



HAL
open science

Développement d'un réseau de neurones impulsionnels sur silicium à synapses memristives

Gwendal Lecerf

► **To cite this version:**

Gwendal Lecerf. Développement d'un réseau de neurones impulsionnels sur silicium à synapses memristives. Electronique. Université de Bordeaux, 2014. Français. NNT : 2014BORD0219 . tel-01137492

HAL Id: tel-01137492

<https://theses.hal.science/tel-01137492>

Submitted on 30 Mar 2015

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.

Thèse de Doctorat de l'université de Bordeaux

École doctorale Sciences Physiques et de l'Ingénieur

Spécialité Électronique

Préparée au laboratoire de l'Intégration du Matériau au Système

Par Gwendal Lecerf

Développement d'un Réseau de Neurones Impulsionnels sur Silicium à Synapses Memristives

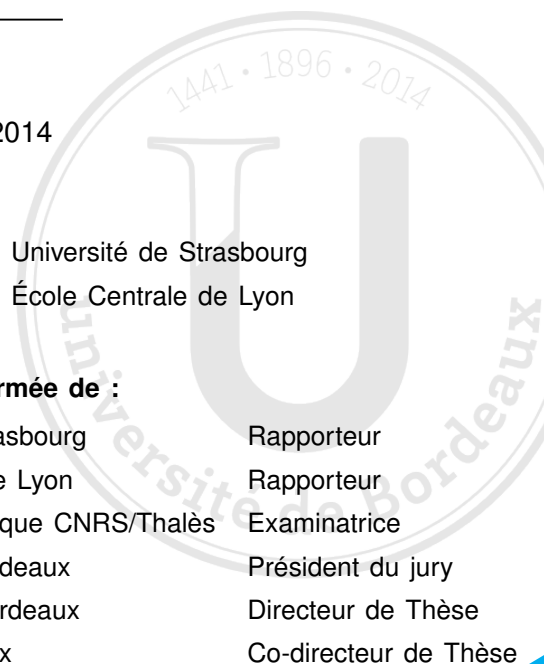
Soutenue le 29 septembre 2014

Après avis de :

M. Luc HÉBRARD	Professeur des Universités - Université de Strasbourg
M. Ian O'CONNOR	Professeur des Universités - École Centrale de Lyon

Devant la commission d'examen formée de :

M. Luc HÉBRARD	Professeur des Universités - Université de Strasbourg	Rapporteur
M. Ian O'CONNOR	Professeur des Universités - École Centrale de Lyon	Rapporteur
M ^{me} Julie GROLLIER	Chargée de Recherche - Unité Mixte de Physique CNRS/Thalès	Examinatrice
M. Claude PELLET	Professeur des Universités - Université de Bordeaux	Président du jury
M. Sylvain SAÏGHI	Maître de Conférence HDR - Université de Bordeaux	Directeur de Thèse
M. Jean TOMAS	Maître de Conférence - Université de Bordeaux	Co-directeur de Thèse



*À mes parents,
À ma sœur.*

Thèse réalisée
dans le Laboratoire de l'**Intégration du Matériaux au Système** de Bordeaux
au sein de l'équipe *Architecture of Silicon Neural Networks*
du groupe **Bioélectronique**
et financée par le projet *Memristive Hardware Analog Neural Network*
soutenue par l'**Agence Nationale de la Recherche**.

Université de Bordeaux
Laboratoire IMS
UMR 5218 CNRS - ENSEIRB-MATMECA
351 cours de la Libération
Bâtiment A31
F - 33405 TALENCE Cedex

REMERCIEMENTS



JE remercie Claude Pellet de m'avoir accueilli au sein du laboratoire IMS Bordeaux et également d'avoir présidé ma soutenance de thèse.

Je remercie Sylvain Saïghi de m'avoir accueilli dans son équipe et de m'avoir fait confiance pendant ces trois années de doctorat. Ses conseils et son soutien m'ont permis de m'épanouir scientifiquement. Je suis très heureux d'avoir pu aider l'équipe à travailler dans une nouvelle direction qui est prometteuse pour l'avenir.

Je remercie tout autant Jean Tomas pour ses idées, ses remarques toujours constructives, son soutien et son écoute (et les échanges de codes \LaTeX aussi).

Je remercie Luc Hébrard et Ian O'Connor pour le temps qu'ils m'ont consacré en tant que rapporteurs de cette thèse.

Je remercie Julie Grollier pour la confiance qu'elle m'a accordée durant le projet, pour sa disponibilité malgré un emploi du temps chargé et pour sa participation à mon jury de thèse.

Je remercie aussi les autres collaborateurs du projet Olivier Temam, Téodora Petrisor, Simon Fossier sans oublier Sören Boyn pour ses conseils et le temps qu'il m'a accordé.

Mes remerciement vont également à mes collègues ingénieurs Ashwin Mangalore et Arnaud Languet pour leurs aides précieuses dans la réalisation de cartes tests et pour leur bonne humeur.

Je remercie également les stagiaires Yoann Robin et Jérémy Fratani qui sont venus travailler à mes côtés. Je remercie plus particulièrement Jérémy pour son travail et son autonomie qui m'ont permis d'achever sereinement ce manuscrit.

Je remercie l'ensemble des personnes de l'IMS, le personnel de soutien à la recherche bien souvent oublié mais indispensable au bon fonctionnement du laboratoire.

Je remercie l'ensemble des collègues du groupe bioélectronique (les équipes AS2N, Élibio et Bio-EM) Sylvie, Noëlle, Yannick, Gilles, Timothée, Bernard, Isabelle, Yann et tous les autres. Je remercie aussi les post-doctorants et stagiaires des différentes équipes.

Je remercie plus particulièrement mes collègues doctorants, et désormais docteurs pour certains, Adam, Filippo, Matthieu, François sans oublier mon voisin de bureau Florian. La bonne ambiance au quotidien, les discussions à l'heure du café (ou du thé pour certains), et l'entraide ont été propices au bon déroulement de chacune de nos thèses. Bon courage aux prochains.

Je n'oublie pas les doctorants et docteurs de l'association AquiDoc avec qui j'ai pris plaisir à organiser les forums de rencontres entre jeunes chercheurs et entreprises.

Je remercie tendrement Aurélie, avec qui je partage désormais ma vie, de m'avoir soutenue dans la dernière année de cette thèse. Sa relecture assidue de mes chapitres, ses bons petits plats et son soutien ont été précieux.

Je remercie également l'ensemble de ma famille, ma sœur, mes parents et grands-parents pour leur compréhension, pour l'affection qu'ils me portent et leurs soutiens infailibles.

RÉSUMÉ

DURANT ces trois années de doctorat, financées par le projet ANR MHANN (Memristive Hardware Analog Neural Network), nous nous sommes intéressés au développement d'une nouvelle architecture de calculateur à l'aide de réseaux de neurones. Les réseaux de neurones artificiels sont particulièrement bien adaptés à la reconnaissance d'images et peuvent être utilisés en complément des processeurs séquentiels.

En 2008, une nouvelle technologie de composant a vu le jour : le memristor. Classé comme étant le quatrième élément passif, il est possible de modifier sa résistance en fonction de la densité de courant qui le traverse et de garder en mémoire ces changements. Grâce à leurs propriétés, les composants memristifs sont des candidats idéaux pour jouer le rôle des synapses au sein des réseaux de neurones artificiels.

En effectuant des mesures sur la technologie des memristors ferroélectriques de l'UMφ CNRS/Thalès de l'équipe de Julie Grollier, nous avons pu démontrer qu'il était possible d'obtenir un apprentissage de type STDP (Spike Timing Dependant Plasticity) classiquement utilisé avec les réseaux de neurones impulsionnels. Cette forme d'apprentissage, inspirée de la biologie, impose une variation des poids synaptiques en fonction des évènements neuronaux.

En s'appuyant sur les mesures réalisées sur ces memristors et sur des simulations provenant d'un programme élaboré avec nos partenaires de l'INRIA Saclay, nous avons conçu successivement deux puces en silicium pour deux technologies de memristors ferroélectriques. La première technologie (BTO), moins performante, a été mise de côté au profit d'une seconde technologie (BFO). La seconde puce a été élaborée avec les retours d'expérience de la première puce. Elle contient deux couches d'un réseau de neurones impulsionnels dédié l'apprentissage d'images de 81 pixels. En la connectant à un boîtier contenant un crossbar de memristors, nous pourrions réaliser un démonstrateur d'un réseau de neurones hybride réalisé avec des synapses memristives ferroélectriques.

Mots Clés

Conception Analogique, Réseau de Neurones Impulsionnels, Memristor, Résistance Memristive, Apprentissage, STDP.

ABSTRACT

SUPPORTED financially by ANR MHANN project, this work proposes an architecture of spiking neural network in order to recognize pictures, where traditional processing units are inefficient regarding this.

In 2008, a new passive electrical component had been discovered : the memristor. Its resistance can be adjusted by applying a potential between its terminals. Behaving intrinsically as artificial synapses, memristive devices can be used inside artificial neural networks.

We measure the variation in resistance of a ferroelectric memristor (obtained from UMR CNRS/Thalès) similar to the biological law STDP (Spike Timing Dependant Plasticity) used with spiking neurons.

With our measurements on the memristor and our network simulation (aided by INRIA-Saclay) we designed successively two versions of the IC. The second IC design is driven by specifications of the first IC with additional functionalities. The second IC contains two layers of a spiking neural network dedicated to learn a picture of 81 pixels. A demonstrator of hybrid neural networks will be achieved by integrating a chip of memristive crossbar interfaced with the second IC.

Keywords

Analog Circuit Conception, Spiking Neural Network, Memristor, Memristive Resistor, Learning, STDP.

Abréviations

MHANN *Memristive Hardware Analog Neural Network*

LTP *Long Term Potentiation*

LTD *Long Term Depression*

STDP *Spike-Timing-Depend Plasticity*

PA potentiel d'action

IF *Integrate & Fire*

HH Hodgkin-Huxley

LIF *Leaky Integrate & Fire*

AdEx *Adaptive Exponential Integrate & Fire*

MN Mihalas-Niebur

BFO à barrière ferroélectrique BiFeO₃

BTO à barrière ferroélectrique BaTiO₃

RMS *Recognition, Mining and Synthesis*

WTA *Winner-Take-All*

MNIST *Mixed National Institute of Standards and Technology*

VHDL *VHSIC (Very High Speed Integrated Circuit) Hardware Description Language*

FPGA *Field Programmable Gate Array*

SpANNWiTA *Spiking Analog Neural Network Winner-Take-All*

CCII convoyeur de courant de seconde génération

TABLE DES MATIÈRES



TABLE DES MATIÈRES	xi
Remerciements	iv
Résumé	vi
Abstract	viii
Abréviations	ix
Table des matières	xi
Introduction	xv
1 De la neurobiologie aux systèmes neuromorphiques et bioinspirés	1
1.1 La neurobiologie	2
1.1.1 La biologie de la cellule	3
1.1.2 La plasticité biologique	7
1.1.3 L'homéostasie	8
1.2 Les systèmes neuromorphiques et bioinspirés	8
1.2.1 Les neurones formels	9
1.2.2 Les neurones impulsionnels	9
1.2.3 La synapse et l'apprentissage	14
1.3 Un projet bioinspiré : MHANN	18
1.4 Conclusion	19
2 Les composants memristifs	20
2.1 L'apprentissage par loi temporelle	23
2.2 L'état de l'art des mesures de STDP avec des composants memristifs	25
2.2.1 Le changement de phase	25
2.2.2 L'oxydoréduction	26
2.2.3 L'organique	29
2.2.4 Les effets purement électroniques	29
2.3 Les mesures réalisées avec des memristors BFO	31
2.3.1 Le comportement memristif	32
2.3.2 Le comportement biologique	33
2.3.3 Les mesures avec des potentiels d'action	37
2.4 Conclusion	43
3 Les réseaux de neurones impulsionnels : de la modélisation à la simulation	44
3.1 La modélisation des réseaux de neurones	46
3.1.1 Des exemples d'architectures de réseaux dédiées à la classification	46
3.1.2 Les réseaux de neurones hybrides dans la littérature	48
3.1.3 Le réseau de neurones du projet MHANN	50

3.2	La modélisation de la plasticité synaptique	52
3.2.1	Un exemple de modélisation d'une loi de variation de conductance de memristor à oxydoréduction	52
3.2.2	La modélisation issue de mesures des lois de variation de conductance des memristors BFO	53
3.3	Les simulations du réseau du projet MHANN	56
3.3.1	Le simulateur événementiel	56
3.3.2	Les simulations avec la modélisation de la STDP du memristor BFO	58
3.4	Conclusion	61
4	La conception sur silicium	62
4.1	La connexion entre une synapse memristive et un neurone impulsionnel	63
4.2	Le détail des blocs et les mesures de la première puce	66
4.2.1	Le convoyeur de courant de seconde génération	67
4.2.2	Le neurone <i>Leaky Integrate & Fire</i> (LIF)	71
4.2.3	Les capteurs d'entrée et la commande numérique	78
4.3	Les améliorations et simulations de la seconde puce	80
4.3.1	La partie analogique	80
4.3.2	La partie numérique	87
4.3.3	Le dessin de la seconde puce	88
4.4	Conclusion	89
	Conclusions & Perspectives	92
	Annexe A D'autres mesures réalisées avec le memristor BFO	96
	Annexe B Les paramètres de la modélisation des formes de STDP mesurées avec des memristors BFO	100
	Annexe C D'autres modélisations de STDP et exemples d'apprentissage	102
	Annexe D Des compléments sur le fonctionnement de la seconde version du réseau de neurones sur silicium	107
	Références bibliographiques de l'auteur	112
	Références bibliographiques	114

Sans imagination il ne pourrait y avoir création.

Albert Jacquard

INTRODUCTION



LE cerveau a toujours été un objet de fantasme. Son mécanisme, encore mal connu, est composé de milliards de milliards de connexions et de milliards de cellules : c'est une unité de calculs parallèles pharaonique qui consomme très peu. En 1943, Warren McCulloch et Walter Pitts créèrent le premier modèle mathématique et informatique de neurone biologique (McCulloch et Pitts, 1943). Issu d'observations neurophysiologiques et anatomiques, ce modèle simple de neurone reste encore de nos jours très utilisé dans les réseaux de neurones artificiels. Grâce aux recherches qui ont suivi sur les réseaux de neurones formels, dont l'AdaLinE (*Adaptive Linear Element*) de Widrow et Hoff (1960) est un des premiers exemples, il a été montré que ces architectures bioinspirées pouvaient être très performantes dans les tâches telles que la reconnaissance d'images, la classification ou encore l'approximation de fonction inconnue. Mais ces recherches sont restées très en marge des communautés électronique et informatique car la miniaturisation du transistor a permis d'amplifier les puissances de calcul des processeurs *i.e.* le nombre d'opérations par seconde, ce qui a rendu obsolète les travaux sur les réseaux de neurones formels. Malgré l'évolution de fabrication des *wafer* et la diminution de la taille des grilles des transistors, les fréquences d'horloge n'ont plus augmenté.

En 2005, les industriels de la microélectronique ont décidé d'arrêter la course à la montée en fréquence de leurs processeurs, notamment dans un souci de maîtrise de la consommation (Lowney, 2006). En effet, pour continuer à accroître la puissance des processeurs, et ainsi rester dans les projections de la loi de Moore, les industriels ont pris la direction de la multiplication des cœurs de calcul, et c'est ce cap qui est maintenu à l'heure actuelle. Pour tirer le meilleur parti de ces nouvelles architectures, les développeurs doivent désormais programmer de façon différente : l'architecture d'un processeur n'est plus seule gage de rapidité puisque le parallélisme du logiciel devient un facteur déterminant. Cette nouvelle façon de penser ouvre de nouvelles perspectives. L'architecture von Neumann n'est plus intouchable. S'il est possible de fractionner un programme pour qu'il soit exécuté par différents cœurs de calcul alors pourquoi ne pas utiliser des calculateurs dédiés ? Les CPU (de l'anglais *Central Processing Unit*) ont été perfectionnés pour le calcul séquentiel : ils utilisent des *pipelines* pour optimiser leurs temps de calculs. Pour faire des calculs en parallèle, ce sont les processeurs graphiques (GPU de l'anglais *Graphic Processing Unit*) qui prennent l'avantage : alors qu'ils étaient dédiés à l'affichage, ils sont désormais une ressource de calcul à part entière. Un nouveau cœur de calcul a fait son apparition dans les catalogues des fondeurs nommé NPU (de l'anglais *Neural Processing Unit*) : composé de neurones, ce processeur a pour ambition d'apprendre des tâches simples sans avoir la nécessité de créer un programme (Kumar, 2013).

Par ailleurs, la généralisation de l'utilisation d'internet et ses nouveaux modes d'accès (smartphones, tablettes), engendrent des flots de données immenses à traiter chaque jour. Pour analyser au mieux ces flux de données en constante augmentation, Intel (Dubey, 2005) a décidé de développer les domaines de reconnaissance, d'exploration et de synthèse ou domaines RMS (de l'anglais *Recognition, Mining and Synthesis*). Ces trois domaines sont l'apanage des réseaux de neurones. Leurs algorithmes sont plus efficaces tout en consommant moins d'énergie (Temam,

2010), c'est pourquoi les industriels ont un intérêt croissant pour les réseaux de neurones artificiels.

En 2008, [Strukov et al. \(2008\)](#) répondent à une théorie énoncée 37 ans plus tôt par [Chua \(1971\)](#), celle du memristor. Dans les années qui ont suivi la parution du papier du théoricien, les physiciens ont voulu trouver une solution matérielle répondant à l'équation $M = d\phi/dq$, mais sans succès. Aussi connu pour être le quatrième élément passif, le memristor a pour principale propriété de modifier la valeur de sa résistance en fonction du courant qui le traverse et de mémoriser cet état une fois que le courant ne le parcourt plus. L'utilisation d'un tel composant dans le monde numérique d'aujourd'hui pourrait réduire la consommation de nos mémoires, et même unifier les différents types de mémoires. Les grands groupes de la microélectronique l'ont bien compris, et cherchent la combinaison de matériaux idéale pour le créer. La propriété plastique de cet élément passif a aussi un autre intérêt : l'utilisation dans des systèmes neuromorphiques ([Chua et Kang, 1976](#)).

Le projet ANR *Memristive Hardware Analog Neural Network (MHANN)*, qui a financé cette thèse, s'inscrit à l'intersection de ces deux événements technologiques : d'une part, la mise au jour d'un nouveau composant passif nommé memristor qui a des propriétés intéressantes pour les électroniciens ; et d'autre part, l'arrêt en montée des fréquences de processeurs depuis 2005 qui a ouvert la porte à de nouvelles architectures de calculs dont les réseaux de neurones.

Beaucoup de théories sur les systèmes neuromorphiques à base de memristors ont été développées ces dernières années ([Zamarreño Ramos et al., 2011](#) ; [Shin et al., 2012](#)), mais peu de chercheurs ont pu faire des expérimentations avec de réels composants memristifs. Grâce à nos partenaires du projet MHANN, et en particulier l'équipe du Dr Julie Grollier de l'UMr CNRS/Thalès, nous avons pu réaliser nos propres mesures sur des dispositifs memristifs ferroélectriques et ainsi créer un système permettant d'utiliser au mieux les nouvelles propriétés de ce composant du futur.

Dans ce manuscrit, nous proposons le développement d'un réseau de neurones sur silicium avec des synapses memristives réelles. Dans un premier chapitre, nous allons nous familiariser avec les notions biologiques liées aux réseaux de neurones. Il nous sera alors possible de présenter le projet [MHANN](#) sur lequel s'appuie cette thèse. Dans un deuxième chapitre, nous détaillerons les propriétés des composants memristifs et leurs avantages à être utilisés dans les réseaux de neurones. Puis nous présenterons des mesures réalisées à l'aide de dispositifs memristifs ferroélectriques qui serviront de synapses dans notre réseau. Dans un troisième chapitre, nous aborderons les réseaux de neurones et leur apprentissage. Nous présenterons des simulations qui s'appuient sur les mesures faites sur les memristors ferroélectriques. Nous pourrons alors décrire, dans un quatrième chapitre, les deux puces sur silicium réalisées durant ces trois années de thèse afin de construire une nouvelle architecture de réseau de neurones hybrides (transistors silicium avec memristors ferroélectriques). Enfin nous achèverons ce manuscrit de thèse par

une rétrospection des trois années du projet [MHANN](#) et une projection du travail qui reste à accomplir dans l'année à venir. Nous donnerons également des pistes pour continuer à améliorer les réseaux de neurones à synapses memristives.

CHAPITRE 1

DE LA NEUROBIOLOGIE AUX SYSTÈMES NEUROMORPHIQUES ET BIOINSPIRÉS

LA neurobiologie est l'étude du système nerveux. C'est à l'intérieur du système nerveux et par extension du cerveau que se trouvent les cellules qui nous intéressent. La puissance de calcul, évaluée entre 10^{13} et 10^{19} instructions par seconde, et la consommation de quelques dizaines de watts du cerveau en font un sujet d'étude des plus intéressants. Mais la complexité cellulaire entrave la bonne connaissance de son fonctionnement. C'est pourquoi, en s'appuyant sur des connaissances biologiques toujours plus importantes, l'ingénierie neuromorphique essaie d'imiter le comportement du cerveau.

Les propriétés des systèmes neuromorphiques ont apporté de nouvelles façons de calculer et de penser le calcul. En effet, en essayant de copier le cerveau, les ingénieurs en neuromorphique ont créé un nouveau paradigme de calcul : les opérations sont faites en parallèle et non plus en série. De plus, il n'est plus question d'exécution d'instructions mais de modification des liaisons entre les unités de calculs. En essayant d'imiter les mécanismes du vivant, certains systèmes neuromorphiques se sont éloignés du fonctionnement biologique : il ne s'agit plus d'imiter le cerveau mais de s'en inspirer.

L'ingénierie neuromorphique se découpe désormais en deux grands axes : la recherche de systèmes visant à imiter le vivant afin de mieux comprendre son fonctionnement, le neuromimétique ; la recherche s'inspirant de certains mécanismes observés sur le vivant pour créer des systèmes plus performants, le bioinspiré.

Dans ce chapitre nous allons esquisser le fonctionnement des cellules biologiques relatives aux réseaux de neurones pour nous permettre de connaître l'origine des mécanismes utilisés dans les réseaux de neurones artificiels ou impulsionnels. Puis nous étudierons les différentes équations mathématiques et structures matérielles cherchant à copier ou imiter les observations biologiques afin d'avoir une idée des différentes approches possibles. Nous ferons un état de l'art des implémentations sur silicium existantes. Et enfin, nous introduirons le projet bioinspiré *Memristive Hardware Analog Neural Network* (MHANN) sur lequel cette thèse repose.

1.1 La neurobiologie

Le système nerveux, composé de neurones, est réparti dans tout le corps. Les neurones sont à la fois des vecteurs d'informations et des actionneurs. Les réactions chimiques à l'intérieur du corps neuronal entraînent des variations de potentiels électriques qui codent l'information. Les neurones communiquent entre eux, ou avec les organes, par l'intermédiaire de synapses qui sont à l'interface avec le monde extracellulaire. Nous allons donner une description biologique succincte des neurones et de leurs synapses, ainsi que certains mécanismes de leurs fonctionnements.

1.1.1 La biologie de la cellule

1.1.1.1 Le neurone

Le neurone illustré dans la figure 1.1 est une unité de prise de décision. En tant que cellule biologique, il possède une frontière avec l'extérieur : sa membrane. Cette dernière est parsemée de canaux ioniques dépendants de la tension. Leurs ouvertures permettent d'accueillir des ions de l'extérieur de la cellule ou au contraire de les évacuer vers l'extérieur. Le potentiel électrique de la membrane dépend donc de la différence de concentration des ions à l'intérieur et à l'extérieur de la cellule neuronale.

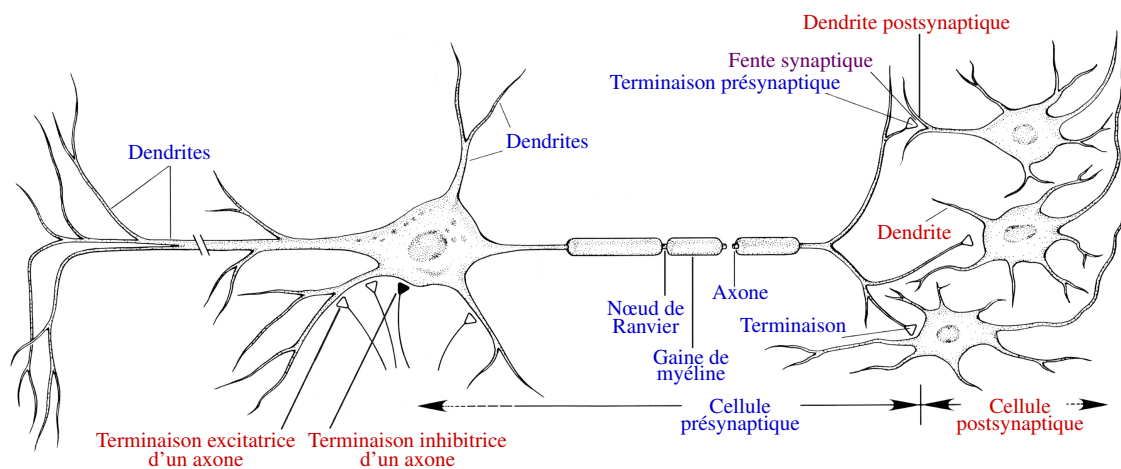


FIGURE 1.1 – Schéma d'un neurone avec les termes biologiques associés, tiré de [Kandel et al. \(2000\)](#).

La stimulation d'un neurone par ses congénères via ses dendrites va influencer les échanges d'ions et ainsi modifier le potentiel de membrane. La réponse d'un neurone à une stimulation est un potentiel d'action (figure 1.2) qui parcourt l'axone vers les terminaisons synaptiques. Le potentiel de repos de la membrane d'un neurone se situe à environ -70 mV. Une stimulation venant d'une synapse excitatrice engendre une dépolarisation la membrane : son potentiel augmente. S'il dépasse le seuil des -50 mV, la dépolarisation devient plus importante et brusque, elle peut atteindre les 30 mV. Le neurone entame ensuite une phase de repolarisation puis d'hyperpolarisation où le potentiel devient inférieur au potentiel de repos. La valeur typique de l'hyperpolarisation est de -80 mV. Enfin, le potentiel de membrane retrouve sa valeur de repos. Ces trois phases décrivent un potentiel d'action et s'échelonnent sur une durée d'environ 1 ms. *A contrario* lorsque le neurone est stimulé via une synapse inhibitrice son potentiel de membrane s'hyperpolarise, il n'a donc aucune chance de générer un potentiel d'action.

Pendant les phases de dépolarisation et de repolarisation, le neurone est dans une période réfractaire absolue *i.e.* le neurone est amorphe, aucune stimulation engendrera un nouveau potentiel d'action. En revanche, durant la phase d'hyperpolarisation, le neurone est dit en période réfractaire relative car s'il est stimulé suffisamment pendant cette période, il peut générer un nouveau potentiel d'action.

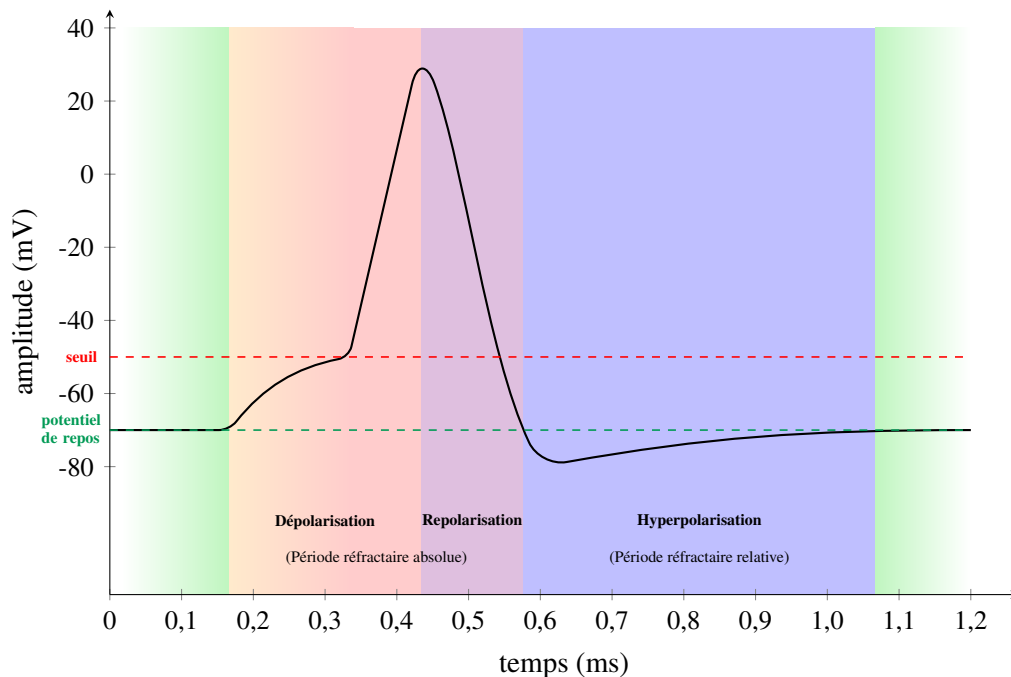


FIGURE 1.2 – Schéma d'un potentiel d'action biologique annoté de ses différentes transitions.

L'axone du neurone peut varier de quelques millimètres à plus d'un mètre. C'est pourquoi il est composé de gaines de myéline entrecoupées de nœuds de Ranvier. Cet ensemble permet au signal électrique de parcourir plus rapidement le neurone. La gaine de myéline isole la membrane du neurone de l'extérieur mais affaiblit l'amplitude du potentiel d'action¹ alors que le nœud de Ranvier concentre des canaux de sodium et potassium dépendants des différences de potentiels et régénère le potentiel d'action. Le potentiel d'action se propage de nœud en nœud, jusqu'à atteindre les terminaisons synaptiques. La transmission du message se fait alors chimiquement ou électriquement par la synapse dont nous détaillerons le fonctionnement dans la partie suivante.

Les chercheurs en ingénierie neuromorphique se sont concentrés sur les comportements électriques des neurones. Ils ont alors établi vingt familles de neurones, visibles à la figure 1.3, suivant les réponses électriques de chacun d'entre eux. Les différentes réponses électriques aux stimulations en courant montrent la diversité des neurones. Par exemple, en (A), le neurone adapte sa fréquence de décharge au fur et à mesure du temps alors que le courant reste constant. Dans le cas (B), le neurone ne s'active qu'une seule fois. En (D), le neurone entame une phase où les potentiels d'action s'enchaînent rapidement, appelée *burst*, avant que son potentiel de membrane ne revienne au repos. Le neurone en (L) intègre les pulses de courant qui lui sont appliqués. Lorsque le seuil est dépassé un potentiel d'action est généré.

Nous venons de voir que le neurone crée une information électrique qui se propage à travers l'ensemble de la cellule. La transmission du signal électrique entre les cellules neuronales se fait par l'intermédiaire des synapses.

¹Comportement comparable à un circuit RC.

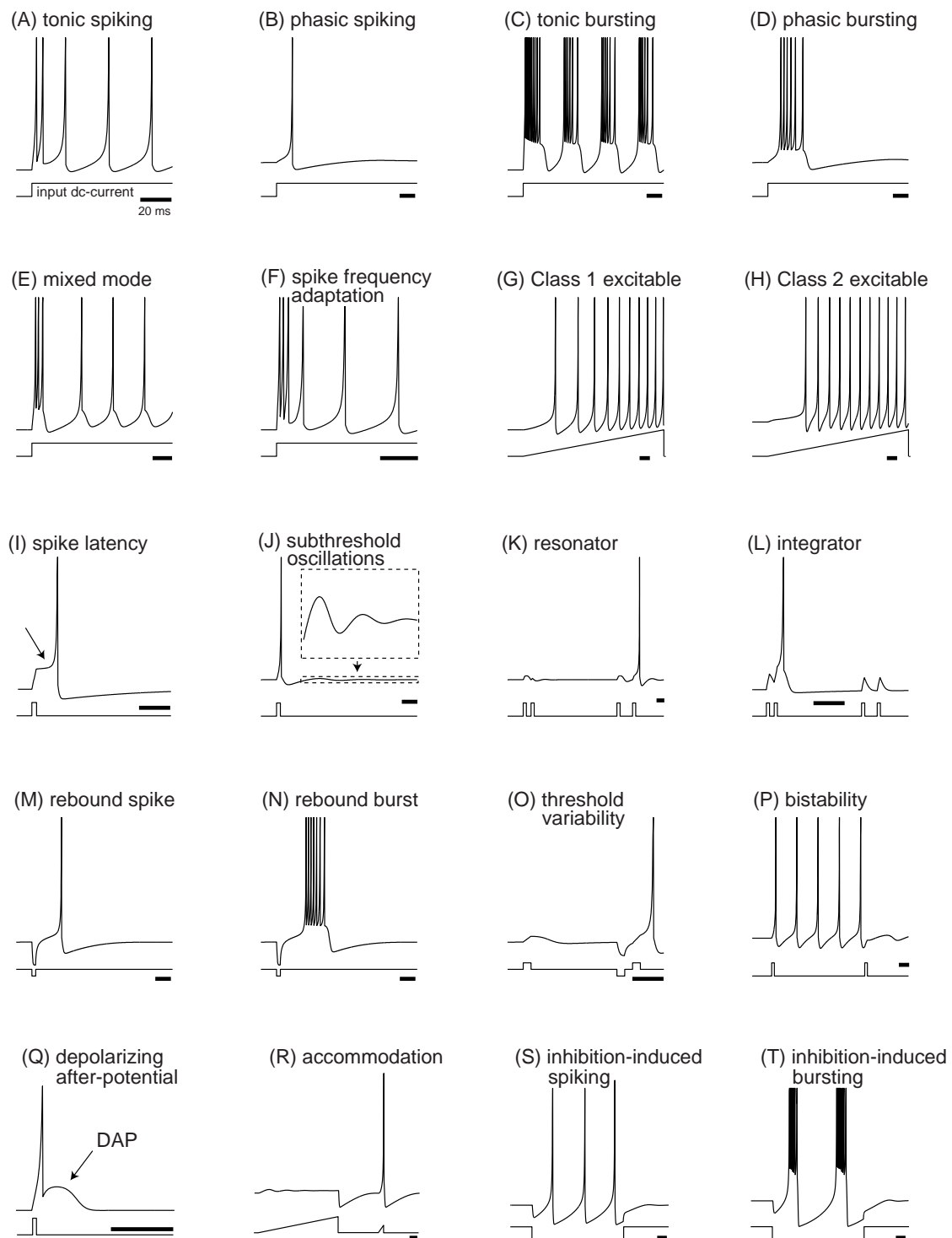


FIGURE 1.3 – Illustration des différentes familles de neurones biologiques. Les illustrations (A) à (T) donnent les réponses électriques du neurone (haut) à un courant de stimulation (bas). Illustration librement accessible en version électronique sur www.izhikevich.com.

1.1.1.2 La synapse

La synapse, illustrée figure 1.4, se situe à l'interface de deux neurones ou d'un neurone et d'un organe. Il existe deux types de synapses : la synapse électrique et la synapse chimique. Cette dernière, bien que moins rapide, est la plus répandue chez les mammifères. En effet, alors que la synapse électrique transmet quasi-directement le potentiel d'action d'un neurone à l'autre, la synapse chimique passe par le biais de neurotransmetteurs. Cette transmission peut être découpée en quatre grandes étapes.

- Les neurotransmetteurs, enveloppés à l'intérieur d'une membrane, forment une vésicule synaptique qui est transportée via des microtubes vers les terminaisons synaptiques qui sont à l'extrémité de l'axone.
- En réponse à un potentiel d'action, des neurotransmetteurs sont libérés au travers de la membrane présynaptique par un processus d'exocytose : la membrane dans laquelle sont les neurotransmetteurs fusionne avec la membrane de la cellule neuronale. Les neurotransmetteurs peuvent alors être expulsés vers l'extérieur, dans le liquide extracellulaire.
- Les neurotransmetteurs franchissent la fente synaptique et se lient à un récepteur situé sur la membrane postsynaptique d'un neurone afférent. L'activation des récepteurs chimiques peut dépolariser ou hyperpolariser la membrane qui subit alors une influence excitatrice² ou inhibitrice³.
- Après avoir produit leur effet, les neurotransmetteurs sont désactivés. Ils disparaissent de la fente synaptique par diffusion ou par dégradation via des enzymes ou encore par captation de l'élément présynaptique.

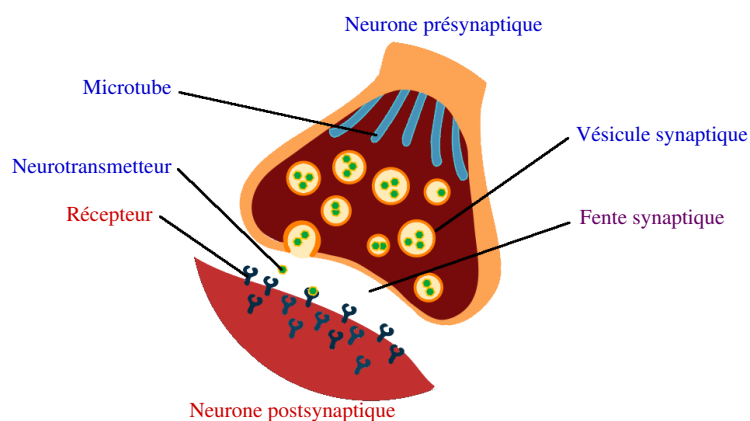


FIGURE 1.4 – Schéma d'une synapse avec les termes biologiques associés.

²Il y a une ouverture de différents canaux ioniques qui permet de faire passer les ions sodium de la fente synaptique vers la dendrite postsynaptique ou les ions potassium en sens inverse.

³Les canaux ioniques ouverts laissent passer les ions chlorure de la fente synaptique vers la dendrite postsynaptique.

Les neurotransmetteurs ne peuvent avoir qu'une seule influence : l'excitation OU l'inhibition de la cellule cible. Les synapses excitatrices se trouvent préférentiellement sur les épines dendritiques alors que les synapses inhibitrices se situent sur le corps cellulaire, comme illustré sur la figure 1.1.

La modulation des ions et des neurotransmetteurs lors de l'apparition du potentiel d'action change l'efficacité de transmission qui a lieu à l'interface de deux neurones. Cette plasticité a été observée par de nombreux biologistes au cours de ces vingt dernières années.

1.1.2 La plasticité biologique

L'expérience modifie l'efficacité de la transmission du signal. La connexion synaptique est donc plastique. Hebb (1949) énonça la théorie suivante sur le fonctionnement de la plasticité des synapses :

When an axon of cell A is near enough to excite B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A's efficiency, as one of the cells firing B, is increased.

Il indique qu'il y a une corrélation entre le renforcement synaptique et l'activité des neurones. La proposition de Hebb a été vérifiée quelques années plus tard (Kandel, 1976). Les synapses qui adoptent ce genre de comportement sont communément appelées synapses *hebbiennes*. Avec ce principe, il est considéré que l'information à enregistrer est conservée au niveau de la synapse.

Les lois de variation de la plasticité ont été ordonnées suivant quatre grandes catégories : l'habituation, la sensibilisation, la potentialisation et la dépression. L'habituation est une réponse dégradée dans le temps pour un même stimuli. La sensibilisation quant à elle est une réponse de plus en plus rapide à un stimuli donné. La potentialisation et la dépression sont deux types d'apprentissages associatifs complémentaires. Ces lois de variation dépendent à la fois du stimuli appliqué et de l'apprentissage précédent.

Des formes de plasticité ont été observées sur différentes régions du cerveau (Abbott et Nelson, 2000) comme illustré à la figure 1.5. Ces formes sont l'association de deux catégories de plasticité : la potentialisation à long terme, en anglais *Long Term Potentiation (LTP)* et la dépression à long terme, en anglais *Long Term Depression (LTD)*. L'évolution de la plasticité des synapses dépend de l'activité dans le temps des neurones pré- et postsynaptiques. Elle est désignée par un terme englobant les deux phénomènes de LTP et de LTD : la *Spike-Timing-Depend Plasticity (STDP)*. Nous décrirons dans la suite du chapitre sa modélisation et nous détaillerons son principe de fonctionnement.

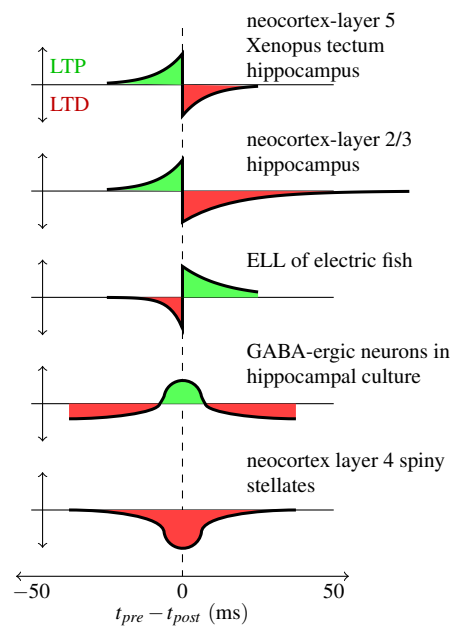


FIGURE 1.5 – Schémas montrant différentes formes de STDP observées en biologie (Abbott et Nelson, 2000).

1.1.3 L'homéostasie

L'homéostasie est la capacité d'un système à conserver un état d'équilibre. Dans les réseaux de neurones, l'homéostasie permet de garder la fréquence de décharge d'un neurone autour d'un niveau moyen (Turrigiano et Nelson, 2000, 2004). Le processus d'homéostasie agit sur les concentrations des ions qui permettent la génération de potentiels d'action (Marder et Goaillard, 2006). Pour son bon fonctionnement, l'homéostasie a besoin d'avoir un niveau de comparaison, des détecteurs de niveau et des régulateurs.

Nous venons de voir les principes biologiques des neurones et des synapses ainsi que le vocabulaire associé. Nous allons maintenant regarder leurs modélisations et leurs implémentations sur silicium.

1.2 Les systèmes neuromorphiques et bioinspirés

Le terme *neuromorphic engineering* (ingénierie neuromorphique) a été utilisé pour la première fois par Mead (1989) et désignait les champs d'application dédiés à l'imitation du système nerveux. Il est désormais aussi employé pour les systèmes inspirés des réseaux de neurones biologiques.

Dans la partie précédente, nous avons vu les différents constituants biologiques permettant la formation d'un réseau de neurones. C'est à partir des observations biologiques que les mathématiciens, électroniciens et informaticiens ont créé des systèmes bioinspirés voire bioréalistes.

Un réseau de neurones artificiels est une simplification de la biologie. Il est structuré par deux entités fondamentales : le neurone et la synapse. Dans les implémentations électroniques et informatiques, il est communément admis d'utiliser une modélisation ponctuelle du neurone alors qu'un neurone biologique évolue dans un espace à 3 dimensions où l'axone, tout comme les dendrites, peuvent avoir une longueur non négligeable. Bien que cela soit une approximation grossière, cette première approche permet tout de même de créer des réseaux de neurones capables d'apprendre.

Dans la suite de ce chapitre nous allons répertorier les formalismes de neurones et de synapses et leurs implémentations matérielles.

1.2.1 Les neurones formels

Dans les années quarante, un modèle mathématique de neurone formel (McCulloch et Pitts, 1943) a été conçu afin de mieux comprendre le fonctionnement global d'un réseau de neurones. Sa construction apporte un nouveau paradigme de calcul, parallèle, qui s'oppose à la machine von Neumann, séquentielle.

En analogie à l'accumulation de charges faite par la membrane du neurone biologique, le neurone formel somme les contributions d'entrées x_i et de biais b (figure 1.6). Puis une fonction d'activation f_a est appliquée au résultat de la somme permettant ainsi de faire une discrimination du résultat. La fonction d'activation utilisée dépend du type de réseau souhaité et de la place du neurone en son sein. Il existe différentes fonctions d'activation linéaires ou non : échelon, tangente hyperbolique, gaussienne, etc. Ce modèle de neurone est encore de nos jours très utilisé en informatique mais reste éloigné de la biologie.

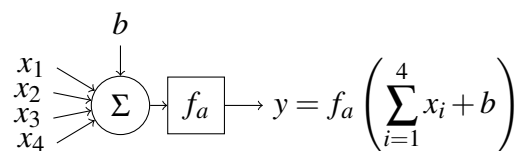


FIGURE 1.6 – Illustration et équation du neurone formel seul.

1.2.2 Les neurones impulsionnels

Les neurones impulsionnels sont un autre type de neurones plus proches de la biologie. Ils transposent l'information en temps. L'instant de l'impulsion aussi bien que la fréquence des impulsions deviennent des vecteurs d'information. Les modèles de neurones impulsionnels peuvent être regroupés dans deux grandes familles : les modèles à conductance et les modèles *Integrate & Fire* (IF).

1.2.2.1 Les modèles à conductance

Les modélisations

En s'appuyant sur des observations biologiques, [Hodgkin et Huxley \(1952\)](#) ont créé un modèle prenant en compte les conductances des canaux ioniques présents à l'interface des neurones. La figure 1.7 présente le circuit électrique équivalent superposé à la membrane et ses canaux ioniques d'une cellule neuronale. Chaque canal ionique est sélectif pour un type spécifique d'ion. Le modèle Hodgkin-Huxley (HH) est fonction de trois canaux : le canal des ions sodium Na^+ , celui des ions potassium K^+ et un canal de fuite L (de l'anglais *leak*, ajout au modèle pour régler la tension de repos et la sensibilité du neurone). La membrane est idéalisée par une capacité électrique. L'équation différentielle résultante est la suivante :

$$C_M \frac{dV_M}{dt} = I - g_{\text{Na}}(V_M - E_{\text{Na}}) - g_{\text{K}}(V_M - E_{\text{K}}) - g_L(V_M - E_L) \quad (1.1)$$

Les conductances des canaux ioniques de Na^+ et de K^+ sont régies par des équations différentielles dépendantes du potentiel de membrane V_M . En décrivant le fonctionnement des canaux ioniques au niveau de la membrane du neurone, le formalisme de HH est proche de la biologie mais il est difficile à résoudre.

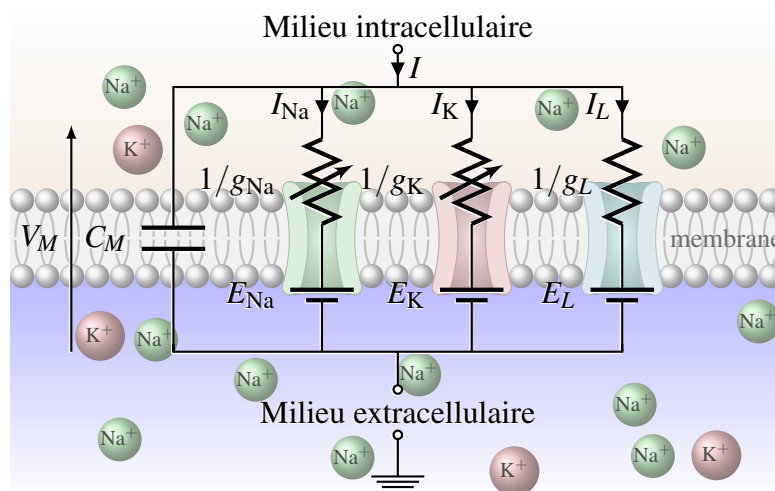


FIGURE 1.7 – Schéma électrique du neurone HH modélisant la membrane d'un neurone et les canaux ioniques à l'interface avec le monde extracellulaire.

Il existe des simplifications du modèle HH. Le modèle de FitzHugh-Nagumo ([FitzHugh, 1955, 1961](#) ; [Nagumo et al., 1962](#)) en réduit le formalisme à deux équations différentielles. À partir du modèle de FitzHugh-Nagumo, [Hindmarsh et Rose \(1984\)](#) proposent un modèle de trois équations différentielles du première ordre couplées.

Les implémentations

L'intégration électronique des équations différentielles non-linéaires et des nombreux paramètres du neurone HH a été réalisée par Renaud *et al.* (2007) et Saïghi *et al.* (2011). Il est difficile mais pas impossible de pouvoir régler correctement le neurone pour obtenir le comportement souhaité (Buhry *et al.*, 2011 ; Grassia *et al.*, 2011). La complexité d'implémentation et de mise en œuvre du neurone HH ne lui permet pas d'être utilisé dans un réseau de neurones à grande échelle.

Les modèles à conductance restent complexes à implémenter. Leurs jeux d'équations différentielles non linéaires avec de nombreux paramètres rendent leurs calibrations particulièrement délicates.

1.2.2.2 Les modèles *Integrate & Fire*

Nous allons décrire quatre modèles de neurones *Integrate & Fire* et leurs implémentations : *Integrate & Fire* (IF) et *Leaky Integrate & Fire* (LIF), *Adaptive Exponential Integrate & Fire* (AdEx) et enfin Mihalas-Niebur (MN).

Les modélisations IF et LIF

Bien avant les modèles à conductance et sur la base d'observations biologiques du système nerveux, Louis Lapicque, en 1907, a modélisé électriquement un premier neurone biologique (Lapicque, 1907). Il a utilisé une capacité de membrane C_M et une résistance de fuite R_L en parallèle (figure 1.8.a). Un tel modèle à lui seul ne peut pas générer de potentiel d'action. Mais Lapicque a postulé que si la tension de membrane V_M dépassait un certain seuil, alors un potentiel d'action serait généré avant de revenir au seuil de repos V_{rest} (figure 1.8.b). Ce modèle, dont l'équation est donnée par (1.2), est un neurone *Leaky Integrate & Fire* (LIF).

En enlevant la résistance de fuite, le modèle est simplifié en *Integrate & Fire* (IF). La décharge du potentiel de membrane, permise par la résistance R_L lorsqu'aucune contribution extérieure n'apparaît, n'existe plus : le potentiel de membrane reste alors constant et somme les contributions sans que leurs temps d'arrivée n'aient une influence. Ces deux modèles de neurones impulsionnels, IF simple et LIF, sont les plus élémentaires. Ils sont décrits par une seule équation différentielle liée au modèle électrique de la capacité et permettent d'appréhender le monde des réseaux de neurones impulsionnels. Avec cette description mathématique, ils ne représentent que la famille neuronale (L) de la figure 1.3.

$$I(t) - \frac{V_M(t) - V_{rest}}{R_L} = C_M \frac{dV_M}{dt} \quad (1.2)$$

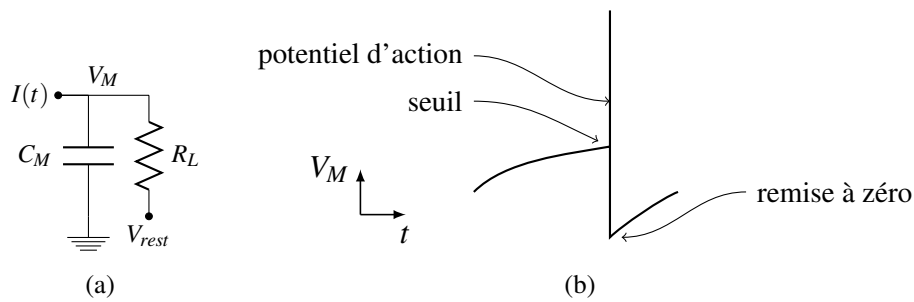


FIGURE 1.8 – Modèle électrique (a) et exemple de tension de membrane (b) du neurone *Integrate & Fire* de L. Lapicque, illustration provenant de Abbott (1999).

Les implémentations IF et LIF

Le premier neurone IF sur silicium a été réalisé par Mahowald et Douglas (1991). En s’aidant des propriétés des transistors relatives aux matériaux, le circuit est peu complexe. De plus, le neurone artificiel possède une constante de temps de l’ordre de la milliseconde équivalente à celle de la biologie. Il est très compact ($\sim 0,1 \text{ mm}^2$) et consomme peu ($\sim 60 \mu\text{W}$). Depuis, de nombreux chercheurs ont continué à améliorer le neurone IF sur silicium et à complexifier son fonctionnement.

La modélisation AdEx

Le modèle *Adaptive Exponential Integrate & Fire* (AdEx) permet d’obtenir plusieurs familles de neurones corticaux (Brette et Gerstner, 2005) grâce à un jeu de deux équations (1.3) qui décrivent le potentiel de membrane V_M et la variable d’adaptation w .

$$\begin{cases} C_M \frac{dV_M}{dt} = -g_L(V_M - E_L) + g_L \Delta_T \exp\left(\frac{V_M - V_T}{\Delta_T}\right) - w + I \\ \tau_w \frac{dw}{dt} = a(V_M - E_L) - w \end{cases} \quad (1.3)$$

Avec C_M la capacité de membrane, g_L la conductance de fuite, E_L le potentiel inverse de fuite, Δ_T le facteur de pente, V_T le seuil, I le courant d’entrée, τ_w la constante de temps d’adaptation et a le paramètre de couplage d’adaptation.

L’implémentation AdEx

Millner *et al.* (2010) ont conçu un neurone AdEx sur silicium. En utilisant différents paramètres, ils ont réussi à imiter quatre comportements neuronaux. Bien qu’ils aient fait fi des optimisations de taille et de consommation, ils montrent qu’ils sont capables de reproduire sur silicium le modèle mathématique de l’AdEx.

La modélisation MN

Le modèle Mihalas-Niebur (MN) (Mihalaş et Niebur, 2009) est une généralisation du neurone IF. Il a l'avantage de contenir des équations linéaires simples (1.4) avec des paramètres biologiques significatifs et permet d'obtenir le comportement complexe de neurones corticaux (Hamilton et van Schaik, 2011). Il est possible de simuler différentes familles de neurones en changeant la valeur de quelques paramètres.

$$\begin{cases} \frac{dI_j}{dt} = -k_j I_j & \text{avec } j = 1, \dots, N \\ \frac{dV_M}{dt} = \frac{1}{C_M} \left(I_{ex} + \sum_j I_j(t) - GV_M \right) \\ \frac{d\Theta}{dt} = aV_M - b\Theta \end{cases} \quad (1.4)$$

L'implémentation MN

Un neurone MN a été réalisé sur silicium par Hamilton et van Schaik (2011). Les premiers résultats montrent qu'il permet de reproduire le comportement du modèle MN. C'est par le biais de courants qu'ils modifient les paramètres et ainsi changent de familles neuronales. Hamilton et van Schaik montrent que ce neurone a l'inconvénient d'occuper une surface plus importante que d'autres modèles de neurone IF mais il décrit plus de familles neuronales différentes.

Il existe d'autres modèles IF comme le *Resonant & Fire* (Izhikevich, 2001) ou le *Quadratic Integrate & Fire* (Gerstner et Kistler, 2002). Leurs équations sont non linéaires, ils sont plus complexes et décrivent des familles neuronales particulières. Des montages électroniques ont été présentés pour leur implémentation sur silicium par Nakada *et al.* (2005) et Basham et Parent (2009) respectivement.

Suivant la complexité du modèle mathématique choisi, toutes les propriétés souhaitées ne sont pas toujours incluses. Par exemple, en utilisant un modèle IF simple, la période réfractaire n'est pas implémentée. La période réfractaire, tout comme l'adaptation du seuil de déclenchement des potentiels d'action ou la génération de potentiels d'action de forme particulière, sont des propriétés qui se traduisent par des blocs électroniques. Il est alors possible de construire « son » neurone en ajoutant ou en enlevant certains d'entre eux. Des exemples de montages sont proposés par Indiveri *et al.* (2011). Ils donnent également un aperçu de certaines implémentations matérielles de neurone de type *Integrate & Fire* et Hodgkin-Huxley.

D'autres modèles mathématiques de neurones existent, comme le *Quartic Integrate & Fire* (Touboul, 2008) ou le modèle d'Izhikevich (Izhikevich, 2003). Ces modèles, où les variables n'ont *a priori* pas d'origine biologique, sont majoritairement utilisés pour des implémentations numériques avec des *Field Programmable Gate Array* (FPGA) par exemple. Toutefois, des

concepteurs analogiques en ont réalisés sur silicium (Wijekoon et Dudek, 2006, 2008, entre autres).

Pour notre réseau, nous avons choisi d’avoir une approche pragmatique. Il n’était pas nécessaire pour nous de disposer de différentes familles neuronales, c’est pourquoi nous avons utilisé le modèle LIF.

Les neurones ne suffisent pas à faire un réseau, ils ont besoin d’être interconnectés les uns aux autres par l’intermédiaire de synapses.

1.2.3 La synapse et l’apprentissage

Nous ne pouvons parler de la synapse sans parler de l’apprentissage. La plasticité de la synapse permet au réseau de garder une trace des évènements passés. La synapse est une mémoire dont la valeur peut évoluer au cours du temps. Quelle que soit sa modélisation, la synapse pondère les informations en amont par l’intermédiaire de sa valeur. Aussi appelée poids synaptique ou conductance synaptique, la valeur de la synapse est modifiée lors du processus d’apprentissage par une loi de variation qui est dépendante de l’architecture du réseau et du type de neurones. Au sein du réseau de neurones, la synapse a donc trois fonctions essentielles à remplir : la plasticité, la mémorisation et la pondération.

Nous allons aborder les différentes formes de plasticités synaptiques qui sont régies par des lois d’apprentissage liées au type neurone utilisé.

1.2.3.1 La plasticité liée aux neurones formels

Pour les neurones formels (figure 1.9), la synapse est représentée par une valeur de poids synaptique w_i . Les entrées x_i du neurone sont alors pondérées par les coefficients synaptiques. Puis, comme nous l’avons vu au 1.2.1, le neurone fait la somme de ces contributions et une fonction d’activation f_a y est appliquée. Lors de l’apprentissage, ce sont les valeurs w_i qui vont être modifiées en fonction d’une loi de variation.

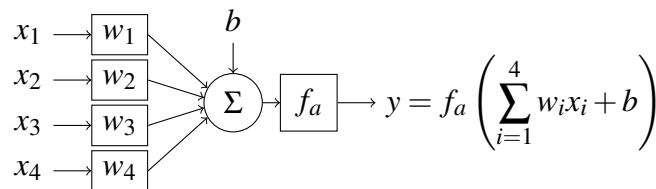


FIGURE 1.9 – Illustration et équation du neurone formel avec les poids synaptiques.

Pour les synapses de neurones formels, les lois d’apprentissage sont mathématiques et n’ont plus de rapport direct avec la biologie. Elles sont liées à l’agencement des neurones entre eux. C’est pourquoi nous reviendrons sur ces lois dans le chapitre 3 qui abordera la problématique des architectures des réseaux.

1.2.3.2 La plasticité liée aux neurones impulsionnels

Dans le cas des neurones impulsionnels, la loi de variation des poids synaptiques s'inspire des observations biologiques. La théorie de [Hebb \(1949\)](#) est une référence. Comme exposée dans la partie biologie, la règle impose un renforcement synaptique entre un neurone de sortie et un neurone d'entrée qui semble l'activer. En renforçant la connexion entre ces deux entités, lorsque le neurone d'entrée s'activera de nouveau, il contribuera davantage à l'activation du neurone de sortie. La connexion sera alors fortifiée jusqu'à atteindre un seuil de saturation. Cette loi a la particularité d'être locale : les variations des valeurs des synapses ne dépendent que des événements des neurones qui y sont connectés.

La plasticité synaptique biologique se décline sous différentes formes. Dans le cas d'un réseau de neurones computationnels, seules deux plasticités nous intéressent plus particulièrement : la potentialisation à long terme (en anglais *Long Term Potentiation*, [LTP](#)) et la dépression à long terme (en anglais *Long Term Depression*, [LTD](#)). La réunion de ces deux phénomènes donne une règle complexe de plasticité synaptique, appelée *Spike-Timing-Depend Plasticity* ([STDP](#)), qui dépend du temps de déclenchement des potentiels d'action ([Bi et Poo, 1998](#)). Certains chercheurs s'intéressent également à la potentialisation et/ou à la dépression à court terme, toujours dans le but de reproduire le comportement biologique, mais ces phénomènes sont très peu utilisés pour des réseaux de neurones bioinspirés.

Des modèles prenant en compte des mécanismes biophysiques précis ont été élaborés. Celui présenté par [Badoual et al. \(2006\)](#) s'appuie sur le formalisme de Hodgkin et Huxley, et décrit l'évolution des différents canaux ioniques au niveau de la synapse. Mais ce modèle reste complexe et nécessite plusieurs équations différentielles pour chaque synapse.

La figure [1.10](#) montre un fonctionnement d'une loi de [STDP](#). Dans la description temporelle (figure [1.10.a](#)), les événements forment des exponentielles décroissantes $P(t)$ et $Q(t)$ dont la valeur est maximale au moment de leur apparition. Dans le cas de la [LTP](#), un événement présynaptique apparaît à t_j avant qu'un événement postsynaptique ne se déclenche à t_i ; il y a alors une augmentation de la conductance d'une valeur $P(t_i - t_j)$. Au contraire, dans le cas de la [LTD](#), un événement présynaptique arrive en t_j après un événement postsynaptique en t_i ; la conductance subit une diminution d'une valeur $Q(t_j - t_i)$. Dans le cas où les événements sont trop éloignés, aucune modification de la conductance n'est réalisée : les événements sont considérés comme étant non-corrélés. Le graphique (figure [1.10.b](#)) montre la variation relative de poids synaptique Δw par rapport à la différence de temps des événements pré- et postsynaptiques. La forme obtenue est caractéristique d'une forme de [STDP](#) que nous qualifierons de « classique ».

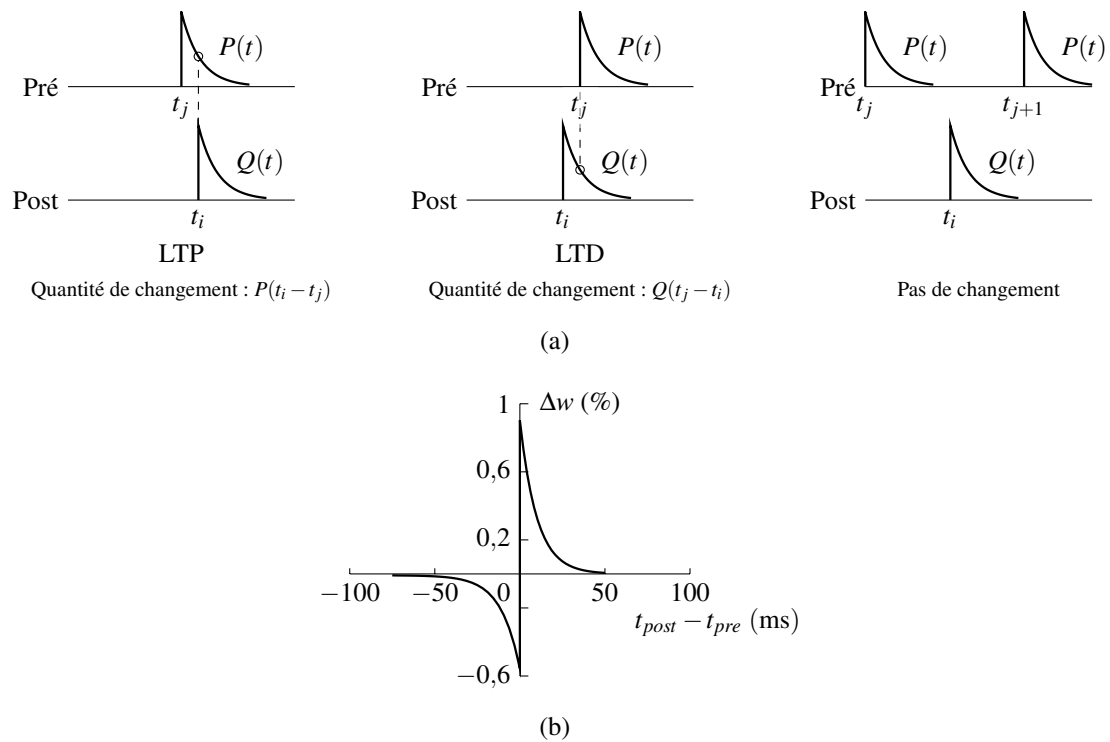


FIGURE 1.10 – (a) Description temporelle de la règle de plasticité. (b) Simulation montrant la STDP résultante dite « classique ». Figure reproduite de la publication [Badoual et al. \(2006\)](#).

1.2.3.3 L'implémentation matérielle des synapses et de leurs lois d'apprentissage

Alors qu'en biologie la synapse est réduite à une fente de quelques dizaines de nanomètres entre deux neurones, pour son implémentation matérielle, elle occupe autant voire plus de place qu'un neurone. La synapse joue le rôle de trois fonctions (plasticité, mémoire et pondération) qui sont généralement décomposées en trois blocs électroniques.

- Des blocs de calculs sont nécessaires pour rendre la synapse plastique.
- Une mémoire dédiée enregistre le poids synaptique. Souvent volatile, la mémoire nécessite un rafraîchissement régulier.
- Un bloc de multiplication permet de pondérer les entrées présynaptiques arrivant sur la membrane du neurone postsynaptique.

La synapse étant l'élément le plus nombreux dans un réseau de neurones, la nécessité de ces trois blocs augmente considérablement la place et la consommation des réseaux de neurones impulsionnels analogiques.

La publication de [Arthur et Boahen \(2006\)](#) donne un exemple de la complexité d'implémentation d'une loi d'apprentissage temporelle. Ils utilisent 21 circuits de STDP par neurone occupant une place non-négligeable de 3800 μm^2 environ sur la puce ($\sim 1520 \mu\text{m}^2$ pour un neurone). [Wijekoon et Dudek \(2012\)](#) montrent plusieurs autres exemples d'implémentations de synapses

avec **STDP**. Dans tous les cas, le circuit occupé par les synapses est plus imposant que celui occupé par les neurones.

Alors que les synapses utilisées dans la majorité des réseaux nécessitent plusieurs niveaux accessibles de poids synaptiques pour permettre d'avoir un apprentissage correct, **Fusi et al. (2000)** présentent une synapse à deux niveaux. Cette simplification permet de réduire à une dizaine le nombre des transistors nécessaires pour construire une synapse.

Dans les premiers modèles de réseaux de neurones, comme l'adaline présenté par **Widrow (1960)**, la synapse était implémentée par un seul composant : un tripôle appelé *memistor*. Ce composant à base de platine peut faire varier sa résistance en fonction de l'intégrale du courant injecté dans une borne de contrôle. Les deux autres extrémités servent alors de résistance. Sa description et son comportement en forme d'hystérésis ont quelques points communs avec le memristor que nous décrirons en détail au chapitre 2. Ce composant d'un nouveau genre et d'une taille de quelques centimètres n'a jamais été intégré.

Diorio et al. (1996, 1997) souhaitent aussi n'utiliser qu'un seul composant pour jouer le rôle de la synapse. Pour ce faire, ils emploient un transistor à grille flottante. En plus d'utiliser sa propriété de mémorisation, ils arrivent à reproduire des formes de **STDP** (**Ramakrishnan et al., 2011**) en appliquant des formes de tension adéquates et en s'appuyant sur la physique de leur composant (utilisation de l'injection de charges et de l'effet tunnel). L'avantage de cette solution est la diminution de l'espace que les synapses occupent sur les gallettes de silicium. Il existe aussi des circuits permettant de contrôler la forme de **STDP** générée par des transistors à grilles flottantes comme le montre **Smith et al. (2014)** mais au prix de l'augmentation de la surface occupée.

Les avancées en microélectronique sur les mémoires ont été utilisées pour les synapses, comme nous venons de le voir avec les transistors à grilles flottantes. C'est aussi en cherchant à réduire la place que peut prendre un bloc mémoire, qu'en 2008, l'équipe HP de R. S. Stanley a mis au point un nouveau composant, le memristor, avec des propriétés qui avaient été décrites dans le papier du théoricien **Chua (1971)**. Bien plus qu'une simple mémoire, comme nous le verrons dans le chapitre 2, les composants memristifs sont des plots de matériaux de quelques centaines de nanomètres de diamètre qui regroupent les trois propriétés d'une synapse. Cela en fait des candidats idéaux pour être utilisés dans les réseaux de neurones en tant que synapses. C'est dans cette dynamique que le projet *Memristive Hardware Analog Neural Network* (**MHANN**) a été créé et a pour but le développement d'un réseau de neurones sur silicium dont les connexions synaptiques sont formées par des composants memristifs ferroélectriques.

1.3 Un projet bioinspiré : MHANN

Nous venons de faire un tour d’horizon des modèles de neurones existants et des possibilités d’implémenter les synapses. De nos jours, malgré l’évolution des technologies, il n’est pas possible de créer des réseaux de neurones à grande échelle de manière simple. Les projets européens comme SenseMaker (Meier, 2002), FACETS (Meier, 2005) ou BrainScaleS (Meier, 2011), ou les projets comme Neurogrid (Benjamin *et al.*, 2014) ou SpiNNaker (Davidson, 2005) montrent qu’il est encore nécessaire d’employer de grands moyens et que la construction de réseaux reste difficile.

La mise en parallèle des cœurs de processeurs due à l’arrêt en montée de leur fréquence, conjuguée à l’apparition des memristors a donné de nouvelles perspectives pour les réseaux de neurones de grande échelle.

En effet, différentes communautés scientifiques ont montré un engouement très important pour les composants memristifs comme le révèle le graphique de la figure 1.11. Dans la communauté électronique et système et la communauté de l’ingénierie neuromorphique représentée avec IEEE et ScienceDirect, aussi bien que dans la communauté des matériaux représentée avec Wiley, le nombre de papiers ayant comme occurrence « memristor » ne cesse d’augmenter.

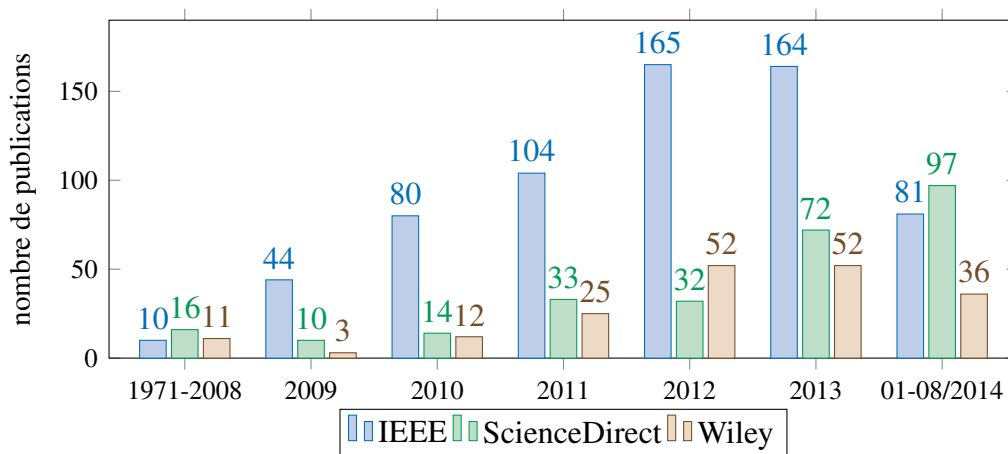


FIGURE 1.11 – Graphique montrant l’évolution du nombre de papiers ayant comme mot clé « memristor » dans différentes bibliothèques électroniques depuis 1971 jusqu’à août 2014.

Des équipes du monde entier cherchent à travailler sur les composants memristifs. Certaines d’entre elles, ayant les moyens et les compétences, ont développé leur propre technologie alors que d’autres modélisent des circuits à base de memristors à partir des premières mesures publiées. Cependant, peu de systèmes réels ont été mis en œuvre.

Le projet *Memristive Hardware Analog Neural Network* (MHANN) s’inscrit dans l’évolution des systèmes bioinspirés et souhaite mettre en place un démonstrateur d’un réseau de neurones sur silicium avec des connexions synaptiques memristives. Ce projet ANR-P2N⁴ a été financé à

⁴Programme de l’Agence National de la Recherche finançant les projets de recherche dans les thématiques des nanotechnologies et des nanosystèmes.

hauteur de 740 k€ pour une durée de 4 ans. Il s'appuie sur la collaboration de trois laboratoires et d'un partenaire industriel :

- l'UMφ CNRS/Thalès apporte la technologie des memristors ferroélectriques qui serviront de synapses ;
- l'INRIA de Saclay étudie les architectures de réseaux de neurones impulsionnels ;
- l'IMS de Bordeaux conçoit un prototype avec une puce comprenant des neurones sur silicium (sujet de cette thèse) ;
- Thalès porte un intérêt à l'utilisation de ce type de technologie dans le futur.

Nous verrons dans la suite de ce manuscrit pourquoi les neurones utilisés sont impulsionnels et comment il est possible d'utiliser des composants memristifs au sein de réseaux de neurones plastiques.

Le projet **MHANN** n'a pas pour but de reproduire le vivant mais de s'en inspirer. Les dynamiques sont similaires à celles observées en biologie mais les constantes de temps sont plus rapides. Alors que le système nerveux fonctionne avec des temps de l'ordre de la milliseconde, voire la centaine de millisecondes, dans ce projet qui repose sur des memristors ferroélectriques, les constantes de temps sont de l'ordre de la centaine de nanosecondes. Le projet **MHANN** veut montrer qu'il est possible d'une part de piloter des composants memristifs ferroélectriques avec des technologies CMOS classiques de la microélectronique, et d'autre part d'obtenir des calculateurs parallèles, à base de memristors, rapides et performants.

1.4 Conclusion

Ce premier chapitre a permis d'introduire le vocabulaire provenant de la neurobiologie. Nous avons examiné le fonctionnement du neurone et la génération de potentiels d'action qui véhiculent l'information. Puis nous avons donné un aperçu des mécanismes de la synapse et introduit les termes liés à l'apprentissage.

Nous avons ensuite exploré les différentes modélisations des neurones, des synapses et leurs implémentations matérielles respectives. Nous avons constaté que les modélisations les plus communes considèrent le neurone comme étant ponctuel et s'affranchissent ainsi de la géométrie des cellules. Nous avons vu qu'il était possible d'utiliser des modèles simples de neurones et de les complexifier en ajoutant des propriétés à l'aide de blocs électroniques.

Enfin nous avons présenté le projet pluridisciplinaire **MHANN** sur lequel s'appuie cette thèse. Le projet a pour ambition de réaliser un prototype d'un réseau de neurones sur silicium connectés avec des synapses memristives.

Avant de détailler le réseau de neurones, nous allons nous attarder sur les composants memristifs en général puis plus particulièrement sur ceux utilisant la technologie ferroélectrique.

CHAPITRE 2

LES COMPOSANTS MEMRISTIFS

DEPUIS 2008 et la publication de [Strukov *et al.* \(2008\)](#), la théorie des memristors, énoncée par [Chua \(1971\)](#), n'est désormais plus orpheline de son composant physique. Bien que des équipes à travers le monde eussent mesuré des comportements électriques particuliers venant de la superposition de différentes couches de matériau, elles n'avaient pas fait le rapprochement avec la théorie formulée dans les années soixante-dix. Cette mise au jour a été faite dans le cadre de recherche de nouvelles technologies pour les mémoires. En effet, les mémoires flashes actuelles sont composées d'un transistor à grille flottante dont la réduction de taille est tributaire des finesses de gravure des technologies silicium classiques qui atteignent leurs limites.

Dès 1995, [Wulf et McKee \(1995\)](#) ont lancé le débat sur les problèmes que peuvent engendrer les mémoires dans les systèmes de calcul. Effectivement, alors que les processeurs voyaient leurs fréquences augmenter, les temps d'accès et d'écriture des mémoires stagnaient. Pour des raisons de consommation, la diminution des périodes d'horloges des processeurs a été stoppée. Pour continuer à respecter la loi empirique sur l'accroissement des puissances de calcul, les fondateurs ont alors augmenté le nombre de cœurs de calcul. Mais [Moore \(2008\)](#) révèle la non-efficacité de l'augmentation du nombre des unités de calculs si l'architecture classique est conservée. Il montre que 2 cœurs sont aussi efficaces en terme de rapidité de calcul que 16. Ce problème est dû à la fois aux temps et aux gestions des accès mémoires qui ralentissent l'efficacité du calcul.

Des recherches ont été menées dans des laboratoires pour trouver d'autres manières de concevoir la mémoire. Le *crossbar* à semiconducteurs est un candidat potentiel à la miniaturisation des points mémoires. Il est constitué de rangées d'électrodes superposées perpendiculairement et reliées par des interrupteurs comme illustré par la figure 2.1. Les interrupteurs électriques forment une mémoire matricielle. Leurs états ouverts ou fermés permettent d'enregistrer des mots binaires.

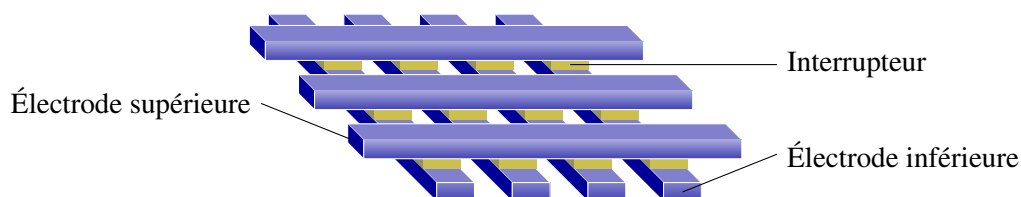


FIGURE 2.1 – Schéma d'un *crossbar*.

Pour que la solution du *crossbar* soit possible, il faut que l'interrupteur puisse changer d'état en fonction soit de la tension présente entre ses bornes, soit du courant qui le traverse. Et pour que ce soit viable, il faut ajouter une condition : la tension de commande ne doit pas être très élevée (de l'ordre du volt) pour rester compatible avec les technologies CMOS actuelles. [Strukov *et al.* \(2008\)](#) ont démontré que l'utilisation d'oxyde de titane permet d'avoir un changement de résistance de l'interrupteur commandé électriquement en utilisant des tensions de l'ordre du volt. En passant d'une résistance forte (R_{OFF}) à une résistance faible (R_{ON}) et inversement, deux états sont discrétisés. Ils ont également mis en évidence des propriétés supplémentaires : la courbe de courant-tension présente une hystérésis. En rapprochant ce comportement de la théorie de Chua,

ils ont ainsi annoncé avoir conçu un memristor.

Dans la théorie de 1971, le memristor est identifié comme étant le quatrième élément passif. Les trois autres éléments passifs, le condensateur, la résistance et l'inductance, sont connus depuis le XIX^e siècle. Ainsi, avec l'apparition de ce nouvel élément, il existe une relation entre chacune des quantités physiques électriques : le courant i , la tension v , la quantité de charge q et le flux magnétique φ (figure 2.2.a).

Le memristor est alors l'élément qui lie la quantité de charge électrique q au flux magnétique φ . La memristance M est donnée par la relation suivante :

$$M = \frac{d\varphi}{dq} \tag{2.1}$$

En 1976, Chua et Kang (1976) précisent les propriétés du memristor et donnent comme caractéristique l'hystérésis de Lissajous (figure 2.2.b) résultante de la relation du courant traversant le composant en fonction de la tension qui se trouve à ses bornes. Le memristor est donc un élément résistif dont la valeur de la résistance dépend du courant qui le traverse. Dans la plupart de ces composants il existe des « zones de seuil », *i.e.* il faut que le courant soit suffisamment fort pour qu'il y ait une variation de résistance. Dans le cas contraire, la résistance ne change pas et reste à la valeur de la dernière modification induite par un courant plus fort : c'est l'effet mémoire. Il n'existe pas de seuil à proprement parler, comme pour une diode par exemple, car les variations de résistance dépendent de la valeur du courant injecté mais aussi du temps pendant lequel le courant est appliqué. Toutefois il est possible de définir un seuil pour un pulse de durée fixe. Il est à noter que nous avons parlé de variation de résistance grâce à un courant injecté, mais il est aussi possible d'obtenir cette variation en appliquant une différence de potentiel aux bornes du memristor. Le phénomène sur les seuils reste le même.

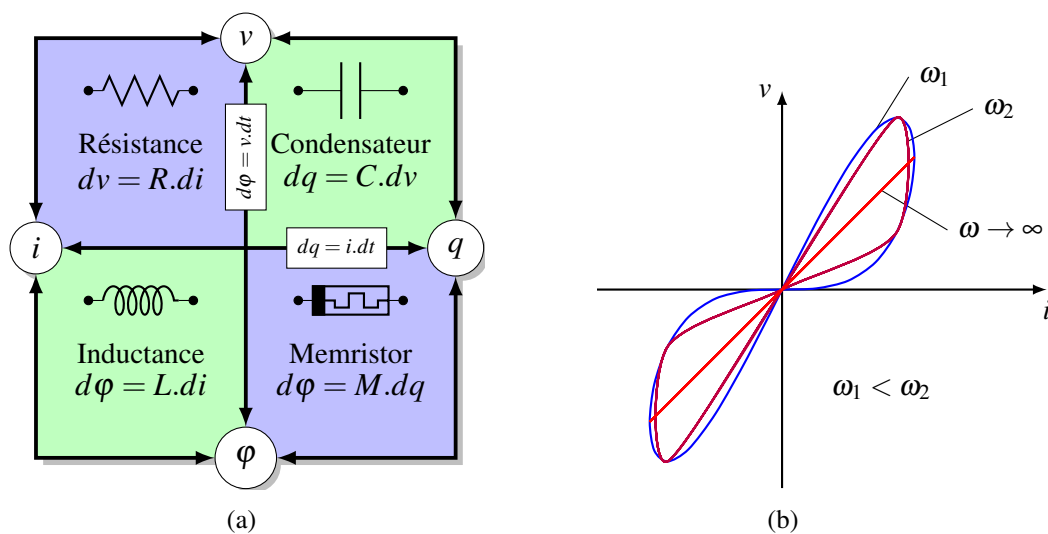


FIGURE 2.2 – (a) Relations entre les grandeurs physiques électriques et les composants électriques passifs. (b) Courbes d'hystérésis de Lissajous caractérisant le comportement memristif d'un composant d'après Chua et Kang (1976).

Les composants memristifs ont donc la capacité de garder en mémoire la valeur de leur résistance, sans aucune consommation de courant supplémentaire. Dans le cas où il est possible d'avoir accès à plusieurs valeurs de résistance, il devient intéressant d'utiliser ce composant pour réaliser une synapse. En effet, comme nous l'avons vu au chapitre précédent, une synapse permet une pondération en fonction de sa valeur de conductance. Qui plus est, l'élément le plus nombreux dans un réseau de neurones est la synapse. Donc avoir une mémoire passive peut réduire drastiquement la consommation du réseau.

2.1 L'apprentissage par loi temporelle

Snider (2008) puis Linares-Barranco et Serrano-Gotarredona (2009b) ont montré qu'il était possible d'utiliser la plasticité intrinsèque des memristors pour implémenter des lois de STDP ayant une dynamique proche de celle observée en biologie. Yu *et al.* (2011), Wu *et al.* (2012) et Kuzum *et al.* (2012b) utilisent des séries de pulses pour obtenir différentes formes de STDP, comme illustré à la figure 2.3. Avec cette solution il est alors indispensable d'avoir un circuit de contrôle pour que les pulses soient synchronisés et ainsi obtenir le bon changement de résistance. Cette méthode est *a priori* applicable à toutes les technologies.

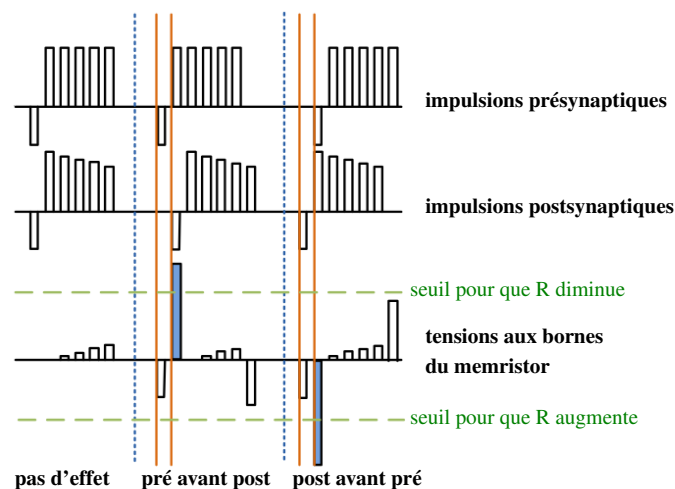


FIGURE 2.3 – Exemple d'implémentation de séries de pulses permettant la génération d'une forme de STDP. Illustration provenant de Wu *et al.* (2012).

Les publications de Linares-Barranco et Serrano-Gotarredona (2009a) et de Zamarreño Ramos *et al.* (2011) démontrent que la forme des impulsions (ou potentiel d'action (PA)) appliquées aux bornes du composant memristif donne les contours de la plasticité résultante, comme illustré à la figure 2.4.

Les règles de STDP imposent d'avoir un ajustement de la conductance synaptique en fonction des temps d'arrivée des évènements pré- et postsynaptiques. Avec les memristors, c'est la valeur de la résistance (inverse de la conductance) qu'il est possible de faire varier en lui appliquant des pulses de tension (ou de courant) suffisamment élevée et de durée limitée. En fonction du

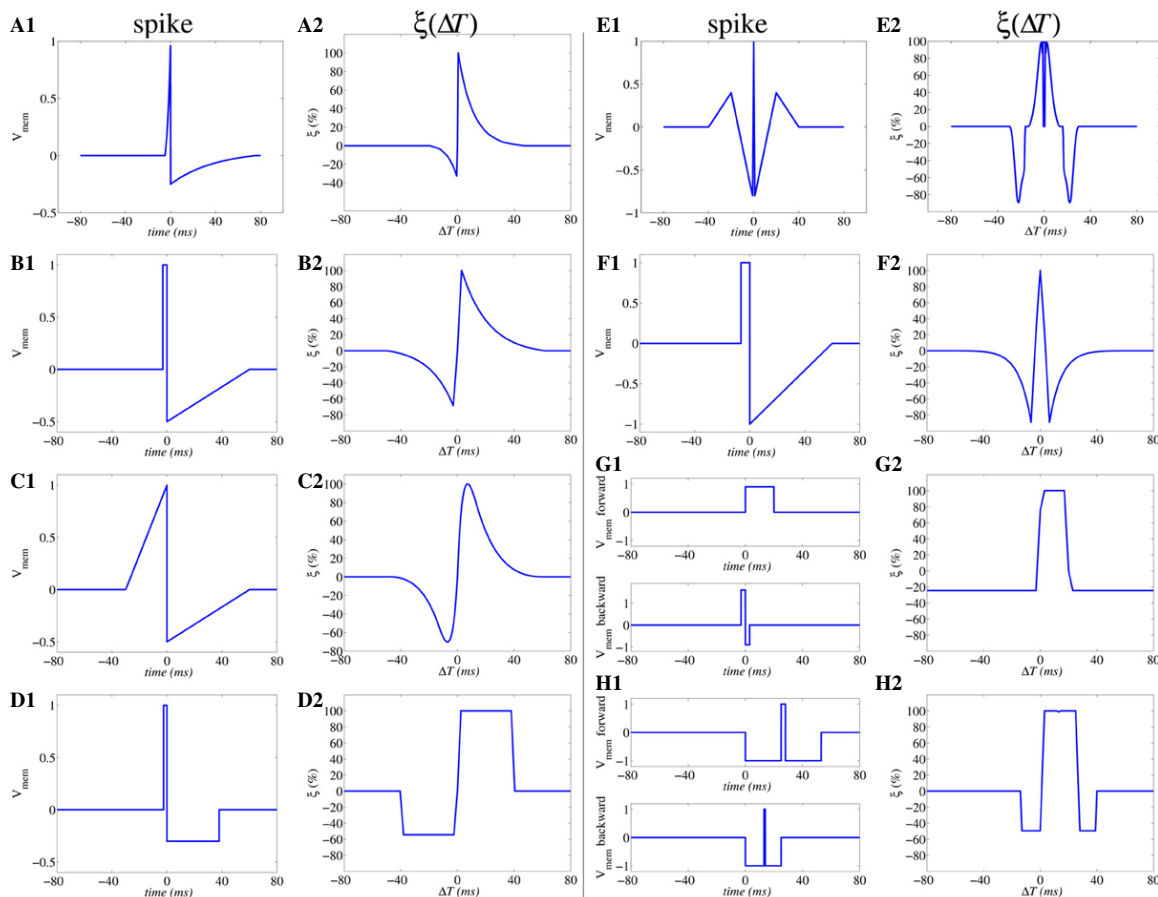


FIGURE 2.4 – Formes des STDP dépendantes des pulses appliqués aux bornes des éléments memristifs. Les formes des PA pré- et postsynaptiques illustrées en X1, avec $X \in [A, E]$, sont appliquées à chacune des bornes du memristor avec un décalage ΔT . Les formes de STDP résultantes sont données en X2, avec $X \in [A, E]$. Le PA présynaptique F1 et le PA postsynaptique de polarité opposée à F1 sont appliqués aux bornes du memristor pour obtenir la forme de STDP F2. Avec les PA en G1 et H1, il est possible de ne pas utiliser une forme de PA identique en position pré- et postsynaptique. Ces résultats de simulation proviennent de Serrano-Gotarredona *et al.* (2013).

temps d'arrivée des PA pré- et postsynaptiques, la coïncidence de leurs formes permet d'avoir des pulses de tension suffisamment élevée pour modifier la valeur de la résistance. Le choix des formes des PA est alors fait en fonction de la forme de STDP voulue et des caractéristiques du composant memristif.

En utilisant la méthode de chevauchement des PA, il n'y a pas besoin de circuit supplémentaire pour changer la valeur de résistance du memristor. De plus, le changement des poids synaptiques est réalisé immédiatement. Deux des trois fonctions de la synapse sont alors remplies : la mémorisation et la plasticité. Pour utiliser cette méthode il est nécessaire d'avoir un memristor qui possède des niveaux intermédiaires de résistance.

Les recherches de ces dernières années se concentrent à trouver et améliorer les assemblages de matériaux pour créer des composants memristifs répondant aux besoins suivants : mémoire binaire et/ou imitation du comportement synaptique. Il existe donc différentes technologies de memristor. Nous allons faire un état de l'art des memristors utilisés en ingénierie neuromorphique

et en particulier ceux utilisés pour implémenter une **STDP**.

2.2 L'état de l'art des mesures de STDP avec des composants memristifs

Dans toutes les technologies existantes, aucun modèle comportemental précis n'a été trouvé car les phénomènes physiques fins mis en jeu ne sont pas encore entièrement compris. Néanmoins les phénomènes physiques prédominants sont connus et nous allons les utiliser pour classer les différents types de composants memristifs répertoriés.

Comme nous l'avons vu dans le chapitre précédent, il existe différentes formes de **STDP**. Néanmoins une **STDP** observée dans le cortex par différentes équipes, et notamment par **Bi et Poo (1998)**, puis modélisée avec des exponentielles par **Badoual et al. (2006)** (figure 1.10.b) est prise comme exemple dans la majorité des papiers pour justifier le comportement biologique du système. Nous qualifierons cette forme de **STDP** (identique à la forme **B2** de la figure 2.4) de « classique ». La forme peut être qualifiée de classique sans toutefois respecter les temps biologiques.

2.2.1 Le changement de phase

Les matériaux sont constitués d'un agencement d'atomes ou de molécules qui définit leur état : solide, liquide ou gazeux. Cet état dépend de la température, la pression, la masse volumique, *etc*, et suit les lois de la thermodynamique. Pour passer d'un état à un autre, le matériau peut transiter par différentes phases. C'est précisément cette propriété qui est utilisée pour créer des interrupteurs résistifs à changement de phase. La technologie est bien connue et maîtrisée car elle reprend le principe appliqué pour le stockage de données sur support optique (CD ou DVD par exemple).

Les matériaux sont à l'état solide et passent d'une phase cristalline à une phase amorphe et inversement. Pour ce faire, des pulses de laser calibré, ou de courant électrique, sont utilisés. Suivant la puissance du laser, ou l'intensité du courant, le matériau change de phase. Il voit alors sa résistance modifiée.

Les composants utilisant cette technologie sont unipolaires, c'est-à-dire que l'intensité du courant utilisée pour changer la valeur de la résistance est toujours injectée dans le même sens. Ce sont la valeur de l'intensité et la durée d'application qui vont permettre d'obtenir un changement de phase différent comme illustré à la figure 2.5.

Kuzum et al. (2012a) utilisent un composant memristif à changement de phase (ou en anglais *Phase Change Memory*, PCM). Ils montrent qu'ils sont capables d'obtenir une **STDP** « classique » (figure 2.6.a) proche de la biologie tant par le changement relatif de la valeur de la conductance que par les différences de temps auxquelles les conductances sont modifiées (de l'ordre de la milliseconde). L'obtention de la **STDP** se fait à l'aide d'une série de pulses positifs pour les

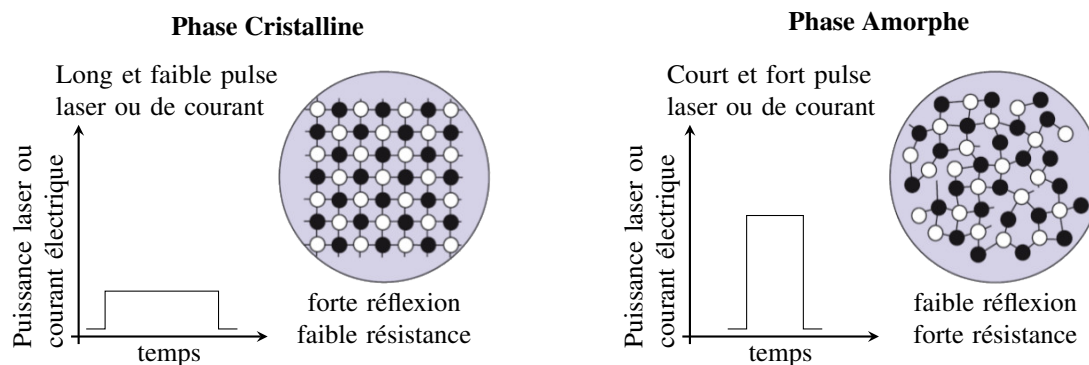


FIGURE 2.5 – Schéma des deux phases d’un matériau à changement de phase et caractéristiques des pulses nécessaires pour la transformation dans la phase indiquée pour permettre la modification de résistance. Illustration tirée de [Wuttig et Yamada \(2007\)](#).

neurones présynaptiques et d’un pulse négatif pour les neurones postsynaptiques. L’amplitude des pulses est inférieure au volt. Comme ils utilisent des séries de pulses, ils ont été en mesure d’obtenir d’autres formes de [STDP](#) (figure 2.6.b).

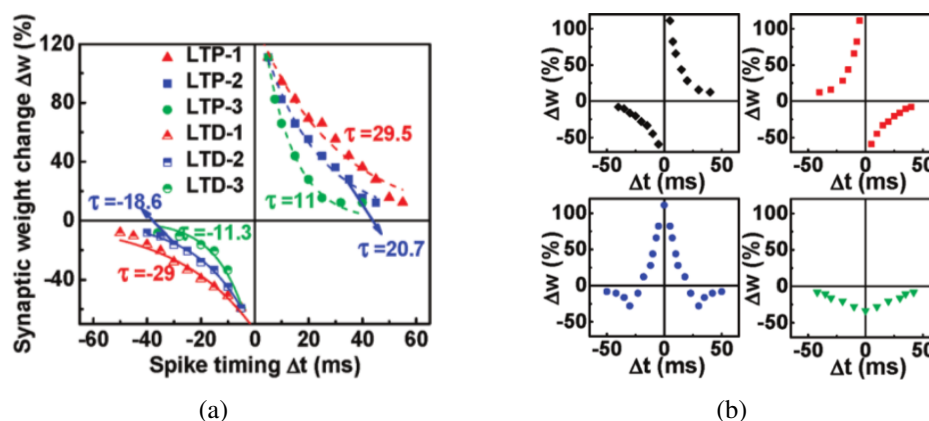


FIGURE 2.6 – Mesures de formes de [STDP](#) obtenues avec des memristors utilisant des matériaux à changement de phase. Illustrations provenant de [Kuzum et al. \(2012a\)](#).

2.2.2 L’oxydoréduction

Une réaction d’oxydoréduction (ou redox) est caractérisée par un transfert d’électrons entre deux éléments chimiques : un oxydant et un réducteur. Le premier subit une réduction, il capte un ou plusieurs électrons, alors que le second va s’oxyder en perdant un ou plusieurs électrons. Néanmoins, les mécanismes produits entre les électrodes sont complexes et dépendent principalement du champ électrique et de l’augmentation de la température induite par effet Joule.

Un des phénomènes observés dans les composants memristifs à base de composé métallique est l’apparition d’un filament d’atomes (figure 2.7). Les atomes provenant de l’électrode active¹

¹Électrode participant aux réactions d’oxydoréduction.

subissent une oxydation et forment des ions. Sous l'effet du champ électrique, les ions se dirigent vers l'électrode passive². Ils subissent alors une réduction. Les atomes ainsi créés s'associent pour former un filament continu de l'électrode passive jusqu'à l'électrode active. Plus le filament est proche de l'électrode active, moins la résistance est grande. La résistance est la plus faible lorsque les deux électrodes sont liées par le filament. En inversant les potentiels aux bornes du composant la réaction d'oxydoréduction change de sens et le filament se désagrège. Ces composants sont donc bipolaires : l'augmentation ou la diminution de résistance se fait suivant le sens du champ électrique.

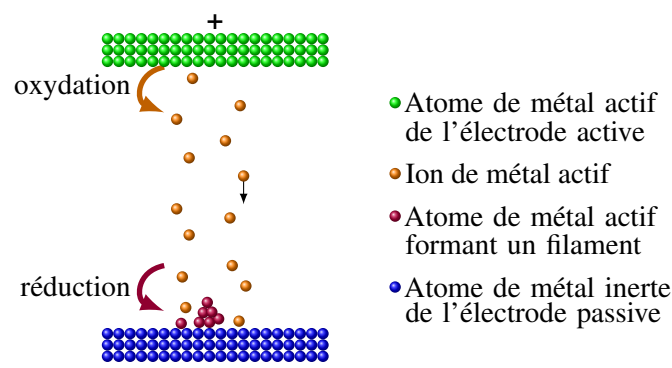


FIGURE 2.7 – Schéma montrant le mécanisme d'oxydoréduction dans un composant memristif. Illustration inspirée de Pan *et al.* (2010).

Le mécanisme présenté est un exemple de fonctionnement d'un composant particulier mettant en œuvre l'oxydoréduction. Il existe d'autres composants n'utilisant pas de matériaux métalliques où les réactions sont quelque peu différentes. De plus, nous avons présenté les réactions dominantes mais il existe d'autres effets moins bien maîtrisés et caractérisés.

Des mesures de **STDP** ont été réalisées avec différents types de matériaux. Jo *et al.* (2010) présentent un memristor bipolaire à composé métallique Ag/Si et Seo *et al.* (2011) une mémoire résistive d'oxyde Pt/Al/TiO_{2-x}/TiO_y/W. Ils présentent des formes de **STDP** « classique » (figure 2.8.a et 2.8.b) réalisées en changeant la durée de leurs pulses appliqués aux bornes composants. Les premiers utilisent des pulses d'amplitude fixe de 3,2 V et – 2,8 V suivant le sens du changement et d'une durée de l'ordre de la centaine de microsecondes. Les seconds emploient des pulses d'une dizaine de millisecondes et d'amplitude non clairement définie. Avec le memristor bipolaire InGaZnO (α -IGZO) de Wang *et al.* (2012) des pulses de ± 5 V et de 50 ms sont utilisés pour obtenir une forme de **STDP** « classique » (figure 2.8.c). Le memristor bipolaire TiN/HfO₂/TiN permet à Subramaniam *et al.* (2013) de tracer des formes de **STDP** « classique » (figure 2.8.d) avec différentes valeurs initiales de résistance. Ils utilisent des pulses d'amplitude avoisinant les 2 V et d'une durée d'une dizaine de millisecondes. Un tracé de **STDP** « classique » (figure 2.8.e) est montré par Lai *et al.* (2010) et réalisé à l'aide d'un nano-fil transistor de polymère/Si. Les tensions appliquées vont jusqu'à 5 V et les durées des pulses sont de l'ordre de la milliseconde.

²Électrode ne participant pas aux réactions d'oxydoréduction : elle reste intacte.

Récemment, *Li et al. (2014)* et *Mandal et al. (2014)* ont montré l'implémentation de formes de **STDP** « classique » (figure 2.8.f et 2.8.g). Les premiers utilisent une structure Ag/AgInSbTe/Ag, et envoient des impulsions similaires à la forme **B1** de la figure 2.4 d'une durée total de 45 μs (dont 5 μs pour la partie positive) et d'une amplitude comprise entre $\pm 0,5$ V ; alors que les seconds utilisent un matériau HfO_2 dopé au manganèse (Mn), et envoient des pulses d'amplitude fixe (2,5 V et $-2,0$ V) et de durée d'une à plusieurs dizaines de millisecondes pour modifier la valeur de leur dispositif. *Li et al. (2014)* ont également montré qu'ils étaient capables de reproduire d'autres formes de **STDP** biologique comme illustré à la figure 2.9.

Park et al. (2012) présentent un système de reconnaissance d'images à base de mémoire résistive ou RRAM. Les points mémoires formés par le composé W/Al/PCMO/Pt sont placés dans un *crossbar*. La **STDP** utilisée est de type « classique » (figure 2.8.h). Contrairement aux autres publications, les temps sont de l'ordre de la centaine de nanosecondes et les pulses ont une amplitude de ± 4 V.

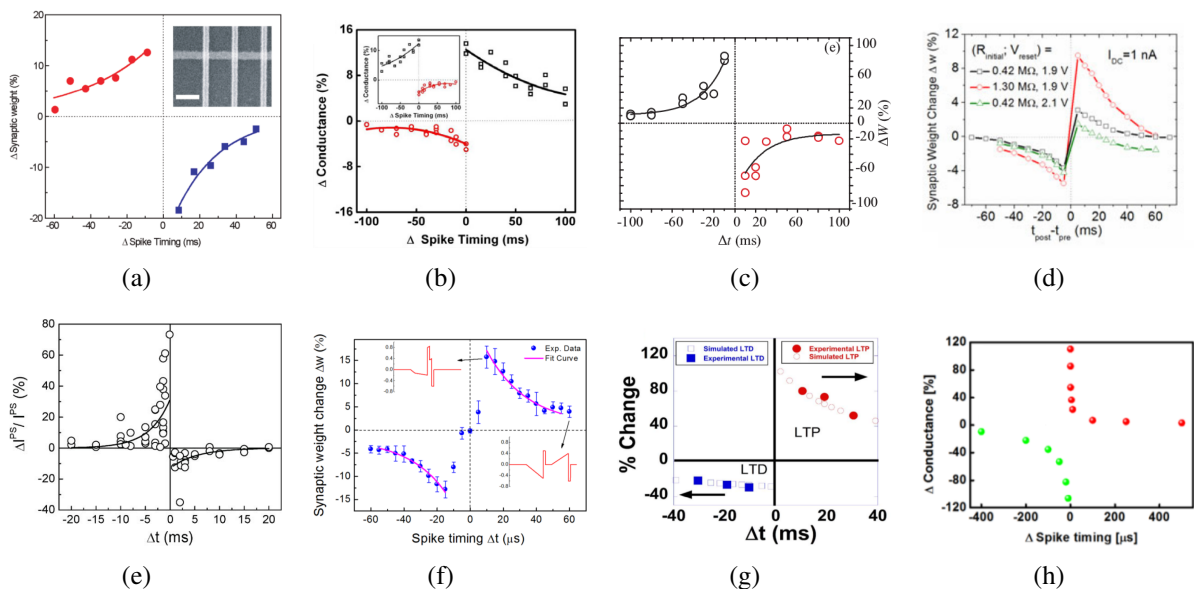


FIGURE 2.8 – Mesures de formes de **STDP** « classique » sur des dispositifs à oxydoréduction provenant de (a) *Jo et al. (2010)*, (b) *Seo et al. (2011)*, (c) *Wang et al. (2012)*, (d) *Subramaniam et al. (2013)*, (e) *Lai et al. (2010)*, (f) *Li et al. (2014)*, (g) *Mandal et al. (2014)* et (h) *Park et al. (2012)*.

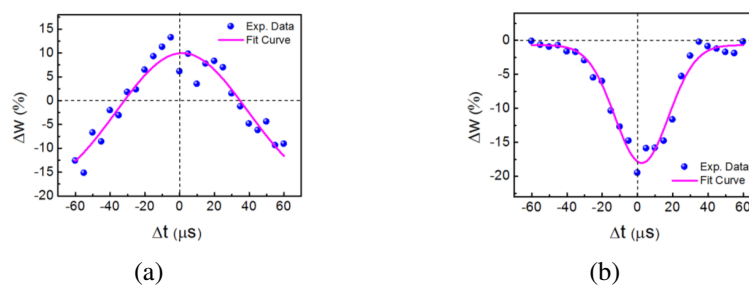


FIGURE 2.9 – Mesures des formes de **STDP** provenant de *Li et al. (2014)*.

2.2.3 L'organique

La technologie basée sur les nanoparticules organiques peut avoir aussi un comportement memristif. En connectant le drain et la grille du transistor NOMFET (de l'anglais *Nanoparticle-Organic Memory Field-Effect Transistor*), [Alibart et al. \(2012\)](#) montrent qu'il est possible d'obtenir une plasticité imitant la biologie. Ce pseudo-dipôle est bipolaire et utilise des pulses d'une durée de 100 ms et d'amplitude de ± 15 V à ± 30 V. Les auteurs présentent deux formes de STDP (figure 2.10) obtenues avec des formes de PA différentes.

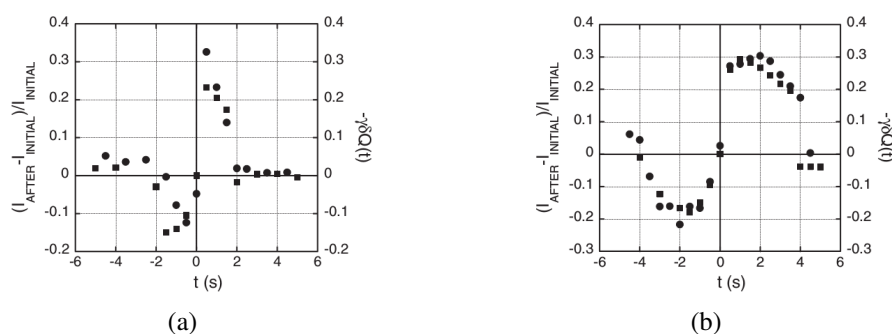


FIGURE 2.10 – Mesure de formes de STDP sur un memristor organique. STDP obtenues avec une forme de PA identique à C1 pour (a) et D1 pour (b) de la figure 2.4. Mesures provenant de [Alibart et al. \(2012\)](#).

2.2.4 Les effets purement électroniques

L'avantage des composants utilisant les effets purement électroniques pour faire varier la valeur de leur résistance est que les processus physiques en jeu ne détériorent pas le matériau. La durée de vie du composant est donc allongée.

Les jonctions tunnels magnétiques utilisent les propriétés magnétiques du matériau. Une barrière isolante est placée entre deux couches de matériaux ferromagnétiques comme illustré à la figure 2.11. La couche du dessous utilise un métal ferromagnétique dont le moment magnétique est fixe, alors que la couche du dessus est divisée en trois parties. Un mur de domaine sépare deux domaines magnétiques de moments opposés. En injectant du courant à la verticale des domaines il est possible de faire se déplacer le mur de domaine. La valeur de la résistance dépend des sens identiques ou opposés des domaines au dessus et en dessous de la couche isolante. Des domaines parallèles (à droite sur la figure) ont une plus faible résistance que des domaines de sens opposés (à gauche sur la figure).

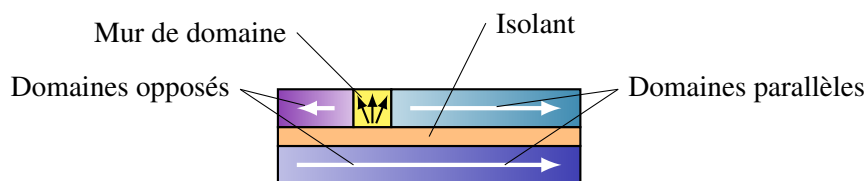


FIGURE 2.11 – Schéma des différentes couches de matériaux composant une jonction tunnel magnétique.

Krzysteczko *et al.* (2012) présentent des STDP « classiques » réalisées à l'aide de leurs technologies (figure 2.12). Ces jonctions ont une résistance de quelques centaines d'ohms ce qui ne permet pas de les utiliser dans des réseaux à grande échelle, sous peine d'avoir une consommation élevée lors des phases de lecture et d'écriture.

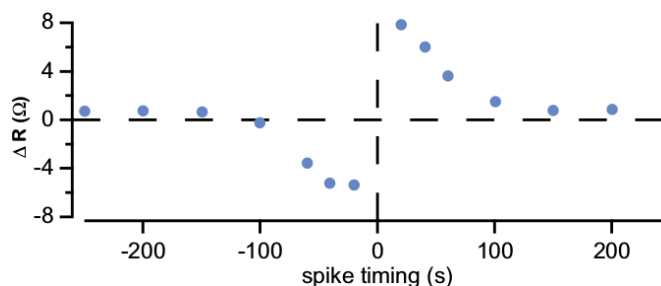


FIGURE 2.12 – Mesures de STDP sur un memristor à effets purement électroniques. Illustration provenant de Thomas (2013).

L'utilisation d'une barrière ferroélectrique entre deux électrodes métalliques permet de créer un composant memristif à effet purement électronique (Chanthbouala *et al.*, 2012). Le matériau ferroélectrique est composé de grains³, ou dipôles, de polarité à orientation variée. Les domaines réunissent les dipôles concomitants qui ont une polarité orientée vers une même direction. Lorsqu'un champ électrique est appliqué aux bornes du composant, les domaines tendent à s'orienter dans le sens du champ. Suivant la direction des domaines, la résistance du composant varie. La partie supérieure de la figure 2.13 montre l'évolution des domaines dans un memristor composé d'une barrière ferroélectrique BiFeO₃. La partie inférieure de la figure montre l'évolution de la valeur de résistance due au changement des domaines.

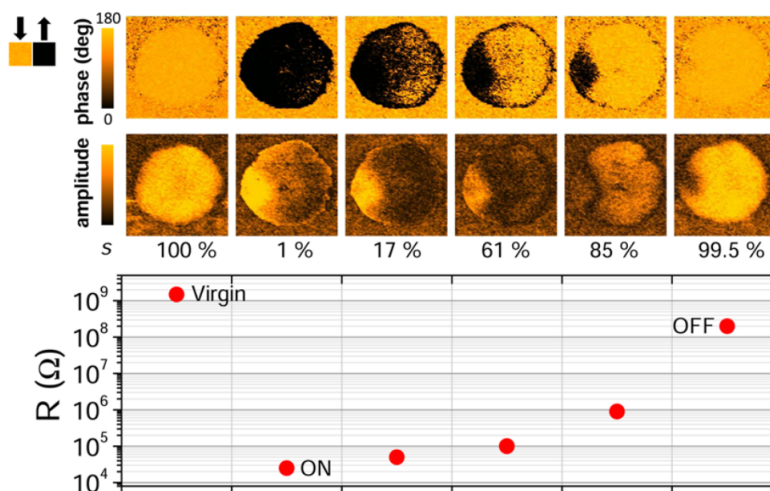


FIGURE 2.13 – Image réalisée à l'aide d'un microscope à force piézoélectrique pour différentes valeurs de résistance mesurées sur une capacité de Pt/Co/BFO/CCMO de 180 nm de large. Une estimation de la fraction des domaines dirigés vers le bas (s) est donnée pour chaque état. La taille des points de mesures de résistance représente l'erreur typique due à la prise de mesure. Illustration tirée de Yamada *et al.* (2013).

³Mailles cristallographiques formant un monocristal.

Il est possible avec un memristor ferroélectrique de mesurer une **STDP** (figure 2.14) comme le montre Schmidt *et al.* (2012). Ils appliquent une différence de **PA** (de forme identique à la forme **B1** de la figure 2.4, mais avec la partie négative réalisée à l'aide d'une exponentielle, et non d'une droite) aux bornes du memristor pour changer sa résistance. La durée de leurs impulsions est de l'ordre de la milliseconde, comparable à l'ordre de grandeur des durées biologiques.

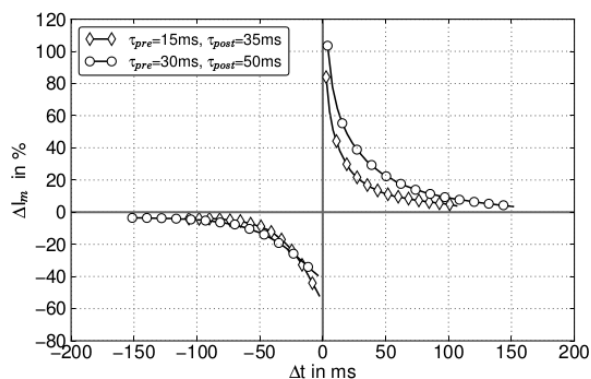


FIGURE 2.14 – Mesures de **STDP** avec un memristor ferroélectrique. Illustration provenant de Schmidt *et al.* (2012).

Nous venons de passer en revue les technologies de memristor avec lesquelles ont été mesurées des formes de **STDP**. C'est la première étape pour pouvoir utiliser le composant afin de réaliser tout ou partie de la synapse dans les réseaux de neurones. Grâce à nos partenaires du projet **MHANN** de l'UMφ CNRS/Thalès, nous avons eu accès à une technologie de memristor à barrière ferroélectrique BiFeO₃ (**BFO**). Nous avons ainsi pu réaliser nos propres mesures détaillées dans la suite de ce chapitre.

2.3 Les mesures réalisées avec des memristors BFO

Dans cette partie nous allons présenter les mesures réalisées au sein du laboratoire sur des memristors seuls (non inclus dans un *crossbar*). Pour ce faire, nous avons mis en place un banc de test (figure 2.15) piloté par ordinateur. Nous utilisons un générateur basse fréquence pour produire les pulses et autres formes d'onde. Nous utilisons un picoampèremètre pour mesurer le courant traversant le composant lorsqu'un niveau DC est appliqué par un générateur du même nom et ainsi connaître la résistance, parfois haute du memristor. Afin d'isoler le niveau continu DC de la partie dynamique AC du montage, un *bias tee* est utilisé.

Pour l'ensemble des mesures présentées, des pulses de lecture sont utilisés pour évaluer le courant qui traverse le memristor et ainsi déterminer sa résistance. Leurs amplitudes ont une valeur fixe de 0,2 V suffisamment faible pour ne pas changer la valeur de la résistance du memristor testé. Leurs durées dépendent du temps que le picoampèremètre prend pour faire une mesure et sont de l'ordre de la centaine de microsecondes.

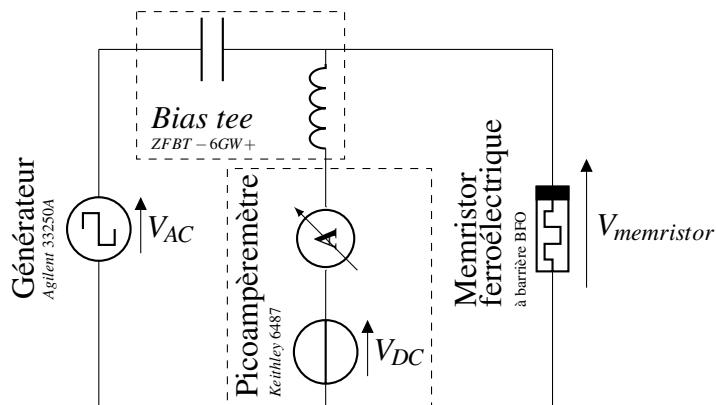


FIGURE 2.15 – Schéma du banc de test des memristors.

2.3.1 Le comportement memristif

Dans un premier temps, pour nous familiariser avec la technologie et ses contraintes, nous avons reproduit les mesures d'hystérésis comme présentées dans la publication de [Chanthbouala et al. \(2012\)](#). Dans cette première série de mesures, il s'agit d'envoyer des pulses en tension de différentes amplitudes et polarités pour tracer l'hystérésis caractérisant le comportement memristif du composant (figure 2.16).

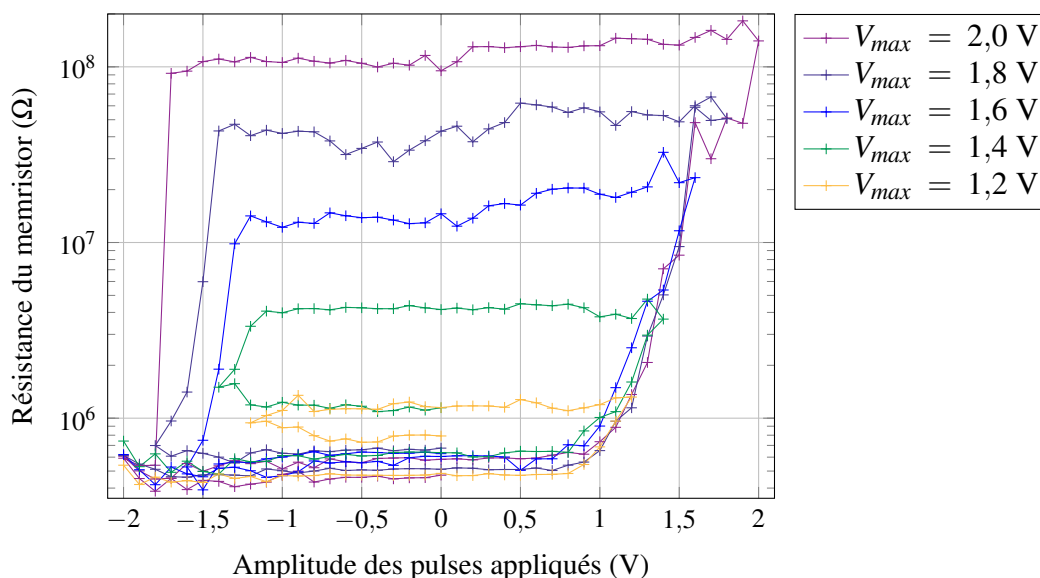


FIGURE 2.16 – Courbes d'hystérésis montrant le comportement memristif du composant. Mesures faites en appliquant des pulses en tension d'une durée fixe de 100 ns.

Nous avons appliqué une séquence de pulses en tension comme illustré à la figure 2.17 de durée fixe de 100 ns et d'amplitude variable. L'amplitude des pulses est fixée à une valeur de départ de -2 V et elle est augmentée par pas de $0,1$ V jusqu'à atteindre une tension maximale V_{max} déterminée au préalable (de $1,2$ V à 2 V). Puis l'amplitude diminue avec le même pas, jusqu'à atteindre l'opposée de la tension maximale (soit de $-1,2$ V à -2 V). Enfin, en gardant le même pas, l'amplitude augmente de nouveau jusqu'à 0 V.

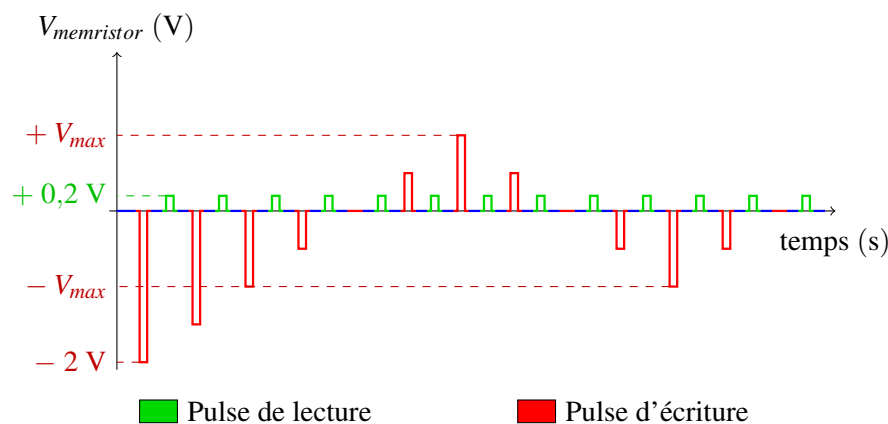


FIGURE 2.17 – Séquence de pulses afin de mesurer le comportement en hystérésis du memristor. Les pulses d'écriture ont une durée fixe de 100 ns.

Ces premières mesures mettent en lumière l'asymétrie du comportement du memristor de cette technologie. En effet, la diminution de résistance se fait de manière plus brutale que l'augmentation, à moins que les valeurs des plus hautes résistances ne soit pas atteintes comme le montrent les mesures pour $V_{\text{max}} = 1,4 \text{ V}$ ou $V_{\text{max}} = 1,2 \text{ V}$.

Grâce à ces premières mesures, il est possible de déterminer des seuils de tension de variation de la résistance lorsque sont appliqués des pulses d'une durée de 100 ns. Nous avons retenu 0,8 V pour passer d'une résistance faible à une résistance forte et $-1,0 \text{ V}$ pour l'inverse. En deçà de ces seuils, le memristor garde en mémoire sa résistance. Ces niveaux de tension sont compatibles avec les technologies CMOS analogiques où l'alimentation peut aller jusqu'à 3,3 V (soit $\pm 1,65 \text{ V}$) voire 5 V (soit $\pm 2,5 \text{ V}$).

2.3.2 Le comportement biologique

Nous avons ensuite réalisé une autre série de mesures avec des pulses pour savoir s'il était possible d'obtenir une plasticité biologiquement réaliste. Nous avons de nouveau utilisé des pulses d'une durée de 100 ns. Nous avons différencié deux catégories de pulses pour cette expérimentation : les pulses d'écriture qui ont une amplitude variable (le pas de variation est de 0,1 V), et entre chacun de ces pulses, les pulses d'initialisation qui ont une amplitude fixe. Le memristor voit donc en alternance un pulse d'initialisation et un pulse d'écriture (figure 2.18).

Dans un premier cas (figure 2.19), nous initialisons la résistance du memristor à une valeur haute (R_{OFF}) avec des pulses de 2 V, et nous appliquons une tension négative (figure 2.18). La valeur de la tension commence à 0 V et est diminuée par pas de 0,1 V entre les applications, jusqu'à atteindre la valeur de -2 V puis elle est augmentée du même pas jusqu'à 0 V. Dans le second cas (figure 2.20), nous initialisons la résistance à une valeur basse (R_{ON}) avec des pulses de -2 V , et nous lui appliquons des pulses de tension dont la valeur commence à 0 V et est augmentée jusqu'à 2 V puis diminuée à 0 V.

Dans la figure 2.19.a, cas où l'initialisation est à R_{OFF} , nous voyons apparaître un phénomène de saturation. Si nous appliquons de façon répétée des pulses positifs tandis que la résistance

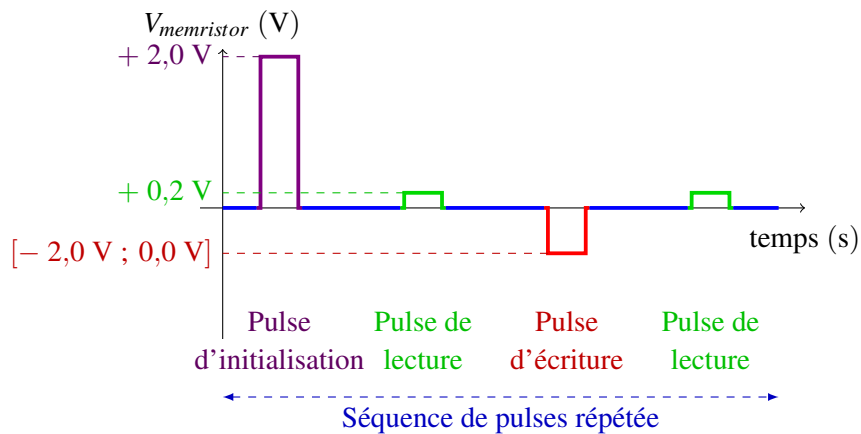


FIGURE 2.18 – Séquence de pulses du cas n° 1 avec initialisation à R_{OFF} . Dans le cas n° 2 l'amplitude des pulses d'initialisation sont de -2 V et l'amplitude des pulses d'écriture sont compris entre 0 V et 2 V . Les pulses d'initialisation et d'écriture ont une durée fixe de 100 ns .

du memristor est dans un état haut, il faut alors lui appliquer des pulses négatifs d'amplitude plus élevée en valeur absolue pour que sa résistance baisse. En effet, des pulses 1 à 42, nous initialisons la résistance du memristor à une valeur haute avec des pulses de 2 V alors que les pulses d'écriture ont une amplitude trop peu élevée pour modifier la valeur de la résistance. Nous appliquons donc plusieurs pulses de 2 V bien que la résistance soit déjà élevée. Ce phénomène est mis en évidence par la dissymétrie du comportement autour du pulse de -2 V , marqué par la flèche en rouge. La figure 2.19.b nous montre la différence de résistance obtenue en fonction de la valeur de la tension d'écriture appliquée. La dissymétrie est d'autant plus apparente sur ce graphique 2.19.b, où le chemin emprunté lorsque nous avons appliqué des tensions croissantes n'est pas le même que pour des tensions décroissantes.

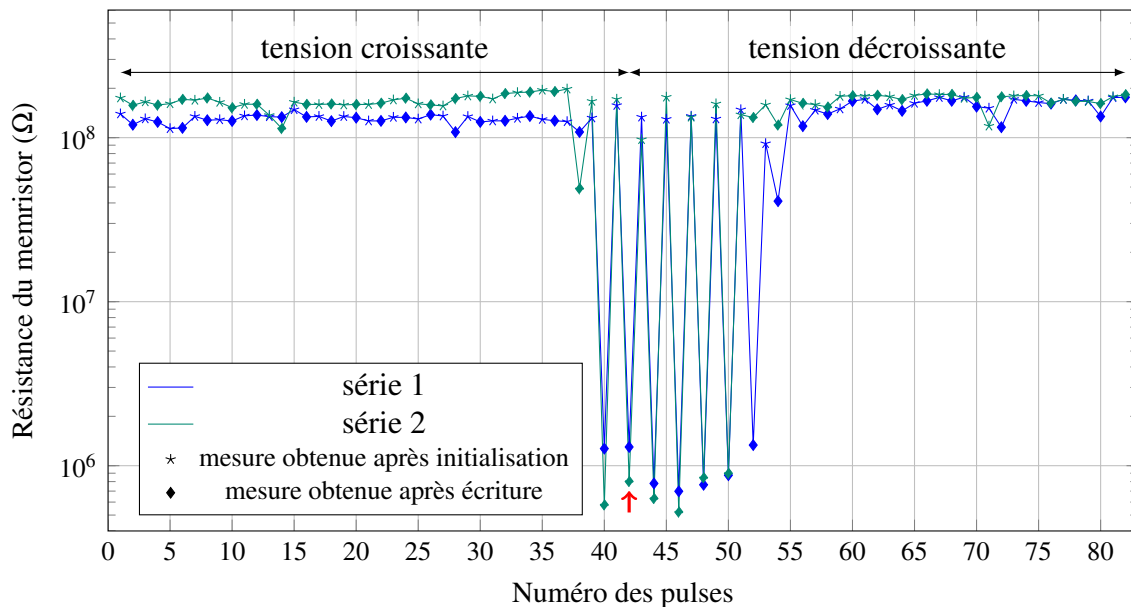
Contrairement aux mesures précédentes, celles obtenues dans le cas où l'initialisation est à R_{ON} (figure 2.20.a) correspondent à ce qui était attendu. Il y a une symétrie autour de la tension appliquée la plus élevée (2 V au pulse numéro 42), et nous avons accès à différentes valeurs de résistance. La figure 2.20.b permet de constater que les variations de résistance sont reproductibles pour une tension croissante ou décroissante.

Les mesures acquises pour ces deux cas (figures 2.19 et 2.20) nous ont servi à tracer une **STDP** fictive présentée dans la figure 2.21. Nous avons calculé la différence de conductance entre les pulses d'initialisation et les pulses d'écriture subséquents. Les valeurs en abscisse ont été obtenues en convertissant l'amplitude des pulses d'écriture en une différence de temps : plus l'amplitude est élevée, plus la différence de temps est faible. Nous avons choisi arbitrairement d'utiliser un pas de 40 ns pour la différence de temps, en commençant par 2 V (à 2 V correspond 40 ns , à $1,9\text{ V}$ correspond 80 ns , etc. et à -2 V correspond -40 ns , etc.).

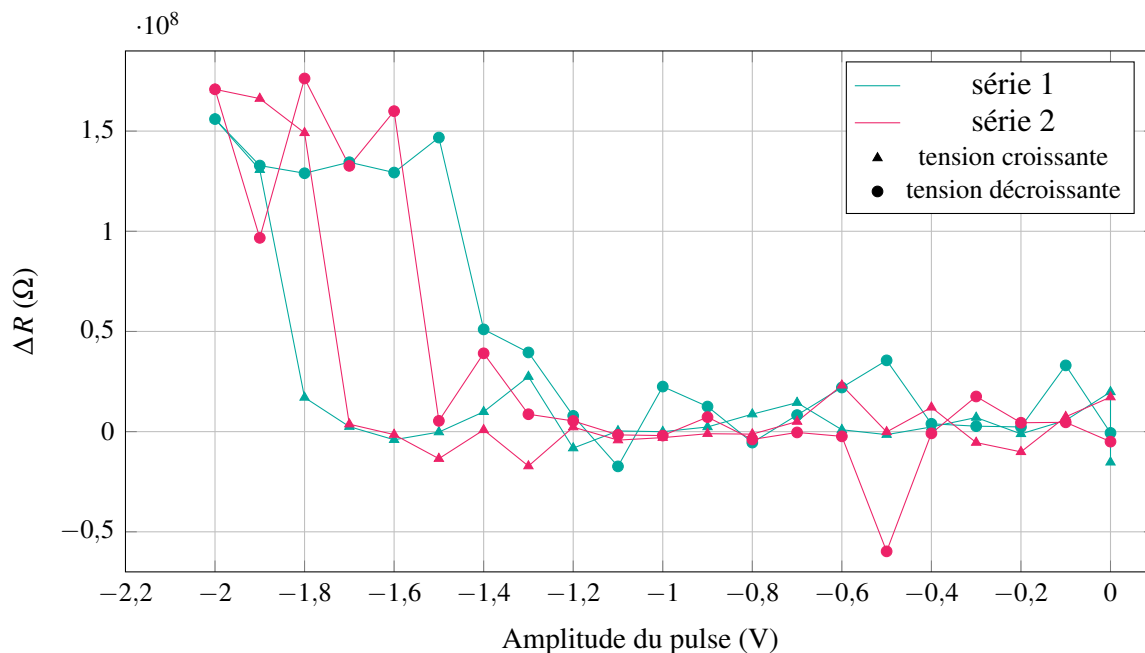
Il apparaît clairement que la **STDP** ainsi obtenue ne répond pas à nos premières attentes : bien que les variations de résistance soient du même ordre de grandeur, le phénomène de saturation empêche le composant d'avoir un comportement symétrique. La dissymétrie n'est pas seulement entre la partie négative et positive de la **STDP** mais aussi dans la partie positive elle-même où le

passage brutal entre R_{OFF} et R_{ON} ne se fait pas pour les mêmes instants.

Bien que les mesures avec des pulses ne nous donnent pas entière satisfaction à ce stade du manuscrit, nous avons expérimenté la plasticité avec des PA appliqués aux bornes du memristor.

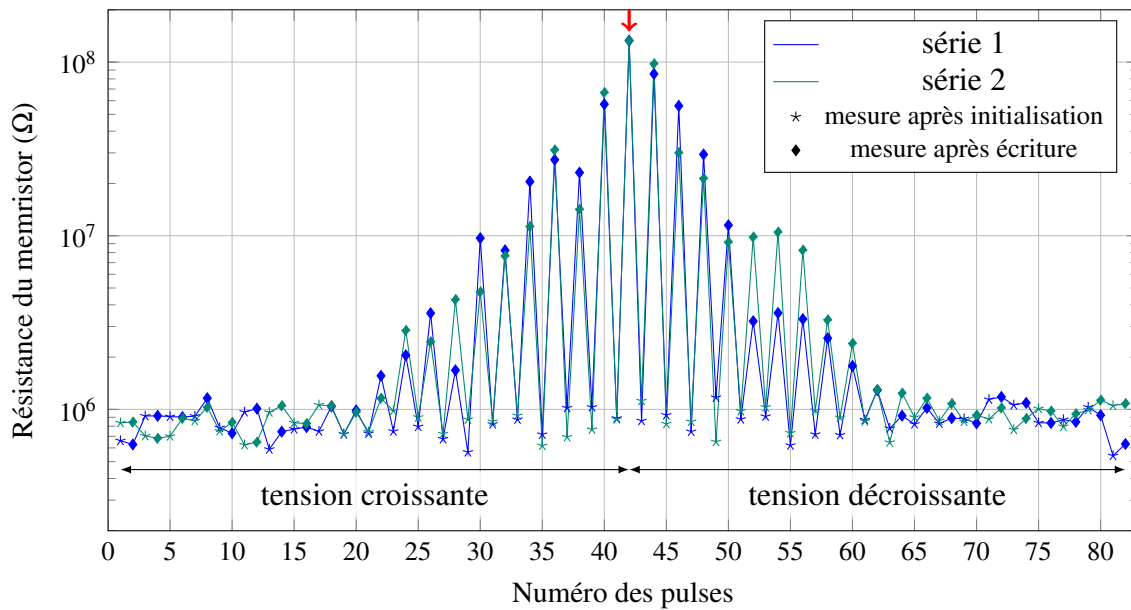


(a) Graphique montrant la variation de résistance pendant le test. La flèche rouge désigne la valeur de la résistance obtenue après le pulse d'écriture de -2 V (pulse numéro 42).

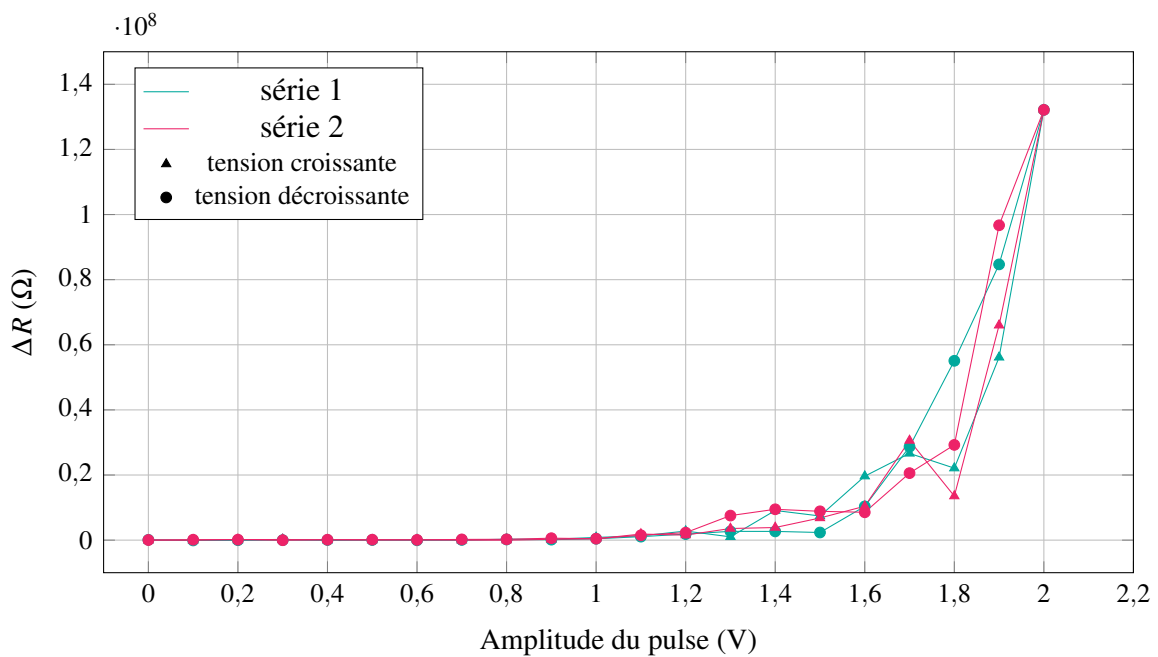


(b) Graphique montrant la différence de résistance obtenue en fonction de la valeur du pulse d'écriture appliqué.

FIGURE 2.19 – Cas n° 1 : initialisation à R_{OFF} . Ces 2 séries de mesures ont été réalisées pour un pulse positif d'écriture de durée 100 ns et avec une variation d'amplitude de 0 V à -2 V puis de -2 V à 0 V (pas de $|0,1$ V). Entre chacun des pulses d'écriture, un pulse d'initialisation (100 ns, 2 V) est envoyé.



(a) Graphique montrant la variation de résistance pendant le test. La flèche rouge désigne la valeur de la résistance obtenue après le pulse d'écriture de 2 V (pulse numéro 42).



(b) Graphique montrant la différence de résistance obtenue en fonction de la valeur du pulse d'écriture appliqué.

FIGURE 2.20 – Cas n° 2 : initialisation à R_{ON} . Ces 2 séries de mesures ont été réalisées pour un pulse positif d'écriture de durée 100 ns et avec une variation d'amplitude de 0 V à 2 V puis de 2 V à 0 V (pas de $|0,1 \text{ V}|$). Entre chacun des pulses d'écriture, un pulse d'initialisation de (100 ns, -2 V) est envoyé.

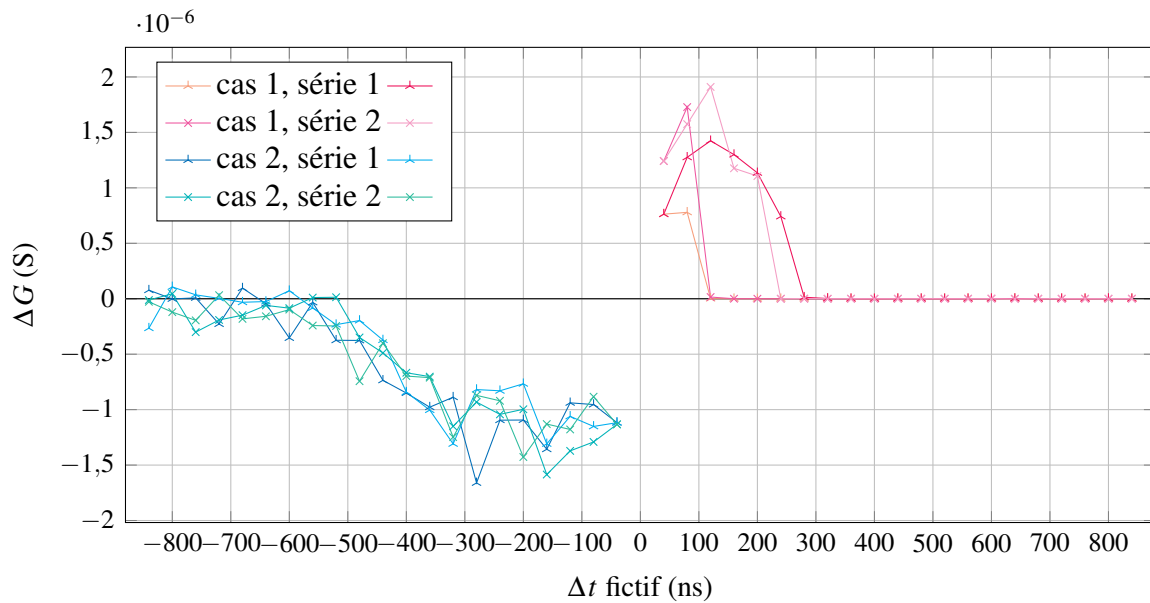


FIGURE 2.21 – **STDP** fictive obtenue avec des pulses d'une durée de 100 ns. Pour tracer ces courbes, nous avons converti l'amplitude des pulses en différence de temps. Le pas utilisé est de 40 ns pour chaque 0,1 V, nous avons commencé avec 2 V à 40 ns et - 2 V à - 40 ns.

2.3.3 Les mesures avec des potentiels d'action

2.3.3.1 La **STDP** classique

Comme nous l'avons vu au début de ce chapitre, la forme du potentiel d'action (**PA**) appliqué aux bornes du memristor engendre une forme de **STDP** particulière. Nous avons choisi d'utiliser le **PA** illustré à la figure 2.22.

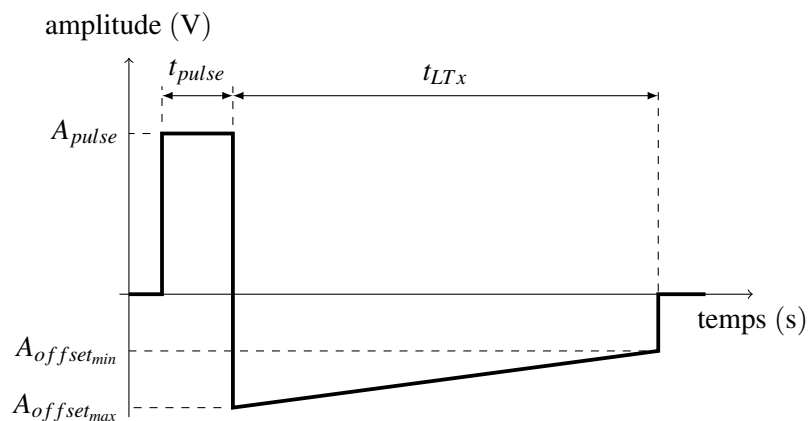


FIGURE 2.22 – Forme du potentiel d'action (ou impulsion ou *spike*).

Cette forme est une simplification d'un **PA** biologique. Elle peut être découpée en deux parties en fonction de sa polarité : la partie positive et la partie négative. La partie positive est le pulse à proprement parler, il possède une durée fixe t_{pulse} . La partie négative peut être vue comme un offset qui va être ajouté au pulse pour franchir le seuil de changement de résistance du memristor lorsque les **PA** coïncident. La durée t_{pulse} permet donc une maîtrise du temps

d'application aux bornes du memristor de la tension maximale. La tension appliquée aux bornes d'un memristor impliquant une variation (lors d'un chevauchement de deux impulsions) sera au maximum de $A_{pulse} + A_{offset_{max}}$ et au minimum de $A_{pulse} + A_{offset_{min}}$. Les valeurs des amplitudes A_{pulse} , $A_{offset_{max}}$ et $A_{offset_{min}}$ sont choisies pour être en deçà des seuils de variation du memristor. La durée de la partie négative t_{LTx} définit le temps durant lequel un pulse peut influencer la valeur de la connexion synaptique. Il s'agit des durées de **LTP** et **LTD**.

Il est possible d'utiliser la même forme de **PA** quelle que soit la position du neurone dans le réseau (en pré- ou postsynaptique). Néanmoins des ajustements de tension sont à faire puisque le memristor ferroélectrique présente une dissymétrie. Nous avons décidé d'utiliser les mêmes valeurs d'amplitude pour la partie négative du pulse en pré- et postsynaptique et d'ajuster la valeur de l'amplitude du pulse positif. Pour ces premières mesures avec des **PA**, nous les avons paramétrés avec les valeurs suivantes : $A_{pulse_{pré}} = 0,8 \text{ V}$, $A_{pulse_{post}} = 1,0 \text{ V}$, $A_{offset_{max}} = -0,8 \text{ V}$, $A_{offset_{min}} = -0,2 \text{ V}$, $t_{pulse} = 100 \text{ ns}$ et $t_{LTx} = 500 \text{ ns}$.

Les mesures de forme de **STDP** sont faites pour des différences de temps d'arrivée de **PA**. L'utilisation d'un seul générateur nous contraint à envoyer la différence des **PA** V_{mem} pour chaque Δt , avec $\Delta t = t_{post} - t_{pré}$, comme illustré à la figure 2.23. Comme précédemment, lorsque $\Delta t > 0$, il y aura une augmentation de la conductance, *i.e.* une diminution de la résistance. Les mesures sont faites avec un pas Δt de 10 ns.

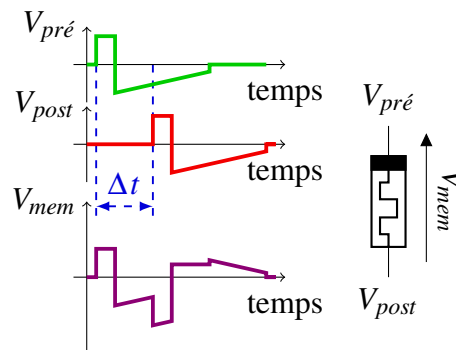
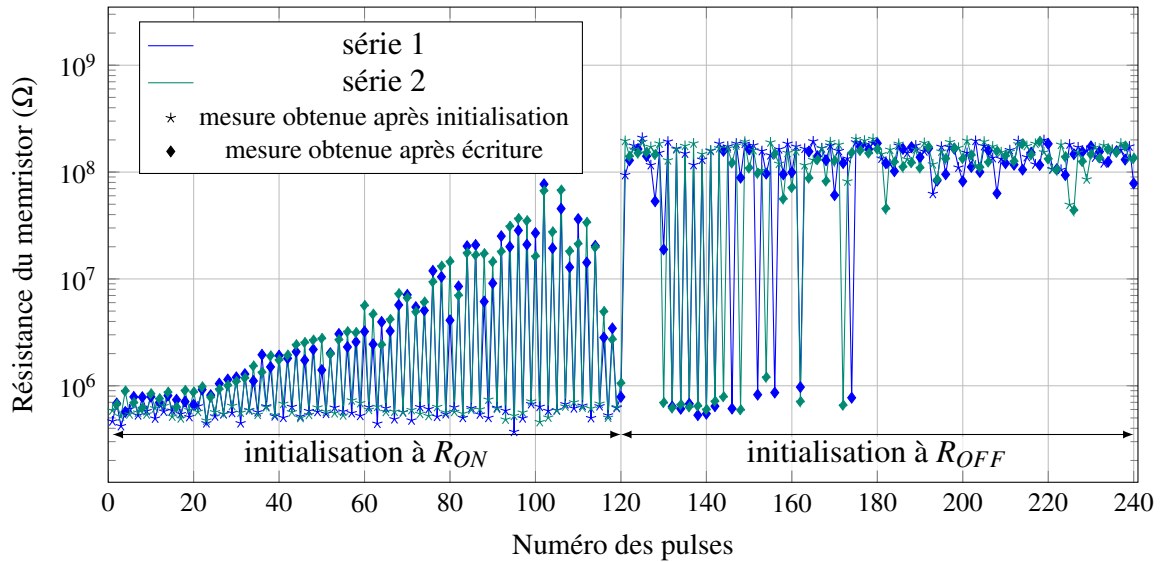


FIGURE 2.23 – Différence de potentiel V_{mem} créée à partir de la différence Δt des **PA** $V_{pré}$ et V_{post} et envoyée par le générateur aux bornes du memristor.

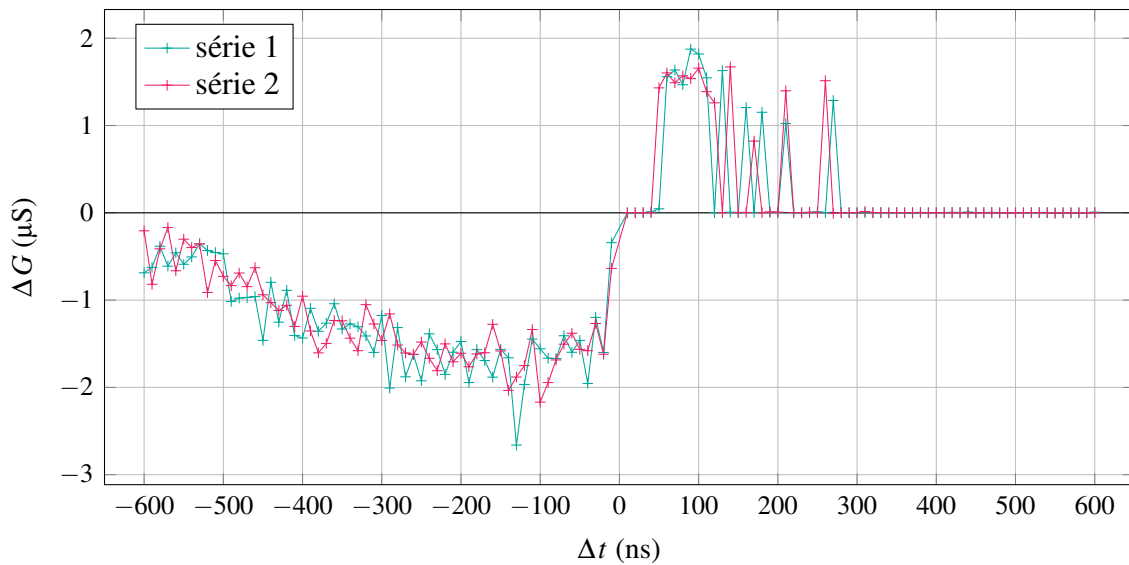
La figure 2.24 a été obtenue en initialisant le memristor avec des pulses de + 2 V pour la résistance haute et - 2 V pour la résistance basse. Afin de limiter le phénomène de saturation du memristor les mesures sont faites de la façon suivante :

1. Initialisation du memristor à une résistance basse (R_{ON}) avec un pulse de - 2 V ;
2. Application de la différence des potentiels d'action correspondant à la différence de temps d'arrivée des évènements pour $\Delta t < 0$;
3. Initialisation du memristor à une résistance haute (R_{OFF}) avec un pulse de + 2 V ;
4. Application de la différence des potentiels d'action correspondant à la différence de temps d'arrivée des évènements pour $\Delta t > 0$.

Ces étapes sont répétées jusqu'à avoir parcouru l'ensemble de la plage de différence de temps entre les potentiels d'action. Par exemple pour la figure 2.24.b, le point pour $\Delta t = -600$ ns est obtenu en premier, puis le point pour $\Delta t = 10$ ns, puis celui pour $\Delta t = -590$ ns, etc.



(a) Graphique montrant la variation de résistance du memristor pendant le test.



(b) Graphique montrant la différence de conductance en fonction de la différence des temps d'arrivée des impulsions pré- et postsynaptiques.

FIGURE 2.24 – Mesures réalisés avec des différences de potentiels d'action $\Delta t = t_{post} - t_{pré}$. Ces graphiques ont été obtenus avec les spécifications d'impulsions suivantes : $A_{pulse_{pré}} = 0,8$ V, $A_{pulse_{post}} = 1,0$ V, $A_{offset_{max}} = -0,8$ V, $A_{offset_{min}} = -0,2$ V, $t_{pulse} = 100$ ns et $t_{LTx} = 500$ ns.

Les mesures reflètent ce que nous avons obtenues avec les pulses. Elles sont satisfaisantes pour $\Delta t < 0$: il y a une évolution décroissante de la différence de conductance en fonction de la différence de temps d'apparition entre deux potentiels d'action. En revanche pour $\Delta t > 0$, l'évolution se fait de manière plus brusque. Entre $\Delta t = 100$ ns et $\Delta t = 300$ ns, le memristor semble réagir de manière aléatoire aux différences de potentiels d'action. Un tel comportement serait peut être intéressant dans la génération de processus stochastique. Il a été montré que les processus stochastiques peuvent être un avantage pour l'apprentissage (Goldberg *et al.*, 2001 ; Suri *et al.*, 2013b). Mais nous n'avons pas creusé dans cette direction car l'objet de cette recherche ne comprend pas cet aspect de l'apprentissage dans les réseaux de neurones.

Précédemment, nous nous sommes aperçus en traçant les hystérésis reflétant le comportement du memristor que la limitation de tension élargit l'accès aux résistances intermédiaires. Lorsque le memristor ne rentre pas en saturation, le passage de la résistance R_{OFF} vers R_{ON} se fait de façon moins abrupte. Nous avons alors réalisé des mesures en contrôlant les résistances avant l'application de la différence de PA. Nous n'initialisons plus avec un pulse de ± 2 V mais nous avons programmé la séquence suivante afin d'avoir la résistance du memristor comprise dans une certaine bande de résistance ($R_{control_{min}} \leq R_{memristor} \leq R_{control_{max}}$) :

1. Envoi d'un pulse négatif de -2 V : la résistance du memristor est alors proche de la résistance basse ;
2. Envoi de pulses positifs croissants (si la limite haute de la bande de résistance $R_{control_{max}}$ souhaitée n'est pas dépassée) jusqu'à franchir la limite basse $R_{control_{min}}$ de la bande de résistance souhaitée ;
3. Envoi d'un pulse négatif si la résistance dépasse la valeur limite haute $R_{control_{max}}$;
4. Envoi d'un pulse positif si la résistance ne dépasse pas la limite basse $R_{control_{min}}$.

Les points 3 et 4 sont répétés jusqu'à obtenir une résistance qui valide les contraintes basse et haute de la plage de résistance imposée. À chaque nouvelle exécution de ces derniers points, l'amplitude du pulse appliqué croît (en valeur absolue).

Le contrôle de résistance fonctionne comme il se doit. La résistance est initialisée avec les valeurs programmées. Nous avons tracé l'enveloppe de la STDP maximale présentée dans la figure 2.25. Pour les valeurs d'initialisation de $500 \text{ k}\Omega \leq R_{init} \leq 650 \text{ k}\Omega$ (en jaune sur la figure), nous avons tracé la partie LTD, et pour les valeurs $9 \text{ M}\Omega \leq R_{init} \leq 15 \text{ M}\Omega$ (en orange sur la figure) nous avons tracé la partie LTP de la STDP. Pour des valeurs initiales de résistance comprises entre ces valeurs extrêmes (en vert, bleu et violet sur la figure), nous constatons que la forme de STDP est englobée dans l'enveloppe de la STDP maximale. La variation de résistance dépend bien de la tension appliquée et de la résistance précédente.

La maîtrise des résistances de départ, nous permet d'avoir une forme de STDP continue pour les parties LTD et LTP ainsi qu'à leur transition. La zone où le composant semblait réagir de façon aléatoire a disparu. Mais cela nous oblige à n'utiliser le memristor que sur une décade

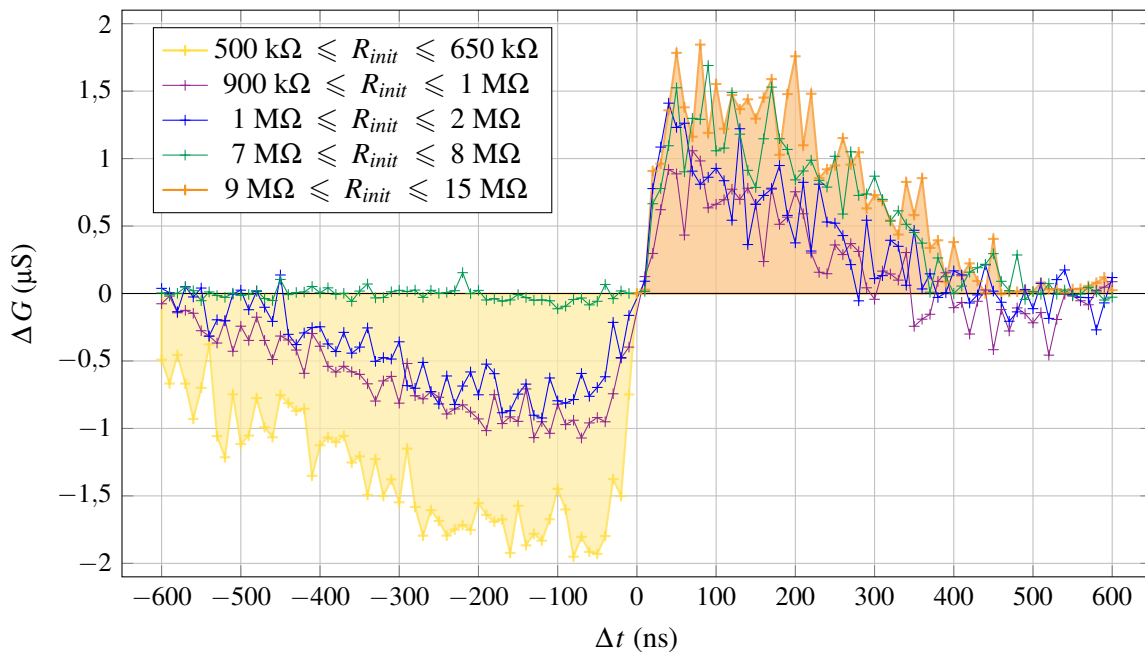


FIGURE 2.25 – Graphique montrant la différence de conductance mesurée sur le memristor en fonction de la différence des temps d'arrivée des impulsions pré- et postsynaptiques ($\Delta t = t_{post} - t_{pré}$). Les impulsions avaient les valeurs suivantes : $A_{pulse_{pré}} = 0,8$ V, $A_{pulse_{post}} = 1,0$ V, $A_{offset_{max}} = -0,8$ V, $A_{offset_{min}} = -0,2$ V, $t_{pulse} = 100$ ns et $t_{LTx} = 500$ ns. Avant chaque différence de PA, nous avons ajusté la valeur de la résistance (R_{init}).

et demi [500 k Ω ; 15 M Ω] au lieu de deux et demi [500 k Ω ; 150 M Ω]. Il est remarquable que lorsque $\Delta t = \pm 10$ ns, il y ait déjà une variation. La superposition des deux impulsions donne un pulse d'une valeur maximale de $A_{pulse} + A_{offset_{max}}$ soit + 1,6 V ou - 1,8 V et d'une durée de 10 ns. Entre 10 ns et 90 ns inclus, la variation de résistance du memristor se fait à amplitude quasi-constante mais à durée variable. Pour la partie positive de la STDP, le maximum est atteint pour des valeurs de Δt entre 50 ns et 70 ns ; alors que pour la partie négative, le minimum est atteint pour des valeurs entre 70 ns et 90 ns. Avec la forme des impulsions que nous avons choisie on pouvait s'attendre à atteindre les extremums pour les valeurs Δt de ± 100 ns.

Avec ce comportement conforme à ce qui se passe en biologie nous avons choisi d'utiliser ce type de PA pour créer notre réseau de neurones. Afin de pouvoir faire des simulations du réseau à grande échelle nous avons réalisé d'autres mesures avec le même protocole. Pour obtenir les performances d'apprentissage du réseau en fonction des variations du comportement des memristors, nous avons procédé à plusieurs mesures sur différents memristors. Les tracés de la figure 2.26 donnent les courbes moyennes de STDP par composant et leurs variations mesurées par Δt . Le graphique montre que les mesures sont reproductibles mais la disparité reste importante. Toutefois ces résultats donnent une première approximation de la variabilité du comportement des memristors et seront exploités dans le chapitre suivant afin de réaliser des simulations d'un réseau utilisant des synapses memristives.

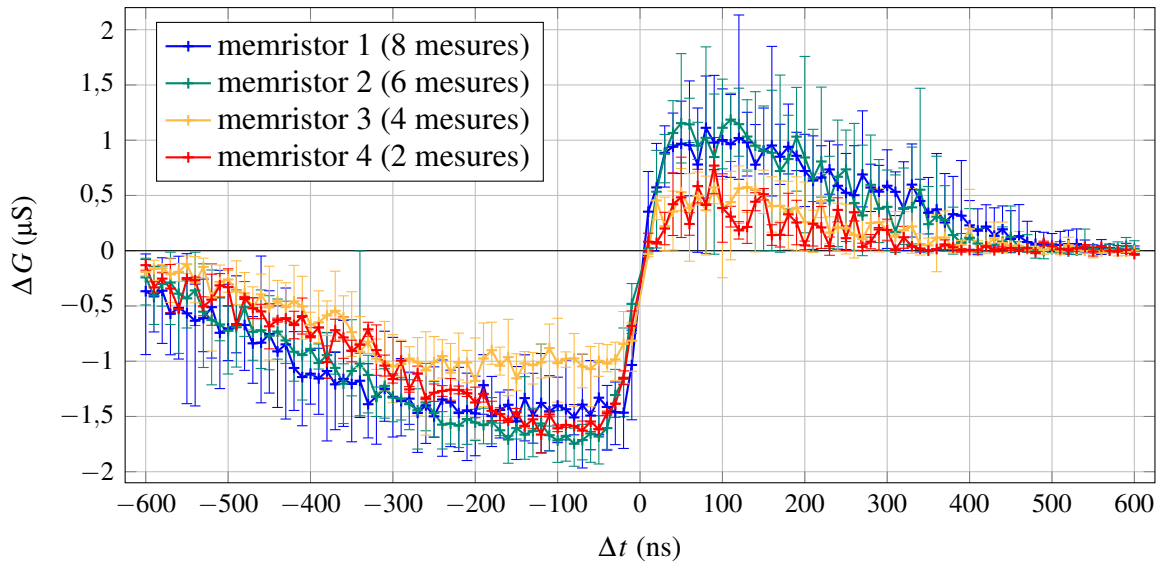


FIGURE 2.26 – Moyenne par Δt des valeurs ΔG obtenues pour plusieurs mesures et pour différents composants montrant la variabilité du comportement des memristors **BFO** dans l'obtention de la forme de **STDP** classique. Les impulsions avaient les valeurs suivantes : $A_{pulse_{pré}} = 0,8 \text{ V}$, $A_{pulse_{post}} = 1,0 \text{ V}$, $A_{offset_{max}} = -0,8 \text{ V}$, $A_{offset_{min}} = -0,2 \text{ V}$, $t_{pulse} = 100 \text{ ns}$ et $t_{LTx} = 500 \text{ ns}$. Avant chaque différence de **PA**, nous avons ajusté la valeur de la résistance avec les conditions suivantes : $500 \text{ k}\Omega \leq R_{ON} \leq 650 \text{ k}\Omega$ et $9 \text{ M}\Omega \leq R_{OFF} \leq 15 \text{ M}\Omega$.

2.3.3.2 Une autre forme de **STDP**

En modifiant la forme du **PA** appliqué aux bornes du memristor, nous avons tracé d'autres formes de **STDP**. La figure 2.27 montre une forme de **STDP** non-biologique mais qui pourrait être utilisée dans des réseaux de neurones impulsionnels computationnels. Elle s'apparente à la forme simplifiée de **STDP** utilisée dans la publication de [Bichler \(2012\)](#). Comme précédemment, la forme des **PA** est identique en pré- et postsynaptique, et les valeurs des amplitudes positives sont différentes. D'autres formes sont présentées dans l'annexe [A](#).

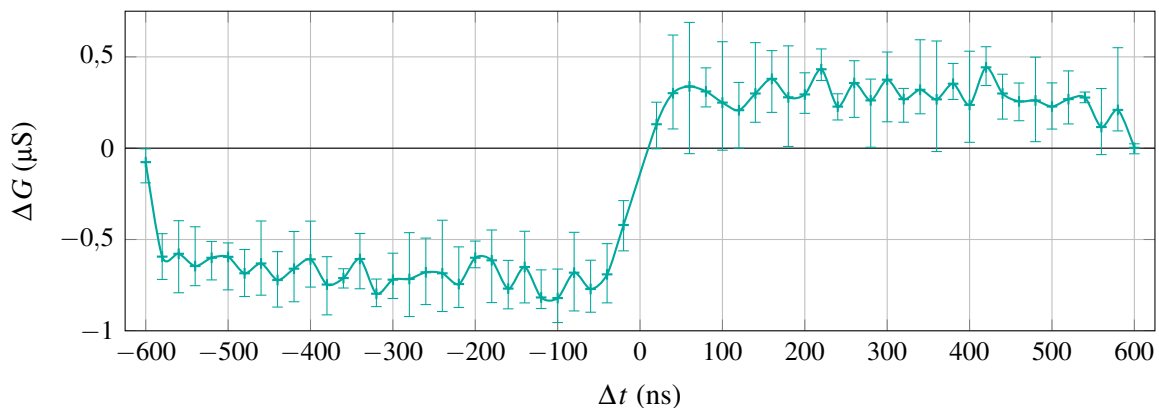


FIGURE 2.27 – Moyenne par Δt des valeurs ΔG obtenues pour 4 mesures montrant la variabilité dans l'obtention de forme de **STDP** non-biologique avec un memristor **BFO**. Les impulsions avaient les valeurs suivantes : $A_{pulse_{pré}} = 0,8 \text{ V}$, $A_{pulse_{post}} = 1,0 \text{ V}$, $A_{offset_{max}} = A_{offset_{min}} = -0,5 \text{ V}$, $t_{pulse} = 100 \text{ ns}$ et $t_{LTx} = 500 \text{ ns}$. Avant chaque différence de **PA**, nous avons ajusté la valeur de la résistance avec les conditions suivantes : $1 \text{ M}\Omega \leq R_{ON} \leq 2 \text{ M}\Omega$ et $9 \text{ M}\Omega \leq R_{OFF} \leq 10 \text{ M}\Omega$.

2.4 Conclusion

L'utilisation de composants memristifs en tant que synapses peut apporter des avancées majeures en termes de surface et de consommation dans les réseaux de neurones en général : le memristor peut être réduit à un plot d'une dizaine de nanomètres de rayon et de hauteur et ses propriétés en font une mémoire passive.

En combinant la plasticité intrinsèque du memristor et la forme des potentiels d'action appliqués entre ses bornes, il est possible de reproduire des formes classiques de *Spike-Timing-Depend Plasticity*. Le memristor ne devient plus seulement une mémoire mais il peut remplacer les algorithmes de plasticité biologique. La limitation de son utilisation provient alors des amplitudes et des durées qu'il est nécessaire de lui appliquer pour obtenir ces changements.

Les mesures de **STDP** issues de la littérature et présentées ici ont été réalisées avec différentes technologies et ont comme dénominateur commun l'imitation du vivant. Les temps, de l'ordre de la milliseconde, sont comparables aux temps biologiques. De plus, certaines combinaisons de matériaux ne permettent pas d'avoir des caractéristiques électriques compatibles avec les technologies CMOS classiques. Dans le cadre de ces travaux de recherches, nous utilisons une technologie de memristor ferroélectrique qui permet de travailler avec des pulses de l'ordre du volt et de la dizaine de nanosecondes. La rapidité de ces changements en font un candidat possible pour créer des co-processeurs à base de neurones impulsionnels.

Les mesures réalisées avec le memristor **BFO** montrent qu'il n'est pas nécessaire d'appliquer des valeurs de tension supérieures à $|2 \text{ V}|$, ce qui est compatible avec la microélectronique. Cependant ces tensions limites amènent le memristor en état de saturation lorsque sa valeur de résistance est proche de la valeur haute (R_{OFF}). Ce phénomène ne permet pas d'obtenir des formes de **STDP** correctes. Nous avons alors proposé une solution en bornant les états initiaux R_{ON} et R_{OFF} . De plus, en modifiant la forme des **PA** appliqués aux bornes du memristor, nous avons obtenu une forme de **STDP** non-biologique.

Le memristor **BFO** utilisé ici peut être qualifié de composant bioinspiré. Il remplit deux des trois fonctions de la synapse : la mémoire et la plasticité. Nous présenterons au chapitre 4 la manière dont nous avons choisi d'inclure le memristor au sein d'un réseau de neurones impulsionnels pour remplir la troisième fonction d'une synapse : la pondération.

Mais auparavant, en s'appuyant sur les mesures réalisées dans ce chapitre, nous allons étudier les performances d'un réseau utilisant le memristor en tant que synapse.

CHAPITRE 3

LES RÉSEAUX DE NEURONES IMPULSIONNELS : DE LA MODÉLISATION À LA SIMULATION

LES réseaux de neurones, dont les propriétés globales sont dues aux interactions locales, s'inspirent du cerveau et en reprennent trois grandes propriétés : l'apprentissage, la robustesse et le parallélisme. L'apprentissage permet aux réseaux de s'adapter grâce à la plasticité synaptique. Il est possible dans certains cas d'opérer des reconversions dans leurs fonctionnements. Le bruit en entrée ou la panne de certains éléments ne mettent pas le réseau en défaut et en font un système robuste car l'information y est distribuée. Le parallélisme autorise une simultanéité dans le traitement des informations par les réseaux.

Comme l'a exposé Intel ([Dubey, 2005](#)), les réseaux de neurones sont performants dans les domaines de reconnaissance, d'exploration et de synthèse (ou en anglais les domaines *Recognition, Mining and Synthesis*, RMS). Ils permettent d'avoir un gain en temps de calculs et en consommation.

Par exemple, pour la reconnaissance d'images, [Pérez-Carrasco et al. \(2010\)](#) ont montré que les réseaux de neurones sont plus rapides qu'un système séquentiel. Pour ce faire, ils utilisent une caméra qui code les variations de luminosité des pixels de l'image en évènements. Au moment de la présentation de l'image, la caméra génère des évènements qui sont envoyés sur la première couche du réseau de neurones engendrant à son tour des évènements sur la couche suivante, jusqu'à la couche de sortie où la reconnaissance de l'objet est donnée en moins de 1 ms. En utilisant une caméra classique, l'image, qui est envoyée dans son entier à intervalle régulier, est traitée dans son ensemble par un processus séquentiel. Elle est classée après avoir subi les différents traitements de chacun des blocs de calculs. S'il faut six étapes de traitement et que chacune d'entre elle nécessite 1 ms, alors l'image est reconnue au bout de 6 ms (figure 3.1).

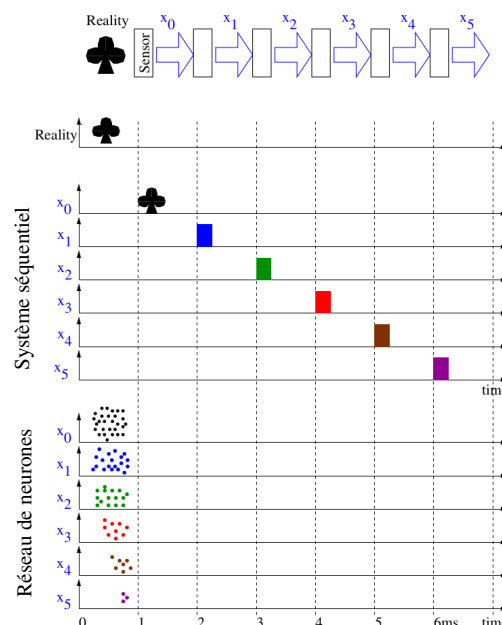


FIGURE 3.1 – Différence des temps d'exécution dans la reconnaissance d'images entre un système séquentiel et un réseau de neurone. Illustration provenant de [Pérez-Carrasco et al. \(2013\)](#).

Dans ce chapitre, nous allons nous intéresser aux architectures de réseaux de neurones et à leurs apprentissages. Nous nous concentrerons sur la modélisation des réseaux qui sont employés

pour la classification. En effet, dans le projet [MHANN](#) nous souhaitons construire un réseau de neurones sur silicium utilisant des synapses memristives et servant à la classification d'images. Ensuite, nous détaillerons la modélisation des lois d'apprentissage utilisées. Puis nous finirons par les simulations du réseau de neurones du projet [MHANN](#).

3.1 La modélisation des réseaux de neurones

Nous allons nous intéresser aux réseaux « plastiques » qui peuvent apprendre à partir d'échantillons appliqués à leurs entrées. À chaque échantillon appliqué en entrée correspond un état des neurones de sortie qui peut être changé en modifiant le poids des synapses qui sont plastiques. Il existe des réseaux de neurones où le poids des synapses est fixe. C'est notamment le cas des réseaux de neurones permettant de générer une rythmique. Ils sont appelés réseaux locomoteurs spinaux, ou *Central Pattern Generator* (CPG), et servent de commande pour le cœur ou la marche d'un individu par exemple.

Nous allons décrire différents types de réseaux de neurones plastiques et leurs lois d'apprentissage. Ces réseaux se sont éloignés des aspects biologiques bien qu'ils en soient inspirés. Ils ont en commun d'être utilisés à des fins de classification.

3.1.1 Des exemples d'architectures de réseaux dédiées à la classification

Chacune des couches des réseaux de neurones définit un espace propre. Les poids des connexions synaptiques permettent de modifier le plan dans lequel sont projetés les échantillons d'entrée. Une rectification des poids est réalisée lorsque la sortie obtenue diffère de la sortie souhaitée. Il existe alors des règles qui permettent de faire varier les poids synaptiques. Ces lois d'apprentissage dépendent à la fois de l'architecture du réseau et du type de neurone qui le compose. Elles sont classées suivant deux catégories :

- les lois supervisées (figure [3.2.a](#)) qui contrôlent la plasticité globale du réseau et nécessitent un gabarit (ou de connaître la sortie désirée) pour calculer l'erreur entre la sortie obtenue et la sortie désirée ;
- les lois non-supervisées (figure [3.2.b](#)) qui agissent au niveau local (entre deux neurones) et n'ont besoin d'aucun système de contrôle.

L'architecture des réseaux ainsi que le type de neurone dépendent de l'application dans laquelle ils vont être utilisés. La figure [3.3](#) donne un aperçu de quatre topologies représentatives détaillées ci-dessous.

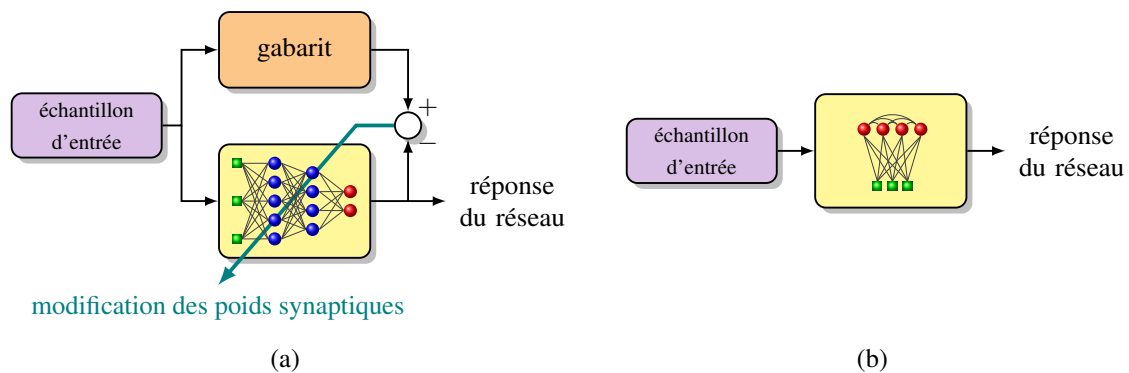


FIGURE 3.2 – Différentes méthodes d'apprentissage. (a) Apprentissage supervisé utilisant un gabarit et une loi globale de modification des poids synaptiques. (b) Apprentissage non-supervisé utilisant une loi locale de modification des poids synaptiques.

Le réseau de neurones multicouches *feedforward* (figure 3.3.a) lorsqu'il est composé de neurones artificiels (non-impulsionnels) est aussi appelé perceptron multicouches (Werbos, 1974 ; Parker, 1982 ; Le Cun, 1985 ; Rumelhart et Zipser, 1986). Il utilise une ou plusieurs couches intermédiaires (ou cachées) entre l'entrée et la couche de sortie. Les couches peuvent être entièrement connectées entre elles. Dans ce cas, le réseau est désigné par le terme réseau *all-to-all*. Les couches cachées permettent de modifier l'espace de travail des entrées. Pour son apprentissage, le perceptron multicouches utilise la règle supervisée de rétro-propagation du gradient basée sur l'algorithme de Widrow-Hoff (Werbos, 1974). Alors que les informations à apprendre sont traitées par le réseau en sens unique de l'entrée vers la sortie, la modification des poids, qui est fonction d'une référence en sortie, traverse le réseau en sens inverse. Les perceptrons multicouches sont utilisés, en outre, pour des estimations ou identifications de fonctions non linéaires et pour de la reconnaissance d'images où ils obtiennent les meilleurs résultats (Cireşan *et al.*, 2012).

La figure 3.3.b présente un réseau de neurones entièrement récurrent. Les réseaux de neurones récurrents, contrairement aux perceptrons, possèdent au moins une connexion entre neurones de la même couche ou neurones de la couche précédente. Ils emploient aussi la règle de rétro-propagation du gradient pour leur apprentissage et sont utilisés pour la reconnaissance d'images ou de l'identification de fonctions. Plusieurs architectures de réseaux de neurones récurrents ont été étudiées et possèdent le nom des personnes qui s'y sont intéressées (Hopfield, 1982 ; Elman, 1990 ; Jordan, 1986).

Les réseaux auto-adaptatifs ou cartes auto-organisatrices ou encore réseaux de Kohonen (figure 3.3.c) ne possèdent pas à proprement parler de couche de sortie. Chacun des neurones est connecté avec tous ses voisins. Ce sont les connexions entre les neurones qui permettent de définir la position relative de chacun des neurones et ainsi de former une carte. Son apprentissage est non-supervisé et dépend de la loi de Grossberg : les poids qui entrent sur le neurone gagnant sont augmentés, s'ils sont trop faibles, pour les rapprocher du vecteur d'entrée. Ils sont utilisés pour la classification.

Les réseaux de neurones compétitifs ou *Winner-Take-All (WTA)* (figure 3.3.d) possèdent une couche de sortie de neurones qui sont connectés les uns aux autres. Les connexions sont inhibitrices entre les neurones et peuvent être réalisées avec l'ensemble des neurones de la couche ou bien uniquement avec les neurones proches. Elles permettent de s'assurer que seul un neurone de l'ensemble connecté est actif. L'apprentissage du réseau est non-supervisé. Ils sont utilisés pour la classification.

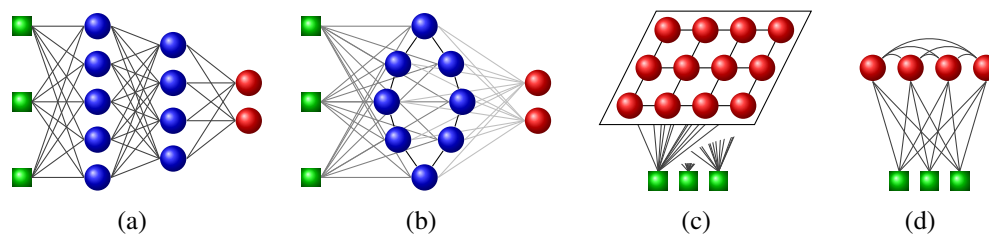


FIGURE 3.3 – Différentes architectures de réseaux de neurones. Les carrés modélisent les entrées et les ronds les neurones du réseau. (a) Réseau multicouche *feedforward*. (b) Réseau entièrement récurrent. (c) Carte auto-organisatrice. (d) Réseau compétitif.

De nombreuses études ont été menées sur les réseaux de neurones dans les domaines de la reconnaissance d'images et de la classification. Nous allons nous intéresser aux réseaux de neurones incluant des composants memristifs ou encore appelés hybrides. Des systèmes réels ont vu le jour alors que d'autres sont pour le moment issus de simulations.

3.1.2 Les réseaux de neurones hybrides dans la littérature

Les réseaux de neurones hybrides désignent les réseaux alliant différentes technologies et notamment ceux utilisant des dispositifs memristifs. Avec l'émergence de ces nouvelles technologies, qui ont la capacité de garder en mémoire la valeur de leur résistance sans aucune consommation supplémentaire, il devient intéressant de les intégrer dans les réseaux de neurones en tant que synapses.

De plus, au chapitre précédent, il a été montré que la plasticité intrinsèque des memristors permet d'implémenter des formes de *STDP* similaires à celles observées en biologie. C'est alors la forme des *PA* appliqués aux bornes du composant qui permet de contrôler la forme résultante de *STDP*. S'agissant d'impulsion à générer et d'une loi temporelle, il devient judicieux d'utiliser des neurones impulsionnels pour construire le réseau.

Des recherches ont montré qu'il était possible d'utiliser des lois de *STDP* pour la classification. Masquelier *et al.* (2009) ont utilisé des neurones impulsionnels et une règle de *STDP* pour simuler l'apprentissage de différentes entrées. Pour ce faire, ils ont choisi une architecture compétitive de réseau de neurones. Les neurones de la couche de sortie sont connectés entre eux par des synapses inhibitrices (Thorpe, 1990).

C'est l'architecture *WTA* que Bichler *et al.* (2012), Suri *et al.* (2013a) et Suri *et al.* (2013b) ont choisie pour simuler leurs réseaux de neurones à synapses memristives. Leurs simulations

ont utilisé le même support : le simulateur Xnet de [Bichler et al. \(2012\)](#). Leurs réseaux sont identiques et possèdent une couche de 16384 entrées et deux couches composées respectivement de 60 et 10 neurones. Ces deux couches de neurones possèdent une inhibition latérale entre chacun des neurones de la couche. Ils utilisent une base de données provenant d'une caméra à événements et discriminent avec leur réseau de neurones les six voies (en sens unique) d'une autoroute californienne. Reprises dans la thèse de [Bichler \(2012\)](#), les simulations concernent trois différentes technologies de mémoire : la NOMFET (cf. 2.2.3), la PCM (cf. 2.2.1) et la CBRAM (c'est une mémoire à électrolytes solides utilisable pour de la programmation binaire).

Sur ce même simulateur (Xnet), [Querlioz et al. \(2013\)](#) ont montré que les réseaux de neurones peuvent s'adapter aux défauts des composants memristifs. Ils ont utilisé une base de données de chiffres écrits à la main (la base *Mixed National Institute of Standards and Technology*, MNIST, que nous décrirons plus en détail dans la partie 3.3). En utilisant un apprentissage non-supervisé et avec une variabilité de 50 % des paramètres des memristors, le réseau peut encore avoir un taux de reconnaissance avoisinant les 80 %.

Il est à noter que les simulations réalisées avec Xnet utilisent une forme de STDP simplifiée avec notamment une dépréciation systématique, facilement réalisable dans un code, des synapses qui n'ont pas participé à l'activation du neurone.

Quant à [Yu et al. \(2012\)](#), ils utilisent des RRAM (cf. 2.2.2) et simulent l'apprentissage de formes provenant d'une base de données personnalisées. L'architecture utilisée est un WTA. Ils ne quantifient pas leur résultat mais ils montrent qu'ils sont capables de faire apprendre différentes formes à leur réseau.

Les résultats obtenus en simulation sont encourageants et démontrent qu'il est possible d'utiliser des dispositifs memristifs au sein des réseaux de neurones. Outre les systèmes simulés, des réseaux ont été construits comme par exemple à l'université de Carlton (Canada) où [Crupi et al. \(2012\)](#) ont développé un petit réseau avec 9 memristors utilisés comme des synapses. Ils parviennent à faire apprendre au réseau mais la technologie utilisée semble avoir une très faible rétention de l'information (de l'ordre d'une quarantaine de secondes).

En utilisant des RRAM (cf. 2.2.2), une technologie plus mature de memristors, [Park et al. \(2012\)](#) parviennent à implémenter un réseau de 10 neurones pour apprendre les 10 chiffres (0 à 9) de dimension 5×6 pixels qu'ils ont eux-mêmes créés. Avec un *crossbar* de 1000 points mémoires qu'ils n'utilisent que partiellement, ils réalisent un apprentissage de leurs chiffres non bruités et arrivent à les classer.

Les simulations et ces premiers dispositifs réels ouvrent la voie à l'intégration de memristors au sein des réseaux de neurones. Le projet MHANN ambitionne de créer un réseau avec des memristors à effet purement électronique.

3.1.3 Le réseau de neurones du projet MHANN

Le réseau de neurones du projet **MHANN** souhaite utiliser les memristors ferroélectriques **BFO** dont les mesures de **STDP** avec des **PA** ont été présentées dans le paragraphe 2.3.3. Pour ce faire, nous avons choisi d'utiliser des neurones impulsionnels *Leaky Integrate & Fire* (**LIF**) et une architecture de réseau compétitive qui permet l'apprentissage de différentes formes telles que des chiffres ou des lettres. Alors que pour l'implémentation matérielle le réseau est dimensionné pour apprendre des images de 81 pixels et de 10 classes différentes, en simulation, il est possible d'augmenter le nombre d'entrées et de sorties pour tester l'apprentissage de formes plus complexes sans changer l'architecture.

Le réseau de neurones du projet **MHANN** dont le schéma de l'implémentation matérielle est présentée à la figure 3.4, est formé d'une couche d'entrée où les éléments sont appelés capteurs, et d'une couche de sortie composée de neurones. Les capteurs traduisent l'information binaire arrivant sur leurs entrées en impulsions. Les neurones, quant à eux, intègrent les informations reçues sur leurs membranes et génèrent une impulsion dès que le seuil de leur membrane est franchi. Les deux couches sont reliées en *all-to-all* avec des synapses memristives excitatrices.

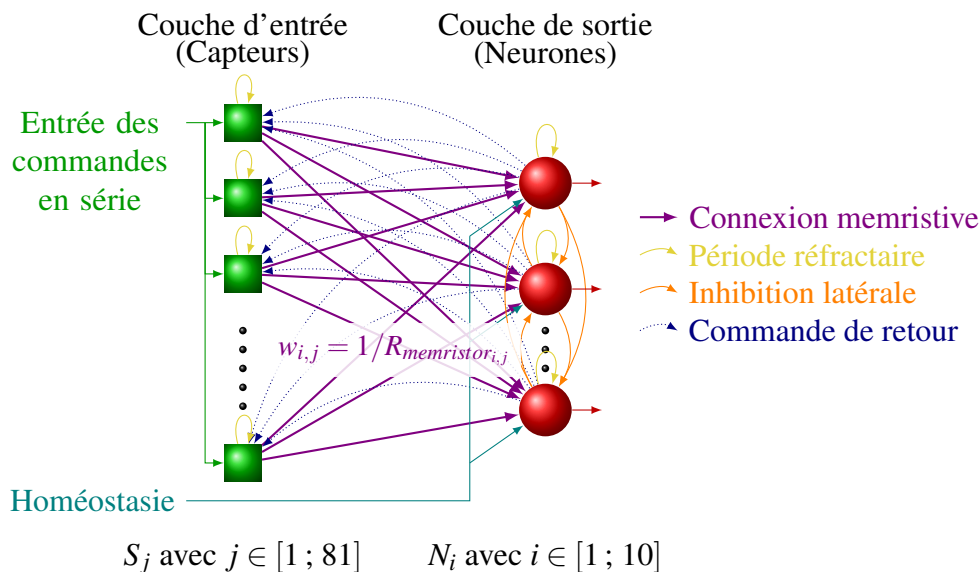


FIGURE 3.4 – Schéma de l'implémentation matérielle du réseau de neurones impulsionnels à synapses memristives.

L'architecture **WTA** impose une inhibition des neurones de la couche de sortie entre eux. Si l'architecture est utilisée pour apprendre différentes formes sans aucune autre propriété, alors il n'est pas possible que l'apprentissage donne satisfaction. En effet, lorsque des **PA** sont générés par les échantillons en entrée, les événements chargent les capacités de membrane des neurones jusqu'à l'activation de l'un d'entre eux. Les synapses se trouvant entre le neurone actif et les capteurs encore actifs (ayant donc permis au neurone de s'activer) sont alors renforcées suivant la règle de **STDP**. Au fur et à mesure que les échantillons s'échelonnent, il y a un renforcement

systématique des poids synaptiques entre le neurone qui devient actif et les capteurs encore actifs. Seul le bruit ajouté aux évènements d'entrée peut engendrer une dépréciation des poids.

Pour éviter que les poids synaptiques soient toujours augmentés sans jamais être dépréciés, le simulateur Xnet détourne la règle de **STDP** en diminuant systématiquement le poids des synapses qui n'ont joué aucun rôle dans l'activation du neurone. Mais la règle de **STDP** est bornée et impose d'avoir deux évènements un pré- et un postsynaptique pour qu'il y ait une variation de poids. De plus, avec une implémentation matérielle où le memristor joue le rôle de la synapse et se trouve entre un capteur et un neurone, la seule possibilité de faire varier le poids est qu'il y ait une différence de **PA**.

C'est pourquoi dans notre réseau, en plus des neurones, les capteurs sont eux aussi dotés d'une période réfractaire. Lorsqu'un neurone ou un capteur est en période réfractaire, qui intervient après la génération d'une impulsion, ils ne peuvent pas générer d'impulsion : ils sont inertes. Dans les réseaux de neurones artificiels, il est remarquable que la couche d'entrée soit pourvue de cette propriété. Combinée à la commande de retour provenant des neurones, elle permet de faire générer des **PA** par les capteurs inactifs suite à l'activation d'un neurone. En effet, les capteurs déjà actifs ou en période réfractaire ne peuvent pas générer à nouveau une impulsion. Des **PA** présynaptiques sont alors produits après un **PA** postsynaptique : les valeurs des poids des synapses entre le neurone actif et les capteurs nouvellement actifs sont alors dépréciées. Cette séquence des impulsions présente lors de l'apprentissage du réseau est illustrée à la figure 3.5.

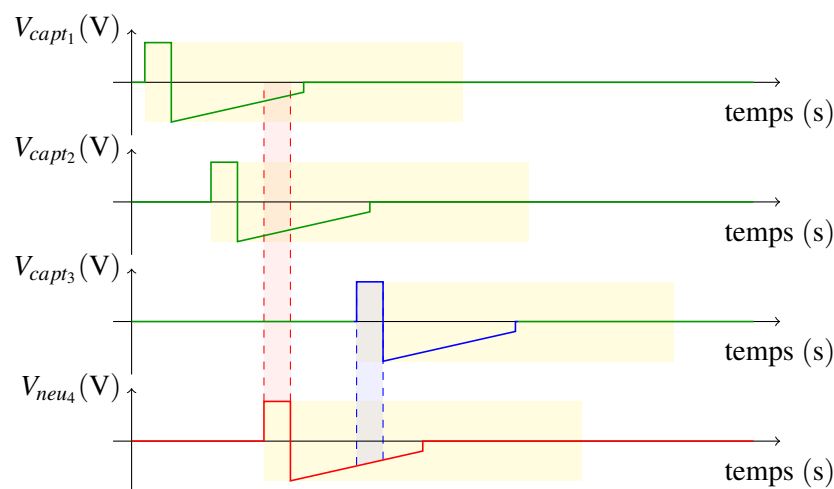


FIGURE 3.5 – Schéma de séquence d'impulsions pendant l'apprentissage. Les bandes jaunes délimitent les zones où les éléments ne peuvent pas être activés. Les impulsions V_{capt1} et V_{capt2} sont générées suite à une commande extérieure. Elles engendrent l'activation du neurone 4 (V_{neu4}). La bande rouge délimite la zone de potentialisation (**PA** pré- puis postsynaptique). Le neurone actif envoie une commande vers la couche d'entrée et permet ainsi la génération d'impulsions par les capteurs qui ne sont pas actifs ou en période réfractaire (V_{capt3}). La bande bleue délimite la zone de dépréciation (**PA** post- puis présynaptique).

L'apprentissage du réseau de neurones que nous utilisons est non-supervisé mais il possède une propriété globale pour que celui-ci se fasse de manière homogène : l'homéostasie. L'homéostasie agit sur les tensions de seuil des neurones. Ainsi lorsqu'un neurone devient actif trop souvent (critère défini dans l'algorithme de l'homéostasie), il voit sa tension de seuil augmenter

pour que d'autres neurones puissent s'activer et donc apprendre. Il est à noter que l'homéostasie peut permettre de superviser l'apprentissage : en modifiant les seuils des neurones en fonction des entrées, l'apprentissage d'un échantillon d'entrée est favorisé pour un neurone choisi. Un exemple de ce type d'apprentissage est présenté dans l'annexe C.

Une propriété intéressante des réseaux **WTA** simple couche est que les poids synaptiques connectés à chacun des neurones représentent les formes apprises par ceux-ci. Il est donc possible d'avoir un aperçu visuel de l'apprentissage.

L'apprentissage du réseau ne peut avoir lieu sans la plasticité des synapses. Dans la suite de ce chapitre nous allons donner différentes modélisations possibles de la plasticité synaptique.

3.2 La modélisation de la plasticité synaptique

Afin de tester l'architecture de notre réseau et de connaître l'influence de ses paramètres, nous avons utilisé différentes modélisations de **STDP**. Comme nous l'avons vu au chapitre 2, il existe différentes technologies de memristors. Dans la majorité de ces technologies, la physique n'est pas entièrement comprise, il n'est donc pas possible d'avoir à ce jour un modèle de memristor fiable basé sur la physique.

Pour présenter les différentes formes de **STDP**, nous avons choisi d'utiliser comme convention $\Delta t = t_{post} - t_{pré}$, avec $t_{pré}$ et t_{post} les instants où commencent les évènements pré- et postsynaptiques respectivement. Suivant la loi de **STDP** « classique » et les conventions ci-dessus, pour $0 < \Delta t \leq t_{LTP}$, il y a potentialisation du poids synaptique que nous désignerons par les termes anglais *Long Term Potentiation* (**LTP**) et pour $-t_{LTD} < \Delta t \leq 0$, il y a dépréciation du poids synaptique que nous désignerons par *Long Term Depression* (**LTD**).

Pour la simulation, nous avons choisi d'avoir un poids synaptique sans dimension qui peut varier de $w_{min} = 1$ à $w_{max} = 10000$.

Après avoir présenté une forme de **STDP** utilisée dans la littérature, nous donnerons un modèle de forme de **STDP** basé sur les mesures que nous avons réalisées sur le memristor ferroélectrique **BFO** de nos partenaires de l'UMφ CNRS/Thalès.

3.2.1 Un exemple de modélisation d'une loi de variation de conductance de memristor à oxydoréduction

Le premier modèle simplifié que nous présentons vient de **Bichler (2012)**. Ils reprennent une modélisation provenant de **Querlioz et al. (2011)** du memristor à oxydoréduction de l'université du Michigan (**Jo et al., 2010**). Ils ont choisi d'utiliser une **STDP** carrée où la différence de temps des évènements donne uniquement le signe du changement de la variation des poids synaptiques.

Cette dernière n'est donc dépendante que du poids précédent, et suit l'équation :

$$F(w) = \begin{cases} \alpha_- \exp\left(-\beta_- \frac{w_{max} - w}{w_{max} - w_{min}}\right) & \text{si } \Delta t < 0 \\ \alpha_+ \exp\left(-\beta_+ \frac{w - w_{min}}{w_{max} - w_{min}}\right) & \text{si } \Delta t \geq 0 \end{cases} \quad (3.1)$$

La figure 3.6 présente un exemple de variation Δw du poids en fonction du poids précédent w (figure 3.6.a). L'évolution du poids suit une fonction exponentielle. Le maximum de la variation de w de la LTP définit le maximum de la plage de variation de Δw en fonction de Δt (figure 3.6.b). Il est en de même pour le minimum de la variation de w de la LTD.

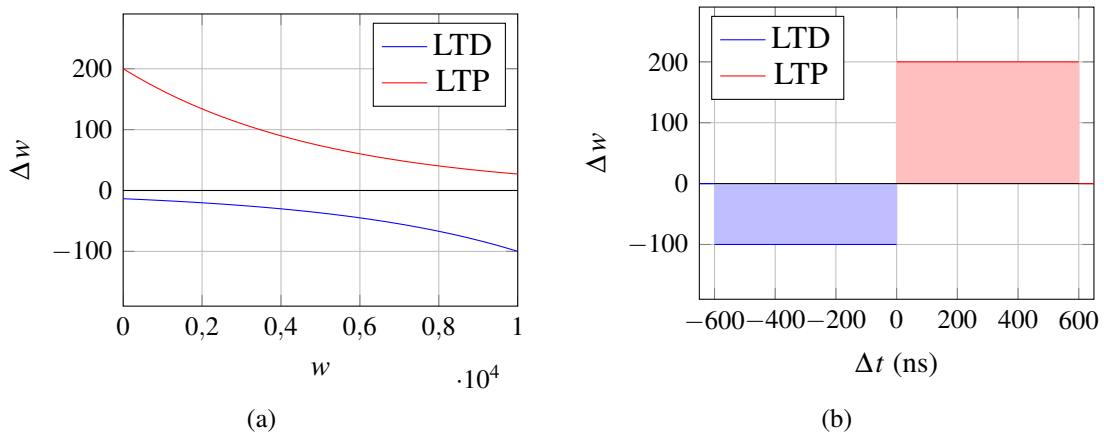


FIGURE 3.6 – Tracés des évolutions de Δw avec les coefficients $\alpha_+ = 200$, $\alpha_- = 100$ et $\beta_+ = \beta_- = 2$. (a) Variation Δw en fonction du poids précédent w . (b) Plage de variation Δw en fonction de Δt formant la loi de STDP.

Nous avons également utilisé d'autres modèles simplifiés de STDP qui sont présentés dans l'annexe C. Ces modèles s'appuient sur des formes de STDP plus biologiquement réalistes avec l'utilisation d'exponentielles pour la potentialisation et la dépression. Nous avons aussi extrait un modèle de formes de STDP à partir des mesures présentées au chapitre précédent dont nous allons donner le détail ci-après.

3.2.2 La modélisation issue de mesures des lois de variation de conductance des memristors BFO

Dans le chapitre précédent, nous avons présenté les mesures de formes de STDP réalisées sur des dispositifs memristifs ferroélectriques avec des PA (figures 2.25 et 2.26). Nous allons ici exploiter ces mesures en modélisant les formes de STDP à l'aide du modèle biologique développé par Song et Abbott (2001) :

$$F(\Delta t) = \begin{cases} A_- \exp(\Delta t / \tau_-) & \text{si } \Delta t < 0 \\ A_+ \exp(\Delta t / \tau_+) & \text{si } \Delta t \geq 0 \end{cases} \quad (3.2)$$

Cette équation donne une forme de **STDP** où le passage entre $\Delta t \geq 0$ et $\Delta t < 0$ est abrupte entre les valeurs A_+ et A_- . Nous l'avons donc modifiée pour obtenir l'équation suivante :

$$\Delta G(\Delta t) = \begin{cases} \frac{\Delta t}{\alpha} \exp(\Delta t / \tau_-) & \text{si } -t_{LTD} \leq \Delta t < 0 \\ \frac{\Delta t}{\alpha} \exp(\Delta t / \tau_+) & \text{si } 0 \leq \Delta t \leq t_{LTP} \end{cases} \quad (3.3)$$

où α est identique pour la partie positive et négative de la **STDP**. En utilisant un algorithme génétique, nous avons identifié les coefficients α , τ_+ et τ_- (annexe B) pour chacune des courbes mesurées (figure 2.26). La figure 3.7 présente deux exemples d'ajustement réalisés à partir des mesures de deux memristors différents. Le modèle biologique de **STDP** choisi s'adapte correctement à la forme de **STDP** du memristor **BFO**. L'ajustement de l'ensemble des mesures réalisées sur les memristors a donné pour les trois paramètres les valeurs moyennes suivantes : $\alpha = 3,53 \cdot 10^{-2}$, $\tau_- = 1,5050 \cdot 10^{-7}$ et $\tau_+ = 8,7026 \cdot 10^{-8}$.

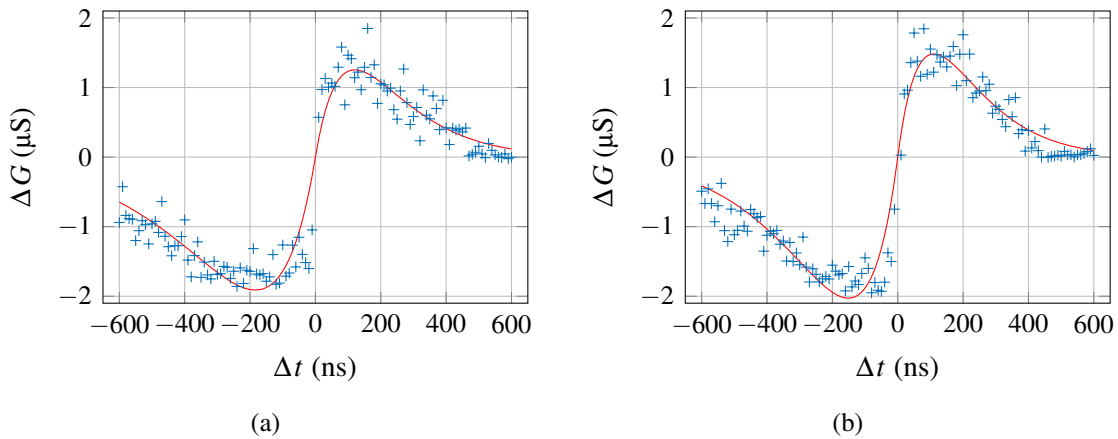


FIGURE 3.7 – Deux exemples d'ajustement des courbes de **STDP** mesurées sur les memristors **BFO**. (a) Ajustement pour une mesure de **STDP** du memristor 1. (b) Ajustement pour une mesure de **STDP** du memristor 2.

Les équations du modèle de variation de ΔG ainsi obtenues ne sont dépendantes que de Δt . Elles donnent les variations maximales de la conductance pour un G_{init} maximum dans le cas de la **LTP** et un G_{init} minimum dans le cas de la **LTD**. Pour les memristors ferroélectriques, la conductance du composant dépend de l'orientation des domaines ferroélectriques. L'équation de la variation de surface de domaine ferroélectrique négatif ds en fonction du temps est donnée dans Chanthbouala *et al.* (2012) :

$$\frac{ds}{dt} = (1 - s) \left\{ \frac{2[t - \tau_N(V)]}{\tau_P^2(V)} \right\} \quad (3.4)$$

où τ_N est le temps de nucléation¹ et τ_P est le temps de propagation. Ces deux variables dépendent

¹Temps pour lequel des changements de polarité s'opèrent.

de la tension appliquée aux bornes du composant. Donc ds dépend de la durée et de l'amplitude du pulse :

$$ds = (1 - s)F(t, V) \quad (3.5)$$

Pour une durée et une amplitude constante, la fonction F est constante, donc la variation de domaine est linéaire et fonction de la fraction de domaine ferroélectrique négatif s . Cette dernière est l'image de la conductance :

$$G = G_{init_{OFF}} + s(G_{init_{ON}} - G_{init_{OFF}}) \quad (3.6)$$

D'après la modélisation du memristor ferroélectrique, nous pouvons déduire que la variation de dG en fonction de G pour un pulse identique est linéaire. La conductance G du memristor est représentée dans les simulations par le poids synaptique w , c'est pourquoi nous avons implémenté l'équation suivante :

$$\Delta w(w, \Delta t) = \begin{cases} (w_{max} - w)A_- \frac{\Delta t}{\alpha} \exp(\Delta t / \tau_-) & \text{si } -t_{LTD} \leq \Delta t < 0 \\ wA_+ \frac{\Delta t}{\alpha} \exp(-\Delta t / \tau_+) & \text{si } 0 \leq \Delta t \leq t_{LTP} \\ 0 & \text{sinon} \end{cases} \quad (3.7)$$

La figure 3.8 montre la plage de variation de Δw en fonction de w (figure 3.8.a). Cette plage est constituée d'un ensemble de droites dont le coefficient directeur dépend de Δt . La loi de **STDP** résultante est donnée à la figure 3.8.b. Elle existe uniquement entre -600 ns et 600 ns et est nulle au-delà.

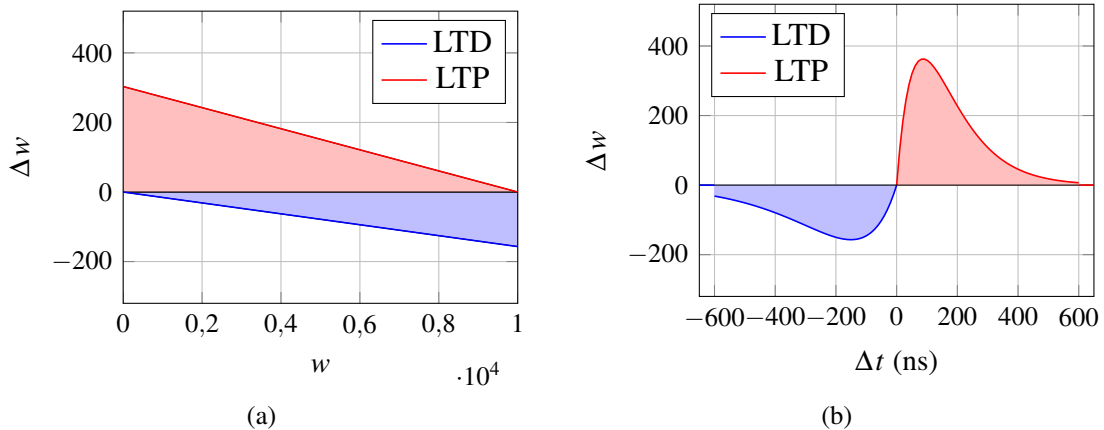


FIGURE 3.8 – Tracés des évolutions de Δw avec les coefficients $\alpha = 3,53 \cdot 10^{-2}$, $\tau_- = 1,5050 \cdot 10^{-7}$, $\tau_+ = 8,7026 \cdot 10^{-8}$, $A_- = 1 \cdot 10^4$ et $A_+ = 4 \cdot 10^4$. (a) Plage de variation Δw en fonction du poids précédent w . (b) Plage de variation Δw en fonction de Δt formant la loi de **STDP**.

3.3 Les simulations du réseau du projet MHANN

Afin d’avoir un point de comparaison avec les réseaux de neurones existants, nous avons décidé de faire des simulations de l’architecture de notre réseau en utilisant la base de données *Mixed National Institute of Standards and Technology* (MNIST)² qui regroupe 70 000 chiffres écrits à la main dont des exemples sont donnés à la figure 3.9 (Le Cun *et al.*, 1998). Les chiffres ont une dimension de 28×28 pixels et la valeur de leur pixel est comprise entre 0 (noir) et 255 (blanc). Ils sont divisés en deux fichiers : un premier groupe de 60 000 chiffres est dédié à l’apprentissage, tandis qu’un second groupe de 10 000 chiffres est utilisé pour tester l’apprentissage du réseau. Deux fichiers complémentaires sont fournis pour étiqueter chacun des chiffres des deux groupes et ainsi connaître leur valeur.



FIGURE 3.9 – Exemples de chiffres tirés de la base de données MNIST.

Le meilleur apprentissage utilisant la base de données MNIST a été obtenu par le réseau à 1100 neurones de Cireşan *et al.* (2012) avec un taux de reconnaissance de 99,77 %. Ce réseau est divisé en 6 couches et les images subissent un prétraitement avant d’être apprises par le réseau. Pour ce projet, notre but n’est pas d’obtenir le meilleur apprentissage possible mais de montrer qu’un réseau utilisant des memristors peut être utilisé pour l’apprentissage car ce type de réseau est un avantage pour une implémentation dans des systèmes embarqués.

Pour tester l’architecture qui a été mise en place, nous avons utilisé un simulateur événementiel avec une couche d’entrée de 784 capteurs (28×28 pixels) et une couche de sortie comprenant entre 50 et 500 neurones. Les résultats obtenus peuvent ainsi être mis en comparaison avec ceux de la littérature. Puis, pour nous aider dans la conception matérielle, nous avons simulé le réseau du projet MHANN avec 81 entrées (soit des images de 9×9 pixels) et 10 sorties en utilisant deux ou trois catégories de chiffres redimensionnés de la MNIST.

3.3.1 Le simulateur événementiel

Nos partenaires de l’INRIA ont codé un simulateur événementiel en C++ qui a l’avantage d’être plus rapide qu’un simulateur temporel (avec un pas de temps fixe) tant que les événements ne sont pas trop nombreux (Bichler *et al.*, 2013), ce qui est notre cas.

L’architecture du code est illustrée à la figure 3.10. Pour chaque image présentée en entrée, un événement inject est créé. Entre chaque événement inject, des événements `in_spike` sont générés suivant la valeur des pixels de l’image en cours de présentation et d’un bruit gaussien d’écart type σ (figure 3.11). Plus un pixel a une valeur élevée, plus la période de ces événements `in_spike` est faible. Pour un pixel d’une valeur de 255 (valeur maximale, le pixel est blanc) la période

²Disponible librement sur le site : <http://yann.lecun.com/exdb/mnist/>

est de $1,2 \mu\text{s}$. L'homéostasie qui régule la fréquence de décharge du neurone en modifiant sa tension de seuil apparaît à une période fixe avec l'évènement `homeo`. L'évènement `out_spike`, quant à lui, est créé lorsque que la tension de membrane a dépassé le seuil du neurone. Il génère un évènement `in_spike` pour tous les pixels non-actifs afin de permettre de déprécier le poids de leurs synapses. Nous avons implémenté une **STDP** bornée ce qui signifie que lorsque le code traite les évènements `in_spike`, il prend en compte les évènements `out_spike` jusqu'à un temps t_{LTD} en amont et déprécie le poids des synapses entre ces évènements. De même, lorsque que le code traite l'évènement `out_spike`, il prend en compte les évènements `in_spike` jusqu'à un temps t_{LTP} en amont et potentialise le poids des synapses entre ces évènements.

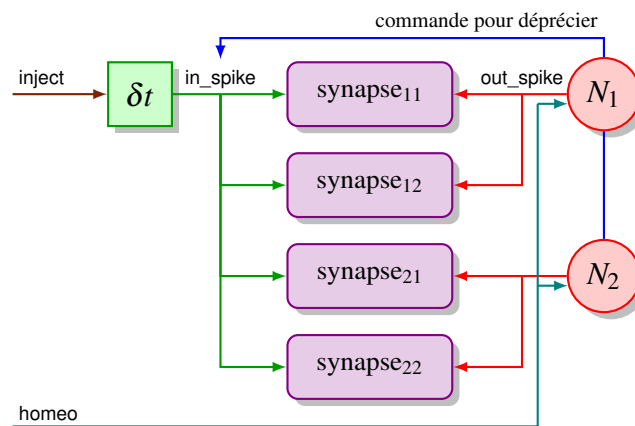


FIGURE 3.10 – Schéma illustrant les évènements du code C++ avec un réseau de 2 neurones et 4 synapses. Les évènements `inject` permettent de créer les évènements `in_spike`. Les évènements `out_spike` proviennent des neurones du réseau qui permettent aussi de générer des `in_spike` via une commande dédiée. Le poids des synapses est modifié suite aux évènements `in_spike` et `out_spike`. Le seuil des neurones est régulé lorsque les évènements `homeo` sont présents.

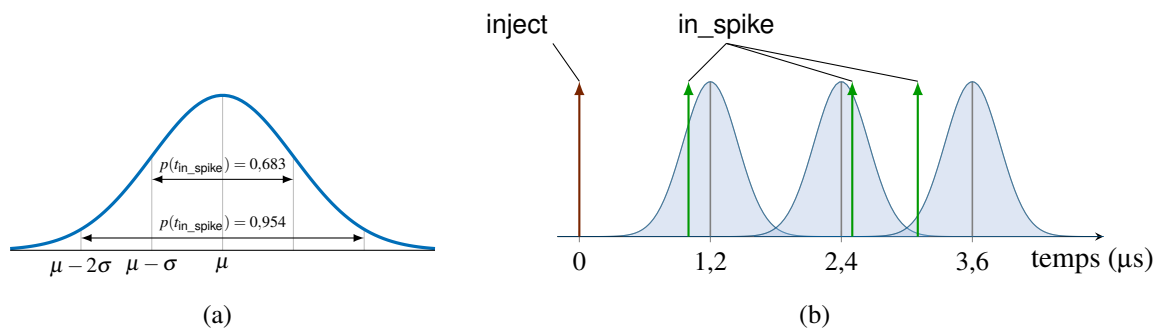


FIGURE 3.11 – Génération des évènements `in_spike` suivant une loi gaussienne et ayant une période dépendante de la valeur du pixel. (a) Loi gaussienne de valeur moyenne μ et d'écart type σ . La probabilité d'avoir un évènement à 1σ est de 68,3 %. (b) Exemple de génération d'évènements `in_spike` pour une période de $1,2 \mu\text{s}$ et $\sigma = 250 \text{ ns}$.

Comme nous l'avons vu dans le chapitre précédent, les temps t_{LTD} et t_{LTP} autour des évènements sont fixés par la forme du potentiel d'action (PA) que nous avons choisie (figure 2.22). Nous souhaitons utiliser des formes de PA identiques dans tout le réseau : le temps t_{LTx} est le

même pour toutes les impulsions. Donc nous avons réalisé les simulations avec des temps t_{LTD} et t_{LTP} similaires.

Les évènements sont gérés par un conteneur C++ appelé `priority_queue`. Il liste l'ensemble des évènements et les classe suivant une fonction de comparaison qui, dans notre cas, permet de trier les évènements par ordre croissant de temps. Les évènements les plus récents sont alors sur le dessus de la pile et sont donc traités en premier. Dès qu'un évènement est traité, il est enlevé de la liste et c'est l'évènement suivant qui est manipulé.

Le simulateur fonctionne suivant trois phases :

- l'apprentissage où les 60 000 premières images de la **MNIST** sont présentées de manière aléatoire ;
- la labélisation où les 60 000 images sont repassées pour déterminer les chiffres appris par chacun des neurones ;
- le test où les 10 000 dernières images de la **MNIST** sont présentées aléatoirement pour connaître le taux de réussite de l'apprentissage.

L'apprentissage non-supervisé nous contraint d'utiliser une phase de labélisation. Durant cette phase, des neurones ayant été actifs pour un trop grand nombre de labels différents sont écartés et ne participent pas à la phase de test. Ces neurones ne sont généralement pas assez spécialisés pour un label. Il est possible de les distinguer des autres par leurs connexions synaptiques qui ne révèlent aucune forme apprise. Pour le test, nous regardons quel neurone a franchi son seuil en premier. La valeur de son label est comparée avec la valeur du label de l'échantillon d'entrée. Le résultat des comparaisons nous donne le taux de réussite de l'apprentissage.

Avec l'aide de Jérémy Fratani, stagiaire de master 2 de l'université de Bordeaux, nous avons modifié le code du simulateur afin d'inclure la loi de **STDP** des memristors **BFO** (figure 3.8.b). Les simulations suivantes présentent les résultats obtenus avec cette loi de **STDP**.

3.3.2 Les simulations avec la modélisation de la STDP du memristor BFO

La figure 3.12 présente un résultat de simulation réalisée avec 100 neurones et ayant obtenu un taux de réussite de 75,8 %. Les poids des connexions synaptiques font apparaître les chiffres reconnus lors de l'apprentissage.

Nous avons réalisé des simulations pour connaître l'influence du niveau de bruit des échantillons d'entrée sur l'apprentissage (figure 3.13). Nous avons constaté qu'il est nécessaire que le bruit soit présent (Scarpetta *et al.*, 2013) car il apporte une amélioration significative du taux de reconnaissance. Pour un réseau de 100 neurones, il y a un bénéfice de 17 % entre les échantillons qui sont présentés sans aucun bruit autour des évènements `in_spike` et ceux présentés avec bruit gaussien d'écart type de 390 ns.

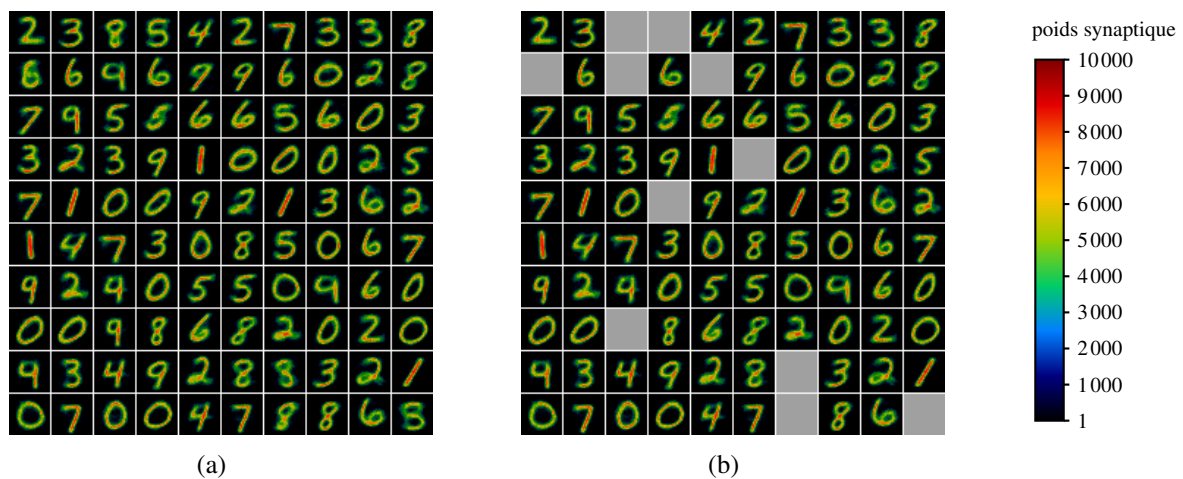


FIGURE 3.12 – Résultat d’un apprentissage de chiffres provenant de la [MNIST](#) avec 100 neurones de sortie ayant obtenu un taux de réussite 75,8 %. (a) Poids des synapses à la fin de l’apprentissage des 60000 images. (b) Élimination de certains neurones après la phase de labélisation.

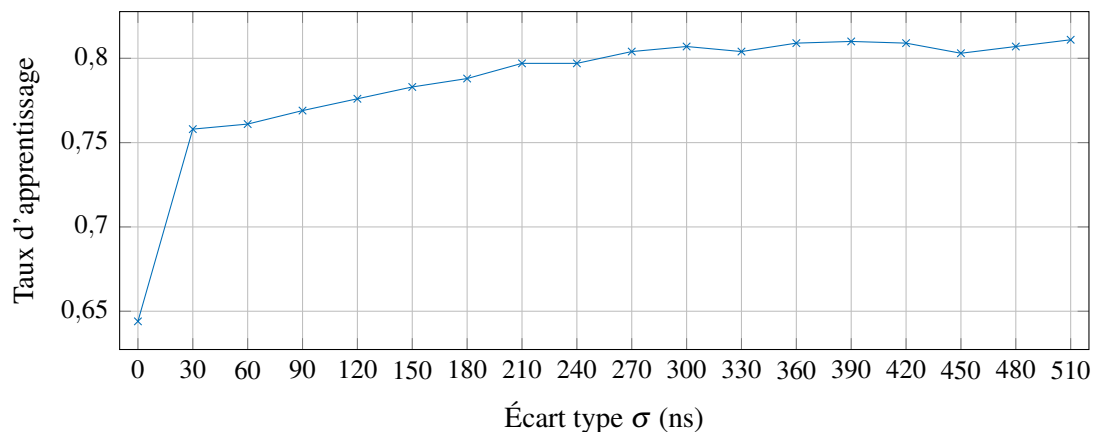


FIGURE 3.13 – Graphique montrant l’influence du bruit ajouté aux événements d’entrée sur l’apprentissage avec un réseau de 100 neurones.

De même, nous avons comparé l’apprentissage de l’architecture avec un nombre différent de neurones de sortie (figure 3.14). L’augmentation du nombre de neurones améliore le taux de réussite. En effet, plus le nombre de neurones est important, plus il est possible de discriminer de petites variations dans la forme pour une même classe de chiffres.

En combinant ces deux paramètres, un bruit d’entrée de $\sigma = 350$ ns et 500 de neurones de sortie, nous avons eu un taux de reconnaissance de 88,8 %. Ce taux de reconnaissance est loin des 99 % obtenus avec des réseaux perceptrons multicouches ([Le Cun et al., 1998](#) ; [Simard et al., 2003](#) ; [Cireşan et al., 2012](#)). Ceci peut s’expliquer par le plus faible nombre de neurones et de connexions synaptiques. De plus, l’implémentation avec une seule couche de neurones impose que les chiffres appris soient semblables aux chiffres de la base de données test. Or un décalage horizontal de quelques pixels fait qu’un 1 ne pourra pas être reconnu par exemple.

En supposant que la distribution des variations des paramètres des memristors à l’intérieur

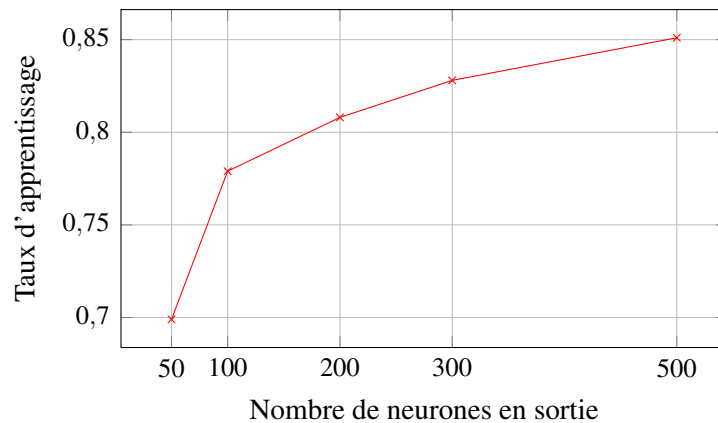


FIGURE 3.14 – Graphique montrant l’influence du nombre de neurones sur l’apprentissage avec un écart type de bruit d’entrée de 120 ns.

d’un même *crossbar* suit une loi gaussienne³, des simulations ont été réalisées en faisant varier les paramètres α , τ_+ et τ_- . Nous avons ainsi déterminé l’écart type maximum autour de la valeur moyenne pour chacun des paramètres pour conserver un taux de reconnaissance correct (table 3.1).

TABLE 3.1 – Valeurs limites de la distribution gaussienne des memristors dans un *crossbar* pour un taux d’apprentissage de 70,2 % d’un réseau de 100 neurones (soit 78400 memristors).

	α	τ_-	τ_+
valeur moyenne	$3,53.10^{-2}$	$1,5050.10^{-7}$	$8,7026.10^{-8}$
écart type	$1,3414.10^{-2}$	$5,179.10^{-8}$	$3,3070.10^{-8}$

Nous avons ensuite testé le réseau que nous souhaitons implémenter matériellement. Nous avons gardé les images de la *MNIST* que nous avons réduites à 9×9 pixels. Les poids synaptiques des connexions après apprentissage sont donnés à la figure 3.15. Les formes des chiffres ainsi réduites ne sont pas réellement reconnaissables. Au vu de ces résultats nous n’utiliserons pas la *MNIST* pour tester le démonstrateur. Nous choisirons d’utiliser des formes personnalisées comme présenté à la figure 3.16.

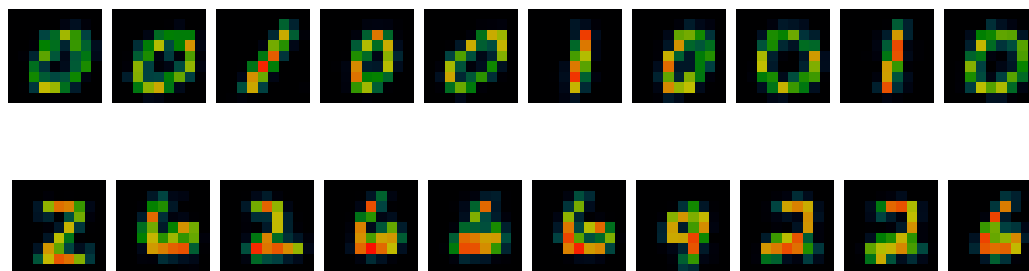


FIGURE 3.15 – Résultat d’un apprentissage de deux chiffres 0 et 1 (en haut) et trois chiffres 2, 6 et 9 (en bas) provenant de la *MNIST* et réduites à 9×9 pixels avec 10 neurones de sortie.

³Les mesures que nous avons réalisées sont trop peu nombreuses pour le confirmer.

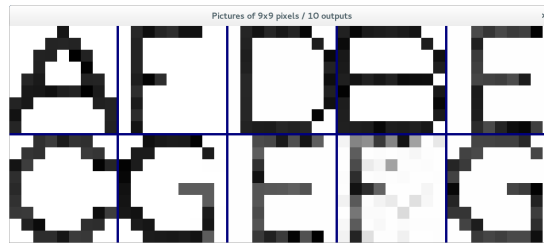


FIGURE 3.16 – Exemples d'échantillons pour tester le réseau de 81 entrées et 10 sorties.

3.4 Conclusion

Dans ce chapitre, nous avons donné un aperçu du principe d'un réseau de neurones plastique. Après avoir exposé les principales architectures de réseaux de neurones pouvant être utilisées pour la reconnaissance de formes avec leurs apprentissages, nous avons donné des exemples de réseaux hybrides issus de la littérature. Puis, nous avons détaillé le choix du réseau utilisé dans le projet [MHANN](#). Afin de pouvoir l'implémenter facilement et de minimiser le nombre de neurones tout en ayant une bonne capacité d'apprentissage, nous avons choisi d'utiliser un réseau compétitif à apprentissage non-supervisé. Il lui a été ajouté des propriétés de période réfractaire et de commande de retour pour utiliser une loi de [STDP](#) bornée « classique ».

Nous avons ensuite donné différentes modélisations de [STDP](#) possibles. Puis, nous avons évalué l'ajustement des formes de [STDP](#) à partir des mesures réalisées sur des memristors [BFO](#) présentées au chapitre 2. La modélisation s'appuie sur une équation de forme de [STDP](#) biologique.

Enfin, nous avons présenté les simulations qui ont été réalisées avec la [STDP](#) des memristors [BFO](#). La base de données [MNIST](#) contenant 70000 chiffres écrits à la main a été utilisée. Nous avons observé que le bruit d'entrée ou bien le nombre de neurones sont des variables importantes pour l'apprentissage des chiffres de la [MNIST](#). Il a également été montré que la variation due au *mismatch* des memristors n'empêche pas l'apprentissage du réseau. Nous sommes en capacité de donner les limites de dispersion des memristors dans un *crossbar* qui permettent de garder un apprentissage correct (70 % de reconnaissance).

Les taux de reconnaissance sont en dessous de ceux trouvés dans la littérature mais ils sont encourageants et permettent de valider l'architecture proposée. En revanche, il a été montré qu'il n'est pas possible d'utiliser la base [MNIST](#) pour l'apprentissage du démonstrateur étant donné le faible nombre de capteurs d'entrée. Une base de donnée créée par nos soins servira, dans la dernière année du projet [MHANN](#), à l'apprentissage du prototype matériel.

Les simulations nous ont permis de jouer avec différents paramètres du réseau comme la valeur de la capacité de membrane, le bruit d'entrée ou les délais de propagation des signaux et de déterminer un ordre de grandeur pour l'ensemble des paramètres du système matériel que nous allons détailler dans le prochain chapitre.

CHAPITRE 4

LA CONCEPTION SUR SILICIUM

LE projet **MHANN** a pour but d'implémenter un réseau de neurones avec des memristors en lieu et place des synapses. Nous avons démontré dans les chapitres précédents que nous pouvions utiliser un neurone impulsionnel. En effet, les évènements générés par ces neurones et la concomitance de leurs formes d'onde permettent d'utiliser la plasticité de la synapse memristive et d'obtenir une modification de la conductance synaptique suivant une loi de **STDP** voulue.

Il n'est pas encore possible, à ce jour, d'avoir des memristors ferroélectriques sur une puce silicium de technologie standard. Dans le cadre du projet, il a été décidé d'utiliser deux puces séparées et connectées sur une même carte électronique : la première inclut une couche de capteurs et une couche de neurones et la seconde renferme un *crossbar* de memristors.

Au début du projet, l'étude du comportement memristif de composants ferroélectriques portait sur une technologie à barrière ferroélectrique BaTiO_3 (**BTO**). Une première puce silicium incluant les deux couches du réseau a été adaptée à cette technologie de memristor. Les memristors **BTO** ont des seuils de variation de résistance de 1,3 V et $-2,45$ V pour un pulse d'une durée de 100 ns. Ces seuils sont plus élevés en valeur absolue que les seuils du memristor à barrière ferroélectrique BiFeO_3 (**BFO**) étudié au chapitre 2 (0,8 V et $-1,0$ V). Développée pendant le projet, c'est cette seconde technologie qui a été choisie pour fabriquer le *crossbar*. Au vu de leur plus faible niveau de seuil, les memristors **BFO** sont de meilleurs candidats pour l'hybridation avec la technologie sur silicium. Mais la première puce silicium créée n'était pas compatible avec ces nouvelles tensions de seuil, une seconde puce a donc été conçue pour réaliser un réseau de neurones dédié aux memristors **BFO**.

Pour concevoir les deux couches du réseau, nous avons choisi la technologie CMOS 0,18 μm avec 6 couches de métallisation d'austriamicrosystems et nous avons utilisé le logiciel de conception cadence[®] avec le simulateur Virtuoso[®] Spectre[®].

Les simulations du réseau en C++, exposées au chapitre précédent, nous ont montré qu'il est possible d'employer le memristor comme une synapse à part entière, *i.e.* en remplissant ses trois fonctions : la plasticité, la mémoire, et la pondération. Dans ce chapitre, nous allons présenter une solution technique permettant d'utiliser ces trois fonctions du memristor. Puis nous allons détailler les mesures de la première puce que nous avons conçue. Nous donnerons ensuite les améliorations apportées dans la seconde puce, illustrées par des simulations. Des mesures ont uniquement été faites sur la première puce puisqu'au moment où nous écrivons ces lignes la seconde puce est en cours de fabrication.

4.1 La connexion entre une synapse memristive et un neurone impulsionnel

Pour que le memristor remplisse le rôle de la synapse, il doit intervenir lors de la charge de la capacité de membrane du neurone postsynaptique lorsqu'un évènement présynaptique est présent (pondération de la synapse). De plus, lorsque les neurones pré- et postsynaptiques

gènèrent des évènements proches (durée inférieure aux temps de **LTD** et **LTP**), le memristor subit des modifications de la valeur de sa résistance (plasticité). Donc le memristor doit se retrouver à la fois en entrée et en sortie du neurone postsynaptique. Cette configuration devient possible en utilisant un convoyeur de courant de seconde génération (**CCII**) placé entre le memristor et le neurone postsynaptique.

Le **CCII** est un tripôle qui possède deux propriétés intéressantes :

- le courant sortant de la borne Z est une copie du courant entrant dans la borne d'entrée X à un coefficient α près : $i_Z \leftarrow \alpha i_X$, avec $\alpha \in \mathbb{R}$;
- il recopie le potentiel de la borne d'entrée Y sur la borne d'entrée X : $v_X \leftarrow v_Y$.

Deux types de convoyeurs de courant sont définis suivant le signe de α : **CCII+** pour $\alpha > 0$ et **CCII-** pour $\alpha < 0$.

Illustré à la figure 4.1, le montage que nous proposons fait l'objet d'un brevet ([Saïghi et al., 2013](#)). Il permet d'utiliser toutes les propriétés du memristor : la plasticité, la mémoire et la pondération.

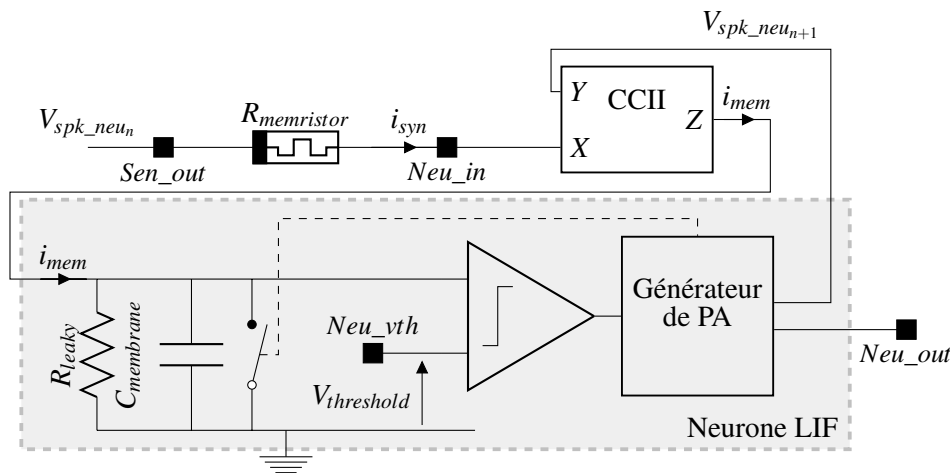


FIGURE 4.1 – Schéma de connexion entre une synapse memristive et un neurone impulsif **LIF**. Les carrés noirs représentent les entrées et sorties de la puce silicium (*Neu_in* et *Neu_vth* sont des entrées ; *Neu_out* et *Sen_out* sont des sorties).

L'implémentation du neurone impulsif **LIF** est séparée en deux. D'un côté, une partie membrane est composée d'une capacité $C_{membrane}$ couplée à une résistance de fuite R_{leaky} et un comparateur, et d'un autre côté, un générateur de **PA** qui permet d'avoir une forme de **PA** adaptée au memristor et à la loi de plasticité choisie.

Sans aucun évènement venant des neurones, le memristor a entre ses bornes le potentiel de repos du neurone présynaptique et le potentiel de repos du neurone postsynaptique qui est recopié par le **CCII** de la borne Y à la borne X. Les deux potentiels de repos étant identiques, le memristor ne voit aucune différence de potentiel entre ses bornes. Lorsqu'un potentiel d'action présynaptique $V_{spk_neu_n}$ arrive aux bornes du memristor, il génère un courant i_{syn} proportionnel

à la résistance du memristor. Ce courant arrive sur la borne X du **CCII**, est recopié à la borne Z et devient le courant i_{mem} qui va permettre de charger la capacité de membrane $C_{membrane}$ du neurone postsynaptique. Dès que le potentiel de membrane dépasse le potentiel de seuil $V_{threshold}$, le neurone génère un **PA** $V_{spk_neu_{n+1}}$ qui est envoyé via Neu_out à la couche suivante du réseau et aussi à la borne Y du **CCII**. Le **PA** postsynaptique se retrouve ainsi aux bornes du memristor. La coïncidence des évènements pré- et postsynaptiques permet donc de modifier directement la résistance du memristor.

Lorsque le potentiel de membrane a dépassé la tension de seuil, un interrupteur est commandé pour décharger la capacité. Le neurone rentre alors dans une phase réfractaire : aucun évènement ne peut venir charger sa capacité de membrane et donc aucune modification de poids synaptique ne peut se faire. En revanche, lorsque les évènements présynaptiques sont trop éloignés, la résistance R_{leaky} du neurone postsynaptique décharge la capacité de membrane.

À ce jour, aucune solution de montage ne propose d'utiliser différentes polarités de synapses memristives. L'originalité du montage que nous proposons avec le **CCII** permet de définir la propriété excitatrice ou inhibitrice des synapses connectées en amont (Lecerf *et al.*, 2013a). En utilisant un *crossbar* de memristors, si les lignes sont connectées à la sortie des neurones présynaptiques, alors les courants résultants de la loi des nœuds de chaque colonne doivent être dirigés vers les entrées des neurones postsynaptiques. Pour ce faire, à chacune des colonnes est connecté un **CCII** dont la polarité définit le type des synapses reliées (figure 4.2.a). Les synapses d'une même colonne ont donc une polarité identique. Il est aussi possible de connecter différentes colonnes de memristors à un neurone en reliant les sorties des **CCII** à l'entrée du même neurone (figure 4.2.b).

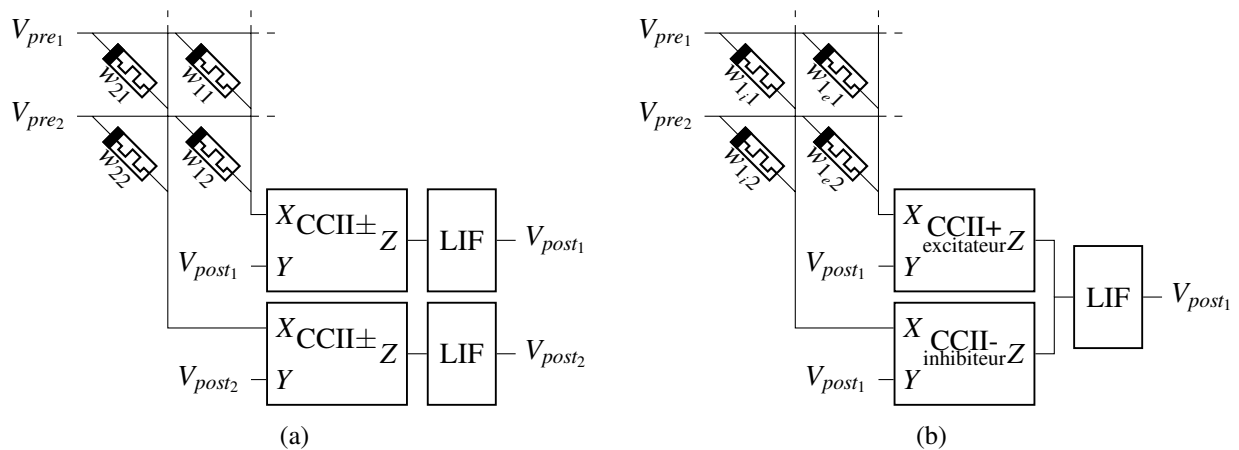


FIGURE 4.2 – Utilisation des convoyeurs de courant avec un *crossbar* de memristor. (a) Un convoyeur est connecté par colonne. (b) Les convoyeurs de courant définissent les propriétés excitatrices ou inhibitrices des synapses connectées aux neurones subséquents.

De plus, cette solution est tournée vers l'extension du réseau. L'utilisation du **CCII** entre le memristor et le neurone permet de cascader les couches de neurones sans changer de structure. Il devient alors facile d'étendre le réseau. Il existe cependant une limitation dépendante du nombre

de memristors que peut piloter un neurone présynaptique et du nombre de memristors qu'il est possible de connecter à l'entrée X du convoyeur de courant (côté postsynaptique). L'ordre de grandeur est de 100 memristors BFO en parallèle pour un courant délivré ou injecté d'environ 1 mA.

Soutenue par le CNRS, la solution de connexion que nous avons proposée est incluse dans un portefeuille de brevet avec celui du memristor ferroélectrique notamment (Bibes *et al.*, 2010), et est portée pour une extension à l'international.

Nous allons maintenant détailler les blocs utilisés dans la première puce afin de mettre en œuvre le montage breveté pour connecter des synapses memristives à des neurones réalisés sur silicium. Les mesures obtenues à l'aide d'une carte test réalisée par Ashwin Mangalore, ingénieur de l'équipe, nous ont permis de mettre en évidence quelques dysfonctionnements de ce premier circuit intégré comme nous allons le constater dans le 4.2. Des solutions seront apportées au 4.3 par la seconde puce qui a de plus été conçue pour des memristors BFO ayant de plus faibles tensions de seuil que les BTO.

4.2 Le détail des blocs et les mesures de la première puce

La première puce nommée *Spiking Analog Neural Network Winner-Take-All (SpANNWiTA)* contient une couche d'entrée du réseau composée de 81 capteurs, une couche de sortie du réseau composée de 10 neurones et des blocs éclatés afin de tester séparément les éléments du réseau (figure 4.3). Un bloc numérique permet de faire rentrer les signaux de commande pour la couche d'entrée.

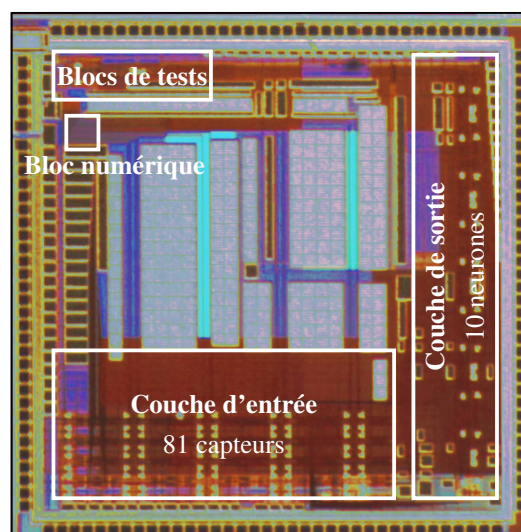


FIGURE 4.3 – Microphotographie de SpANNWiTA (142 plots, surface = 9,175 mm²).

Le cahier des charges de l'ensemble des blocs est dépendant des caractéristiques des PA à générer. Pour la première puce, s'agissant de PA utilisés avec des memristors BTO, leurs

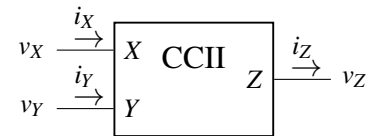
caractéristiques étaient (suivant les conventions utilisées à la figure 2.22) : $A_{pulse_{pré}} = 0,8 \text{ V}$, $A_{pulse_{post}} = 1,6 \text{ V}$, $A_{offset_{max}} = -1,3 \text{ V}$, $A_{offset_{min}} = -0,9 \text{ V}$, $t_{pulse} = 100 \text{ ns}$ et $t_{LTx} = 500 \text{ ns}$.

4.2.1 Le convoyeur de courant de seconde génération

Le montage

Le fonctionnement du convoyeur de courant de seconde génération peut être résumé sous forme de matrice :

$$\begin{bmatrix} i_Y \\ v_X \\ i_Z \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & \pm \alpha & 0 \end{bmatrix} \begin{bmatrix} v_Y \\ i_X \\ v_Z \end{bmatrix}$$



Il existe différents montages pour implémenter un CCII. Nous avons choisi d'utiliser celui présenté par [Toumazou et al. \(1993\)](#), et illustré à la figure 4.4. Pour satisfaire l'équation $v_X = v_Y$, un amplificateur opérationnel est monté en suiveur, et pour répondre à l'équation $i_Z = \pm \alpha i_X$, un montage suiveur de courant est cascadié à l'amplificateur. Dans l'illustration, ce sont des doubles miroirs de courant simple qui sont utilisés pour recopier le courant i_X en entrée et générer le courant i_Z en sortie.

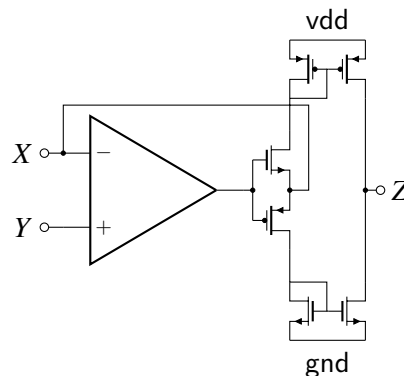


FIGURE 4.4 – Schéma de principe du convoyeur de courant tiré de [Toumazou et al. \(1993\)](#).

Dans la couche de sortie du réseau, nous avons uniquement employé des CCII+ en amont des neurones, car nous n'utilisons que des synapses excitatrices dans le réseau (cf. 3.1.3). Toutefois, pour pouvoir faire des mesures, nous avons implanté sur la puce un CCII- et un CCII+ avec leurs entrées et sorties accessibles de l'extérieur. Nous n'avons conçu que des CCII avec un coefficient α unitaire.

Pour l'étage de sortie des CCII, nous avons choisi d'utiliser un double miroir de courant cascadié : double car l'impulsion est positive puis négative, donc le courant traversant le memristor va dans les deux sens ; cascadié car cela permet d'avoir un gain en courant plus précis¹ sans devoir rajouter une tension de polarisation supplémentaire. En revanche, cette technique réduit

¹Le miroir de courant cascadié permet d'imposer un V_{DS} identique aux transistors du miroir simple.

l'excursion de la tension de sortie du montage qui est limitée par la somme des tensions $2V_{DS_{sat}} + V_{th}$ des transistors de sortie. Avec ces tensions de déchet d'environ 1,1 V et une alimentation de $\pm 1,65$ V, il ne devient plus possible de recopier correctement le PA. Nous avons donc décidé d'augmenter la plage d'alimentation du CCII en passant de $\pm 1,65$ V à $\pm 2,5$ V².

Pour l'étage d'entrée, nous avons conçu un amplificateur à compensation par effet Miller avec un double étage différentiel à charge cascodée repliée en entrée (figure 4.5). Nous avons choisi cet étage *rail-to-rail* car les spécifications du PA n'étaient pas encore figées au moment de la conception de l'amplificateur. La compensation par capacité Miller sur l'étage de sortie permet d'avoir la stabilité du montage en boucle fermée. Des modifications ayant été apportées suite au changement de niveau de tension du CCII, nous avons gardé l'étage d'entrée *rail-to-rail*. En revanche, nous avons modifié l'étage de sortie en utilisant un simple montage source commune car finalement l'impulsion à copier a été figée, elle possède une tension négative $A_{offset_{max}}$ fixe de $-1,3$ V, mais une tension positive A_{pulse} réglable. Nous avons donc un amplificateur qui est *rail-to-rail* en entrée mais pas en sortie.

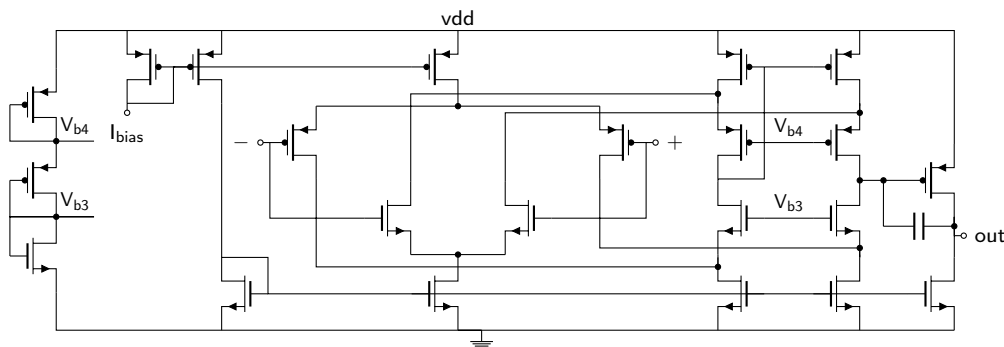


FIGURE 4.5 – Schéma transistor de l'amplificateur opérationnel du CCII.

Le schéma électrique des $CCII_{\pm}$ est visible à la figure 4.6. Un $CCII-$ (figure 4.6.b) est obtenu en croisant les sorties des miroirs de courant du $CCII+$ (figure 4.6.a) et en y ajoutant un double étage de miroirs de courant cascodés.

²Nous avons utilisé les transistors de grille semi-épaisse de la technologie AMS 0,18 μm dont les tensions d'alimentation peuvent être au maximum de $\pm 2,5$ V.

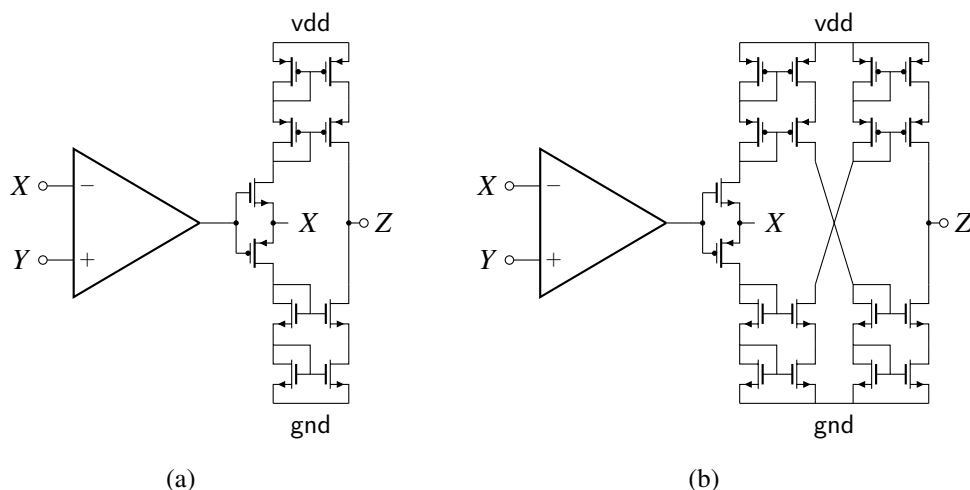


FIGURE 4.6 – Schéma électrique des convoyeurs de courant de seconde génération implémentés dans la première puce. (a) CCII+ ; (b) CCII-.

Les mesures

Dans un premier temps, nous avons procédé à des mesures en statique sur les CCII du bloc test, présentées à la figure 4.7. Que ce soit pour le CCII+ ou le CCII-, les mesures statiques montrent la linéarité des recopies aussi bien en courant (figure 4.7.a et 4.7.b) qu'en tension (figure 4.7.c et 4.7.d).

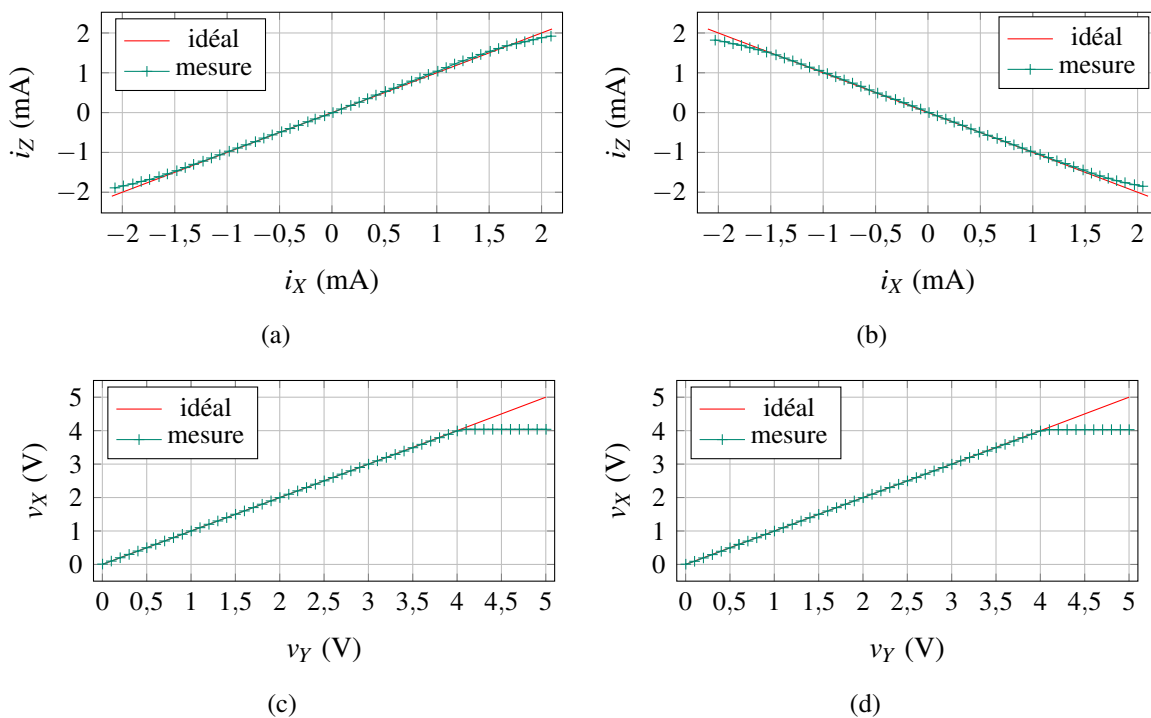


FIGURE 4.7 – Mesures statiques faites sur les CCII±. Tests de la copie de courant de la borne X à la borne Z des CCII+ (a) et CCII- (b) avec une charge de 1 kΩ aux bornes X et Z. Tests de la reproduction de la tension de la borne Y sur la borne X des CCII+ (c) et CCII- (d).

Puis nous avons réalisé des mesures avec les potentiels d'action aux caractéristiques suivantes : $A_{pulse} = 0,8 \text{ V}$, $A_{offset_{max}} = -0,8 \text{ V}$, $A_{offset_{min}} = -0,2 \text{ V}$, $t_{pulse} = 100 \text{ ns}$ et $t_{LTx} = 500 \text{ ns}$ (figures 4.8 et 4.9). Les mesures de la figure 4.8.a montrent la recopie d'un PA sur la borne X initialement appliqué sur la borne Y. Le temps d'établissement de l'impulsion v_X pour la montée est d'environ 80 ns. Puis le pulse a une durée d'environ 100 ns et le temps de descente est aussi de 100 ns. L'impulsion recopiée accuse un retard d'environ 200 ns quand elle est de nouveau superposée avec l'impulsion d'entrée.

Nous avons réalisé des simulations de plasticité résultante (figure 4.8.b) en considérant d'une part une impulsion présynaptique parfaite (avec des temps de montée de l'ordre de la nanoseconde) appliquée sur la borne présynaptique d'un memristor et d'autre part l'impulsion déformée obtenue après recopie du CCII appliquée sur la borne postsynaptique. Comparée à l'application de deux PA idéaux, l'altération de l'impulsion postsynaptique déforme la loi locale de STDP. Les extremums sont déplacés, la conséquence peut être un ralentissement de l'apprentissage dû à des erreurs de variation de conductance. Si les deux formes ont des pentes similaires, la STDP retrouve sa symétrie. L'apprentissage par chevauchement des formes de PA nécessite une symétrie des impulsions.

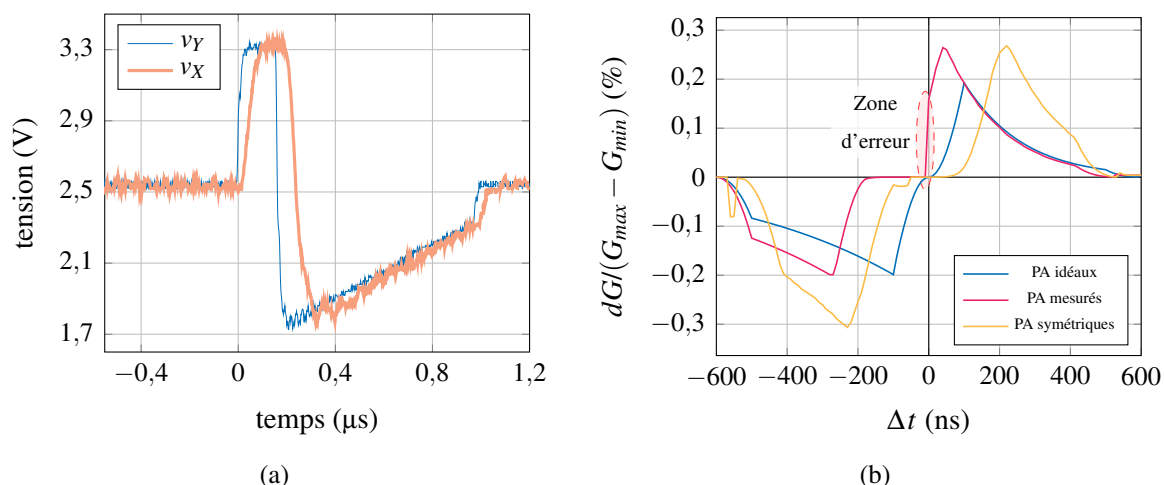


FIGURE 4.8 – Mesures de recopie de la tension réalisées sur le CCII+ avec un PA. (a) L'impulsion v_Y générée par un GBF connecté à la borne Y du CCII est dupliquée à la borne X (v_X). (b) Simulation Matlab[®] des STDP résultantes pour des signaux idéaux et mesurés. La troisième simulation « PA symétriques » a été réalisée avec l'envoi de deux PA déformés entre les bornes d'un memristor.

Les courbes de la figure 4.9 ont été obtenues en appliquant un PA $V_{pré}$ à une charge résistive de $1 \text{ M}\Omega$ connectée à l'entrée X du CCII+. Une résistance de $100 \text{ k}\Omega$ était connectée à la sortie Z du CCII+. La forme du pulse de courant i_Z du CCII est arrondie. Pour la recopie en courant, il n'est pas nécessaire d'avoir la forme exacte du PA, mais il est important que le courant i_Z puisse charger la capacité de membrane du neurone connecté, ce qui est tout à fait possible avec la forme obtenue.

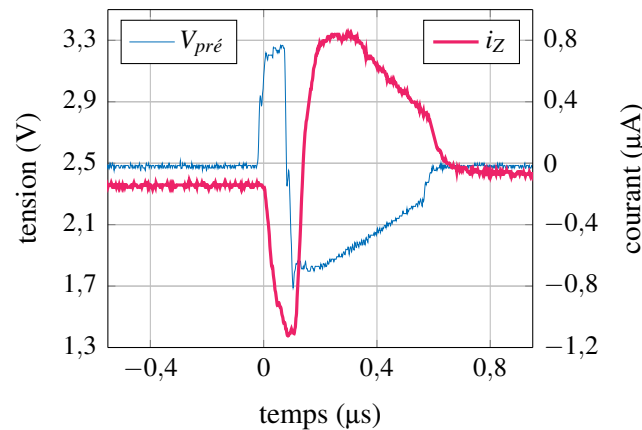


FIGURE 4.9 – Mesures de recopie du courant réalisées sur le **CCII+** avec un **PA**. L'impulsion V_{pre} engendre un courant en traversant une résistance de test de $1\text{ M}\Omega$ connectée à la borne X du **CCII**. i_Z est le courant de la borne Z du **CCII** qui est la copie du courant entrant à la borne X .

Les convoyeurs de courant remplissent en partie le cahier des charges :

- les mesures en statique montrent une recopie à l'identique des courants de l'entrée X vers la sortie Z avec une limite de $\pm 1,5\text{ mA}$;
- les mesures en statique montrent une recopie à l'identique des tensions de l'entrée Y vers l'entrée X avec une saturation à 4 V ;
- les mesures en dynamique montrent une recopie en courant non parfaite mais satisfaisante pour notre application ;
- les mesures en dynamique montrent une recopie en tension avec des temps lents de montée et de descente : il est nécessaire que les formes des impulsions pré- et postsynaptiques soient similaires.

La sortie du convoyeur de courant est connectée au neurone **LIF** que nous allons détailler dans la partie suivante.

4.2.2 Le neurone **LIF**

Avec l'utilisation de formes de **PA** adaptées à l'apprentissage avec des memristors, le neurone **LIF** doit être découpé en deux parties : la membrane et le générateur d'impulsion. La membrane intègre les courants générés suite aux impulsions des neurones de la couche précédente. Lorsqu'il y a dépassement du seuil du neurone, la membrane envoie un signal au générateur de **PA**. Il va alors créer deux impulsions :

- une impulsion postsynaptique qui se retrouve sur les memristors connectés aux neurones de la couche précédente grâce au **CCII** ;
- une impulsion présynaptique qui arrive sur les memristors connectés aux neurones de la couche suivante.

4.2.2.1 La membrane

Le montage

Le composant principal de la membrane (figure 4.10) est une capacité. Elle permet de faire la somme des contributions de chaque impulsion. En parallèle de celle-ci, un transistor NMOS en régime de saturation modélise la résistance de fuite (R_{leaky}). Le transistor est polarisé par un montage CMOS assurant une tension V_{GS} fixe et assure un courant de fuite constant. Un second transistor NMOS se trouve en parallèle de la capacité, il s'agit d'un interrupteur permettant de faire court-circuit aux bornes de la capacité de membrane et ainsi de la décharger. La tension de grille est commandée par le signal *resetCmem* qui vient de la sortie d'une porte OU.

Un comparateur est utilisé pour détecter si la tension aux bornes de la capacité est supérieure au seuil $V_{threshold}$. Dès que la tension aux bornes de la capacité dépasse la tension de seuil, le comparateur passe d'un niveau bas à un niveau haut. L'information est transmise au générateur d'impulsion qui à son tour envoie une commande de décharge de la capacité. La tension aux bornes de la capacité commence donc à chuter passant alors en dessous du seuil. La sortie du comparateur quant à elle passe alors du niveau haut vers le niveau bas. Les délais de ces commutations sont de l'ordre de la nanoseconde, bien inférieurs aux temps nécessaires pour commander le montage qui génère le potentiel d'action. Un monostable se trouve en sortie du comparateur afin d'avoir un pulse calibré (sa durée doit être comprise entre t_{pulse} et $t_{pulse} + t_{LTx}$). La durée du pulse créé par le monostable est donc ajustée à 300 ns.

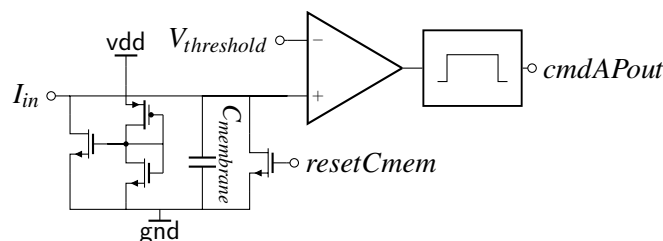


FIGURE 4.10 – Schéma de la membrane du neurone.

La capacité de membrane $C_{membrane}$ de type MIM³ occupe une surface de $20 \times 17,5 \mu\text{m}^2$ pour une valeur de 708,8 fF. La fuite de courant venant de R_{leaky} est d'environ 60 nA. La résistance du memristor étant comprise entre 100 k Ω et 10 M Ω , avec un PA il est alors possible de charger la capacité initialement déchargée entre 60 mV et 6,8 V. La tension maximale de charge peut être supérieure aux tensions d'alimentation puisque la charge sera coupée par le dépassement de la tension de seuil. La tension de charge minimale doit être inférieure aux tensions d'alimentation qui limitent la valeur maximale de $V_{threshold}$.

³Métal-Isolant-Métal de l'anglais *Metal-Insulator-Metal*.

Les mesures

Nous avons effectué les mesures de la membrane du bloc test à l'aide du convoyeur de courant positif (figure 4.11.a). Un niveau continu de 2,5 V est imposé sur les entrées X et Y du CCII. La sortie Z du CCII est connectée à l'entrée du bloc membrane *TestNeuIn*. La tension de seuil $V_{threshold}$ est fixée à 1,5 V et l'entrée *resetCmemIn* est connectée à gnd. La valeur de la résistance de test de 1 kΩ nous permet de charger la membrane du neurone à une valeur d'environ 2,8 V avec un seul PA (figure 4.11.b).

Sur la figure 4.11.b, nous pouvons voir que l'envoi du PA sur V_{in} charge la capacité de membrane $V_{TestNeuIn}$. Puis la capacité se décharge à travers la résistance de fuite R_{leaky} . Lorsque la tension de la membrane dépasse la tension de seuil, un pulse de 3,3 V d'amplitude et d'environ 300 ns devrait apparaître en sortie du bloc membrane $V_{cmdAPout}$. Or nous n'observons qu'un léger changement de potentiel de quelques dizaines de millivolts, valeur insuffisante pour commander le générateur de PA connecté en suivant. La membrane du bloc test ne nous permet donc pas de vérifier si le monostable envoie un pulse suffisamment long pour commander le générateur de PA.

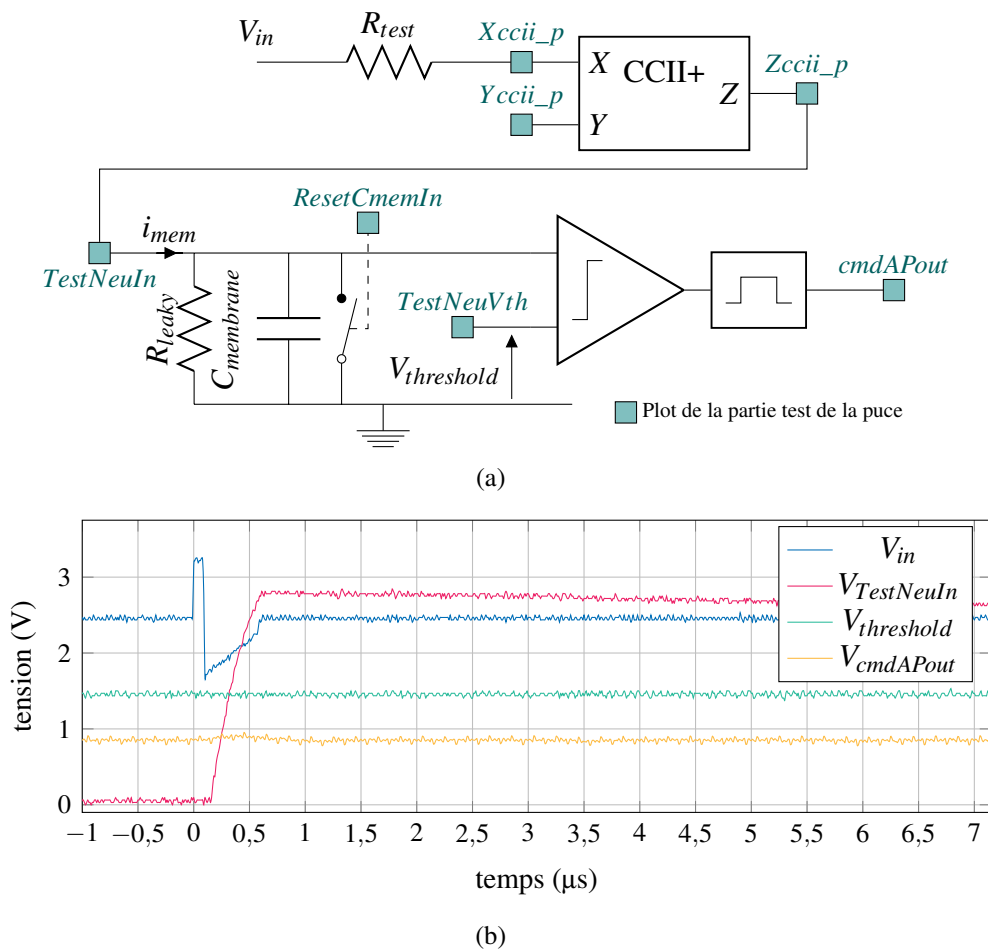


FIGURE 4.11 – Mesures de la membrane du bloc test. (a) Schéma des entrées/sorties de la membrane du bloc test. (b) Graphique montrant les tensions de sortie de la membrane après application d'un PA.

Par ailleurs, nous avons mesuré une décharge en 130 μs pour une charge de la capacité de 4,7 V. Pour une même valeur de charge, la simulation donne un temps de décharge de 55 μs . L'utilisation de capacité MIM peut expliquer en partie cette différence. Les valeurs des capacités surfaciques sont données avec une approximation $\pm 15\%$.

Les mesures de la membrane montrent que la capacité $C_{membrane}$ et le transistor jouant le rôle de la résistance de fuite R_{leaky} sont fonctionnels. En revanche le monostable ne donne pas satisfaction. Nous verrons dans la suite de ce document que les monostables des neurones, dont aucune mesure n'est possible, fonctionnent et permettent de générer des PA.

4.2.2.2 Le générateur de PA

Le montage

Le schéma du générateur d'impulsion est donné à la figure 4.12. Un signal de commande $cmdAP$ appliqué en entrée permet de générer deux potentiels d'action en sortie : un pré- et un postsynaptique, V_{spk_pre} et V_{spk_post} respectivement. Le générateur d'impulsion envoie également un signal de remise à zéro $resetCmem$ pour décharger la capacité de membrane du neurone lorsque ce dernier est actif⁴.

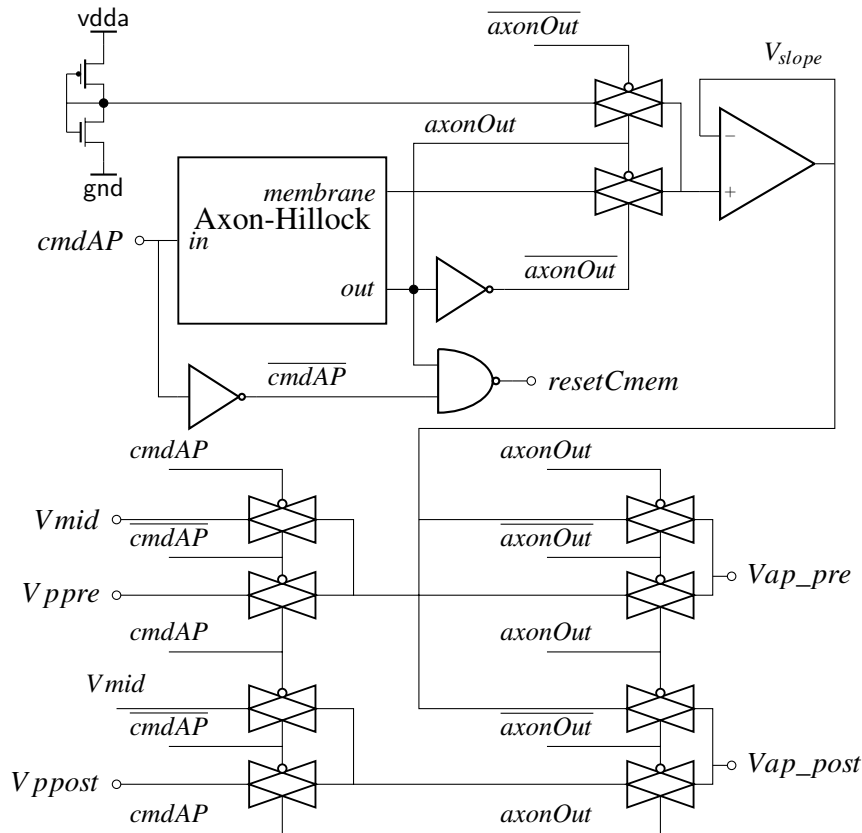


FIGURE 4.12 – Schéma du générateur d'impulsion du neurone.

⁴Un neurone actif est un neurone qui génère un PA.

Les tensions V_{mid} , V_{ppre} et V_{ppost} définissent respectivement le potentiel de repos, et les amplitudes $A_{pulse_{pre}}$ et $A_{pulse_{post}}$. Elles sont générées à l'extérieur de la puce, il est donc possible de les régler. Ces tensions sont aiguillées par des interrupteurs commandés à l'aide d'un circuit *axon-hillock* pour créer les impulsions pré- et postsynaptiques.

Le circuit *axon-hillock*, tiré de Mead (1989), est un neurone IF qui génère une forme de potentiel d'action triangulaire proche de la forme des impulsions que nous souhaitons avoir. Nous avons modifié le circuit d'origine (figure 4.13.a) afin d'obtenir les durées t_{pulse} et t_{LTx} ainsi que la rampe de polarité négative (figure 4.13.b) de l'impulsion choisie au chapitre 2 (figure 2.22).

Lorsque le circuit (figure 4.13.a) est sur son point de repos : V_{out} vaut $vdda$; $cmdAP$ est à $vssa$; la capacité C_{mem} a une tension qui se situe entre $vdda$ et le seuil de basculement des inverseurs (sa valeur exacte n'est pas importante dans notre cas).

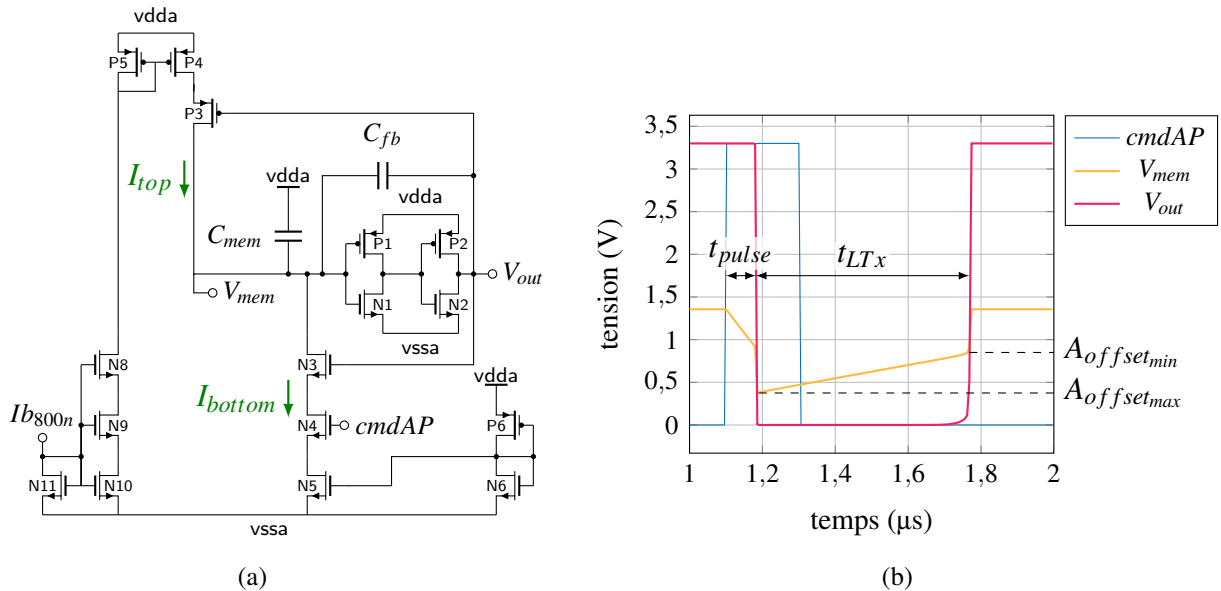


FIGURE 4.13 – Circuit *axon-hillock* de Mead (1989) modifié. (a) Schéma transistor de l'*axon-hillock* implémenté sur la puce. (b) Simulation Spectre® de l'*axon-hillock*.

Lorsqu'un pulse d'amplitude $vdda$ arrivant par l'entrée $cmdAP$ est reçu sur la grille du NMOS N4, la branche appartenant à ce transistor devient active (voir courbe de simulation à la figure 4.13.b). Le courant I_{bottom} contrôlé par le transistor N5 (polarisé par P6 et N6) décharge la capacité C_{mem} jusqu'à ce que sa tension devienne suffisamment basse pour faire basculer les deux inverseurs (P1-N1 et P2-N2). La sortie des inverseurs V_{out} passe alors de $vdda$ à $vssa$, les charges accumulées dans C_{mem} se répartissent équitablement dans C_{mem} et C_{fb} donc la tension de la capacité C_{mem} chute. Le basculement de V_{out} a coupé le courant déchargeant la capacité C_{mem} , mais il a activé la branche du PMOS P3 reliée à la sortie du deuxième inverseur. Le courant I_{top} est alors intégré dans la capacité C_{mem} . La tension V_{mem} croît linéairement, jusqu'à franchir de nouveau le seuil de basculement des inverseurs. V_{out} passe de $vssa$ à $vdda$, les charges de la capacité C_{fb} sont transférées dans la capacité C_{mem} donc la tension de la capacité C_{mem} augmente (les charges dans C_{fb} n'ayant pas bougé, la tension V_{mem} augmente d'autant qu'elle a diminué

lors du premier transfert de charge). Le nouveau basculement des inverseurs coupe le courant I_{top} venant du PMOS P3. $cmdAP$ étant passée entre temps au niveau bas, plus aucun courant ne vient modifier la valeur de la tension aux bornes de la capacité C_{mem} . Le montage est de nouveau sur son point de repos et attend un pulse arrivant sur $cmdAP$.

Comme nous l'avons précisé dans la partie précédente relative à la membrane, il est nécessaire que la durée du pulse de commande soit supérieure à t_{pulse} pour que la capacité de membrane se décharge suffisamment et franchisse le seuil des inverseurs. Sa durée doit être inférieure à $t_{pulse} + t_{LTx}$, sinon le circuit générerait de nouveau une impulsion.

Un premier calcul nous a permis d'obtenir une approximation des valeurs des amplitudes et des temps que nous voulions avoir pour le PA. En utilisant les équations de charge et de capacité des condensateurs, nous obtenons le jeu d'équations suivant :

$$t_L = \frac{C_{fb} + C_{mem}}{I_{top}} \Delta V_{mem} = \frac{C_{fb}}{I_{top}} V_{dda} \quad (4.1)$$

$$t_H = \frac{C_{fb} + C_{mem}}{I_{bottom} - I_{top}} \Delta V_{mem} = \frac{C_{fb}}{I_{bottom} - I_{top}} V_{dda} \quad (4.2)$$

Avec une contrainte sur les tensions :

$$\Delta V_{mem} = \frac{C_{fb}}{C_{mem} + C_{fb}} \Delta V_{out} \quad (4.3)$$

Après calculs et simulations, nous avons obtenu les caractéristiques suivantes : $I_{top} = 265$ nA, $I_{bottom} = 200$ nA, $C_{fb} = 55$ fF, $C_{mem} = 282$ fF, $\Delta V_{out} = 3,3$ V et $\Delta V_{mem} = 0,88$ V.

Les mesures

Les mesures réalisées sur le bloc test (figure 4.14.a) nous donnent des impulsions pré- et postsynaptiques identiques mais de mauvaise qualité (figure 4.14.b). Les temps mesurés sont plus longs que ceux obtenus en simulation : $t_{pulse} \approx 200$ ns pour $t_{pulse} = 120$ ns en simulation et $t_{LTx} > 1$ μ s pour $t_{LTx} = 590$ ns.

Les écarts dans les valeurs de durée des impulsions entre les simulations et les mesures peuvent s'expliquer par l'utilisation de capacité MIM. Comme noté précédemment, la valeur est donnée par le fondeur avec une certaine approximation, mais cela ne justifie pas entièrement des durées multipliées par un facteur deux.

4.2.2.3 Les neurones de la couche de sortie

Les mesures

Après avoir fait des mesures sur les blocs séparément, nous avons également testé les neurones de la couche de sortie du réseau. Le circuit de test reprend le schéma de la figure 4.1 en remplaçant le memristor par une résistance de test de 1 k Ω . Des mesures ont été réalisées sur les

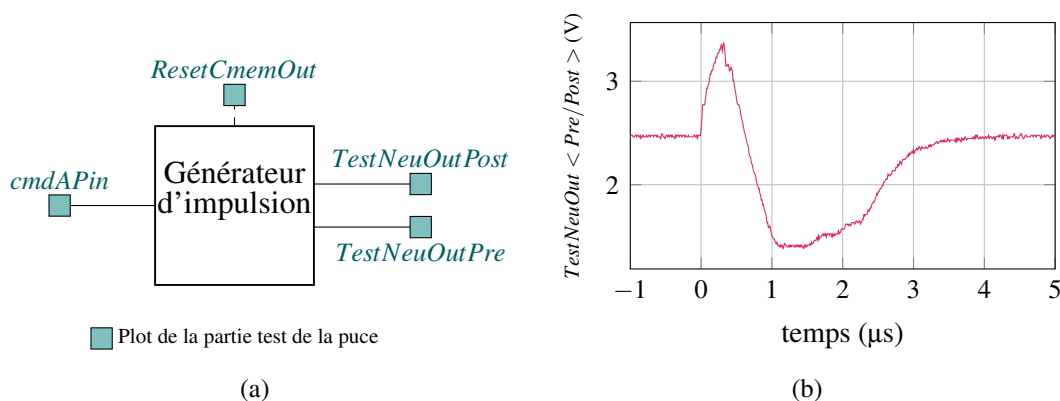


FIGURE 4.14 – Mesures du générateur de PA du bloc test. (a) Schéma des entrées/sorties du générateur d’impulsion du bloc test. (b) Impulsion en sortie du générateur de PA de test.

10 neurones en les stimulant avec des PA générés par un GBF. Toutes les mesures ont donné des résultats similaires illustrés à la figure 4.15 par l’exemple du neurone 2. Seul le neurone 7 n’a généré aucune impulsion lors de nos tests.

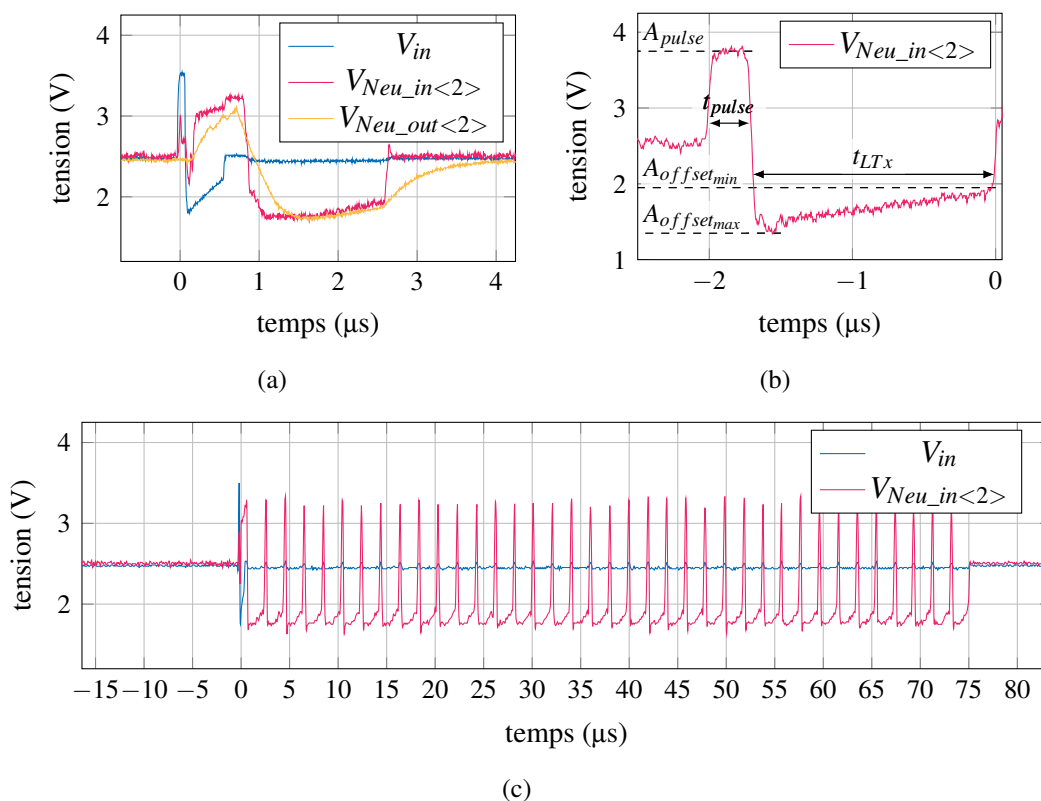


FIGURE 4.15 – Mesures du neurone 2 : réponse $V_{Neu_out<2>}$ du neurone après l’envoi d’un PA chargeant la membrane à travers une résistance de 1 kΩ via le CCII. $V_{Neu_in<2>}$ donne l’image de l’impulsion recopiée par le CCII. La tension de seuil du neurone $V_{Neu_vth<2>}$ est réglée à 4,125 V. (a) Un seul PA est généré en sortie. (b) Un PA seul. (c) Un burst est généré en sortie.

Pour ces mesures, un PA est appliqué sur l’entrée V_{in} et traverse une résistance de 1 kΩ. Le courant résultant de la loi d’Ohm arrive sur la borne X du CCII. Le courant est recopié en Z et

permet de charger la capacité de membrane du neurone. Le potentiel de membrane ayant franchi le seuil $V_{Neu_vth<2>}$ ajusté à 4,125 V, un PA $V_{Neu_out<2>}$ est généré en sortie du neurone. Le potentiel $V_{Neu_out<2>}$ visible à la figure 4.15.a est de mauvaise qualité. Néanmoins sur l'entrée $V_{Neu_in<2>}$ du neurone, la recopie de l'impulsion de sortie sur la borne X du CCI fait apparaître un PA avec des fronts raides : la forme est plus proche de celle attendue. La figure 4.15.b montre un PA isolé, mesuré sur l'entrée $V_{Neu_in<2>}$. Les amplitudes A_{pulse} , A_{offset_max} et A_{offset_min} sont correctes mais les durées sont supérieures à celles qui ont été obtenues en simulation : environ 300 ns pour t_{pulse} au lieu des 120 ns en simulation et 1,7 μ s pour t_{LTx} au lieu de 590 ns.

De plus, les mesures ont révélé un phénomène de *burst*⁵ présent sur tous les neurones fonctionnels (figure 4.15.c). Ce phénomène, non souhaité, peut être dû au monostable qui génèrerait un pulse de commande de durée plus importante que l'impulsion de sortie du neurone.

Durant le test, nous avons observé l'exclusivité d'un neurone à être actif (propriété *Winner-Take-All*, WTA, du réseau). Le neurone qui est stimulé génère au moins un PA. Mais la génération de plusieurs PA, visible à la figure 4.15.c, est un problème. Ce phénomène peut être dû à une mauvaise décharge de la capacité de membrane du neurone ou une mauvaise durée du pulse généré par le monostable de la membrane. Un autre dysfonctionnement a été observé : il arrive que le neurone commence à générer la partie positive du PA (le pulse) mais ne génère pas la partie négative du PA (la rampe). Ce problème peut encore être dû à un mauvais calibrage de la durée du pulse du monostable de la membrane.

4.2.3 Les capteurs d'entrée et la commande numérique

Les capteurs sont des générateurs de PA commandés par le bloc numérique d'entrée (figure 4.16.b). Le bloc numérique de la première puce est composé d'un registre à décalage dont l'entrée est *dataIn* (figure 4.16.a). Les commandes des 81 capteurs sont introduites en série cadencées par l'horloge *clk*. Dès que les données sont entièrement prépositionnées, un bit de validation *vld* est envoyé sur l'entrée d'horloge des bascules D servant de mémoire. Leurs valeurs d'entrée se retrouvent en sortie et permettent de commander les générateurs d'impulsion des capteurs. Lorsqu'un 1 est reçu en entrée du capteur, ce dernier génère une impulsion. Cette commande a été réalisée dans le but d'apprendre des images en noir et blanc (sans niveau de gris).

Nous avons créé un code VHDL permettant de générer les entrées *clk*, *dataIn* et *vld* avec un FPGA connecté à la carte test de la puce. Nous visualisons ces signaux et le comportement du capteur 1, *cmdSen < 1 >* sur la figure 4.17. Les autres capteurs ont été testés et ont un comportement identique au capteur 1.

L'absence de remise à zéro pour les bascules D et l'absence de monostable en entrée des capteurs ne permettent pas un contrôle satisfaisant des générateurs d'impulsion. Dès qu'un 1 est présent à l'entrée d'un capteur, ce dernier devient actif jusqu'à ce qu'il ait un 0 à son entrée. Il génère alors non pas un PA mais un *burst* : ce comportement n'est pas voulu, et il sera corrigé

⁵Succession continue de PA.

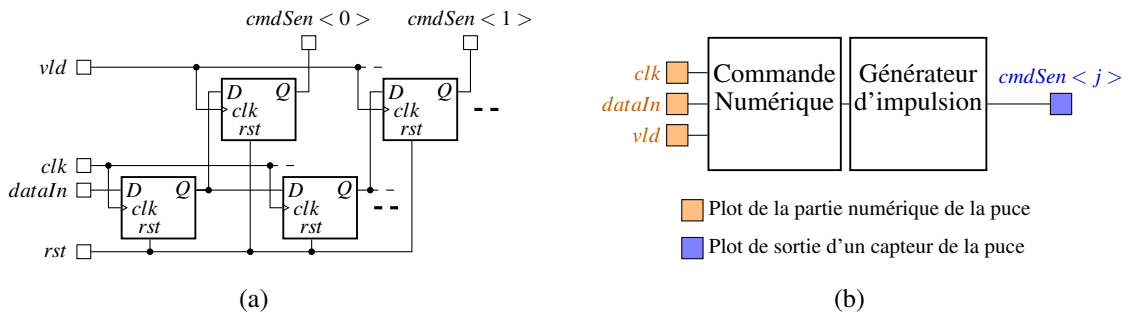


FIGURE 4.16 – (a) Schéma du registre à bascules D de la partie numérique de la puce. (b) Schéma des entrées/sorties de la partie numérique et des capteurs de la couche d'entrée du réseau.

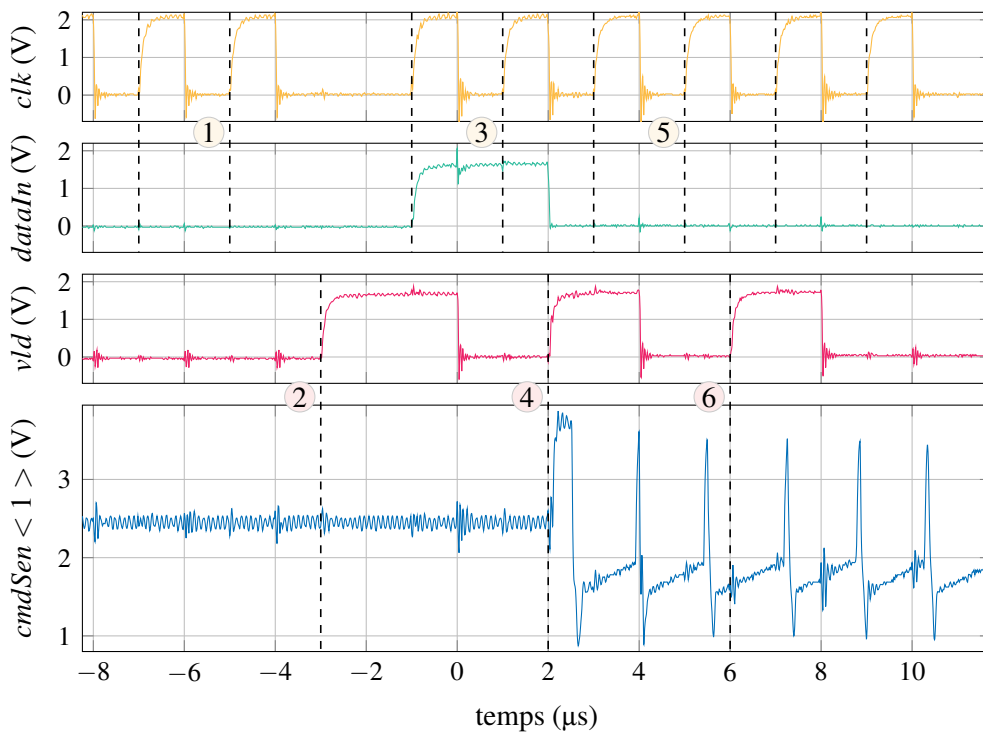


FIGURE 4.17 – Mesures réalisées sur le capteur 1. (1) Fin d'une série de 81 coups d'horloge clk avec $dataIn$ à 0. (2) Le front montant de vld permet d'appliquer les 0 sur l'entrée des capteurs. (3) Entrée d'un 1 sur la première bascule D du registre. (4) Le 1 est appliqué sur le premier capteur qui devient actif et commence à générer un PA ($cmdSen <1>$). (5) Entrée d'un 0 sur la première bascule D du registre. (6) Le 0 est appliqué sur le premier capteur, mais le capteur ne le prend pas en compte et continue de générer des PA.

dans la seconde version de la puce. De plus, les capteurs peuvent être contrôlés à l'activation mais pas à la désactivation : l'envoi de 0 sur un capteur actif ne change pas son comportement et il a été observé un temps de latence d'une durée allant de 200 ms jusqu'à 1 s.

Nous venons de voir que la première puce n'a pas donné entière satisfaction lors des mesures. Il apparaît qu'il y a un problème d'adaptation avec les plots de sortie : les PA ont une forme correcte lorsqu'ils sont mesurés par l'intermédiaire du CCII, le monostable de test semble ne pas générer de pulse alors que les neurones donnent une réponse suite à la stimulation d'un

courant. Nous avons aussi constaté que les durées des PA étaient supérieures à celles obtenues en simulation. La cause probable est l'utilisation de capacités qui ont une précision relative de 15 %. Néanmoins l'architecture globale du système n'est pas remise en cause et dans la conception de la seconde puce nous avons pris en compte les défauts de la première mouture.

4.3 Les améliorations et simulations de la seconde puce

Entre les deux versions de la puce, des simulations plus précises du réseau de neurones ont été réalisées et ont permis d'apporter des modifications nécessaires à l'implémentation du système. Nous allons donc détailler les améliorations apportées et présenter les simulations transistors réalisées avant la fabrication de la seconde puce. Ce second circuit intégré doit nous être retourné début septembre.

4.3.1 La partie analogique

La partie analogique comprend les générateurs d'impulsion des capteurs et les neurones de la couche de sortie. La première version de la puce a été conçue pour répondre aux caractéristiques d'une technologie particulière de memristor (BTO) générant des impulsions aux caractéristiques figées. Dans la conception de la seconde puce, nous avons décidé de rendre le potentiel d'action entièrement paramétrable (durées et amplitudes) afin d'avoir une souplesse d'utilisation. La puce est compatible avec les valeurs de seuils plus faibles (pour un pulse appliqué de 100 ns) du memristor BFO.

4.3.1.1 Le nouveau générateur d'impulsion

Le nouveau générateur d'impulsion (figure 4.18) garde la même base que le précédent : ce sont des interrupteurs commandés par un circuit *axon-hillock* qui permettent de créer l'impulsion voulue.

Nous avons remanié le circuit *axon-hillock* de la première version en remplaçant le premier des deux inverseurs (réalisé par le couple P1-N1 sur la figure 4.13.a) par un comparateur à hystérésis. De plus, nous avons rendu les courants I_{top} et I_{bottom} paramétrables de l'extérieur. Avec ces modifications, il devient possible de contrôler la valeur des amplitudes $A_{offset,max}$, $A_{offset,min}$ et des durées t_{pulse} et t_{LTx} .

Afin d'éviter les *bursts* de PA observés lors des mesures, nous avons ajouté une bascule RSH personnalisée. Sa table de vérité est donnée dans le tableau 4.1 : la mise à 1 est synchrone alors que la mise à 0 est asynchrone.

Au repos, la sortie Q de la bascule RSH est à 0 et un niveau haut est appliqué sur son l'entrée S par la sortie *out* de l'*axon-hillock*. La sortie de la bascule passe à 1 lorsqu'un front montant arrive sur son entrée d'horloge. Puis la bascule est remise à 0 lorsque le signal *refractory* (ou *rst*) passe à 1. Le signal *refractory* provient d'un monostable commandé par le passage de 1 à 0

TABLE 4.1 – Table de vérité de la bascule RSH.

<i>clk</i>	<i>S</i>	<i>R</i>	<i>Q</i>	\overline{Q}
–	X	0	0	1
–	X	1	Q_{n-1}	$\overline{Q_{n-1}}$
↑	0	0	Q_{n-1}	$\overline{Q_{n-1}}$
↑	1	0	1	0

de la sortie de l'*axon-hillock*. Avec la bascule RSH, il n'est donc plus nécessaire d'utiliser le monostable de la membrane. Dans la seconde version de la puce, la membrane est donc réduite au comparateur avec la capacité, la résistance de fuite et l'interrupteur de décharge en parallèle.

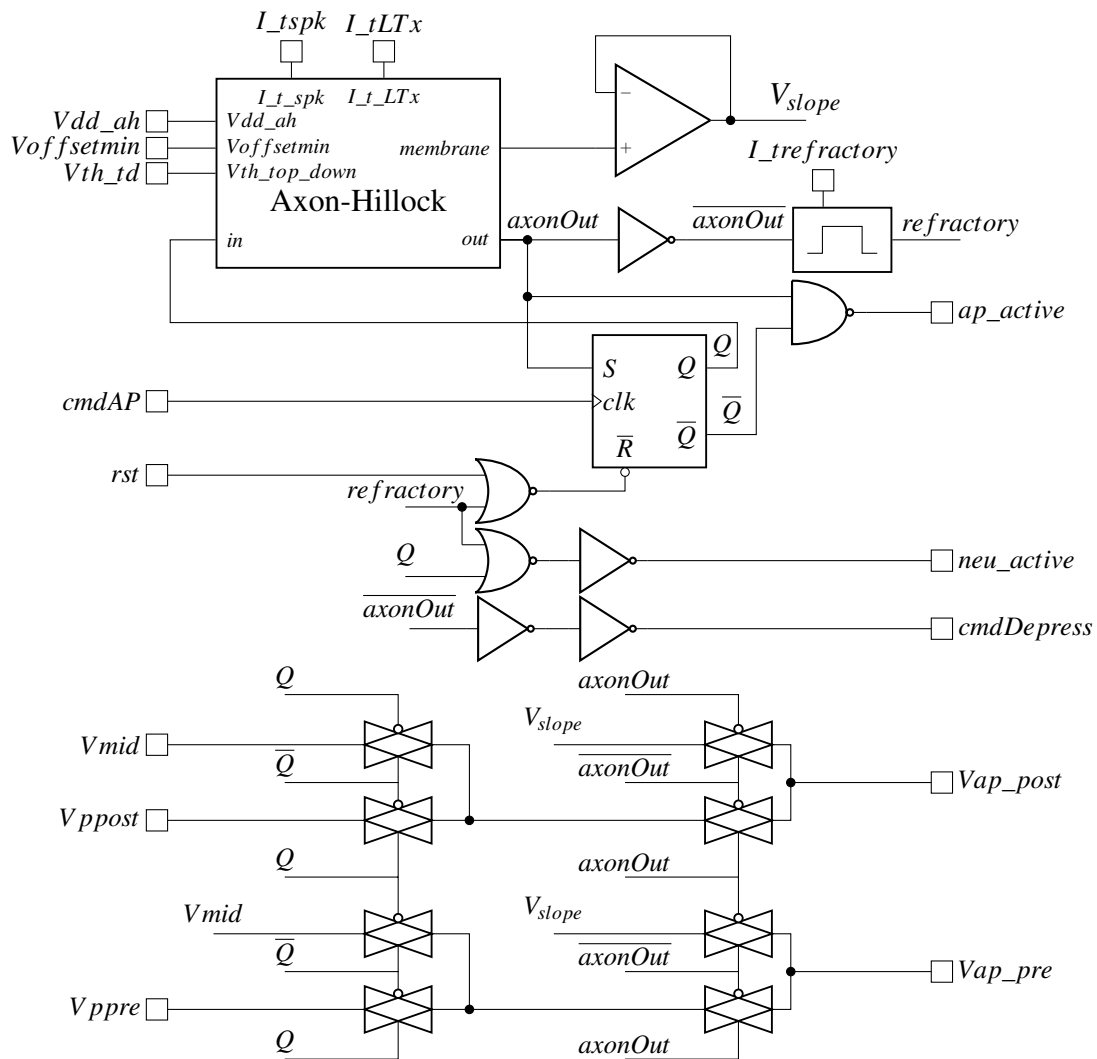


FIGURE 4.18 – Schéma du générateur d'impulsion du neurone de la seconde puce.

Nous avons également ajouté et modifié certaines propriétés du neurone. En effet, dans la première version de la puce nous avons utilisé un modèle de neurone LIF simplifié avec une période réfractaire égale au temps du PA. Or comme nous l'avons vu dans le chapitre précédent, des simulations plus précises du réseau ont montré qu'il est nécessaire que les neurones aient une

période réfractaire plus longue et au moins égale à deux fois la durée d'une impulsion⁶. Nous avons donc décidé d'ajouter un monostable ajustable par un courant extérieur $I_{refractory}$ qui autorise une modification de la durée de la période réfractaire du neurone représentée par le signal *refractory*. Une période réfractaire a aussi été ajoutée aux capteurs, elle est modifiable et peut être désactivée. Les durées des périodes réfractaires des neurones et des capteurs peuvent être différentes : elles sont ajustables par deux courants différents.

Avec cette modification, une sortie a été créée : la commande de dépréciation *cmdDepress*. Cette commande est connectée aux capteurs et a la possibilité d'être désactivée. Elle permet au neurone devenu actif de commander la génération de PA aux capteurs qui ne sont pas actifs à cet instant. Ce PA étant créé pendant l'impulsion du neurone, les conductances des synapses connectées entre les capteurs nouvellement actifs et le neurone actif sont alors dépréciées.

4.3.1.2 Le nouveau convoyeur de courant

Le convoyeur de courant de la première version répond à nos besoins techniques mais il a été conçu avec un niveau d'alimentation différent des autres éléments de la puce. Pour la seconde version, nous avons conçu un CCII qui reprend la même architecture que précédemment (un amplificateur monté en suiveur cascadé avec un étage suiveur de courant) et qui possède un niveau d'alimentation identique au reste de la puce.

Avec la possibilité d'avoir des impulsions modifiables et pour que le convoyeur de courant ne soit pas l'élément limitant, il devient nécessaire d'avoir un amplificateur en tension *rail-to-rail* en entrée et en sortie (figure 4.19.a). Cette spécification est également nécessaire pour l'étage suiveur de courant. L'amplificateur opérationnel, visible dans la figure 4.19.a, est composé d'un double étage différentiel à charge cascodée repliée en entrée, et d'un étage de sortie de classe AB avec une capacité Miller scindée en deux (Hogervorst *et al.*, 1994). Nous avons implémenté l'étage de recopie en courant de classe AB proposé par Zeki et Kuntman (2000) et illustré à la figure 4.19.b. La sortie de la grille des transistors de courant est asservie à un amplificateur réalisé par un étage différentiel.

Bien que les deux étages soient *rail-to-rail*, une fois cascadé, l'étage de recopie en courant limite la tension lorsqu'une impulsion apparaît sur la borne Y : des saturations du signal sont alors visibles sur la borne X. Dans le cadre du réseau de neurones que nous implémentons, l'utilisation des deux propriétés du CCII ne se font pas au même instant :

- la recopie en tension est nécessaire lorsque le neurone devient actif et donc crée son impulsion ;
- la recopie en courant a lieu lorsque le neurone est inactif.

⁶Ceci permet d'éviter qu'un neurone présynaptique généré pour déprécier la valeur de la connexion synaptique charge la membrane du neurone, par exemple.

Nous avons donc décidé d'insérer des interrupteurs qui désactivent l'étage de recopie en courant lorsque le neurone devient actif (figure 4.20). Ainsi l'amplificateur est seul en suiveur et il n'y a plus de saturation du signal recopié.

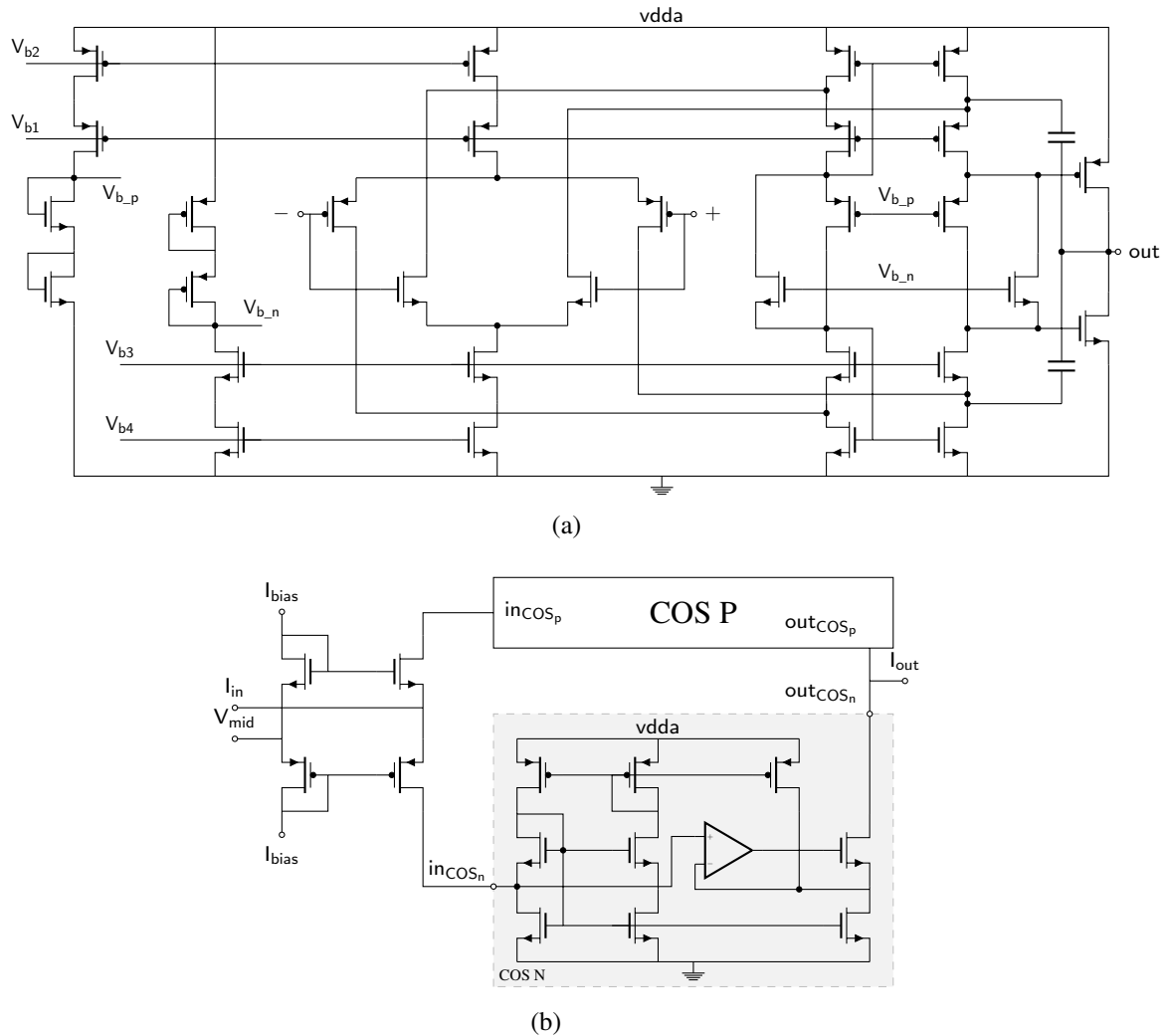


FIGURE 4.19 – Schémas transistors des blocs composant les CCII. (a) L'amplificateur opérationnel du CCII. (b) L'étage suiveur de courant du CCII. Il est composé de deux blocs complémentaires de recopie de courant COS N et COS P (COS de l'anglais *Current Output Stage*).

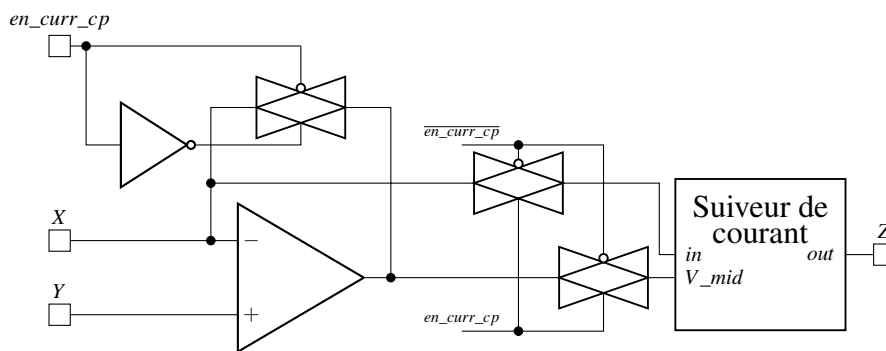


FIGURE 4.20 – Schéma du convoyeur de courant de la seconde puce.

4.3.1.3 Le circuit de décision

Bien que nous n'ayons pas observé de réponse simultanée de deux neurones durant les mesures, nous avons ajouté un bloc de décision entre les membranes des neurones et leurs générateurs. Le bloc de décision, ou circuit *Winner-Take-All* (WTA), a pour entrée les signaux de dépassement des 10 membranes des neurones du réseau et il possède 10 sorties reliées aux entrées des générateurs de PA des neurones. Avec l'exclusivité de sa sortie, ce bloc assure qu'il n'y ait qu'un seul générateur de PA qui soit activé. Dans l'architecture précédente, en cas de dépassement de seuil simultané au niveau de deux membranes de deux neurones, il n'était pas possible de garantir l'exclusivité d'activation d'un neurone. Néanmoins, avec le circuit WTA, si deux neurones sont actifs avec une nanoseconde d'écart, la commande donnée par le bloc de décision peut être erronée : le bloc peut activer un neurone tiers qui n'a pas eu de dépassement au niveau de sa membrane. Toutefois, un seul neurone aura été actif. Le circuit WTA est remis à zéro lorsque le neurone activé a terminé de générer son impulsion.

4.3.1.4 Le circuit et la simulation de la partie analogique

La figure 4.21 présente toute la partie analogique composée de l'ensemble des blocs détaillés précédemment et connectée avec le memristor extérieur à la puce. La simulation, dont le détail est dans figure 4.22 et sa légende, montre le résultat d'un neurone N_3 stimulé par deux capteurs d'entrée S_1 et S_2 . Des memristors modélisés avec un code Veriloga ont pris la place des synapses pour connecter les deux capteurs au neurone.

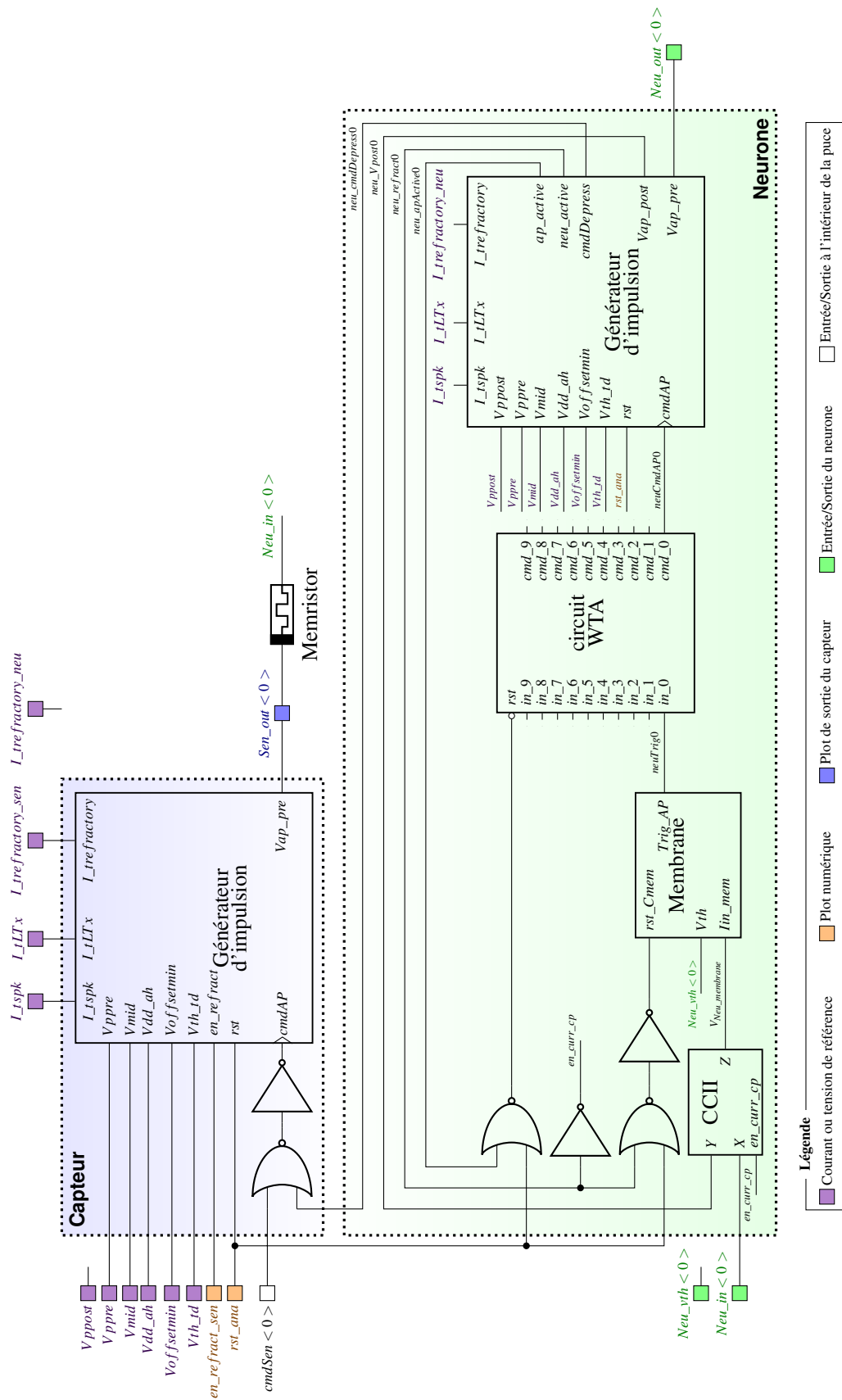


FIGURE 4.21 – Schéma de la partie analogique de la seconde puce.

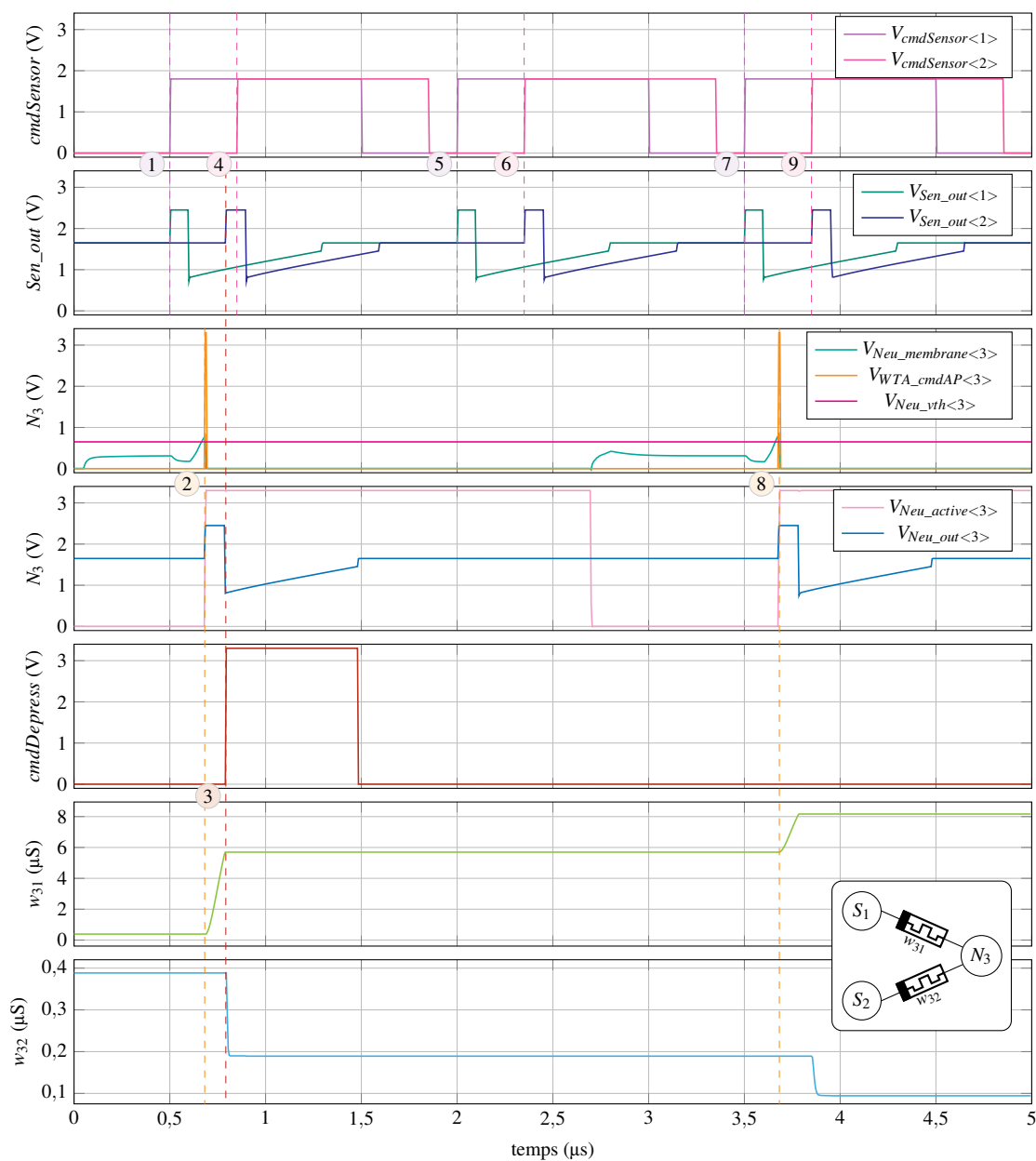


FIGURE 4.22 – Simulation de deux capteurs (S_1 et S_2) connectés par des memristors à un neurone (N_3). À $3,2 \mu\text{s}$ le signal en_rfrct_sen est désactivé : le signal $cmdDepress$ n'est alors plus généré. (1) Une commande est envoyée sur S_1 qui génère un PA. (2) Le PA a chargé la capacité de membrane du neurone N_3 jusqu'au seuil $V_{Neu_vth<3>}$ ajusté à $0,65 \text{ V}$. N_3 génère alors à son tour un PA, la différence de potentiel aux bornes du memristor w_{31} devient suffisamment grande pour augmenter sa conductance. (3) Le signal de commande $cmdDepress$ venant de N_3 est envoyé aux capteurs. Le capteur non-actif S_2 génère alors un PA, la valeur de la conductance du memristor entre S_2 et N_3 est alors dépréciée. (4) Une commande est envoyée sur S_2 qui est déjà actif, donc rien ne change. (5) et (6) Des PA sont générés par S_1 puis S_2 , aucune charge de la membrane N_3 n'est constatée car le neurone est considéré comme actif : $Neu_active < 3 >$ est à 1. (7) et (8) sont similaires à (1) et (2). en_rfrct_sen étant désactivé N_3 ne génère pas de signal $cmdDepress$. (9) C'est donc par la commande $cmdSen < 1 >$ qu'est généré le PA de S_2 . Le PA de S_2 est arrivé après celui de N_3 donc pour une seconde fois, w_{32} est dépréciée.

4.3.2 La partie numérique

Les simulations du chapitre précédent ont montré qu’il était nécessaire d’introduire un bruit dans le temps de présentation des images. Avec le convertisseur série-parallèle d’entrée de la première puce, réalisé avec des bascules D, l’image ne peut être présentée qu’entièrement : l’information est prépositionnée par la première série de bascules D, puis elle est envoyée aux capteurs par l’intermédiaire de la deuxième série de bascules D. La présentation d’une nouvelle image ne peut avoir lieu que 810 ns plus tard, pour une fréquence d’horloge de 100 MHz. Il n’est donc pas possible d’introduire du bruit dans la présentation des images.

Afin de pouvoir introduire du bruit dans la présentation, nous avons changé le bloc numérique de la puce. De plus, nous avons modifié la façon d’apprendre du réseau en ne se limitant pas qu’aux images en noir et blanc, mais en utilisant des images en niveau de gris. Le niveau de gris est alors transposé en fréquence. Pour une image sur fond noir, plus le pixel est blanc, plus grande sera la fréquence des PA codant le pixel.

La figure 4.23 présente la partie numérique de la seconde puce, une mémoire y est incorporée. Cette dernière conserve les temps de déclenchement entre les PA pour chacun des 81 pixels. Il est possible de sauvegarder 10 évènements de PA par pixel. Les temps sont enregistrés avec une précision de 9 bits, soit la possibilité d’un déclenchement entre 10 ns et 5,12 µs pour une horloge de 100 MHz.

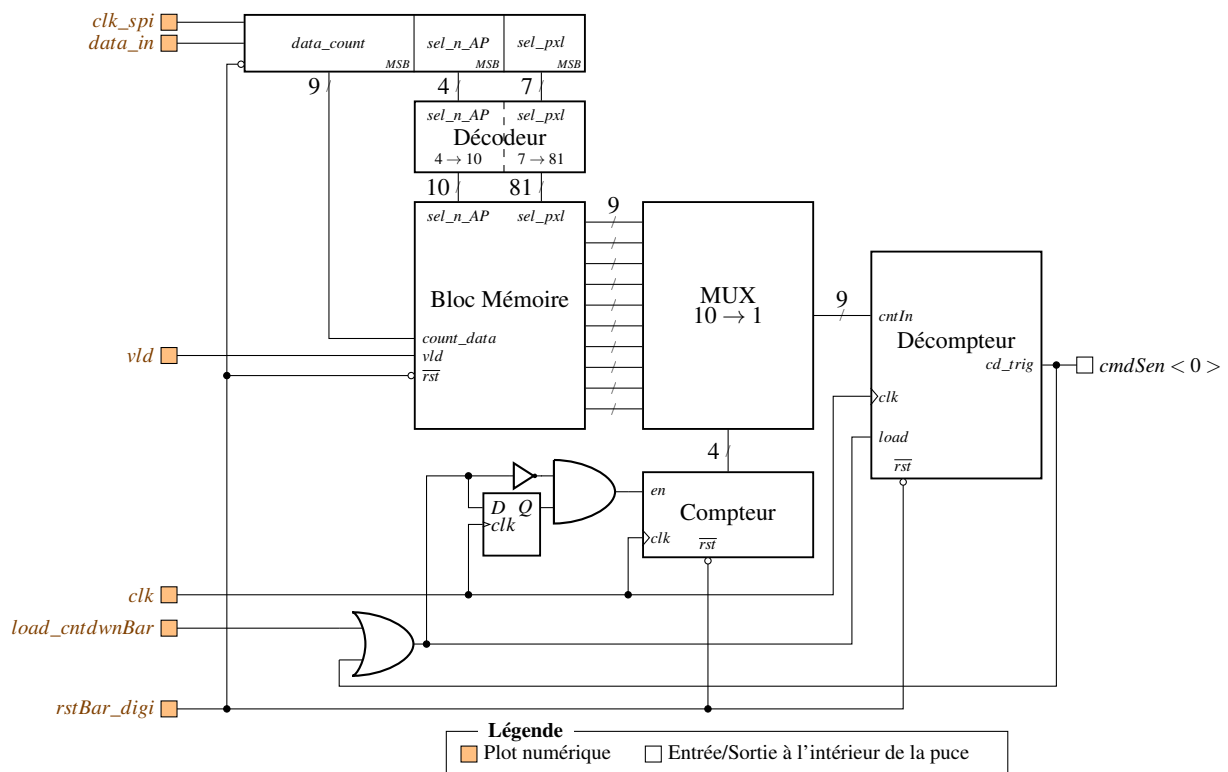


FIGURE 4.23 – Schéma de la partie numérique de la seconde puce.

La technologie 0,18 μm d'AMS possède une bibliothèque d'éléments numériques alimentés sous 1,8 V. Le pilotage de ces entrées se fait par un **FPGA** dont les sorties sont en 0 V/3,3 V. Des montages d'ajustement de tension sont donc présents pour les entrées numériques de la puce afin de faire la conversion 3,3 V vers 1,8 V.

Les simulations du bloc numérique ont été réalisées avec un simulateur Spectre-Verilog. La figure 4.24 montre le résultat d'une simulation qui consiste à rentrer dans la mémoire de la puce quatre évènements pour le capteur 1 puis de les faire générer par ce dernier pour l'apprentissage. Dès que tous les évènements sont sauvegardés (en les mémorisant par l'envoi d'un 1 sur *vld*), l'entrée *load_cntdwnBar* est passée à 1 : le premier évènement est alors chargé dans le décompteur. Lorsque *load_cntdwnBar* passe à 0 l'apprentissage commence. Le décompte démarre jusqu'à atteindre 0. Au passage de 1 à 0 un signal est généré en sortie du compteur pour charger la valeur suivante. Le multiplexeur servant d'aiguillage est piloté par un compteur dont la valeur est incrémentée un coup d'horloge après le chargement des décompteurs. Si aucun évènement n'est mémorisé ou si le compteur a atteint 10, le décompteur reste à 0.

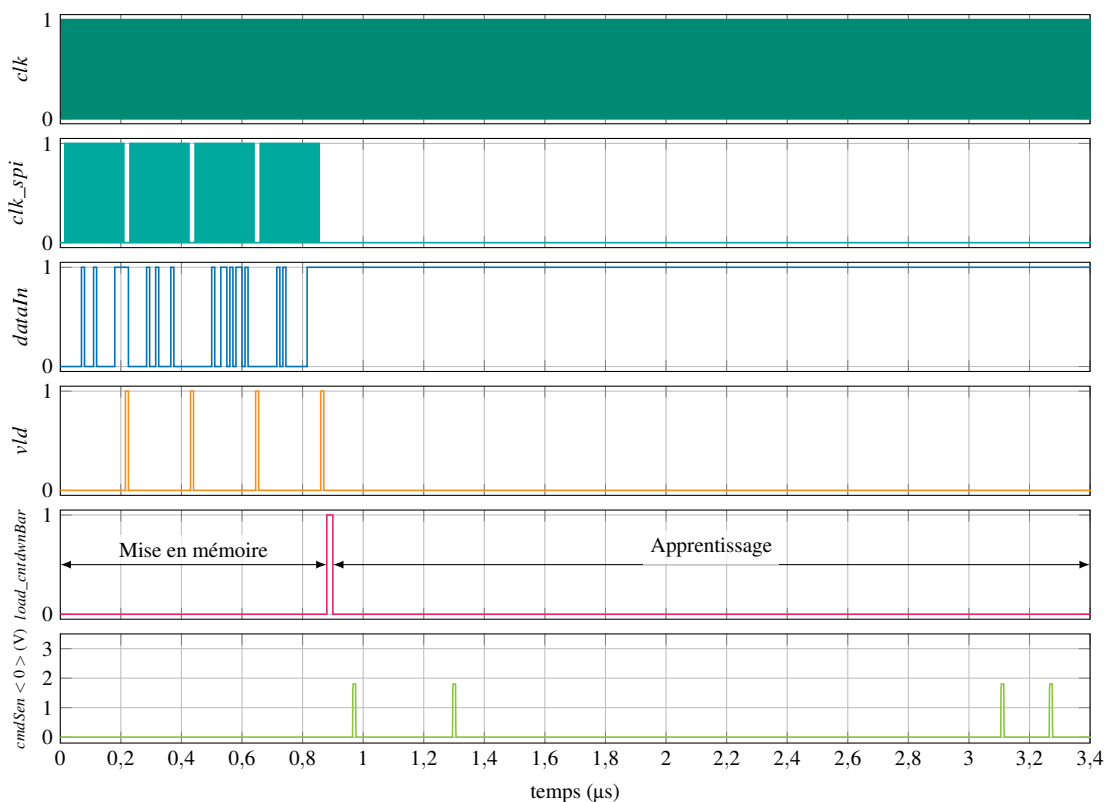


FIGURE 4.24 – Simulation Spectre-Verilog de la seconde puce **SpANNWiTA**.

4.3.3 Le dessin de la seconde puce

Le dessin de la puce est présenté à la figure 4.25. En plus des deux couches du réseau, la puce contient des blocs tests : un **CCII** avec deux sorties en courant (une positive et une négative) et un générateur d'impulsion. Le bloc numérique de la puce *full-custom* a été placé et routé par

le Circuits Multi-Projets (CMP) avec les outils cadence. Partie à la fonderie fin mai 2014, la puce est attendue pour début septembre.

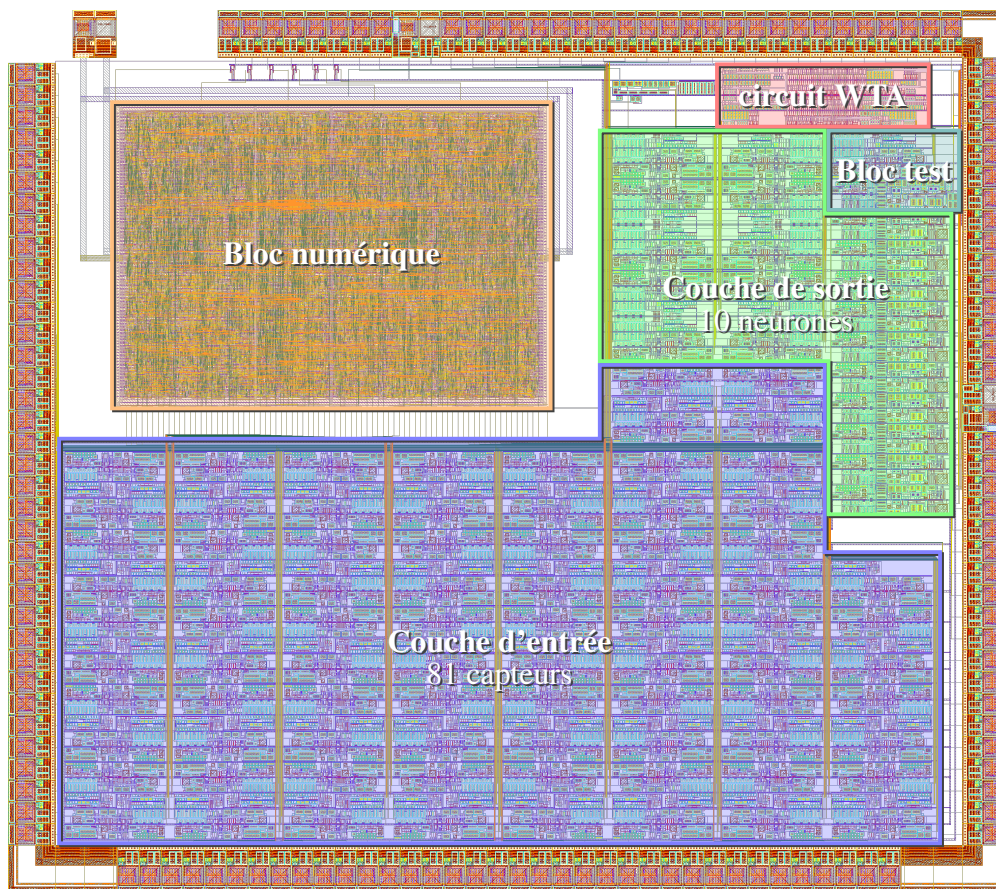


FIGURE 4.25 – Dessin des masques de la seconde puce **SpANNWiTA** (144 plots, surface = 10,535 mm²).

4.4 Conclusion

Le projet **MHANN** s'est penché sur la manière de mettre en œuvre des synapses memristives ferroélectriques avec de l'électronique classique.

Nous avons proposé une solution de montage qui a été brevetée afin d'utiliser un memristor comme une synapse avec des neurones impulsionnels. Grâce aux propriétés du convoyeur de courant de seconde génération placé entre le neurone et la synapse, cette dernière pondère l'information reçue en amont et charge la membrane du neurone en aval. Avec l'utilisation d'une forme de **PA** adéquate, le **CCII** permet aussi d'utiliser la plasticité du memristor pour implémenter une loi de **STDP** choisie : les modifications de résistance se font directement grâce aux **PA** des neurones.

Nous avons réalisé une première puce **SpANNWiTA** contenant une couche d'entrée (81 capteurs) et une couche de sortie (10 neurones) du réseau de neurones impulsionnels. Les mesures de cette puce dédiée aux memristors **BTO** révèlent des imperfections pouvant ralentir

l'apprentissage du réseau. De plus, la forme figée des impulsions ne permet pas d'utiliser cette première puce avec différentes technologies de memristor.

Une seconde puce s'enrichissant des erreurs de la première a été conçue et est en cours de fabrication. Sur cette nouvelle puce, les PA générés aussi bien par les neurones que par les capteurs sont entièrement configurables (en amplitudes et en durées). De plus, les générateurs d'impulsion et le bloc numérique d'entrée ont été complexifiés pour répondre à des exigences apportées par des simulations informatiques. Pour cette seconde version de SpANNWiTA, seules des simulations sont présentées car la puce reviendra de fabrication début septembre.

Les puces conçues sur silicium vont permettre de piloter des memristors. À ce jour, seules quelques mesures ont été réalisées avec un memristor et un convoyeur de courant (Lecerf *et al.*, 2014a) et ont donné satisfaction. Il est prévu d'utiliser d'autres dispositifs memristifs pour confirmer les premières mesures. Un *crossbar* de memristors BFO est en cours de réalisation par le laboratoire UMφ CNRS-Thalès. Il sera connecté par l'intermédiaire d'une carte test avec la puce SpANNWiTA pour créer le réseau de neurones du projet MHANN.

CONCLUSIONS & PERSPECTIVES



CETTE thèse conclut les trois premières années du projet collaboratif *Memristive Hardware Analog Neural Network (MHANN)* qui a permis de réunir des personnes du monde de l'informatique, de l'électronique et du matériau pour mettre au point un démonstrateur en montrant la faisabilité d'un réseau de neurones impulsionnels avec des synapses memristives ferroélectriques. De plus, en s'appuyant sur l'ensemble des propriétés des dispositifs ferroélectriques (la rapidité de modification de résistance, la faible consommation, la mémoire et la plasticité intrinsèque), le projet *MHANN* souhaite mettre en évidence l'avantage d'utiliser des réseaux de neurones hybrides en tant que co-processeurs.

Synthèse

Comme nous l'avons souligné dans le chapitre 1, les chercheurs en ingénierie neuromorphique tentent d'imiter le cerveau pour mieux le comprendre mais aussi pour créer des architectures de calculs plus rapides et moins consommatrices en énergie. Nous avons vu que l'encéphale contient des neurones qui communiquent entre eux grâce à des potentiels d'action par l'intermédiaire de fentes synaptiques. Avant de pouvoir créer des réseaux de neurones complexes, il est nécessaire de modéliser le comportement du neurone et de la synapse, respectivement unité de décision et élément mémoire. La synapse est dotée d'une plasticité : en évoluant en fonction des événements neuronaux, les valeurs des poids synaptiques vont moduler les réponses d'un réseau à un échantillon d'entrée. Nous avons notamment montré qu'il existe une loi de *Spike-Timing-Depend Plasticity (STDP)* directement inspirée d'observations biologiques.

L'amélioration de l'implémentation du neurone sur silicium a permis de réduire sa surface et sa consommation. Bien que les implémentations électroniques des synapses aient aussi été optimisées, il reste difficile de condenser ses rôles de plasticité, mémoire et pondération. Une première réponse a été apportée avec les transistors à grilles flottantes mais un nouveau composant dont la taille ne dépend pas des contraintes de la microélectronique classique a fait son apparition. En effet, le memristor ne possède pas de grille, c'est un empilement de matériaux qui forme un dipôle de taille nanoscopique.

Le chapitre 2 nous a permis de faire un tour d'horizon des mesures de formes de *STDP* réalisées avec différentes technologies de memristor. Certaines d'entre elles ne sont pas compatibles directement avec les technologies de la microélectronique à cause de leurs niveaux de tension élevés. La majorité d'entre elles a des temps de modification similaires aux temps biologiques de l'ordre de la milliseconde. Ces technologies ne sont pas destinées, à court terme, à être utilisées dans des systèmes de calculs mais plutôt dans la recherche à l'imitation du comportement biologique du système nerveux. En revanche, la technologie des memristors ferroélectriques de l'UMφ CNRS/Thalès allie la rapidité de variation et les faibles niveaux de tension. Nos mesures réalisées sur des memristors à barrière ferroélectrique BiFeO_3 (*BFO*) montrent qu'il est possible, en appliquant un potentiel d'action (*PA*) dédié, d'obtenir des formes de *STDP* comparables à celles observées en biologie mais avec des temps de l'ordre de la centaine de nanosecondes.

Après avoir introduit différentes architectures de réseaux de neurones dédiées à la classification dans le chapitre 3, nous avons présenté l'architecture du réseau du projet **MHANN** qui utilise des neurones impulsionnels et une loi de **STDP** bornée dans le temps. Composé d'une couche d'entrée de 81 capteurs et d'une couche de sortie de 10 neurones, le réseau utilise une architecture compétitive avec une connexion *all-to-all* entre les deux couches et un apprentissage non-supervisé. L'utilisation originale de périodes réfractaires à la fois sur les neurones de sortie et sur les capteurs d'entrée nous permet d'employer des **STDP** « classiques » pour l'apprentissage.

Nous avons modélisé les mesures réalisées sur les memristors **BFO** afin de les inclure dans un simulateur et ainsi connaître les performances de notre architecture avec des dispositifs memristifs. Les simulations montrent des résultats corrects (88,8% de reconnaissance avec 500 neurones) sans toutefois rivaliser avec les architectures plus classiques à apprentissage supervisé. Néanmoins, nous sommes capables de donner les limites de disparité des memristors inclus dans un *crossbar* pour garder un taux d'apprentissage acceptable.

Dans le chapitre 4, nous avons détaillé les puces silicium qui ont été conçues durant ces trois années de thèse. L'implémentation électronique du réseau repose sur l'utilisation d'un convoyeur de courant de seconde génération (**CCII**). Il permet de connecter un neurone impulsionnel à des synapses memristives qui sont utilisées pour leur plasticité intrinsèque, pour leur capacité à garder en mémoire leurs changements et pour pondérer les signaux en amont. Cette solution simple permet d'utiliser pleinement le memristor ; elle fait l'objet d'un brevet français en vue d'une internationalisation. De plus, le système memristor - convoyeur de courant - neurone forme un motif simple qui, dans l'absolu, est répétable à l'infini pour une construction de réseau à grande échelle.

Les mesures que nous avons réalisées sur la première puce **SpANNWiTA** n'ont pas donné entière satisfaction mais ont permis de valider une partie de l'architecture. En s'appuyant sur ces mesures, nous avons conçu une seconde puce. Seules des simulations ont été présentées car cette puce est actuellement en cours de fabrication. Couplée à un *crossbar* de memristors, elle permettra de construire le prototype du projet **MHANN**.

Durant cette thèse réalisée en 3 ans, nous avons mis en place une architecture de réseau de neurones et réalisé un montage électronique qui nous permettra de piloter les memristors ferroélectriques **BFO**. Le travail se poursuivra durant la dernière année du projet **MHANN** afin de montrer le bon fonctionnement d'un réseau de neurones à synapses memristives ferroélectriques.

Perspectives

Du projet MHANN

Les premières mesures obtenues en connectant un convoyeur de courant avec un memristor sont de bon augure ([Lecerf et al., 2014a](#)). Mais il faut confirmer ces résultats sur d'autres memristors [BFO](#). Il est également nécessaire de continuer à faire des mesures de [STDP](#) sur différents dispositifs [BFO](#) afin d'alimenter la base de données du simulateur et d'en améliorer les performances.

La modélisation du memristor en général et celle du memristor ferroélectrique en particulier est un point clé pour l'avancement des recherches avec ces technologies. Alors que le premier memristor à barrière ferroélectrique BaTiO_3 ([BTO](#)) de l'UM ϕ avait un modèle qui permettait de faire des simulations avec des impulsions, la seconde technologie, le memristor [BFO](#) qui a été perfectionné durant le projet [MHANN](#), ne possède pas encore de modèle. Il sera important pour nous de développer un modèle précis qui nous permettra de réaliser des simulations pour aider à son implémentation dans des systèmes complexes.

Le réseau de neurones du projet [MHANN](#) ne se fera pas sans le *crossbar* de memristors qui est notre prochain challenge. La dernière année du projet [MHANN](#) sera consacrée à la caractérisation et la mise en place de ce *crossbar* avec la seconde puce [SpANNWiTA](#).

Des réseaux de neurones à synapses memristives

Le XXI^e siècle est qualifié de « siècle sur le cerveau ». Le projet européen *Human Brain Project* (HBP) et l'initiative *Brain Research through Advancing Innovative Neurotechnologies* (BRAIN) des États-Unis d'Amérique montrent l'intérêt grandissant pour la recherche sur le cerveau. L'ingénierie neuromorphique peut être un moyen de faire des avancées vers la compréhension du cerveau. Mais avant cela, ne serait-il pas nécessaire de comprendre encore mieux les réseaux de neurones impulsionsnels qui ont longtemps été délaissés ?

L'une des grandes difficultés des réseaux de neurones impulsionsnels est que les différents paramètres qui permettent son fonctionnement (taille du réseau, agencement des neurones, valeur de la capacité de membrane, intensité des courants, période d'inhibition, période réfractaire, période et valeur d'homéostasie, *etc.*) sont tous corrélés et ont tous une influence les uns sur les autres. Il reste encore difficile de donner l'impact exact de chacun d'entre eux. De plus, l'utilisation de lois locales pour un apprentissage global n'est pas une évidence : l'apprentissage des réseaux de neurones avec une [STDP](#) est encore mal maîtrisé. Les neurones ont longtemps été au cœur des recherches en ingénierie neuromorphique et la lumière reste à faire sur les synapses et leur mécanisme de plasticité.

La technologie des memristors ouvre de nouvelles perspectives pour les réseaux de neurones. Mais la physique des composants memristifs n'est pas entièrement élucidée. Il est important de

continuer les recherches sur ces composants du futur car ils ont un fort potentiel. Les industriels de la microélectronique l'ont bien compris et beaucoup d'entre eux ont investi pour mettre au point leur propre technologie. Leur but immédiat n'est pas d'en faire des synapses mais des mémoires binaires. L'investissement réalisé permet une meilleure maîtrise des *process* de fabrication et une meilleure fiabilité des composants. Néanmoins pour la mémorisation analogique, il n'en est pas de même, il subsiste encore des problèmes de fiabilité et de rétention de l'information.

Il reste beaucoup à faire tant au niveau des technologies memristives que pour les réseaux de neurones impulsionsnels. Mais les systèmes hybrides peuvent apporter une nouvelle révolution des systèmes électroniques tant en terme de consommation, de surface que de puissance de calcul.

ANNEXE A

**D'AUTRES MESURES RÉALISÉES AVEC LE
MEMRISTOR BFO**

DANS le chapitre 2 sont présentées les mesures de formes de STDP que nous avons réalisées avec le memristor BFO de l'équipe du D^r Julie Grollier de l'UMφ CNRS/Thalès. Dans cette annexe, nous exposons d'autres mesures de formes de STDP faites avec ces même memristors. Mais tout d'abord nous revenons sur le comportement en hystérésis du memristor.

Un petit retour sur l'hystérésis

La figure A.1 présente des mesures similaires à la figure 2.16 mais sous la forme $i-v$ (dans l'encart de la figure A.1 l'hystérésis est présenté sous sa forme $R = f(v)$). En effet, c'est la courbe $i = f(v)$ qui caractérise le comportement d'un composant. Chua a montré en 1976 qu'un memristor dessine une courbe d'hystérésis de Lissajous (figure 2.2.b) lorsqu'est tracé sa courbe $i = f(v)$, d'où sa célèbre phrase :

If it's pinched it's a memristor.

La figure A.1 montre donc l'hystérésis de la courbe $i = f(v)$ qui est caractéristique d'un memristor. En ingénierie neuromorphique, il est préférable de regarder l'évolution de la résistance ou de la conductance synaptique en fonction de la tension appliquée comme nous l'avons fait au chapitre 2 puisque c'est à partir de la conductance que les courbes de STDP sont tracées.

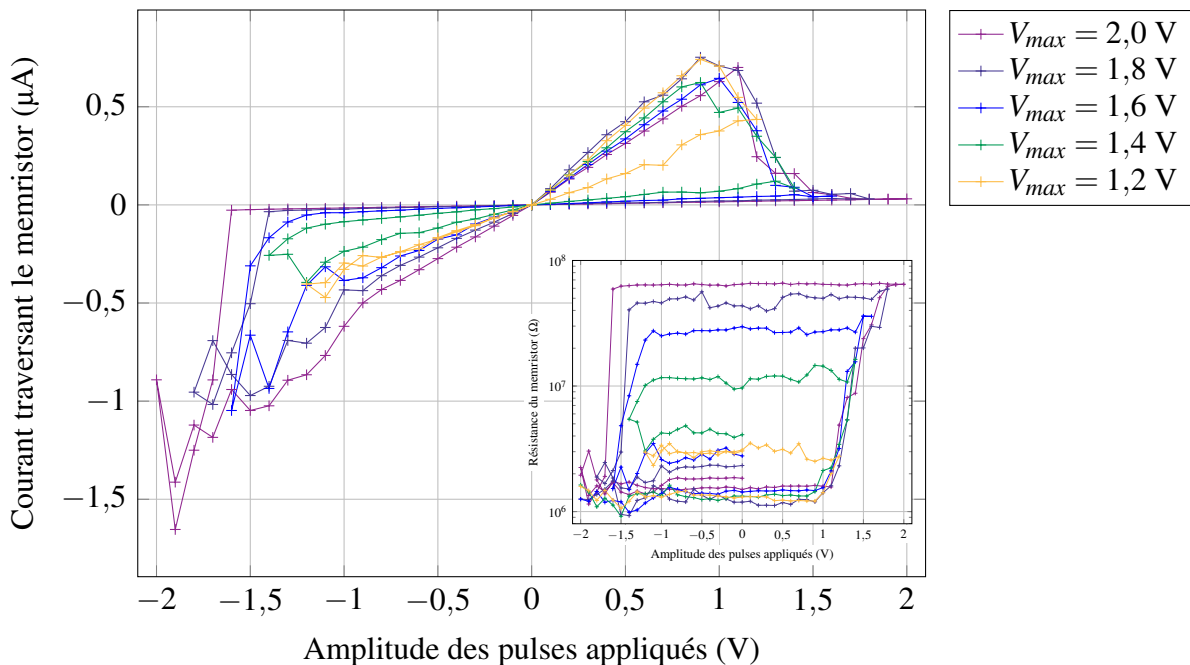


FIGURE A.1 – Mesures de l'hystérésis de courant en fonction de la tension d'un memristor BFO.

D'autres formes de STDP

Nous avons vu à la fin du chapitre 2 (2.3.3.2) qu'il était possible d'avoir d'autres formes de STDP (figure 2.27). La figure A.2 présente une autre forme de STDP réalisée avec un PA présynaptique ayant comme spécifications (suivant la convention de la figure 2.22) : $A_{pulse} = 0,8 \text{ V}$, $A_{offset_{max}} = -1,0 \text{ V}$ et $A_{offset_{min}} = -0,4 \text{ V}$. Le PA postsynaptique est, quant à lui, identique mais avec des tensions opposées. Les durées des deux PA étaient $t_{pulse} = 100 \text{ ns}$ et $t_{LTx} = 500 \text{ ns}$.

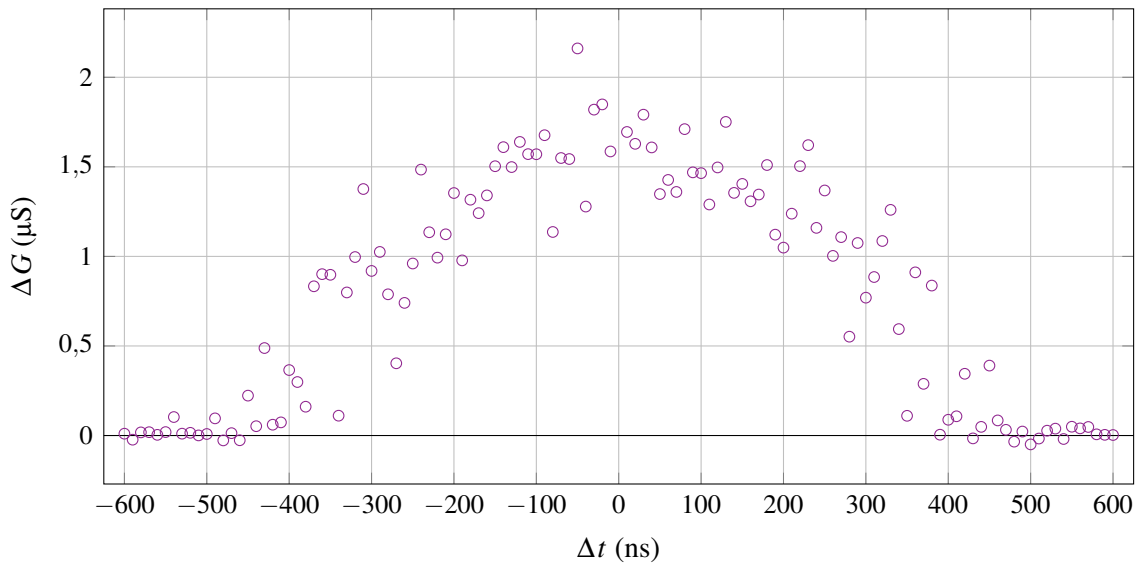


FIGURE A.2 – Mesures de STDP obtenues pour les PA présynaptiques suivants : $A_{pulse} = 0,8 \text{ V}$, $A_{offset_{max}} = -1,0 \text{ V}$ et $A_{offset_{min}} = -0,4 \text{ V}$. Les PA postsynaptiques avaient des amplitudes opposées. Les durées des deux PA étaient $t_{pulse} = 100 \text{ ns}$ et $t_{LTx} = 500 \text{ ns}$.

La forme de STDP ainsi obtenue peut être qualifiée de purement *hebbienne* car les poids synaptiques sont augmentés quelle que soit la valeur de Δt .

Alors que les formes précédentes de STDP ont été obtenues en faisant varier le potentiel aux bornes du memristor avec un pulse d'une durée constante, nous avons réalisé des mesures (figure A.3) en faisant varier le temps d'application de la tension maximale. Les PA choisis pour ces mesures sont formés d'un pulse d'une durée de 100 ns et d'amplitude positive puis d'un pulse d'une durée identique de 100 ns et d'amplitude négative. Lorsque $\Delta t = 100 \text{ ns}$, le memristor voit entre ses bornes un pulse d'une durée de 100 ns et d'amplitude maximale. L'amplitude est égale à la différence d'amplitude des deux PA soit 1,8 V lorsque $\Delta t < 0$ ou -2,0 V lorsque $\Delta t > 0$.

Pour Δt négatif, le comportement du memristor est approximativement linéaire de -190 ns à -20 ns. Et pour la partie où Δt est positif, on a une valeur minimale pour 90 ns. Ces mesures ne correspondent pas à nos attentes. Au vu de la forme des PA utilisés, nous nous attendions à avoir une forme identique de la STDP (au signe près) pour Δt négatif et positif, la forme des PA étant faite pour avoir un extrémum à -100 ns et 100 ns. Pour le moment, nous ne savons pas pourquoi nous obtenons cette forme particulière.

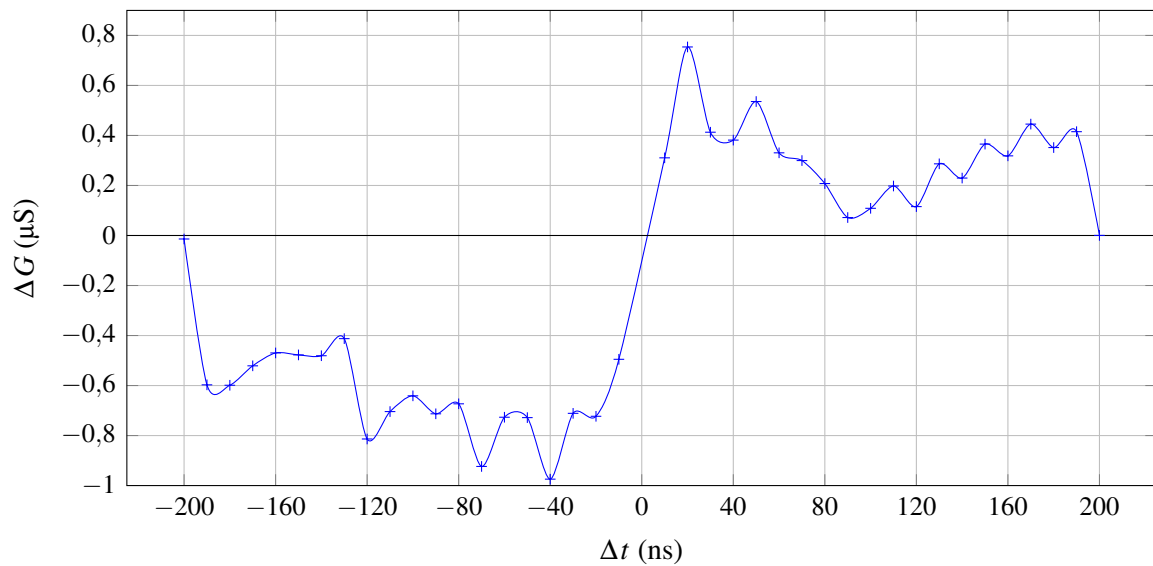


FIGURE A.3 – Mesures de STDP obtenues pour les PA suivants : $A_{pulse_{pré}} = 0,8 \text{ V}$, $A_{pulse_{post}} = 1,0 \text{ V}$, $A_{offset_{max_{pré}}} = A_{offset_{min_{pré}}} = -1,0 \text{ V}$, $A_{offset_{max_{post}}} = A_{offset_{min_{post}}} = -0,8 \text{ V}$, $t_{pulse} = 100 \text{ ns}$ et $t_{LTx} = 100 \text{ ns}$.

ANNEXE B

LES PARAMÈTRES DE LA MODÉLISATION DES FORMES DE STDP MESURÉES AVEC DES MEMRISTORS BFO

CETTE annexe donne, pour la modélisation des formes de STDP mesurées avec des memristors BFO, les paramètres extraits grâce à un algorithme génétique. L'équation de la modélisation choisie est proche des formes de STDP biologiques :

$$F(\Delta t) = \begin{cases} \frac{\Delta t}{\alpha} \exp(\Delta t/\tau_-) & \text{si } \Delta t < 0 \\ \frac{\Delta t}{\alpha} \exp(-\Delta t/\tau_+) & \text{si } \Delta t \geq 0 \end{cases}$$

TABLE B.1 – Paramètres pour le memristor 1.

	α	τ_-	τ_+
série 1	$3,538488.10^{-2}$	$1,837947.10^{-7}$	$1,206029.10^{-7}$
série 2	$4,251570.10^{-2}$	$1,930170.10^{-7}$	$1,189641.10^{-7}$
série 3	$3,436116.10^{-2}$	$1,588310.10^{-7}$	$1,064806.10^{-7}$
série 4	$3,308872.10^{-2}$	$1,576774.10^{-7}$	$9,807659.10^{-8}$
série 5	$3,304105.10^{-2}$	$1,557140.10^{-7}$	$9,562441.10^{-8}$
série 6	$3,801744.10^{-2}$	$1,621394.10^{-7}$	$1,003428.10^{-7}$
série 7	$3,551175.10^{-2}$	$1,476252.10^{-7}$	$9,269234.10^{-8}$
série 8	$3,526321.10^{-2}$	$1,414749.10^{-7}$	$8,175064.10^{-8}$

TABLE B.2 – Paramètres pour le memristor 2.

	α	τ_-	τ_+
série 1	$2,741359.10^{-2}$	$1,511501.10^{-7}$	$1,104850.10^{-7}$
série 2	$2,739023.10^{-2}$	$1,382136.10^{-7}$	$8,905909.10^{-8}$
série 3	$2,795341.10^{-2}$	$1,464999.10^{-7}$	$9,273307.10^{-8}$
série 4	$2,930477.10^{-2}$	$1,395850.10^{-7}$	$8,218367.10^{-8}$
série 5	$2,845904.10^{-2}$	$1,414349.10^{-7}$	$8,143539.10^{-8}$
série 6	$2,995060.10^{-2}$	$1,366576.10^{-7}$	$7,202598.10^{-8}$

TABLE B.3 – Paramètres pour le memristor 3.

	α	τ_-	τ_+
série 1	$4,499561.10^{-2}$	$1,444548.10^{-7}$	$8,579776.10^{-8}$
série 2	$4,114071.10^{-2}$	$1,444317.10^{-7}$	$6,908864.10^{-8}$
série 3	$4,464463.10^{-2}$	$1,452456.10^{-7}$	$7,426937.10^{-8}$
série 4	$6,048297.10^{-2}$	$1,605382.10^{-7}$	$7,176105.10^{-8}$

TABLE B.4 – Paramètres pour le memristor 4.

	α	τ_-	τ_+
série 1	$3,016317.10^{-2}$	$1,367120.10^{-7}$	$5,364492.10^{-8}$
série 2	$2,754139.10^{-2}$	$1,247986.10^{-7}$	$4,350590.10^{-8}$

ANNEXE C

**D'AUTRES MODÉLISATIONS DE STDP ET
EXEMPLES D'APPRENTISSAGE**

DANS la première partie de cette annexe nous présentons deux autres formes de **STDP** implémentées dans le code C++ servant à la simulation de réseau de neurones. Dans la seconde partie nous donnons des exemples d'apprentissage non-supervisé puis supervisé avec un réseau de 100 neurones.

D'autres modélisations de STDP

Le premier modèle s'appuie sur une forme de **STDP** biologiquement réaliste avec l'utilisation d'exponentielles pour la potentialisation et la dépression :

$$\Delta w(w, \Delta t) = \begin{cases} -\eta_- w \exp(\Delta t / t_{LTD}) & \text{si } -t_{LTD} \leq \Delta t < 0 \\ \eta_+ (w_{max} - w) \exp(-\Delta t / t_{LTP}) & \text{si } 0 \leq \Delta t \leq t_{LTP} \\ 0 & \text{sinon} \end{cases} \quad (\text{C.1})$$

La figure C.1 présente un exemple de variation Δw du poids en fonction du poids précédent w (figure C.1.a) et la plage de variation résultante de Δw en fonction de Δt (figure C.1.b).

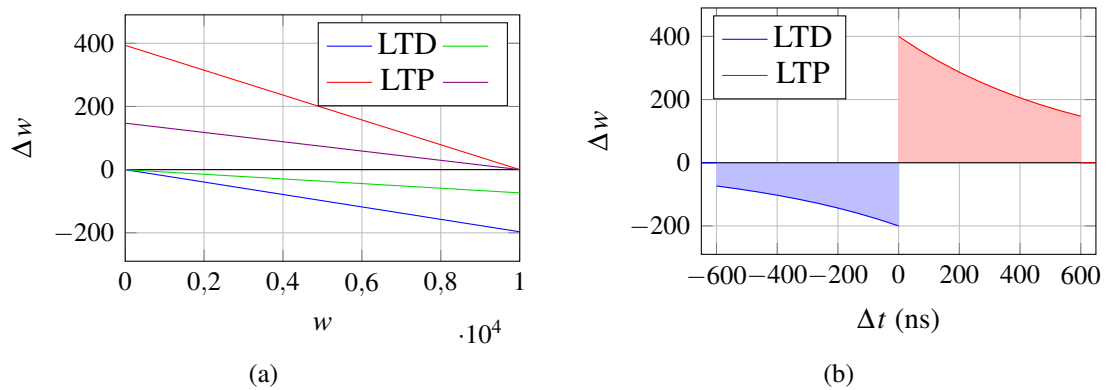


FIGURE C.1 – Tracés des variations de Δw avec les coefficients $\eta_- = 0,02$, $\eta_+ = 0,04$ et $t_{LTD} = t_{LTP} = 600$ ns. (a) Variation Δw du poids en fonction du poids synaptique précédent w pour $\Delta t = \pm 10$ ns (en rouge et bleu) et $\Delta t = \pm 600$ ns (en violet et vert). (b) Plage de variation Δw du poids en fonction de Δt .

La deuxième forme est semblable à la première mais présente une partie linéaire autour de 0. Le passage abrupte entre la partie LTD et LTP de la STDP est supprimé. Cette forme se rapproche de la forme obtenue avec les memristors.

$$\Delta t(w, \Delta t) = \begin{cases} -w\eta_- \exp(\Delta t / t_{LTD}) & \text{si } -t_{LTD} \leq \Delta t < -t_{linear} \\ -w\Delta t \frac{\eta_-}{t_{linear}} \exp(t_{linear} / t_{LTD}) & \text{si } -t_{linear} \leq \Delta t < 0 \\ (w_{max} - w) \frac{\eta_+}{t_{linear}} \exp(-t_{linear} / t_{LTP}) & \text{si } 0 \leq \Delta t \leq t_{linear} \\ (w_{max} - w)\eta_+ \exp(-\Delta t / t_{LTP}) & \text{si } t_{linear} < \Delta t \leq t_{LTP} \\ 0 & \text{sinon} \end{cases} \quad (C.2)$$

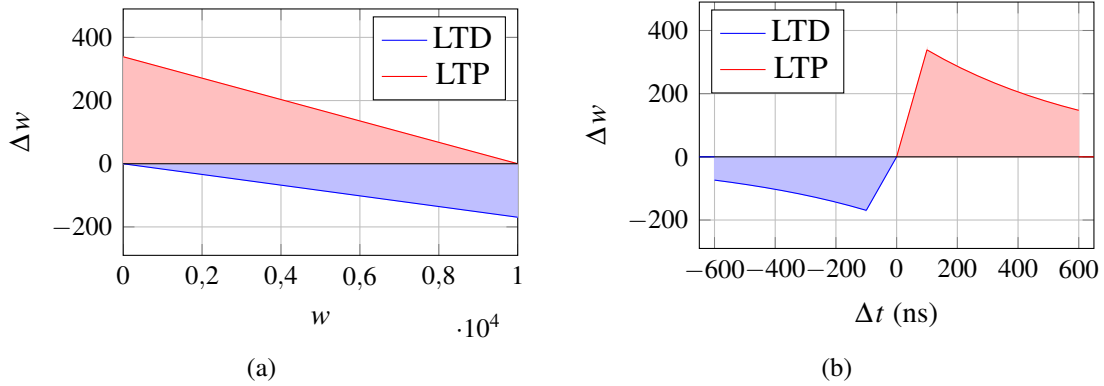


FIGURE C.2 – Tracés des variations de Δw avec les coefficients $\eta_- = 0,02$, $\eta_+ = 0,04$, $t_{linear} = 100$ ns, et $t_{LTD} = t_{LTP} = 600$ ns. (a) Variation Δw du poids en fonction du poids précédent w pour $\Delta t = \pm 10$ ns (en rouge et bleu) et $\Delta t = \pm 600$ ns (en violet et vert). (b) Plage de variation Δw du poids en fonction de Δt .

Deux exemples d'apprentissage

Cette partie de l'annexe présente deux exemples d'apprentissage : un non-supervisé et un supervisé.

Apprentissage non-supervisé : évolution des poids synaptiques

La figure C.3 présente une évolution typique des poids des synapses durant un apprentissage non-supervisé d'un réseau de 100 neurones.

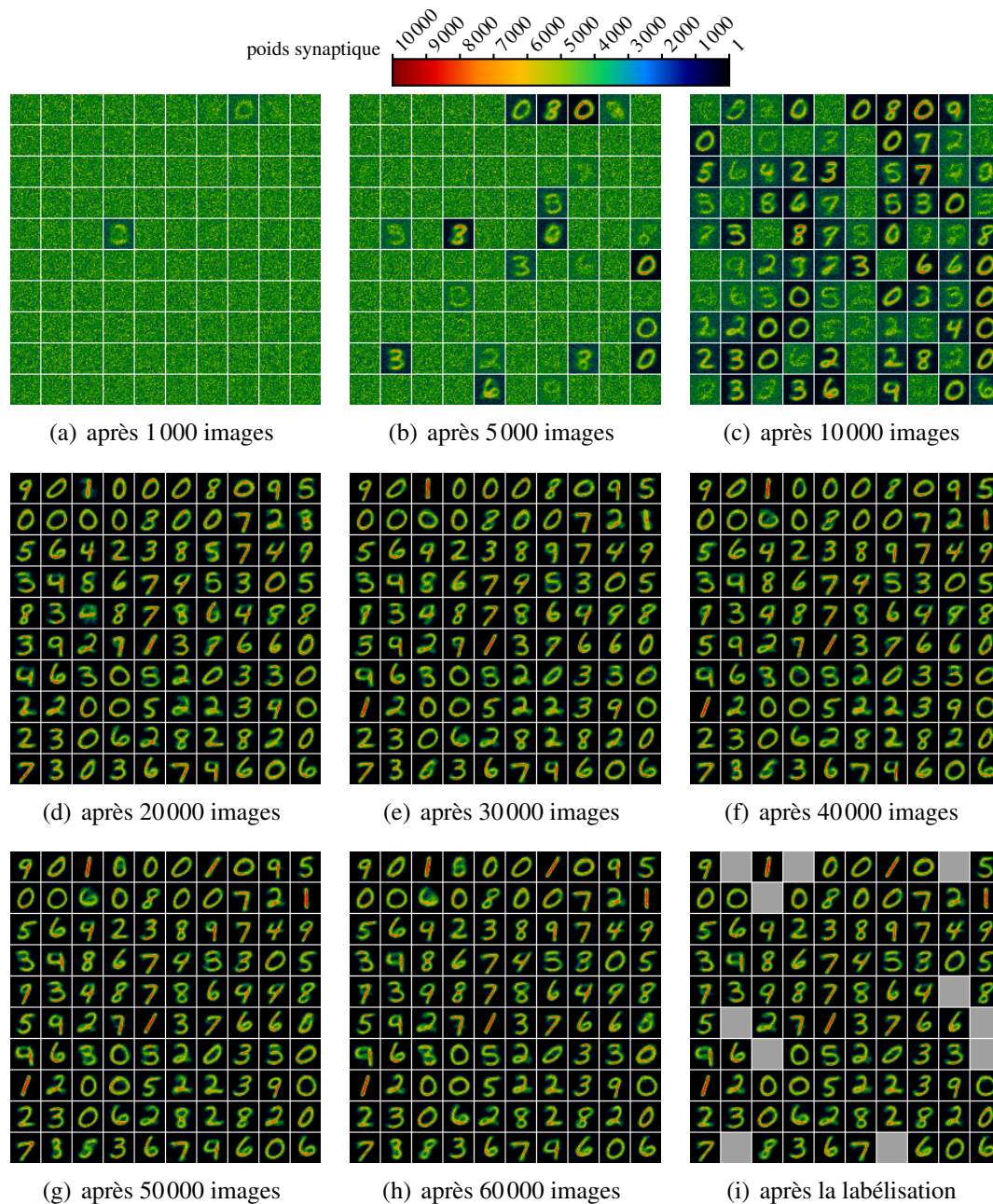


FIGURE C.3 – Évolution d'un apprentissage de 60 000 chiffres écrits à la main de 28×28 pixels provenant la base de données [MNIST](#). Il a été réalisé avec un réseau de 100 neurones.

Apprentissage supervisé

Nous avons testé un apprentissage supervisé en maîtrisant le seuil de déclenchement des neurones (figure C.4). La régulation a été contrôlée suivant le label de l'échantillon d'entrée. Une augmentation de seuil est appliquée à tous les neurones qui ne sont pas concernés par le label. Chacune des 10 classes de labels possède 10 neurones. Dans notre cas, il n'y a qu'une légère amélioration du taux de reconnaissance par rapport à un apprentissage non-supervisé (78 % contre 75,8 % de taux de réussite pour 100 neurones) mais nous n'avons pas optimisé les paramètres pour ce type d'apprentissage.

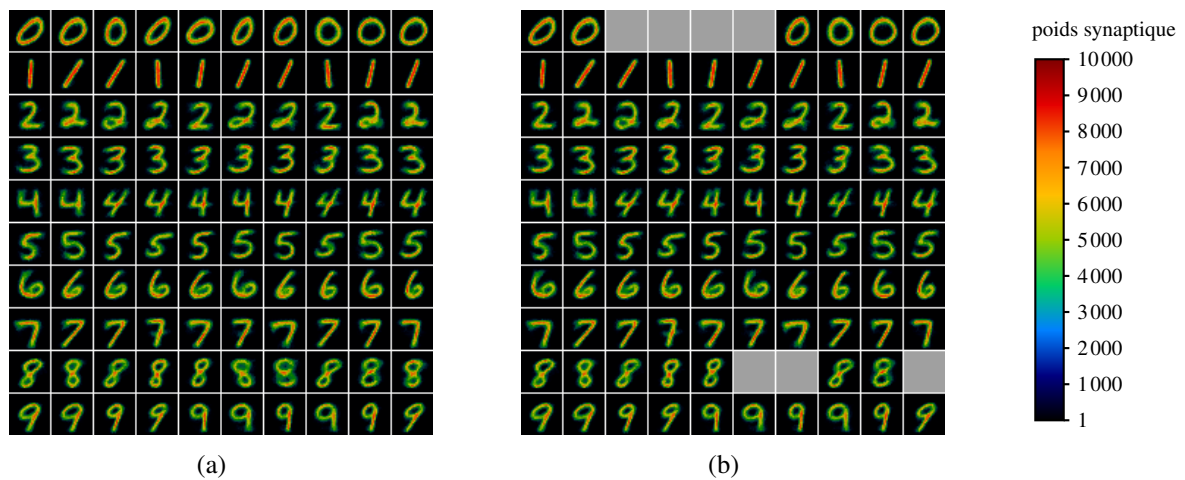


FIGURE C.4 – Résultat d'un apprentissage pseudo-supervisé de chiffres provenant de la [MNIST](#) avec 100 neurones de sortie ayant obtenu 78 % de réussite. (a) Poids des synapses à la fin de l'apprentissage des 60 000 images. (b) Élimination de certains neurones après la phase de labélisation.

ANNEXE D

**DES COMPLÉMENTS SUR LE
FONCTIONNEMENT DE LA SECONDE
VERSION DU RÉSEAU DE NEURONES SUR
SILICIUM**

DANS cette annexe, nous apportons un complément sur la seconde version de la puce. La figure D.1 présente la couronne de plots. La suite de l'annexe montre le chronogramme du fonctionnement de la puce et détaille le circuit de prise de décision WTA.

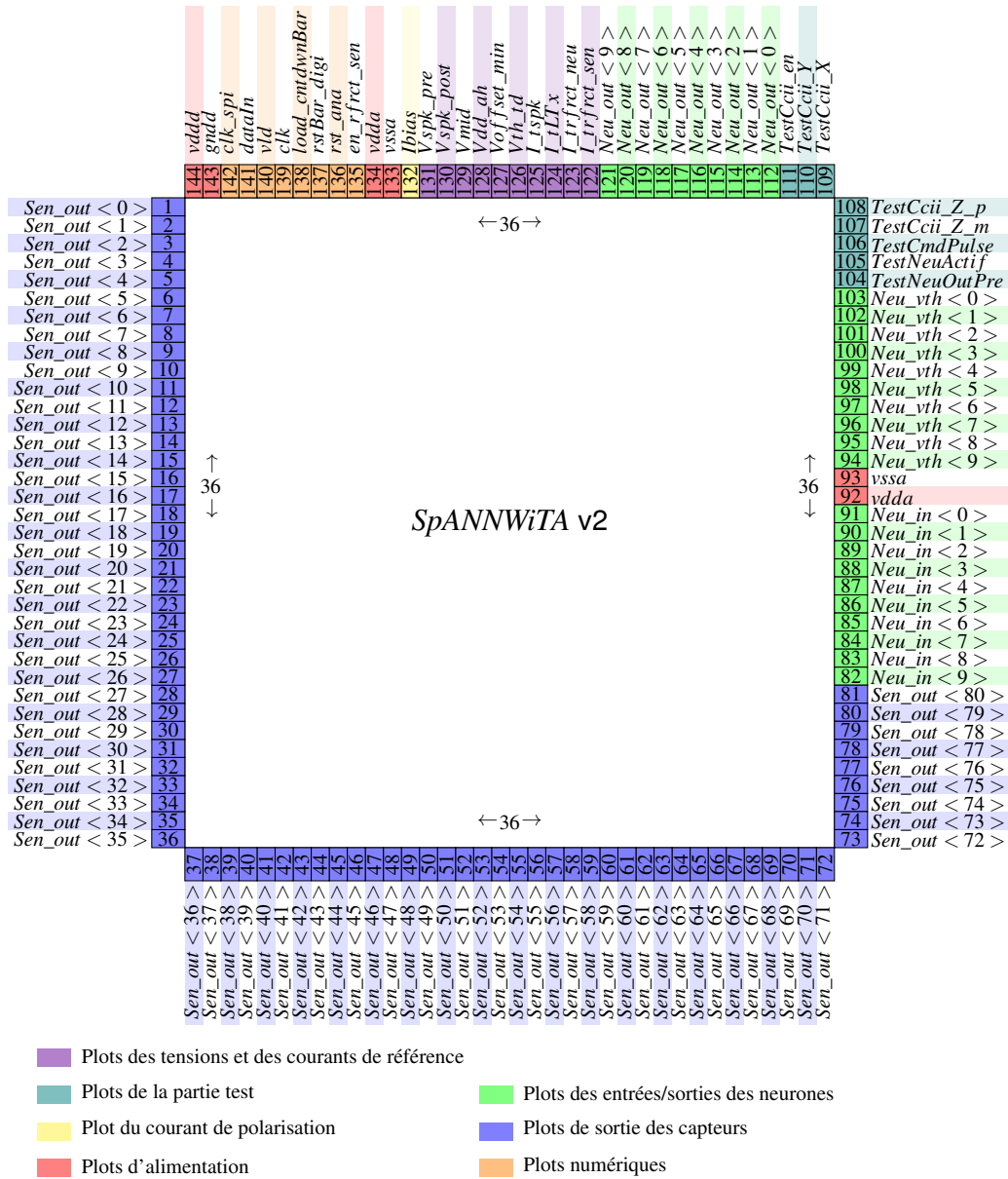


FIGURE D.1 – Couronne de plots de la seconde version de SpANNWiTA.

Le détail du fonctionnement de la seconde puce

La seconde version de la puce **SpANNWiTA** bénéficie d'un bloc numérique plus élaboré (cf. 4.3.2) que la première version. Le fonctionnement global de la puce a donc changé et il est détaillé dans le chronogramme de la figure D.2. Le mot binaire permettant de rentrer en mémoire les valeurs des décomptes est donné à la figure D.3.

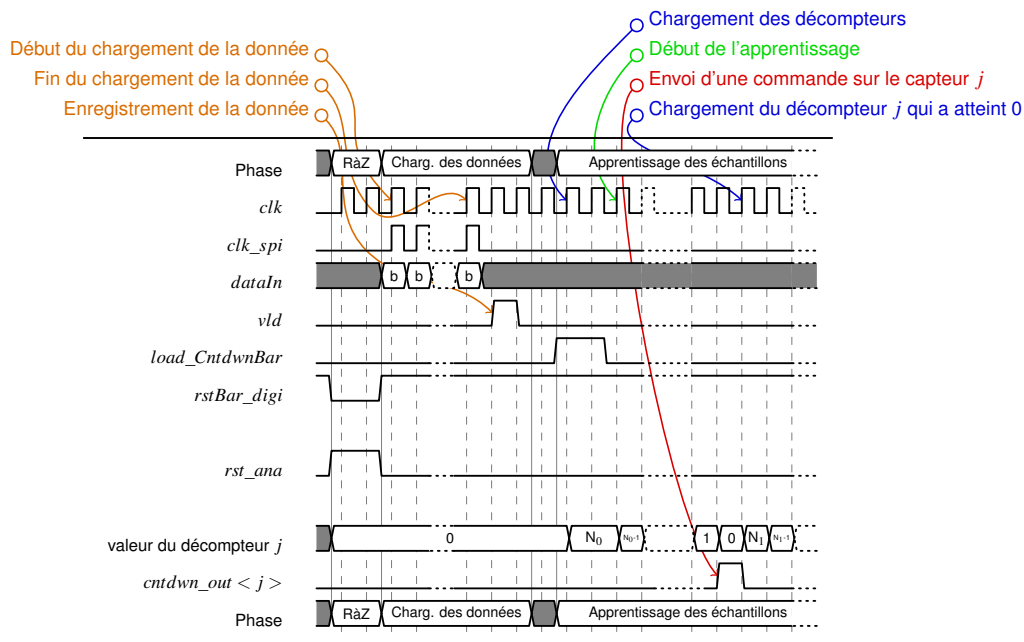


FIGURE D.2 – Chronogramme détaillant le fonctionnement de la seconde puce **SpANNWiTA**. La puce fonctionne suivant deux principales phases : la mise en mémoire des données puis leur apprentissage.

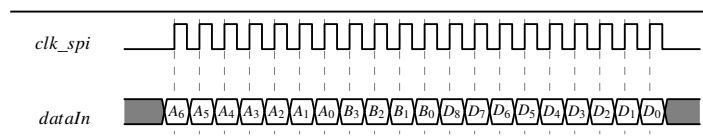


FIGURE D.3 – Mot binaire permettant de charger la mémoire de la puce. Un mot contient 7 bits A_i pour sélectionner le numéro du pixel (de 1 à 81), puis 4 bits B_j pour sélectionner le numéro du **PA** (de 1 à 10) et 9 bits D_k pour la donnée de décompte (MSB¹ en premier).

¹De l'anglais *Most Significant Bit*, il désigne le bit de poids fort.

Un complément sur le circuit WTA

La seconde version de la puce inclut un bloc de décision qui n'est pas présent dans la première version (cf. 4.3.1.3). Bien qu'une approche analogique existe (Lazzaro *et al.*, 1989) mais par méconnaissance au moment de la conception, nous avons utilisé une approche numérique présentée à la figure D.4.

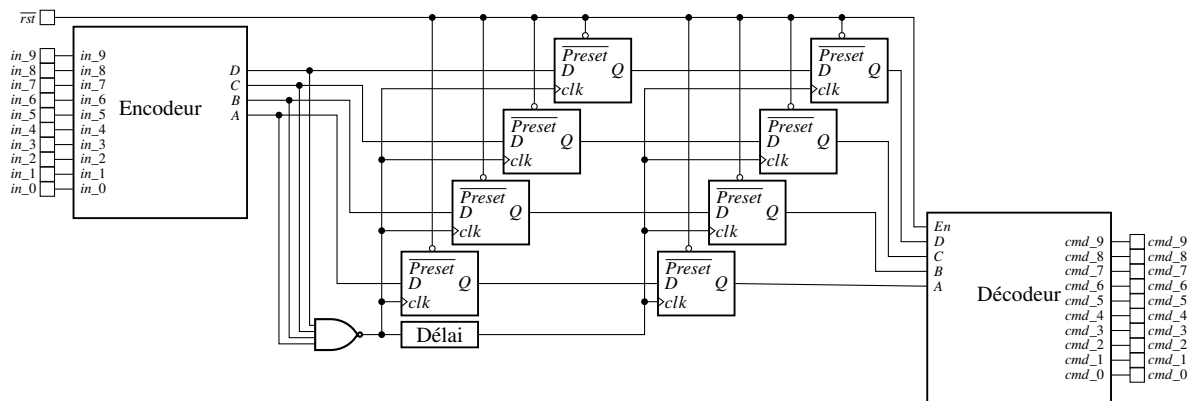


FIGURE D.4 – Circuit de décision WTA.

L'encodeur d'entrée est en logique inversée : les sorties sont à 1 lorsque la tension de membrane est inférieure au potentiel de seuil et passe à 0 dans le cas inverse. La porte ET-NON, connectée aux sorties de l'encodeur, passe donc de 0 à 1 lorsqu'il y a un changement dans l'état des sorties et permet à la première rangée de bascules D d'enregistrer la valeur binaire de sortie. En cas de dépassement de seuil de plusieurs potentiels de membrane dans un laps de temps limité (quelques nanosecondes), la valeur de l'encodeur peut changer alors que l'enregistrement dans la première rangée de bascules D est en cours. Des valeurs de sortie de bascules peuvent être amenées à changer. Pour éviter que cela ne crée deux valeurs décodées, nous avons ajouté une seconde rangée de bascules D. Ainsi la valeur enregistrée par les premières bascules est figée, et peut être envoyée sur les entrées du décodeur par l'intermédiaire de la seconde rangée de bascules D. Le décodeur garantit ainsi l'activation d'une seule sortie.

**RÉFÉRENCES BIBLIOGRAPHIQUES DE
L'AUTEUR**

Brevet

SAÏGHI, S., TOMAS, J. et LECERF, G. (2013). Organe à neurone artificiel et memristor. *Brevet Français n° FR 3003062*. Paris : Institut National de la Propriété Industrielle.

Conférences

LECERF, G., TOMAS, J., BOYN, S., GIROD, S., MANGALORE, A., GROLLIER, J. et SAÏGHI, S. (2014a). Silicon Neuron dedicated to Memristive Spiking Neural Networks. *In 2014 IEEE International Symposium on Circuits and Systems (ISCAS2014)*, p. 1568–1571.

LECERF, G., TOMAS, J., BOYN, S., GIROD, S., MANGALORE, A., GROLLIER, J. et SAÏGHI, S. (2014b). Silicon Neuron dedicated to Memristive Spiking Neural Networks. *In Journée Neurostic 2014 (GdR ISIS)*.

LECERF, G., TOMAS, J. et SAÏGHI, S. (2013a). Excitatory and Inhibitory Memristive Synapses for Spiking Neural Networks. *In 2013 IEEE International Symposium on Circuits and Systems (ISCAS2013)*, p. 1616–1619.

LECERF, G., TOMAS, J. et SAÏGHI, S. (2013b). Réseau de Neurones Impulsionnels avec Synapses Memristives. *In Colloque national 2013 du GdR SoC-SiP*.

RÉFÉRENCES BIBLIOGRAPHIQUES

- ABBOTT, L. F. (1999). Lapicque's introduction of the integrate-and-fire model neuron (1907). *Brain Research Bulletin*, 50(5-6):303–304.
- ABBOTT, L. F. et NELSON, S. B. (2000). Synaptic plasticity: taming the beast. *Nature Neuroscience*, 3 Suppl:1178–83.
- ALIBART, F., PLEUTIN, S., BICHLER, O., GAMRAT, C., SERRANO-GOTARREDONA, T., LINARES-BARRANCO, B. et VUILLAUME, D. (2012). A Memristive Nanoparticle/Organic Hybrid Synapstor for Neuroinspired Computing. *Advanced Functional Materials*, 22(3):609–616.
- ARTHUR, J. V. et BOAHEN, K. (2006). Learning in silicon: Timing is everything.
- BADOUAL, M., ZOU, Q., DAVISON, A. P., RUDOLPH, M., BAL, T., FRÉGNAC, Y. et DESTEXHE, A. (2006). Biophysical and phenomenological models of multiple spike interactions in spike-timing dependent plasticity. *International Journal of Neural Systems*, 16(2):79–97.
- BASHAM, E. J. et PARENT, D. W. (2009). An analog circuit implementation of a quadratic integrate and fire neuron. *Conference proceedings: Annual International Conference of the IEEE Engineering in Medicine and Biology Society.*, p. 741–744.
- BENJAMIN, B., PEIRAN GAO, MCQUINN, E., CHOUDHARY, S., CHANDRASEKARAN, A., BUSSAT, J.-M., ALVAREZ-ICAZA, R., ARTHUR, J., MEROLLA, P. et BOAHEN, K. (2014). Neurogrid: A Mixed-Analog-Digital Multichip System for Large-Scale Neural Simulations. *Proceedings of the IEEE*, 102(5):699–716.
- BI, G. Q. et POO, M. M. (1998). Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and postsynaptic cell type. *The Journal of Neuroscience : the Official Journal of the Society for Neuroscience*, 18(24):10464–72.
- BIBES, M., BARTHÉLÉMY, A., GROLLIER, J. et MAGE, J.-C. (2010). Dispositif ferroélectrique à résistance ajustable. *Brevet Internationnal n° WO 2010142762*. Paris: Institut National de la Propriété Industrielle.
- BICHLER, O. (2012). *Contribution à la conception d'architecture de calcul auto-adaptative intégrant des nanocomposants neuromorphiques et applications potentielles*. Thèse de doctorat, Université Paris Sud.
- BICHLER, O., QUERLIOZ, D., THORPE, S. J., BOURGOIN, J.-P. et GAMRAT, C. (2012). Extraction of temporally correlated features from dynamic vision sensors with spike-timing-dependent plasticity. *Neural Networks : the Official Journal of the International Neural Network Society*, 32:339–48.

- BICHLER, O., ROCLIN, D., GAMRAT, C. et QUERLIOZ, D. (2013). Design exploration methodology for memristor-based spiking neuromorphic architectures with the Xnet event-driven simulator. In *2013 IEEE/ACM International Symposium on Nanoscale Architectures (NANOARCH)*, p. 7–12.
- BRETTE, R. et GERSTNER, W. (2005). Adaptive exponential integrate-and-fire model as an effective description of neuronal activity. *Journal of Neurophysiology*, 94(5):3637–42.
- BUHRY, L., GRASSIA, F., GIREMUS, A., GRIVEL, E., RENAUD, S. et SAÏGHI, S. (2011). Automated parameter estimation of the Hodgkin-Huxley model using the differential evolution algorithm: application to neuromimetic analog integrated circuits. *Neural Computation*, 23(10):2599–625.
- CHANTHBOUALA, A., GARCIA, V., CHERIFI, R. O., BOUZEHOUE, K., FUSIL, S., MOYA, X., XAVIER, S., YAMADA, H., DERANLOT, C., MATHUR, N. D., BIBES, M., BARTHÉLÉMY, A. et GROLLIER, J. (2012). A ferroelectric memristor. *Nature Materials*, 11(10):860–4.
- CHUA, L. O. (1971). Memristor-The missing circuit element. *IEEE Transactions on Circuit Theory*, 18(5):507–519.
- CHUA, L. O. et KANG, S. M. (1976). Memristive devices and systems. *Proceedings of the IEEE*, 64(2):209–223.
- CIREŞAN, D., MEIER, U. et SCHMIDHUBER, J. (2012). Multi-column Deep Neural Networks for Image Classification. *Technical Report*, p. 3642–3649.
- CRUPI, M., PRADHAN, L. et TOZER, S. (2012). Modelling Neural Plasticity with Memristors. *IEEE Canadian Review*, p. 10–14.
- DAVIDSON, S. (2005). SpiNNaker project. <http://apt.cs.manchester.ac.uk/projects/SpiNNaker/>.
- DIORIO, C., HASLER, P., MINCH, A. et MEAD, C. (1996). A single-transistor silicon synapse. *IEEE Transactions on Electron Devices*, 43(11):1972–1980.
- DIORIO, C., HASLER, P., MINCH, B. et MEAD, C. (1997). A floating-gate MOS learning array with locally computed weight updates. *IEEE Transactions on Electron Devices*, 44(12):2281–2289.
- DUBEY, P. (2005). Recognition, Mining and Synthesis Moves Computers to the Era of Tera. *Technology@Intel Magazine*, 9:1–10.
- ELMAN, J. L. (1990). Finding Structure in Time. *Cognitive Science*, 14(2):179–211.
- FITZHUGH, R. (1955). Mathematical models of threshold phenomena in the nerve membrane. *The Bulletin of Mathematical Biophysics*, 17(4):257–278.

- FITZHUGH, R. (1961). Impulses and Physiological States in Theoretical Models of Nerve Membrane. *Biophysical Journal*, 1(6):445–66.
- FUSI, S., ANNUNZIATO, M., BADONI, D., SALAMON, A. et AMIT, D. J. (2000). Spike-driven synaptic plasticity: theory, simulation, VLSI implementation. *Neural computation*, 12(10):2227–58.
- GERSTNER, W. et KISTLER, W. M. (2002). *Spiking Neuron Models: Single Neurons, Populations, Plasticity*. Cambridge University Press.
- GOLDBERG, D., CAUWENBERGHS, G. et ANDREOU, A. (2001). Analog VLSI spiking neural network with address domain probabilistic synapses. In *ISCAS 2001. The 2001 IEEE International Symposium on Circuits and Systems*, volume 2, p. 241–244.
- GRASSIA, F., BUHRY, L., LÉVI, T., TOMAS, J., DESTEXHE, A. et SAÏGHI, S. (2011). Tunable neuromimetic integrated system for emulating cortical neuron models. *Frontiers in Neuroscience*, 5(134):1–12.
- HAMILTON, T. J. et van SCHAİK, A. (2011). Silicon implementation of the generalized integrate-and-fire neuron model. In *2011 Seventh International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, p. 108–112.
- HEBB, D. (1949). *The Organization of Behavior: A Neuropsychological Theory*. Wiley.
- HINDMARSH, J. L. et ROSE, R. M. (1984). A model of neuronal bursting using three coupled first order differential equations. *Proceedings of the Royal Society of London. Series B, Containing papers of a Biological character. Royal Society (Great Britain)*, 221(1222):87–102.
- HODGKIN, A. L. et HUXLEY, A. F. (1952). A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of Physiology*, 117(4):500–44.
- HOGERVORST, R., TERO, J., ESCHAUZIER, R. et HUIJSING, J. (1994). A compact power-efficient 3 V CMOS rail-to-rail input/output operational amplifier for VLSI cell libraries. In *Proceedings of IEEE International Solid-State Circuits Conference - ISSCC '94*, p. 244–245.
- HOPFIELD, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, 79(8):2554–2558.
- INDIVERI, G., LINARES-BARRANCO, B., HAMILTON, T. J., van SCHAİK, A., ETIENNE-CUMMINGS, R., DELBRUCK, T., LIU, S.-C., DUDEK, P., HÄFLIGER, P., RENAUD, S., SCHEMMELE, J., CAUWENBERGHS, G., ARTHUR, J. V., HYNNA, K., FOLOWOSELE, F., SAÏGHI, S., SERRANO-GOTARREDONA, T., WIJEKOON, J. H. B., WANG, Y. et BOAHEN, K. (2011). Neuromorphic silicon neuron circuits. *Frontiers in Neuroscience*, 5(73):1–23.

- IZHIKEVICH, E. M. (2001). Resonate-and-fire neurons. *Neural Networks*, 14(6-7):883–894.
- IZHIKEVICH, E. M. (2003). Simple model of spiking neurons. *IEEE Transactions on Neural Networks / a publication of the IEEE Neural Networks Council*, 14(6):1569–72.
- JO, S. H., CHANG, T., EBONG, I., BHADVIYA, B. B., MAZUMDER, P. et LU, W. (2010). Nanoscale memristor device as synapse in neuromorphic systems. *Nano Letters*, 10(4):1297–301.
- JORDAN, M. I. (1986). Serial Order: A Parallel Distributed Processing Approach. In *Neural-Network Models of Cognition — Biobehavioral Foundations*, p. 471–495. Advances in Psychology.
- KANDEL, E. R. (1976). *Cellular Basis of Behavior : An Introduction to Behavioral Neurobiology*. W. H. Freeman and Company.
- KANDEL, E. R., SCHWARTZ, J. H. et JESSELL, T. (2000). *Principles of Neural Science*. McGraw-Hill Medical.
- KRZYSTECZKO, P., MÜNCHENBERGER, J., SCHÄFERS, M., REISS, G. et THOMAS, A. (2012). The memristive magnetic tunnel junction as a nanoscopic synapse-neuron system. *Advanced Materials (Deerfield Beach, Fla.)*, 24(6):762–6.
- KUMAR, S. (2013). Introducing Qualcomm Zeroth Processors: Brain-Inspired Computing.
- KUZUM, D., JEYASINGH, R. G. D., LEE, B. et WONG, H.-S. P. (2012a). Nanoelectronic programmable synapses based on phase change materials for brain-inspired computing. *Nano Letters*, 12(5):2179–86.
- KUZUM, D., JEYASINGH, R. G. D., YU, S. et WONG, H.-S. P. (2012b). Low-Energy Robust Neuromorphic Computation Using Synaptic Devices. *IEEE Transactions on Electron Devices*, 59(12):3489–3494.
- LAI, Q., ZHANG, L., LI, Z., STICKLE, W. F., WILLIAMS, R. S. et CHEN, Y. (2010). Ionic/electronic hybrid materials integrated in a synaptic transistor with signal processing and learning functions. *Advanced Materials (Deerfield Beach, Fla.)*, 22(22):2448–53.
- LAPICQUE, L. (1907). Recherches quantitatives sur l'excitation électrique des nerfs traitée comme une polarisation. *J Physiol Pathol Gen*, 9:620–635.
- LAZZARO, J., RYCKEBUSCH, S., MAHOWALD, M. A. et MEAD, C. A. (1989). Winner-take-all networks of $O(n)$ complexity. *Advances in Neural Information Processing Systems 1*, p. 703–711.
- LE CUN, Y. (1985). A learning scheme for asymmetric threshold networks. *Proceedings of Cognitiva 85*, p. 599–604.

- LE CUN, Y., BOTTOU, L., BENGIO, Y. et HAFFNER, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.
- LI, Y., ZHONG, Y., ZHANG, J., XU, L., WANG, Q., SUN, H., TONG, H., CHENG, X. et MIAO, X. (2014). Activity-dependent synaptic plasticity of a chalcogenide electronic synapse for neuromorphic systems. *Scientific Reports*, 4(4906):1–7.
- LINARES-BARRANCO, B. et SERRANO-GOTARREDONA, T. (2009a). Exploiting memristance in adaptive asynchronous spiking neuromorphic nanotechnology systems. *In 9th IEEE Conference on Nanotechnology*, p. 601–604.
- LINARES-BARRANCO, B. et SERRANO-GOTARREDONA, T. (2009b). Memristance can explain Spike-Time-Dependent-Plasticity in Neural Synapses. *Nature Precedings*, (3010):1–4.
- LOWNEY, G. (2006). Why Intel is designing multi-core processors. *In Proceedings of the eighteenth annual ACM symposium on Parallelism in Algorithms and Architectures - SPAA '06*. ACM Press.
- MAHOWALD, M. et DOUGLAS, R. (1991). A silicon neuron. *Nature*, 354(6354):515–8.
- MANDAL, S., EL-AMIN, A., ALEXANDER, K., RAJENDRAN, B. et JHA, R. (2014). Novel synaptic memory device for neuromorphic computing. *Scientific Reports*, 4(5333).
- MARDER, E. et GOAILLARD, J.-M. (2006). Variability, compensation and homeostasis in neuron and network function. *Nature Reviews. Neuroscience*, 7(7):563–74.
- MASQUELIER, T., GUYONNEAU, R. et THORPE, S. J. (2009). Competitive STDP-based spike pattern learning. *Neural Computation*, 21(5):1259–76.
- MCCULLOCH, W. S. et PITTS, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, 5(4):115–133.
- MEAD, C. (1989). *Analog VLSI and neural systems*. Addison-Wesley Longman Publishing Co., Inc.
- MEIER, K. (2002). SenseMaker project. http://cordis.europa.eu/result/rcn/42027_en.html.
- MEIER, K. (2005). FACETS project. <http://facets.kip.uni-heidelberg.de/>.
- MEIER, K. (2011). BrainScaleS project. <https://brainscales.kip.uni-heidelberg.de/index.html>.
- MIHALAŞ, S. et NIEBUR, E. (2009). A generalized linear integrate-and-fire neural model produces diverse spiking behaviors. *Neural Computation*, 21(3):704–18.

- MILLNER, S., GRÜBL, A., MEIER, K., SCHEMMELE, J. et SCHWARTZ, M.-o. (2010). A VLSI Implementation of the Adaptive Exponential Integrate-and-Fire Neuron Model. *In Advances in Neural Information Processing Systems*, p. 1642–1650.
- MOORE, S. K. (2008). Multicore is bad news for supercomputers. *IEEE Spectrum*.
- NAGUMO, J., ARIMOTO, S. et YOSHIKAWA, S. (1962). An Active Pulse Transmission Line Simulating Nerve Axon. *Proceedings of the IRE*, 50(10):2061–2070.
- NAKADA, K., ASAI, T. et HAYASHI, H. (2005). Silicon resonate-and-fire neuron based on the volterra system. *In Int. Symp. on Nonlinear Theory and its Applications*, p. 82–85.
- PAN, F., CHEN, C., WANG, Z.-s., YANG, Y.-c., YANG, J. et ZENG, F. (2010). Nonvolatile resistive switching memories-characteristics, mechanisms and challenges. *Progress in Natural Science: Materials International*, 20:1–15.
- PARK, S., KIM, H., CHOO, M., NOH, J., SHERI, A., JUNG, S., SEO, K., PARK, J., KIM, S., LEE, W., SHIN, J., LEE, D., CHOI, G., WOO, J., CHA, E., JANG, J., PARK, C., JEON, M., LEE, B., LEE, B. H. et HWANG, H. (2012). RRAM-based synapse for neuromorphic system with pattern recognition function. *In 2012 International Electron Devices Meeting*, p. 10.2.1–10.2.4.
- PARKER, D. B. (1982). Learning-logic.
- PÉREZ-CARRASCO, J. A., ACHA, B. n., SERRANO, C., CAMUNAS-MESA, L., SERRANO-GOTARREDONA, T. et LINARES-BARRANCO, B. (2010). Fast vision through frameless event-based sensing and convolutional processing: application to texture recognition. *IEEE Transactions on Neural Networks / a publication of the IEEE Neural Networks Council*, 21(4):609–20.
- PÉREZ-CARRASCO, J. A., ZHAO, B., SERRANO, C., ACHA, B. n., SERRANO-GOTARREDONA, T., CHEN, S. et LINARES-BARRANCO, B. (2013). Mapping from frame-driven to frame-free event-driven vision systems by low-rate rate coding and coincidence processing—application to feedforward ConvNets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(11):2706–19.
- QUERLIOZ, D., BICHLER, O., DOLLFUS, P. et GAMRAT, C. (2013). Immunity to Device Variations in a Spiking Neural Network With Memristive Nanodevices. *IEEE Transactions on Nanotechnology*, 12(3):288–295.
- QUERLIOZ, D., DOLLFUS, P., BICHLER, O. et GAMRAT, C. (2011). Learning with memristive devices: How should we model their behavior? *In 2011 IEEE/ACM International Symposium on Nanoscale Architectures*, p. 150–156.

- RAMAKRISHNAN, S., HASLER, P. E. et GORDON, C. (2011). Floating gate synapses with spike-time-dependent plasticity. *IEEE Transactions on Biomedical Circuits and Systems*, 5(3):244–52.
- RENAUD, S., TOMAS, J., BORNAT, Y., DAOUZLI, A. et SAÏGHI, S. (2007). Neuromimetic ICs with analog cores: an alternative for simulating spiking neural networks. In *2007 IEEE International Symposium on Circuits and Systems*, p. 3355–3358.
- RUMELHART, D. E. et ZIPSER, D. (1986). Feature discovery by competitive learning. *Cognitive Science*, 9(1):75–112.
- SAÏGHI, S., BORNAT, Y., TOMAS, J., LE MASSON, G. et RENAUD, S. (2011). A library of analog operators based on the Hodgkin-Huxley formalism for the design of tunable, real-time, silicon neurons. *IEEE Transactions on Biomedical Circuits and Systems*, 5(1):3–19.
- SCARPETTA, S., GIACCO, F., LOMBARDI, F. et de CANDIA, A. (2013). Effects of Poisson noise in a IF model with STDP and spontaneous replay of periodic spatiotemporal patterns, in absence of cue stimulation. *Bio Systems*, 112(3):258–64.
- SCHMIDT, H., MAYR, C., STÄRKE, P., PARTZSCH, J., CEDERSTROEM, L., SCHÜFFNY, R., SHUAI, Y. et DU, N. (2012). Waveform Driven Plasticity in BiFeO₃ Memristive Devices: Model and Implementation. In *Advances in Neural Information Processing Systems 25*, p. 1–9.
- SEO, K., KIM, I., JUNG, S., JO, M., PARK, S., PARK, J., SHIN, J., BIJU, K. P., KONG, J., LEE, K., LEE, B. et HWANG, H. (2011). Analog memory and spike-timing-dependent plasticity characteristics of a nanoscale titanium oxide bilayer resistive switching device. *Nanotechnology*, 22(25):254023.
- SERRANO-GOTARREDONA, T., MASQUELIER, T., PRODROMAKIS, T., INDIVERI, G. et LINARES-BARRANCO, B. (2013). STDP and STDP variations with memristors for spiking neuromorphic learning systems. *Frontiers in Neuroscience*, 7(2):1–15.
- SHIN, S., SACCHETTO, D., LEBLEBICI, Y. et KANG, S.-M. S. (2012). Neuronal spike event generation by memristors. In *2012 13th International Workshop on Cellular Nanoscale Networks and their Applications*, p. 1–4.
- SIMARD, P., STEINKRAUS, D. et PLATT, J. (2003). Best practices for convolutional neural networks applied to visual document analysis. In *Seventh International Conference on Document Analysis and Recognition, 2003. Proceedings.*, volume 1, p. 958–963.
- SMITH, A., MCDAID, L. et HALL, S. (2014). A compact spike-timing-dependent-plasticity circuit for floating gate weight implementation. *Neurocomputing*, 124:210–217.

- SNIDER, G. S. (2008). Spike-timing-dependent learning in memristive nanodevices. *In 2008 IEEE International Symposium on Nanoscale Architectures*, p. 85–92.
- SONG, S. et ABBOTT, L. F. (2001). Cortical Development and Remapping through Spike Timing-Dependent Plasticity. *Neuron*, 32(2):339–350.
- STRUKOV, D. B., SNIDER, G. S., STEWART, D. R. et WILLIAMS, R. S. (2008). The missing memristor found. *Nature*, 453(7191):80–3.
- SUBRAMANIAM, A., CANTLEY, K. D., BERSUKER, G., GILMER, D. C. et VOGEL, E. M. (2013). Spike-Timing-Dependent Plasticity Using Biologically Realistic Action Potentials and Low-Temperature Materials. *IEEE Transactions on Nanotechnology*, 12(3):450–459.
- SURI, M., BICHLER, O., HUBERT, Q., PERNIOLA, L., SOUSA, V., JAHAN, C., VUILLAUME, D., GAMRAT, C. et DESALVO, B. (2013a). Addition of HfO₂ interface layer for improved synaptic performance of phase change memory (PCM) devices. *Solid-State Electronics*, 79:227–232.
- SURI, M., QUERLIOZ, D., BICHLER, O., PALMA, G., VIANELLO, E., VUILLAUME, D., GAMRAT, C. et DESALVO, B. (2013b). Bio-Inspired Stochastic Computing Using Binary CBRAM Synapses. *IEEE Transactions on Electron Devices*, 60(7):2402–2409.
- TEMAM, O. (2010). The Rebirth of Neural Networks. *In International Symposium on Computer Architecture*.
- THOMAS, A. (2013). Memristor-based neural networks. *Journal of Physics D: Applied Physics*, 46(9):093001.
- THORPE, S. J. (1990). Spike arrival times : A highly efficient coding scheme for neural networks. *Parallel Processing in Neural Systems*, p. 91 – 94.
- TOUBOUL, J. (2008). *Modèles non-linéaires et stochastiques en neuroscience*. Thèse de doctorat, École Polytechnique.
- TOUMAZOU, C., LIDGEY, F. J. et HAIGH, D. (1993). *Analogue IC Design: The Current-Mode Approach*. IET, The Institution of Engineering and Technology, Michael Faraday House, Six Hills Way, Stevenage SG1 2AY, UK.
- TURRIGIANO, G. G. et NELSON, S. B. (2000). Hebb and homeostasis in neuronal plasticity. *Current Opinion in Neurobiology*, 10(3):358–364.
- TURRIGIANO, G. G. et NELSON, S. B. (2004). Homeostatic plasticity in the developing nervous system. *Nature Reviews. Neuroscience*, 5(2):97–107.
- WANG, Z. Q., XU, H. Y., LI, X. H., YU, H., LIU, Y. C. et ZHU, X. J. (2012). Synaptic Learning and Memory Functions Achieved Using Oxygen Ion Migration/Diffusion in an Amorphous InGaZnO Memristor. *Advanced Functional Materials*, 22(13):2759–2765.

- WERBOS, P. J. (1974). *Beyond regression : new tools for prediction and analysis in the behavioral sciences*. Thèse de doctorat, Harvard University.
- WIDROW, B. (1960). An Adaptive 'Adaline' Neuron Using Chemical 'Memistors'. *Stanford Electronics Laboratories Technical Report*, p. 1553–2.
- WIDROW, B. et HOFF, M. E. (1960). Adaptive Switching Circuits. *IRE WESCON Convention Record*, 4:96–104.
- WIJEKON, J. H. B. et DUDEK, P. (2006). Simple Analogue VLSI Circuit of a Cortical Neuron. In *2006 13th IEEE International Conference on Electronics, Circuits and Systems*, p. 1344–1347.
- WIJEKON, J. H. B. et DUDEK, P. (2008). Compact silicon neuron circuit with spiking and bursting behaviour. *Neural Networks : the Official Journal of the International Neural Network Society*, 21(2-3):524–34.
- WIJEKON, J. H. B. et DUDEK, P. (2012). VLSI circuits implementing computational models of neocortical circuits. *Journal of Neuroscience Methods*, 210(1):93–109.
- WU, Y., YU, S., WONG, H.-S. P., CHEN, Y.-S., LEE, H.-Y., WANG, S.-M., GU, P.-Y., CHEN, F. et TSAI, M.-J. (2012). AlOx-Based Resistive Switching Device with Gradual Resistance Modulation for Neuromorphic Device Application. In *2012 4th IEEE International Memory Workshop*, p. 1–4.
- WULF, W. A. et MCKEE, S. A. (1995). Hitting the memory wall. *ACM SIGARCH Computer Architecture News*, 23(1):20–24.
- WUTTIG, M. et YAMADA, N. (2007). Phase-change materials for rewriteable data storage. *Nature Materials*, 6(11):824–32.
- YAMADA, H., GARCIA, V., FUSIL, S., BOYN, S., MARINOVA, M., GLOTER, A., XAVIER, S., GROLLIER, J., JACQUET, E., CARRÉTÉRO, C., DERANLOT, C., BIBES, M. et BARTHÉLÉMY, A. (2013). Giant electroresistance of super-tetragonal BiFeO₃-based ferroelectric tunnel junctions. *ACS Nano*, 7(6):5385–90.
- YU, S., GAO, B., FANG, Z., YU, H., KANG, J. et WONG, H.-S. P. (2012). A neuromorphic visual system using RRAM synaptic devices with Sub-pJ energy and tolerance to variability: Experimental characterization and large-scale modeling. In *2012 International Electron Devices Meeting*, p. 10.4.1–10.4.4.
- YU, S., WU, Y., JEYASINGH, R., KUZUM, D. et WONG, H.-S. P. (2011). An Electronic Synapse Device Based on Metal Oxide Resistive Switching Memory for Neuromorphic Computation. *IEEE Transactions on Electron Devices*, 58(8):2729–2737.

Zamarreño RAMOS, C., Camuñas MESA, L. A., PÉREZ-CARRASCO, J. A., MASQUELIER, T., SERRANO-GOTARREDONA, T. et LINARES-BARRANCO, B. (2011). On spike-timing-dependent-plasticity, memristive devices, and building a self-learning visual cortex. *Frontiers in Neuroscience*, 5(26).

ZEKI, A. et KUNTMAN, H. (2000). High-linearity low-voltage self-cascode class AB CMOS current output stage. In *2000 IEEE International Symposium on Circuits and Systems. Emerging Technologies for the 21st Century.*, volume 4, p. 257–260. Presses Polytech. Univ. Romandes.