiter la surface de la synapse et le layout de l’un d’entre eux est représenté sur la Figure IV.25.

Figure IV.25 : Layout du comparateur.

Ce comparateur a été utilisé pour la réalisation d’un prototype de circuit avec apprentissage comprenant une matrice de 4x4 synapses.

A ce jour, les valeurs des tensions V_l et V_h sont mal connues, on sait seulement qu’elle doivent être proches de la moitié de la tension d’alimentation. Ainsi, pour la réalisation du prototype, nous avons choisi de ne pas les fixer et de les imposer au travers d’un plot du circuit. Un système de génération de ces tensions devra être conçu. Elles devront être symétriquement réparties autour de la moitié de la tension d’alimentation et leur différence devra être contrôlable.

IV.6- GENERATEUR ALEATOIRE OPTOELECTRONIQUE

La réalisation d’un générateur aléatoire optoélectronique a été exposée au paragraphe (§ III.2.2). On a vu qu’à partir de l’échantillonnage de l’image d’un speckle optique, on peut construire deux types de générateurs aléatoires :

- un générateur de nombres binaires aléatoires,
- ou un générateur de nombres réels aléatoires à répartition sigmoïdale.

Dans chacun des cas, on a besoin d'une matrice d'acquisition d'image à photodiode, cependant le premier système, grâce à son interface numérique, peut être conçu comme un système autonome alors que le deuxième ne le peut pas.

La Figure IV.26 représente le circuit d'échantillonnage de l'image du speckle d'un système autonome de génération de nombre binaires aléatoires. Il est constitué d'une matrice de cellules contenant chacune une photodiode à laquelle est associé un comparateur. Le comparateur est chargé de seuiller la tension aux bornes de la photodiode à l'issue de la période de décharge. Les nombres binaires sont extraits de la matrice par un registre à décalage.

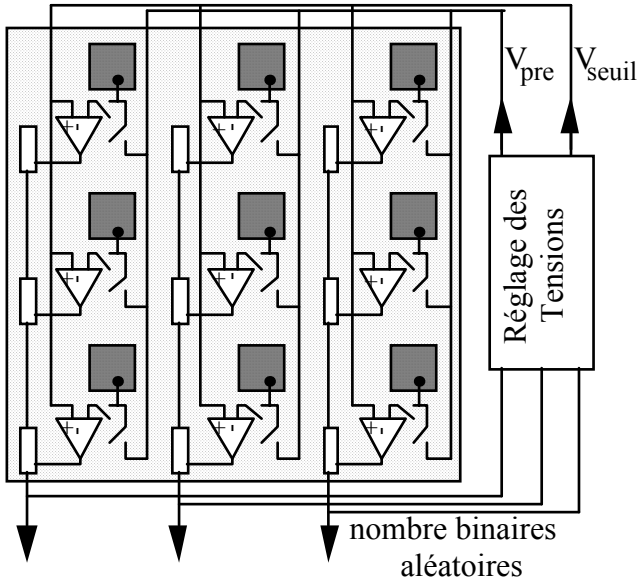


Figure IV.26 : Générateur de nombres binaires aléatoires

Ces circuits sont réalisés sur une technologie CMOS classique en utilisant les diffusions de drain pour la réalisation des photodiodes. Les photodiodes sont préchargées par une tension V_{pre}, et la tension qui sert de référence au comparateur pour le seuillage est V_{seuil}. Ces tensions doivent être choisies de telle sorte que le probabilité de sortie d'un nombre binaire aléatoire soit égale à 1/2. Le système est donc doté d'une unité de génération de ces tensions à partir d'une accumulation des échantillons de sorties.

Nous avons d'une part réalisé un prototype de la matrice de photodiodes sur la technologie numérique 2µm ES2 à caisson n-well. Les photodiodes sont réalisées avec des diffusions dopées n+ sur le substrat dopé p-. Pour ce circuit, j'ai réalisé un

comparateur semblable à celui décrit sur la Figure IV.24. On pourra trouver plus de détail dans [Mad90].

J'ai d'autre part collaboré avec J.C. Rodier pour la réalisation d'un prototype du générateur de tensions. Celui a été testé sur le circuit à photodiodes décrit plus haut. Il utilise un intégrateur à capacité commutée pour générer une tension de seuil variable. La tension de précharge des photodiodes est fixée à V_{dd}. On pourra trouver plus de détail dans [Rod92].

La faible taille du prototype réalisé devrait permettre d'en placer un grand nombre dans chaque circuit et de les associer à un faible nombre de photodiodes. Ainsi, les défauts d'uniformité de l'intensité lumineuse du speckle et de sensibilité des photodiodes seront en grande partie corrigés.

La réalisation d'un générateur de nombres réels aléatoires à répartition sigmoïdale impose de modifier la matrice décrite ci-dessus en remplaçant la tension de seuil par une tension différente pour chaque cellule et représentant la contribution du réseau (voir Figure III.12). Chaque cellule remplit alors les fonctions d'un neurone complet. La tension de précharge peut alors être asservie en utilisant une mesure de la tension aux bornes de la photodiode après décharge, ou bien une mesure binaire après seuillage avec une contribution nulle. Les variations de la Température T sont obtenues par modulation de la puissance de la source Laser.

Tous les travaux présentés dans ce paragraphe sont issus d'une collaboration entre l'Institut d'Electronique Fondamentale et l'équipe du Dr P. Chavel de l'Institut d'Optique Théorique et Appliquée.

IV.7- MBA1 : CIRCUIT AVEC 1024 SYNAPSES SANS APPRENTISSAGE

IV.7.1- Introduction

Le circuit appelé ***mba1*** (pour **M**achine de **B**oltzmann **A**nalogique **1**), est composé d'une matrice de synapses de 16 lignes et 64 colonnes et des interfaces numériques avec le réseau d'interconnexions introduit au paragraphe (§ III.3.4). Son organisation, conçue avec Patrick Garda, est basée sur la description de l'architecture d'une machine sans apprentissage du paragraphe (§ III.3.3).

Il a été réalisé sur la technologie CMOS 3µm, 2 Métaux, 2 Polysiliciums de *Mietec Alcatel*®. Nous avons eu accès à cette technologie par l'intermédiaire d'un MPW (Multi-Project Wafer) de la division *Invomec* de *Imec* à Leuven en Belgique.

Une partie du circuit a été entièrement dessinée ("full-custom"), alors que pour le reste, nous avons utilisé des éléments de la bibliothèque mixte analogique-numérique Mietec de cellules pré-caractérisées.

L'alimentation du circuit se fait par des tensions de 0-5V. Les alimentations des parties numériques et analogiques sont physiquement séparées et la partie analogique utilise une tension de référence à 2,5V.

Ce circuit peut calculer la contribution V_i^n de 16 neurones différents ayant pour entrée 64 neurones communs. Les états des neurones sont binaires et notés X_j^n, les poids synaptiques sont notés W_{ij} et ils sont analogiques.

On obtient donc ici :

$$V_i^n = \sum_{j \neq i} W_{ij}.X_j^n$$

avec $i \in \{0...15\}$ et $j \in \{0...63\}$ et $X_j^n \in \{0, 1\}$

Si on veut calculer des contributions avec en entrée un plus grand nombre de neurones, ces circuits pourront être facilement cascadés par une simple interconnexion de leurs plots de sortie. En effet, la contribution du réseau est représentée en sortie par un courant I_i.

mba1 est destiné à être utilisé avec le circuit *mba2* décrit au paragraphe suivant pour la réalisation d'un module de traitement prototype (cf § IV.9).

Les vues générales de la schématique et du layout de ce circuit sont représentées sur les Figures IV.35 et IV.36.

La surface totale du circuit est 55mm^2, sa consommation sera inférieure à 200mW et il est constitué d'environ 35.000 transistors. Ce circuit peut calculé 1024 connections à une fréquence de relaxation de 300 kHz : soit une puissance de calcul de 300 MOPS (Millions d'Opérations par Seconde).

IV.7.2- Organisation Interne

a- La Synapse

Chaque élément de la matrice (synapse) est constitué d'une capacité de stockage du poids synaptiques, du convertisseur tension-courant *convvi* décrit au paragraphe (§ IV.2).

A chaque synapse est adjoint un point mémoire de configuration. Lorsque une valeur nulle est stockée dans ce point mémoire, l'étage de sortie du convertisseur *convvi* est inhibé et la synapse est inutilisée. On peut donc ainsi changer entièrement la configuration du réseau de neurones pour lequel *mba1* est utilisé.

Une schématique générale de la synapse, ainsi qu'une vue détaillée de la partie numérique servant à la configuration de la synapse, sont représentées sur la Figure IV.27.

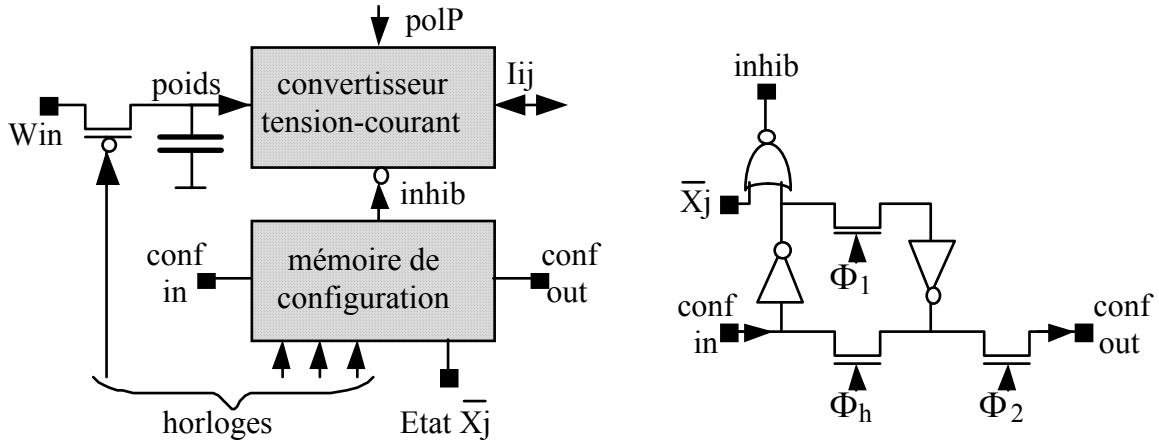


Figure IV.27 : la synapse de *mba1*.

Le point mémoire de configuration est un élément de registre semi-statique. La sortie de ce registre et le complément de l'état du neurone sont connectés à une porte "nor". Ainsi, on peut déduire la table de vérité représentée sur le Tableau IV.4.

$\overline{X_j}$	$conf_{in}$	inhib
0	0	0
1	0	0
0	1	1
1	1	0

Tableau IV.4 : configuration de la synapse.

La synapse est donc inhibée si le point mémoire est chargé par son entrée $conf_{in}$ avec un "0". L'entrée $\overline{X_J}$ doit être égale au complément de l'état du neurone.

La synapse a été dessinée entièrement en "full-custom". Une valeur de 2,5 a été choisie pour le paramètre "a" de l'équation (IV.3). Ainsi, d'après le Tableau IV.2, le rapport entre les transistors d'entrée (P_1 et P_2 de la Figure IV.2) et les transistors de

dégénérescence (P_3 et P_4) est égal à 6. Ce rapport est obtenu en plaçant en parallèle 2 transistors identiques de taille élémentaire pour réaliser P_1 et P_2, et 3 autres en série pour réaliser P_3 et P_4. La taille du transistor élémentaire est w=6µm, l=10µm. Une vue de la schématique du convertisseur tension-courant ainsi qu'un résultat de simulation électrique montrant sa fonction de transfert sont représentés sur la Figure IV.34.

La surface totale du layout de la synapse est 18850µm^2, la moitié environ étant occupée par la partie numérique de configuration. Une vue de ce layout est reproduite sur la Figure IV.28.

Figure IV.28 : Layout de la synapse.

Au lieu d'utiliser un point mémoire pour inhiber une cellule synapse inutilisée, on pourrait charger la synapse avec un poids nul. Cependant, dans ce cas l'influence des offsets des amplificateurs à transconductance n'est pas éliminée et la synapse inutilisée intervient en fait dans la valeur du poids W_{is} de la connexion de seuil (voir équation I.2).

Les mesures effectuées sur ce prototype sont destinées à permettre de conclure sur l'utilité de ce point mémoire. Il faut toutefois noter que dans le cas de la réalisation de matrices de synapses avec apprentissage, ce point mémoire est tout-à-fait nécessaire mais on peut peut-être se limiter à une configuration par colonne.

b- Cellule de Polarisation

Le convertisseur est polarisé par une tension appelée polP. Cette tension doit être réglée de telle sorte les fonctions de transfert poids-courant de toutes les synapses d'un module de traitement soient identiques. Cette tension, nous servira donc à minimiser les dispersions de caractéristiques pouvant exister entre circuits.

Un bloc ajustant automatiquement la polarisation polP, en fonction d'une référence de courant extérieure, n'a pas été développé pour ce circuit. J'ai préféré pour ce prototype développer une cellule de polarisation me permettant de contrôler le courant de polarisation I_0 en fonction d'une tension extérieure.

La schématique de cette cellule est représentée sur la Figure IV.29. Elle est inspirée d'une cellule de polarisation décrite dans [Vit85]. Ce montage permet d'obtenir des courants de polarisation très faibles allant de la zone de faible inversion au bas la zone de forte inversion. Le rapport du miroir de courant (P_1, P_2) est égal à 1

alors que le rapport $ß_3/ß_4$ entre les tailles des transistors (N_3, N_4) est inférieur à 1. Le transistor N_5 est utilisé en résistance contrôlée par la tension d'entrée appelée pol.

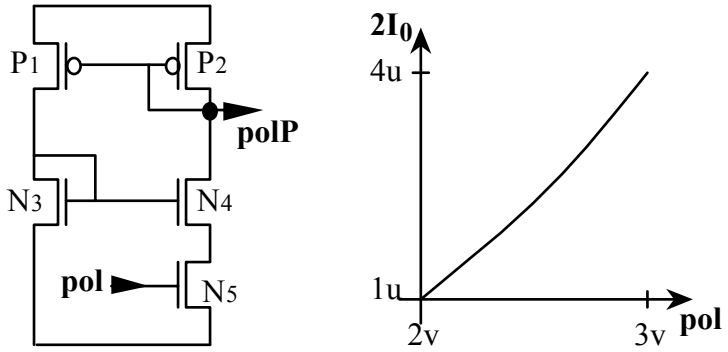


Figure IV.29 : cellule de polarisation.

Les tailles des transistors N_4 et N_5 ont été optimisées dans le but de minimiser l'effet d'une dispersion des paramètres W et L des transistors sur la tension de polarisation polP.

Le courant de polarisation $2I_0$ de la cellule convvi obtenu pour une tension d'entrée pol variant entre 2 et 3 Volt est reporté sur la Figure IV.29.

Par le réglage de la tension pol, on règle donc le paramètre I_M de l'équation (IV.7). On peut dire que sur ce circuit, ce paramètre est aisément réglé dans l'intervalle [0,5µA, 4µA].

c- La Matrice Synaptique

La Figure IV.30 présente l'organisation générale de la matrice de synapses et de ses entrées-sorties.

Les états X_j des neurones U_j sont distribués en colonne alors que les courants I_i, représentants les contributions du réseau, sont collectés en ligne. Le courant I_i est généré par une interconnexion horizontale des sorties des convertisseurs tension-courant.

Le paramètre Z de l'équation (III.13) vaut 8, je rappelle qu'il est défini comme étant le nombre d'états de neurones multiplexés dans le temps sur un même plot numérique. Ainsi, le circuit comprend 8 plots d'entrées numériques (X<0:7>) pour l'entrée en série des états de neurone et 8 convertisseurs série-parallèle 8 bits pour la restitution des 64 états (X_J<0:63>). Les convertisseurs sont appelés "s1p8mux".

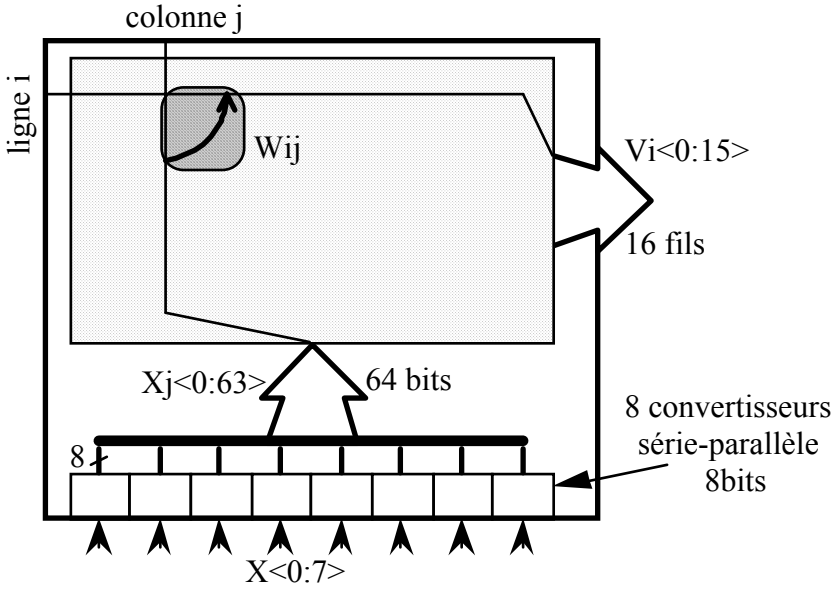


Figure IV.30 : les entrées-sorties de la Matrice.

Les contributions sont disponibles sur 16 plots analogiques V_i<0:15>.

Les différents points mémoires de configuration sont connectés en lignes pour former un registre à décalage de longueur 1024 comme il est représenté sur la Figure IV.31.

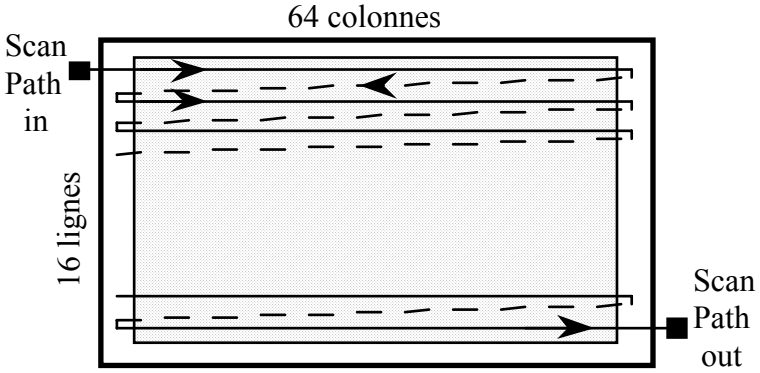


Figure IV.31 : Organisation des points mémoires de configuration.

La configuration de la matrice se fait donc en entrant en série sur un plot numérique la configuration d'un circuit complet. La sortie de ce registre à décalage est connectée à un plot de sortie numérique afin de simplifier le test de bon fonctionnement de ce "Scan Path".

d- Stockage et Rafraîchissement des Poids

Les poids sont stockés dans la matrice et rafraîchis par l'intermédiaire d'un convertisseur numérique-analogique (DAC) 8 bits. Ce convertisseur est extrait de la bibliothèque Mietec de cellule pré-caractérisée.

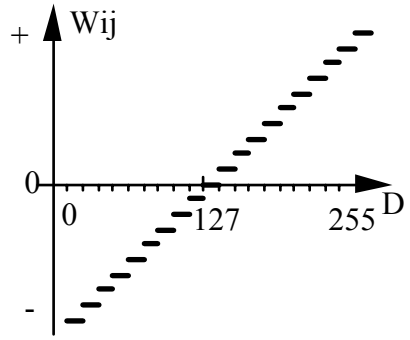


Figure IV.32 : codage des poids

Le codage des poids est bipolaire, c'est à dire que les poids sont signés (positifs ou négatifs) et qu'un poids nul est représenté par une donnée numérique D de 127, (voir Figure IV.32).

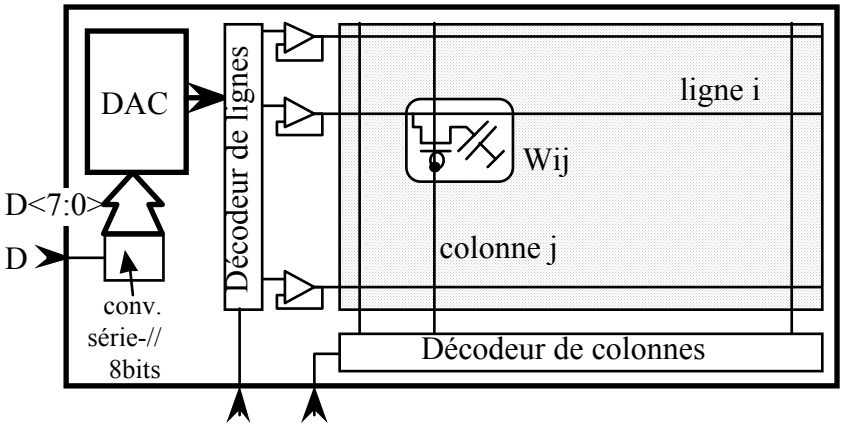


Figure IV.33 : Stockage des poids.

Le principe de fonctionnement de la partie stockage et rafraîchissement des poids est illustré par la Figure IV.33.

Tout d'abord, les 8 bits de données du DAC sont entrés en série sur un seul plot numérique puis convertis et latchés par un convertisseur série-parallèle 8bits (appelé "s1p8").

Le stockage des poids dans la matrice se fait par colonnes entières :

- Dans une première phase, les données numériques correspondants aux 16 poids d'une colonne sont entrés dans le circuit et les 16 valeurs analogiques correspondantes sont successivement stockés sur les grilles d'entrée de 16 suiveurs par adressage de ligne.
- Dans une deuxième phase, la colonne considérée est adressée et les suiveurs chargent les 16 capacités représentant les poids de la colonne par l'intermédiaire des transistors canal-p (voir aussi la Figure IV.27).

L'adressage du suiveur à connecter au DAC se fait par un décodeur de ligne (appelé "rowscan16") et celui de la colonne se fait par un décodeur de colonne (appelé "colscan64"). Cet adressage se faisant toujours de manière séquentiel (pas d'accès aléatoire à un poids particulier), les décodeurs sont en fait des registres à décalage bouclés sur eux-même.

Les poids doivent être rafraîchis en permanence.

IV.7.3- Interface Utilisateur

Le séquencement des convertisseurs, des décodeurs de ligne et de colonne, ainsi que celui des différents convertisseurs série-parallèle, n'a pas été inclus dans le circuit. Seule la matrice de synapse a été réalisée en "full-custom", les autres cellules sont extraites de la bibliothèque Mietec de cellules pré-caractérisée.

Outre le mode de fonctionnement normal conforme à la description ci-dessus, le circuit possède un mode de test et un mode d'initialisation.

Dans le mode d'initialisation, les décodeurs de ligne et de colonne sont reliés en entrée et en sortie à des plots numériques alors qu'ils sont bouclés sur eux-même dans en fonctionnement normal. Afin d'éviter une consommation et un bruit numériques inutiles les décodeurs ne sont pas reliés au plot de sortie en fonctionnement normal. Le mode d'initialisation est sélectionné lorsqu'un signal numérique appelé "LOOP" prend la valeur 0.

Le mode de test est utilisé pour tester les 8 convertisseurs série-parallèle 8 bits d'entrée des états de neurones. Lorsque le signal numérique appelé "XjSpMODE" prend la valeur 1, les 8 convertisseurs sont mis en séries pour former un registre à décalage 64 bits. L'entrée du premier convertisseur (X<0>) sert d'entrée à ce registre, alors que sa sortie est connectée au plot de sortie numérique normalement relié à la sortie du scanner de configuration de la matrice (ScanPathOut).

Le tableau IV.5 fait le récapitulatif des différents modes de fonctionnement.

LOOP	XjSpMODE	fonctionnement	Mode
0	0	• convertisseurs 8 bits : normal • décodeurs : initialisation	1
0	1	• convertisseurs 8 bits : test • décodeurs : initialisation	2
1	0	• convertisseurs 8 bits : normal • décodeurs : normal	3
1	1	• convertisseurs 8 bits : test • décodeurs : normal	4

Tableau IV.5 : configuration de la synapse.

Remarque : Dans le mode de test des convertisseurs série-parallèle, 8 colonnes de neurones sont directement reliés aux 8 plots d'entrée X<0:7> sans traverser de cellules séquentielles. Nous avons choisi d'implanter ce mode de fonctionnement asynchrone afin de pouvoir d'une part caractériser plus facilement la cellule synapse, mais aussi afin de rendre utilisable ce circuit pour le test d'un réseau de neurone asynchrone et à temps continu.

IV.7.4- Figures

Cette section présente quelques Figures illustrant la réalisation du circuit *mba1*.

Tout d'abord, la Figure IV.34 présente le convertisseur tension-courant (convvi) et son élément de polarisation. Les fonctions de transfert de ces deux éléments sont reportées sur la même Figure. Elles sont obtenues avec le simulateur électrique Hspice.

Ensuite, la Figure IV.35 présente une vue générale de la schématique du circuit. On reconnaîtra sur cette Figure, la matrice de 16x64 synapses appelée "matsyn16x64". La tension de référence à 2,5V est nommée sur cette Figure : v_{ddgnd}.

Ensuite, la Figure IV.36 présente une vue générale du Layout du circuit. Les parties analogiques et numériques sont parfaitement découplées sur ce Layout. On reconnaîtra aisément la matrice matsyn16x64 dans le zone en haut à droite. A l'intérieur de la matrice, les seules portes logiques changeant d'état en fonctionnement normal sont les portes "nor" inhibant l'état du neurone. Le bruit

numérique dû à un couplage entre la partie numérique et la partie analogique devrait être faible.

Figure IV.34: Vue schématique du convertisseur tension-courant et de son élément de polarisation.

Figure IV.35 : Vue schématique du circuit *mba1*.

Figure IV.36 : Vue du layout de *mba1*.

IV.8- MBA2 : CIRCUIT AVEC 32 NEURONES

IV.8.1- Introduction

Le circuit appelé ***mba2*** (pour **M**achine de **B**oltzmann **A**nalogique **2**), est composé de 32 neurones et d'un générateur pseudo-aléatoire à automate cellulaire introduit au paragraphe (§ III.3.4.d).

Il a été réalisé sur la technologie CMOS 2,4µm, 2 Métaux, 2 Polysiliciums de *Mietec Alcatel*®. Nous avons eu accès à cette technologie par l'intermédiaire de l'organisation Européenne de fabrication de circuit intégrés *Eurochip*. Une partie du circuit a été entièrement dessinée ("full-custom"), alors que pour l'autre partie, nous avons utilisé des éléments de la bibliothèque Mietec mixte analogique-numérique de cellules pré-caractérisées.

L'alimentation du circuit se fait par des tensions de 0-5V. Les alimentations des parties numériques et analogiques sont physiquement séparées et la partie analogique utilise une tension de référence à 2,5V.

Ce circuit peut calculer, à l'instant n, les états X_i^n de 32 neurones différents à partir des 32 courants I_i représentant contributions du réseau sur ces neurones.

En sortie du circuit on obtient donc, d'après les équations (I.5) et (IV.7), la probabilité de l'état du neurone i :

$$P(X_i=0) = \frac{1}{\left[1 + \exp\left(\frac{I_i}{I_M.T}\right)\right]}$$

où i □ {0...32} et I_M est le paramètre de normalisation qui caractérise le codage en courant de la contribution du réseau, voir le paragraphe (§ IV.3.1).

Le produit du paramètre I_M par la température T est une grandeur caractéristique du circuit. On a vu sur l'équation (IV.10) qu'il est égal au produit de deux autres paramètres R_{ct} et K_{pat}. Ces deux paramètres caractérisent les deux éléments fondamentaux du neurone : le convertisseur courant-tension et la sigmoïde. Ils seront explicités dans les paragraphes qui suivent.

mba2 est destiné à être utilisé avec le circuit *mba1* décrit au paragraphe (§ IV.7) pour la réalisation d'un module de traitement prototype (cf § IV.9).

Les vues générales de la schématique et du layout de ce circuit sont représentées sur les Figures IV.40 et IV.41.

La surface totale du circuit est 33mm^2, alors que les 3/4 de la surface du cœur du circuit est occupée par le générateur aléatoire. Il consomme environ 350mW.

IV.8.2- Organisation Interne

a- Organisation générale

La Figure IV.37 présente l'organisation générale du circuit *mba2*. Il est tout d'abord constitué de 32 convertisseurs courant-tension permettant la variation de la température T, décrits au paragraphe (§ IV.3), ainsi que de 32 blocs sigmoïde-comparateur, décrits au paragraphe (§ IV.4).

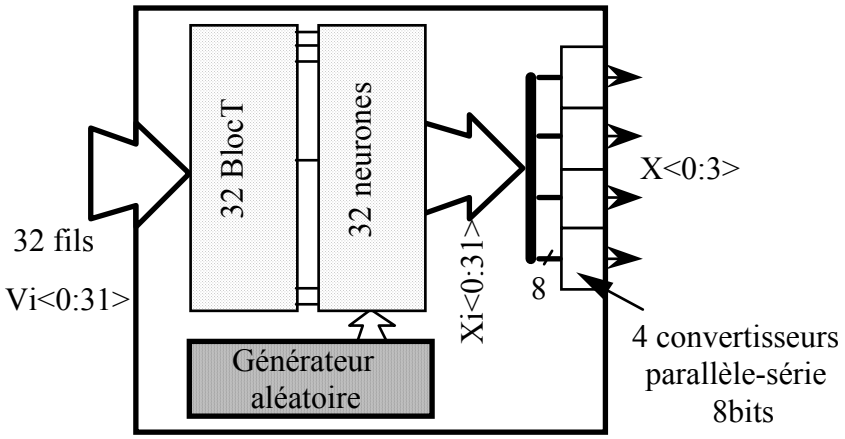


Figure IV.37 : Organisation du circuit.

Les 32 courants I_i sont distribués aux 32 convertisseurs, appelés BlocT. Les sorties des 32 convertisseurs sont reliées chacune à un bloc neurone A (intégrateur-sigmoïde-comparateur).

Le paramètre Z (cf. équation III.13) vaut 8, et ainsi les 32 états de neurones en sortie de ces derniers blocs (X_i<0:31>) sont multiplexés dans le temps sur 4 plots de sortie numérique (X<0:3>) et 4 convertisseurs parallèle-série 8 bits sont donc nécessaires. Les convertisseurs sont appelés "p8s1".

b- Bloc Neurone.

La conception du bloc Neurone a été décrite tout au long du paragraphe (§ IV.4) pour aboutir finalement à la schématique du Neurone utilisé dans ce circuit,

(voir Figure IV.18). On sait déjà, d'après l'équation (IV.13), que l'utilisation de transistors bipolaires latéraux pour la réalisation de la sigmoïde conduit à :

$$K_{pat} = \frac{1}{U_T} = \frac{q}{kT} \approx 38{,}5\ V^{-1} \text{ à } 300°K \qquad \text{(IV.14)}$$

Sur la Figure IV.18, les Entrées Bit et $\overline{Bit}$ sont le nombre binaire aléatoire et son complément et sont fournis par la générateur pseudo-aléatoire décrit ci-dessous.

Les tensions polP, B2 et B3 sont générées par le bloc décrit sur la Figure IV.19 à partir de la tension extérieure polP2.

Les tensions V_l et V_h ne sont pas générées en interne et doivent être imposées et réglées par l'utilisateur.

Le séquencement du circuit est fait par une horloge à deux phases (Φ_1, Φ_2). D'après les simulations électriques de cette cellule, la fréquence de cette horloge pourrait être de 300 kHz.

c- Convertisseur courant-tension

Figure IV.38 : schématiques de decodeT et blocT.

Le principe du convertisseur courant-tension a été décrit au paragraphe (§ 4.3.3) et en particulier sur la Figure IV.13. Il permet d'obtenir une variation de la température T par l'intermédiaire de 3 signaux numériques S_1, S_2, S_3 et d'un signal analogique R_c. Les trois signaux numérique sont générés par un décodeur (appelé decodeT) à partir du complément des 2 bits A<1:0>.

L'amplificateur opérationnel et les portes logiques sont des éléments de la bibliothèque Mietec de cellules pré-caractérisées.

Les schématiques du convertisseur (blocT) et du décodeur sont présentés sur la Figure IV.38.

D'après le Tableau IV.3 et la schématique de decodeT, on peut déduire le Tableau IV.6 représentant la valeur de R_{ct} en fonction de A<1:0> et de la valeur R_{ct0} en A=0.

A<1:0>	A<1>	A<0>	R_{ct}

0	0	0	R_{ct0}
1	0	1	$R_{ct0}/2$
2	1	0	$R_{ct0}/4$
3	1	1	$R_{ct0}/8$

Tableau IV.6 : R_{ct} en fonction de A.

Afin de déterminer la valeur du paramètre R_{ct0} en fonction de la tension de commande R_c, utilisons le modèle du transistor dans sa zone de conduction présenté dans le Tableau IV.1 et complété par l'effet de modulation de la tension de seuil.

Si l'on suppose que la $|v_{gs} - V_T| >> |v_{ds}|$, on peut alors écrire que le courant i_d est égal à :

$$i_d = ß.(v_{gs} - V_T).v_{ds}$$

et le transistor est équivalent à une résistance de valeur :

$$R_{on} = \frac{1}{ß.(v_{gs} - V_T)}$$

La résistance R_{ct0} est définie par la mise en série de 2 résistances, l'interrupteur commandé par la tension v_{dd} et la résistance variable commandée par la tension R_c. On a donc, compte tenu de l'équation IV.1 et des tailles des transistors de la schématique de la Figure IV.38 :

$$R_{ct0} = \frac{2,4}{4.\mu_n C_{ox}.(v_{dd} - v_{ddgnd} - V_T)} + \frac{9,6}{4\mu_n C_{ox}.(R_c - v_{ddgnd} - V_T)} \qquad \text{(IV.15)}$$

μ_n est la mobilité du transistor canal-n et vaut ici 70,2 μm^2/V.ns.

C_{ox} est la capacité d'oxyde et vaut ici 0,812 fF/μm^2,

V_{Tn} est égal à la tension de seuil du transistor canal-n pour une tension de canal-substrat de 2,5V (vddgnd). Elle est égale à :

$$V_{Tn} = V_{T0} + \gamma.\left(\sqrt{2|\Phi_b| + v_{ddgnd}} - \sqrt{2|\Phi_b|}\right) = 1,14V$$

V_{T0} est la tension à $v_{sb} = 0$ et vaut ici 0,9V,

Φ_b est le potentiel de substrat et vaut ici 0,7V,

γ est coefficient d'effet de substrat et vaut ici 0,3.

Et donc :

$$\boxed{R_{ct0} = \left(7{,}7 + \frac{42{,}1}{R_c - 3{,}64}\right) \ k\Omega} \qquad \text{(IV.16)}$$

par exemple, si R_c = 4,6V, sachant que K_{pat}=38,5 (éq.IV.14) et que I_M=1µA, on déduit de l'équation (IV.10) que T=0,5.

d- Générateur aléatoire

L'architecture du générateur aléatoire à automate cellulaire a été décrite au paragraphe (§ III.2.2.d). La schématique de l'automate réalisé est représentée sur la Figure IV.39.

Alors que, l'équation (III.6) décrit le fonctionnement des sites, j'ai ajouté une porte "nor" et un signal numérique de contrôle (C) pour permettre l'initialisation de l'automate cellulaire. L'équation (III.6) devient :

$$X_m^n = \left[X_{m-1}^{n-1} \, H \, (X_m^{n-1} + X_{m+1}^{n-1})\right] . \overline{C} \ + \left[X_{m-1}^{n-1}\right] .C \qquad \text{(IV.17)}$$

Lorsque le signal C est à "1", l'automate cellulaire se comporte comme un registre à décalage.

Les nombres binaires d'un site sur 8 sont utilisés comme nombres aléatoires, et donc l'automate est constitué de 256 sites dans le circuit. Les extrémités de l'automate sont reliées à des plots du circuit. Ainsi, si on utilise plusieurs circuits neurones, on peut chaîner ces automates afin d'en construire en plus long ; les séquences aléatoires sont ainsi plus longues et les statistiques de l'automate meilleures.

Cet automate ayant été réalisé avec des cellules de la bibliothèque Mietec de cellules pré-caractérisées, il occupe les 3/4 de la surface du cœur du circuit. Cependant, sa structure est très répétitive et le développement en "full custom" d'une nouvelle cellule incluant les fonctions complètes d'un site permettra un gain de surface important.

IV.8.3- Interface Utilisateur

Le séquencement des convertisseurs parallèle-série, du générateur aléatoire, et des neurones n'a pas été inclus dans le circuit. Ainsi ces horloges doivent être générées par l'utilisateur :

- horloge biphase (Φ_1, Φ_2) pour le neurone,

- déclenchement sur un front du générateur aléatoire,
- déclenchement sur un front des convertisseurs série-parallèles.

IV.8.4- Figures

Cette section présente quelques Figures illustrant la réalisation du circuit *mba2*.

Tout d'abord, la Figure IV.39 représente le générateur pseudo-aléatoire à automate cellulaire.

La Figure IV.40 montre une vue générale de la schématique du circuit. On reconnaîtra sur cette Figure, le bloc sigmoïde-comparateur appelé "neurone", le convertisseur courant-tension appelé "BlocT", ainsi que le générateur aléatoire appelé "geneAlea". La tension de référence à 2,5V est nommée sur cette Figure : V_{ddgnd}.

Enfin, la Figure IV.41 présente une vue générale du Layout du circuit.

Figure IV.39: Vue schématique du générateur aléatoire.

Figure IV.40 : Vue schématique du circuit *mba2*.

Figure IV.41 : Vue du layout de *mba2*.

IV.9- MBA1, MBA2 : LA BASE D'UN MODULE DE TRAITEMENT

Les circuits *mba1* et *mba2* ont été conçus pour former les éléments de base d'un module de traitement d'une Machine de Boltzmann sans apprentissage. Un tel module de traitement a été évoqué dans le paragraphe (§ III.3.4).

Le schéma du module de traitement est illustré sur la Figure IV.42.

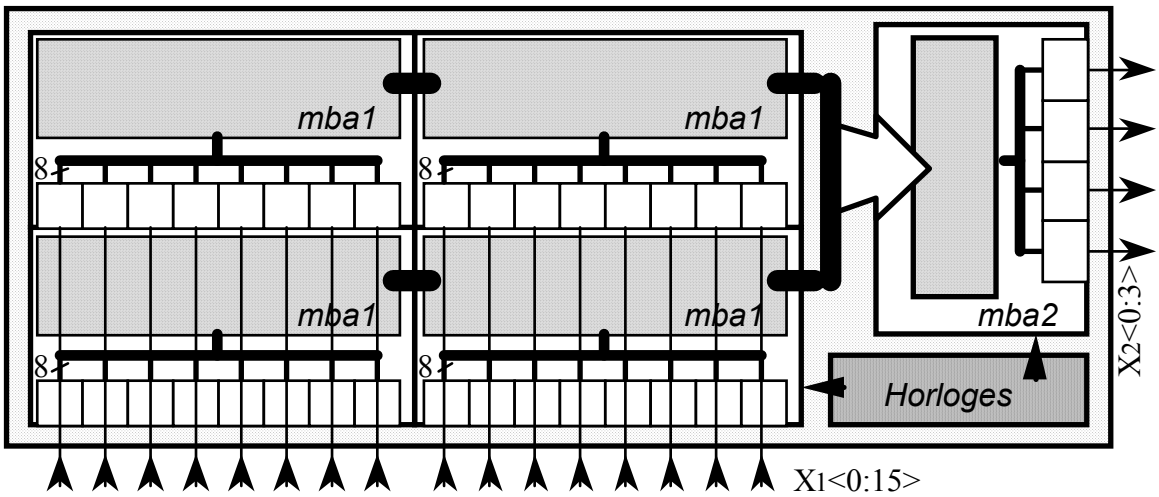


Figure IV.42 : Module de traitement.

Ce module regroupe 4 circuits *mba1* et 1 circuit *mba2* pour former une matrice de 32x128 synapses sans apprentissage permettant le calcul de nouveaux 32 états de neurones à partir de 128 autres en entrée. Le module de traitement comprend donc 16 fils pour l'entrée en série des états de neurones et 4 fils pour la sortie des nouveaux états.

La durée nécessaire à une relaxation complète d'une machine utilisant ce module devrait être inférieur à 5µs.

IV.10- CONCLUSION

J'ai présenté dans ce chapitre, la conception analogique des différentes cellules nécessaires à la réalisation des circuits spécialisés d'une Machine de Boltzmann avec ou sans apprentissage.

J'ai ensuite décrit la conception de deux circuits permettants d'aboutir à la construction d'un module de traitement 32x128 sans apprentissage.

La suite immédiate de ce travail sera la conception d'un circuit incluant toutes les spécificités d'un module de traitement et en particulier la génération des polarisations et la génération des signaux d'horloges.

CHAPITRE V

TEST DES CELLULES DE BASE ET DES CIRCUITS

V.1- INTRODUCTION

Dans ce chapitre je vais tout d'abord présenter les tests de plusieurs cellules décrites au chapitre IV. Ces tests ont été faits sur des circuits qui avait été spécialement réalisés dans ce but :

- le circuit MBAT4 comprenant des synapses réalisées, sans le système de compensation de la linéarisation mais avec une large plage d'entrée (input range),
- le circuit MBAT5 comprenant des synapses linéarisées,
- le circuit MBAT8 comprenant un prototype de sigmoïde et ayant permis le test des transistors bipolaires latéraux.

Je présenterai, ensuite, le test des circuits de plus grande taille présenté au chapitre IV et en particulier du circuit MBA1.

Les circuits MBAT4, MBAT5, MBAT8 et MBA1 ont été fabriqués sur la technologie CMOS 3µm, 2 Métaux, 2 Polysiliciums de Mietec Alcatel par l'intermédiaire du MPW (Multi Project Wafer) organisé par la division *Invomec* de Imec en Belgique.

V.2- TEST DES CIRCUITS MBAT4 ET MBAT5

V.2.1- Organisation

L'organisation interne des deux circuits MBAT4 et MBAT5 est identique. Ces deux circuits font partis de la série des circuits MBAT (**M**achine de **B**oltzmann **A**nalogique **T**est) qui ont été construits pour tester le fonctionnement des différentes

cellules présentées aux chapitre III et IV et que l'on retrouve en partie dans les circuits MBA1, MBA2.

MBAT4 et MBAT5 sont destinés à tester le fonctionnement des convertisseurs tension-courant et de la synapse sans apprentissage.

Ils sont organisés en matrice de 4 lignes et 10 colonnes de synapses. Une tension analogique simulant une valeur de poids est commune à toute une colonne de synapses. La présence des neurones est simulée par des amplificateurs opérationnels internes qui sont connectés selon le principe de la Figure V.1. La résistance de contre-réaction est externe au circuit et les trois lignes du bas de la matrice sont reliées au même neurone. En l'absence de la résistance externe, les synapses sont isolées de l'influence de l'amplificateur opérationnel et on peut alors tester la réalisation d'un grand réseau de neurone par interconnection de plusieurs circuits synapses.

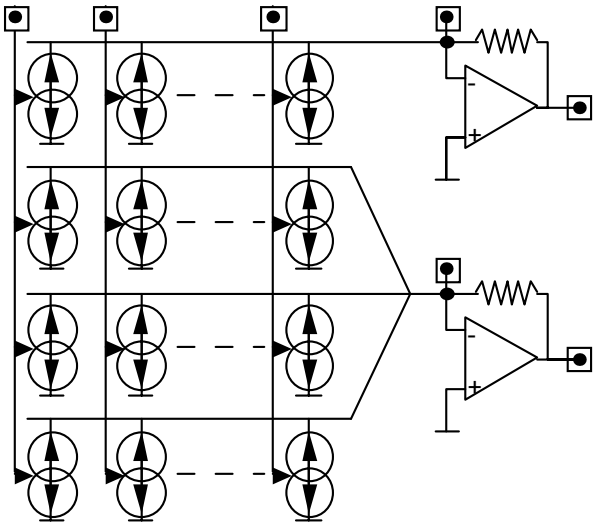


Figure V.1 : Principe des circuits.

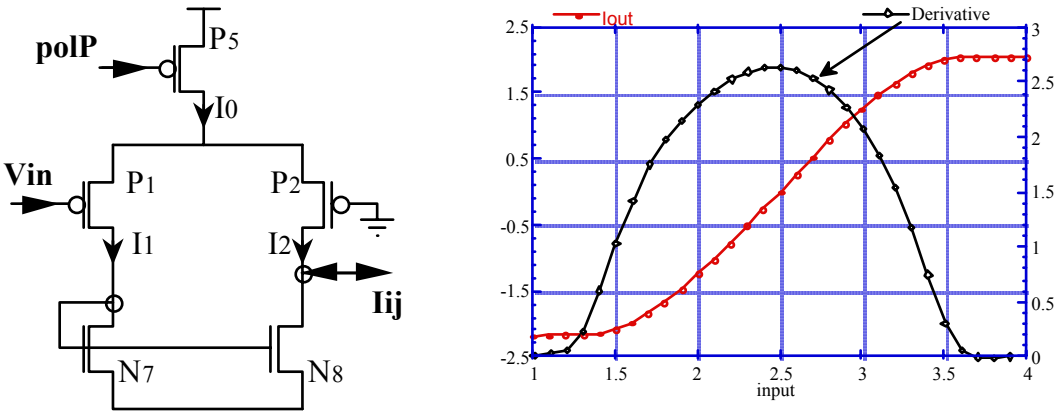


Figure V.2 : MBAT4 : schématique, courbe de transfert et dérivée.

La synapse du circuit MBAT4 est en fait une paire différentielle de transistors MOS, à large dynamique d'entrée mais sans correction de la linéarisation. La schématique d'une synapse, la caractéristique de transfert mesurée sur un prototype, ainsi que la dérivée de celle-ci sont représentées sur la Figure V.2.

La synapse du circuit MBAT5 est conforme au schéma de la Figure IV.2. La valeur du paramètre "a" est ici de 2,240 pour un rapport $ß_1/ß_3$ des tailles des transistors P_1 et P_3 de 4,96. Je n'ai pas eu recours, pour le layout, à l'utilisation d'un transistor de taille élémentaire. Une mesure de fonction de transfert est représentée sur la Figure V.3.

Les états de neurone sont entrés dans le circuit MBAT5 par un convertisseur série-parallèle 10 bits et le schéma général du circuit est représenté sur la Figure V.3.

Sur MBAT4 et MBAT5, la tension de polarisation polP des synapses (voir les Figures V.2 et IV.2) est générée par une cellule de polarisation telle que celle décrite sur la Figure IV.34.

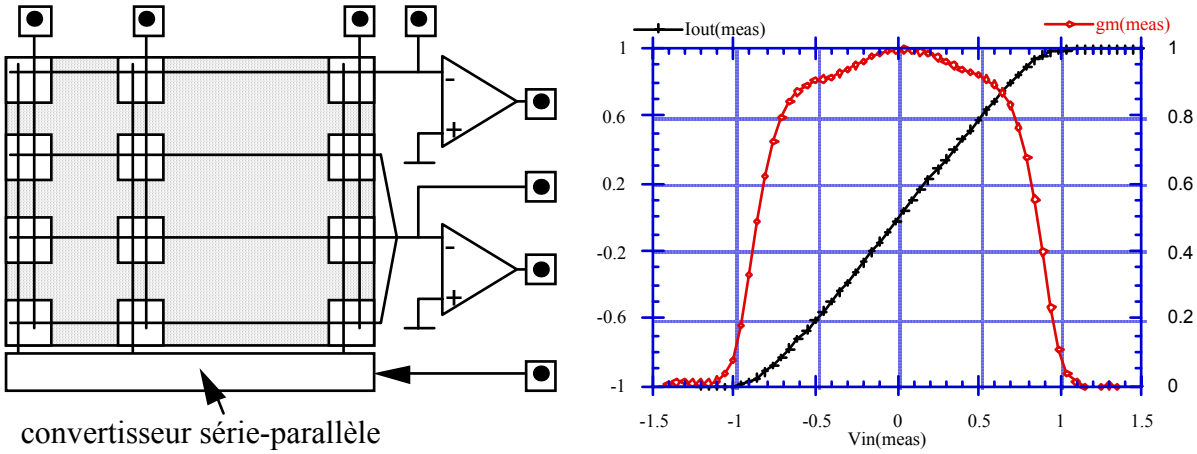


Figure V.3 : MBAT5 : schématique, courbe de transfert et dérivée.

Je vais maintenant présenter les tests de dispersion des caractéristiques effectués sur le circuit MBAT4 et les tests fonctionnels et de mise en cascade des cellules synapses effectués sur le circuit MBAT5. Je disposais de 40 circuits MBAT4 et 40 circuits MBAT5 montés sur boitier et j'ai pu dans les deux cas faire des mesures statistiques sur des lots de 38 circuits. Dans les deux cas, il s'est avéré que 2 circuits n'avaient pas des comportements standards et ils ont été rapidement écartés des tests.

La tension d'alimentation est 5V, la tension de référence est 2,5V (v_{ddgnd}). La masse réelle ("terre") pouvait être à 0V (gnd) ou à "v_{ddgnd}" selon l'essai à réaliser.

Le circuit MBAT4 a été soumis au MPW/Imec n°21 organisé en Janvier 1990. Le circuit MBAT5 a été soumis au MPW/Imec n°22 organisé en Mars 1990.

V.2.2- Test statique du circuit MBAT4.

La même tension est appliquée sur toutes les colonnes et on mesure un courant de sortie moyen sur la premiére ligne de 10 synapses, ceci dans le but de faire des mesures de dispersion entre circuits. Le courant de polarisation est 2µA et la résistance extérieure est de l'ordre de 100K.

La Figure V.2 représente la caractéristique de transfert du circuit étiqueté n°2 dans le lot, la dérivée de cette caractéristique calculée à partir de ces mesures est tracée sur le même graphique.

On peut voir sur la Figure V.4, les histogrammes donnant une idée des dispersions des offsets et des courants de saturation, la tension d'entrée de la cellule de polarisation étant constante.

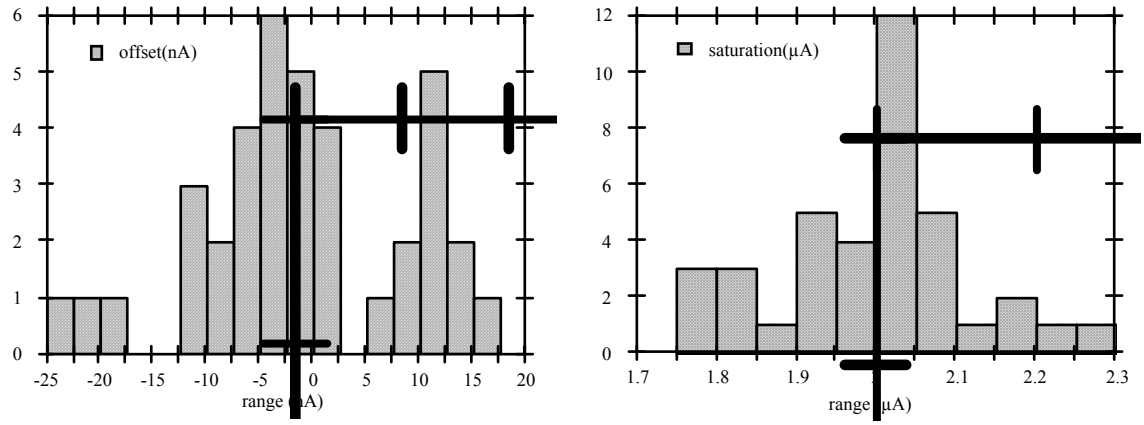


Figure V.4 : Histogramme des "offset" en courant et des courants de saturation.

On peut faire les observations suivantes sur les deux histogrammes :

a) les offsets sont assez dispersés, mais plus de la moitié d'entre eux sont limités à ±6nA par synapse (soit 0,3% du courant de saturation).

b) La faible taille des cellules synapses ainsi que leurs faibles courants ont pour conséquence une dispersion sur les courants de saturation de l'ordre de 15%. Cette valeur importante m'a conduit, pour le circuit MBAT5, à soigner le "layout" et j'ai ainsi pu mesurer une valeur inférieure à 8%. Je pense que cette dispersion est due, pour une grande part, à la cellule de polarisation et j'ai cherché à mieux optimiser sa conception pour le circuit MBA1.

Malgré tout, la polarisation de chaque circuit synapse doit être calibrée afin de maintenir le courant de saturation maximum constant. C'est un essai de ce type que j'ai cherché à faire dans le test suivant.

V.2.3- Test statique du circuit MBAT5

a- Présentation

Dans cet essai du circuit MBAT5 j'ai alimenté les 10 colonnes avec la même tension et observé la sortie de l'amplificateur connecté aux trois lignes de synapses, soit à 30 synapses. J'ai réglé la tension de polarisation de telle sorte que le courant de saturation ($2.I_0$) soit égal à 1µA par synapse (soit 30µA en tout) pour une tension d'entrée égale à v_{dd}. J'ai ainsi pu observer la dispersion entre circuits de la caractéristique de transfert moyenne des circuits pour une tension d'entrée comprise entre 1 et 4 Volt.

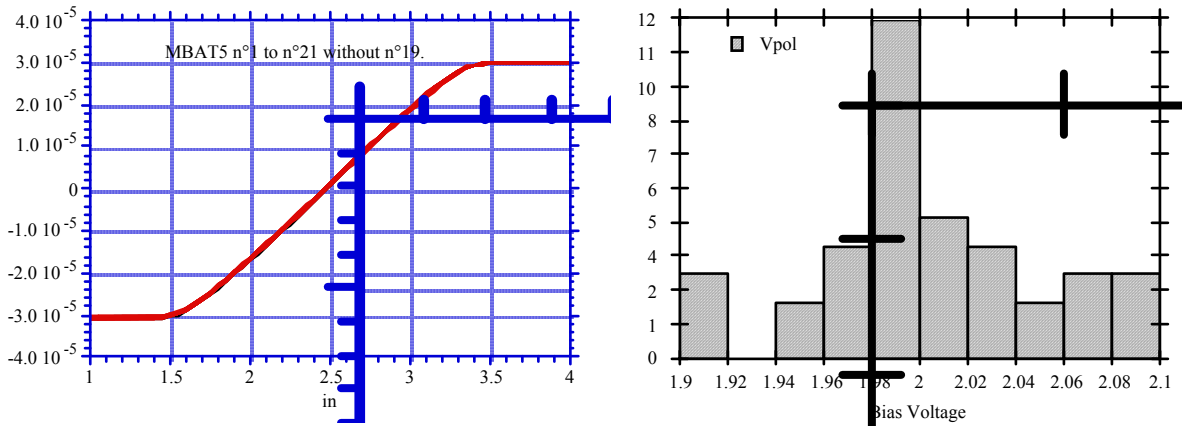


Figure V.5 : Superposition des fonctions de transfert (MBAT5) et histogramme typique des tensions de polarisation (MBAT4).

La Figure V.5 représente la superposition des caractéristiques de transfert de 20 circuits différents. Sur la même Figure, j'ai représenté un histogramme des tensions de polarisation nécessaires, ce relevé ayant toutefois été fait lors d'un essai semblable avec le circuit MBAT4.

b- dispersion des transconductances à l'origine.

Lors de cet essai, j'ai pu observer une très faible dispersion de la pente à l'origine des caractéristiques moyennes. La pente à l'origine représente 30 fois la

valeur moyenne de la transconductance à l'origine du convertisseur tension-courant. Elle dépend du rapport des tailles des transistors de dégérescence et de source de la paire différentielle du convertisseur tension-courant.

D'après les équations (IV.2) et (IV.3), on peut déduire aisément que cette pente est égale à :

$$\left(\frac{\partial I_{out}}{\partial V_{in}}\right)_{V_{in}=0} = \frac{1}{a} \cdot \sqrt{2I_0} \cdot \sqrt{ß_1} \qquad (V.1)$$

Rm : $V_{in} = 0$ correspond à une tension d'entrée réelle proche de 2,5V.

La transconductance à l'origine a été calculée d'après les mesures de fonctions de transfert décrites ci-dessus. Les statistiques de ces valeurs sur le lot de 39 circuits sont présentée sur le Tableau IV.7.

Minimum	1,20 $\mu A.V^{-1}$
Maximum	1,24 $\mu A.V^{-1}$
Points	39.000
Moyenne	1.2218 $\mu A.V^{-1}$
Ecart Type	0.013563 $\mu A.V^{-1}$ (1.1%)

Tableau IV.7 : Statistiques du gain à l'origine sur MBAT5.

On peut observer, sur ce Tableau, un écart type entre les valeurs de l'ordre de 1% et donc une dispersion relativement faible des caractéristique. Dans cet essai, le courant I_0 est constant et la dispersion des transconductances a pour origine une variation de $ß_1$ ou à une variation du paramètre "a". Pour le layout du convertisseur tension-courant utilisé dans le circuit MBA1, le paramètre "a" est réalisé par duplication d'un transistor de taille élémentaire et sa dispersion sera plus faible, améliorant ainsi la dispersion sur la transconductance.

V.2.4- Etude des dispersions internes du circuit MBAT5

Ce test était destiné à évaluer la dispersion des caractéristiques des synapses d'un circuit. Il a été fait sur la rangée de 10 synapses du circuit MBAT5.

Une même tension d'entrée est appliquée sur les entrées de poids des 10 synapses et une seule synapse est sélectionnée à la fois par son entrée d'inhibition. La

caractéristique de transfert de la synapse est alors relevée entre 1V et 4V par pas de 0,1V (soit au total 310 valeurs).

On peut voir sur la Figure V.6 diverses représentations de ces mesures.

La Figure (V.6.a) représente la superposition des caractéristiques relevées alors que la Figure (V.6.b) représentent la superposition des dérivées de ces caractéristiques.

La Figure (V.6.c) représente le courant de saturation (I_{sat}=2.I_0) en fonction du numéro de synapse alors que la Figure (V.6.d) représente la valeur de la dérivée du courant à l'origine.

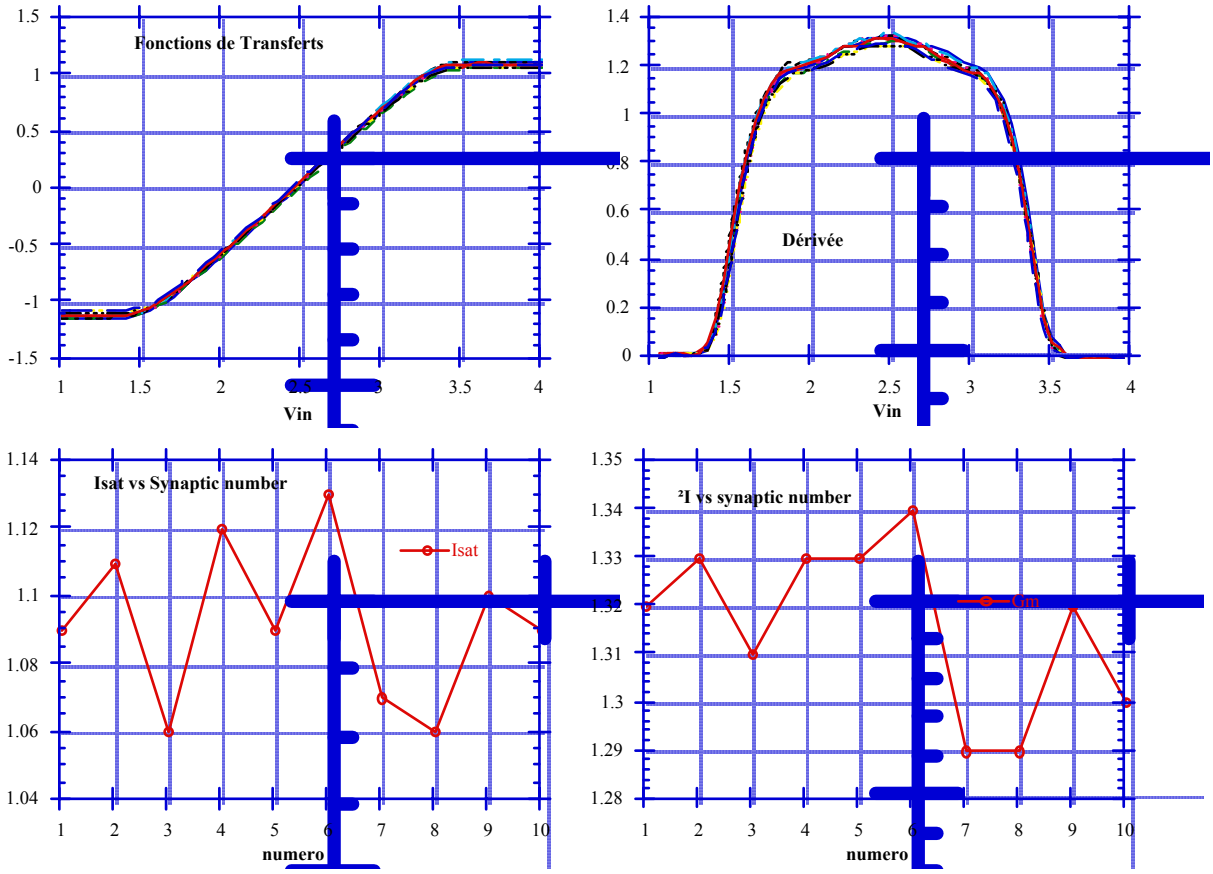


Figure V.6 : (**a** **b** / **c** **d**)

Le tableau IV.8 présente les statistiques des mesures du courant de saturation et de la dérivée à l'origine des Figures (V.6.c) et (V.6.d).

La dispersion sur le courant maximum traduit directement la dispersion des transistors P_5 et P_6, des miroirs de courant N_7-N_9, N_8-N_{10} et P_{11}-P_{12} de la Figure (IV.2). Sur ce circuit, la longueur L de ces transistors est de 10µm alors que

leur largeur est de 5µm. La dispersion observée est sans aucun doute due à leur faible largeur.

$Isat=2.I_0$		$(\partial I/\partial V_{in})_{V_{in}=0}$	
Minimum	1,06 µA	Minimum	1,29 $\mu A.V^{-1}$
Maximum	1,13 µA	Maximum	1,34 $\mu A.V^{-1}$
Points	10	Points	10
Moyenne	1,092 µA	Moyenne	1,316 $\mu A.V^{-1}$
Ecart Type	0,022 soit 2%	Ecart Type	0,017 soit 1,2%

Tableau IV.8 : Statistiques internes de MBAT5.

V.2.5- Test dynamique

a- Circuit MBAT4

J'ai fait le test dynamique du circuit MBAT4 en mettant les dix entrées de poids en parallèle sur le même générateur de tension. La tension de sortie observée est celle de l'amplificateur opérationnel de la rangée de 10 synapses. La tension d'entrée est une tension carrée à une fréquence de 100 Khz. La réponse indicielle du système est avec un dépassement de 23% et un temps de réponse à 5% de l'ordre de 4 µs, l'unité de temps pour l'utilisateur de ce convertisseur est de l'ordre de la µs. La Figure V.7 représente une vue d'écran d'oscilloscope de ce test dynamique.

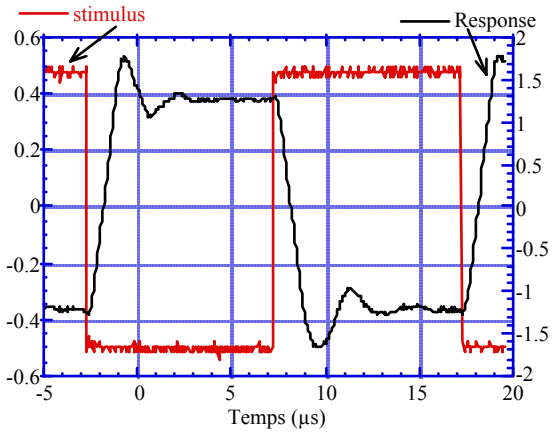


Figure V.7 : Image d'oscilloscope.

b- Circuit MBAT5

Les fonctionnalités du circuit MBAT5 nous permettaient de le tester dans des conditions similaires à une relaxation de la Machine de Boltzmann.

La Figure V.8 présente des vues d'oscilloscope relevées pendant ce test. Une capacité C a été placées en parallèle avec la capacité R de rebouclage de l'amplificateur opérationnel. Les valeurs de C, de R ainsi que la période d'horloge du convertisseur série-parallèle sont indiquées sur ces deux vues.

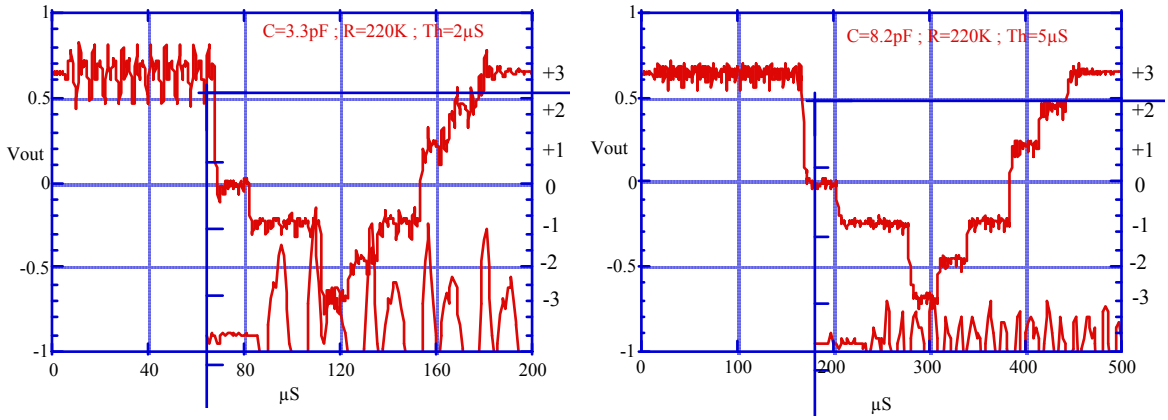


Figure V.8 ([a] - [b]) : Images d'oscilloscope

Le convertisseur série-parallèle étant initialisé avec des "1" (tous les neurones sont à 1), ce test commence par une mise à zéro des états de neurones (à l'instant t≈70µs pour la vue [a], et à t≈175µs pour la vue [b]).

Ensuite, à l'instant t≈80µs pour la vue [a] (t≈205µs pour la vue [b]), l'état d'un neurone est mis à "1", le poids du convertisseur tension-courant correspondant est à une valeur que nous appelerons "-1".

A l'instant t≈100µs pour la vue [a] (t≈250µs pour la vue [b]), les états de deux autres neurones sont mis à "1", un des poids valant "-1" et l'autre valant "+1". La contribution du réseau totale reste cependant égal à "-1".

A l'instant t≈110µs pour la vue [a] (t≈280µs pour la vue [b]), l'état d'un autre neurone est mis à "1", le poids correspondant valant "-2". La contribution du réseau prend alors la valeur "-3".

Aux instants t≈125µs et t≈135µs pour la vue [a] (t≈310µs et t≈340µs pour la vue [b]), l'état d'un autre neurone est mis à "1", le poids correspondant valant "+1". La contribution du réseau prend alors la valeur "-1".

A l'instant t≈155µs pour la vue [a] (t≈385µs pour la vue [b]), l'état de deux neurones est mis à "1", les poids correspondants valant "+1". La contribution du réseau prend alors la valeur à "+1".

Aux instants t≈165µs et t≈175µs pour la vue [a] (t≈415µs et t≈445µs pour la vue [b]), l'état d'un autre neurone est mis à "1", le poids correspondant valant "+1". La contribution du réseau prend alors la valeur à "+3".

Un poids de valeur "+1" correspondont à une tension de 0,5V au dessus de la tension de référence.

On peut voir sur les vues d'oscilloscope que le bruit numérique augmente avec le nombre d'états de neurones à "1". Les tensions de poids sont en effet apportées sur le circuit au travers de plots périphériques et elles sont ensuite connectées au convertisseur tension-courant par l'intermédiaire de fils en métal traversant tout le circuit et croisant donc les horloges du convertisseur série-parallèle. Sur le circuit MBA1, ceci ne se produira pas et le bruit sera donc beaucoup plus faible dans un circuit réel.

V.2.6- Comparaison Théorie-Simulations-Mesures

La Figure V.9 représente une superposition de courbes que j'ai pu tracé d'aprés :

- la théorie exposée au paragraphe (§ IV.2.3),
- la simulation électrique du convertisseur utilisé dans le circuit MBAT5 avec Spice2g6 et les paramètres technologiques fournis par Mietec pour ce run,
- les mesures faites sur le circuit MBAT5.

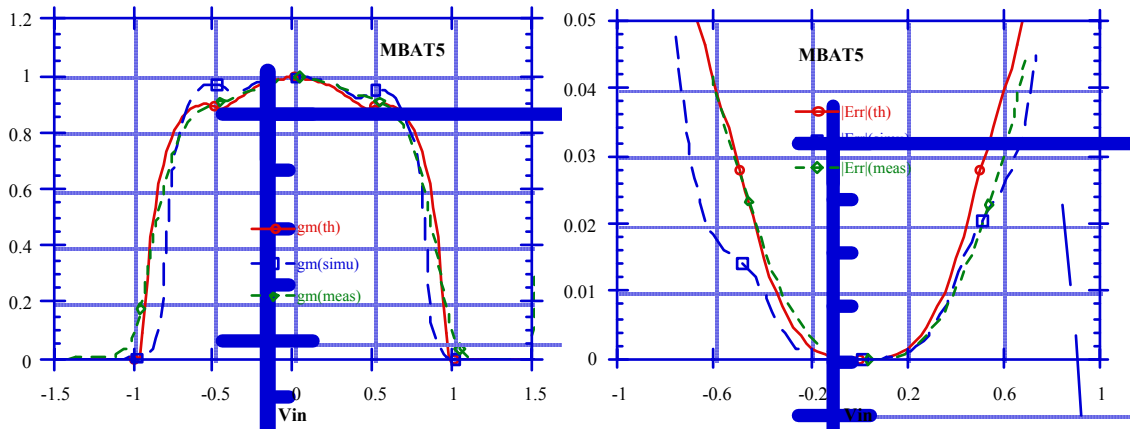


Figure V.9 ([a] - [b]) : Dérivée et Erreur absolue du courant.

V.3- TEST DU CIRCUIT MBAT8

Le circuit MBAT8 contient quatres cellules neurones à peu près semblables à celle représentée sur la Figure IV.17. Ce circuit a permis de valider l'utilisation d'un transistor bipolaire latéral pour la réalisation de la sigmoïde. Deux des cellules neurones ont été cablées avec les grilles des transistors bipolaires latéraux connectées à l'alimentation "vdd", alors que les deux autres sont connectées à un plot de test et elles peuvent être imposées par l'intermédiaire d'une pointe fine.

Ce circuit a montré que les grilles n'avaient pas besoin d'être connectées à une tension supérieure à vdd pour cette application et que la connexion à la tension d'alimentation était suffisante pour le bon fonctionnement de cet cellule.

La Figure V.10 représente la caractéristique sigmoïdale qui a pu être obtenue sur ce circuit en relevant la tension sur l'entrée aléatoire qui provoque le changement d'état du neurone pour une contribution du réseau donnée.

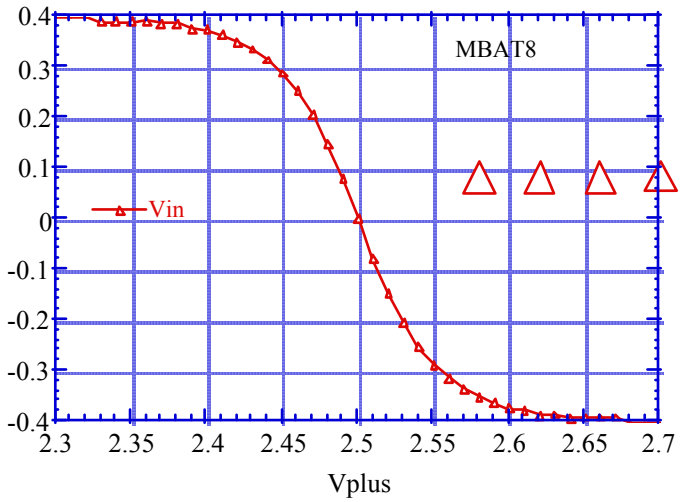


Figure V.10 : Test de la sigmoïde.

V.4- TEST DU XOR

Les circuits MBAT5, MBAT8 et MBAT6 ont été utilisés pour tester une relaxation sur l'exemple du "ou-exclusif" à architecture 2-1-1, selon le schéma de la Figure V.11.

Le circuit MBAT6 n'a pas été décrit ici. Il est constitué entre-autre d'un générateur pseudo-aléatoire à 8 cellules comme celui décrit au paragraphe (§ III.2.2.d), et d'un filtre/intégrateur décrit au paragraphe (§ III.2.2.b). Il a été conçu par Patrick Garda et testé par Zhu Yiming. Ce circuit a été soumis au MPW/Imec n°25 organisé en Septembre 1990.

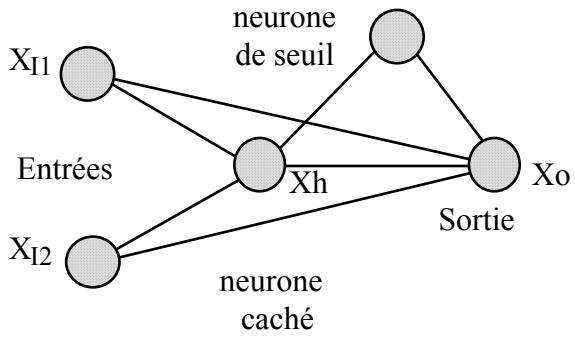


Figure V.11 : XOR 2-1-1.

Figure V.12 : Relaxation du XOR-2-1-1 avec les entrées (0,0).

Les Figures V.12 et V.13. représente une photo d'oscilloscope et une photo de l'écran du DAS (Digital Analysis System) de relaxations respectivement dans le cas où les deux entrées sont égales à 0 et dans le cas où les deux entrées sont différentes. On remarquera sur les photos du DAS : les entrées XI1 et XI2, le neurone caché Xh, la sortie Xo et la sortie du générateur aléatoire RANDOM.

La fréquence de relaxation a été limitée à 50kHz à cause des fortes capacités de la plaquette de test ayant servi au cablage de ce prototype.

Figure V.13 : Relaxation du XOR-2-1-1 avec les entrées (1,0).

V.5- CONCLUSION

Dans ce chapitre, j'ai présenté le test d'un certain nombre de cellules faisant partie des circuits MBA1 et MBA2.

Ensuite, j'ai présenté la relaxation d'un XOR construite en utilisant des petits circuits que nous avons réalisé.

Enfin, les premiers tests que j'ai pu faire sur le circuit MBA1 ont été présentés.

CONCLUSION

Dans ce manuscrit, j'ai décrit l'étude d'une réalisation analogique de la Machine de Boltzmann. Mon but était d'apporter un élément de réponse à la question suivante : Pourra-t-on un jour profiter à la fois de la grande puissance de calcul des circuits analogiques, et de la grande capacité d'intégration des technologies VLSI, pour le traitement de l'information grâce aux Réseaux de Neurones Formels ?

Au chapitre I, j'ai montré l'intérêt que présente l'étude de la Machine de Boltzmann pour répondre à cette question. On a vu tout d'abord que ce modèle est très long à simuler sur un ordinateur conventionnel et que la réalisation d'une machine spécialisée pour son expérimentation est donc très intéressante. De plus, le fait que les états de neurones soient binaires simplifie le test de ces circuits et leur intégration dans un système. Enfin cet algorithme présente de meilleures performances de reconnaissances des formes que les autres modèles [Koh89].

Au chapitre II, j'ai présenté un état de l'art succinct des réalisations de Réseaux de Neurones Formels au début de cette étude

Au chapitre III, j'ai tout d'abord montré comment les équations de la Machine de Boltzmann peuvent être calculées physiquement par un système analogique et j'ai pour cela défini deux cellules de base : la cellule neurone et la cellule synapse. J'ai décrit ces cellules par des blocs fonctionnels. J'ai ensuite montré comment ces deux cellules peuvent être assemblées dans une architecture cohérente et simple à mettre en œuvre. Ceci m'a conduit à la définition de deux types d'architectures de matrices synaptiques : monodirectionnelles et bidirectionnelles.

J'ai ensuite présenté l'architecture d'unités de traitement analogiques dont la taille est adaptée aux applications à traiter et qui peuvent être construites par assemblage de plusieurs circuits analogiques cascadables. Les interfaces de ces unités

de traitement sont uniquement numériques et elles peuvent donc être interconnectées par un réseau numérique programmable.

J'ai aussi étudié une méthode de détermination de la taille de la matrice synaptique à implanter dans chaque circuit en fonction de la technologie de fabrication.

Dans le chapitre IV, j'ai présenté la conception des éléments constituants les cellules neurone et synapse avec une technologie CMOS. J'ai donc présenté tout d'abord la conception du convertisseur tension-courant qui est utilisé dans la synapse et qui est réalisé à l'aide d'une paire différentielle linéarisée. J'ai présenté ensuite celle de deux convertisseurs courant-tension et de la fonction sigmoïde qui sont nécessaires pour la réalisation de la cellule neurone.

J'ai ensuite décrit les cellules qu'il faut adjoindre à la synapse pour lui fournir des capacités d'apprentissage de son poids : compteur analogique et comparateur. Enfin pour pouvoir calibrer aisément la conception de ces cellules, j'ai fait une étude originale de la réalisation de la température de relaxation T.

Enfin, j'ai présenté les circuits prototypes MBA1 et MBA2 que j'ai réalisés et qui servirons de base à un module de traitement complet sans apprentissage.

Dans le chapitre V, le test des premiers prototypes des cellules décrites au chapitre VI est présenté. Une grande part de ce chapitre est dédiée à l'étude complète des dispersions entre synapses d'un même circuit et des dispersions entre circuits. J'ai aussi pu tester la mise en cascade de plusieurs circuits ce qui permet de répartir la contribution du réseau sur un neurone sur plusieurs circuits et de construire des modules de traitement plus grands.

Enfin, l'expérimentation d'une relaxation de l'exemple du XOR est présentée. Elle a été réalisée par interconnexion de plusieurs circuits prototypes n'incluant chacun qu'une petite partie des fonctionnalités nécessaires.

Les éléments de calcul que j'ai présenté dans ce manuscrit sont voisins par certaines caractéristiques de ceux développés dans les autres laboratoires de recherche. Ainsi, nous obtenons des densités de cellules et des vitesses de relaxation comparables à celles des circuits contemporains réalisés sur une technologie CMOS 2µm, [Als90] (31x32 synapses et 32 Neurones), [Ros89] (5x5 synapses et 15mm^2), [Hol89].

Par contre cette étude se distingue par un certain nombre de points très importants. Nous avons tout d'abord construit des cellules fidèles au modèle mathématique afin d'assurer la convergence de la machine. Nous avons ainsi choisi

un générateur aléatoire de grande qualité statistique et nous avons amélioré la linéarité des différentes cellules.

De cette manière, nous avons aussi voulu dépasser les problèmes de précision liés à l'analogique et mis en évidence par une équipe d'ATT dans [Bos91] où une partie des connections doivent être réalisés en numérique et où les taux d'erreur de reconnaissance augmentent de 5% à cause de l'utilisation de l'analogique. C'est pour valider ce choix que nous avons conçu les circuits MBA1 et MBA2.

Dans les réseaux entièrement connectés, le nombre de synapses en entrée d'un neurone augmente linéairement avec le nombre total N de neurones (la taille du réseau). Au contraire, dans les réseaux multi-couches, le nombre de synapses en entrée d'un neurone est en pratique limité à 128 ou 256. Nous avons pu ainsi utiliser des techniques de calcul analogique sans rencontrer le problème d'imprécision mis en évidence sur le projet MIND-128 [Gam91].

Nous utilisons de plus l'algorithme synchrone développé par Azencott qui rend valide la commutation simultanée des neurones et permet de s'affranchir des problèmes de convergence posés par l'utilisation du modèle de Hinton et Sejnowski dans ces conditions et qui ont été cités par Goser dans [Kre88].

Une prise en compte plus complète du modèle mathématique permet aussi de faire des systèmes plus grands. Là encore, contrairement à d'autres études, la réalisation d'un grand système est rendue possible par l'adjonction d'un réseau d'interconnexions numérique programmable à nos unités de traitement analogique. Nous avons en effet, dès le début de l'étude, pris en compte cet aspect alors que d'autres ont choisi de reporter le problème [Als87a].

Les perspectives de ce travail sont tout d'abord de réaliser un système avec MBA1 et MBA2 et de chercher à valider les choix de conception énoncés ci-dessus.

Les progrès des technologies CMOS permettront de plus la réalisation de grands systèmes dont nos choix de conception garantiront le fonctionnement. Le passage d'une technologie 2,4µm à une technologie 0,8µm se traduit par exemple par une multiplication de toutes les dimensions par 1/3 et ceci donne donc une surface d'encombrement égale à environ 1/9 de la surface originale.

Dès aujourd'hui, en fonction des résultats que nous avons déjà obtenus, nous pouvons envisager la conception d'un circuit plus gros et incluant toutes les fonctionnalités d'une unité de traitement. En se référant aux tailles des prototypes réalisés sur des technologies 2,4µm et 3µm, nous estimons qu'il sera possible de réaliser une unité de traitement incluant une matrice de 64x300 synapses et 64 Neurones en utilisant une technologie 0,8µm. De tels circuits rendent alors possible

l'accomplissement de tâches de reconnaissances avec des applications de la taille de NETtalk en moins de 500µs et en utilisant seulement deux circuits.

Enfin, le choix du générateur aléatoire opto-électroniques ou du générateur aléatoire pseudo-aléatoire à automate cellulaire nous permet d'envisager un construction de Machines de Boltzmann à temps continu. En effet, le modèle de le machine de Boltzmann que nous avons utilisé ici est à temps discret (les circuits sont commandés par une horloge et les états des neurones sont échantillonnés). La première évolution de cette étude sera d'abandonner tout séquencement et de concevoir des circuits à temps continu. Le circuit MBA1 a été conçu pour permettre ce type de fonctionnement.

Ce mode de fonctionnement ouvre de réelles perspectives de recherche pour les circuits analogiques cellulaires. En effet, le temps de reconnaissance d'une application sera alors diminué d'au moins un ordre de grandeur. De plus, les signaux ne seront plus synchronisés les uns par rapport aux autres, et ceci est bien adapté aux circuits intégrés construits sur une grande surface où cette synchronisation devient justement difficile à conserver. Cependant ceci pose de nouvelles difficultés de modélisation et d'observation de l'expérience.

Une deuxième évolution de cette étude sera la conception de systèmes à états de neurones continus. Les sorties des cellules neurones ne seront plus alors seulement binaires, mais pourront prendre n'importe laquelle d'un nombre fini (ou infini) de valeurs. Les signaux d'entrée et de sortie de ces circuits seront alors analogiques ce qui est le mode de communication normal des capteurs et autres transducteurs. La quantité d'informations par ligne de communication est alors maximale et on minimisera ainsi le nombre d'interconnexions entre les différentes parties d'un même système. De plus, ce mode de fonctionnement devrait permettre de diminuer la consommation des cellules neurones et de concevoir ainsi des circuits contenant plus de cellules à puissance consommée constante. La modélisation et l'expérimentation de ces systèmes multivalués et à temps continu sont encore très peu abordées, on accède ici à un domaine où un grand nombre de problèmes restent ouverts.

REFERENCES BIBLIOGRAPHIQUES

[Als87a] J. Alspector et R.B. Allen,
"A Neuromorphic VLSI Learning System".
Dans *Advanced Research in VLSI,* (P. Losleben, Ed.), MIT Press, Cambridge, MA, Proceedings of the 1987 Stanford Conference on VLSI, pp. 313-349, 1987.

[Als87b] J. Alspertor, R. Allen, V. Hu, et S. Satyarayana,
"Stochastic learning networks and their electronic implementation".
Dans *Proceedings of NIPS 87,* American Institut of Physics, 1987.

[Als89] J. Alspector, B. Gupta, et R.B. Allen,
"Performance of a Stochastic Learning Microchip".
Dans *Advances in Neural Information Processing Systems I,* (D. Touretzky, Ed.), pp. 748-760, 1989.

[Als90] J. Alspector, R.B. Allen, A. Jayakumar, T. Zeppenfeld, et R. Meir,
"Relaxation Networks for Large Supervised Learning Problems".
Dans *Proceedings of NIPS,* 1990.

[Arb87] M.A. Arbib,
"*Brains, Machines and Mathematics",*
Springer-Verlag, 1987.

[Ari91a] Y. Arima, K. Mashiko, K. Okada, T. Yamada, A. Maeda, H. Kondoh, et S. Kayano,
"A Self-Learning Neural Network Chip with 125 Neurons and 10K Self-Organization Synapses".
IEEE Journal of Solid State Circuits, vol. 26, pp. 607-611, Avril 1991.

[Ari91b] Y. Arima, K. Mashiko, K. Okada, T. Yamada, A.M.H. Notani, H. Kondoh, et S. Kayano,
"A 336-Neuron, 28K-Synapse, Self-Learning Neural Network Chip with Branch-Neuron-Unit Architecture".
IEEE Journal of Solid State Circuits, vol. 26, pp. 1637-1644, Novembre 1991.

[Aze90a] R. Azencott,
"Synchronous Boltzmann Machines and Gibbs Field : Learning Algorithms".
Dans *Neurocomputing,* (F. Fogelman-Soulié et J. Hérault, Eds.), Springer-Verlag, pp. 51-63, 1990.

[Aze90b] R. Azencott,
"Synchronous Boltzmann Machines and Artificial Vision".
Dans *Proc. of the Conf. Entretiens de Lyon on Neural Networks,* Springer Verlag, Paris, 1990.

[Aze90c] R. Azencott, A. Doutriaux, et L. Younes,
"Synchronous Boltzmann Machines and outline based classification".
Dans *International Neural Network Conference 90 Paris,* IEEE, Kluwer Academic Publishers, pp. 7-10, Juillet 1990.

[Bel89] E. Belhaire et P. Garda,
"Key points for a fully analog implementation of Boltzmann Machines".
Dans *IFIP Workshop on Parallel Architecture on Silicon,*
Grenoble, France, Decembre 1989.

[Bel90] E. Belhaire et P. Garda,
"An analog Chip Set for Synchronous Boltzmann Machines".
Dans *ITG/IEEE Workshop on Microelectronics for Neural Networks,*
Dortmund, RFA, pp. 80-87, Juin 1990.

[Bel91] E. Belhaire et P. Garda,
"Design of a Linear Sum-of-Product Circuit for Boltzmann Machines".
Dans *IEEE International Symposium on Circuits and Systems,* Singapore,
pp. 1299-1302, Juin 1991.

[Bes91] P. Bessiere, A. Chams, A. Guérin, J. Herault, C. Jutten, et J.C. Lawson,
"From Hardware to Software: Designing a Neurostation".
Dans *VLSI Design of Neural Networks,* (U. Ramacher et U. Rückert, Eds.),
Kluwer Academic Publishers, Dortmund RFA, pp. 311-335, 1991.

[Bos91] B.E. Boser, E. Säckinger, J. Bromley, Y. LeCun, et L.D. Jackel,
"An Analog Neural Network Processor with Programmable Topology".
IEEE Journal of Solid State Circuits, vol. 26, n° 12, pp. 2017-2025, Decembre 1991.

[Can86] M. Cand, E. Demoulin, J.L. Lardy, et P. Senn,
"*Conception de Circuits Intégrés MOS : éléments de bases - perspectives*",
Eyrolles et CNET-ENST, 1986.

[Coh85] J. Cohen,
"Application of an adaptive auditory model to speech recognition".
Dans *110th Meeting of the Acoustical Society of America,* Novembre 1985.

[Darpa88] "*DARPA, Neural Network Study*",
AFCEA International Press, Novembre 1988.

[Dev91] F. Devos,
"Retines Intelligentes".
Seminaire de Laboratoire, IEF, Orsay, France, Mars 1991.

[Dur89] M. Duranton, J. Gobert, et N. Mauduit,
"A Digital VLSI Module for Neural Networks".
Dans *I.D.S.E.T.,* (L. Personnaz et G. Dreyfus, Eds.), Paris, pp. 720-724, 1989.

[Ebe88] S. Eberhardt, A. Moopenn, et A. Thakoor,
"Considerations for hardware implementations of neural networks".
Dans *22nd Asilomar Conference,* Maple Press, Pacific Grove,
pp. 649-653, Novembre 1988.

[Gam91] C. Gamrat, A. Mougin, P. Peretto, et O. Ulrich,
"The Architecture of MIND Neurocomputers".
Dans *2nd International Workshop on Microelectronics for Neural Networks,*
Munich, pp. 463-469, 1991.

[Gar90] P. Garda et E. Belhaire,
"An analog chip set with digital I/O for Synchronous Boltzmann Machine".
Dans *VLSI for ArtificiaI Intelligence and Neural Network*,
(J.G. Delgado-Frias et W.R. Moore, Eds.), Boston, Kluwer Academic, 1990.

[Gra86] H.P. Graf, L. Jackel, R. Howard, B. Straughn, J. Denker, W. Hubbard, et D. Schwartz,
"VLSI Implementation of a Neural Network Memory With Several Hundreds of Neurons".
Dans *AIP Conference Proceedings on Neural Network for Computing,* (J.S. Denker, Ed.),
American Institut of Physics, Snowbird, Utah, pp. 183-187, 1986.

[Gue89] A. Guerin, C. Jutten, et J. Herault,
"Neurocomputer : CRASY a Cost-Effective Solution".
Dans *I.D.S.E.T.,* (L. Personnaz et G. Dreyfus, Eds.), Paris, pp. 756-765, 1989.

[Heb49] D.O. Hebb,
"The Organization of the Behavior",
Wiley, 1949.

[Hin84] G.E. Hinton, T.J. Sejnowski, et D.H. Ackley,
"Boltzmann Machines: Constraint satisfaction networks that learn".
Tech. Rept. CMU-CS-84-119, Carnegie-Mellon University, Pittsburgh,
PA 15213, Mai 1984.

[Hin85] G.E. Hinton et T.J. Sejnowski,
"Learning in Boltzmann machines".
Dans *Cognitiva 85,* Paris, pp. 283-290, 1985.

[Hin86] G.E. Hinton et T.J. Sejnowski,
"Learning and Relearning in Boltzmann Machines".
Dans *Parallel Distributed Processing*, (D.E. Rumelhart et J.L. McClelland, Eds.),
pp. 282-317, Cambridge, MA, Bradford-MIT Press, 1986.

[Hol89] M. Holler, S. Tam, H. Castro, et R. Benson,
"An Electrically Trainable Artificial Neural Network (ETANN) with 10240 Floating Gate Synapses".
Dans *Proceedings of the International Annual Conference on Neural Networks,*
IEEE, pp. II-191--II-196, 1989.

[Hop82] J.J. Hopfield,
"Neural networks and physical systems with emergent collective computational abilities".
Proceedings of the National Academy of Science of USA, vol. 79,
pp. 2554-2558, Avril 1982.

[Koh89] T. Kohonen, R. Chrisley, et G. Barna,
"Statistical Pattern Recognition with Neural Networks : Benchmarking Studies".
Dans *Neural Networks from Models to Applications,* (L. Personnaz et G. Dreyfus, Eds.),
I.D.S.E.T., Paris, pp. 160-167, 1989.

[Kre88] I. Kreuzer et K. Goser,
"A Modified Model of Boltzmann Machines for WSI-Realization".
Dans *Signal Processing IV : Theories and Applications*, (J.L. Lacoume, A. Chenikian, N. Martin, et J. Malbors, Eds.), pp. 327-330, Elsevier Science B.V, North-Holland, 1988.

[Kru88] F. Krummenacher et N. Joehl,
"A 4 Mhz CMOS Continious-Time Filter with On-Chip Automatic Tuning".
IEEE Journal of Solid-State Circuits, vol. 23, pp. 750-758, Juin 1988.

[Kru89] F. Krummenacher,
"Design Considerations in High Frequency CMOS Transconductance Amplifier Capacitor (TAC) Filters".
Dans *IEEE International Symposium on Circuits and Systems,*
Portland, Oregon, Mai 1989.

[Lal89] P. Lalanne,
"Les réseaux de neurones formels et leurs réalisations optoélectroniques. Génération optique de tableaux de nombres aléatoires",
Thèse de doctorat, Université de Paris XI - Orsay, 1989.

[Lal90] P. Lalanne, J.C. Rodier, H. Richard, P. Chavel, E. Belhaire, K. Madani, et P. Garda,
"2-D Optical Generator of Updating Probabilities for VLSI Implementation of Boltzmann Machines.".
International Journal of Optical Computing, vol. 1, pp. 25-30, 1990.

[Mad90] K. Madani,
"Etude de structure électroniques pour réseaux de processeurs stochastiques",
Thèse de doctorat, Université de Paris XI - Orsay, 1990.

[Mad91] K. Madani, P. Garda, E. Belhaire, et F. Devos,
"Two Analog Counters for Neural Networks Implementation".
IEEE Journal of Solid State Circuits, vol. 26, pp. 966-974, Juillet 1991.

[McC43] W. McCulloch et W. Pitts,
"A logical calculus of the ideas immanent in Nervous Activity".
Bulletin of Mathematical Biophysics, vol. 5, pp. 115-137, 1943.

[Mea89a] "*Analog VLSI Implementation of Neural Systems"*,
(C. Mead et M. Ismail, Eds.), MA: Kluwer Academic, 1989.

[Mea89b] C.A. Mead,
"*Analog VLSI and neural systems"*,
Addison Wesley, 1989.

[Mez90] M. Mezard et J.P. Nadal,
"Réseaux de Neurones et Physique Statistique".
Le Courrier de CNRS : Image de la Physique, 1990.

[Min69] M.L. Minsky et S.A. Papert,
"*Perceptrons"*,
MA: MIT Press, 1969.

[Min88] M.L. Minsky et S.A. Papert,
"Perceptrons - Expanded Edition",
MA: MIT Press, 1988.

[Nai90] P. Naish et P. Bishop,
"Conception des ASICs",
Masson, Traduit de l'anglais par F. Devos, T. Maurin et A. Mérigot, 1990.

[Oua91] J. Ouali, G. Saucier, et J. Trilhe,
"Fast Design of Digital Dedicated Neuro Chips".
Dans *VLSI Design of Neural Networks,* (U. Ramacher et U. Rückert, Eds.),
Kluwer Academic Publishers, Dortmund RFA, pp. 187-203, 1991.

[Par85] D.B. Parker,
"Learning Logic".
Tech. Rept. TR-47, Center for Computational Research in Economics and Management Science, MIT, Avril 1985.

[Per89] L. Personnaz, A. Johannet, G. Dreyfus, et M. Weinfeld,
"Toward a Neural Network Chip : a Performance Assessment and a Simple Example".
Dans *I.D.S.E.T.,* (L. Personnaz et G. Dreyfus, Eds.), Paris, pp. 682-691, 1989.

[Per90] P. Peretto, R.V. Zurk, A. Mougin, et C. Gamrat,
"The Semi-Parallel Architectures of Neuro-Computers".
Dans *Neurocomputing : Algorithms, Architectures and Applications,*
(F. Fogelman et J. Hérault, Eds.), Springer-Verlag, pp. 195-204, 1990.

[Puj91a] H. Pujol, E. Belhaire, et P. Garda,
"Local Interconnection Through Splitted Busses for Synchronous Boltzmann Machines".
Dans *Artificial Neural Networks : Proceeding of ICANN-91,* (T. kohone , K. Mäkisara, O. Simula, et J. Kangas, Eds.), Elvesier Science Publishers B.V. (North-Holland), Espoo, Finland, pp. 679-684, Juin 1991.

[Puj91b] H. Pujol, E. Belhaire, et P. Garda,
"Local Interconnection Through Splitted Busses for Multilayered Boltzmann Machines".
Dans *Proceedings of 3rd symposium sur les architectures nouvelles de machines,*
CNRS© - PRC ANM, Palaiseau, pp. 197-207, Juin 1991.

[Rod92] J.C. Rodier,
"Thèse d'Ingénieur CNAM en cours",
Master's thesis, CNAM, 1992.

[Ros62] F. Rosenblatt,
"Principle of Neurodynamics: Perceptron and the Theory of Brain Mechanisms",
Spartan Books, 1962.

[Ros89] O. Rosetto, C. Jutten, J. Herault, et I. Kreuzer,
"Analog VLSI Synaptic Matrices".
IEEE Micro, vol. 9, pp. 56-63, Decembre 1989.

[Rum86] D.E. Rumelhart, G.E. Hinton, et R.J. Williams,
"Learning Internal Representations by Error Propagation".
Dans *Parallel Distributed Processing*, (D.E. Rumelhart et J.L. McClelland, Eds.),
Ch. 8, pp. 318-362, Cambridge, MA: MIT Press, 1986.

[Sej87] T.J. Sejnowski et C.R. Rosenberg,
"Parallel Networks that Learn to Pronounce English Text".
Complex systems, vol. 1, pp. 145-168, 1987.

[Siv86] M.A. Sivilotti, M.R. Emerling, et C.A. Mead,
"VLSI Architectures for Implementation of Neurals Networks".
Dans *AIP Conference Proceedings on Neural Network for Computing,* (J.S. Denker, Ed.),
American ®Institut of Physics, Snowbird, Utah, pp. 408-413, 1986.

[Sku90] M. Skubiszewski,
"A Hardware Emulator for Binary Neural Networks".
Dans *International Neural Network Conference 90 Paris,* IEEE,
Kluwer Academic Publishers, Paris, pp. 555-558, Juillet 1990.

[The90] J.B. Theeten, M. Duranton, N. Mauduit, et J.A. Sirat,
"The LNEURO-CHIP: A Digital VLSI With On-Chip Learning Mechanism".
Dans *International Neural Network Conference 90 Paris,* IEEE,
Kluwer® Academic Publishers, Paris, pp. 593-596, Juillet 1990.

[Tom90] J. Tomberg, H. Raittinen, et K. Kaski,
"VLSI Architecture of the Boltzmann Machine Algorithm".
Dans *International Neural Network Conference 90 Paris,* IEEE,
Kluwer ®Academic Publishers, Paris, pp. 568-571, Juillet 1990.

[Tsi85] "*Design of MOS VLSI Circuits for Telecommunications*",
(Y. Tsividis et P. Antognetti, Eds.), Prentice-Hall, 1985.

[Tsi86] Y. Tsividis, M. Banu, et J. Khoury,
"Continous-Time MOSFET-C Filters in VLSI".
IEEE Transactions on Circuits And Systems, vol. CAS-33, pp. 125-140, 1986.

[Tsi87] Y.P. Tsividis,
"Analog MOS integrated circuits-Certain new ideas, trends and obstacles".
IEEE Journal of Solid-State Circuits, vol. SC-22, pp. 317-321, 1987.

[Ver89] M. Verleysen et P. Jespers,
"An Analog VLSI Implementation of Hopfield's Neural Network".
IEEE Micro, vol. 9, Decembre 1989.

[Ver90] M. Verleysen et P. Jespers,
"Precision of Sum-of-Product in Analog Neural Network".
Dans *Proceedings of the 1st International Workshop on Microelectronics for Neural Networks,*
Dortmund, RFA, Juin 1990.

[Vit83] E.A. Vittoz,
"MOS Transistors Operated in the Lateral Bipolar Mode and Their Application in CMOS Technology".
IEEE Journal of Solid-state Circuits, vol. SC-18, pp. 273-279, Juin 1983.

[Vit85] E.A. Vittoz,
"Micropower techniques".
Dans *Design of MOS VLSI Circuits for Telecommunications*, (Y. Tsividis et P. Antognetti, Eds.), Ch. 4, Englewood Cliffs, NJ, Prentice-Hall, 1985.

[Vit89a] E. Vittoz et X. Arreguit,
"CMOS integration of Herault-Jutten cells for separation of sources".
Dans *Analog VLSI implementation of neurals systems*, (C.A. Mead et M. Ismaïl, Eds.), Ch. 3, pp. 57-84, Kluwer Academic, 1989.

[Vit89b] E. Vittoz, E. Sorouchyari, P. Heim, X. Arreguit, et F. Krummenacher,
"Analog VLSI implementation of a Kohonen Map".
Dans *Journées d'Electronique de Lausanne,* Ecole®Polytechnique Fédérale de Lausanne, Presses Polytechniques Romandes, pp. 291-302, Octobre 1989.

[Wan90a] Z. Wang,
"2-Mosfet transresistor with extremely low distortion for output reaching supply voltage".
Electronics Letters, vol. 26, pp. 951-952, Juin 1990.

[Wan90b] Z. Wang,
"Wideband Class AB (Push-Pull) Current Amplifier in CMOS Technology".
Electronics Letters, vol. 26, pp. 543-545, Avril 1990.

[Wei90] M. Weinfeld,
"Integrated Artificial Neural Networks : Components for Higher Level Architectures with New Properties".
Dans *Neurocomputing : Algorithms, Architectures and Applications,*
(F. Fogelman et J. Hérault, Eds.), Springer-Verlag, pp. 123-130, 1990.

[Wid60] G. Widrow et M.E. Hoff,
"Adaptative switching circuits".
Dans *Western Electric Show and Convention, Convention Record, Part 4,*
Institut of Radio Engineers, pp. 96-104, 1960.

[Wol86] S. Wolfram,
"*Theory and Applications of Cellular Automata",*
World Scientific Publishing Co. Pte. Ltd., 1986.

[Zhu91] Y. Zhu, P. Garda, et Y. Ni,
"A Novel CMOS Analog Counter for Boltzmann Machines".
Dans *IEEE International Symposium on Circuits and Systems,* Singapore, Juin 1991.

BIBLIOGRAPHIE

Livres sur les Réalisation de Circuit Intégrés Analogiques CMOS

- P.R. Gray et R.G. Meyer,
  "*Analysis and Design of Analog Integrated Circuits*",
  John Wiley & sons, 2nd, 1984.

- "*Design of MOS VLSI Circuits for Telecommunications*",
  (Y. Tsividis et P. Antognetti, Eds.), Prentice-Hall, 1985.

- Y.P. Tsividis,
  "*Operating and Modeling of the MOS transistor*",
  Mc Graw-Hill International, 1988.

Livres sur les Réseaux d'automates et de Neurones Formels

- S. Wolfram,
  "*Theory and Applications of Cellular Automata*",
  World Scientific Publishing Co. Pte. Ltd., 1986.

- "*AIP Conference Proceedings on Neural Network for Computing*",
  (J.S. Denker Ed.), American Institut of Physics, Snowbird, Utah, 1986.

- "*Advanced Research in VLSI : proceedings of the 1987 Stanford Conference on Very Large Scale Integration*",
  (P. Losleben Ed.), MIT Press, 1987.

- "*DARPA, Neural Network Study*",
  AFCEA International Press, Novembre 1988.

- "*Neural Networks from Models to Applications*",
  (L. Personnaz et G. Dreyfus, Eds.), I.D.S.E.T., 1989.

- "*Analog VLSI Implementation of Neural Systems*",
  (C. Mead et M. Ismail, Eds.), MA: Kluwer Academic, 1989.

- C.A. Mead,
  "*Analog VLSI and neural systems*",
  Addison Wesley, 1989.

- "*Neurocomputing : Algorithms, Architectures and Applications*",
  (F. Fogelman et J. Hérault, Eds.), Springer-Verlag, Vol. F 68, NATO ASI Series, 1990.

- *"VLSI Design of Neural Networks"*,
  (U. Ramacher et U. Rückert, Eds.), Kluwer Academic Publishers, 1991.

Articles interessants et non cités dans le manuscrit

- P. Lalanne,
  "Optical Implementation of Neural Networks : state-of-the-art and perspectives",
  Dans *Journées d'Electronique de Lausanne,* Ecole Polytechnique Fédérale de Lausanne,
  pp. 251-264, Octobre 1989.

- E.A. Vittoz,
  "The design of high performance analog circuits on digital CMOS Chips",
  IEEE Journal of Solid-state Circuits, vol. SC-20, pp. 657-665, Juin 1985.

- E. Vittoz, H. Oguey, M.H. Maher, O. Nys, E. Dijktra, et M. Chevrolet,
  "Analog Storage of Adjustable synaptic weights",
  Dans *Proceedings of the First Internationnal Workshop on MicroElectronics for Neural Networks,*
  Dortmund RFA, pp. 69-79, Juin 1990.

- A.F. Murray et A.V. Smith,
  "Asynchronous VLSI neural networks using pulse stream arithmetic",
  IEEE Journal of Solid State Circuits, vol. 23, pp. 688-697, 1988.

- A.F. Murray, A. Hamilton, H.M. Reekie, S. Churcher, Z. Butler, et L. Tarassenko,
  "Innovations in pulse stream neural VLSI - arithmetic and communications",
  Dans *Proceedings of the First Internationnal Workshop on MicroElectronics for Neural Networks,*
  Dortmund, pp. 8-27, Juin 1990.

- *"Neural Network Solutions"*,
  Intel Corporation, Santa Clara, CA, 1991.

- B.E. Boser, E. Säckinger, J. Bromley, Y. LeCun, R.E. Howard, et L.D. Jackel,
  "An Analog Neural Network Processor and its Application to High-Speed Character Recognition",
  Dans *International Joint Conference on Neural Network,* 1991.

ANNEXES

ANNEXE I :

"2-D Optical Generator of Updating Probabilities for VLSI Implementation of Boltzmann Machines."

P. Lalanne, J.C. Rodier, H. Richard, P. Chavel,
E. Belhaire, K. Madani, et P. Garda

Publication parue dans
International Journal of Optical Computing, vol. 1, pp. 25-30, 1990.

ANNEXE II :

"Two Analog Counters for Neural Networks Implementation"

K. Madani, P. Garda, E. Belhaire, et F. Devos

Publication parue dans

IEEE Journal of Solid State Circuits, Vol. 26, N°7, Juillet 1991.

TABLE DES MATIERES

Introduction :

Réseaux de Neurones Formels et Traitement de l'information Analogique

Circuits Numériques et Circuits Analogiques en 1991. 3

Les Architectures Neuronales 5

Chap I :

Présentation du Problème

I.1- Réseaux de Neurones Formels 9

I.1.1- Qu'est qu'un Réseau de Neurones Formels ? 9

a- Historique 9

b- Description d'un modèle simplifié de RNF 11

I.1.2- Intérêt des Réseaux de Neurones Formels ? 12

I.2- La Machine de Boltzmann 13

I.2.1- Introduction 13

I.2.2- Caractéristiques 14

I.2.3- Performances comparées de Reconnaissances des formes 16

I.2.4- Relaxation stochastique 18

I.2.5- Apprentissage dans la Machine de Boltzmann 21

a- Introduction 21

b- Apprentissage dans le modèle Asynchrone 21

c- Apprentissage dans le modèle Synchrone 23

I.2.6- Température T du réseau 23

I.2.7- Exemples d'Utilisation : Reconnaissance à partir de contour 24

I.3- Conclusions 25

Chapitre II :

Réalisations Analogiques de Réseaux de Neurones : Etat de l'Art

II.1- Réalisations de Réseaux de Neurones Formels 27

II.2- Machine de Boltzmann 28

II.2.1- Laboratoires Bellcore (USA) 28

II.2.2- Autres Réalisations de Machine de Boltzmann 29

II.3- Autres Réalisations de Réseaux de Neurones 30

II.4- Etat de l'art des sommateurs 31
II.5- Conclusion 32

Chapitre III :
Architecture

III.1- Introduction 33
III.2- Description Fonctionnelle de la Cellule Neurone 34
III.2.1- Premiers choix architecturaux 34
III.2.2- Génération de nombres aléatoires 35
a- Introduction 35
b- Loi uniforme à partir de nombres aléatoires binaires 37
c- Génération des nombres binaires : Le speckle Optique 38
d- Générateur Pseudo-Aléatoire à Automate Cellulaire. 40
e- Génération de Nombre aléatoires à répartition sigmoïdale 42
III.2.3- Cellule Neurone A 43
III.2.4- Cellule Neurone B 44
III.2.5- Cellule Synapse 45
III.3- Un ensemble de circuits pour la Machine de Boltzmann 47
III.3.1- Introduction 47
III.3.2- Machine avec apprentissage 47
a- Présentation 47
b- Exemple d'utilisation 50
c- Circuits avec apprentissage 51
d- Réalisation de NETtalk 54
III.3.3- Machine sans apprentissage 55
III.3.4- Système Complet sans apprentissage 56
III.4- Conclusion 57

Chapitre IV :
Conception et Simulation

IV.1- Introduction 59
IV.2- Convertisseur tension-courant : convvi 60
IV.2.1- Introduction 60
IV.2.2- Description du fonctionnement 61
IV.2.3- Technique de “linéarisation” 62
IV.2.4- Simulation 67
IV.2.5- Conclusion 69
IV.3- Conversion courant-tension 70

IV.3.1- Introduction 70
IV.3.2- Convertisseur simple 72
a- Introduction 72
b- Convertisseur à deux transistors 72
c- Convertisseur classe AB 73
IV.3.3- Convertisseur et Variation de Température 75
IV.4- Conception de la Sigmoïde et du comparateur 76
IV.4.1- Introduction 76
IV.4.2- Sigmoïde 76
IV.4.3- Transistor Bipolaire Latéral 78
IV.4.4- Comparateur 79
IV.4.5- Conception du Neurone A 81
IV.5- Les éléments pour l'Apprentissage de la Synapse 82
IV.5.1- Introduction 82
IV.5.2- Compteur Analogique 82
a- Définition 82
b- Principe des compteurs 1 et 2 83
c- Principe du compteur 3 83
IV.5.3- Comparateur 85
IV.6- Générateur aléatoire optoélectronique 87
IV.7- mba1 : Circuit avec 1024 synapses Sans Apprentissage 89
IV.7.1- Introduction 89
IV.7.2- Organisation Interne 90
a- La Synapse 90
b- Cellule de Polarisation 92
c- La Matrice Synaptique 93
d- Stockage et Rafraîchissement des Poids 94
IV.7.3- Interface Utilisateur 96
IV.7.4- Figures 97
IV.8- mba2 : circuit avec 32 Neurones 101
IV.8.1- Introduction 101
IV.8.2- Organisation Interne 102
a- Organisation générale 102
b- Bloc Neurone. 102
c- Convertisseur courant-tension 103
d- Générateur aléatoire 105
IV.8.3- Interface Utilisateur 106
IV.8.4- Figures 106

IV.9- mba1, mba2 : La base d'un module de traitement 110
IV.10- Conclusion 110

Chapitre V :
Test des Cellules de Base et des Circuits
V.1- Introduction 113
V.2- Test des circuits MBAT4 et MBAT5 113
V.2.1- Organisation 113
V.2.2- Test statique du circuit MBAT4. 116
V.2.3- Test statique du circuit MBAT5 117
a- Présentation 117
b- dispersion des transconductances à l'origine 118
V.2.4- Etude des dispersions internes du circuit MBAT5 118
V.2.5- Test dynamique 120
a- Circuit MBAT4 120
b- Circuit MBAT5 121
V.2.6- Comparaison Théorie-Simulations-Mesures 122
V.3- Test du circuit MBAT8 123
V.4- Test du XOR 123
V.5- Conclusion 126

Conclusion 127

Références Bibliographiques 131

Bibliographie 139
Livres sur les Réalisation de Circuit Intégrés Analogiques CMOS 139
Livres sur les Réseaux d'automates et de Neurones Formels 139
Articles interessants et non cités dans le manuscrit 140

Annexes
Annexe I : 142
Annexe II : 149