Conception de réseaux de neurones sur silicium à l'aide de synapses memristives : application au traitement d'image

Charly Meyer

université de BORDEAUX

THÈSE PRÉSENTÉE

POUR OBTENIR LE GRADE DE

DOCTEUR DE L'UNIVERSITÉ DE BORDEAUX

ÉCOLE DOCTORALE : Sciences physiques et de l'ingénieur

SPÉCIALITÉ : Électronique

Par Charly Meyer

Conception de réseaux de neurones sur silicium à l'aide de synapses memristives : application au traitement d'images

Sous la direction de : Sylvain SAIGHI
(co-encadrant : Jean TOMAS)

Soutenue le 08/07/2021

Membres du jury :

M. UHRING Wilfried	Professeur des Université - Université de Strasbourg	Rapporteur
M. QUERLIOZ Damien	Chargé de recherche - Université Paris Saclay	Rapporteur
Mme GROLLIER Julie	Directrice de recherche - Université Paris Saclay	Examinateurs
M. LARRAS Benoit	Maître de conférences - Junia	Examinateurs
M. TARIS Thiery	Professeur des Université - Bordeaux INP	Examinateurs
M. SAIGHI Sylvain	Professeur des Universités - Université de Bordeaux	Directeur de thèse
M. TOMAS Jean	Maitre de conférences - Université de Bordeaux	Co-encadrant

À mes parents,
à mon épouse.

Remerciements

J'aimerais tout d'abord remercier mon directeur de thèse Sylvain Saïghi et mon co-encadrent Jean Tomas pour m'avoir fait confiance pour mener ces travaux de recherches. Ils m'ont également épaulé, encouragé et m'ont fait sortir de mes retranchements afin que j'atteigne les connaissances et compétences acquises durant cette thèse.

Je remercie les membres du jury, Julie Grollier, Thiery Taris et Benoit Larras d'avoir été présents pour ma soutenance, et plus particulièrement mes rapporteurs de thèse Damien Querlioz et Wilfried Uhring pour leur temps consacré à l'étude mon manuscrit.

Je remercie Thomas Finateu et Etienne Reynaud qui font partie de l'équipe de conception de Prophesee. Ils étaient d'une grande aide à la réalisation de la puce Hermes, ils m'ont encouragé et donné de précieux conseils techniques.

Je remercie Farad et Pierre doctorants de notre équipe qui m'ont beaucoup aidé dans mes travaux, et pour les bons moments passés ensemble.

J'ai eu la chance d'avoir baigné dans un groupe de recherche multidisciplinaire avec des spécialistes en biologie et en électronique. Les échanges et discussions que nous avons eus ont été intéressants et fructueux. L'ambiance de notre groupe a également été extraordinaire. Je remercie donc Sylvie, Noëlle, Yannick, Gilles, Timothée, Adrien, Isabelle, Yann, mais également les doctorants Clémence, Anne, Marie, Lorenza, Romain, Loïc, Alexandre, Yann, et nos anciens doctorants, Antoine, Amélie, Jonathan et Hermanus.

Enfin pour finir je remercie tous les personnels du laboratoire qui nous ont permis de réaliser nos activités de recherches dans les meilleures conditions possible, en particulier Valérie Abel notre support administratif, pour son aide et qui a toujours trouvé les bons mots pour nous mettre de bonne humeur.

Résumé

L'intelligence artificielle est une technologie en plein essor et utilisée de plus en plus dans diverses applications. Leurs algorithmes sont très gourmands en énergie. Pour des enjeux environnementaux et sociétaux, il est indispensable de réduire la consommation de ces algorithmes. Pour cela il est nécessaire de s'affranchir de la classique architecture de von Neumann. Le projet européen ULPEC dont cette thèse fait partie, a pour but de concevoir un réseau de neurones évènementiels à base de synapses memristives inspiré du fonctionnement du cerveau qui est par nature économe en énergie. Le memristor est une résistance variable commandable en tension qui peut, suivant les conditions d'utilisation, avoir le même comportement que des synapses biologiques.

Le projet ULPEC consiste à concevoir une puce sur laquelle est présent une caméra évènementielle, un crossbar de memristors de 784x100 et un réseau de neurones analogiques. Le réseau de neurones est constitué de 784 neurones d'entrées et 100 de sorties. Les neurones d'entrées permettent de transformer un évènement venant de la caméra en un signal capable d'être lu par le réseau de neurones. Les neurones de sorties permettent la reconnaissance d'images spécifiques en modifiant la valeur des résistances des memristors afin de réaliser le processus d'apprentissage.

Les neurones utilisés sont modélisés par un LIF (Leaky Integrate and Fire en anglais). Ce modèle a pour particularité d'être proche de la biologie et le grand avantage d'être facilement implantable électroniquement. Nous utiliserons un convoyeur de courant dans nos neurones post-synaptiques, ce qui permet de piloter aisément les synapses memristives. Nous avons réalisé des simulations proches du comportement physique de notre réseau et nous avons obtenu 67% de reconnaissance sur une base de données composée de chiffres manuscrits filmés par une caméra évènementielle.

Ce manuscrit de thèse a pour but d'explorer la faisabilité d'un tel projet en se focalisent sur la conception des neurones en pointant les problèmes rencontrés et les résolvant dans la mesure du possible. Ces travaux ont permis d'explorer la faisabilité du premier réseau de neurones à base de synapses memristives de cette envergure et permettront de donner des pistes de recherche pour de futurs projets.

Mots clés

Réseau de neurones évènementiels ; Memristor ; Faible consommation ; Caméra évènementielle ; Apprentissage non supervisé.

Abstract

Nowadays, the artificial intelligence is a technology used more and more in diverse features. The usage of their algorithms need a huge consummation of energy, for environmental and societal issues it is necessary to reduce their power consumption. The classical von Neumann architecture used in traditional computer is not efficient for artificial intelligence algorithms in terms of energy and calculation speed. The European project ULPEC which this thesis takes part, has in aim to design an ultra-low power bio-inspired neural network based on memristive synapses and an event-based camera. Memristors which play the role as synapses, are variable resistors controllable by voltage at their terminals.

The purpose of this project is to make a chip embedding an event camera, a matrix of 784x100 memristors and design the analog neural network to achieve the best learning as possible. The neural network is composed of 784 input neurons and 100 output neurons, where each input neuron is directly connected to a single one pixel from the camera. Output neurons make the images recognition and modify the weight of their synapses to realize the learning.

Neurons are modelized by a LIF neuron (Leaky Integrated and Fire), this model is close to the biology and has the advantage to be easy to design and implemented by electronics. We use a current conveyor into the output neuron to implement this model. We have built a simulator close to the physics of the neural network, and we have obtained 67% of recognition on handwritten figures database.

This thesis has aimed to explore the feasibility of the neural network base on synapses memristive realization to highlight the issue and solve it with the technical solution when it is possible. These works allowed to explore the feasibility of this project and give clue for future projects.

Keywords

Spiking neural networks; Memristor; Low power; Event camera; Unsupervised learning

Unité de recherche

[UMR, 52 18, 351 cours de la libération 33405 Talence]

ABREVATION

AER : Protocole de communication (Address Event Representation)

BTO : Memristor ferroélectrique de structure cristalline (Béryllium Titane Oxygène)

CCII : Convoyeur de courant de seconde génération

CNN : Réseau de neurones convolutionnel (Convolutional Neural Network)

DCB : Le contrôleur numérique de notre puce

IA : Intelligence artificielle

LIF : Modèle de neurone signifiant Leaky Integrate and Fire

MNIST : Base de données de chiffres manuscrits

N-MNIST : Base de données de chiffres manuscrits filmés par une caméra événementielle

SNN : Réseau de neurones évènementiels (Spiking Neural Network)

STDP : Spike Timing Dependent Plasticity

ULPEC : Projet européen Ultra-Low Power Event-based Camera

TABLE DES MATIERES

INTRODUCTION

Le machine learning (apprentissage automatique en français) a depuis les années 40 où le premier réseau de neurones a été capable d'apprendre par lui-même [MCCULLOCH W. S 1943] intéressé les chercheurs. Cet intérêt a perduré jusqu'à nos jours et s'et même renforcé pendant la dernière décennie. Les réseaux de neurones sont devenus au cours du temps de plus en plus performants notamment grâce à l'augmentation de la puissance des ordinateurs, l'augmentation de l'efficacité des algorithmes et à l'accès à une grande collection de données. Ils se retrouvent dans tous les domaines de la vie courante et professionnelle. Ils sont capables de résultats exceptionnels, au-delà de la capacité du cerveau humain dans des domaines bien précis tels que de faire un diagnostic médical ou faire de la reconnaissance d'objet à partir d'images. Notre vie à l'avenir sera probablement bouleversée par l'intelligence artificielle qui s'installera encore plus et de façon durable dans le monde professionnel et à la maison, afin de libérer du temps pour les tâches les plus créatives en suppriment les tâches rébarbatives [SHABBIR J. 2015].

Cependant les algorithmes de machine learning nécessitent de faire une quantité colossale de calculs. Ce traitement massif des données entraîne une augmentation considérable des ressources énergétiques nécessaires pour réaliser ces applications, mais également de grandes quantités de ressources pour la communication des appareils vers les centres de calcul. Leur utilisation est nécessaire, car les appareils mobiles ne disposent pas aujourd'hui de l'autonomie et des capacités pour embarquer seuls ces algorithmes d'intelligence artificielle.

Le cerveau, malgré sa faible puissance de 20W, est capable de réaliser des tâches complexes de la vie quotidienne que même les intelligences artificielles utilisant des super calculateurs ne peuvent reproduire qu'avec difficulté, ne serait-ce que pour établir un dialogue et de comprendre toutes ses subtilités par exemple. Afin d'abaisser la consommation énergétique et de pouvoir réaliser des tâches aussi complexes que celles du cerveau, on peut s'inspirer de son fonctionnement pour réaliser un autre type de plateforme matérielle dédiée à l'intelligence artificielle. Les neurones artificiels à l'instar des neurones biologiques y codent leurs informations par des évènements. Cependant ce nouveau paradigme de calcul n'est pas celui utilisé à l'heure actuelle par les algorithmes les plus efficaces qui sont basés sur des réseaux de neurones formels.

De nombreuses et fructueuses recherches ont été faites dans les réseaux de neurones bio-inspirés montrant principalement en simulation une diminution de la consommation énergétique par rapport aux réseaux de neurones formels. Ils ont également la propriété comme le cerveau de pouvoir faire un apprentissage sans supervision, contrairement aux réseaux classiques qui nécessitent une grande quantité de données labellisées, en général faites « à la main ». Les réseaux de neurones évènementiels

sont potentiellement plus efficaces sur des architectures matérielles dédiées que sur un ordinateur avec une architecture classique de type von Neumann. L'architecture classique des ordinateurs ne permet pas de faire des calculs massivement parallèles, contrairement au cerveau biologique.

Parallèlement à cette approche en rupture des architectures de calculs dédiées à l'intelligence artificielle, cette dernière décennie a vu l'émergence de nano composants appelés memristors. Les memristors sont des résistances programmables par tension et non volatiles. Leur plasticité, leur capacité à modifier la résistance suite à l'application d'une tension, leur confère un comportement similaire aux synapses biologiques, ce qui en fait d'excellents candidats pour les réseaux de neurones évènementiels bio-inspirés matériels. Ces réseaux à base de synapses memristives sont à ce jour très étudiés, mais les réalisations physiques se font encore attendre.

Ces travaux de thèse se placent dans le contexte du projet européen ULPEC (ICT 03-2016 Smart System Integration). L'objectif d'ULPEC est de réaliser un démonstrateur composé d'une caméra événementielle ultra basse consommation, d'un réseau de neurones CMOS et d'un réseau de memristors ferroélectriques assemblés sur le circuit silicium par une technique de report 3D. Le système sera exploité pour la reconnaissance des panneaux de signalisation routière pour l'aide à la conduite voire même pour les véhicules autonomes (https://ulpecproject.eu/). Mes travaux de thèse ont pour but d'étudier la faisabilité de réaliser un apprentissage non supervisé du démonstrateur ULPEC et de concevoir la puce système à l'exception de la caméra événementielle dont le layout a été fourni par Prophesee, partenaire du projet ULPEC.

Le manuscrit est organisé en quatre parties suivies par une conclusion. Le premier chapitre est consacré aux généralités des réseaux de neurones matériels et de la caméra événementielle. Le deuxième chapitre se focalise sur les memristors et leurs utilisations comme synapses. Le troisième chapitre s'intéresse à la conception électronique des blocs constituant la partie silicium du réseau de neurones. Enfin le quatrième chapitre fera l'objet de la puce système composée de la caméra, du réseau de neurones et de la logique de pilotage du système autorisant l'apprentissage et l'inférence. La puce doit être en capacité de recevoir le report de la matrice de memristors avec ses 884 connexions.

CHAPITRE 1 : ETAT DE L'ART

1.1 Introduction

Les réseaux de neurones sont performants et capables de réaliser des tâches plus rapidement que les hommes dans de nombreux domaines, mais sont énergivores. Avec l'apparition de l'intelligence artificielle (IA) dans de plus en plus de secteurs, la consommation en énergie de l'IA devient un enjeu majeur de la recherche. Pour atteindre les performances de l'état de l'art tout en réduisant les coûts énergétiques, l'architecture de von Neumann doit être abandonnée au profit de nouveaux paradigmes de calcul tel que les architectures neuromorphiques.

Le cerveau ne dépense qu'une vingtaine de Watt, mais est d'une telle efficacité qu'aucune IA ne peut l'égaler dans les domaines de la vie quotidienne. Pour une réduction de consommation, de nombreuses architectures se sont inspirées du cerveau. Ces réseaux appelés réseaux de neurones évènementiels peuvent être couplés à des capteurs dédiés bio-inspirés fonctionnant en synergie, qui eux aussi sont conçus dans le but de réduire drastiquement leur consommation.

Dans ce chapitre il y sera montré pourquoi les réseaux de neurones évènementiels sont très prometteurs et méritent d'être explorés. Il y sera présenté différents types de réseaux de neurones bio-inspirés. Dans notre projet il a été choisi de réaliser un réseau de neurones à base de synapses memristives, il sera brièvement expliqué dans ce chapitre leur fonctionnement.

1.2 Les réseaux de neurones conventionnels

Les réseaux de neurones sont aujourd'hui largement répandus dans de multiples applications que ce soit pour de la classification de données, de la reconnaissance d'image ou vocale, utilisés dans des disciplines aussi diverses que la biologie, la médecine l'environnement ou l'énergie [HAO L. 2017]. Les réseaux de neurones peuvent être considérés comme une boite noire ou l'on met en entrée par exemple des photos d'animaux, des radiographies de patients ou des enregistrements vocaux et ils classifient en sortie s'il s'agit d'un chien ou d'un chat, si le patient a ou non un cancer ou nous donnent encore la phrase reconnue. Pour en arriver à une telle efficacité, il a fallu que les réseaux de neurones apprennent en leur fournissant des données en entrée et qu'ils se renforcent en se corrigeant au gré de leur réussite ou erreur.

Les réseaux de neurones sont composés de synapses et de neurones. Le fonctionnement d'un neurone est illustré à la (figure 1.1). Son but est de sortir une valeur y_i en fonction de l'addition des entrés x_i multipliés par les poids respectifs w_i de chaque synapse. L'apprentissage permet de modifier chaque poids w_i individuellement en fonction des erreurs et des entrées x_i, pour maximiser la sortie. La fonction

non linéaire f du neurone varie selon l'application, elle peut être binaire +1 ou -1, y_j vaut par exemple 1 si la somme $\sum_{i=i}^{n} x_i w_i$ est supérieure à un seuil [WIDROW B. 1990]. Les fonctions softmax et sigmoïde sont également très répandues et font office de fonction f sur la (figure 1.1) [SHIBATA, K 1999].

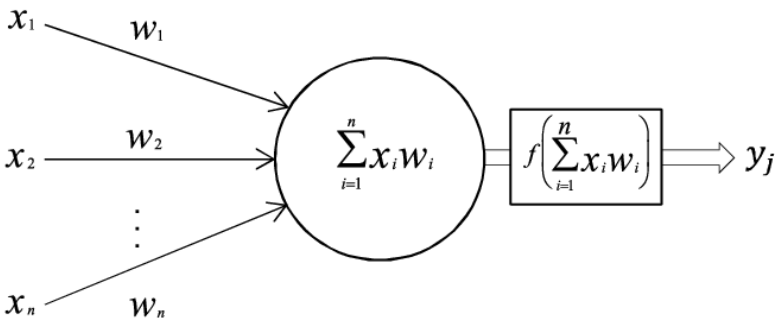


Figure 1.1 Fonctionnement d'un neurone : sortie en fonction de la somme des produits synaptiques w_i et de leur entrée respective x_i.

La classification linéaire pour un perceptron se fait très simplement avec un seul neurone et deux entrées la valeur en x_1 pour les ordonnées et en x_2 pour les abscisses comme montré sur la figure 1.2-a). On ajoute à ce neurone de la (figure1.2-c) une constante B avec un poids w_B pour qu'il puisse plus facilement apprendre et trouver une droite qui sépare les deux échantillons. Si le problème est linéaire (figure1.2-a), on peut résoudre le problème avec un seul neurone et trouver une droite en modifiant les poids w_1, w_2 et w_B, qui ont pour but de modifier la pente et sa valeur à l'origine. Dans cet exemple la valeur des poids w_1=-1, w_2=1 et w_B=0 convient pour la séparation de la droite.

En revanche, les problèmes non linéaires comme sur la (figure 1.2-b), ne peuvent pas être résolus par un seul neurone. La solution est d'utiliser plusieurs couches successives de neurones, appelées couches cachées.

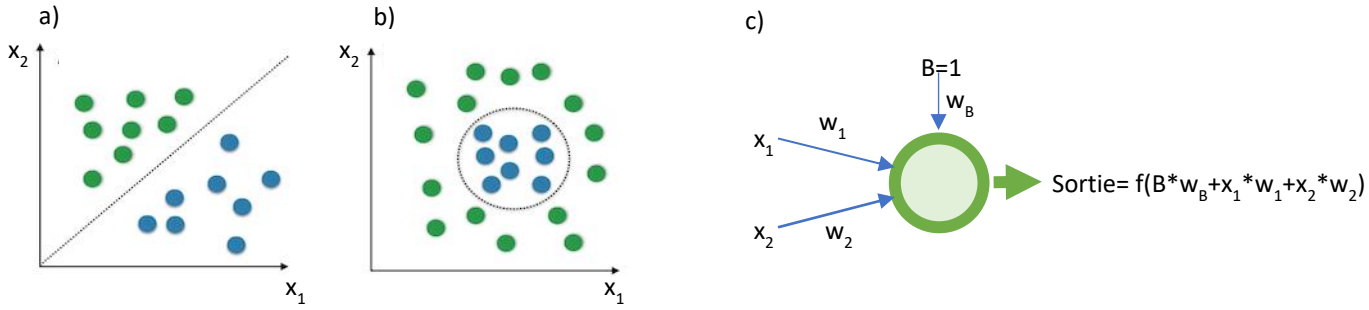


Figure 1.2 : a) problèmes linéaires, b) problèmes non linéaires. c) neurone à deux entrées.

Un réseau de neurones est un ensemble de neurones connectés entre eux. Le perceptron fait partie de l'un des premiers réseaux de neurones à être utilisé dans les années 1960 dans le but de faire de la classification de données [WIDROW B. 1990]. Le perceptron est un réseau de neurones composé d'une ou de plusieurs couches cachées, où toutes les sorties des neurones sont connectées à toutes les entrées des neurones de la couche supérieures. Le principe de ce réseau est de faire circuler l'information uniquement dans un seul sens, de l'entrée vers la sortie. La (figure 1.3) illustre un Perceptron à deux couches cachées. Pour permettre un apprentissage et réussir la classification voulue, il faut modifier les poids synaptiques du réseau. Pour modifier les poids synaptiques, on doit entrainer le réseau à reconnaitre ce que l'on souhaite en choisissant quel neurone de sortie doit être activé par le réseau pour chaque donnée à classifier.

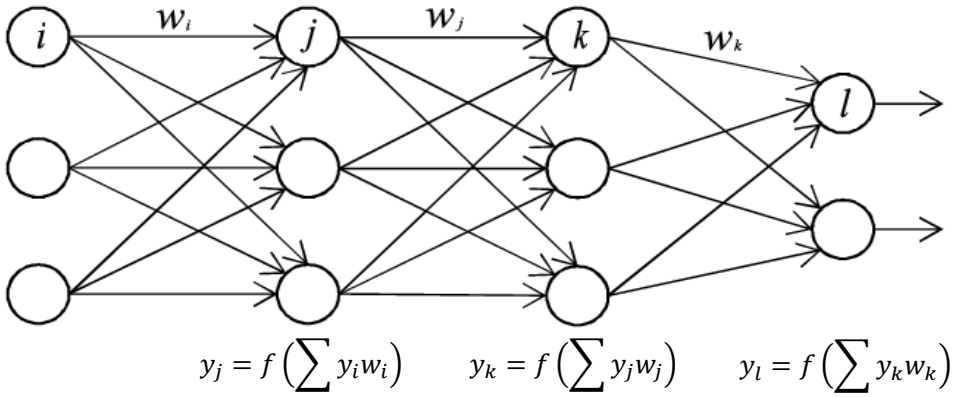


Figure 1.3 : Réseau de neurones à deux couches cachées

Pour modifier le poids du réseau de neurones on utilise un algorithme dit de « back-propagation » [WIDROW B. 1990] qui consiste à comparer le résultat obtenu à celui attendu et à modifier les poids des synapses du neurone de sortie en question en utilisant un algorithme de descente de gradient, dans le but de diminuer cet écart, l'erreur est également propagée couche par couche de la sortie vers l'entrée. Afin d'éviter de tomber dans un minimum local avec l'algorithme de back-propagation, une solution est d'utiliser un algorithme génétique qui consiste à créer d'autres réseaux de neurones avec des poids bien différents, et de garder celui qui répond le plus à nos attentes [SEIFFERT U. 2001]. Cet algorithme de back-propagation a permis l'émergence des réseaux de neurones et demeure largement utilisé.

ImageNet est une base de données recueillant des millions d'images labellisées par catégorie comme le sport, les animaux ou les fruits, voir (figure 1.4). Ce recueil d'images permet d'entrainer des algorithmes afin de réaliser de la reconnaissance d'image et de les comparer aux autres algorithmes réalisés par les chercheurs du monde entier. Jusqu'en 2012 le taux d'erreur de classification était supérieur à 25% pour la base de données ImageNet. C'est pour cela que les premiers réseaux ont été abandonnés au profit des

réseaux de neurones convolutionnels (CNN). Les premiers CNN ont conduit à 15,3% d'erreur en 2012 qui a diminué d'année en année pour atteindre 3,6% en 2015 et 2,3% en 2017. L'augmentation du nombre de couches cachées est prépondérante. À titre de comparaison, le taux d'erreur commis par l'humain à la reconnaissance de ces séries d'images est de 5,1%, ce qui permet de mettre en avant l'efficacité des réseaux de neurones pour la reconnaissance d'image.

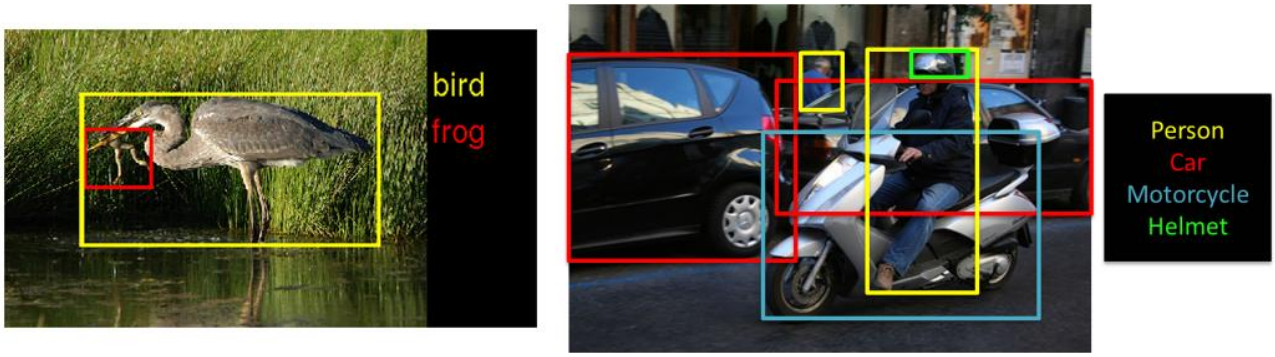


Figure 1.4 : Exemple d'images labellisées de ImageNet.

Dans les réseaux de neurones convolutionnels, les neurones ont la même fonction, mais le changement majeur réside dans la connectivité du réseau. La sortie d'un neurone d'une couche n'est plus connectée à tous les neurones de la couche suivante comme pour le perceptron, seule une partie de l'image est connectée aux neurones suivants. Il y a bien entendu un balayage de toute l'image pour que chaque zone soit recouverte (voir figure 1.5). Le but de ces réseaux est de reconnaitre des petits motifs récurrents de l'image comme des lignes ou des courbes [LECUN Y. 1998]. La méthode d'apprentissage reste identique en utilisant la back-propagation et en diminuant ou renforçant le poids des synapses concernées. Dans une optique d'une réduction du temps de calcul, une variante consiste à utiliser des filtres de Gabor plutôt que des synapses à poids variables pour réaliser les convolutions. Ces filtres sont fixes et comprennent déjà les motifs à reconnaitre et permettent ainsi d'éviter de calculer le poids des synapses [SHAKIB S. 2017].

Il existe une multitude d'autres architectures de réseaux de neurones, qui ont chacune d'entre elles une spécificité distincte pour différentes applications. Nous ne les présenterons pas ici, car elles nous éloigneraient du propos de ce manuscrit de thèse.

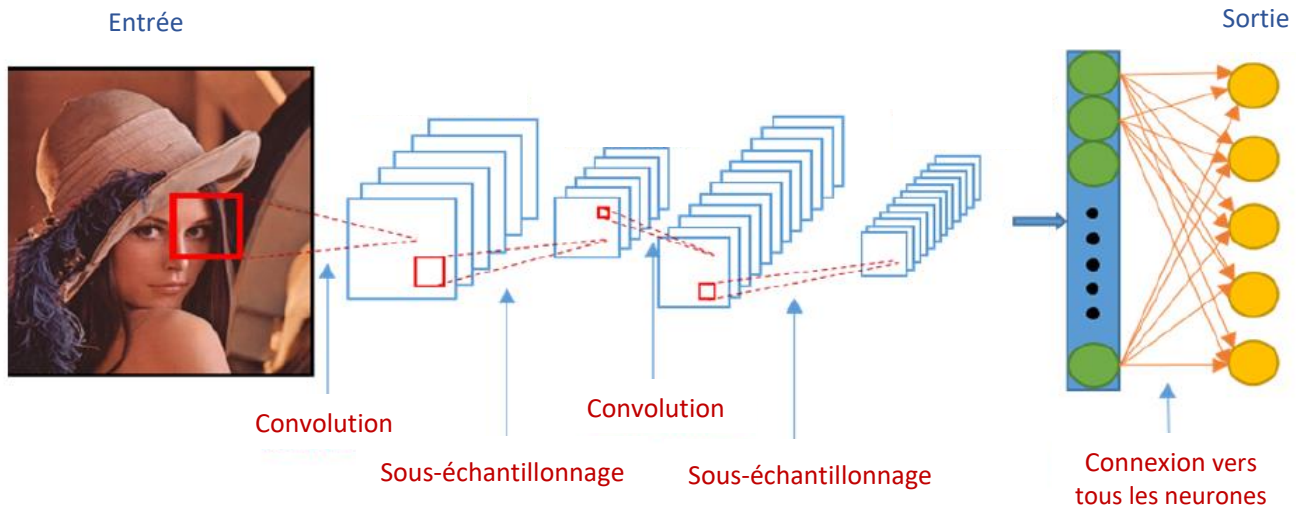


Figure 1.5 Architecture d'un réseau de neurones convolutionnels.

1.3 Les problèmes rencontrés

Il y a aujourd'hui environ 30 milliards d'objets connectés consommant chacun de l'énergie pour sa propre alimentation, mais aussi pour la connexion au réseau. La régulation de la consommation énergétique des appareils mobiles est primordiale pour la durée de vie de la batterie. Pour augmenter la durée de vie de cette batterie certains calculs sont déportés vers des data center. Aujourd'hui la majeure partie de l'énergie dépensée se passe durant la mise en veille des appareils en maintenant les mémoires volatiles actives, mais est également dépensée dans les convertisseurs analogiques numériques [CAPRA M. 2019]. Ce sont deux problèmes que les réseaux de neurones événementiels à base de synapses memristives peuvent résoudre.

Au fil des années, les appareils mobiles ont tendance à voir leurs performances augmentées tout en consommant moins. La loi de Moore a permis de faire des avancées significatives en termes de performance et de réduction de consommation par la diminution de taille des transistors. La réduction de la taille des transistors est certes toujours envisagée, mais n'apporte plus les mêmes bénéfices, entre la performance et la consommation un compromis doit être fait [BOHR M. T. 2017] [KRESTINSKAYA O. 2019]. Pour atteindre une augmentation de performance, mais également une réduction de la consommation énergétique, un changement de paradigme dans l'architecture des organes de calcul est nécessaire.

L'électronique en général et en particulier les objets de technologie de l'information et de la communication (TIC) comme les objets connectés, les téléphones mobiles, les ordinateurs ou les

téléviseurs ont une influence importante dans la consommation d'électricité mondiale [JONES N. 2018]. Il est même prévu qu'en 2030 20,9% de la consommation énergétique y soit consacré. L'électricité consommée par les data centers et la communication vers ceux-ci occupe une grande part de l'électricité mondiale et continue d'augmenter (voir figure 1.6). L'augmentation des flux d'informations dans le réseau et data centers est due en partie à l'augmentation des assistants personnels intelligents tels que Siri de Apple, Google Now ou encore Cortana de Microsoft. Ils sont utilisés pour effectuer des ordres ou répondre à une question posée de vive voix par l'utilisateur. Ils doivent être capables d'associer un son à un mot et de comprendre la sémantique de la phrase prononcée. Ce sont des réseaux de neurones profonds d'une grande complexité qui sont utilisés pour ce genre d'application. Les calculs sont réalisés sur des serveurs informatiques, car ils sont trop complexes pour être fait depuis un téléphone ; les ressources énergétiques mises en jeu et le temps de réponse seraient trop importants [KANG Y. 2017].

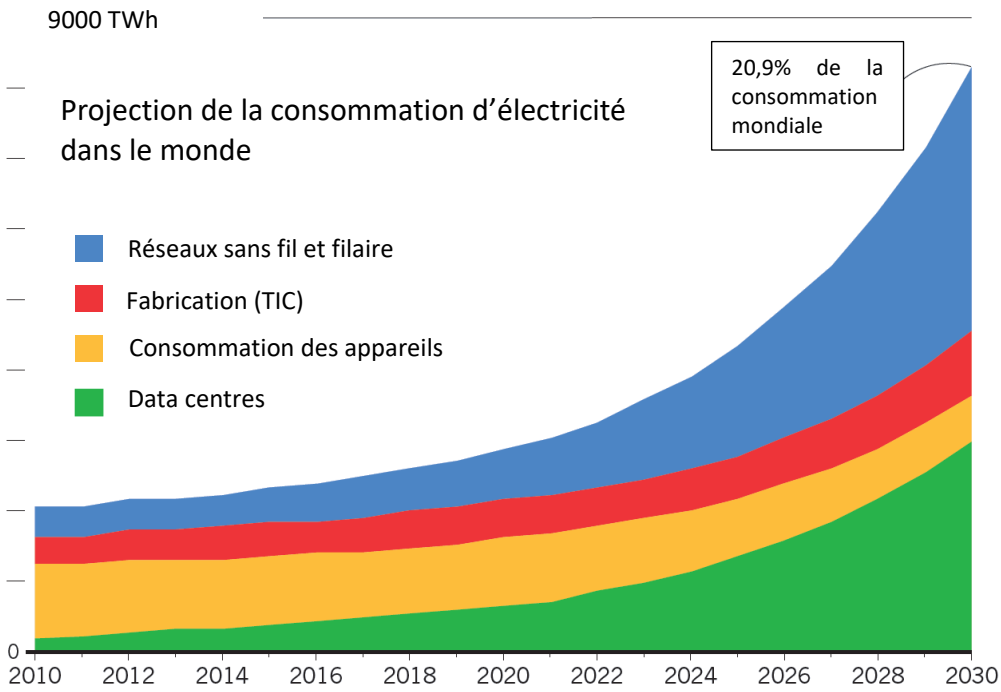


Figure 1.6 : Projection de la consommation mondiale d'électricité liée à l'électronique (TIC) [JONES N. 2018]

De plus en plus d'appareils sont connectés au « cloud » et non uniquement pour les applications liées à l'IA, ce qui crée une saturation de la communication vers les serveurs. Avec le nombre grandissant des appareils connectés, le problème risque de s'aggraver. L'arrivée de la 5G devrait réduire le phénomène de congestion [YU W. 2017], mais ne résoudra pas le problème de la consommation énergétique lié à cette augmentation de trafic.

Depuis la fin des années 70 la question de s'affranchir de la classique architecture de von Neumann afin augmenter la performance des IA se posait déjà [PELAEZ E. 1990], et il devient aujourd'hui nécessaire de remettre en question cette architecture au profit d'autres plus efficaces. Les réseaux de neurones ont une particularité intéressante en vue d'une implémentation matérielle à savoir que chaque neurone peut être

traité indépendamment des autres, ce qui autorise intrinsèquement les calculs parallèles. Or un processeur classique est une unité de calcul qui réalise les opérations l'une après l'autre, ce qui n'est donc pas adapté à un calcul massivement parallèle. De plus dans les architectures classiques, l'unité de calcul et la mémoire sont séparées par un bus de communication, la vitesse maximale est donc limitée par la communication entre ces deux blocs. La vitesse de calcul est d'autant plus impactée lorsque l'on échange beaucoup d'information avec la mémoire, ce qui est le cas pour les réseaux de neurones où chaque poids synaptique doit être recherché en mémoire [MEROLLA P. A. 2014]. Les processeurs utilisent pour pallier ce problème plusieurs niveaux de mémoire cache, mais ce n'est pas suffisant pour les réseaux de neurones. La (figure 1.7-a) présente le goulot d'étranglement causé par l'architecture de von Neumann au niveau de la communication entre le CPU et la mémoire qui supportent respectivement le calcul des neurones et les poids synaptiques. La (figure 1.7-b) présente une architecture distribuée dans l'espace des unités de calcul et de la mémoire semblable à celle du cerveau. Un avantage des réseaux de neurones est que le poids des synapses mis en mémoire est utilisé localement, ce qui rend l'architecture de von Neumann classique inutile.

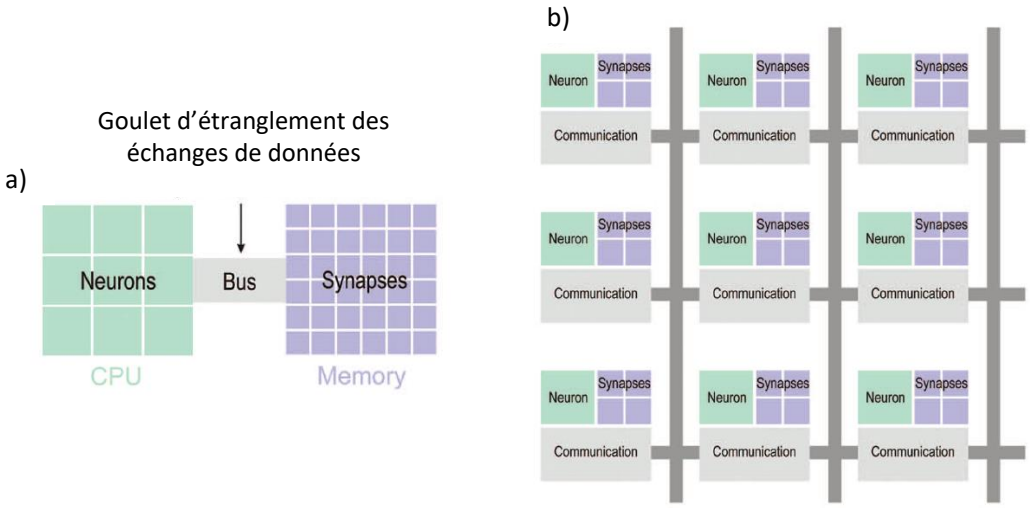


Figure 1.7 : a) Architecture de von Neumann avec son goulet d'étranglement au niveau du bus de communication entre le CPU et la mémoire. b) Architecture conceptuelle où les synapses sont au plus proche des neurones comme dans le cerveau.

Pourquoi faire du neuromorphique ?

Le cerveau a un fonctionnement particulier : chaque neurone communique via des potentiels d'actions qui représentent des évènements. Les réseaux de neurones bio-inspirés sont appelés réseaux de neurones évènementiels ou « spiking neural network » (SNN en anglais). Un autre avantage du cerveau est qu'il apprend seul, il n'y a pas besoin d'aide extérieure pour modifier le poids de ses synapses, cette technique est appelée apprentissage non supervisé.

Depuis l'aube de l'informatique, dans les années 1950 l'un des pères fondateurs de l'ordinateur, A. Turing, avait déjà pensé utiliser une machine dédiée au calcul neuromorphique [TURING A. M. 1950]. Depuis ce temps, l'IA a fait l'objet de recherches avec un net intérêt depuis la dernière décennie. L'équipe de J. S. Plank a consulté plus de 3000 publications portant sur les plateformes matérielles dédiées aux réseaux de neurones évènementiels depuis les années 80 [SCHUMAN C. D. 2017], ils ont fait un excellent travail de synthèse en montrant les objectifs, les technologies utilisées et leurs applications. Dans cet article, il oppose l'architecture de von Neumann classique aux machines matérielles conçues pour le calcul neuromorphique. Il est montré qu'une grande part des dispositifs matériels ont pour objectif de réduire la consommation énergétique, ou d'augmenter la performance de calcul qui inclut calcul temps réel, parallélisme et vitesse, comme illustré dans la (figure 1.8). D'autres objectifs secondaires sont liés à la réalisation de ce type d'architecture comme pouvoir être plus tolérant aux défauts, avoir la possibilité de faire des simulations du cerveau humain, mais aussi trouver une architecture différente de celle de von Neumann.

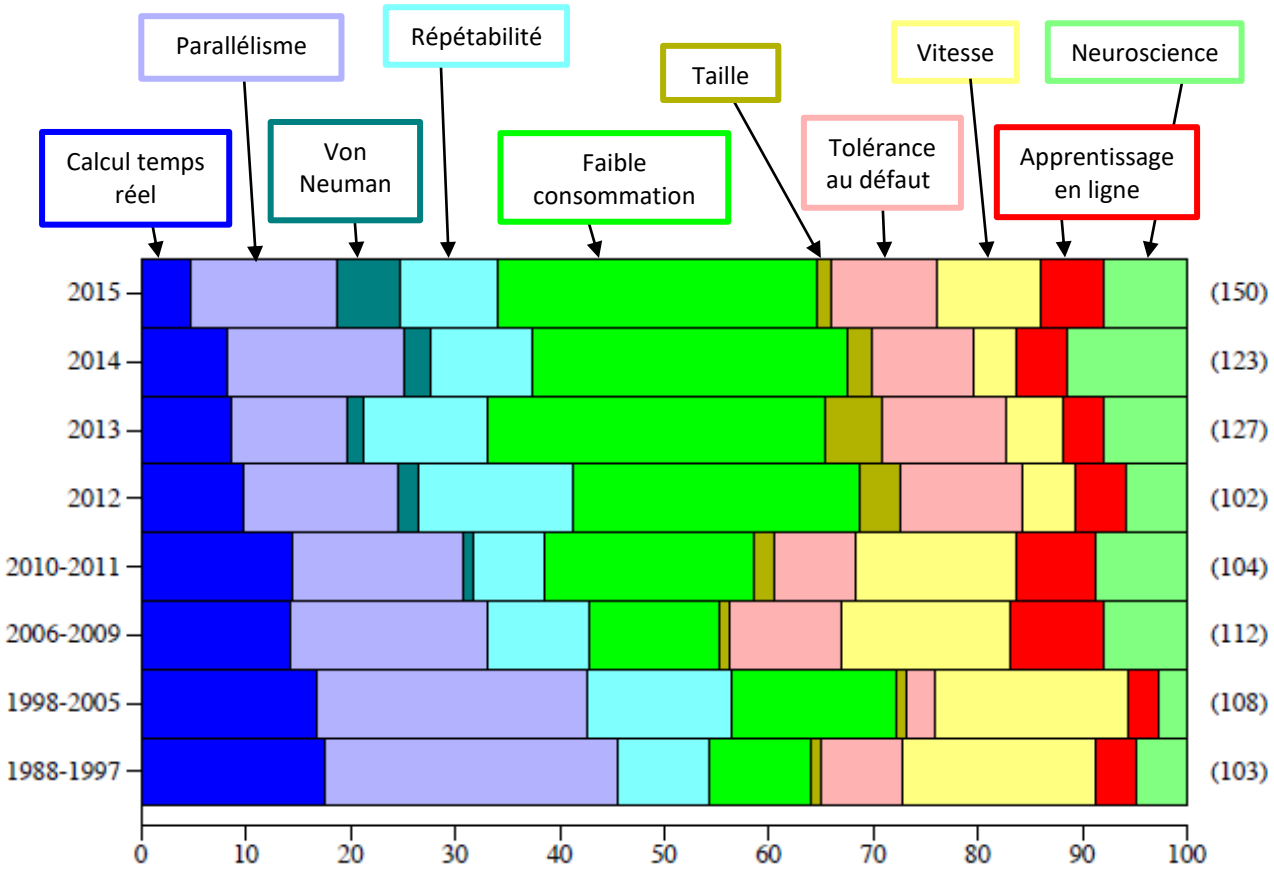


Figure 1.8 : Répartition des objectifs de créer une plateforme neuro-morphique hardware dédiée en fonction des années et du nombre de publications par année [SCHUMAN C. D. 2017]

Une autre caractéristique intéressante des réseaux de neurones évènementiels est qu'ils ont un fonctionnement et un système d'apprentissage semblable au cerveau, qui a la particularité de faire un apprentissage non supervisé.

Bien que les SNN sur des plateformes matérielles dédiées puissent être encore améliorés, notamment en trouvant des algorithmes plus performants, ils sont devenus populaires pour des raisons d'efficacité et surtout de leur potentiel de faible consommation énergétique comparée à leurs équivalents sur des ordinateurs classiques utilisant l'architecture de von Neumann [SCHUMAN C. D. 2017].

1.4 Un réseau de neurones bio inspiré

Avec ses 90 milliards de neurones pour seulement 20 W, le cerveau présente un ratio capacité de calcul versus consommation énergétique inégalé. C'est pourquoi de nombreux chercheurs travaillent à l'élaboration d'un processeur dédié aux réseaux de neurones à potentiel d'action [YOUNG A. R. 2019]. À titre d'exemple, la simulation sur un ordinateur personnel d'une partie d'un réseau de neurones de souris comprenant 2,5 millions de neurones est 9000 fois plus lente et 40 000 fois plus gourmande en énergie que la réalité [VARKEY B. 2014]. La simulation entière d'un cerveau humain serait de l'ordre d'un demi gigawatt. Le neurone est composé d'un corps appelé soma et de branches de connexion appelées dendrites pour sommer les signaux afférents (voir figure 1.9). Les signaux qui parcourent le neurone s'appellent potentiels d'action ou spike en anglais. C'est un signal électrique d'une tension crête à crête d'environ 120mV. Le potentiel d'action est un phénomène tout ou rien qui se déclenche quand on atteint la tension de seuil. Le soma possède une fuite qui permet au potentiel de retrouver un état d'équilibre. Autrement dit, si le neurone cesse d'être stimulé, il finira par « oublier » ce qu'il vient de recevoir comme information. Cette représentation du fonctionnement du neurone est modélisée par le neurone LIF « Leaky Integrated and Fire » [ABBOTT L. F. 1999], qui est une approximation grossière du modèle du fameux modèle de Hodgkin et Huxley [HODGKIN L. 1952]. La synapse réalise la jonction entre un neurone et les dendrites du suivant. Le poids de la synapse varie en fonction de l'activité des neurones, c'est ce que l'on appelle l'apprentissage.

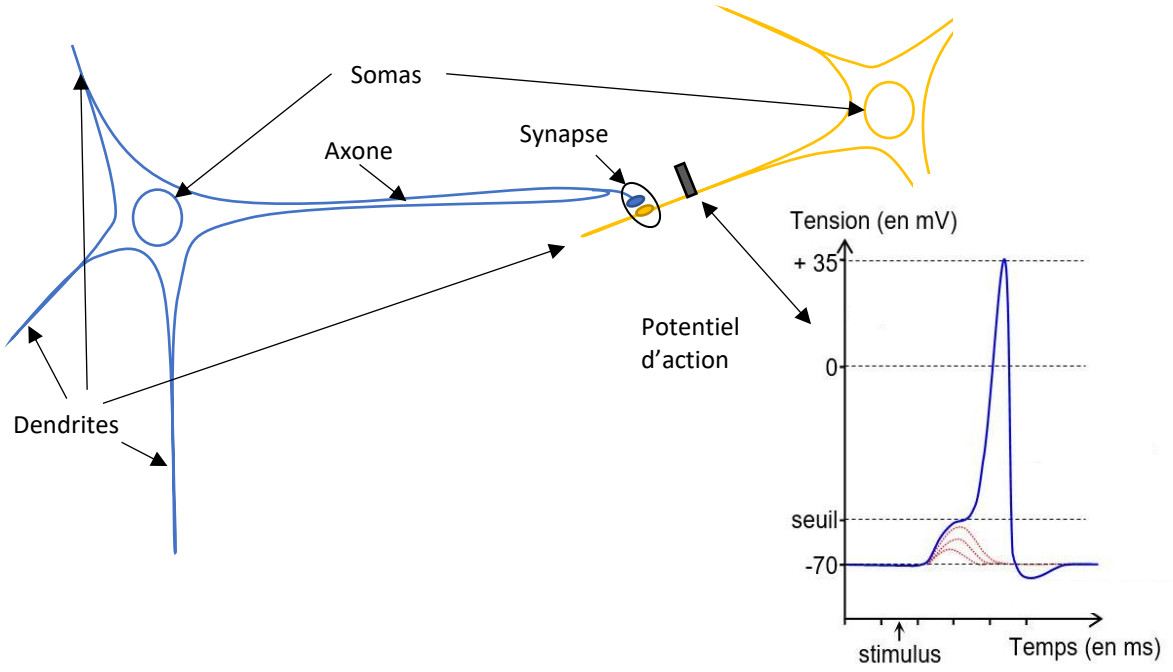


Figure 1.9 : Illustration de deux neurones connectés par une synapse et d'un potentiel d'action

Dans le but de limiter la consommation énergétique tout en augmentant la puissance des réseaux de neurones, de grands projets ont été lancés ces dernières années en créant un nouveau type d'architecture s'inspirant des principes cérébraux. Les puces TrueNorth, Neurogrid, SpiNNaker et BrainScaleS, et ont à eux quatre fait un grand pas dans l'avancement des processeurs neuromorphiques [BOUVIER M. 2019].

True north

True North est une plateforme neuromorphique développée par IBM. Elle est composée de 4096 puces neuro-synaptiques. Chacune de ces puces est composée de 256 neurones, de 256 entrées appelées axone et de 65 536 synapses connectées en matrice où chaque axone est relié à tous les neurones [AKOPYAN F. 2015]. Au total, True North contient 1 million de neurones et 256 millions synapses. Cette plateforme a été conçue pour les applications mobiles.

True North utilise des puces numériques en fonctionnement synchrone et asynchrone. Une horloge de 1 kHz interne à chacune des puces synchronise les évènements asynchrones arrivant sur le buffer. Pour des raisons de facilité d'implémentation et de performance, il a été choisi que les neurones utilisent le modèle LIF « Leaky Itegred and fire » plutôt que des modèles plus proches de la biologie. Bien qu'inspiré par l'architecture de von Neumann, le système est un mixte entre un fonctionnement synchrone et asynchrone, de plus la mémoire de chaque synapse est localisée dans chaque puce, il n'y pas de mémoire commune à tous les neurones.

Neurogrid

Cette plateforme permet de faire des simulations en temps réel de réseaux de neurones évènementiels pour simuler une activité cérébrale. Elle simule 1 million de neurones en temps réel en consommant moins d'une dizaine de Watt. Un ordinateur standard peut simuler 2,5 millions de neurones 9000 fois plus lentement pour une puissance d'une centaine de Watt. Cette performance est permise grâce à un circuit électronique dédié permettant le calcul analogique et contrôlé par une partie numérique. La configuration de Neurogrid est souple, elle permet également de simuler un réseau de neurones classique de type perceptron [VARKEY B. 2014].

SpiNNaker

Cette plate-forme a été développée pour partie dans le cadre du grand projet européen « Human Brain Project », dont le but est de faire une simulation du cerveau humain. La solution retenue est d'utiliser des puces contenant 18 processeurs ARM ayant en tout 128 MB de mémoire locale et partagée. L'échange de données est asynchrone. Les modèles de neurones possibles sont ceux d'Izhikevich et le LIF, ce qui permet une plus grande rapidité de calcul que des modèles plus complets décrivant la biologie du cerveau. La force de cette plateforme est la capacité de traiter et de transmettre de grandes quantités de données sur l'ensemble du circuit imprimé pour créer des connexions complexes à l'instar du cerveau. Le circuit est composé de 48 puces, ce qui fait un total de 864 processeurs ARM sur la même carte électronique. SpiNNaker est décliné en plusieurs versions de la plus simple qui contient une seule carte à la plus complexe qui en compte 1200 pour une puissance totale de 75 kW [FURBER S. B. 2014].

BrainScaleS

Le système BrainScaleS est le second du projet européen « Human Brain Project ». C'est un assemblage mixte de neurones analogiques et numériques permettant une simulation de 10 000 à 100 000 fois plus rapide que le cerveau. Pour cette plateforme le modèle LIF est utilisé avec des règles d'apprentissage programmables, mais a été conçu initialement pour un apprentissage par STDP (Spike Timing Dependent Plasticity). Une première version est composée de plusieurs puces constituées pour chacune d'une matrice de 32 x 32 synapses connectée à 32 neurones. Les synapses utilisées sont des résistances réglables [AAMIR S. A. 2018]. La seconde version a pour but d'augmenter la vitesse de calcul et de baisser sa consommation. Pour cela, la technique est d'utiliser une multiplication matricielle analogique qui permet de faire des milliers d'opérations simultanément. Cette puce contient 512 neurones analogiques qui sont connectés chacun à 256 synapses. Le système est piloté par des processeurs numériques. Des convertisseurs analogiques numériques sont utilisés pour récupérer le résultat du calcul analogique. Cette plateforme permet de réaliser de bon score de reconnaissance, en dépassant les 98% sur la MNIST, une base de données de chiffres manuscrits [WEIS J. 2020].

1.5 Caméra évènementielle

Le projet ULPEC vise à intégrer en plus d'un réseau de neurones memristifs une caméra évènementielle. Le réseau de neurones permettra de reconnaitre une série de chiffres manuscrits de manière automatique et non supervisée. La caméra évènementielle offre de nombreux avantages dont la basse consommation et l'absence de convertisseur analogique numérique qui consomme une grande part de l'énergie dans les caméras classiques [CAPRA M. 2019]. Selon sa configuration, elle peut traiter plus de 5000 « images » par seconde en acceptant une dynamique de luminosité de 140dB contre 60dB pour des caméras traditionnelles [REBECQ H. 2019].

Cette caméra évènementielle est inspirée du fonctionnement de la rétine en détectant la variation de luminosité. Contrairement aux caméras conventionnelles synchrones qui envoient à l'unité de calcul la totalité de l'image 25 fois par seconde, les caméras évènementielles fonctionnent différemment. Chaque pixel est indépendant et envoie un évènement de façon asynchrone uniquement si le pixel en question détecte une variation de luminosité. Si la variation de luminosité est positive un spike ON est déclenché, si la variation est négative c'est un spike OFF qui est déclenché. Sur la (figure 1.10 a) est montré un schéma du pixel. En b) est montrée l'évolution des signaux Vlog et Vdif. Vlog est l'image de la tension de la diode photoréceptrice, sa tension évolue logarithmiquement en fonction du courant I_{PH}., le courant est quant à lui proportionnel à la luminosité. La tension Vdif suit la tension Vlog, mais elle est remise à zéro quand elle atteint la tension « ON threshold » ou « OFF thershold » et la caméra émet respectivement des spike ON ou OFF du pixel concerné [POSCH C. 2014].

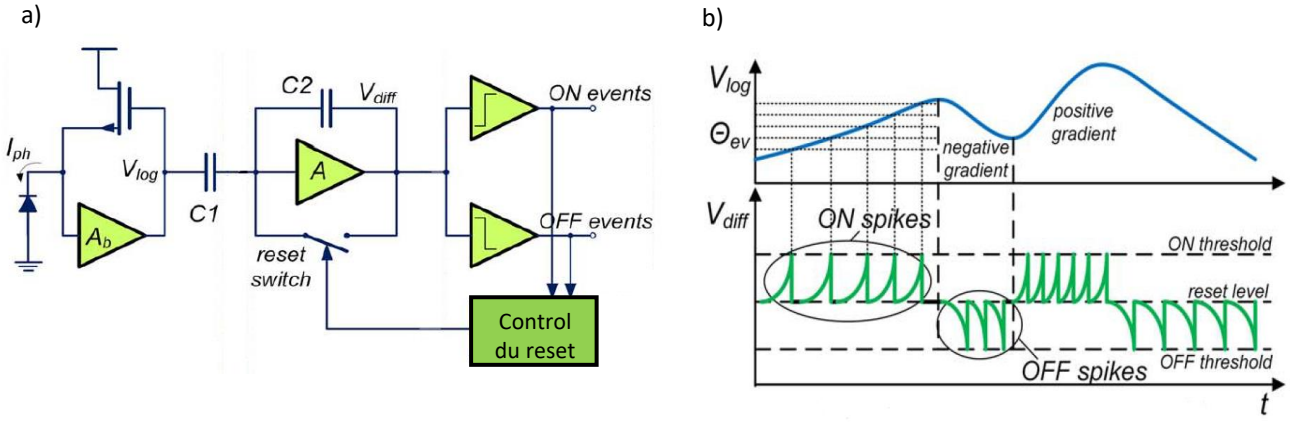


Figure 1.10 : a) Schéma du pixel de la caméra évènementielle b) Spike émis par la caméra évènementielle en fonction de la variation de luminosité.

Sur la (figure 1.11), on peut voir la différence entre une caméra standard en a) et la caméra évènementielle de Prophesee qui est partenaire du projet EU ULPEC en b). La caméra évènementielle ne détecte que les variations de luminosité, et ne prend donc que les objets en mouvement. Elle affiche en noir ou blanc les pixels qui ont reçu une variation de luminosité, et en gris celles qui n'ont rien reçu. La

dynamique de sensibilité de la caméra événementielle est nettement mise en évidence : une personne non visible sur la caméra standard le devient sur la caméra évènementielle.

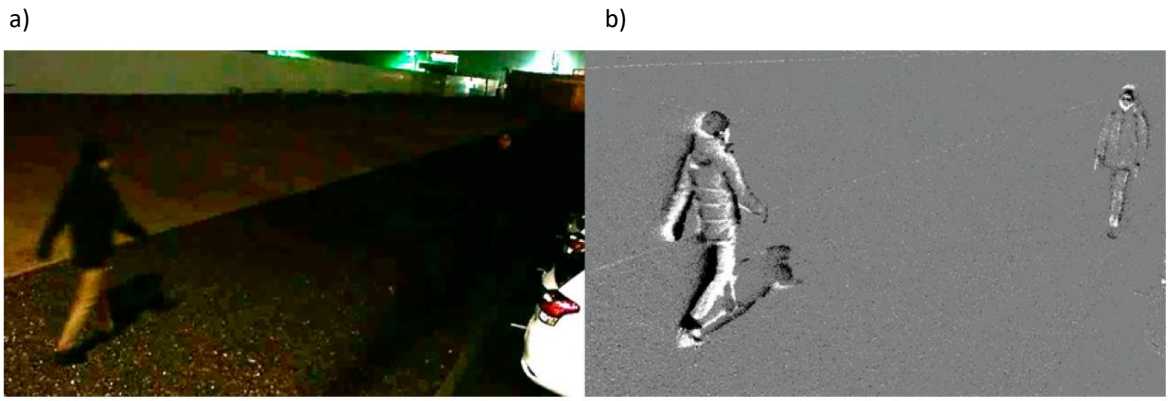


Figure 1.11 : a) Caméra conventionnelle b) Caméra évènementielle de Prophesee.

Les caméras évènementielles ont un fonctionnement intrinsèque asynchrone. Pour le traitement de l'image, il est plus judicieux d'utiliser un traitement asynchrone comme les réseaux de neurones à spike plutôt qu'un traditionnel ordinateur afin d'obtenir de meilleurs résultats en termes de rapidité et de consommation énergétique [GALLEGO G. 2019], [STROMATIAS E. 2017]. Les réseaux de neurones à spike fonctionnant comme les caméras évènementielles sont donc naturellement adaptés au traitement des données en sortie de caméra.

1.6 Réseau de neurone à spike

Codage par fréquence d'impulsions

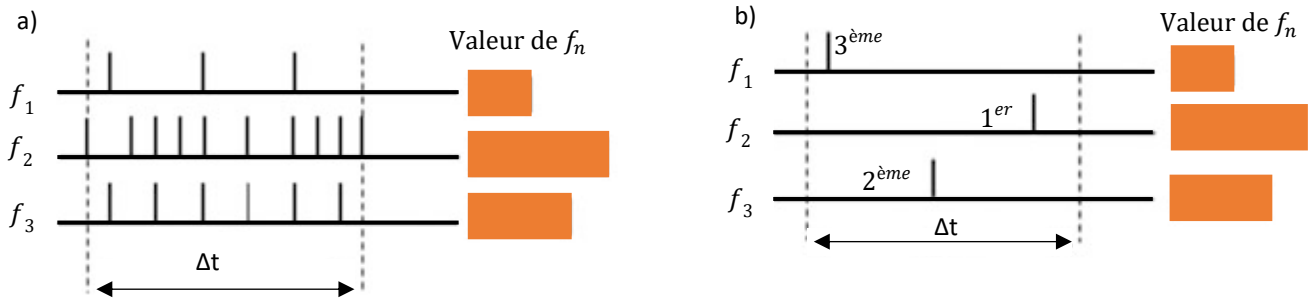


Figure 1.12 : Conversion d'une valeur en évènement : a) fréquentiel, b) : par ordre d'arrivée

Pour utiliser les réseaux de neurones évènementiels dans le monde réel il faut pouvoir transformer les informations visuelles, sonores ou autres en évènement. Il y a plusieurs façons de coder la valeur reçue par le capteur. Si par exemple on souhaite transformer une valeur f_n correspond à grandeur physique en évènement, il y a plusieurs moyens d'y parvenir. On peut transformer cette grandeur physique en fréquence (a), la fréquence est proportionnelle à la valeur de la grandeur physique mesuré (voir figure 1.12 a). On peut également transformer l'information reçue par des évènements ordonnés (figure 1.12 b), le premier à arriver aux neurones correspond à la valeur la plus importante, le temps de retard est

inversement proportionnel à cette valeur [KASABOV, N. 2019]. On utilisera pour notre réseau de neurones le codage par fréquence d'évènements qui est sensible à la variation de luminosité.

Modèle du neurone

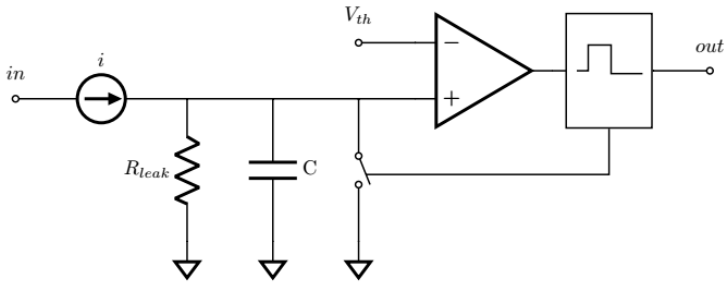


Figure 1.13 : Schéma électrique d'un neurone LIF

Le modèle du neurone LIF présenté (figure 1.13) [SCHWARTZ E. L. 1997] sous forme électrique, permet de comprendre le fonctionnement d'un neurone, mais permet en particulier d'avoir un schéma électrique permettant de faire un réseau de neurones matériel. La source de courant représente la somme des évènements externes multipliés par les poids synaptiques respectifs des synapses afférentes. Quand suffisamment de courant a chargé la capacité C et que la tension de cette dernière est supérieure à V_{th}, le neurone déclenche un spike. La résistance R_{leak} permet de décharger lentement la capacité de membrane C pour qu'un évènement qui a chargé cette capacité longtemps auparavant ne soit plus pris en compte et ne perturbe pas l'évènement à venir.

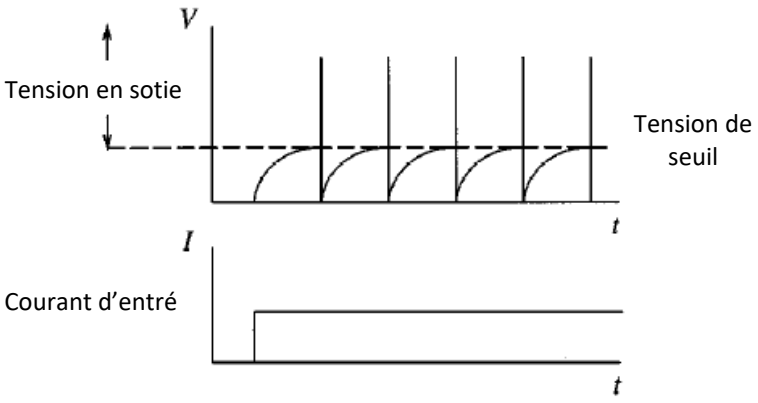


Figure 1.14 : Simulation de la charge puis du déclenchement d'un neurone LIF [SCHWARTZ E. L. 1997]

La (figure 1.14) présente une simulation électrique du comportement d'un neurone LIF sous un courant d'entrée continu. La valeur minimale du courant à injecter pour que le neurone déclenche est $i = \frac{V_{th}}{R_{leak}}$; en dessous de cette valeur le neurone ne pourra jamais déclencher [SCHWARTZ E. L. 1997]. Dans le cas général, le courant n'est pas constant, mais dépend des évènements ainsi que du poids de leurs synapses.

Apprentissage

Spike Timing Dependent Plasticity

En étudiant les mécanismes d'apprentissage du cerveau, Hebb en 1949 décrit une loi d'apprentissage non supervisé. Cette loi est basée sur la corrélation temporelle entre les évènements qui arrivent depuis les neurones pré-synaptiques et l'évènement qui est généré par le neurone post-synaptique (voir figure 1.15).

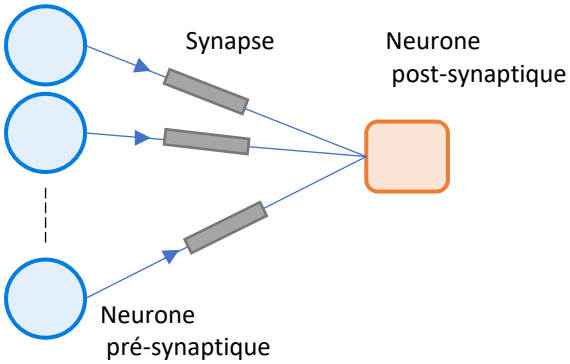


Figure 1.15 : Réseau de neurones vers un seul neurone post-synaptique.

Le contrôle de la modification du poids des synapses est l'essence des règles d'apprentissages. La STDP (Spike Timing Dependent Plasticity) est une théorie de modification de poids de la synapse en fonction des potentiels d'actions pré et post-synaptique. Le modèle proposé par Badoual M. correspond à la variation de poids d'une synapse biologique sur une longue durée. Dans ce modèle la variation de poids synaptique évolue selon deux fonctions exponentielles $Q(t_j\text{-}t_i)$ et $P(t_i\text{-}t_j)$ en fonction des temps de déclenchement t_j du neurone pré-synaptique et t_i du neurone post-synaptique [BADOUAL M. 2006]. Cette modification du poids est due au potentiel d'action pré-synaptique arrivant avant ou après le potentiel d'action post-synaptique [PURVE D. 2014]. Sur la (figure 1.16) on peut voir la variation de poids synaptique en fonction de la différence entre t_j et t_i, dans le premier cas (figure 1.16 a) on a une augmentation du poids, dans le deuxième (figure 1.16 b)on a une diminution, enfin dans le troisième (figure 1.16 c) si $(t_i\text{-}t_j)$ ou $(t_j\text{-}t_i)$ est trop grand il ne se passe rien, car les évènements pré et post-synaptique sont trop éloignés. On peut voir le résultat du changement de poids de la synapse relative en fonction du temps de déclenchement des neurones pré et post synaptique en (figure 1.16d).

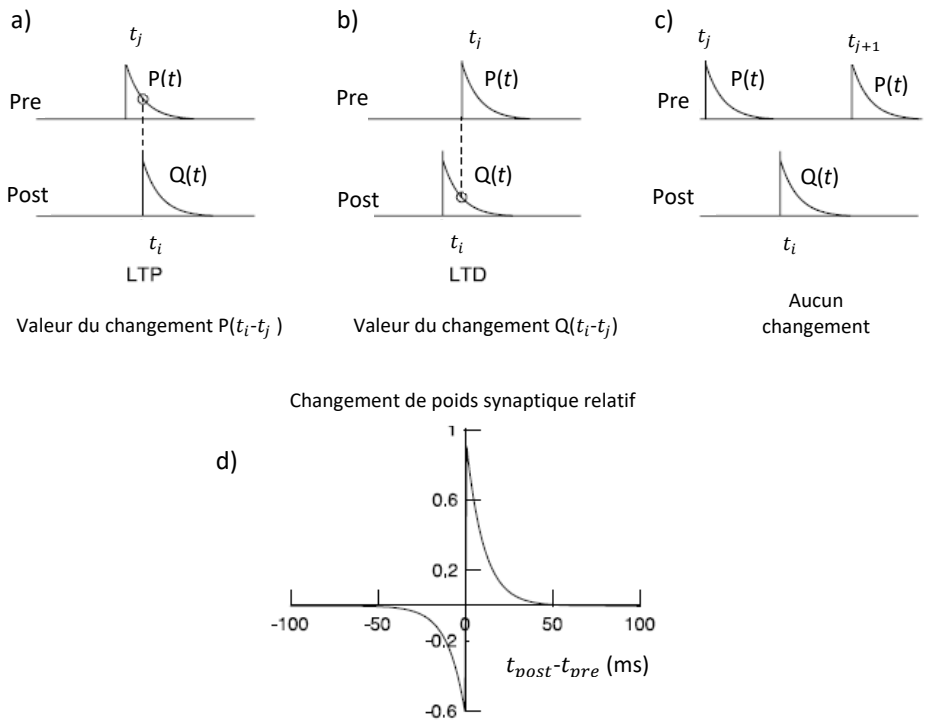


Figure 1.16 : Evolutions du poids synaptique en fonction du temps de déclenchement. a) Augmentation du poids. b) Diminution du poids. c) Aucun changement de poids. d) Changement de poids synaptique relatif en fonction des temps de déclenchement des neurones pré et post-synaptique.

1.7 Crossbar de memristor

Dans les réseaux de neurones évènementiels matériels, on utilise des synapses artificielles qui peuvent être des memristors comme dans le cas du projet ULPEC. Le principe physique du memristor lui confère la capacité de changer son poids synaptique de la même manière que les synapses biologiques. La conductance (l'inverse de la résistance) des memristors est le poids synaptique [SERRANO-GOTARREDONA T. 2013]. Les memristors ont une résistance électrique que l'on peut faire varier en appliquant une tension à leurs bornes. Les synapses pour les réseaux de neurones impulsionnels matériels faites à base de memristors ont le grand avantage de voir leur poids se modifier en fonction de l'activité du réseau. Pour cela on code les évènements générés par les neurones pré et post-synaptique par des impulsions de tensions spécifiques. La valeur des tensions aux bornes des memristors générées par les impulsions à des temps bien précis permet de modifier leurs poids de la même manière que la STDP [BILL J. 2014].

Un crossbar de memristors est une matrice de memristors qui permet par construction de connecter chacune des neurones d'entrée à tous les neurones de sortie (voir figure1.17). Un crossbar est simple de

fabrication et pratique pour faire un réseau de neurones dit « all to all » où toutes les entrées sont connectées à toutes les sorties.

Même si les simulations n'offrent pas de bonnes performances en termes d'énergie, elles sont utiles pour comprendre le fonctionnement de ces réseaux de neurones avec memristors. De nombreux articles ont évalué la performance des simulations de réseaux de neurones matériels avec un crossbar de memristors commandé par des neurones LIF [SHAMSI J. 2018], [ZHENG N. 2018] [BILL J. 2014]. Les neurones LIF sont particulièrement bien adaptés au crossbar de memristors. En maintenant constant les tensions de sortie du crossbar, le courant qui circule est uniquement dépendant de la tension en entrée du réseau et de la résistance des memristors qui est l'inverse du poids synaptique.

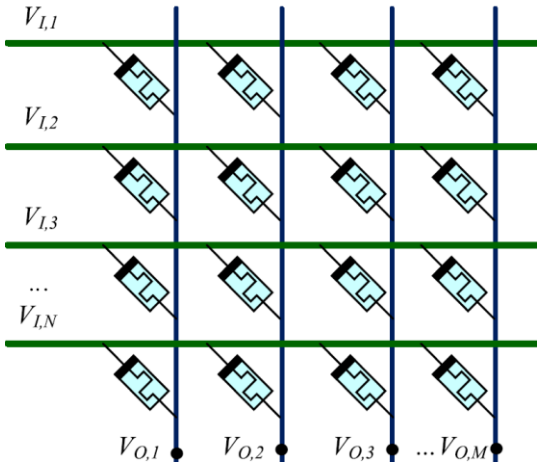


Figure 1.17 : Crossbar de memristors.

Quelques crossbars de memristors ont été réalisés. Ils sont de deux types différents, les passifs et les actifs. Les crossbars passifs sont de simples quadrillages de memristors comme présenté à la (figure 1.17). Les crossbars actifs possèdent en plus un transistor relié à chaque memristor. Les applications de ces réseaux de memristors ne sont pas uniquement réservées aux réseaux de neurones. Grâce à leurs configurations, ils peuvent faire du calcul analogique en parallèle très rapidement, pour des applications dans le filtrage d'image par exemple. L'avantage des réseaux de neurones actifs est qu'ils peuvent modifier ou contrôler chaque memristor indépendamment. La taille de ces réseaux est variable pouvant aller de 128*64 memristors pour les crossbars actifs à 32*32 pour les passifs [XIA Q. 2019]. Bien que les réseaux de memristors pilotables par transistor soient plus pratiques, ils sont néanmoins plus compliqués à produire.

Dans le cadre du projet ULPEC, nous visons la réalisation et l'utilisation d'un crossbar passif composé de 784 x 100 memristors. Il sera piloté par des neurones LIF et la modification des poids se fera par STDP.

1.8 ULPEC

Le projet européen ULPEC (ICT 03-2016 Smart System Integration) auquel contribues cette thèse a pour ambition de réaliser un réseau de neurones bio-inspiré à base de memristors et d'une rétine artificielle dans le but de faire de la reconnaissance d'image et d'ouvrir une voie vers le calcul neuromorphique avec une architecture alternative à celle de von Neumann. ULPEC pour « Ultra Low Power Event-based Camera » est un projet qui compte neuf partenaires, répartie en quatre universités et quatre partenaires industriels ainsi que le CNRS, et est financé par l'Union européenne à hauteur de 4,9 M€ pour une durée de quatre ans. Les partenaires sont Université de Bordeaux (France), Chronocam SAS devenu Prophesee (France), IBM Research GMBH (Switzerland), Robert Bosch GMBH (Germany), Universiteit Twente (Netherlands), Université Pierre et Marie Curie devenue Université de Sorbonne Paris (France), CNRS (France), Twente Solid State Technology B.V. (Netherlands) et Eidgenoessische Technische Hochschule Zuerich (Switzerland)

1.9 Conclusion

Avec l'augmentation rapide et envahissante de l'IA dans de nombreux domaines de notre vie et encore plus à l'avenir [SHABBIR J. 2018], la puissance de calcul nécessaire devra augmenter. L'amélioration continue depuis 1965 des performances des processeurs résultants de la loi de More n'est plus envisageable, une augmentation de la performance se fait désormais au détriment de la consommation énergétique malgré la réduction de la taille des transistors [BOHR M. T. 2017]. Les processeurs avec leur architecture actuelle ne sont pas efficaces pour traiter les algorithmes de machine Learning. Dans un but d'économie d'énergie pour des applications embarquées entre autres, mais également à l'échelle mondiale, il est nécessaire de trouver une solution adaptée si l'on souhaite le développement de l'intelligence artificielle pour faciliter nos tâches quotidiennes.

Dans ce premier chapitre, nous avons montré l'engouement de la recherche sur les réseaux des réseaux de neurones matériels pour s'affranchir de la classique architecture de von Neumann, limitée en vitesse de calcul et gourmande en énergie. De nombreuses recherches fructueuses ont montré la possibilité de faire de l'apprentissage et de la reconnaissance avec un réseau de neurones hardware inspiré du fonctionnement du cerveau, qui est en plus d'être efficace est très peu consommateur d'énergie. D'autre part, les memristors sont de bons candidats pour jouer le rôle de synapses dans les réseaux de neurones bio-inspirés. Les caméras évènementielles consomment moins d'énergie et offrent une dynamique plus grande que les caméras classiques. De plus, les évènements qu'elles génèrent sont directement utilisables par les réseaux de neurones évènementiels. C'est dans ce contexte que se place le projet ULPEC et ces travaux de thèse. Dans la suite de ce manuscrit, il sera expliqué plus en détail le fonctionnement des memristors, la conception des circuits électroniques permettant leur pilotage et la présentation du système ULPEC.

CHAPITRE 2 : LES SYNAPSES MEMRISTIVES

2.1 Introduction

Les réseaux de neurones artificiels matériels ont pour objectif de réduire la consommation énergétique tout en augmentant la performance de calcul en faisant de la reconnaissance d'image ou du traitement du langage verbal en temps réel. Plusieurs solutions sont proposées pour la substitution des synapses biologiques, l'une d'entre elles est l'utilisation de synapse à base de memristors qui semble être prometteuse. C'est cette technologie qui a été retenue pour le projet ULPEC. Ce chapitre présentera le fonctionnement d'un memristor ferroélectrique et son modèle Verilog A qui permet l'étude des capacités d'apprentissage du futur système ainsi que la conception de son électronique.

2.2 Memristor : origine et application

En 1971 Leon O. Chua [CHUA L. 1971] décrit d'un point de vue théorique le fonctionnement d'un nouveau composant appelé memristor, en s'inspirant des 3 composants classiques, la résistance, le condensateur et l'inductance. Il définit la memristance notée M de la fonction suivante :

$$M(q) = \frac{d\varphi}{dq}$$

où φ est le flux et q la charge qui traverse le memristor. La memristance est homogène à une résistance. Dans son modèle Chua définit la memristance comme une variable dépendante des charges injectées dans le memristor (q(t)). Cependant les différents memristors actuels et en particulier celui que nous utilisons ont un fonctionnement différent. La variation de la résistance est causée par une application d'une tension spécifique. Le memristor agit comme une mémoire analogique où l'on garde en information la valeur de sa résistance quand cette tension n'est plus appliquée. Ils sont largement répandus dans les réseaux de neurones matériels, mais présents également dans quelques autres applications décrites ci-dessous.

Applications possibles des memristors

Les memristors en utilisant leurs caractéristiques de mémoire non volatile ont un avenir dans la lutte contre le gaspillage énergétique. Dans les mémoires SRAM, pour garder en mémoire les données il faut que les transistors soient sous tension. En utilisant des memristors plutôt que la traditionnelle architecture à base de transistors dans les mémoires RAM [HALAWANI Y. 2015], on peut dans certaines applications où le cycle de lecture et écriture est lent, atteindre des économies d'énergie de l'ordre de 80%.

D'autres applications un peu plus originales ont été étudiées. Certains les utilisent comme portes logiques dans le but de réduire la consommation et augmentant l'intégrabilité [TETZLA R. 2014]. Une dernière application envisageable, est d'utiliser la mémoire analogique non volatile des memristors, pour qu'avec un seul composant on puisse avoir une mémoire sur « 6.5 bits », soit 45 états en tout [STATHOPOULOS S. 2017].

Synapse et application sur les réseaux de neurones

Dans le but de se soustraire à la classique architecture de von Neumann ainsi que de faire du traitement d'image, nous avons choisi de nous inspirer de la biologie pour faire un réseau de neurones à base de memristors. Nous utilisons un modèle LIF du neurone qui est largement utilisé et reconnu efficace. Pour une intégration matérielle, les synapses physiques doivent respecter des caractéristiques précises pour le bon fonctionnement des réseaux de neurones. Une bonne synapse pour l'intégration physique devrait respecter ces quatre caractéristiques [WANG J. 2019] :

- La synapse doit pouvoir garder en mémoire sa valeur et doit pouvoir tout aussi facilement augmenter son poids que de le diminuer ;
- La sortie vers le neurone de la couche supérieure de la synapse doit être le produit entre son poids synaptique et son entrée ;
- La valeur du poids après application des signaux est dépendant du poids initial de la synapse et des signaux qui lui sont appliqués ;
- Les synapses doivent être les plus compactes possible, dans le but d'une grande intégrabilité et doit transférer l'information vers un seul neurone tout en étant peu gourmandes en énergie.

Les memristors ferroélectriques utilisés dans le cadre du projet ULPEC ont ces quatre caractéristiques, ce qui leur permet d'être aisément utilisés dans un réseau de neurones matériel.

Le memristor n'est pas le seul composant respectant ces caractéristiques et peut être utilisé dans des réseaux de neurone matériel. Il y a entre autres les transistors à grille flottante qui peuvent être utilisés dans l'application de réseaux de neurones [GOPALAKRISHNAN R. 2015]. Dans ce cas, les transistors se comportent comme des synapses, où la conductance est modifiée en changeant la tension de grille du transistor. Dans la publication [MOHMOODI D. 2018], la synapse est substituée par une mémoire flash dont le fonctionnement est similaire à celui d'un transistor à grille flottante, où l'on injecte un courant dans la grille du transistor pour modifier le poids de la synapse. Il existe également des synapses à base de micro miroirs commandables pour une application sur les réseaux de neurones optiques [BUENO J. 2018].

D'un autre côté, il existe plusieurs technologies de memristors. Celle que l'on utilise fait partie des memristors ferroélectriques. Ce choix a été fait pour plusieurs raisons, la résistance R_{on} est élevée de l'ordre du méga Ohm et le rapport R_{off}/R_{on} est supérieur à 100. Les tensions à appliquer pour modifier la

conductance sont compatibles avec des valeurs de la microélectronique, de l'ordre de 2 V, de plus il est également possible d'intégrer ce type de memristor dans un crossbar, ce qui est une condition indispensable à la conception d'un réseau de neurones physique fonctionnel.

2.3 Fonctionnement memristor ferroélectrique

Ce paragraphe détaille brièvement la physique du memristor permettant au lecteur d'appréhender les travaux de recherche présentés ultérieurement

Effet ferroélectrique

Un matériau ferroélectrique est un matériau qui a une polarisation électrique spontanée, ou rémanente. C'est-à-dire qu'au repos, sans application d'un champ électrique la polarisation du matériau est non nulle (figure 2.4). Cette polarisation est modifiée avec l'application d'un champ électrique suffisamment important. Les matériaux ferroélectriques ont la particularité de garder « en mémoire » l'état de polarisation une fois que le champ électrique est retombé à zéro.

Un matériau cristallin n'est pas purement homogène, il est composé d'une multitude de granularités qui sont appelées domaines, chaque domaine du matériau ferroélectrique a une polarisation ainsi qu'une orientation du cristal bien distinct (figure 2.1) La polarisation totale est la somme vectorielle de chacune des polarisations des domaines que le matériau contient [UCHINO K. 2010].

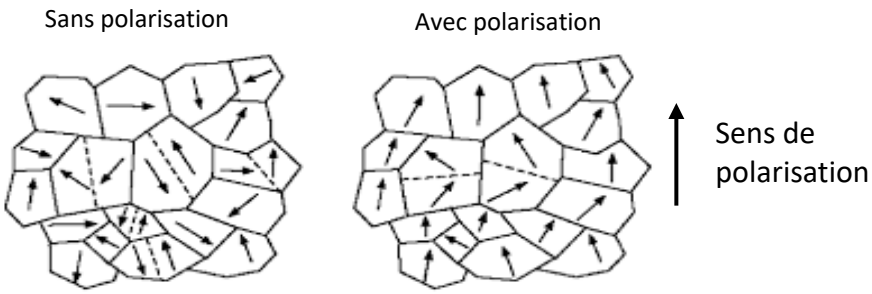


Figure 2.1 : Domaines ferroélectriques avec et sans polarisation

La polarisation électrique est un phénomène qui se passe au niveau atomique dans une structure cristalline. Lorsqu'un atome chargé (anion ou cation) subit un déplacement par rapport aux autres ions de la maille, il en résulte l'apparition d'un dipôle électrique. Le sens de la polarisation dépend du sens de déplacement des charges électriques. Plus la distance entre les charges positives et les charges négatives est grande, plus la polarisation est importante. L'application d'un champ électrique suffisamment important peut entrainer un déplacement des ions et donc changer la polarisation du matériau. Quand le champ électrique disparait, les ions qui se sont déplacés restent à leur place et créent une polarisation rémanente.

Structure du memristor

Le memristor est composé d'une épaisseur de quelques nanomètres d'un matériau ferroélectrique pris en sandwich entre deux métaux différents. Le matériau ferroélectrique qui est utilisé pour la fabrication du memristor est du baryum de titane (BTO), qui a une structure cristalline de Pérovskite. Sa structure est de la forme ABO_3 avec le béryllium aux site A, le titane en B et les oxygènes sur les sites O (figure 2.2).

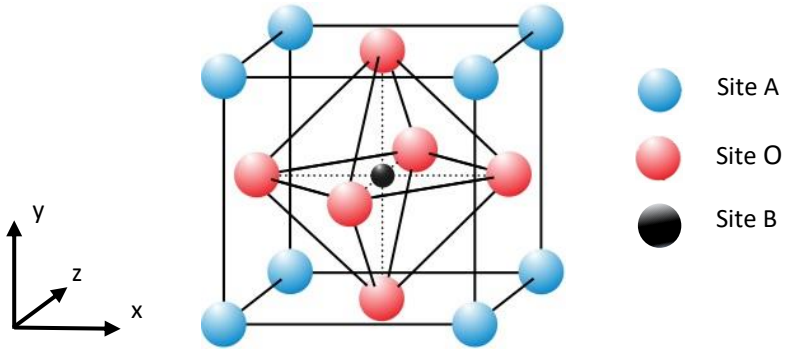


Figure 2.2 : Structure Pérovskite du titane de baryum

A température ambiante la position d'équilibre du titane est de plus ou moins 0,06 Å par rapport aux sites du béryllium selon l'axe x, y ou z [COHEN R. 1992]. Le point d'équilibre est l'endroit qui minimise l'énergie globale du système, en dessous de la température de Curie qui est de 123°C [SAKAYORI K. 1995], il y a six emplacements qui minimisent cette énergie et donc création d'une polarisation orientée selon l'un des six sens possibles du déplacement du titane. Au-delà de cette température, le point d'équilibre du titane est au centre de la maille, il en résulte une absence de polarisation [MARTIN L. 2016]. Il ne faut donc pas que le memristor dépasse les 123°C au cours de son utilisation au risque de perdre sa polarisation et sa mémoire.

Théorie de Landau-Devonshire

Cette théorie a été établie pour traiter les transitions de phase dans un matériau ferroélectrique et l'appliquer à notre memristor. Cette théorie complexe qui est le résultat de la mécanique quantique sera abordée de manière superficielle, mais permettra de comprendre les grands principes du changement de phase quand une tension est appliquée sur le memristor.

La théorie de Landau exprime l'énergie libre du système (G) qui dépend de la polarisation orientée du matériau (p) de la température (T) ainsi que d'un champ électrique externe (E), la fonction suivante a été tronquée au sixième ordre de la série polynomiale et donne [CHANDRA P. 2007] [ZHI M 2017] :

$$(2.1)\ G = -Ep + \frac{a}{2}p^2 + \frac{b}{4}p^4 + \frac{c}{6}p^6$$

avec $a = a_0 * (T - T_0)$, T_0 la température de Currie, b et c des paramètres non constants. La position stable est la valeur de la polarisation qui crée un minimum local de l'énergie : $\frac{dG(p)}{dp} = 0$. Pour E=0 et si $T<T_0$ alors il y a deux valeurs de polarisations, égales en valeur absolue qui minimise l'énergie du cristal, si $T>T_0$ la seule valeur qui minimise l'énergie du cristal est pour p=0 (figure 2.3a). A température ambiante, si l'on applique un champ électrique, on change l'énergie interne du matériau (figure 2.3 b, c et d) et si ce champ est suffisamment important (figure 2.3 d) alors il n'y a qu'une seule valeur de p qui satisfasse l'équation : $\frac{dG(p)}{dp} = 0$ [BOYN S. 2016] et le titane peut changer de position et donc de polarisation.

Il faut apporter un champ électrique suffisamment important pour qu'il y ait un changement de polarisation et donc de résistance du memristor. Après changement de polarisation et une fois le champ électrique disparu, la polarisation reste à la dernière valeur obtenue. Sur la (figure 2.4), on peut voir l'évolution de la polarisation en fonction du champ électrique appliqué ainsi que le déplacement de l'emplacement du titane dans le cristal.

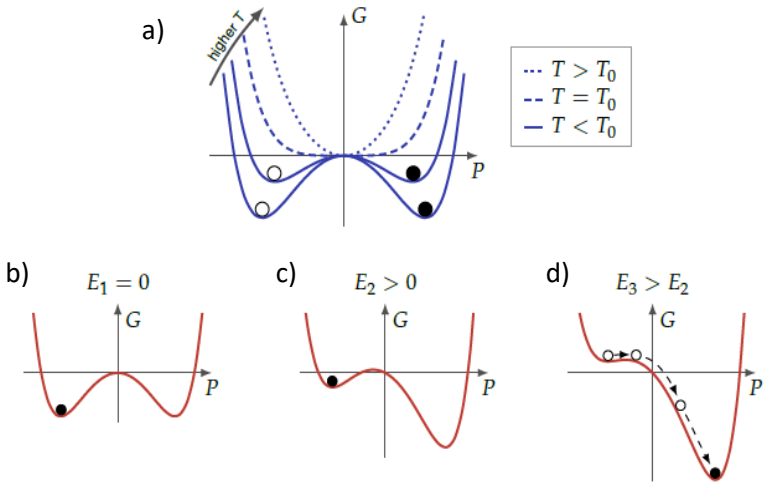


Figure 2.3 : Énergies libres du cristal en fonction de la polarisation et de la température en a) et en fonction de la polarisation et d'un champ électrique externe en b), c) et d).

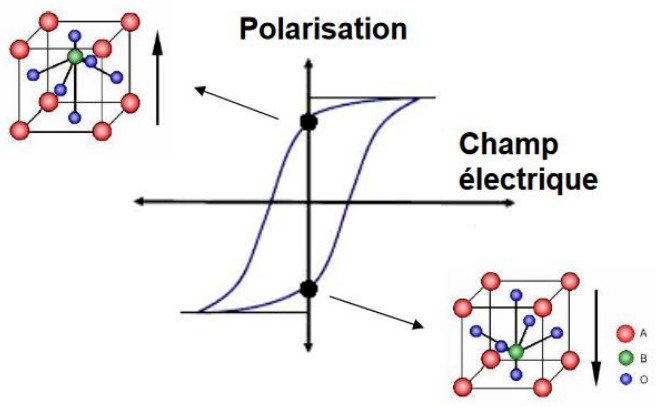
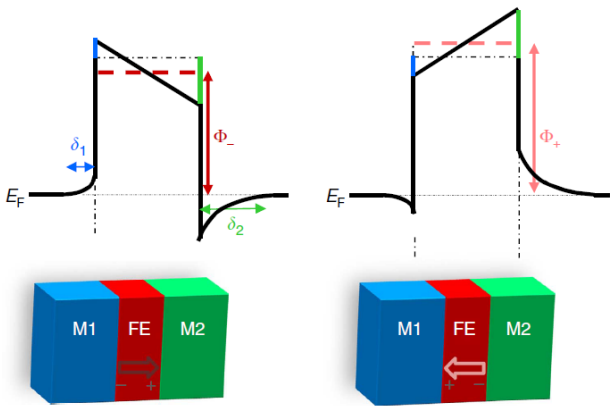


Figure 2.4 : Polarisations du matériau en fonction de la position du titane dans le cristal et du camp électrique qui a permis la modification de cette polarisation.

Origine des courants du memristor

Le principe de jonction tunnel ferroélectrique a été montré par Esaki en 1971. L'origine de la variation de résistance est due à une variation de polarisation dans la fine couche du matériau ferroélectrique qui génère une variation de la barrière ferroélectrique. À température ambiante les courants originaires de l'effet tunnel sont dominants par rapport à ceux liés aux effets thermiques, ces types de courant sont dépendants du potentiel de barrière dans une jonction tunnel ferroélectrique et varient exponentiellement avec cette dernière [GARCIA V. 2014].



Polarisation droite : petite barrière

Polarisation gauche : grande barrière

Figure 2.5 : Évolution de la barrière de potentiel au sein du memristor en fonction des deux différentes polarisations.

Quand il y a polarisation vers la gauche ou la droite, cette polarisation est de même intensité, de sens opposé et génère une barrière de potentiel de même hauteur. Le matériau ferroélectrique ne permet pas à lui seul de créer une variation de résistance, car la hauteur de barrière est identique dans ces deux cas.

En ajoutant deux matériaux différent M1 et M2 de chaque côté du matériau ferroélectrique FE, on crée à leurs interfaces une asymétrie de la marche de potentiel qui permet d'avoir une barrière de taille différente entre la polarisation gauche et droite (figure 2.5). Cela implique qu'il y a une différence de courant pour une même tension appliquée et donc une variation de résistance du memristor.

Temps de changement de polarisation / résistance du memristor

Le changement de polarisation se fait par petites parties indépendantes que l'on appelle nucléations [DAWBER M. 2005]. En fonction de la tension, de la durée et du nombre de créneaux, les domaines se polarisent plus ou moins rapidement et modifient proportionnellement la conductance électrique du memristor.

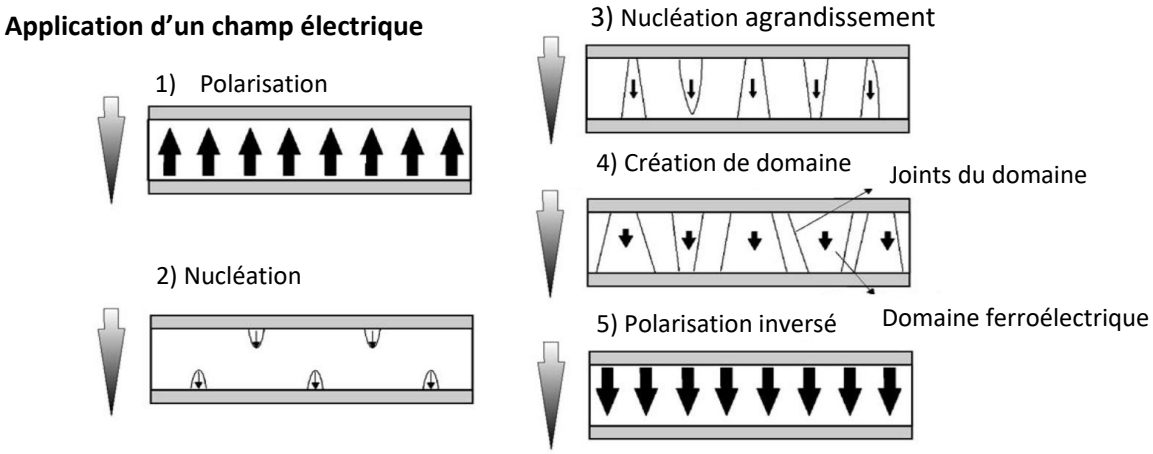


Figure 2.6 : Évolutions de la polarisation étape par étape quand un champ électrique est appliqué.

Lorsqu'un champ électrique est appliqué sur le matériau déjà polarisé de sens opposé, il y a transition de polarisation, mais elle prend un certain temps, illustration (figure 2.6) [DAWBER M. 2005]. Le changement de polarisation commence par l'apparition de nucléation au niveau des deux interfaces du matériau. Ces nucléations sous l'application du champ électrique s'étirent jusqu'à l'interface opposée, et forment un domaine indépendant qui croît sur la largeur du matériau avant l'inversion totale de la polarité. Ces étapes ne se font pas instantanément, elles prennent un certain temps qui varie en fonction du champ électrique (de la tension appliquée aux bornes du memristor).

Dans notre application, il est avantageux que le changement de résistance ne se fasse pas instantanément de R_{off} à $R_{on,}$ ou inversement, on évite ainsi d'avoir des poids synaptiques binaires. Si le changement de résistance est trop brutal, le réseau de neurones risque de se bloquer dans certaines valeurs et de ne plus rien apprendre. Le temps de renversement permet de changer la résistance du memristor progressivement et d'obtenir ainsi une palette importante de résistances possibles.

Inversion de polarisation des domaines ferroélectriques quand des créneaux de tension sont appliqués au memristor

La (figure 2.7) montre l'évolution de la résistance du memristor en fonction du nombre de créneaux de tension appliqués. Les mesures du memristor ont été faites après chaque application d'un créneau de tension. Les créneaux ont tous la même durée de 20ns et la même tension positive et négative sur chacune des figures a) et b). Les points rouges sur la (figure 2.7), correspondent à des créneaux positifs de 2,9V en (a) et 3V en (b), les points bleus à des créneaux négatifs, de -2,7V en (a) et -3V en (b), les points en noir correspondent à la réinitialisation de la résistance du memristor à la valeur minimale R_{on}. Cette figure montre la variation progressive et la saturation de la résistance du memristor, elles sont d'autant plus rapides que la tension en valeur absolue appliquée sur le memristor est importante.

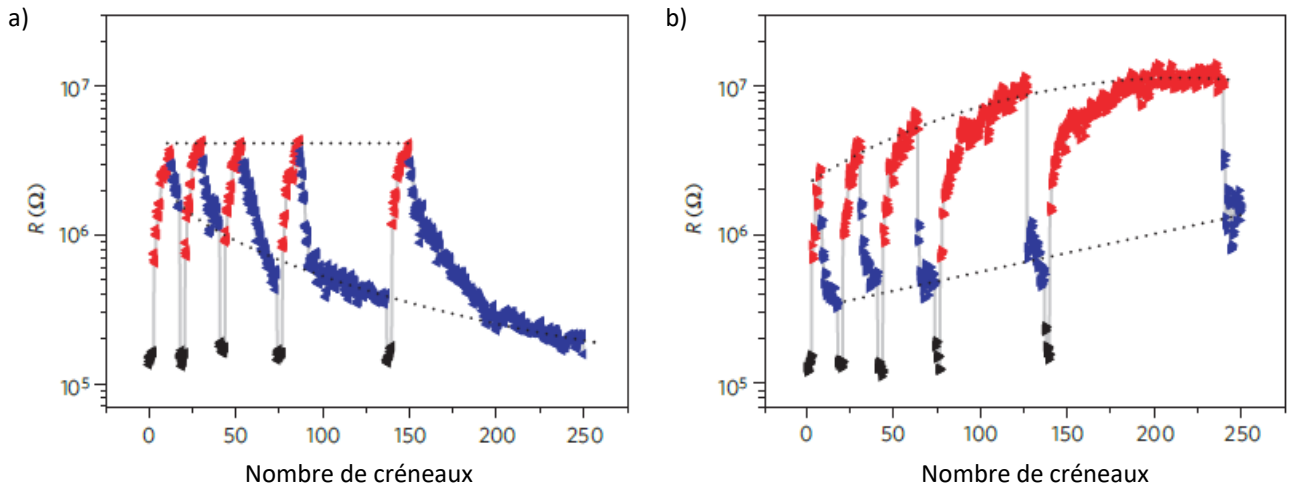


Figure 2.7 : Variations de la résistance en fonction des pulses de tension appliqués, d'une durée de 20ns [CHAMBOULA A. 2012].

L'évolution de la résistance est fortement dépendante de la tension appliquée aux bornes du memristor. En appliquant toujours des créneaux de 20 ns, mais de tension variable, en partant et arrivant à 0V, en passant par 4V et -5V, on obtient la courbe de l'évolution de la résistance (figure 2.8). On peut déjà analyser sur la partie (a) qu'il faut dépasser un seuil de tension pour commencer à faire varier la résistance du memristor. La variation de résistance du memristor dépend de la tension appliquée, mais également de sa résistance, plus elle est élevée, plus la tension appliquée doit être importante. La partie (b) présente l'évolution de polarisation des domaines ferroélectriques dans le memristor en fonction de sa résistance et des pulses qui lui sont appliqués. Dans les encadrés rouges, on représente la phase de polarisation du memristor où on applique des créneaux positifs afin d'augmenter sa résistance, en bleu la phase de polarisation pendant la réduction de la résistance, où l'on applique des créneaux négatifs.

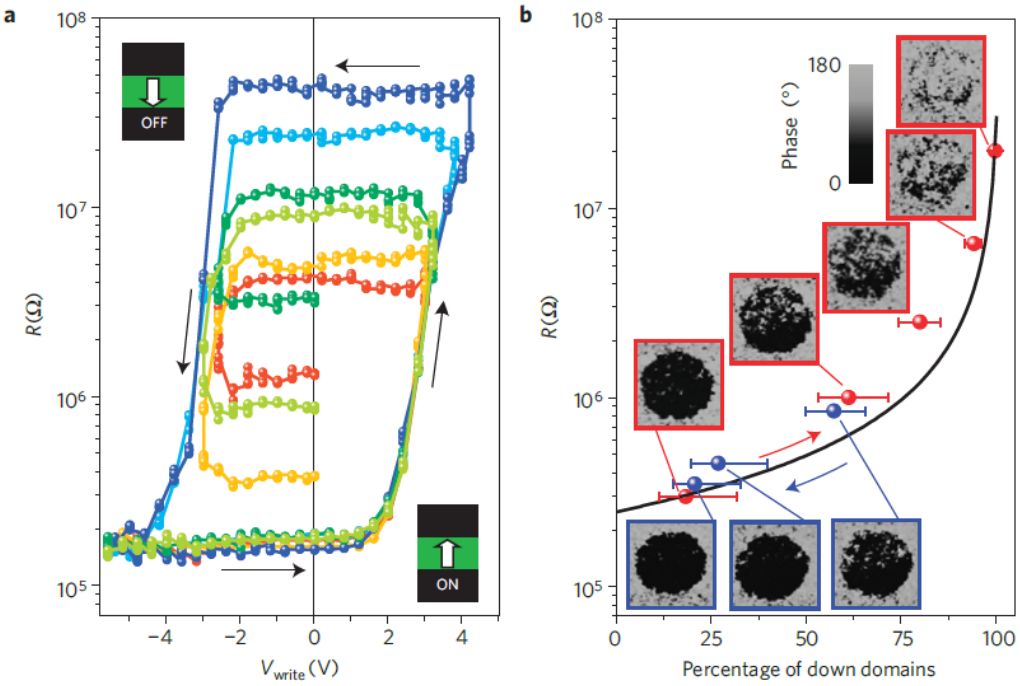


Figure 2.8 : La mesure de la résistance se fait sous une tension de 100mV. a) Hystérésis correspondant à la résistance du memristor avec l'application du créneau de tension variable. b) Fraction des domaines « fermés » en fonction de la résistance du memristor. [CHAMBOULA A. 2012]

Les domaines en blancs sont « fermés », la phase de polarisation est de 180°, la résistance de ces domaines y est très élevée. Les domaines en noir sont « ouverts », la polarisation a une phase de 0° et la résistance y est plus faible. On définit S comme le ratio de surface des domaines fermés par rapport à toute la surface de la section. Il y a proportionnalité entre la conductance du memristor (1/R) et S qui est définie par la formule suivante [CHAMBOULA A. 2012] :

$$(2.2)\ \frac{1}{R} = S\left(\frac{1}{R_{off}} - \frac{1}{R_{on}}\right) + \frac{1}{R_{on}}$$

La résistance minimale R_{on} du memristor est atteinte quand tous les domaines sont ouverts (S=0) et la résistance maximale R_{off} l'est quand tous les domaines sont fermés (S=1).

2.4 Modèle Verilog A

Pour nous permettre la conception de nos neurones analogiques dans le cadre du projet ULPEC, nous avons eu besoin d'intégrer un modèle du memristor dans notre logiciel de conception. Nous avons fait le choix d'un modèle comportemental écrit en Verilog A. Le modèle du memristor ferroélectrique présenté dans le cadre de ce manuscrit est basé sur les mesures réalisées sur ce composant [MEYER C. 2018]. Nous

avons pris pour point de départ un modèle déjà existant, mais qui ne prenait pas en compte les variabilités des valeurs minimales et maximales ainsi que la variabilité du changement de valeur de la résistance du memristor lors de l'application des créneaux de tension [BOYN S. 2017].

Mesure du memristor en vue d'un modèle.

Une série de mesures sur le memristor a été faite pour établir sa capacité d'apprentissage suivant la règle de STDP. Pour cela on applique les tensions $V_{pré}$ et V_{post} sur le memristor (figure 2.9 b), puis on mesure la variation de conductance du memristor après l'application de ces signaux en fonction de $\Delta t = t(V_{post}) - t(V_{pré})$.

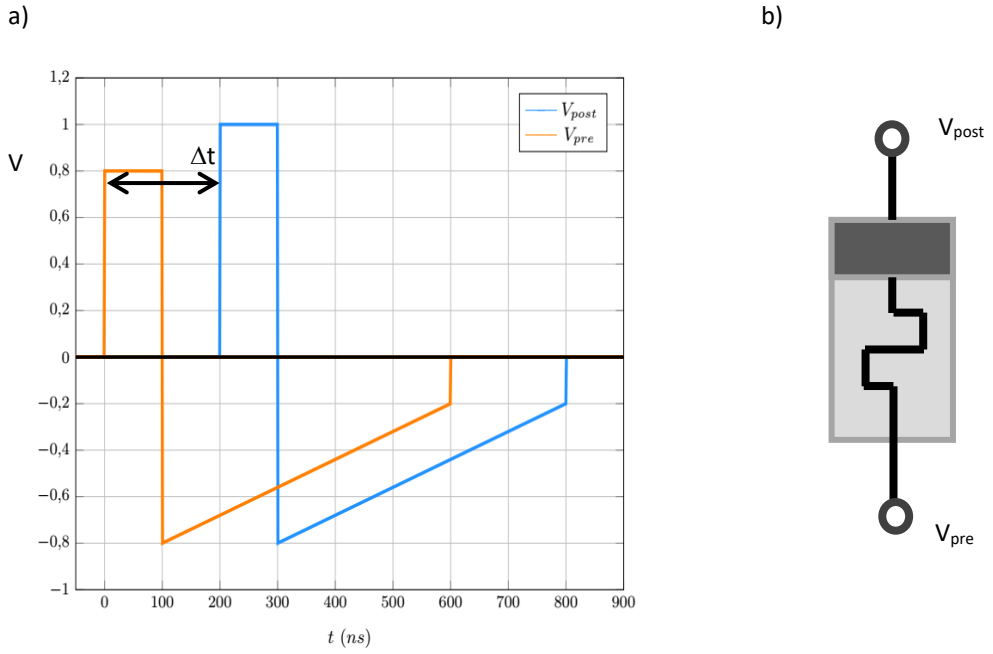


Figure 2.9 : Forme des potentiels d'action pré- et post-synaptiques appliqués sur le memristor.

Les signaux $V_{pré}$ et V_{post} correspondent respectivement au pulse généré par les neurones pré-synaptique et post-synaptique du réseau de neurones créé pour l'utilisation de ce type de memristor [LECERF G. 2015.]. Il existe plusieurs types de memristors ferroélectriques avec des caractéristiques différentes, celui que nous avons mesuré en vue de faire notre modèle Verilog A a une couche ferroélectrique de type $BaFeO_3$, ce qui confère à ce type de memristor un seuil de transition de l'ordre de 1V. Pour permettre de charger la capacité de membrane du neurone post-synaptique sans modifier la résistance du memristor, les signaux doivent avoir une tension inférieure à 1V. Durant la modification du poids synaptique, la tension présentée aux bornes du memristor doit excéder 1V. Cela sera réalisé grâce à une différence de

potentiel (V_{pre} - V_{post}) de 1,6V ou -1,7V pour les valeurs maximales et minimales respectivement. Les signaux $V_{pré}$ et V_{post} sont présentés (figure2.9).

Pour mesurer la variation de conductance du memristor (donc le poids de la synapse) on l'initialise dans une gamme de résistance prédéfinie, puis on applique les créneaux pré- et post-synaptiques sur le memristor décalé de Δt (figure 2.9) et enfin on mesure la variation de conductance après l'application de ces créneaux. La variation du poids synaptique est dépendante de la résistance initiale du memristor. Plusieurs valeurs initiales de résistance ont été utilisées pour obtenir le résultat de mesures de la (figure 2.10). La mesure de la résistance du memristor se fait en mesurant le courant sous une tension de 100mV à ses bornes.

La modélisation du memristor effectuée par [BOYN S. 2017] est toujours valable pour nos mesures. Sur la (figure 2.10 b), le modèle est représenté en rouge, et les huit courbes bleues sont issues d'une série de huit mesures de la STDP réalisée en appliquant les créneaux de tension de la (figure 2.10) et avec une initialisation de la résistance comprise dans l'intervalle [500kΩ; 650kΩ] pour Δt < 0ns et [9MΩ; 15MΩ] pour Δt >0ns. La simulation du modèle a été réalisée sous Matlab dans la même condition que le Verilog A, en appliquant exactement les mêmes créneaux et en prenant le même intervalle de résistance d'initialisation que les mesures. La résistance évolue en fonction du ratio des domaines fermés sur le nombre de domaines total S selon l'équation (2.2.2) avec les paramètres suivants qui ont été pris. Pour la simulation du modèle (figure 2.10 b), on a choisi R_{init} = 650 kΩ pour Δt < 0ns et R_{init} = 9.5 MΩ pour Δt > 0ns, les résistances Ron=400 kΩ et Roff = 100MΩ. Le changement des domaines (S) évolue en fonction de la tension appliquée aux bornes du memristor (V) de la manière suivante :

$$(2.3)\quad S_{\pm}(t,V) = \frac{1}{2} \pm \frac{1}{\pi}\, arctan \frac{log\big(t_{mean}(V)\big) - log(t)}{\Gamma(V)}$$

Avec $\Gamma(V)$ la largeur et $\log\big(t_{mean}(V)\big)$ le centre de la distribution Lorentzienne pour une valeur V appliquée aux bornes du memristor, du logarithme des temps de nucléation du matériau ferroélectrique. Le modèle reproduit l'allure des mesures de la variation de conductance. La variation de conductance mesurée pour un même Δt est causée par plusieurs facteurs différents, ceux qui ont été pris en compte sont la variation de transition et la variation de résistance initiale.

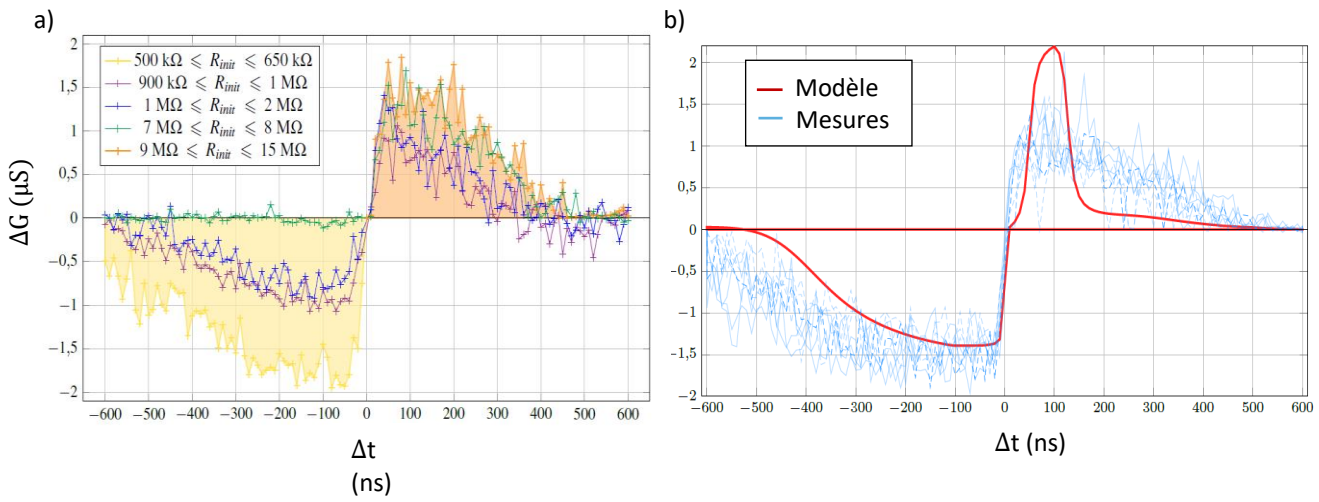


Figure 2.10 : A) Mesure de la variation du poids synaptique du memristor en fonction de Δt et de sa résistance initiale. B) Modèle du memristor qui reproduit 8 mesures de STDP sur un même memristor avec une R_{init} appartenant à l'intervalle [500kΩ; 650kΩ] pour Δt <0ns et une R_{init} appartenant à l'intervalle [9MΩ; 15MΩ] pour Δt >0ns.

Modèle avec bruit de transition

Le modèle comprenant le bruit de transition de résistance est basé sur le modèle précédent auquel on a ajouté une variabilité sur le temps d'application de la tension sur le modèle original. Sur la (figure 2.11) sont présentées deux simulations du modèle comprenant le bruit pris en compte pour la comparer avec les mesures qui ont été faites et évaluer le bon fonctionnement du modèle. Le bruit ajouté au modèle permet d'être plus fidèle aux mesures, en termes de variation, mais également plus fidèle aux mesures générales comparées au modèle sans bruit (figure 2.10 b).

La résistance initiale pour Δt<0ns est comprise entre 500kΩ et 650kΩ avec une résistance du memristor minimale (Ron) de 400 kΩ et maximal (Roff) de 100MΩ . Pour se mettre le plus près possible des conditions de mesures, durant la simulation la résistance initiale est tirée aléatoirement entre 500kΩ et 650kΩ et en appliquant bien sûr les mêmes signaux que les mesures. On tire également aléatoirement la résistance initiale du memristor comprise entre 9MΩ et 15MΩ pour les Δt >0ns.

Pour les hautes valeurs de résistance, si la tension appliquée au memristor est négative dans le but de la faire décroitre, on ajoute un bruit sur la résistance minimale Ron uniquement lors de la transition. La valeur minimale revient à l'origine après. La variation de résistance est la suivante :

$$(2.4)\ Ron_{equ} = Ron * \left(1 + \frac{Rinit}{12 * 10^6}\right) + Aléa1 * \frac{Rinit}{45 * Aléa2}$$

avec Ron_{equ} la résistance minimale qui sert uniquement pour les calculs intermédiaires de la résistance, Rinit la résistance avant son changement, Aléa1 et Aléa2 deux variables aléatoires différentes tirées d'une distribution gaussienne de moyenne nulle et d'écart type 1. Dans le modèle Verilog A, on calcule la

variation de domaine S qui est directement proportionnelle à la résistance du memristor. La résistance du memristor calculée à partir de S dépend de Ron et Roff. Pour calculer la résistance finale R avec le bruit on prend la formule (2.2) $\frac{1}{R} = S\left(\frac{1}{R_{off}} - \frac{1}{R_{on}}\right) + \frac{1}{R_{on}}$ en remplaçant Ron par Ron_{equ} ce qui nous donne $\frac{1}{R} = S\left(\frac{1}{R_{off}} - \frac{1}{R_{equ}}\right) + \frac{1}{R_{equ}}$. Enfin pour que le modèle soit fonctionnel on recalcule la valeur de S en remettant Ron à sa valeur et R par la résistance calculée précédemment et nous donne $S = \frac{\frac{1}{R} - \frac{1}{R_{on}}}{\frac{1}{R_{off}} - \frac{1}{R_{on}}}$.

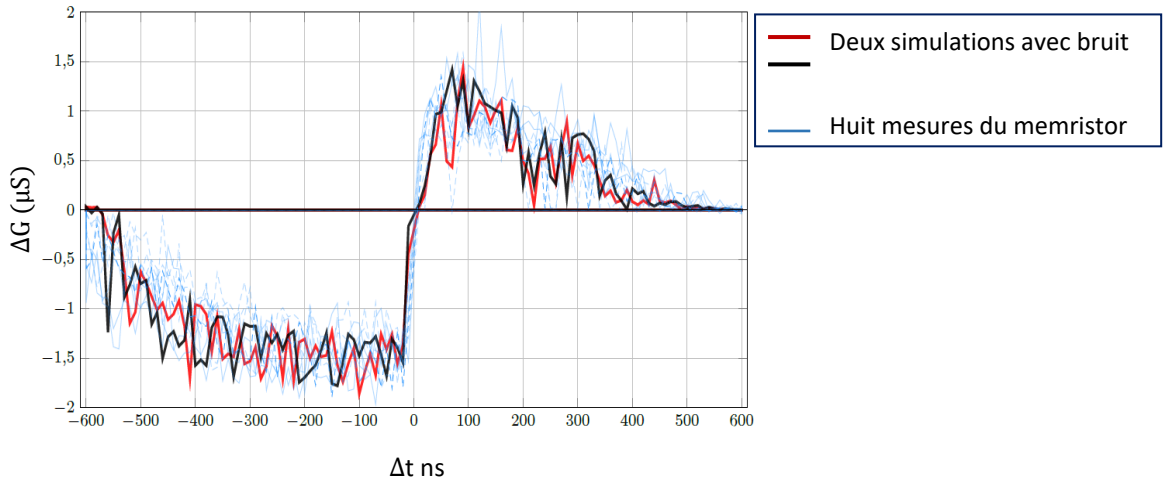


Figure 2.11 : Modèle du memristor avec variabilité de transition qui reproduit 8 mesures de STDP sur un même memristor avec un R_{init} appartenant à l'intervalle [500kΩ; 650kΩ] pour Δt <0 et [9MΩ; 15MΩ] pour Δt >0.

Distribution de résistance minimale et maximale

Le modèle Verilog A prend également en compte la variation de résistance minimale R_{on} et maximale R_{off} du memristor. Ces données sont obtenues à partir des mesures faites par notre partenaire l'Unité Mixte de Physique CNRS-Thales. La variation est un peu plus complexe qu'une simple distribution gaussienne. Si on passe en échelle logarithmique les ordonnées (échelle des résistances), il apparait une répartition d'une somme de deux lois normales sur les distributions des R_{on} et R_{off}. Pour établir ce modèle, on cherche tout d'abord à trouver deux sommes de deux distributions Gaussiennes qui s'approchent de a) et b) sur la (figure 2.12) en échelle logarithmique. Les deux distributions sont de la forme suivante :

-R_{on} : X_{on} = distribution gaussienne 65% de probabilité d'avoir une moyenne 4,1 et un écart type de 1,7 et 35% de probabilité d'avoir une moyenne 7,77 et un écart type de 2.

-R_{off} : X_{off} = distribution gaussienne 60% de probabilité d'avoir une moyenne 9,83 et un écart type de 0,55 et 40% de probabilité d'avoir une moyenne 10,2 et un écart type de 4.

Pour obtenir la valeur de la résistance Ron et Roff à partir de X_{on} et X_{off} afin de la convertir depuis l'échelle logarithmique on applique la formule suivante :

$$R_{on} = 89k\Omega * 1.25^{X_{on}} = 89k\Omega * \exp(X_{on} * \ln(1.25))$$
$$R_{off} = 3.55Mk\Omega * 1.259^{X_{off}} = 3.55Mk\Omega * \exp\big(X_{off} * \ln(1.259)\big)$$

La fonction de répartition de R_{on} et R_{off} est fonction des variables aléatoires X_{on} et X_{off} qui obéissent aux deux lois normales décrites précédemment. Elle est également fonction du paramètre de la résistance minimale de chaque histogramme 89kΩ pour R_{on} et 3,33MΩ pour R_{off}, mais également du rapport entre chaque barre d'histogramme qui est de 1,25 pour les R_{on} et de 1,259 pour les R_{off}.

Sur la série de mesures réalisée, nous n'avons pas constaté de corrélation entre les valeurs de résistances minimales et maximales sur un memristor. Un memristor peut avoir une grande résistance « on » et une faible résistance « off » ou inversement.

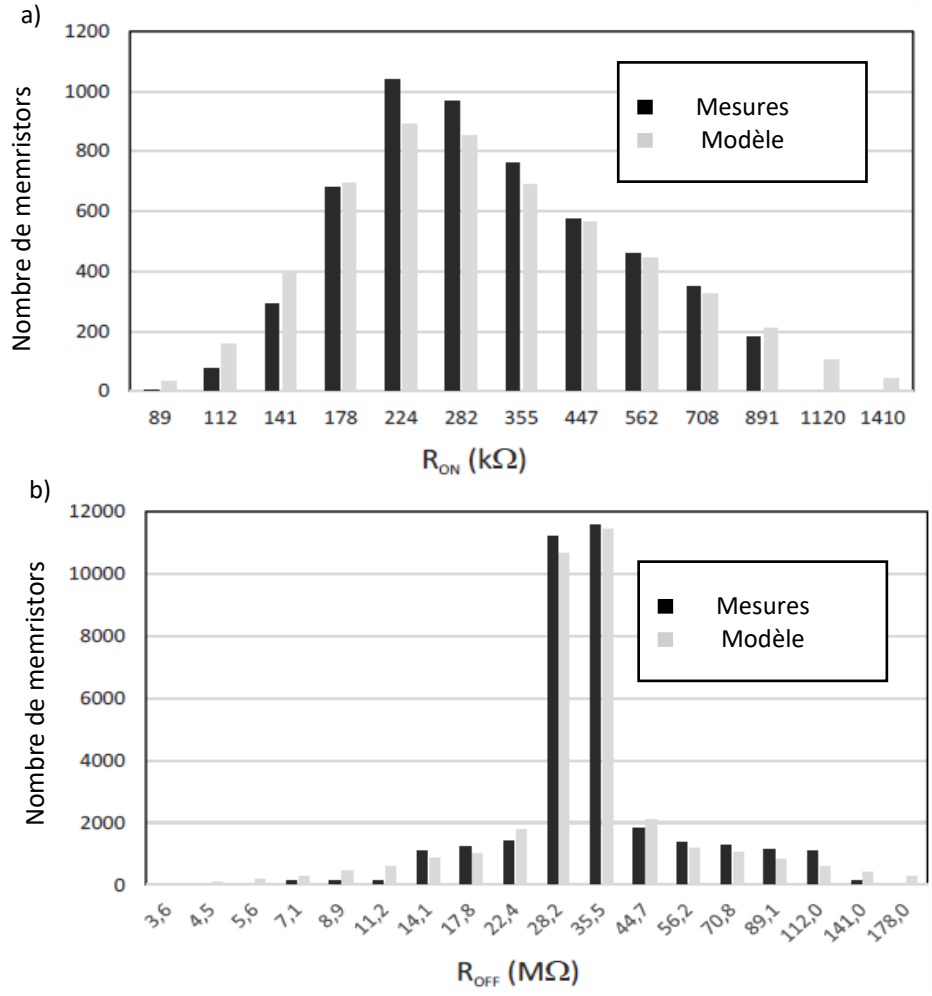


Figure 2.12 : Distribution de R_{on} et R_{off} sur les memristors ferroélectriques BTO.

Cette modélisation de résistance Ron et Roff est valable pour un procédé de fabrication donné. Si les valeurs du modèle ne peuvent pas être généralisées sur tous les memristors ferroélectriques, ce modèle présente tout de même l'intérêt de la prise en compte de toutes les variabilités de ce composant : résistance minimale, résistance maximale et variation de la résistance suite à l'application d'une tension au-delà du seuil.

Temps de transition

Pour notre modèle Verilog A, il faut aussi prendre en compte la variation de tension aux bornes du memristor et modifier en temps réel sa résistance. Or dans le modèle [BOYN S. 2017], la variation de la résistance est appliquée au memristor une fois que la tension aux bornes du memristor est remise à zéro, et n'est valable que pour des signaux carrés. Il a donc fallu l'adapter pour qu'il puisse prendre en compte les signaux transitoires et calculer sa résistance à chaque moment.

Lorsque l'on souhaite faire une simulation temporelle sur Cadence, il faut pouvoir changer la résistance du memristor en temps réel. On ne peut pas se permettre de changer la résistance du memristor une fois l'application du signal terminée comme dans le modèle de [BOYN S. 2017] car d'une part la durée du pulse est inconnue et d'autre part la tension aux bornes du memristor n'est pas forcément constante. Le modèle Verilog A doit nous permettre de connaitre la résistance du memristor à chaque instant pour être intégrés à l'outil de conception Cadence.

Lors d'une mesure sur le memristor, on a l'information concernant la résistance uniquement avant et après l'application de la tension, la variation de la résistance étant inconnue. Le modèle original effectue les calculs de la modification de résistance du memristor et donne les mêmes résultats que l'on applique un seul pulse à ses bornes de durée Δt ou de n pulses de durée $\Delta t/n$ chacun. De ce fait, le modèle s'adapte bien à l'échantillonnage. Nous avons choisi un temps d'échantillonnage de 10ns comme compromis : pas de temps pour réaliser les calculs rapidement tout en restant dans le même ordre de grandeur que les signaux appliqués dans le projet ULPEC qui sont de l'ordre de quelques centaines de nanosecondes. Une autre contrainte apparait, celle du choix de la valeur de tension appliquée pour le changement de résistance du memristor.

2.5 Implantation de l'apprentissage dans le système ULPEC

A l'origine le memristor qui doit être utilisé pour le crossbar et pour lequel le modèle Verilog A a été modélisé est un ferroélectrique de type $BaFeO_3$. Cependant après des difficultés de fabrication ce type de memristor, on utilisera finalement un memristor avec matériau ferroélectrique de conception $BaTiO_3$. Il en résulte des caractéristiques physiques différentes dont notamment une augmentation de la résistance minimum et maximum, mais surtout une tension de seuil du memristor de l'ordre de 2V, ce qui pose des

contraintes supplémentaires pour la conception du circuit intégré qui fonctionnera avec une alimentation de 3,3V.

Architecture

Les memristors du projet ULPEC sont réalisés sous la forme d'une matrice, ce qui confère au réseau la propriété d'une interconnexion dite all-to-all : chaque neurone pré-synaptique est connecté à tous les neurones post-synaptiques par un memristor (figure 2.13).

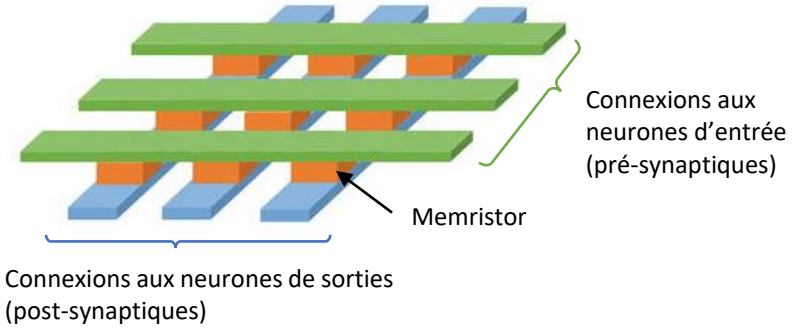


Figure 2.13 : Représentations 3D de la matrice de memristors.

Notre réseau de neurones est constitué de 784 (28 x 28) neurones d'entrée, qui sont chacun connectés directement à un pixel de la caméra évènementielle. Quand la caméra détecte une variation de luminosité et qu'elle est suffisamment importante, qu'elle soit positive ou négative, la caméra envoie un évènement au neurone d'entrée correspondant à son pixel. Le neurone d'entrée génère alors un potentiel d'action « down » pour charger les neurones de sortie appelés neurones post-synaptiques. Il y a 100 neurones de sortie qui doivent chacun apprendre un unique motif. Durant la phase d'apprentissage, dès qu'un neurone post-synaptique a reconnu un motif, il y a augmentation du poids des synapses reliant ce neurone post-synaptique aux neurones pré-synaptiques qui ont participé à son déclenchement. Il y a diminution du poids de toutes les autres synapses du réseau, tous les neurones sont ensuite remis à zéro. La remise à zéro des neurones post-synaptiques est nécessaire, car nous souhaitons faire la reconnaissance d'images qui sont considérées comme des apprentissages séparés les uns des autres. Cette méthode est appelée « winer takes all » et est utilisée dans les circuits bio-inspirés utilisant la STDP [FERRE P. 2018], [KAPPEL D. 2014]. Cette technique permet de faire apprendre tous les neurones chacun à son tour de manière la plus équitable possible.

Tension d'application

Pour des raisons de facilité de conception, plutôt que d'utiliser un potentiel d'action avec une rampe sur la partie négative comme présentée à la (figure 2.9), les potentiels d'action appliqués ont des formes carrées voir (figure 2.14). Les potentiels d'actions présynaptiques apparaissant avant le déclenchement

du neurone post-synaptique utilisent uniquement la partie négative, et ceux venant après uniquement la partie positive. Pour cette raison on a décidé de séparer le potentiel d'action en deux signaux distincts « up » et « down ». Le potentiel d'action des neurones post-synaptiques est quant à lui bien en une seule partie. La tension au repos se trouve à 1,1V, la tension « up » à 2,2V et la tension « down » à 0.1V, ces tensions sont les mêmes pour les neurones pré et post-synaptique. Le potentiel d'action « down » permet de charger la capacité de membrane des neurones post-synaptique en fonction du poids des synapses et permet potentiellement d'augmenter le poids d'une synapse. Sa durée est comprise entre 5µs et 10µs. Le potentiel d'action « up » à la seule utilité de diminuer le poids des synapses concernées, il dure entre 200ns et 500ns. Sur la (figure 2.14) sont présentés les potentiels d'action pré-synaptique « up » et « down », respectivement Vpre1 et Vpre2. On voit la tension résultante sur les deux memristors en Vmem1 et Vmem2. Le memristor 1 a une tension à ses bornes qui dépasse le seuil des 2V, sa résistance sera augmentée. Quant au memristor 2, il subit une tension inférieure à -2V et verra sa résistance diminuer.

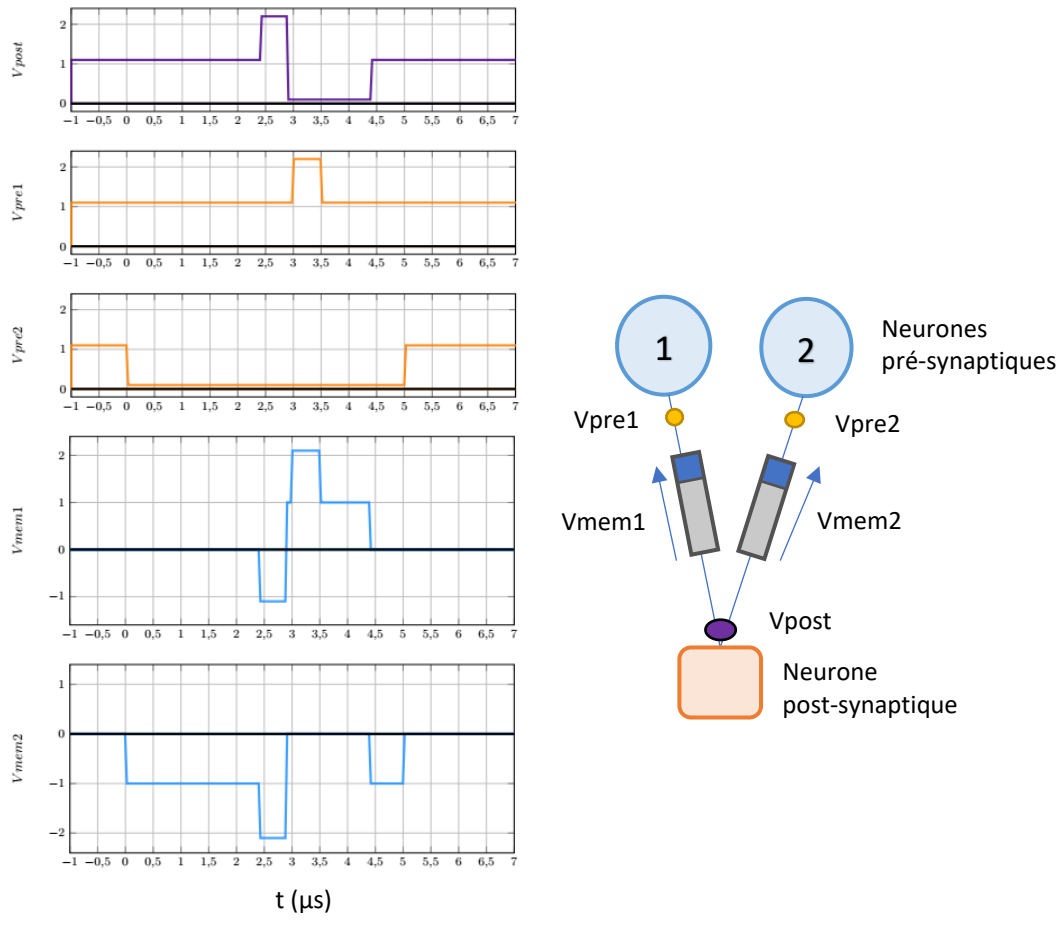


Figure 2.14 : Potentiels d'action appliqués sur les memristors par les neurones pré et post-synaptiques.

2.6 Conclusion

Ce chapitre nous a permis de comprendre le fonctionnement du memristor qui est utilisé comme synapse dans notre réseau de neurones. Nous avons expliqué ses propriétés ferroélectriques qui lui permettent de changer sa résistance quand on lui applique des tensions supérieures à environ 2V en valeur absolue. Si les tensions sont inférieures en valeur absolue à 2V le memristor se comporte comme une résistance dont sa valeur reste fixe. Nous avons ensuite présenté un modèle Verilog A du comportement du memristor ferroélectrique qui prend en compte la variabilité des valeurs minimales et maximales de la résistance ainsi que la variabilité du changement de valeur de cette résistance lors de l'application des créneaux de tension. Le modèle est compatible avec le pas de calcul du logiciel de conception Cadence. Nous avons par la suite présenté comment nous implanterons la méthode d'apprentissage non supervisé.

Dans la suite de ce manuscrit nous allons voir les contraintes rencontrées sur le réseau de neurones et présenter en détail les solutions retenues pour la conception des blocs numériques et analogiques.

CHAPITRE 3 : CONCEPTION ELECTRONIQUE DES NEURONES

Cette partie a pour objectif d'introduire les contraintes et problèmes pour comprendre le fonctionnement du réseau de neurones et de présenter les blocs analogiques qui les résolvent. L'objectif est de faire un réseau de neurones en utilisant la caméra évènementielle de Prophesee ainsi que les memristors comme synapses permettant le traitement et la reconnaissance d'image en utilisant principalement la MNIST (liste de 60 000 chiffres manuscrits et labellisés). Pour interpréter les données de la caméra, nous avons créé un bloc numérique permettant la lecture de la caméra évènementielle et faisant la gestion de l'apprentissage, il permet de déclencher les neurones pré et post-synaptiques désirés. Ce circuit numérique sera expliqué en détail dans le prochain chapitre.

Concernant les blocs analogiques, les neurones pré et post-synaptiques doivent pouvoir modifier le poids des memristors en appliquant les tensions adéquates, suffisamment rapidement tout en délivrant assez de courant pour maintenir les tensions souhaitées. Le point crucial de la conception concerne la charge de toutes les capacités de membrane des neurones post-synaptiques, qui doit être proportionnelle à la valeur de la conductance du memristor qui est connecté au neurone pré-synaptique qui a déclenché. La matrice de memristors de grande taille génère de grosses contraintes qui sont non négligeables pour les réseaux d'une telle taille. Comprendre et résoudre ces contraintes est essentiel afin de pouvoir réaliser un apprentissage avec notre réseau de neurones.

3.1 Description du réseau de neurones et contraintes

Description du réseau de neurones

Notre réseau de neurones comporte 784 neurones d'entrée appelés neurones pré-synaptiques, qui sont chacun d'entre eux connectés aux 100 neurones de sortie appelés neurones post-synaptiques, la connexion du réseau est dit « all to all ». Les 78 400 memristors font le lien entre les neurones pré et post-synaptiques (figure 3.1). Les neurones d'entrée reçoivent les informations de la caméra évènementielle qui détecte la variation de luminosité, chaque neurone pré-synaptique est commandé par un seul et unique pixel de la caméra. Chaque neurone post-synaptique a une capacité de membrane, qui se charge par le courant provenant de tous les memristors reliés à ce neurone ; le courant qui traverse un memristor est proportionnel à sa conductance.

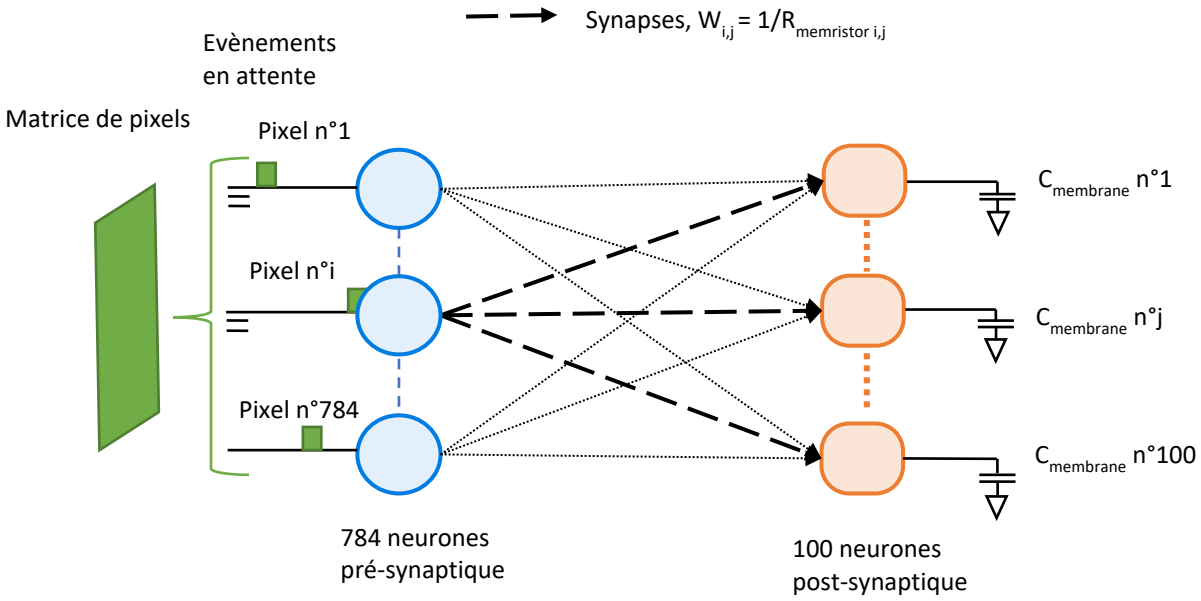


Figure 3.1 : Représentation du réseau de neurones

Les neurones pré-synaptiques génèrent une tension à leur sortie dans le but de charger les capacités de membrane et de pouvoir durant une autre phase écrire sur les memristors. Les neurones post-synaptiques quant à eux imposent une tension à leur entrée dans le but dans un premier temps de maintenir une tension et d'imposer un courant dans les memristors afin de charger la capacité de membrane et de pouvoir dans un second temps réaliser un apprentissage en changeant la conductance des memristors. Les neurones pré et post-synaptiques sont connectés entre eux par un memristor qui joue le rôle de synapse, dont la valeur de son poids synaptique correspond à la valeur de sa conductance électrique. Au repos, quand aucun évènement n'arrive depuis la caméra, toutes les tensions générées par les neurones pré et post-synaptiques sont à une tension égale à Vmid (1.1V). De ce fait, au repos aucun courant ne circule dans les memristors.

Le déclenchement d'un neurone post-synaptique représente le moment où il a reconnu un motif envoyé par la caméra sur le réseau de neurones. Ce déclenchement a lieu quand la tension de la capacité de membrane d'un neurone post-synaptique a atteint un certain seuil, ce seuil étant le même pour tous les neurones post-synaptiques.

La charge de l'ensemble des capacités de membranes peut être représentée sous forme de matrice. Le crossbar de memristors est utilisé pour faire un produit matriciel entre les neurones d'entrée et la conductance des memristors, le résultat est homogène à un courant. $W_{i,j}$, représente le poids d'une synapse de la ligne i et de la colonne j du crossbar de memristor.

Le courant provient uniquement des neurones pré-synaptiques qui ont été déclenchés par la caméra. Le courant qui entre dans le neurone j a un instant t est suivant l'équation (3.1) :

$$(3.1)\ i_{neurone\ j} = \sum_{i=1}^{i=784} (Vpre_i - Vmid) * W_{ij}$$

Le courant entrant est copié du neurone pré-synaptique vers la capacité de membrane du neurone post-synaptique en subissant une division du courant K. Le courant doit être divisé pour ne pas que la capacité de membrane se charge trop rapidement, les capacités intégrées étant de très petites valeurs.

L'équation 3.2 donne la variation de tension de membrane des neurones post-synaptiques dans un intervalle de temps donné. Vmembrane représente la matrice des tensions de membrane de chaque neurone post-synaptique, Vpré la matrice des tensions des neurones pré-synaptiques. $C_{membrane}$ est la capacité de membrane de tous les neurones post synaptiques et sont identiques pour tous les neurones aux variations de fabrication près, i_{leak} est le courant de fuite de chaque neurone postsynaptique également identique.

$$(3.2)\ \Delta V_{membrane\ j} = \frac{\int_{t0}^{t1} (\sum_{i=1}^{i=784} W_{i,j} * (Vmid - Vpre_i))dt/K\ - i_{leak} * (t1 - t0)}{C_{membrane}}$$

Quand la tension de membrane d'un des neurones post-synaptiques a atteint la tension de seuil $Vth_{membrane}$, le neurone en question déclenche et met à zéro la tension de membrane de tous les autres neurones.

3.2 Présentation du neurone post-synaptique

Dans le but de faire faire un réseau de neurones bio-inspiré, nous utiliserons un modèle de neurone biologique facilement implantable en utilisant l'électronique. Le modèle du neurone LIF qui en anglais signifie « Leaky Integrate and Fire » est suffisamment fidèle pour être utilisé avec la loi de Hebb pour réaliser un apprentissage proche de la biologie. Ce modèle LIF est également bien plus rapide à la simulation qu'un modèle plus complet et réaliste [GUPTA A. 2009], ce qui a l'avantage de simuler le réseau de neurones et de faire le dimensionnement nécessaire en vue de la conception bien plus rapidement.

Notre neurone post de type LIF présenté (figure 3.2), est composé d'un convoyeur de courant qui impose la tension Vpost de Y en X et copie le courant de X en Z. Autrement dit, le courant traversant les memristors charge la capacité de membrane $C_{membrane}$. Dès que la tension de membrane dépasse le seuil Vth le bloc Vpost génère un pulse de tension qui est copié de Vy vers Vx par le convoyeur de courant. Le bloc Vpost remet aussi à zéro la tension de Cmembrane.

La capacité de membrane que nous utilisons a une valeur de 1pF, par conséquent les courants doivent être de faible intensité pour ne pas charger ou décharger cette capacité trop rapidement. Le courant de décharge i_{leak} est de l'ordre de 100pA. Dans le modèle LIF original [ABBOTT L. 1999], une résistance

permet de faire une décharge en courant exponentielle, ce qui pour nos ordres de grandeur de courant est irréalisable avec une résistance qui devrait être de l'ordre de 10GΩ. Nous utilisons donc un transistor en fonctionnement sous le seuil piloté, ce qui a pour conséquence d'avoir un courant de décharge constant et non exponentiel.

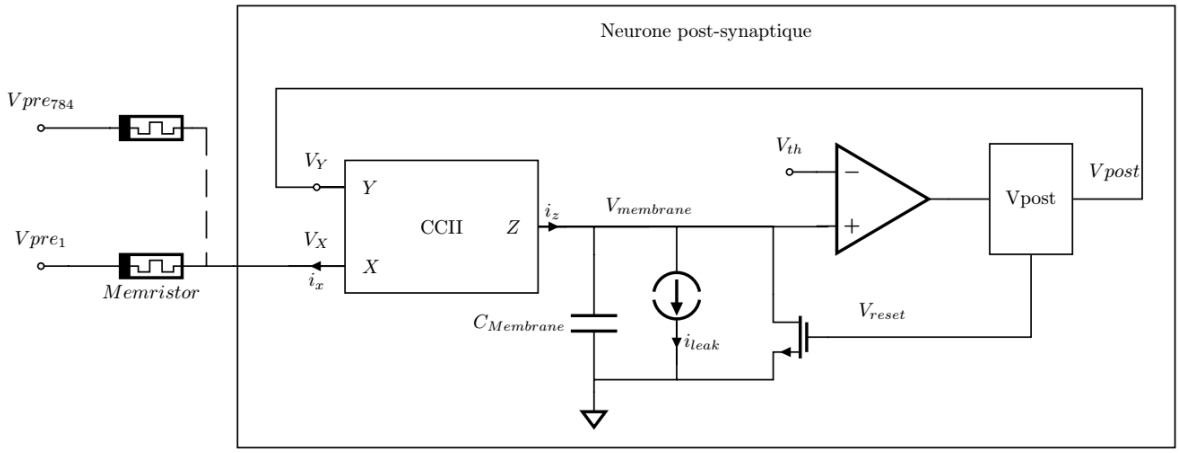


Figure 3.2 : Représentation du neurone LIF postsynaptique.

Contrainte du réseau de neurones en microélectronique.

Pour s'assurer que tous les signaux maintiennent la tension qu'il leur est imposée, les neurones pré et post-synaptiques doivent pouvoir fournir un courant suffisamment important dans toutes les configurations du réseau. Les 784 neurones pré-synaptiques sont connectés à chacun des 100 neurones de sortie par un memristor. Les neurones doivent alors pouvoir délivrer une tension allant de 0,1V à 2,2V quelle que soit la valeur des memristors pour la génération des signaux pré et post-synaptiques. La résistance minimale des memristors utilisés dans ces travaux est de 1MΩ et leur résistance maximale est supérieure à 100MΩ. En sortie du neurone post-synaptique le convoyeur de courant doit pouvoir délivrer une tension de 2,2V ou 0,1V sur une résistance équivalente de 1.28kΩ ce qui correspond à la mise en parallèle des 784 memristors à leur valeur minimale de résistance à 1MΩ. De l'autre côté, les neurones pré-synaptiques doivent pouvoir délivrer également 0,1V ou 2,2V sur une résistance équivalente minimale du réseau de 10kΩ, ce qui correspond à 100 résistances en parallèle de 1MΩ. Il faudra donc concevoir et s'assurer que les circuits électroniques puissent fournir ces courants tout en respectant la forme des tensions désirées.

L'offset sur l'entrée X du convoyeur de courant est un problème majeur qu'il faut résoudre. Pour illustrer son importance on fera une étude de cas avec et sans offset. En se focalisant uniquement sur un seul neurone post-synaptique, et lorsqu'un et un seul neurone pré-synaptique a déclenché un pulse « down », un courant circule dans le memristor relié à aux neurones pré et post-synaptique, le CCII copie ce courant

et l'envoie vers la capacité de membrane (voir figure 3.3). Le déclenchement du neurone pré-synaptique fait chuter la tension sur l'entrée X du convoyeur de courant, le CCII effectue une correction en tension de l'ordre du milli Volt pour ramener la tension en X à Vmid, ce qui génère ainsi un courant qui sera copié vers la capacité de membrane, illustration (figure3.3). Dans le cas du déclenchement du neurone pré-synaptique, Vpre est égale à Vdown soit 0.1V. La chute de tension que le CCII doit corriger est fonction de la résistance du memristor, de la résistance équivalente de tous les autres memristors mis en parallèle ($R_{network}$), des tensions Vmid et Vdown. Si la tension en sortie du convoyeur de courant est parfaitement égale à Vmid, aucun courant ne circule dans Rnetwork, cela implique que tout le courant passant dans le memristor est copié vers la capacité de membrane.

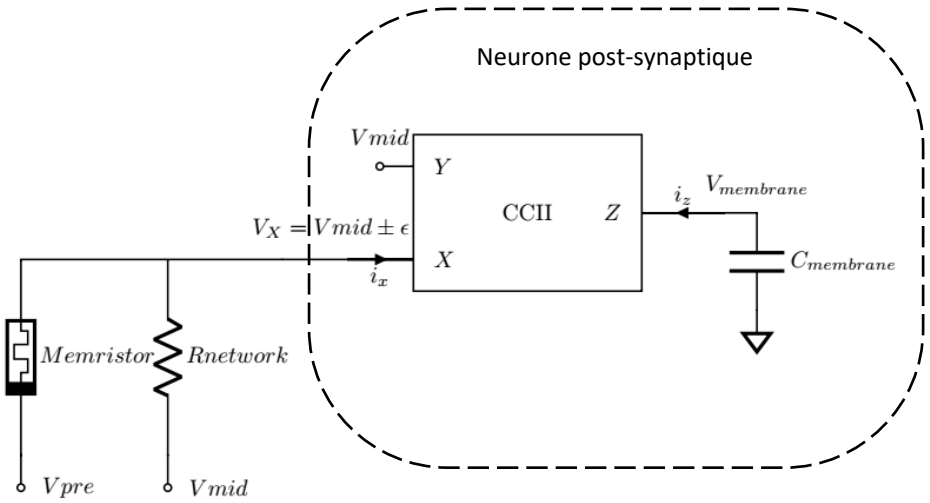


Figure 3.3 : Memristor et résistance équivalente de réaux de memristor (Rnetwork) vu du CCII du neurone post-synaptique.

La tension sur l'entrée X du convoyeur de courant à cause de l'offset ne sera pas égale à Vmid soit 1.1V. Il y aura donc toujours un courant qui circulera dans la branche X du CCII. Dans le cas où Vpre est égale à Vmid soit 1,1V, le courant i_x du CCII est le suivant (3.3)

$$(3.3)\ i_x = -\frac{\varepsilon}{R_{network}} - \frac{\varepsilon}{R_{memristor}}$$

Avec ε l'offset du convoyeur de courant. Dans le cas ou l'offset est négatif (Vx supérieur à Vmid) le courant i_x est négatif, il sort donc du CCII et charge en continu la capacité de membrane, le neurone risque de déclencher de manière intempestive. Il faut donc impérativement que l'offset du convoyeur de courant soit positif.

Dans le cas où la tension Vpre est égale à Vdown soit 0,1V le courant i_x est défini suivant l'équation suivante (3.4) :

$$(3.4)\ i_x = -\frac{\varepsilon}{R_{network}} + \frac{-V_{mid} + V_{down} - \varepsilon}{R_{memristor}}$$

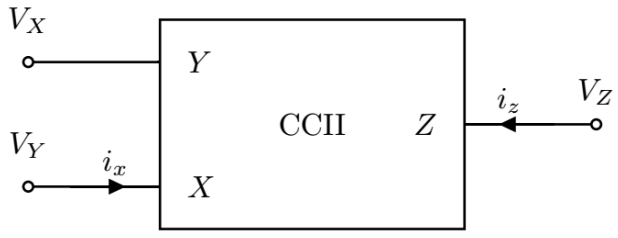
Pour illustrer le problème, et à titre d'exemple si la valeur de tous les memristors est identique, la valeur de $R_{networok} = R_{memristor}/783$. On se placera également dans le cas où l'offset est positif pour que le réseau de neurones soit fonctionnel, le courant i_x vaut ainsi la valeur suivante (3.5) :

$$(3.5)\ i_x = -\frac{783 * \varepsilon}{R_{memristor}} + \frac{-V_{mid} + V_{down} - \varepsilon}{R_{memristor}}$$

Pour pouvoir charger la capacité de membrane le courant i_x doit être négatif, et donc $\frac{-V_{mid}+V_{down}-\varepsilon}{R_{memristor}}$ dois être inférieur à $\frac{783*\varepsilon}{R_{memeristor}}$, sinon le courant i_x est positif et ne chargera pas la capacité de membrane et le neurone ne déclenchera jamais. Avec nos valeurs, l'offset doit être inférieur à 1,28mV pour qu'un évènement de provenant de la caméra puisse charger la capacité de membrane. Lors du fonctionnement du réseau, il y aura surement plusieurs neurones pré-synaptiques qui déclencheront leur pulse « down » en même temps. Il est nécessaire de corriger l'offset de chacun de nos 100 neurones post-synaptiques indépendamment et que leur valeur soit le plus proche possible de 0V tout en étant impérativement positif. L'offset du convoyeur de courant estimé par simulation est de plus ou moins 25mV ce qui rendra le réseau non fonctionnel. Une correction de l'offset s'impose donc.

Convoyeur de courants de seconde génération (CCII)

Le convoyeur de courant a été inventé par Sedra et Smith en 1968 [SEDRA A. 1989]. Nous utilisons celui de seconde génération dans notre neurone post-synaptique. Le principe de fonctionnement du convoyeur de courant est présenté (figure 3.4) avec sa matrice liant les tensions et les courants. Il permet de copier la tension de V_y vers V_x, et de copier le courant de i_x vers i_z ou - i_z. Dans notre cas nous utilisons un diviseur de courant qui permet de réduire la quantité de courant qui est copié de X vers Z. Il n'y a aucun courant qui rentre en Y.



$$\begin{pmatrix} i_Y \\ V_X \\ i_Z \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & \pm 1 & 0 \end{pmatrix} * \begin{pmatrix} V_Y \\ i_X \\ V_Z \end{pmatrix}$$

Figure 3.4 : Convoyeur de courant de seconde génération avec sa matrice hybride donnant les les recopies en tensions et en courants

Nous pouvons distinguer deux phases bien distinctes concernant l'apprentissage de notre réseau de neurones. La première est la phase qui consiste à charger la capacité de membrane : le convoyeur doit maintenir avec précision la tension Vmid en X et copier le courant qui y circule vers la capacité de membrane. Dans cette configuration, le courant en X n'est pas très grand, car il n'y a que quelques neurones pré-synaptiques qui ont déclenché en même temps. Dans la seconde phase, le convoyeur de courant doit recopier en X un pulse de tension provenant du neurone post-synaptique dans le but de modifier le poids des memristors. Durant cette phase le courant est très important, mais durant cette phase il est inutile de copier le courant pour charger la capacité de membrane, car le neurone post-synaptique a déjà déclenché.

Les topologies conventionnelles des CCII (figure 3.5) utilisent une polarisation en courant dans la branche X et Z [FERRI G. 2004]. Dans les deux cas, la tension est copiée de Y vers X, soit en utilisant des miroirs de courant (a) soit une paire différentielle (b). Le courant est copié de X vers Z par des miroirs de courants.

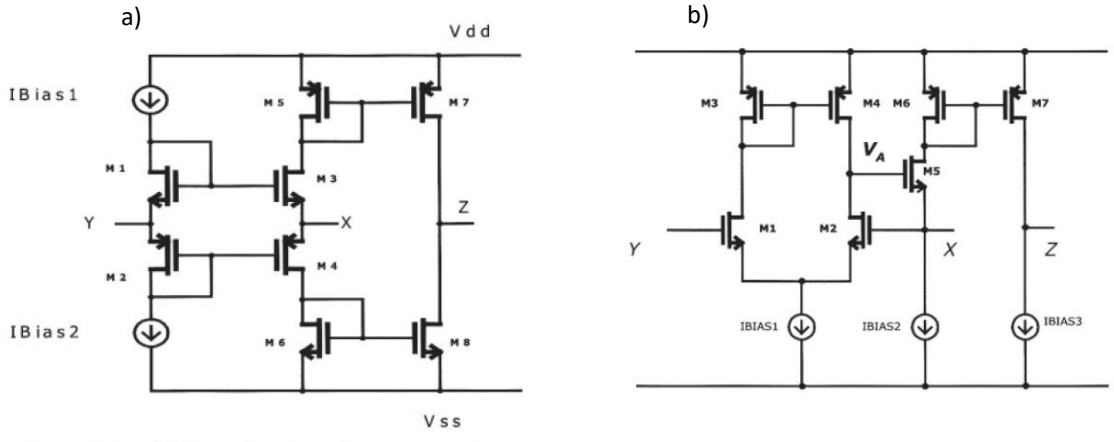


Figure 3.5 : Topologies de base pour un convoyeur de courant a) topologie basée sur des miroirs de courant b) topologie basée sur une paire différentielle.

Le problème que nous rencontrons avec ces deux topologies est que la variation de copie du courant par les miroirs de courant n'est pas parfaite, ce qui implique que le courant sortant en Z est différent de X. Cela provoque deux problèmes distincts. Le premier, si le courant sortant en Z est supérieur à celui en X, la capacité de membrane se charge sans qu'il y ait de courant venant des memristors, ce qui peut entrainer un déclenchement involontaire de ce neurone et donc nuire à l'apprentissage du réseau. Dans le second cas si le courant sortant de Z est inférieur à celui de X, alors un courant venant des memristors qui devrait charger la capacité de membrane peut être trop faible, il peut même décharger la capacité de membrane en continu et donc nuire également à l'apprentissage du réseau de neurones. Ces deux cas rendent notre réseau de neurones inutilisable. La charge ou décharge constante éventuelle est causée par

le courant de polarisation présent dans les branches X et Z. Il faudra donc concevoir un convoyeur de courant sans polarisation dans les branches X et Z. Dans notre utilisation nous souhaitons uniquement charger la capacité de membrane sans la décharger, nous ne copierons donc seulement le courant dans un seul sens.

Le convoyeur de courant utilisé est inspiré de la topologie de la (figure 3.5 b) à laquelle nous avons supprimé les polarisations des branches X et Z (IBAIS2 et IBAIS3) et remplacer la paire différentielle par un AOP, ce qui donne le schéma du circuit de la (figure 3.6 a). Avec cette topologie, l'avantage est de copier le courant seulement si Mn fournit du courant, donc seulement si la tension aux bornes de X est supérieure à la tension du réseau de neurones pré-synaptiques. Si la tension Vmid du réseau est légèrement plus élevée que celle en X, le courant ne peut pas circuler dans le convoyeur et est absorbé par les neurones pré-synaptiques, ce n'est en soi pas un problème. Cependant la tension V_{AOP} devient égale à zéro, car l'AOP est non contre-réactioné. Il faut alors un certain temps avant que Mn entre en conduction. De plus la mise en conduction génère quelques rebonds qui peut être préjudiciable pour la recopie du courant vers Z qui ne correspondrait plus au courant qui est traversé par le memristor. Pour remédier à ce problème et en s'inspirant de l'article de [SEDRA A.S 1990], le transistor Mp a été rajouté pour implémenter un push-pull (figure 3.6 b). Le push-pull utilisé est dimensionné pour que seul un des transistors Mn ou Mp soit conducteur à la fois. Le transistor M1 est présent pour augmenter la tension de grille de Mn par rapport à la tension V_{AOP} ce qui permet de pouvoir de changer la conduction entre Mp et Mn rapidement sans que les deux ne soient passants en même temps. Le courant de polarisation Ibias qui traverse M1 est absorbé par l'AOP. La capacité C1 permet durant la transition de transmettre directement la variation de la tension V_{AOP} générée par l'AOP vers Mn, évitant ainsi les rebonds, ce qui permet une meilleure copie en courant.

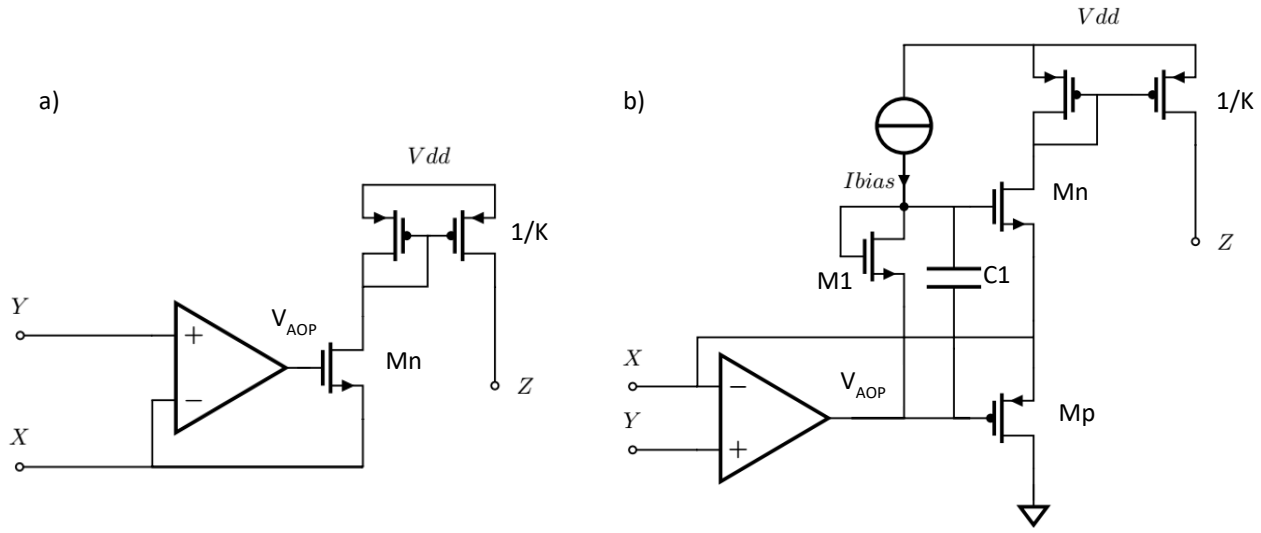


Figure 3.6 : a) topologie de base de notre CCII b) topologie de notre CCII amélioré par un push-pull.

Cette topologie (figure 3.6 b) fonctionne très bien pour copier le courant qui circule dans le memristor sans qu'il n'y ait de charge ou décharge intempestive de la capacité de membrane du neurone post-synaptique. La copie en courant peut avoir une variabilité de l'ordre d'une dizaine de pour cent, mais ce n'est pas un problème majeur pour un réseau de neurones qui est résilient à la variabilité de ses composants.

Le problème de cette topologie est qu'elle ne peut pas fournir beaucoup de courant et il est donc impossible de générer le pulse post-synaptique sur une charge de faible résistance. Cependant lorsque nous avons besoin de générer un pulse, la copie en courant devient inutile et le courant peut venir d'un autre transistor. Sur la (figure 3.7), le transistor Mn2 a été rajouté dans le but de fournir une grande quantité de courant et est toujours commandé par l'AOP. Il est indispensable de rajouter une polarisation forte en courant uniquement pendant la durée du pulse post-synaptique, ce qui en fait un amplificateur de classe A. Par les courants importants à fournir sur une grande plage de tension, allant de 0.1V à 2.2V, une technologie limitée à 3.3V avec une contrainte supplémentaire d'utiliser l'AOP pour commander la tension, il est difficile de se passer de l'architecture d'amplificateur de classe A qui est par nature énergivore. Cependant l'utilisation d'une telle configuration est très peu fréquente, utilisée uniquement pour le déclenchement du pulse post-synaptique sur un seul neurone à la fois et pour une durée de 1.5µs environ.

Sur la (figure 3.7) les switches S1, S2 et S3 ne sont pas des interrupteurs qui s'ouvrent et se ferment instantanément. Ils sont faits à base de transistors et laissent passer progressivement le courant pour éviter les variations de charge trop importante qui risqueraient de détériorer les memristors. Le transistor Mn2 est très large (w=100µm, L=0.28µm) pour pouvoir fournir beaucoup de courant tout en délivrant 2,2V sur l'entrée X du CCII. Il est en configuration suiveur par rapport à V_{AOP}. Pour une valeur de tension de 2,2V appliquée à sa source, et sachant que la tension de seuil de Mn2 est de l'ordre de 0,8V, il faut au moins 3V sur sa grille pour que Mn2 soit passant. Nous arrivons ainsi à la limite du 3,3V utilisé pour cette technologie. Il n'était pas envisageable d'utiliser un transistor de type P à la place de Mn2, car nous aurions créé un étage supplémentaire avec un retard supplémentaire et la marge de phase de notre convoyeur de courant aurait été diminuée. Il deviendrait alors plus compliqué de le stabiliser. Avec un transistor en suiveur, la stabilité n'est pas dégradée. Il y a deux polarisations en courant créées par Mpol1 et Mpol2, avec respectivement un courant de 150µA et de 2mA. Le courant de polarisation de 150µA est utilisé pour la stabilité du pulse « up ». Plus ce courant est important moins le rebond est prononcé lors du pulse up, mais en contrepartie, moins de courant peut être délivré par le convoyeur. Le courant de polarisation de 2mA quant à lui permet de générer le pulse « down » dont sa tension est contrôlée par le transistor Mn2.

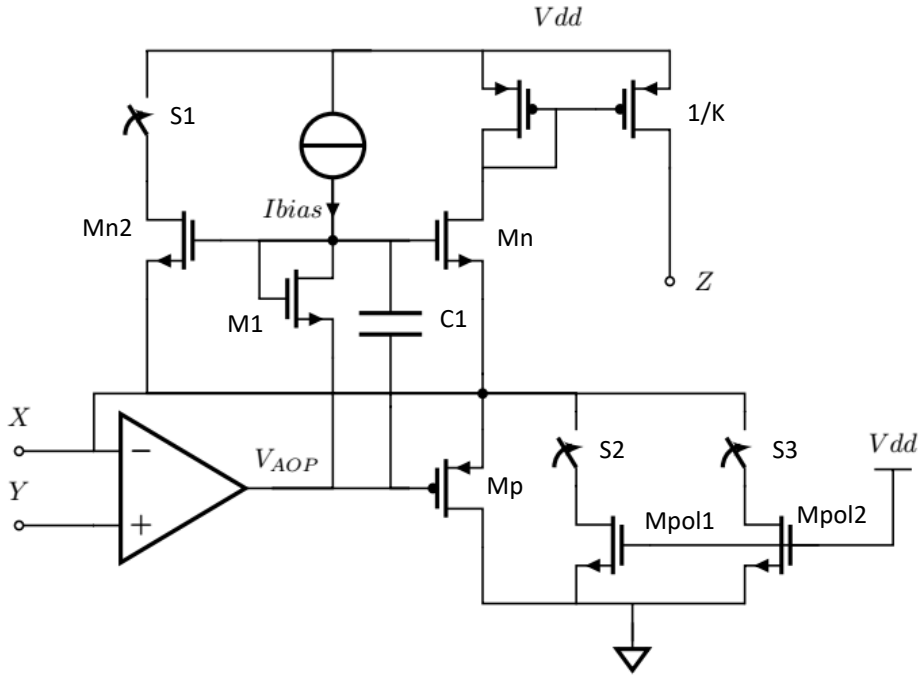


Figure 3.7 : CCII de classe A commandé par le transistor Mn2 avec les courants de polarisation variable en X de Mpol1(150µA) et Mpol2 (2mA).

Copie en courant par miroirs de courant

Le CCII copie les courants qui traversent les memristors pour charger la capacité de membrane. La capacité de membrane étant faible, 1pF, le courant doit être copié avec réduction pour ne pas charger le condensateur trop rapidement. La réduction du courant est dépendante de l'application visée par le système. Quand beaucoup d'évènements arrivent depuis la caméra, la réduction doit être plus forte que lorsque peu d'évènements arrivent. En fonction des différentes configurations souhaitées, nous pouvons choisir une division en courant de 25, 125 ou 600.

Le premier diviseur de courant présent sur la puce Orion – première puce développée durant ces travaux de thèse – n'était pas opérationnel à cause de l'utilisation de MOS sous le seuil, la division en courant était moins importante que prévue. La (figure 3.8) présente le schéma du nouveau diviseur de courant dans lequel les transistors sont dimensionnés pour ne pas être sous le seuil bien que les courants puissent être très faibles, de l'ordre de 20nA. Les transistors utilisés pour la recopie en courant avec division sont appairés et sont de même longueur. Pour une bonne division par 5 nous utilisons 5 fois le même transistor placé en parallèle au niveau des transistors équivalant M1, M3 et M8, et 8 transistors en parallèle pour une division par 8 au niveau du transistor équivalent M10. Les transistors M12 M13 et M14 placés en série permettent une division supplémentaire par 3 [BKER J. R. 2010]. Les transistors M25, M125 et M600 sont utilisés pour sélectionner la division en courant souhaitée. Ils ne servent que

d'interrupteurs et sont connectés à la sortie Z du convoyeur de courant qui est relié à la capacité de membrane.

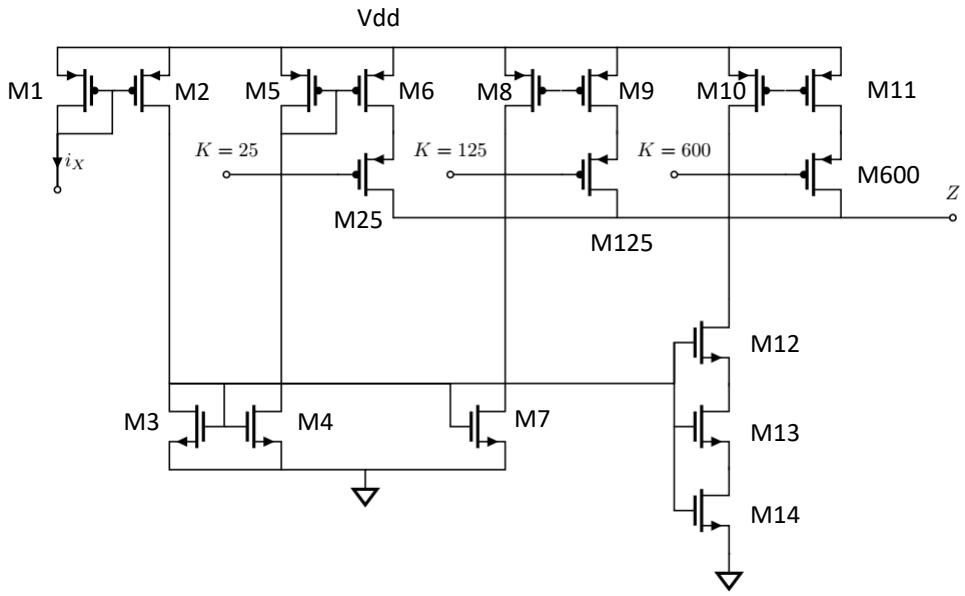


Figure 3.8 : Diviseur de courant du convoyeur de courant.

Convoyeur de courant complet

Après avoir expliqué le fonctionnement schématique du convoyeur de courant, nous allons présenter le schéma interne du convoyeur de courant (figure 3.9). Sur la (figure 3.7) nous avons présenté des switches. Ils étaient là pour illustrer la fonctionnalité du convoyeur de courant plutôt que de montrer le fonctionnement exact. Les trois switches sont commandés par deux signaux $V_{x\_controle}$ et V_{com_down} qui sont contrôlés par le générateur de pulse post-synaptique. Le switch S1 permet quand il fermé, si $V_{x\_controle}$=1, de contrôler le transistor Mn2 par l'AOP, sa tension de grille est égale à V1. Quand S1 est ouvert, c'est à dire quand $V_{x\_controle}$=0, le transistor Mn2 est bloqué. Durant le même temps où Mn2 est passant, Mpol1 fournit un courant de polarisation, car $V_{x\_controle}$=1. Le transistor Mpol2 fournit quant à lui un courant important uniquement pour la génération de la partie du pulse post-synaptique qui est égale à 0.1V ; Mpol2 est passant uniquement si V_{com_down}=1. Le signal V_ix qui sort du convoyeur est utilisé pour régler l'offset du convoyeur et sera expliqué par la suite.

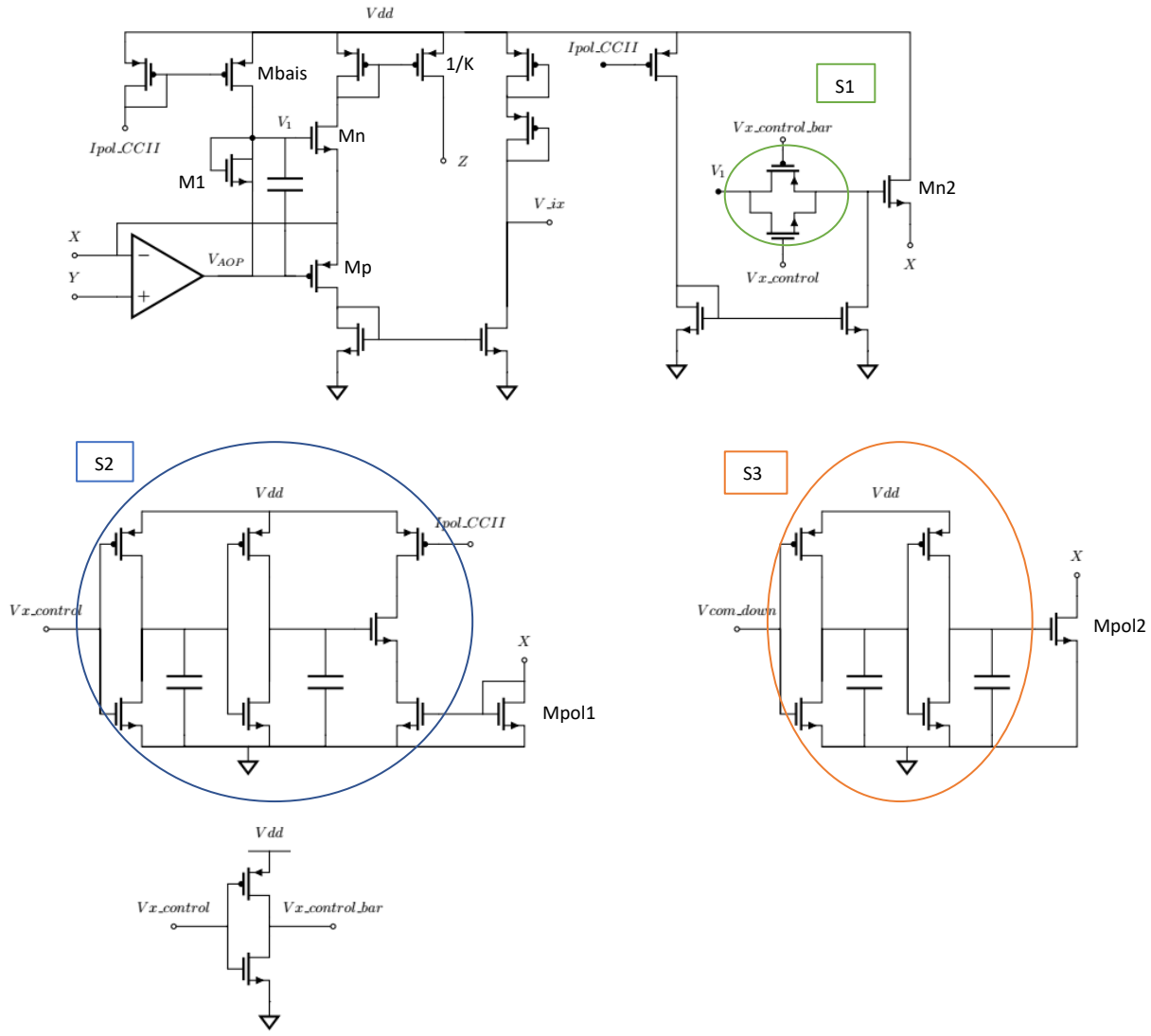


Figure 3.9 : Schéma complet interne du convoyeur de courant sans la division en courant.

Mesure

Les mesures de la recopie et de la division en courant faites sur la puce Orion sont présentées à la (figure 3.10) Il est montré l'évolution du rapport de division en courant en fonction du courant i_x et du facteur de division K. Nous aurions dû observer, si le bloc était parfaitement fonctionnel, des courbes horizontales pour K=25 (courbe verte), K=125 (orange) et K=600 (bleu) quelle que soit la valeur de i_x. Or le gain en division diminue fortement quand le courant i_x diminue. Ce problème peut être causé par le fait que les transistors utilisés sont sous le seuil pour les faibles courants i_x. Ce problème a été corrigé sur la puce Hermes ; les transistors ne sont plus sous le seuil. Le schéma du nouveau diviseur de courants est celui de la (figure 3.8).

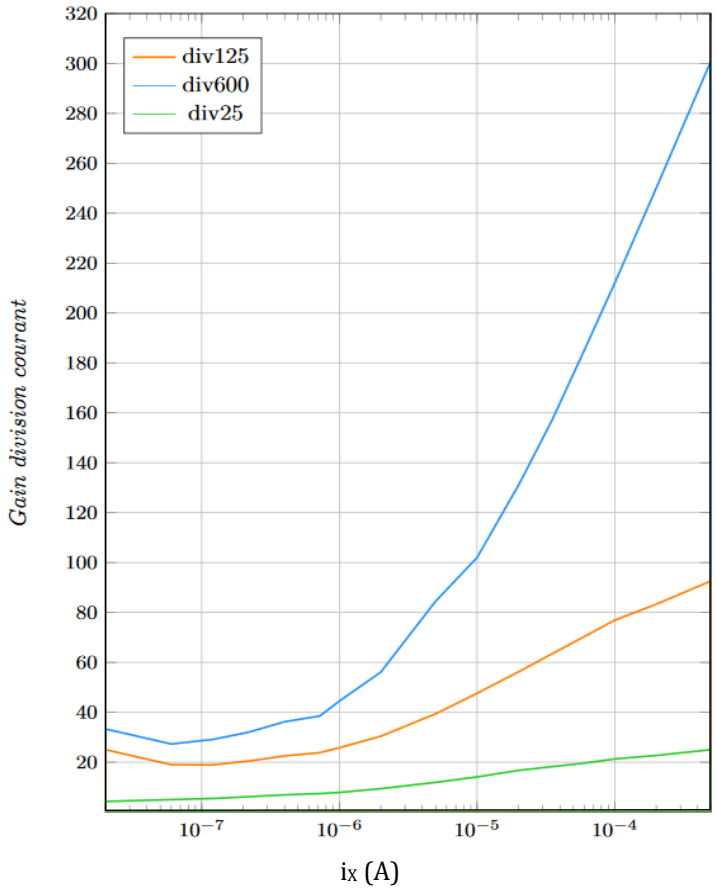
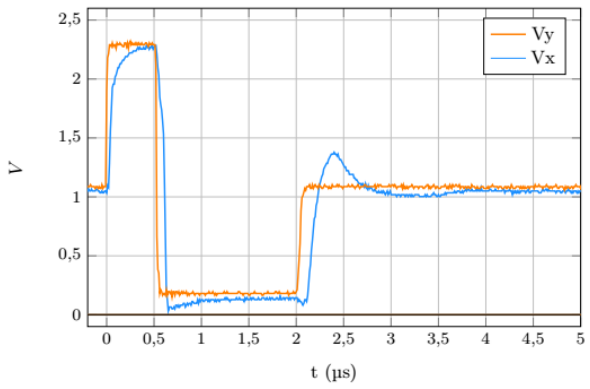


Figure 3.10 : Mesures de la division du courant en fonction du courant i_x

Sur la puce Orion nous avons également mesuré le pulse généré par le convoyeur de courant. Les résultats de mesure sont présentés à la (figure 3.11). En a) nous avons la mesure du pulse post-synaptique sous une charge de 1.2kΩ connectée en Vmid et en b) la mesure du même signal sans charge résistive. Cependant même sans charge résistive appliquée il y a toutefois la résistance de l'oscilloscope de 1MΩ conecté à la masse et les capacités parasites en parallèles qui ont été mesurées à 45pF. Pour la mesure du pulse sans charge, nous obtenons un slew-rate de 18V/µs pour la montée et un slew-rate de 16V/µs pour la descente. À la fin du pulse, le rebond observé est le résultat de notre choix de topologie du convoyeur de courant sans courant de polarisation (voir figure 3.6 b). En n'ayant aucun courant de polarisation dans les branches X et Z du convoyeur de courant, la tension V_x retrouve l'équilibre beaucoup plus lentement. Ce rebond n'a aucun impact sur l'apprentissage, car ce neurone qui vient de déclencher sera inhibé juste après.

a)



b)

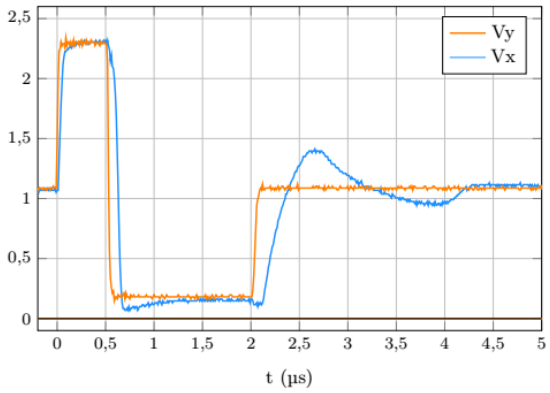


Figure 3.11: Mesures du pulse post du convoyeur de courant. a) Mesure avec une charge de 1.2kΩ connectée à Vmid et une capacité C=45pF. b) Mesure sans charge résistive, mais avec une résistance de 1MΩ de l'oscilloscope et une capacité C=45pF.

3.3 Correcteur offset

Le problème d'offset expliqué précédemment doit être résolu pour que le réseau de neurones puisse réussir un apprentissage. L'offset sera corrigé sur chacun des 100 neurones post-synaptiques.

La correction d'offset permet d'ajuster la tension en X pour qu'il n'y ait pas de courant qui charge la capacité de membrane quand il n'y a aucun évènement sur les neurones d'entrée. Nous ne corrigeons pas l'offset par rapport à la tension Vy imposée au convoyeur de courant, mais par rapport à la tension Vmid présente en X et légèrement altérée par le réseau des neurones pré-synaptiques. Une correction parfaite de l'offset étant impossible, il y aura toujours un petit courant qui passera par la borne X de notre convoyeur de courant. Comme le convoyeur de courant ne copie que le courant sortant de X pour charger la capacité de membrane, en imposant un très faible courant entrant en X il n'y aura aucun problème sur la variation de tension de capacité de membrane. Nous choisissons donc de corriger l'offset en plaçant la tension Vx du convoyeur très légèrement inférieur à la tension Vmid générée par les neurones pré-synaptiques, ce qui entraine un faible courant entrant dans le convoyeur qui ne sera pas copié vers la membrane.

Principe retenu

Dans la littérature pour corriger l'offset d'un AOP, nous trouvons de nombreuses publications et brevets sur la commutation pour réduire le bruit en 1/f. L'offset peut être considéré comme un bruit à f=0Hz [WU P.C 2013], [PIERANGELO C. 1985]. L'utilisation de commutation est totalement inadaptée dans notre cas, car elle génèrerait du bruit haute fréquence causé l'horloge et entrainerait des courants entrants et sortants du convoyeur de courant ce qui par conséquent chargerait la capacité de membrane et pourrait déclencher un neurone post-synaptique de manière intempestive.

La solution retenue pour réaliser la correction de l'offset est de modifier symétriquement la tension appliquée sur la borne substrat des deux transistors PMOS qui constituent la paire différentielle de l'AOP

d'entrée du convoyeur de courant [MOHAN, C 2017] (voir figure 3.12). Une fois ces tensions configurées, elles demeureront constantes durant la phase d'apprentissage du réseau de neurones. En jouant sur la transconductance de bulk nous modifions légèrement le courant des transistors d'entrée de la paire différentielle. Lorsque le courant augmente sur un transistor, il diminue sur l'autre et l'offset est corrigé quand les deux deviennent égaux.

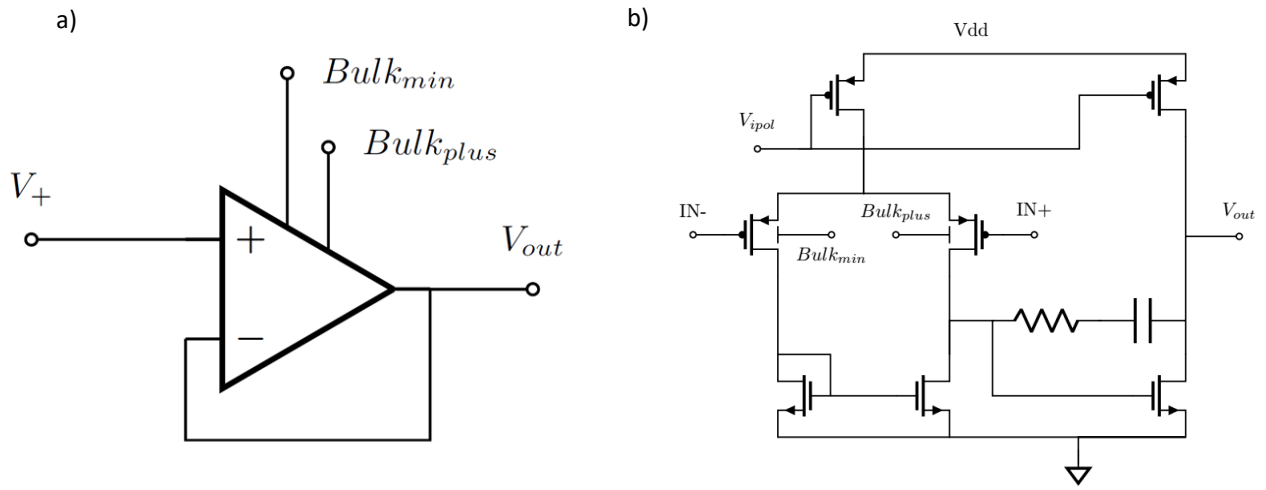


Figure 3.12 : AOP du convoyeur de courant a) schéma bloc, b) schéma électrique interne.

Circuit de commande de la correction d'offset

La (figure 3.13) décrit le système de commande de la correction d'offset. Cette correction s'effectue au moment de l'initialisation du système. Elle est pilotée par 4 signaux provenant du FPGA : CLK, rest, en_kill et en_counter.

L'objectif est de corriger la tension d'offset du convoyeur de courant en appliquant une tension de contrôle sur l'AOP. Cette tension de contrôle doit rester fixe une fois que l'offset est corrigé, nous n'avons donc pas de contrainte dynamique pour la conception, mais nous devons toutefois avoir une fonction qui mémorise la valeur de la correction.

Pour appliquer la tension de correction aux bornes des bulks des transistors d'entrés de la paire différentielle nous utilisons un convertisseur numérique analogique (DAC) qui génère une tension $Vbulk_{min}$ - $Vbulk_{plus}$; elle diminue progressivement et linéairement grâce à un compteur six bits [ARBET D. 2012].

Si l'offset du convoyeur de courant est trop important et que la tension en X est beaucoup plus basse que prévus, il n'y a pas de problèmes vis-à-vis de l'intégrité de l'ensemble du système, ce neurone déclenchera peu, voire pas du tout.

Si la tension en X est trop grande et reste supérieure à Vmid malgré la correction appliquée, l'intégrité de l'ensemble du réseau peut être corrompue dans ce cas. Ce neurone aura une charge constante de sa capacité de membrane et pourra déclencher de manière totalement anarchique et nuira à l'apprentissage du réseau de neurones. Il faudra donc « tuer » ce neurone post-synaptique pour qu'il ne charge plus du tout sa capacité de membrane et ne déclenche jamais. Pour cela après la correction nous envoyons un signal *en_kill*, qui permet d'activer la bascule RS si l'offset n'est pas corrigé. Le signal *kill_neurone* permet de décharger la capacité de membrane en permanence si ce signal est actif.

L'objectif est de corriger la tension d'offset du convoyeur de courant en appliquant une tension de contrôle sur l'AOP. Cette tension de contrôle doit rester fixe une fois que l'offset est corrigé, nous n'avons donc pas de contrainte dynamique pour la conception, mais nous devons toutefois avoir une fonction qui mémorise la valeur de la correction.

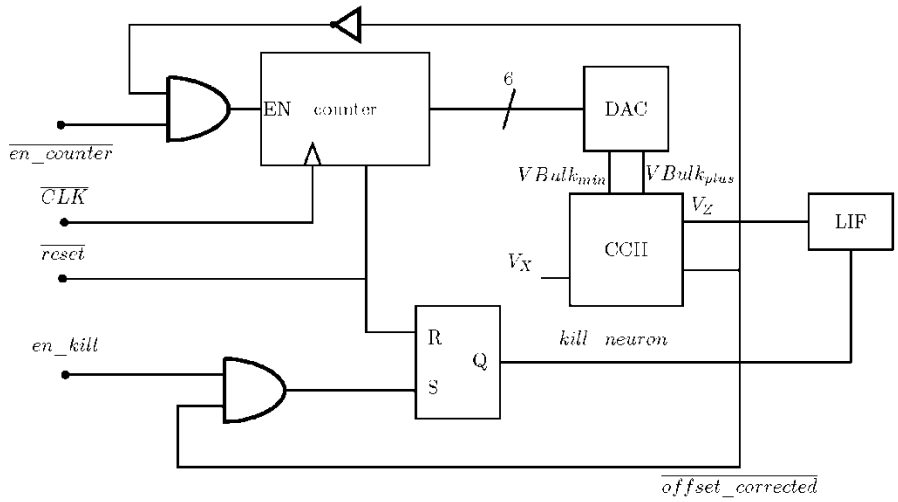


Figure 3.13 : Schéma du fonctionnement du correcteur d'offset pour le CCII.

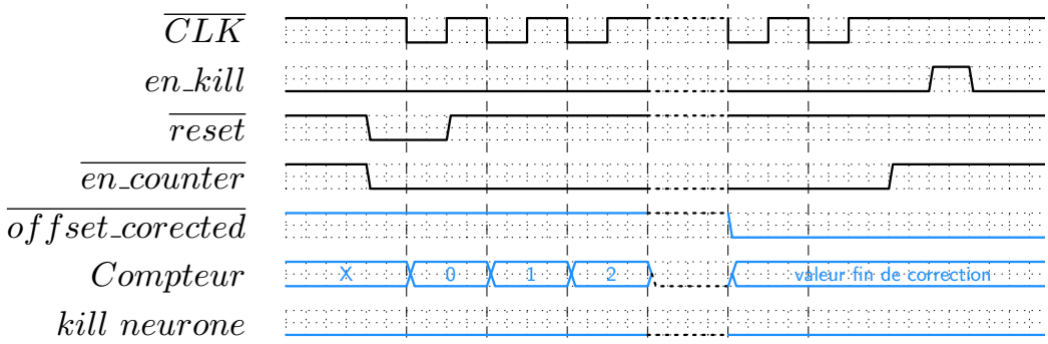


Figure 3.14 : Chronogrammes pour l'offset corrigé. En noir les signaux appliqués par le FPGA.

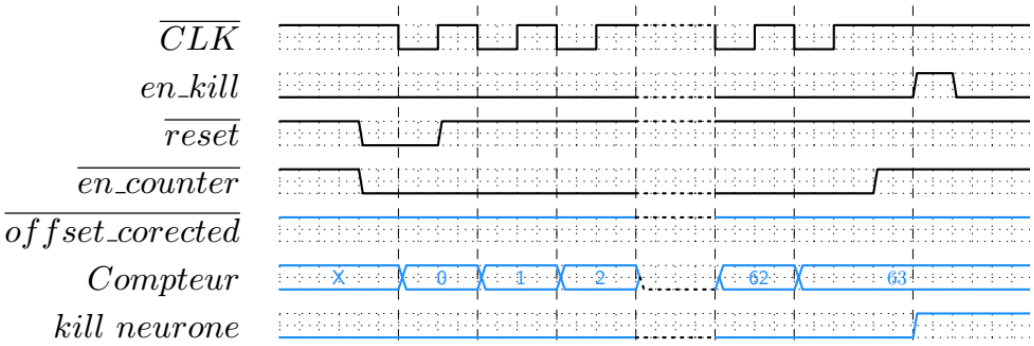


Figure 3.15 : Chronogrammes pour l'offset non corrigé. En en noir les signaux appliqués par le FPGA.

Convertisseur numérique analogique

Le DAC utilisé est un réseau R-2R qui permet de moduler le courant dans R1 et R2, les tensions ainsi créées en V_1 et V_2 (voir figure 3.16) sont connectées aux $bulk_{plus}$ et $bulk_{min}$ de l'AOP. La variation de tension de substrat est comprise entre 3,23 Volt et 3.3 Volt.

Le DAC 6 bits présenté en (figure 3.16) est de type R-2R. Il a l'avantage d'utiliser seulement deux valeurs de résistance différentes qui sont du même ordre de grandeur, ce qui est indispensable pour la fabrication sur silicium. Dans notre cas nous utiliserons deux transistors MOS différents jouant le rôle de résistance, avec l'un qui a une longueur de canal double par rapport à l'autre pour obtenir le double de la résistance, contrairement à l'article [ARBET D. 2012] qui propose d'utiliser 2 transistors en série pour doubler la résistance. En effet, la mise en série des transistors engendre plusieurs problèmes, la tension V_{GS} des deux transistors est différente. Il se peut aussi que dans certaines configurations, les transistors ne soient pas dans le même mode de fonctionnement, l'un en linéaire et l'autre en saturé, ce qui entraine une résistance différente pour les deux transistors. Pour remédier à ce problème, il était préférable d'utiliser un seul transistor avec un canal plus long. En contrepartie il faut éviter la longueur minimale des transistors pour qu'ils soient moins affectés par la variation de longueur quand le transistor est passant. La taille de ces transistors est (L=2µm ; W=2µm) pour l'un et (L=4µm ; W=2µm) pour le second.

Un autre problème qu'il faut également résoudre est la variation de la non-linéarité entre chaque quantum. L'un des problèmes vient de la tension de drain de M1 et M2 ne sont pas identiques si les

courants sont différents dans M1 et M2, ce qui arrive la plupart du temps. Pour remédier à ce problème, la solution est d'utiliser de larges transistors pour la copie en courant (M1 et M2), qui ont comme propriété d'avoir une très faible variation V_{DS} quand le courant varie.

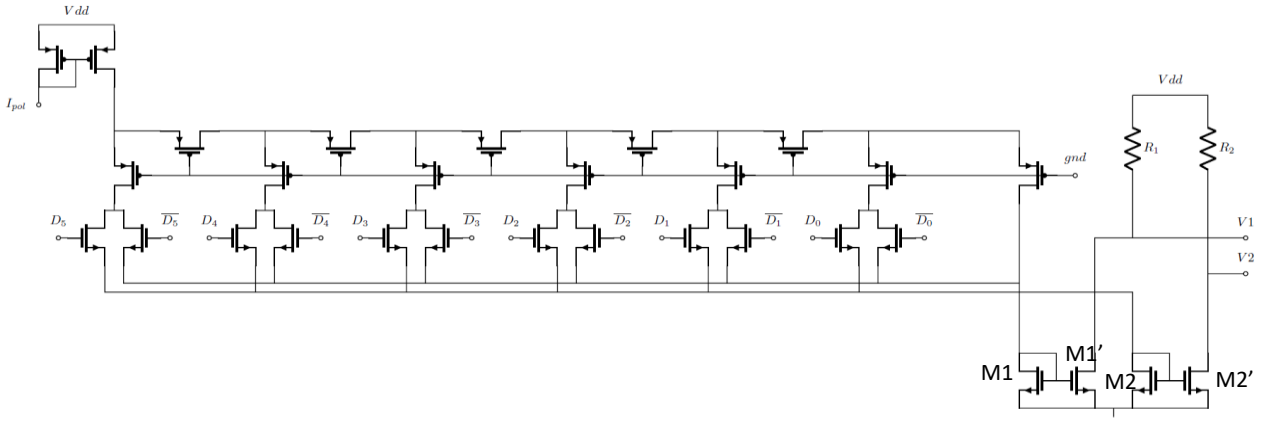


Figure 3.16 : Schéma du DAC utilisé pour la correction d'offset.

Pour faire varier la tension générée par le convertisseur numérique analogique et mémoriser cette valeur, nous utiliserons un compteur 6 bits à base de bascule D. Les bascules D sont réalisées à l'aide de transistors sur la base de la topologie (figure3.17). [KO U. 2000]. Cette configuration assure son bon fonctionnement sans qu'il y ait de difficultés de conception, mais au détriment de la surface prise sur le layout. La consommation dynamique et la rapidité ne sont pas nos priorités dans la mesure où il y peu de composants et qu'ils ne commutent plus durant la phase d'apprentissage du réseau de neurones.

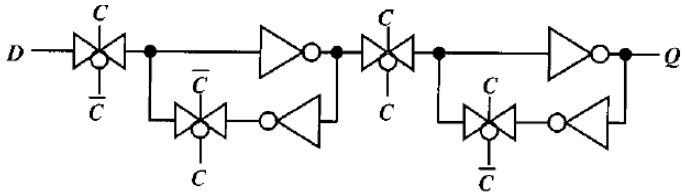


Figure 3.17 : Schéma simplifié de la bascule D où C représente l'horloge D l'entrée et Q la sortie.

Comment détecter la bonne correction de l'offset, visualisation avec simulations

Maintenant que nous avons le compteur qui contrôle le convertisseur numérique analogique et génère une tension appliquée sur les bulks des transistors d'entrée de la paire différentielle de l'AOP afin de corriger l'offset, il faut désormais détecter à partir de quelle tension l'offset est corrigé pour chacun des 100 neurones post-synaptiques. Il n'est pas envisageable de corriger « à la main » l'offset de chaque neurone. Il faut donc détecter la tension Vx et la comparer non pas à la tension Vmid de Vy, mais à la

tension Vmid altérée par le réseau de neurones pré-synaptiques et arrêter la correction dès que la tension Vx devient inférieure à la tension de référence. Le défi est de comparer une tension de l'ordre du millivolt en utilisant des comparateurs qui ont des offsets de plus ou moins 25mV.

À l'état initial, quand le compteur vaut 0, la tension Vx est maximale. À chaque coup d'horloge, le compteur s'incrémente et réduit la tension Vx. Pour détecter le moment où l'offset est corrigé, nous regardons le sens dans lequel le courant circule dans la branche X. Quand la tension Vx est supérieure à Vmid le courant du convoyeur de courant provient de VDD, passe par Mn avant de sortir vers le réseau de memristors (voir 3.18). Quand la tension Vx devient inférieure à Vmid, le courant vient du réseau de memristors passe par Mp et M2 avant d'aller à gnd. Le but est donc de détecter le moment ou le courant change de sens. Le courant entrant dans le convoyeur de courant est copié de M2 vers M3 et traverse M4 et M5, la tension V_i_X décroit et est reportée sur l'entrée non inverseuse du comparateur. Dès que cette tension devient inférieure à Vth (2,2V), enable passe à 0, ce qui stoppe le compteur et le DAC maintient la tension de contrôle. La tension en X du convoyeur de courant est alors corrigée. Une simulation montrant ce fonctionnement est montrée (Figure 3.19) avec 10kΩ comme résistance équivalente des 784 memristors.

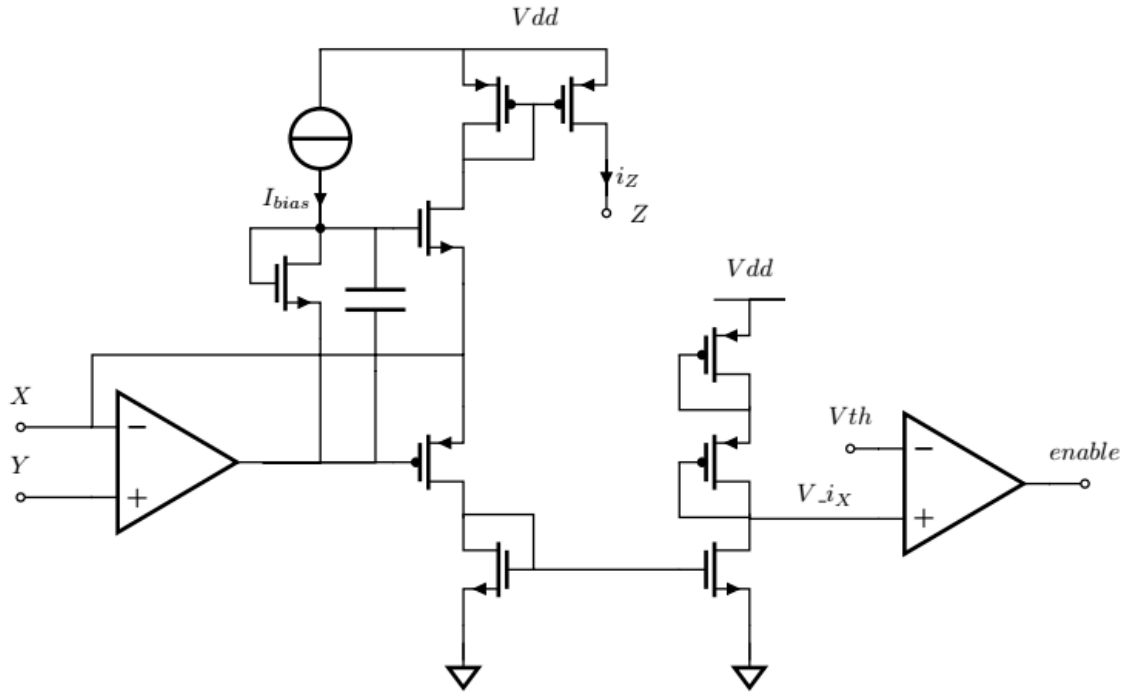


Figure 3.18 : CCII avec détecteur d'offset corrigé.

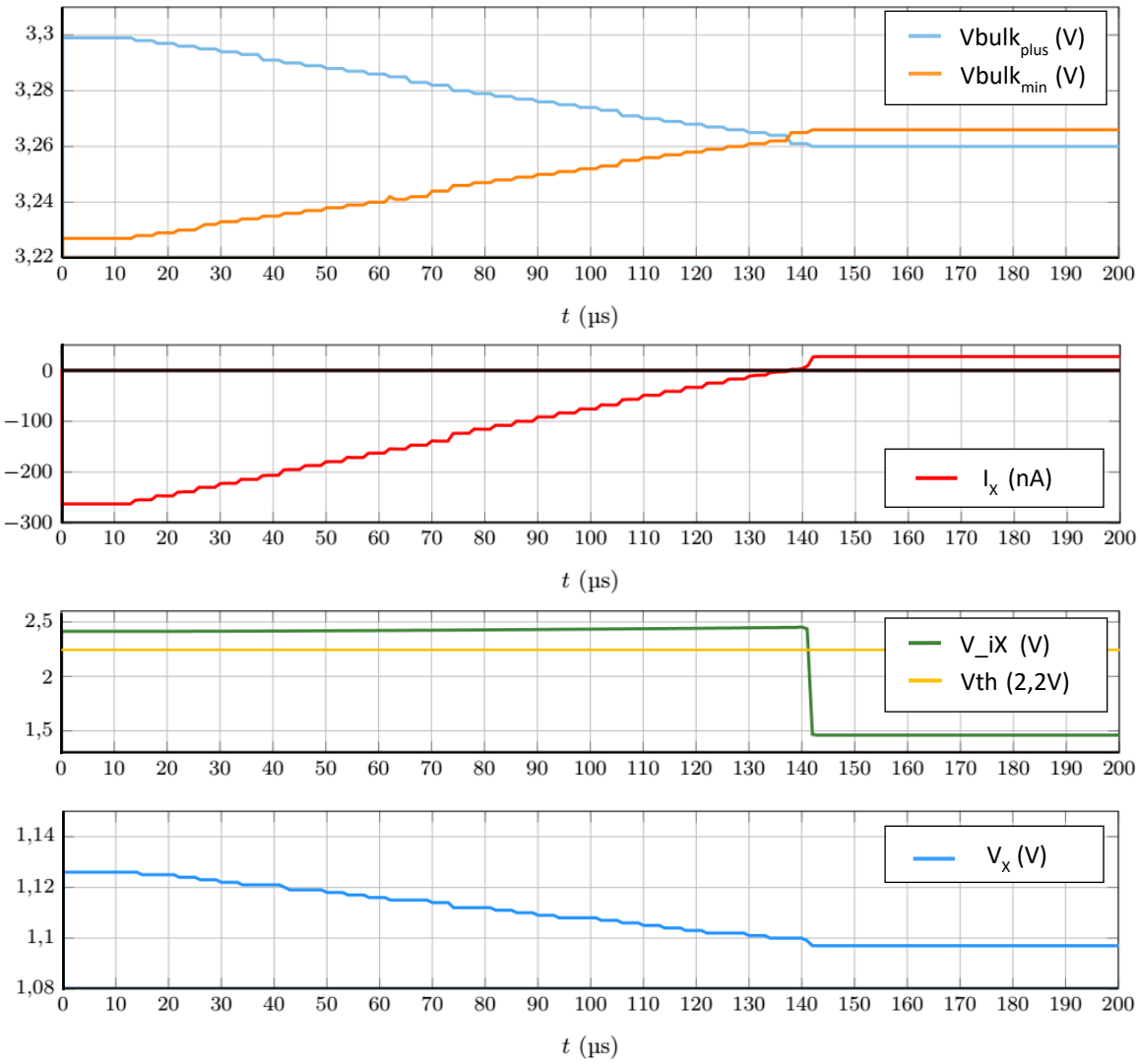


Figure 3.19 : Simulation de la correction d'offset montrant l'évolution des tensions de bulk, du courant Ix, de la tension V_iX permettant de détecter le changement de sens du courant et la tension VX.

Mesure de la correction d'offset

Les mesures ont été faites suivant le montage la (figure3.20). avec Vy = Vmid et Vx1 la tension appliquée au réseau d'une valeur précise de 1,124V avec Rx1 = 10kΩ. Vx est la tension mesurée par un oscilloscope qui a une résistance de 1MΩ reliée à 0V. La tension en Vx lorsque que convoyeur de courant ne génère aucun courant est la suivante $Vx = \frac{Vx1 * R_{oscilloscope}}{R_{x1} + R_{oscillosope}}$, soit Vx = 1,113V. Au démarrage de la correction la tension Vx est à son maximum, puis diminue progressivement jusqu'à trouver le point où la tension générée par le CCII devient inférieure à 1,113V. En dessous de cette valeur, le courant n'est plus copié de X vers Z et la tension Vz devient nulle.

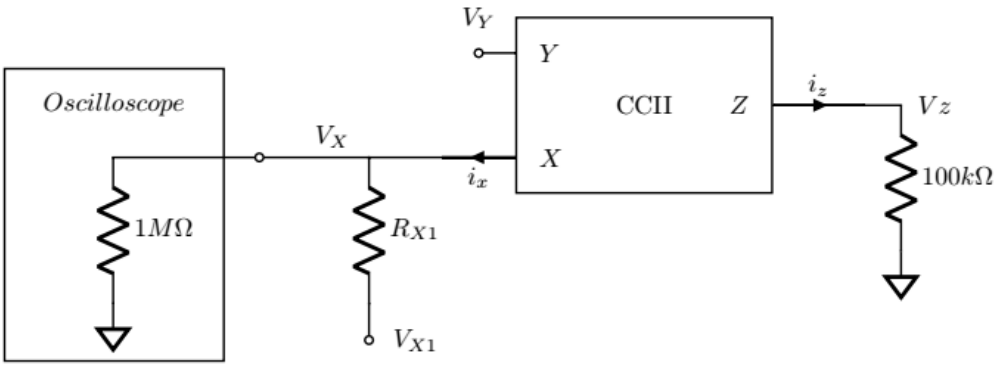


Figure 3.20 : Schéma de la mesure de la correction d'offset du CCII.

Sur la (figure 3.21), les créneaux en gris représentent l'horloge (clk) qui contrôle le compteur et donc la tension en sortie du DAC. Le signal offset corrected s'annule quand l'offset est corrigé. V_Z est l'image du courant i_Z en mesurant la tension. Nous pouvons voir que le courant ainsi que la tension décroissent à chaque front d'horloge jusqu'à ce que la valeur du courant devienne nulle. La tension Vx après correction est bel et bien en dessous de 1,113V comme calculé précédemment. La correction de l'offset nous permet de maitriser la tension en sortie des neurones post-synaptiques, ce qui est indispensable pour l'apprentissage du réseau de neurones.

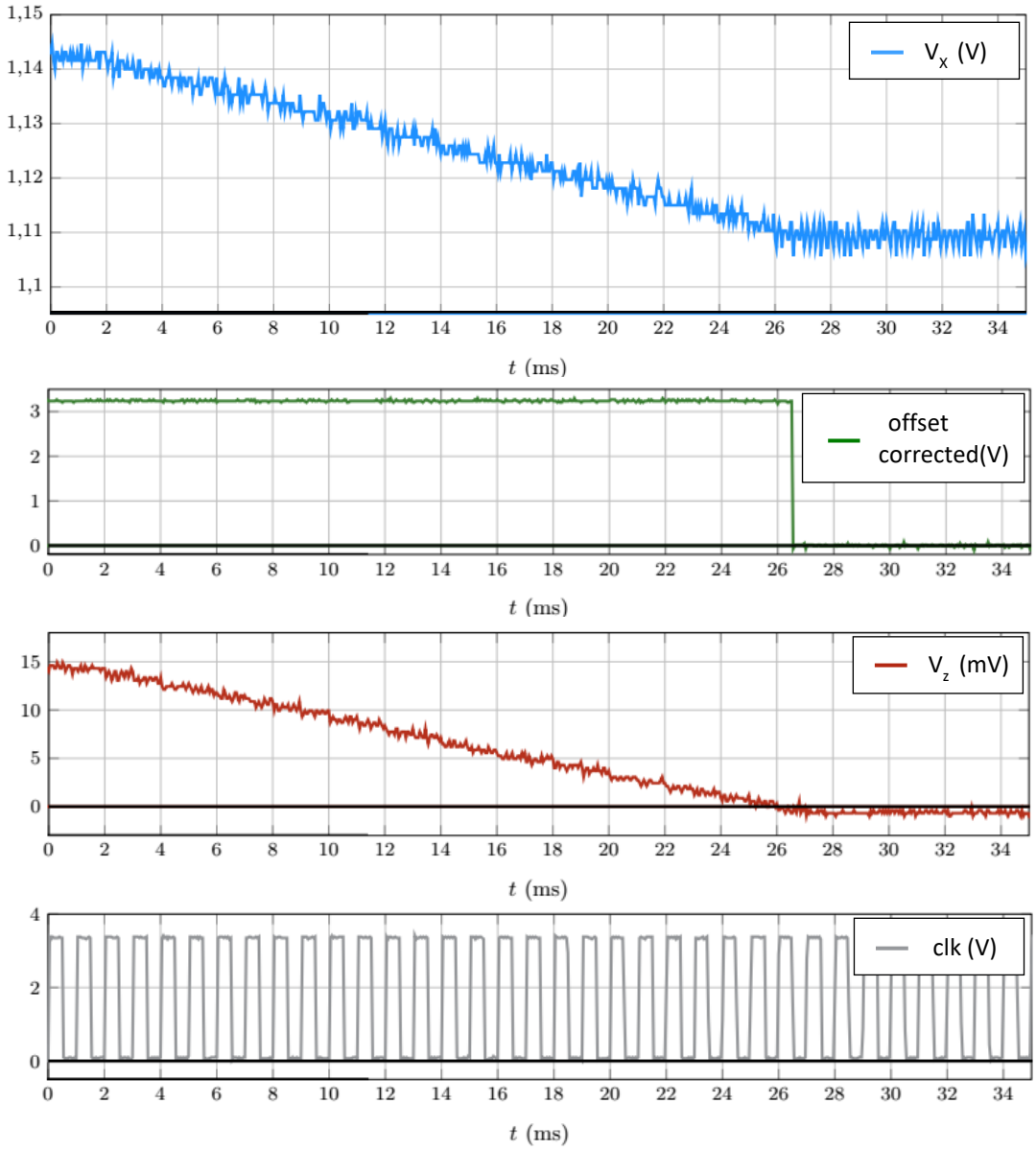


Figure 3.21 : Mesure de la correction d'offset du CCII

3.4 Neurone pré-synaptique

Dans une optique de faible consommation, le but est de limiter au maximum le courant dans les memristors, mais surtout le courant de polarisation continu. Cependant il faut que le neurone pré-synaptique fournisse un courant suffisamment important quand les 100 memristors sont à leur résistance minimale. Quand le neurone pré-synaptique a été conçu et finalisé, la valeur minimale des memristors était de 100kΩ, ce qui impliquait une résistance équivalente de 1kΩ connectée à 1,1V et donc un courant de 1mA pour le pulse « down » et 1,1mA pour le pulse up. Depuis la valeur des résistances Ron des memristors sont passées à 1MΩ, cependant le neurone post-synaptique étant déjà conçu et opérationnel, nous avons décidé de ne pas le modifier.

Le neurone pré-synaptique a pour but de fournir la tension Vmid au repos, le pulse « up » et « down » lorsque nous envoyons une impulsion respectivement sur trig_up et trig_down. Il ne peut pas être déclenché si un pulse est déjà en cours. Pour contrôler les paramètres temporels des pulses, nous utilisons une rampe de tension qui est créée en chargeant une capacité avec un courant constant (voir (figure 3.22 a). Pour déclencher la rampe, nous mettons à zéro V_{rampe} par un pulse du signal V_{rest}. La tension évolue ensuite linéairement, pour obtenir la durée souhaitée, nous plaçons un comparateur avec un seuil Vth. Le pulse commence quand la tension V_{rampe} devient inférieure à Vth et s'arrête quand elle redevient supérieure à ce même seuil (voir figure 3.22 b). Le temp t_{pulse} est défini par l'équation suivante $t = \frac{V_{th} * C_{rampe}}{i_C}$, avec i_c le courant chargeant le condensateur qui est différant de I_{pol}. Le temps des deux pulse « up » et « down » sont généré avec le même courant de polarisation I_{pol} =1µA. Le premier pour la durée du pulse « up » avec V_{th} = 1,7V, C_{rampe} = 420fF, et i_c = 2µA pour un temps calculé de t_{pulse} = 360ns, avec les retards, les temps de propagation et la largeur de V_{reset} on obtient un temps de 500ns pour le pulse « up ». Le second pour la durée du pulse « down » on a V_{th} = 2,2V, C_{rampe} = 810fF, et i_c = 0,175µA pour un temps t_{pulse} = 10 µs.

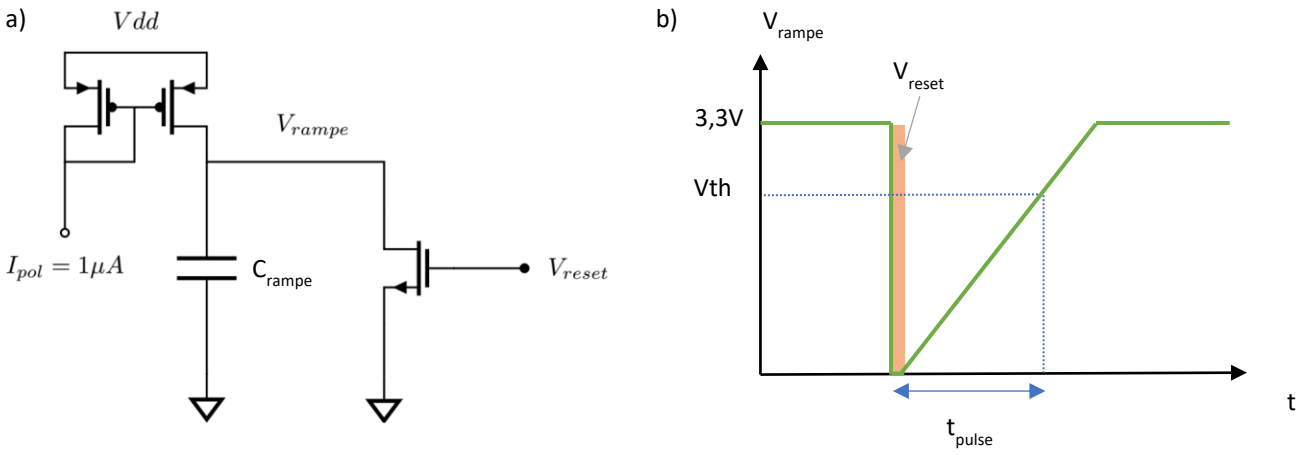


Figure 3.22 : Rampe de tension a) Schéma électrique, b) Chronogramme.

Quand un évènement arrive sur trig_down ou trig_up (voir figure 3.23), nous envoyons respectivement une brève impulsion sur V_{rest} du générateur de rampe de tension « down » ou « up » (respectivement t_down_rampe et t_up_rampe) pour déclencher et contrôler le temps du pulse correspondant. La logique de commande de tension reçoit les informations des rampes de tension et envoie la commande de tension V_{up} V_{mid} ou V_{down} sur l'entrée non inverseuse de l'AOP afin qu'il puisse délivrer la tension désirée. Ce bloc génère également enable_trig, ce signal permet de valider le déclenchement d'une rampe de tension si aucune n'est active.

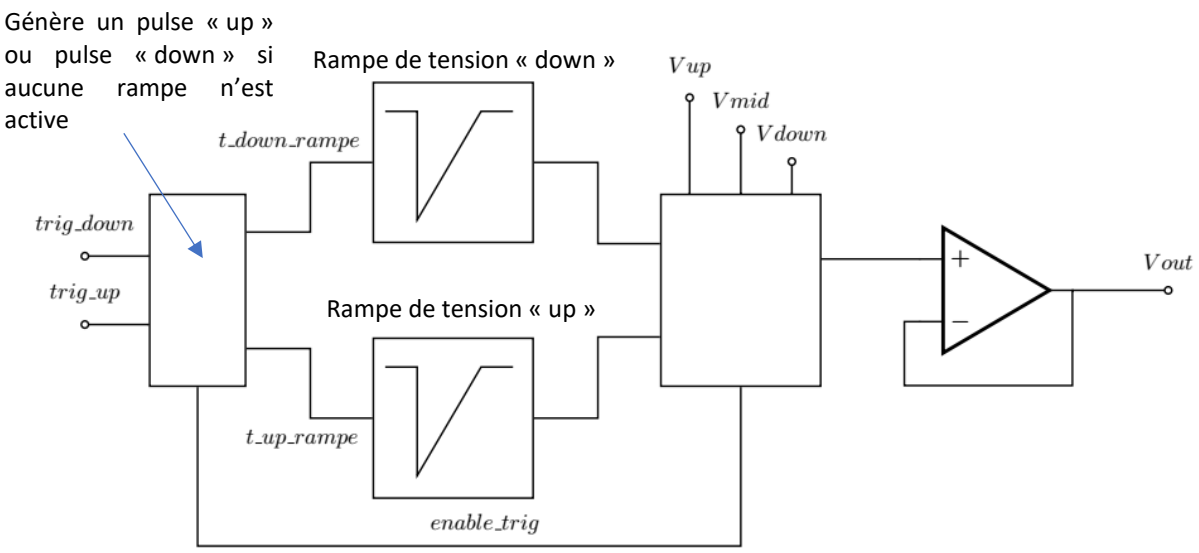


Figure 3.23 : Neurone pré-synaptique

Pour limiter la consommation du système, il est important de limiter le courant au repos de l'AOP. Pour cela nous utiliserons un AOP de classe AB. L'AOP cascodé utilisé (voir figure 3.24) est inspiré de [HUIJSING J. 2017]. Quelques modifications ont dû toutefois être réalisées. Les transistors M8 et M9 ne sont pas en charge cascodée, ce qui permet d'avoir une plus grande amplitude de tension sur la tension de grille de Mp et Mn, mais c'est au détriment du gain de l'AOP en réduisant la résistance du deuxième étage. Le transistor M16 permet d'obtenir une tension Vb entre la tension de grille de Mp et Mn dans le but de réduire la consommation au repos de l'AOP. Il y a un juste milieu à trouver entre une faible consommation statique et la stabilité de l'AOP. La solution adoptée est de préférer un AOP avec une plus grande consommation statique et donc d'améliorer la marge de phase pour ainsi éviter les rebonds appliqués sur le réseau de memristors. Les transistors M14 et M15 permettent de diminuer la tension Vb lorsque la demande de courant se fait importante en sortie de l'AOP (voir figure 3.25). Nous appliquons V_{polmid} sur la grille de ces deux transistors qui permet de rendre ces deux transistors conducteurs quand la tension de grille de Mn diminue et celle Mp augmente. Le condensateur Cc et Rz quant à eux permettent d'augmenter la marge de phase de l'AOP. La tension V_{polmid} valant 1.6V est une tension de polarisation qui est appliquée sur les transistors M10 et M11 pour la résistance cascodée, ainsi que pour M14 et M15 pour la variation dynamique de Vb.

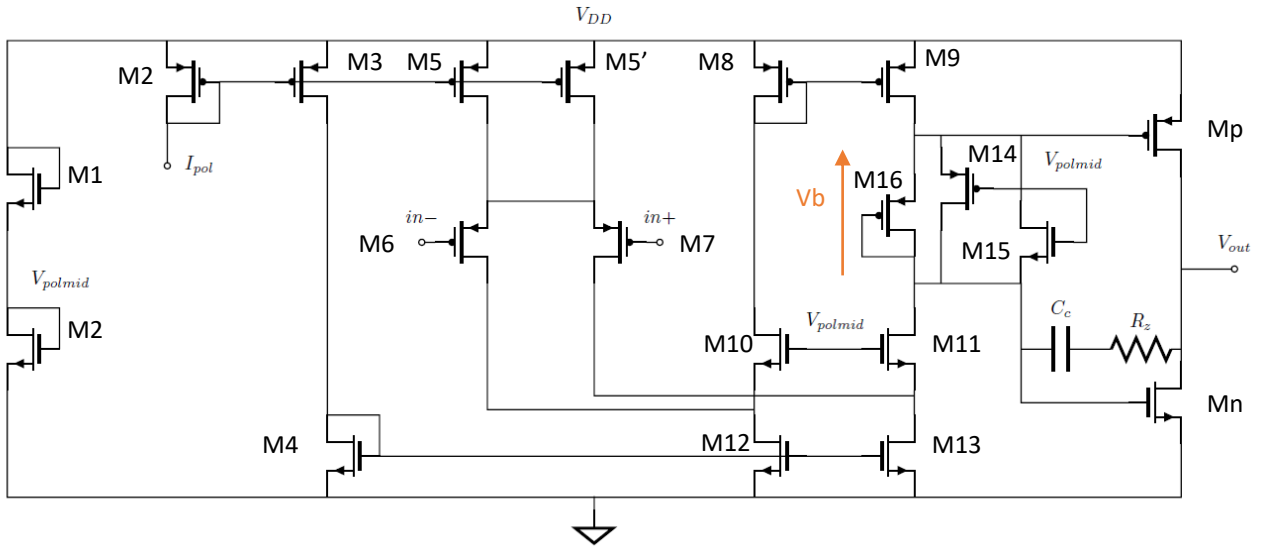


Figure 3.24 : AOP du neurone pré-synaptique.

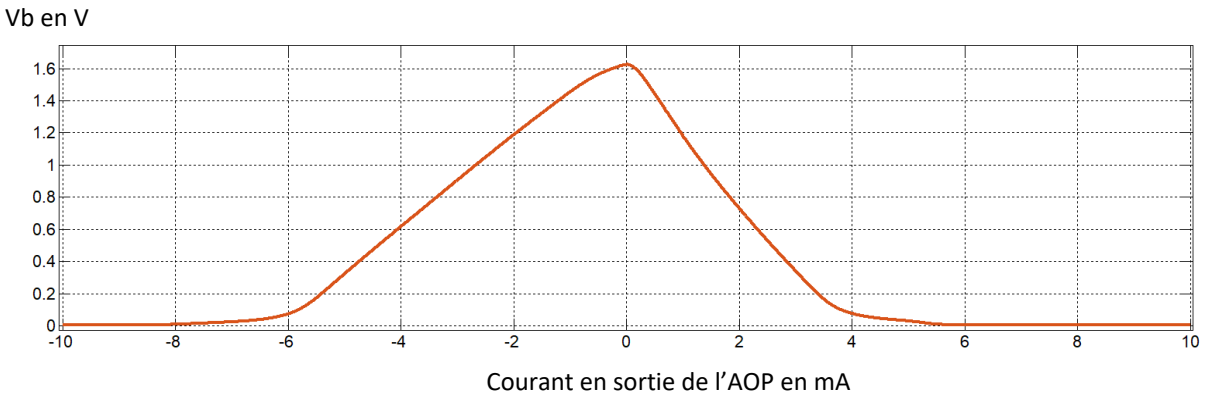


Figure 3.25 : Tension Vb en fonction du courant en sortie de l'AOP.

Dans les valeurs extrêmes, la tension de grille de Mn peut aller atteindre 3,27V, et celle de Mp peut descendre à 53mV. L'AOP peut alors délivrer jusqu'à 4mA, pour un courant total au repos total 146,2µA.

Mesures

Pour être le plus proche possible de la réalité physique, nous prenons en compte les impédances parasites causées par le montage et les mesures, qui ne sont pas négligeables pour une puce conçue pour de l'électronique intégrée. La connaissance de ces impédances parasites permet de prédire sur Cadence l'évolution de V_{out}. Les capacités parasites en ajoutant celle de l'oscilloscope sont de l'ordre de 45pF. L'oscilloscope a également une résistance de 1MΩ reliée à la masse. Pour le dimensionnement de l'électronique, il est nécessaire de connaitre la valeur de la capacité du crossbar. Cette valeur étant inconnue, mais étant donné que le crossbar est intégré sur la puce, la valeur de cette capacité doit être

relativement faible, nous avons supposé qu'elle serait inférieure à 50pF. Le dimensionnement de l'électronique est donc fait en prenant 50pF comme valeur maximale de la capacité de sortie. Les mesures du neurone pré-synaptique se sont faites en mesurant V_{out} (voir figure 3.26) en prenant en compte des résistances et capacités parasites une résistance de charge R_L connectée à Vmid valant (1,2kΩ ou ∞).

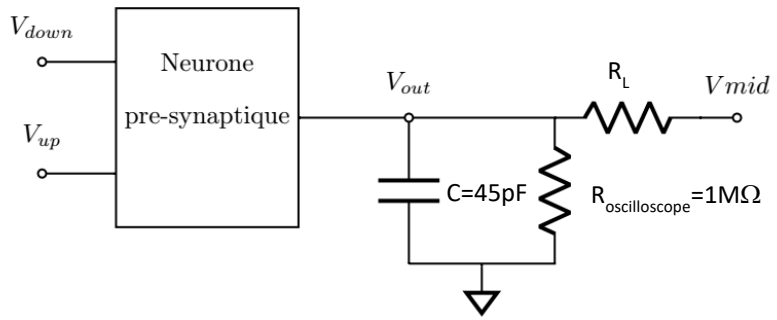


Figure 3.26 : Schéma de mesure du neurone pré-synaptique.

Pour la génération du pulse « up » ou « down », nous appliquons un évènement respectivement sur V_{up} ou V_{down} (voir figure 3.26). Si nous envoyons une impulsion sur V_{up} pour déclencher le pulse « up », il faut un certain temps avant que la tension V_{out} commence à varier, environ 70ns, puis 50 ns sont nécessaires pour atteindre la valeur de 2V (voir figure 3.27) Pour les pulses « down » il faut également 70ns pour attendre une variation de la sortie de V_{out} et 50ns supplémentaires pour atteindre la valeur de 0,1V (voir figure 3.28). L'attente de 70ns est causée par le délai du trig_down et trig_up, le déclenchement de la rampe de tension et le temps de propagation des comparateurs avant de sélectionner la tension désirée par la logique de commande de tension.

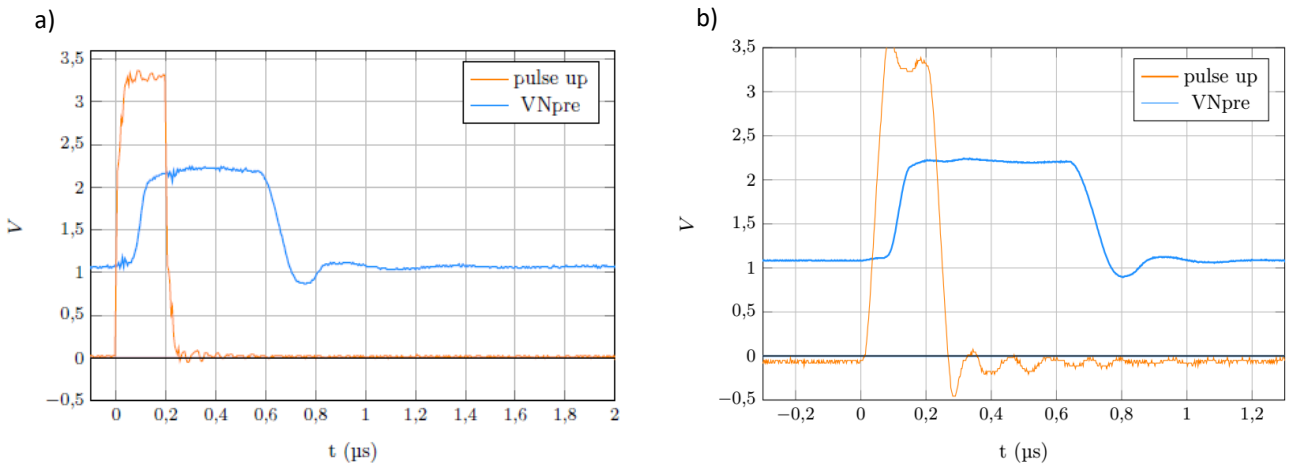


Figure 3.27 : Mesure du pulse « up » du neurone pré-synaptique. a) charge $R_{oscilloscope}$=1MΩ, C=45pF et R_L = ∞ b) charge $R_{oscilloscope}$=1MΩ C=45pF et R_L = 1,2kΩ

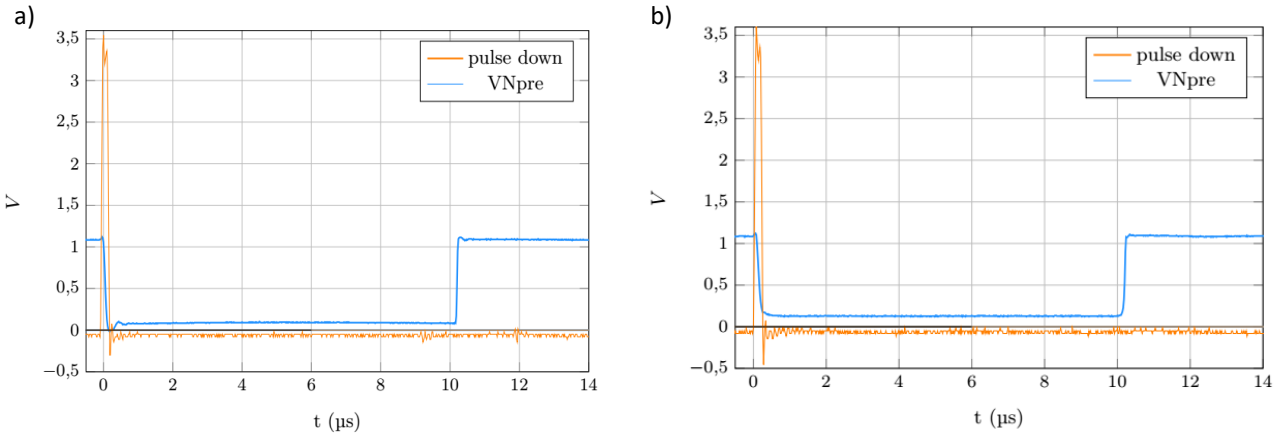


Figure 3.28 : Mesure du pulse « down » du neurone pré-synaptique a) charge $R_{oscilloscope}$ = 1MΩ, C = 45pF et R_L = ∞ b) charge $R_{oscilloscope}$ = 1MΩ, C = 45pF et R_L = 1,2kΩ.

3.5 Conclusion

Cette partie nous a permis de comprendre les problèmes liés au réseau de neurones évènementiel à base de matrices de memristors de grandes tailles et de proposer des solutions pour les résoudre. La correction de l'offset du convoyeur de courant est un point essentiel qu'il a fallu résoudre afin de pouvoir réaliser un apprentissage correct. Les signaux pré et post-synaptiques sont bien générés et correspondent à nos attentes. Le seul point sensible est la division en courant qui ne correspond pas à nos attentes et sera corrigé dans la seconde puce, appelée Hermes, qui contiendra l'ensemble du réseau de neurones.

Dans le dernier chapitre de ce manuscrit, nous allons présenter l'architecture de la puce Hermes, voir comment le réseau de neurones est orchestré par le circuit numérique, et quelle stratégie d'apprentissage nous avons choisie. Seront également présentées les simulations d'apprentissage proche de l'électronique du réseau neurone afin de montrer son potentiel et de choisir la plus adapté.

CHAPITRE 4 : PUCE SYSTEME, HERMES

Dans ce dernier chapitre, nous allons voir comment est construite la puce et par quel moyen le réseau de neurones ainsi constitué est piloté. Nous allons dans un premier temps présenter l'ensemble de la puce, évoquer les contraintes ainsi que les problèmes matériels éventuels liés au crossbar de memristors, à la caméra et à leur intégration sur la puce. Les stratégies d'apprentissages sont des éléments majeurs pour améliorer le fonctionnement du réseau, nous allons par la suite voir comment elles sont implémentées et déterminer par simulation les plus efficaces à implémenter.

4.1 Présentation de Hermes

La puce est constituée de trois dispositifs, la caméra évènementielle conçue chez Prophesse, le crossbar de memristors conçu et fabriqué par l'université de Twente, puis implémenté par flip-chip par IBM-Zurich et enfin le réseau de neurones conçut à l'université de Bordeaux. La caméra a déjà été conçue et a donc déterminé le choix de la technologie utilisé, on utilisera pour la conception de la puce la technologie UMC en 180nm et avec une dernière couche de métallisation en 20Å d'épaisseur. Cette puce est fabriquée selon un engineering run et nous permet d'avoir à disposition un wafer entier, nous avons fait produire en tout 6 wafers contenant chacun 144 puces, 6 de plus restent en attente de fabrication.

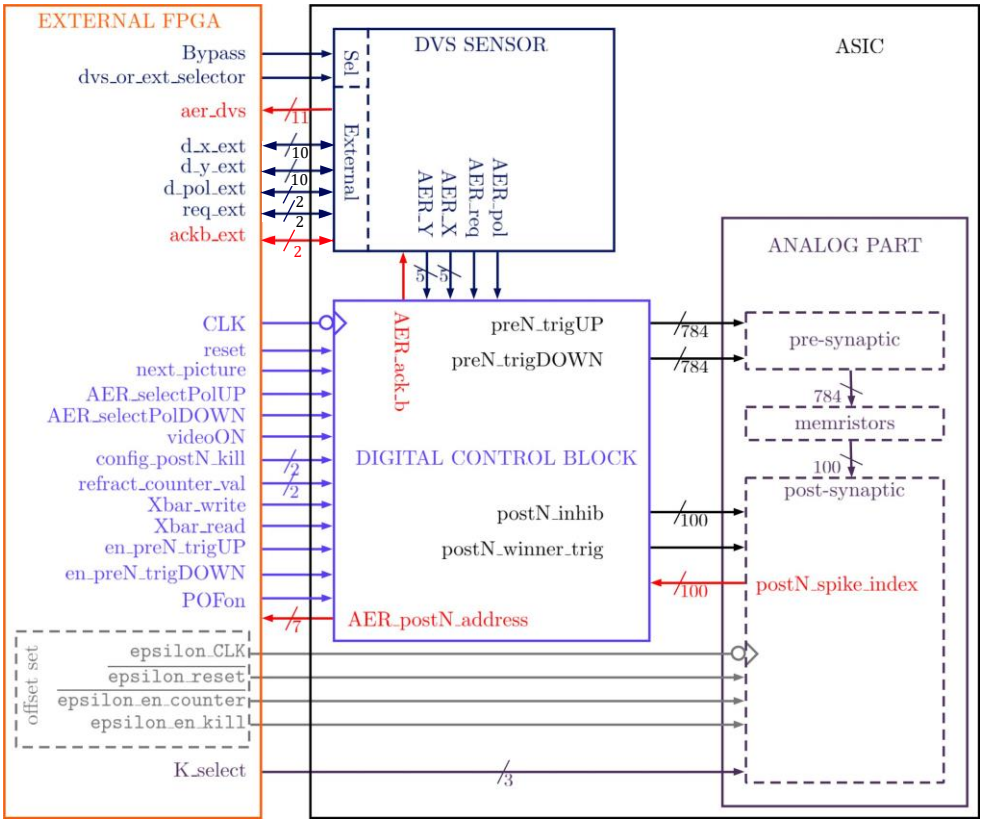


Figure 4.1 : Schéma global de la puce et du FPGA qui la contrôle.

Notre réseau de neurones analogique doit interpréter les évènements numériques en sortie de la caméra évènementielle pour le déclenchement du neurone concerné, pour cela nous utilisons un contrôleur

numérique appelé DCB (Digital Control Block). Le DCB est la partie centrale de la puce permettant la communication vers le réseau de neurones depuis la caméra et une communication depuis et vers l'extérieur de la puce, il permet aussi de contrôler l'apprentissage du réseau de neurones. Le schéma global de la puce est présenté (figure 4.1). La communication vers le DCB se fait avec un FPGA.

Toutes les entrées sorties numériques qui sortent de la puce pour aller vers le PGFA sont en 3,3V, toutes celles qui entrent dans de la puce sont quant à elles en 1,8V. Un translateur de niveaux est placé à l'interface de la puce permettant également si nécessaire de fournir un courant important via un buffer. La caméra est alimentée en 1,8V tandis que les blocs analogiques et le DCB sont alimentés en 3,3V. Il y a également des translateurs de niveaux dans la puce. Leur architecture est présentée (figure 4.2). Ils permettent de faire le lien entre le contrôleur numérique et la caméra mais sont aussi alloués pour la communication vers l'extérieur de la puce depuis le DCB. Le choix d'utiliser un contrôleur numérique en 3,3V est dû au nombre important de connexions qui doivent se faire vers le réseau de neurones, plus de 1600 connexions qui auraient nécessité un translateur de niveau sur chacune d'entre elles. Les signaux numériques sortants sont connectés à un FPGA, qui est alimenté en 1,8V. Il faudra sur la carte de test qui permet d'accueillir la puce Hermes et le FPGA mettre des translateurs de niveaux de 3,3V vers 1,8V pour tous les signaux numériques.

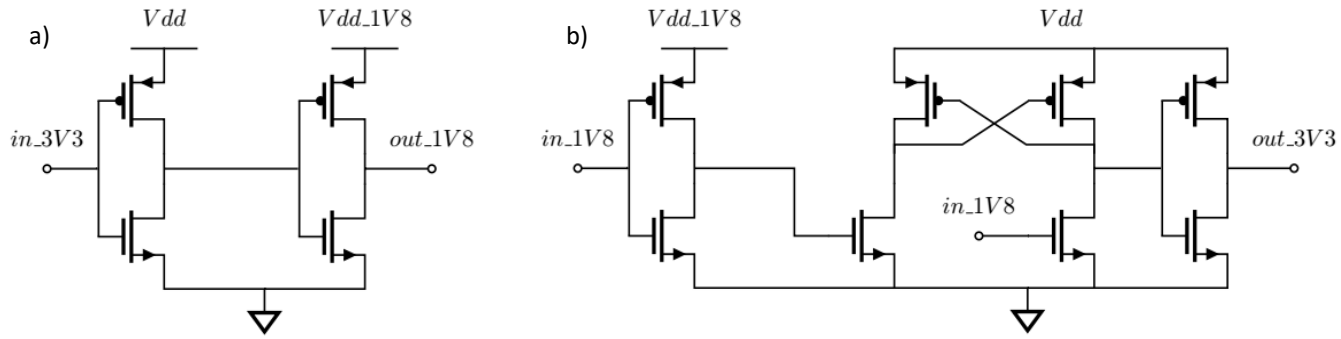


Figure 4.2 : a) translateur de niveaux de 3,3V vers 1,8V. b) translateur de niveaux de 1,8V vers 3,3V.

Layout de la puce

Le layout de la puce avec notre réseau de neurones, la caméra et les connexions vers le crossbar est montré (figure 4.3). Sa taille est de 160mm^2 (12.8mm x 12.6mm). Notre réseau de neurones est composé d'une matrice de 28x28 neurones pré-synaptiques, du DCB et de la matrice de 10x10 neurones post-synaptiques. La caméra est placée à gauche de la puce, suffisamment éloignée du crossbar pour éviter que l'ombre de ce dernier perturbe l'optique. Nous avons suffisamment de place pour disposer nos 133 plots d'entrés sorties qui sont répartis de la manière suivante : les 34 plots du haut sont reliés aux neurones pré et post-synaptiques, les 63 de gauches sont connectés à la caméra et les 36 du bas sont eux

connectés au DCB. Le réticule fabrication contient deux exemplaires de la puce comme présenté en (figure 4.4). Une photo de la puce Hermès a été prise dans son boitier est montré (figure 4.5).

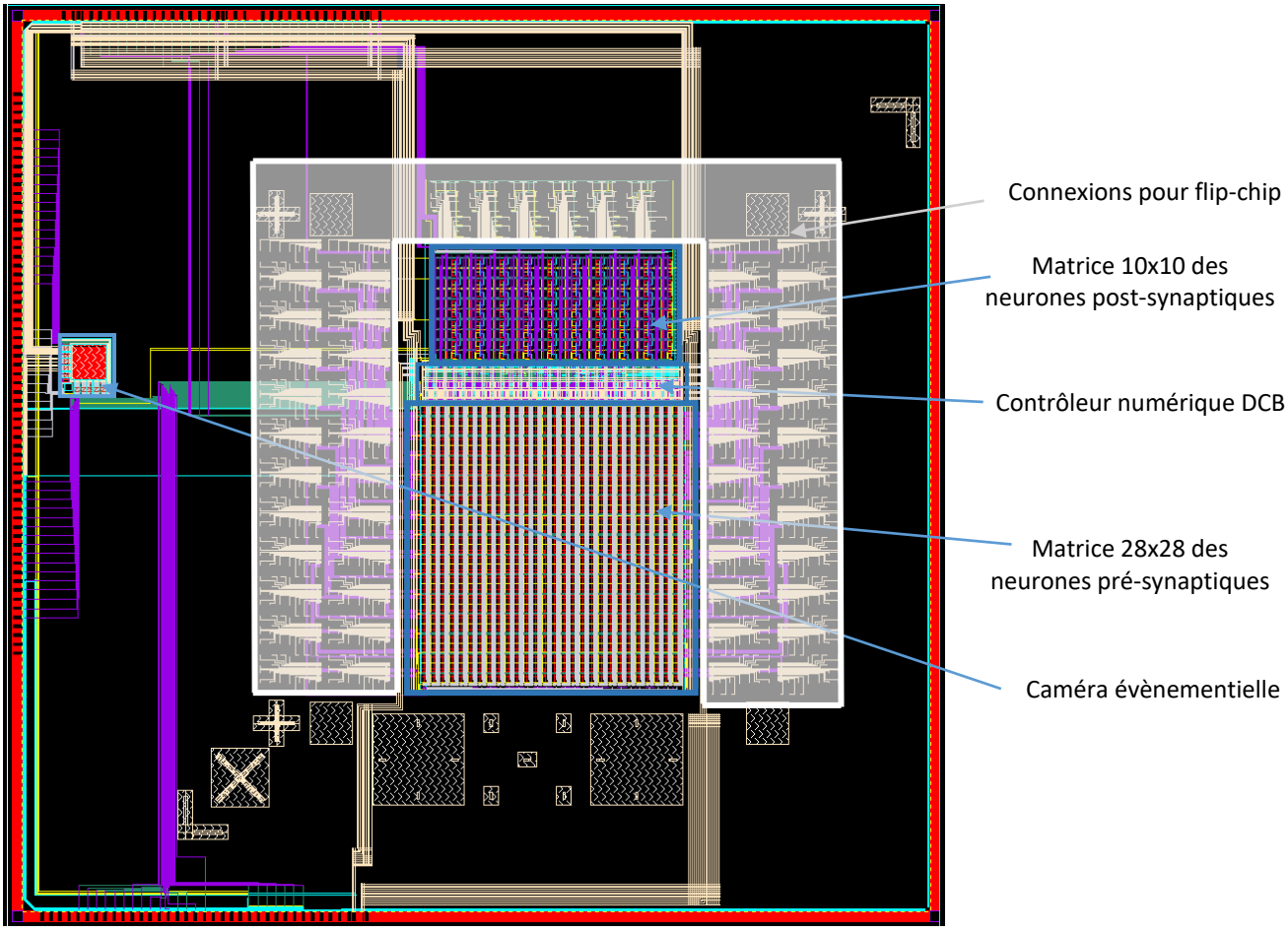


Figure 4.3 : Layout de la puce Hermes (Δx : 12.8 mm ; Δy : 12.6mm)

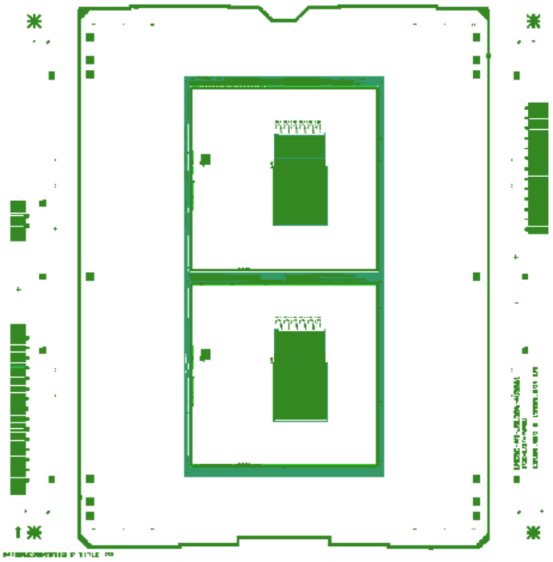


Figure 4.4 : Masque d'un réticule de notre puce.

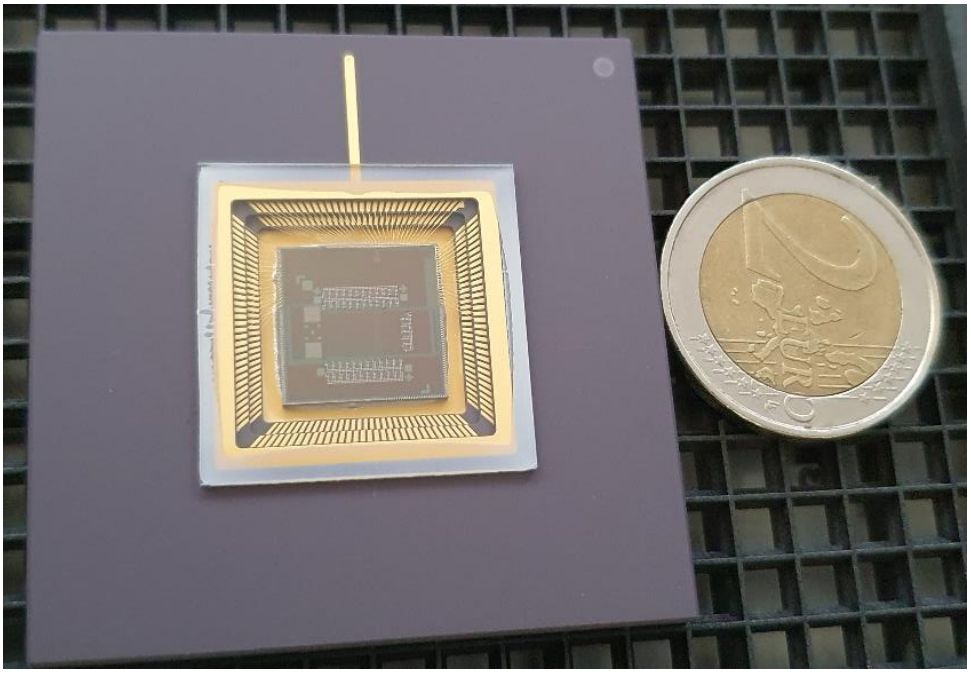


Figure 4.5 : Photo de la puce Hermes dans son boitier avec à ses coté une pièce de deux euros pour estimer sa taille.

Layout des sous-ensembles

Nous avons présenté le fonctionnement des neurones pré et post-synaptique dans le chapitre précédent, leur layout ainsi que celui du DCB sont présentés ci-après. L'objectif est de pouvoir assembler les différents sous-ensembles le plus aisément possible malgré les plus de 2500 connexions qui les unissent.

Le layout du neurone post-synaptique et celui de la matrice de neurones post-synaptiques 10x10 sont présentés (figure 4.6). Le convoyeur de courant et le convertisseur numérique analogique permettant la correction d'offset du CCII prennent une part importante de la surface du layout. La capacité de membrane bien qu'elle ne fasse qu'un pico Farad, sa surface est non négligeable. La taille d'un neurone post-synaptique est de 140µm x 330µm.

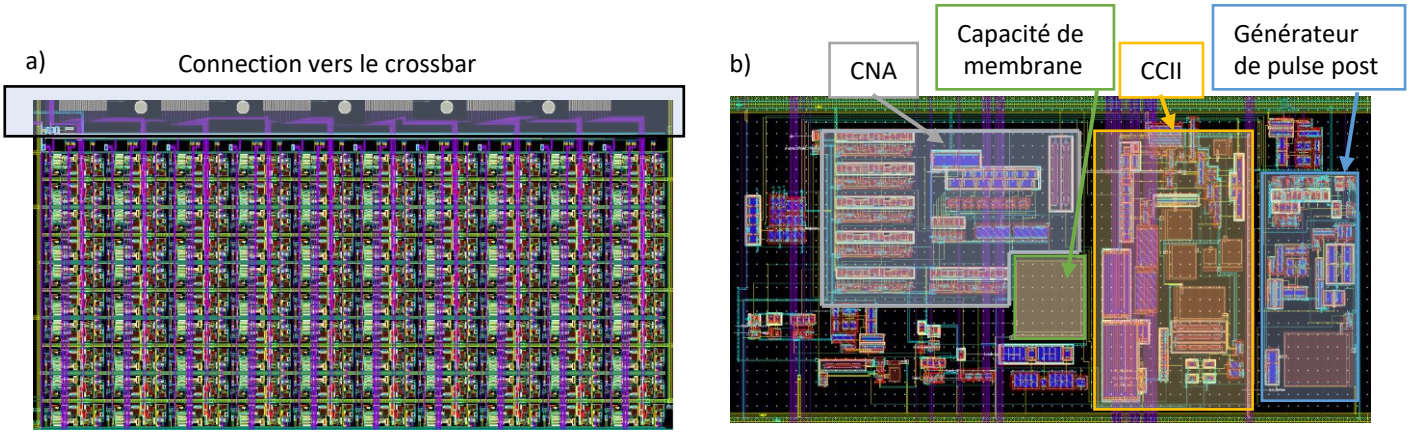


Figure 4.6 : a) Matrice de neurones post-synaptiques 10x10. b) Layout d'un neurone post-synaptique. Le layout du convertisseur numérique analogique (CNA) est présenté en gris, en jaune le convoyeur de courant (CCII), en bleu le générateur de pulse post-synaptique et en vert la capacité de membrane.

Le layout du neurone pré-synaptique et celui de la matrice de neurones pré-synaptiques 28x28 sont présentés (figure 4.7) Une grande partie de la surface du neurone pré-synaptique est recouvert de fils de connexions allant vers le crossbar, le DCB ou servent de courant de polarisation au neurone. Toutes les capacités ont été placées dans un coin pour laisser la place à ces fils. Pour éviter les effets d'antenne (antenna effect), on récupère les fils par une couche de métal supérieur. Cela permet de séparer de longs fils qui peuvent subir une variation transitoire de tension lors de la fabrication, le courant ne pouvant pas passer par les grilles des transistors la tension du fils peut devenir suffisamment importante pour qu'il ait un claquage de la grille. La taille d'un neurone pré-synaptique est de 128µm x 136µm.

Le layout du DCB est présenté (figure 4.8). Les 1568 connexions du bas sont connectées vers les neurones pré-synaptiques et permettent le déclenchement des pulses « up » et « down ». Les 201 connexions du haut sont connectées vers les neurones post-synaptiques, ils permettent de faire un reset individuel de la capacité de membrane de chaque neurone post-synaptique, de détecter lequel a déclenché et permet d'envoyer un signal à tous les neurones pour générer le pulse post-synaptique.

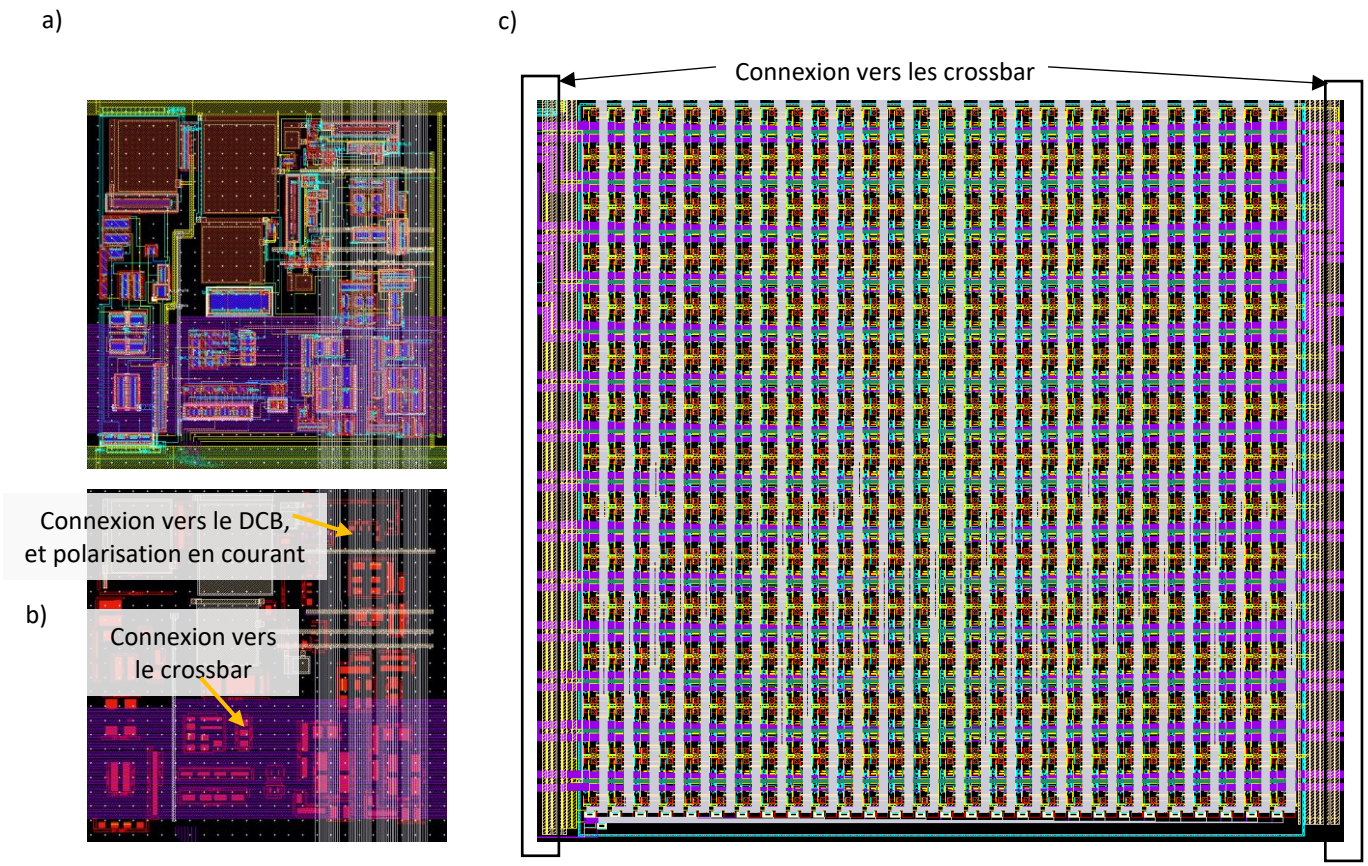


Figure 4.7 : a) Layout d'un neurone pré-synaptique, b) fils de connexions vers le crossbar métal4 (violet), et de la polarisation en courant métal 5 (gris). c) matrice de neurones pré-synaptiques 28x28.

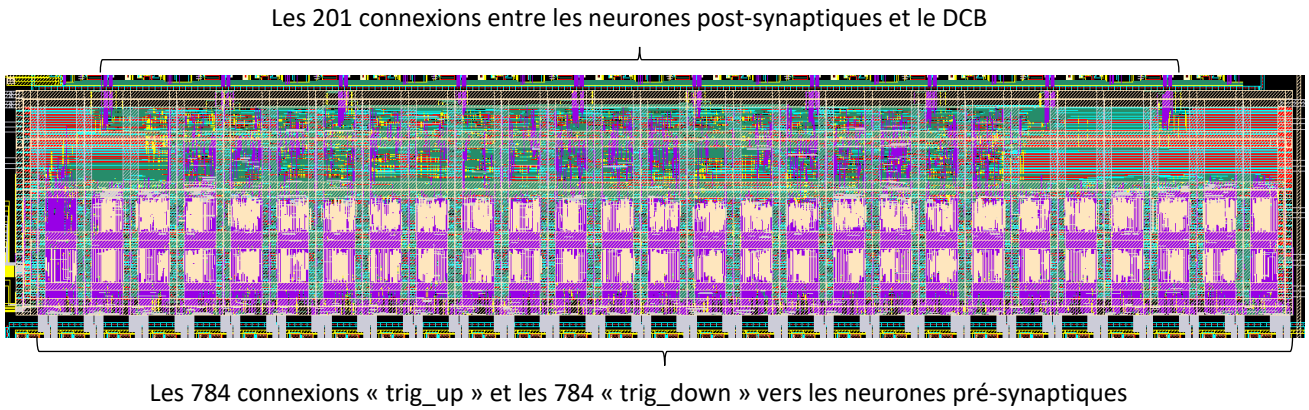


Figure 4.8 : Layout du DCB, avec en bas les 1568 connexions vers les neurones pré-synaptiques. En haut les 201 connexions vers les neurones post-synaptiques.

4.2 Le crossbar de memristors

Le crossbar que nous utilisons est composé d'une matrice de 784 lignes et 100 colonnes de memristors soit 78 400 memristors au total. Sa surface est de 10 mm x 10 mm et il fait 1mm d'épaisseur. Pour le

placer sur la puce, on utilisera la méthode du flip-chip. Ce procédé permet de connecter deux puces différentes entre elles sans utiliser de pads d'entrée sortie de la couronne externe de la puce. Le premier avantage est qu'il n'y a pas toutes les impédances parasites liées à un pad d'entrée sortie et que les connexions peuvent être placées au plus près de l'électronique. Le second avantage est que l'on peut aisément connecter nos 884 entrées sorties entre les deux puces via les solder balls.

Pour faire la connexion avec le flip-chip, on doit placer des billes de soudure sur la puce une par une sur disque de diamètre 60µm au dernier niveau de métal, sur lequel il y a eu une dépassivation pour enlever la couche d'isolation. Une fois que les billes ont été placées, le crossbar est mis par-dessus, on fait refondre les billes pour que la jonction soit bien réalisée. IBM Zurich fera le flip-chip, on a reçu de leur part l'empreinte pour pouvoir réaliser le layout avec la dépassivation des solder balls. Cette empreinte est présentée (figure 4.9), sur la partie de gauche il y a la vue de l'ensemble du layout, sur la partie de droite est montré un zoom d'un module de 17 solder balls et la connexion associée vers le réseau de neurones. Pour garantir un bon alignement, il est nécessaire de mettre des croix de placement, ces croix sont dessinées sur la puce sur le dernier niveau de métal. L'emplacement vide au centre est destiné à notre réseau de neurones.

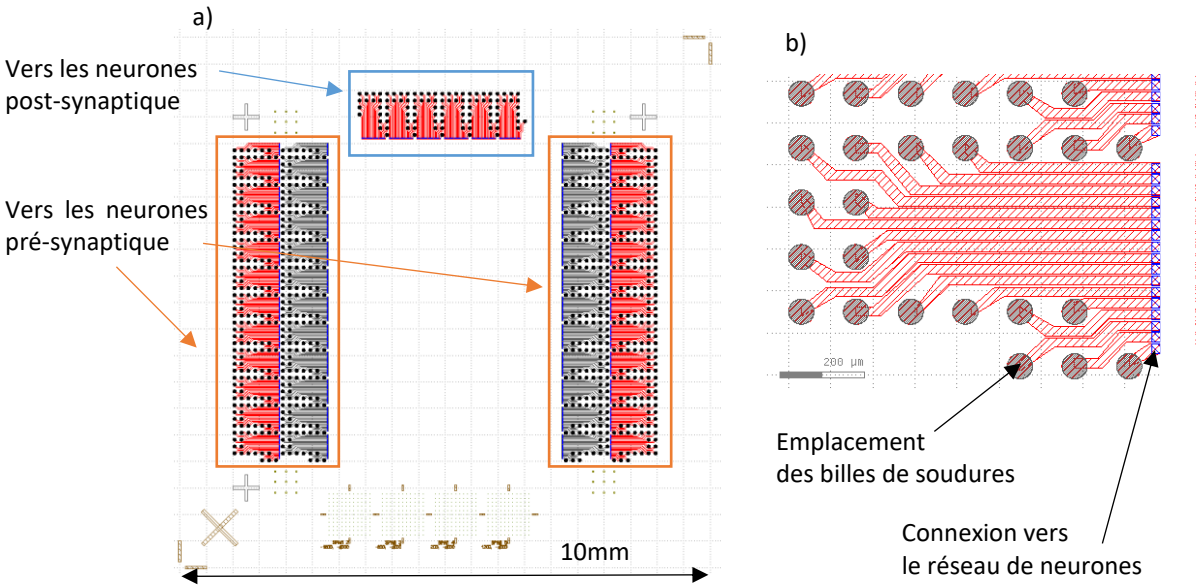


Figure 4.9 : Schéma fourni par IBM Zurich pour le flip-chip a) Le layout b) Zoom sur une partie.

Les chutes résistives du crossbar

Le crossbar de memristor n'est pas parfait, il présente des résistances de ligne qui sont proportionnelles à la longueur de la piste. Il a été montré dans un article [TRUONG S. N. 2019] l'effet néfaste des résistances parasites dans les crossbars de memristors utilisés dans un réseau de neurones, pouvant nuire à

l'apprentissage et à la reconnaissance. Les résistances parasites ont un plus fort impact quand la résistance des memristors est faible ou quand la matrice de memristors est de grande dimension.

Notre crossbar est composé d'une matrice de memristor de 784 par 100, ce qui correspond respectivement au nombre de neurones pré-synaptiques et post-synaptiques. La résistance de ligne, celle qui relie chaque memristor entre eux n'est pas nulle. La résistance r qui représente la résistance entre chaque memristor est estimée à r=1Ω par nos partenaires faisant le crossbar, illustration des résistances parasite (figure 4.10). Cette résistance de 1Ω peut paraitre dérisoire si l'on compare à la valeur minimale des memristors qui est de 1MΩ, mais dans certaines conditions où de grandes quantités de courant sont générées par le réseau, la chute de tension peut être importante. Les 784 neurones pré-synaptiques sont connectés par un memristor à chacun des 100 neurones post-synaptiques, la résistance de ligne parasite change en fonction de la position des neurones. Le neurone pré-synaptique N_{pre1} voit une résistance parasite totale de 784Ω sur la ligne des neurones post-synaptique, tandis que N_{pre784} ne voit que 1Ω.

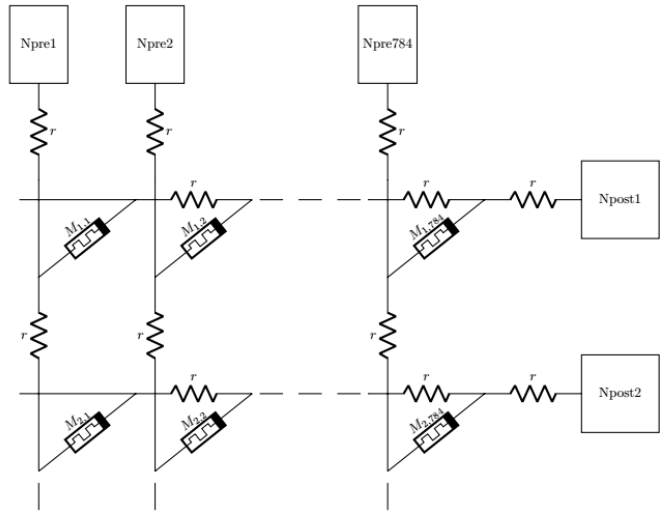


Figure 4.10 : Schéma du crossbar avec r la résistance d'accès parasite entre chaque memristor.

Une simulation d'une seule ligne du crossbar a été faite pour évaluer la chute de tension le long d'une seule ligne de transmission. Le but de cette simulation est de montrer l'effet auquel il faut être attentif et les problèmes qui en découlent. Les neurones post-synaptiques ne générant pas un signal pour la modification des poids sont fortement limités en courant. Le courant de forte intensité est uniquement généré par le neurone post-synaptique qui a déclenché, et on étudiera uniquement cette ligne de memristors. Sur la (figure 4.11) est présenté le schéma utilisé lors de la simulation, avec Vs=1V qui correspond à la tension du neurone post-synaptique, r=1Ω et les résistances R_mem correspondant aux résistances des memristors qui sont toutes identiques et valant 1MΩ (valeur minimale de nos memristors), afin de se placer dans le pire des cas possibles tous les memristors sont mis à 0V au niveau des neurones pré-synaptiques.

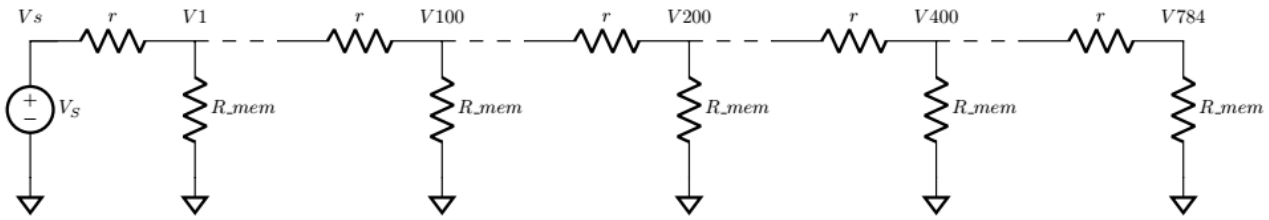


Figure 4.11 : Schéma d'une ligne du crossbar avec une résistance de ligne r = 1Ω entre chaque memristor.

Sur la (figure 4.12) est présentée la chute de tension simulée à partir du schéma de la (figure 4.11). On trace la chute de tension au niveau des 100-ème (V100), 200-ème (V200), 400-ème (V400) et dernier memristor (V784) en fonction de la résistance R_mem, que l'on a considérée identique pour tous les memristors et qui correspond donc à la résistance moyenne de chaque memristor. La résistance R_mem varie de la résistance minimale des memristors de 1MΩ à 15MΩ, 15MΩ correspond à la valeur moyenne des résistances du réseau après apprentissage [LEWDEN P. 2019]. Dans ces simulations on place toutes les résistances R_mem à 0V. Dans ces conditions on observe une chute de potentiel de 55mV, 102mV, 172mV et de 224mV, respectivement pour les tensions V100, V200, V400 et V784 avec R_mem = 1MΩ. Dans ces conditions et avec cette chute de tension, les derniers memristors de la ligne auront plus de mal à voir leur résistance augmentée.

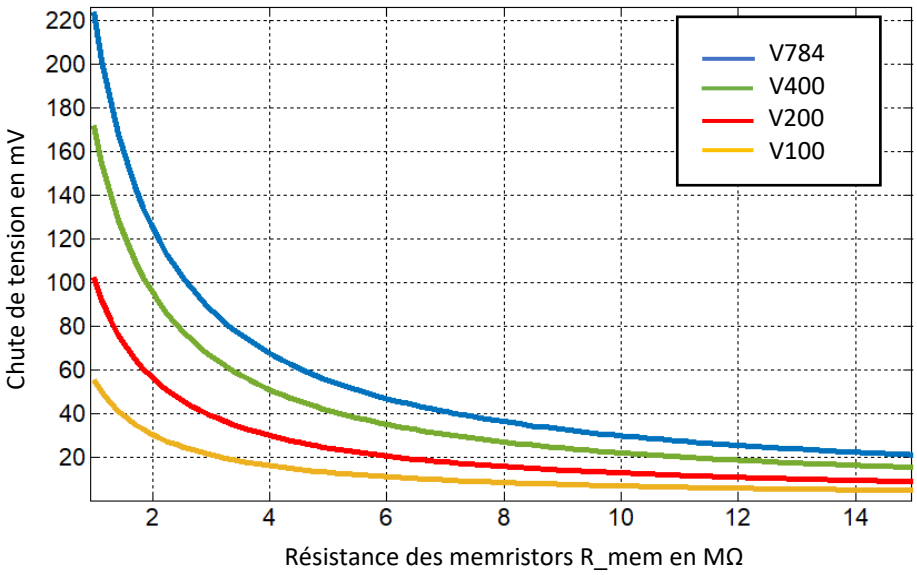


Figure 4.12 : Chute de potentiel le long de la ligne en fonction de la résistance des memristors R_mem.

Sur le réseau de neurones réel, la configuration où les résistances sont toutes à leur valeur minimale est très improbable, la résistance minimale de chaque memristor est potentiellement supérieure à 1MΩ, de plus les neurones pré-synaptiques ne seront pas tous au même potentiel. Il faut cependant faire attention

lors de l'initialisation à ce que les memristors n'aient pas une résistance trop faible au risque de nuire à la modification des poids synaptiques, en particulier pour les derniers memristors de la ligne. Dans ce cas il a la possibilité d'augmentée la tension Vup du neurone pré-synaptique des quelques centaines de millivolts pour pouvoir plus facilement augmenter la résistance des memristors. Il n'y a cependant pas de possibilité d'augmenter la tension Vup du côté des neurones post synaptiques, ce qui potentiellement pourrait nuire à la diminution la résistance des memristors à cause des chutes résistives. Cependant la chute de tension intervient pour de faibles résistances de memristors, si leurs résistances sont déjà à leur valeur minimale, leurs réductions deviennent inutiles.

Les résistances parasites d'accès peuvent poser problème pour des crossbars de très grande taille. Dans notre cas le problème rencontré est la chute de tension qui peut empêcher la modification correcte de poids, cette chute de tension est raisonnable et peut être compensée par l'initialisation ou l'augmentation de la tension Vup des neurones pré-synaptiques par exemple.

4.3 La caméra évènementielle

La caméra évènementielle placée proche du bord de la puce permet de filmer et d'envoyer de manière asynchrone l'information du pixel qui a déclenché. Chaque pixel est indépendant et est sensible à la variation de luminosité. Quand une variation de luminosité positive ou négative est détectée par un pixel la caméra envoie l'adresse de ce dernier via un bus de données vers le réseau de neurones ainsi que le sens de la variation de la luminosité.

La caméra que Prophesee nous a fournie est de format 32x32 pixels. Pour des raisons de contraintes physiques afin de notamment diminuer le courant nous sommes passés à 784 neurones équivalant à une matrice de 28x28. Cette taille correspond également à la taille des images de chiffre manuscrit de la base de données MNIST. Dans cette configuration les deux couronnes de pixels externes ne seront donc pas prises en compte.

Cette caméra est connectée à 18 tensions de polarisation externes permettant entre autres de changer la sensibilité du pixel. Pour la communication entre le réseau de neurones et la caméra, 13 fils sont utilisés permettant de donner l'adresse du pixel qui a déclenché en x et y par Data_x (0 : 4) et Data_y (0 : 4), le sens de polarisation par Data_pol et les commandes de synchronisation par req et ackb (figure 4.13). On peut également communiquer avec la caméra en ordonnant de déclencher un pixel, ou de connaitre lequel a déclenché. Pour cela on utilise les mêmes signaux que précédemment envoyés au réseau de neurones, mais ceux-ci sont envoyés cette fois vers le FPGA et sont présents dans les deux sens pour avoir une connexion entrante et sortante.

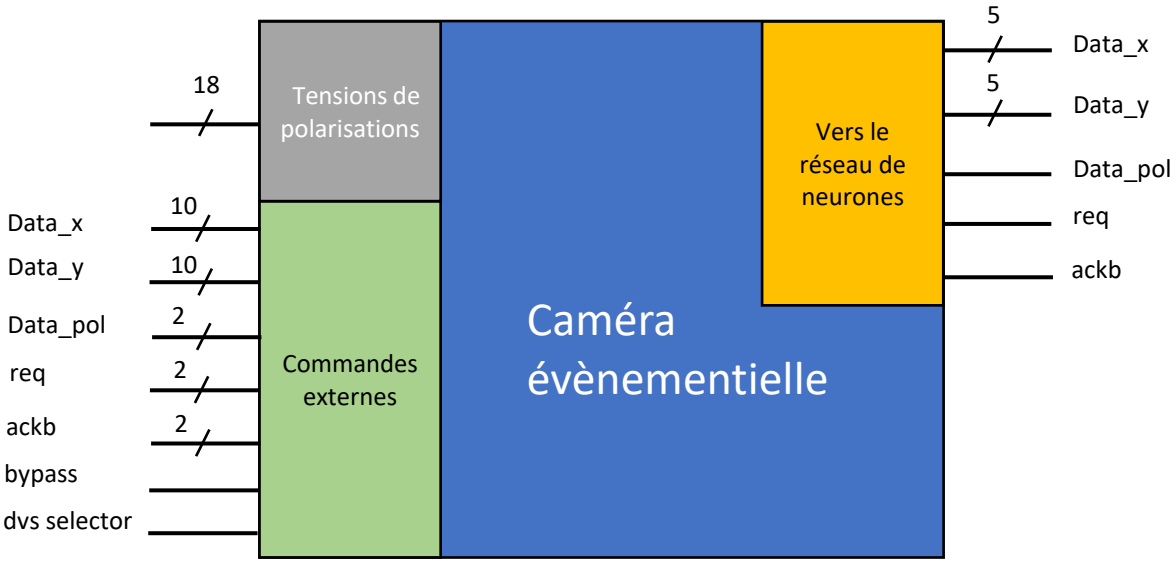
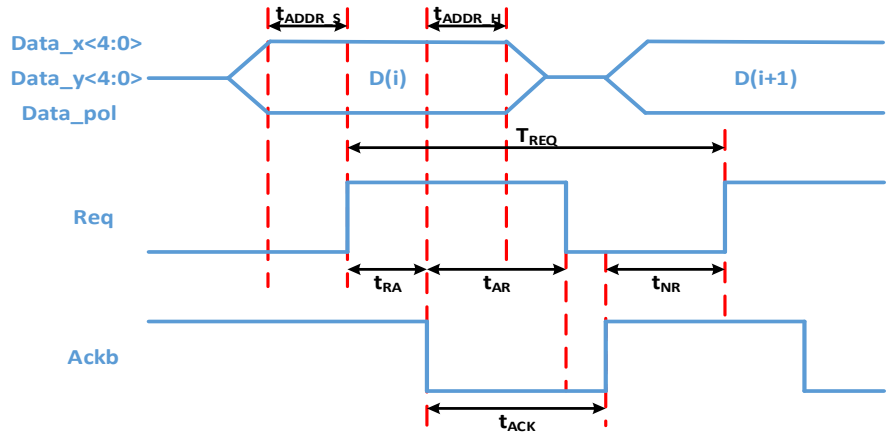


Figure 4.13 : Schéma bloc de la caméra évènementielle.

Pour traiter les signaux reçus de la caméra, on utilise le DCB, il interprète l'adresse du pixel et fait déclencher le neurone pré-synaptique correspondant. La communication entre le DCB et la caméra se fait en utilisant le protocole de conomination AER, décrit ci-après. Sur la (figure 4.14) sont représentés les différents signaux envoyés par la caméra et leur timing. Les signaux Data_x, data_y et data_pol sont envoyés ensemble et attendent l'arrivée de Req qui arrive 7ns ($t_{ADDR\ S}$) plus tard et sont synchronisés à ce moment-là par le DCB pour laisser le temps aux données de se stabiliser. Le signal Ackb est renvoyé vers la caméra 8ns (t_{RA}) après avoir reçu le front descendant Req pour confirmer la réception de l'adresse envoyée et met fin au signal de donné 4ns ($t_{ADDR\ H}$) après la réception puis met Req à 0 8ns (t_{AR}) après. Le signal Ackb est le complémentaire de Req décalé de 7ns. De nouvelles données contenant l'adresse peuvent arriver au plus vite toutes les 30ns (T_{REQ}).



t_{ADDR_S}	t_{ADDR_H}	T_{REQ}	t_{RA}	t_{AR}	t_{NR}	t_{ACK}
7ns	4ns	30ns	8ns	8ns	5ns	16ns

Figure 4.14 : Protocole de communication de la caméra.

4.4 Le contrôleur numérique (DCB)

Le contrôleur numérique est cadencé par une horloge de 5MHz fournie par le FPGA, tous les évènements qu'il génère ainsi que les fonctions internes sont synchronisées sur le front descendant de cette horloge. Il permet de faire l'interface entre les données reçues de la caméra et le réseau de neurones en décodant l'adresse du pixel qui a déclenché. Il permet aussi avec une grande souplesse de contrôler l'apprentissage du réseau en fonction des différentes stratégies choisies et permet également la communication vers l'extérieur pour informer quel neurone post-synaptique a déclenché. L'ensemble des neurones pré-synaptiques est contrôlé par le DCB il permet de déclencher le pulse « up » ou « down » de ses neurones, il contrôle également la mise à zéro des capacités de membrane et le déclenchement du pulse des neurones post-synaptiques.

Le DCB a été réalisé en VHDL et testé à petite échelle sur un FPGA avant de l'implémenter sur la puce. Il est composé de deux parties principales, l'une qui contrôle les neurones pré-synaptiques et la seconde les neurones post-synaptiques.

Détection de l'adresse.

La caméra évènementielle envoie une adresse au réseau de neurones qui correspond au pixel qui a détecté une variation de luminosité, dans les conditions les plus rapides elle peut en envoyer au réseau une nouvelle information toutes les 30 ns. Or la fréquence de fonctionnement du contrôleur numérique qui décode l'adresse est de 5 MHz soit une période de 200ns. Il n'est pas envisageable d'utiliser un fonctionnement synchrone pour détecter les adresses qui sont émises par la caméra, on risquerait que certains évènements ne soient pas pris en compte. Il faut donc que le contrôleur numérique décode

l'adresse de manière asynchrone par rapport à l'horloge principale. Cependant pour que l'intégrité des signaux puisse être vérifiée avec les outils numériques de synthèse, il n'est pas possible d'utiliser de bascule RS par exemple. L'outil numérique permet de vérifier que les signaux respectent le bon timing, en prenant les critères de coordination des portes logiques. Il est toutefois possible de faire un reset asynchrone sur une bascule D tout en assurant le bon fonctionnement du système, le reset asynchrone d'une bascule permet de récupérer l'adresse d'un pixel. On utilise le signal req provenant de la caméra pour valider le décodage de l'adresse. Afin que les signaux d'adresse soient stables, le signal req est retardé de 7ns, par rapport à l'apparition de l'adresse du pixel.

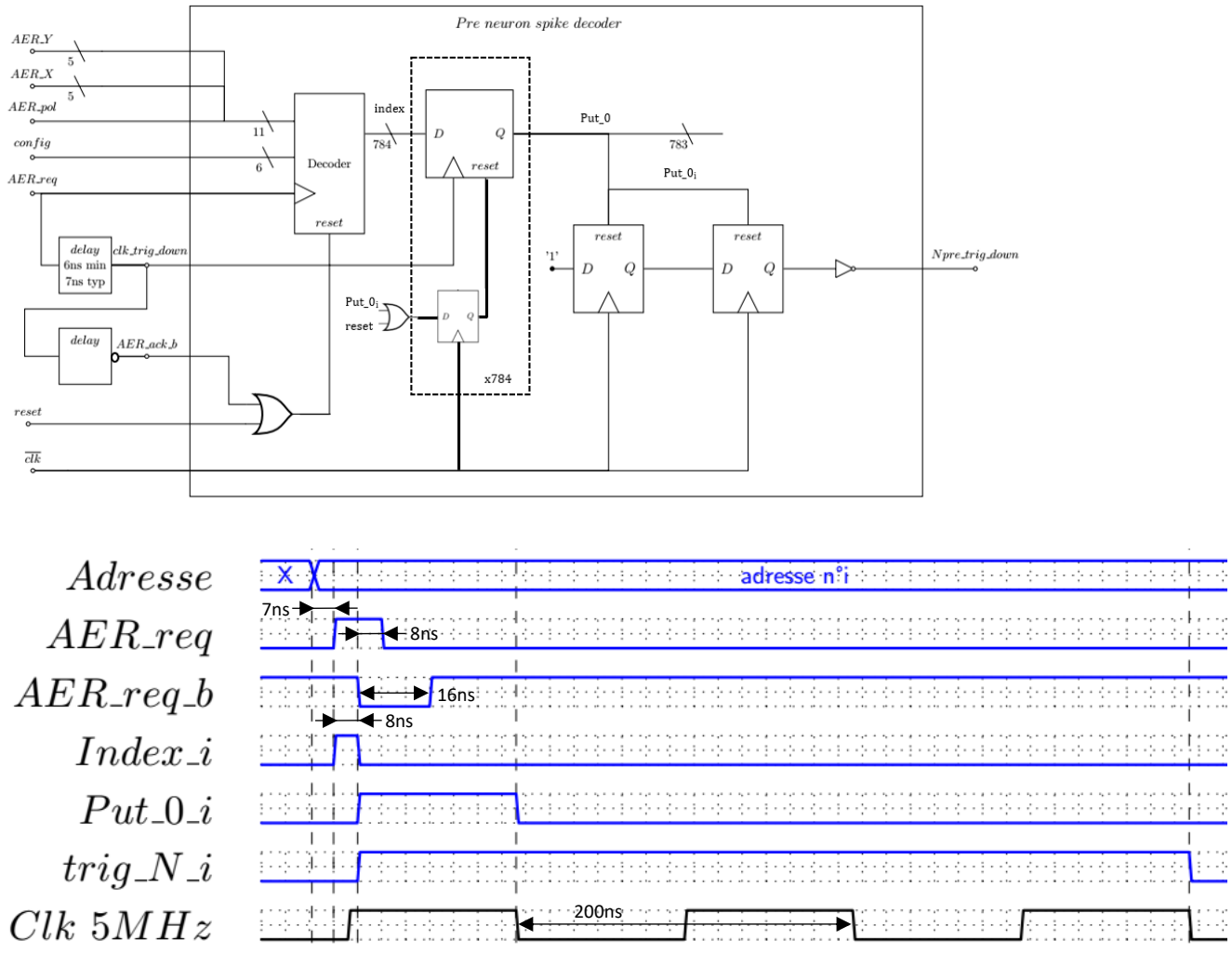


Figure 4.15 : Schéma électronique de détection d'une adresse dans le contrôleur numérique avec son chronogramme de fonctionnement.

Le schéma de fonctionnement de la détection d'adresse est présenté sur la (figure 4.15). Quand l'adresse est reçue par le contrôleur numérique, le décodeur interprète l'adresse et fournit l'index du neurone pré-synaptique correspondant. L'adresse correspond à AER_X et AER_Y appelé Data_X et Data_Y sur les figures 4.13 et 4.14, la polarisation AER_pol est appelé Data_pol et AER_req est appelé Req. Le décodeur est synchronisé avec le front montant de AER_req arrivant 7ns après l'arrivée des adresses depuis la

caméra. Il y a en sortie du décodeur 783 bits à '0' et un seul à '1', celui qui correspond à l'adresse du neurone. Put_0 permet de faire un reset asynchrone sur les deux bascules D. Pour que la commande puisse générer le pulse « down » dans un temps contrôlé, une bascule D permettant la remise à zéro de Put_0 a été placée entre le décodeur et les deux bascules D qui subissent un reset asynchrone. Une fois que ces bascules ont été mises à zéro, Npre_trig_down (trig_N_i sur la figure 4.15) passe à '1', fait déclencher le signal « down » du neurone pré-synaptique correspondant et est mis en mémoire, le signal Npre_trig_down reste actif au minimum durant les deux périodes d'horloge suivantes.

L'apprentissage

L'objectif de cette puce est de pouvoir faire de la reconnaissance d'image, pour cela on utilisera entre autres la base de données de la MNIST filmée depuis la caméra évènementielle intégrée, ou de la N-MNIST en envoyant directement les adresses depuis le FPGA, en entrainant le réseau pour qu'il modifie le poids de ses synapses sans intervention externe. Ces deux bases de données de chiffres manuscrits représentent les mêmes chiffres sous deux formats différents, la première sous forme d'images la seconde par des vidéos filmées par une caméra évènementielle. Notre caméra ne peut apercevoir que les variations de luminosité, si l'on souhaite montrer un résultat d'apprentissage cette caméra, il faudra que les pixels de l'image présentée subissent une variation de lumière en créant une saccade de la caméra par exemple. La base de données de la N-MNIST permettra toutefois de contrôler les données entrantes et de les comparer les résultats d'apprentissage à ceux de la littérature.

La STDP est notre règle d'apprentissage inspirée de la biologie. Cette règle nous permet de modifier le poids des memristors, mais contrairement au chapitre n°2 dans lequel les signaux avaient une forme bien particulière, les signaux générés par les neurones et appliqués sur le crossbar sont légèrement différents. Afin de réaliser l'apprentissage de notre réseau, nous utilisons trois signaux différents, deux venants des neurones pré-synaptiques et un des neurones post-synaptiques présentés (figure 4.16). Toutes les tensions et temps sont réglables afin de pouvoir optimiser le taux d'apprentissage quand on réalisera les tests sur la puce, les valeurs qui nous ont donné de bons résultats d'apprentissage en simulation sont les suivants : Vdown = 0.1V, Vmid = 1.1V, Vup =2.2V, t_1 = 10µs, t_2 = 0.5µs, t_3 = 0.5µs et t_4 = 1.5µs.

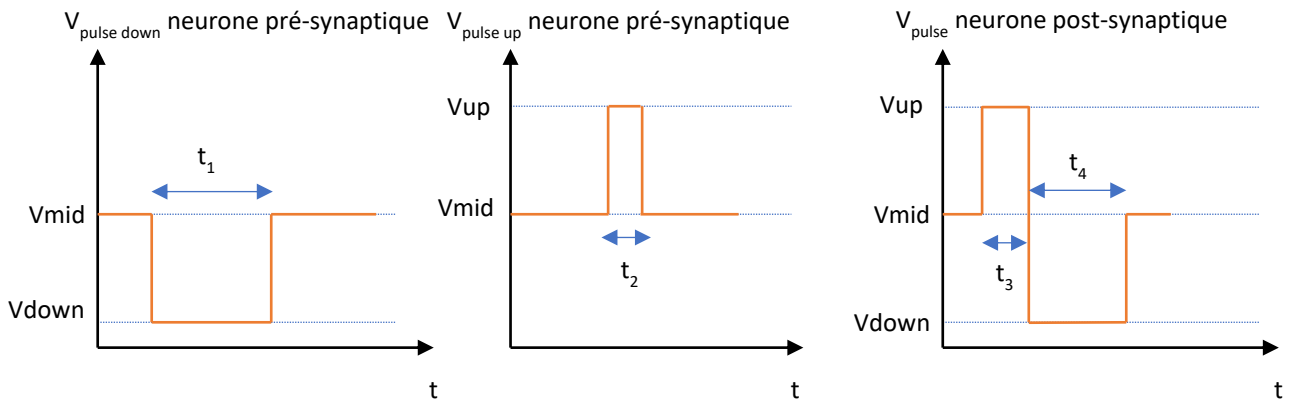


Figure 4.16 : Les trois différents signaux générés par les neurones pré et post-synaptiques.

Nous avons décidé de séparer le pulse pré-synaptique en deux parties, car elles ont deux fonctions bien distinctes, la première partie $V_{pulse\ down}$ permet de charger la capacité de membrane des neurones post-synaptiques, elle permet également d'augmenter le poids des synapses. La seconde partie est le signal $V_{pulse\ up}$, qui permet quant à lui la diminution des poids synaptiques et est déclenché peu de temps après V_{pulse} .

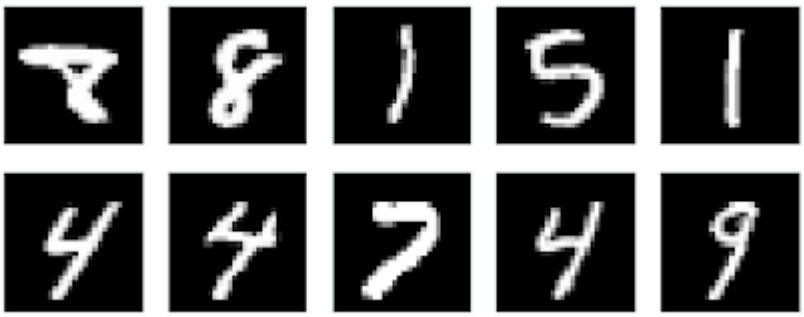


Figure 4.17 : Exemple de chiffres de la MNIST.

Quand un neurone post-synaptique a généré un potentiel d'action, la règle de la STDP stipule qu'il y a augmentation du poids de la synapse associé au neurone pré qui a généré un pulse « down » peu de temps avant et diminution du poids de la synapse associé au neurone qui a reçu un pulse « up » juste après. Sur les images que nous utilisons, nous souhaitons diminuer le poids des synapses qui sont reliées aux neurones pré-synaptiques qui n'ont pas déclenché d'événement et n'en déclencheront jamais. Les neurones qui ne déclencheront pas, sont ceux connectés au pixel de la caméra qui filme le font noir (figure 4.17) car il n'a pas de variation de lumière visible par la caméra sur ce font et il n'y aura donc aucun déclenchement du pixel concerné.

Pour pouvoir réaliser un apprentissage dans ces conditions, nous utilisons une technique d'apprentissage proche du « winer takes all » décliné en plusieurs versions afin d'optimiser l'apprentissage de notre réseau de neurones, et sont nommées 0P0D, 0P1D et 1P1D.

De nombreux paramètres permettent de mettre en œuvre les stratégies d'apprentissage implantées en contrôlant le déclenchement des neurones pré et post-synaptiques, d'initialiser le poids des synapses afin d'améliorer le taux de reconnaissance et sont décrits dans le tableau 4.1 suivant.

Tableau 4.1 : Liste des signaux de configuration des stratégies d'apprentissage

PIN	Fonction
CLK	Génère l'horloge du contrôleur numérique, le front descendant est actif
reset	Permet une initialisation au démarrage
Next_picture	Signale le changement d'image
AER_ selectPolUP	Sélectionne uniquement les pixels qui voient leur variation de lumière augmentée
AER_ selectPolDOWN	Sélectionne uniquement les pixels qui voient leur variation de lumière diminuée
videoON	Sélectionne le mode vidéo ou image par image
Config_post_N_ kill [2]	Permet une initialisation pour choisir quel neurone post-synaptique sera ou non inhibé tout le temps.
Refract_conter_ val [2]	Permet de choisir la valeur du compteur réfractaire
Xbar_read	Permet uniquement la lecture du crossbar sans modification des poids
Xbar_write	Permet la modification des poids des synapses
En_preN_trigUP	Permet ou non la génération des pulses « up »
En_preN_ trigDOWN	Permet ou non de générer les pulses « down »
POFon	Sélectionne le mode POF ou DIF

Les stratégies permettant l'initialisation

Activation des pulses « up » et « down »

L'état initial du crossbar de memristors peut être quelconque, on peut alors rencontrer des difficultés pour un apprentissage correct du réseau. L'initialisation idéale est une répartition aléatoire des résistances. Il faudra donc pouvoir uniquement augmenter ou uniquement diminuer le poids des synapses que l'on souhaite. En_preN_trigUP et En_preN_trigDOWN permettent d'activer ou de désactiver respectivement les pulses « up » et « down » uniquement lors de la modification des poids et permet la sélection de respectivement la diminution des poids. Les pulses « down » resteront activés pour charger les capacités de membrane des neurones post-synaptiques.

Inhibition des neurones post-synaptiques

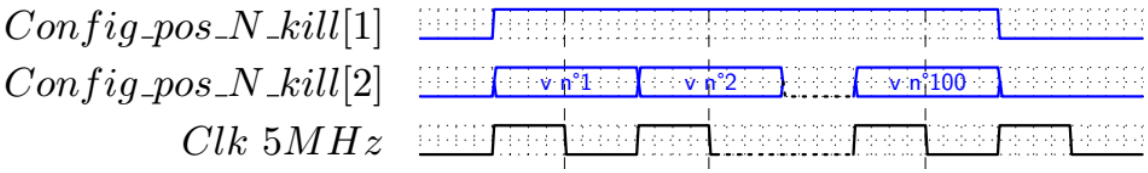


Figure 4.18 : L'inhibition des neurones post-synaptiques est modifiée quand Config_post_N_kill [1] passe à 1 et inhibe le neurone n°i si la valeur n°i (v n°i) de Config_post_N_kill [2] est à 1.

Durant l'initialisation on peut choisir les neurones post-synaptiques qui seront actifs ou non, ce qui implique que seuls les memristors reliés aux neurones actifs verront leur résistance modifiée. Il est également possible d'inhiber les neurones post-synaptiques défaillant durant la phase d'apprentissage. Config_post_N_kill est un signal sur deux bits qui permet l'inhibition d'un neurone post-synaptique particulier. Le premier bit valide le fait de changer le paramètre d'inhibition sur les neurones post-synaptique, le second donne la valeur si le neurone concerné est actif. On commence par modifier le paramètre d'inhibition sur le premier neurone (Valeur n°1) jusqu'au 100ème neurone (Valeur n°100), l'horloge permet de valider à chaque front descendant le neurone concerné, voir (figure 4.18).

Les stratégies permettant une amélioration de la reconnaissance

Compteur réfractaire

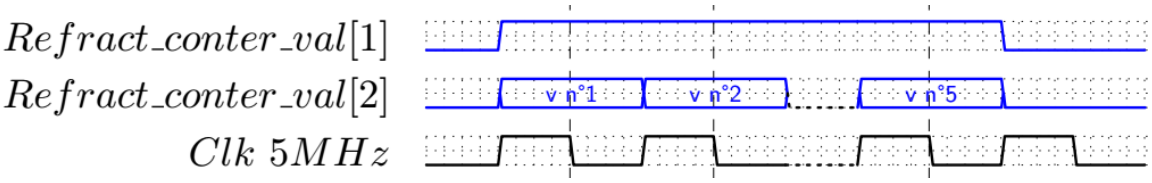


Figure 4.19 : La valeur du compteur réfractaire est modifiée quand Refract_conter_val [1] passe à 1 et donne au bit n°i la valeur n°i (v n°i) de Refract_conter_val [2].

Selon l'état initial et les disparités de fabrication des memristors du crossbar, il est possible que quelques neurones post-synaptiques soient connectés à des memristors dont la résistance est bien plus élevée ou plus faible que la moyenne de l'ensemble du réseau de neurones. Ceci a pour conséquence que certains neurones déclencheront trop souvent tandis que d'autre pas assez, ralentissant voir réduisant la capacité d'apprentissage du réseau. Pour éviter ce problème, il a été décidé d'inhiber un neurone post-synaptique qui vient de déclencher un certain nombre de tours. Un tour correspond au déclenchement d'un autre neurone post-synaptique. Le compteur qui s'occupe d'inhiber les neurones est appelé compteur réfractaire, sa valeur est stockée sur 5 bits et peut donc aller de 0 à 31. Pour régler cette valeur, on utilise Refract_conter_val qui est sur 2 bits, le premier donnant l'autorisation de modifier la valeur du compteur réfractaire, le second change un par un les bits de ce compteur (voir figure 4.19), le changement s'effectue

sur le front descendant. Chaque neurone post-synaptique est muni de son propre compteur réfractaire. Un neurone est inactif quand on met la valeur de son compteur à 0, et le reste tend que son compteur réfractaire est inférieur à la valeur fixée par Refract_conter_val.

Sélection de la variation de luminosité

Quand un pixel déclenche, il a vu une variation de luminosité soit positive, soit négative et envoie cette information à notre réseau de neurones. Nous ne traitons pas différemment les variations de luminosité positive ou négative, le même pulse « down » est envoyé par le neurone pré-synaptique correspondant. Il est cependant possible de choisir quel sens de variation de luminosité on souhaite garder. Si AER_selectPOLUP est à '1' on permet la détection de luminosité positive et si AER_selectPolDOWN est à '1' ce sont les variations négatives de luminosité qui sont prises en compte. Il est bien entendu possible d'activer en même temps les deux sens de polarité.

Sélection vidéo ou image par image

Il y a possibilité de choisir soit de présenter les images une par une aux réseaux de neurone, soit sous forme d'un film continu. Les images sont bien entendu représentées sous forme de films, cependant quand un neurone post-synaptique a déclenché, on change d'image et on informe le réseau que l'on a changé d'image en faisant passer next_picture à '1'. Pour sélectionner le mode « film » on met videoON à '1', s'il est à '0' le mode « image par image » est choisi. Le mode POFon permet de sélectionner le mode d'apprentissage 1P1D s'il est actif, sélectionne le mode 0P1D sinon. Ces deux modes d'apprentissage sont expliqués dans le paragraphe suivant. La stratégie d'apprentissage 0P0D n'a cependant pas été implantée dans le contrôleur numérique, car elle n'était pas optimale pour notre réseau.

Priorisation de la sélection du neurone post synaptique

Quand le contrôleur numérique a détecté le déclenchement d'un neurone post-synaptique, c'est-à-dire quand la capacité de membrane d'un neurone post-synaptique a atteint la tension de seuil, il envoie un signal au DCB. Si plusieurs neurones ont déclenché en même temps, le DCB n'en garde qu'un seul, celui qui a l'index le plus faible, puis met la capacité de membrane à 0V pour tous les autres neurones.

Présentation des stratégies d'apprentissage

Les trois stratégies d'apprentissage présentées et utilisées lors des simulations sont les suivants 0P0D, 0P1D et 1P1D [LEWDEN P. 2020]. La règle 0P0D permet d'envoyer un pulse « down » au neurone correspondant au pixel de la caméra qui a détecté une variation de luminosité. Quand un neurone post-synaptique a été déclenché, un pulse « up » est émis par tous les neurones pré-synaptiques qui ne sont pas en train d'émettre un pulse « down », dans le but de diminuer la valeur des poids des synapses correspondantes. Les synapses qui sont connectées aux neurones actifs avant le déclenchement du neurone post-synaptique voient leurs poids augmenter.

La règle 0P1D, comme pour la règle précédente, seules les synapses qui sont connectées aux neurones actifs qui ont précédé au déclenchement du neurone post-synaptique du temps t1 (figure 4.16), voient leurs poids augmenter. Mais contrairement à la règle précédente, seuls les neurones pré-synaptiques qui n'ont pas émis du tout de pulse « down » émettent un pulse « up », ce qui fait diminuer le poids des synapses correspondantes.

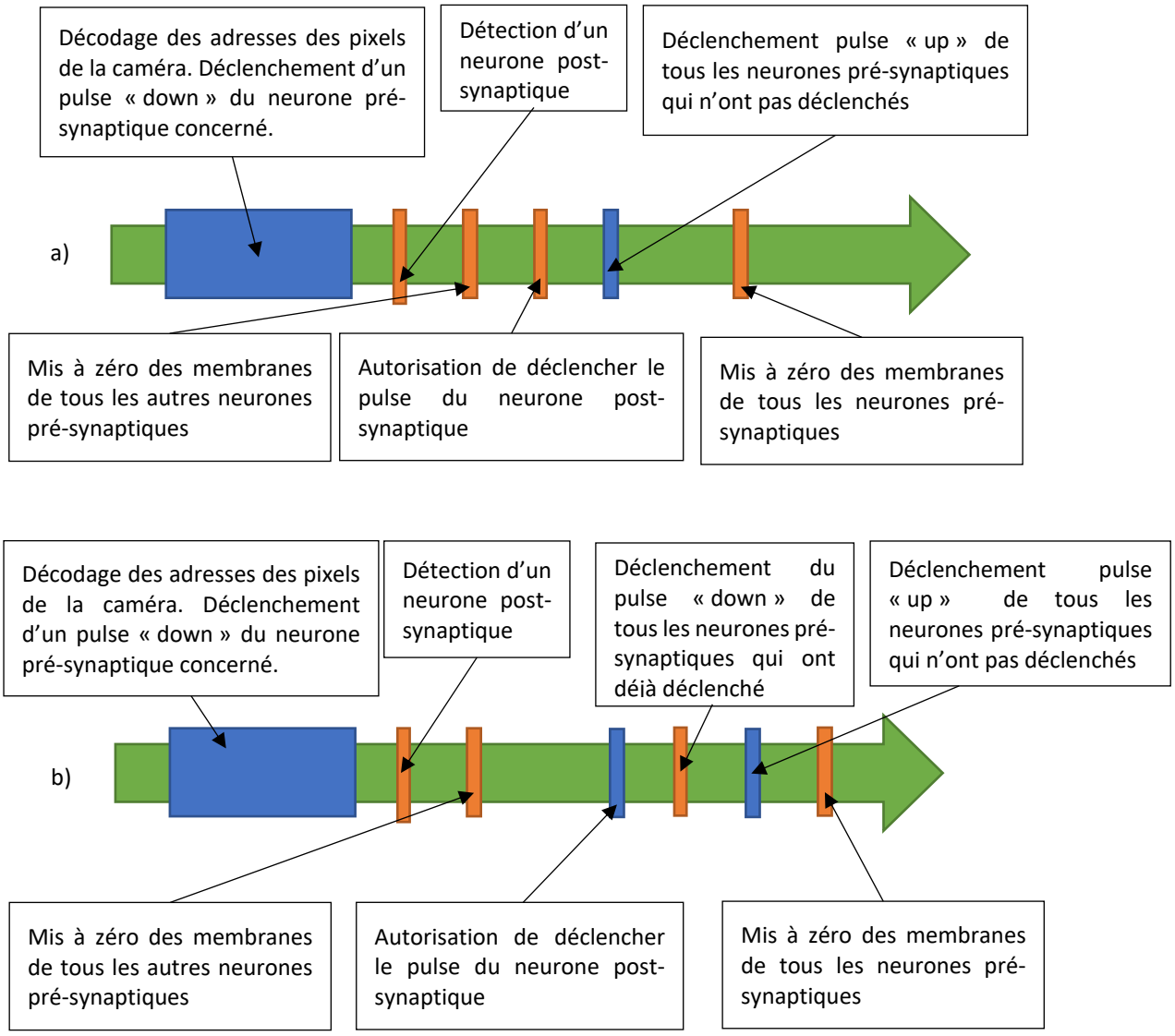


Figure 4.20 : Suite d'évènements au cours du temps orchestrée par le contrôleur numérique. a) stratégie d'apprentissage 0P1D. b) stratégie d'apprentissage 1P1D.

La dernière règle 1P1D permet de regénérer après le déclenchement d'un neurone post-synaptique un pulse « down » par tous les neurones pré-synaptiques qui en ont déjà envoyé au moins un avant son déclenchement, ce qui implique que toutes les synapses qui leur sont reliées voient leur poids augmenter. Comme la règle précédente 0P1D, après qu'un neurone post-synaptique a déclenché, tous les neurones pré-synaptiques qui n'ont pas reçu d'évènement de la caméra se voient émettre un pulse « up » qui

permet de diminuer le poids des synapses reliées à ces neurones. Finalement et quelle que soit la règle d'apprentissage, on met la capacité de membrane du neurone qui a déclenché à 0V, son compteur réfractaire passe à zéro et on incrémente le compteur réfractaire tous les autres neurones. La suite de ces évènements orchestrée par le contrôleur numérique est présentée sous forme de schéma des événements sur la (figure4.20) et représente en a) la règle 0P1D et en b) la 1P1D.

4.5 Simulation des règles d'apprentissage

Pour comprendre le fonctionnement de ces trois différents types d'apprentissages, des simulations ont été faites avec les 3 différentes règles, 0P0D, 0D1P et 1P1D. Le but de la simulation est de reproduire le fonctionnement du réseau de neurone hardware. Un doctorant de notre équipe a réalisé un modèle compact du convoyeur de courant sur Python à partir des simulations faites sur Cadence, il permet de calculer le courant qui charge à chaque instant la capacité de membrane du neurone post synaptique concerné, les memristors sont modélisés par une résistance. La division en courant utilisée lors de la simulation par le convoyeur de courant est de 100. On a réalisé deux types de simulation, la première pour vérifier le bon apprentissage des stratégies proposées, la seconde pour évaluer l'apprentissage en ajoutant une variabilité de l'offset du convoyeur de courant et une variabilité de la capacité de membrane.

Simulation du réseau avec les trois différentes règles d'apprentissage

Les premières séries de simulations se feront sur la base de la N-MNIST avec en entrée 1156 neurones correspondant à la taille en pixel de la base de données qui présente une image de 34 x 34. Les trois saccades d'un seul chiffre sont présentées sous forme d'un film de 300ms, on utilisera pour notre simulation uniquement les premières 100ms du film qui correspond à la première saccade, uniquement les variations de lumière positives seront prises en compte. Notre système génère des signaux qui sont rapides par rapport aux films de la N-MNIST, si on se réfère à la (figure 4.16) le temps qui permet la charge de la capacité de membrane est t_1 = 10µs. Les autres temps ne sont pas pris en compte, ils ne sont pas nécessaires dans la simulation, où seules les variations des poids synaptiques sont calculées. Par construction, notre réseau de neurones a une dynamique de fonctionnement plus rapide qu'un film de la N-MNIST. Pour remédier à ce problème les simulations ont été faites avec une accélération jusqu'à 100 fois, le taux de reconnaissance est présenté (figure 4.21). Le taux d'apprentissage maximal est de 67,7% établi avec la règle 1P1D sans accélération de la N-MNIST. En augmentant la vitesse de lecture de la base de données, la règle d'apprentissage 0P0D fait des progrès spectaculaires en passant de 0% à 46,8% pour une accélération allant de 1 à 100.

On peut voir également un léger progrès avec la règle 0P1D et une accélération de la N-MNIST, pour les mêmes raisons, plus d'évènements arrivent en même temps et donc plus de synapses voient leur poids augmenter en même temps. Cependant le taux de reconnaissance à faible vitesse est non nul car il n'y a pas de diminution du poids des synapses reliées aux neurones qui ont déjà émis un pulse « down ». La variation de conductance des memristors évolue de la manière suivante :

$$(4.1)\ \Delta G = \begin{cases} +A * (G_{max} - G_0)\ si\ V_{sys} < 1{,}2V \\ -A * (G_0 - G_{min})\ si\ V_{sys} > 1{,}2V \\ 0 \qquad\qquad\qquad sinon \end{cases}$$

Avec ΔG la variation de conductance du memristor, A le taux d'apprentissage valant 0,05, G_{max} compris entre 0.,1µS et 1µS, G_{min} compris entre 1nS et 10nS, G_0 la conductance actuelle du memristor et enfin V_{sys} la résultante de la tension sur le memristor des tensions des neurones pré et post-synaptiques. V_{sys} est supérieur à 1,2V uniquement quand un neurone pré-synaptique a déclenché un pulse « up » après un neurone post-synaptique, la tension V_{sys} est inférieure à 1,2V uniquement si un neurone pré-synaptique un pulse « down » avant le pulse d'un neurone post-synaptique, et suit donc la variation de la conductance de la STDP.

Sur la (figure 4.22) on voit la conductance de chaque synapse après apprentissage. Chaque case correspond à un neurone post-synaptique et chaque pixel de cette case correspond au poids synaptique d'un memristor. On voit des chiffres apparaitre dans les tons jaune/vert correspondant au plus fort poids des synapses.

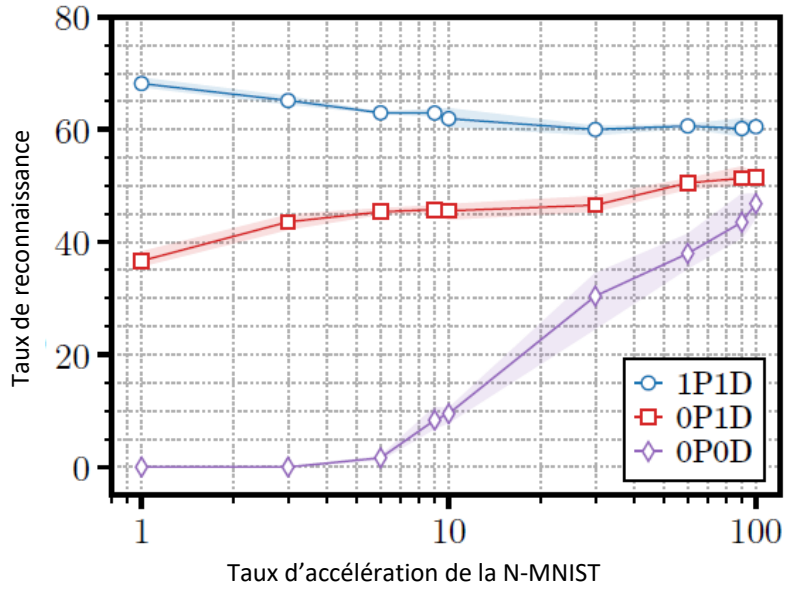


Figure 4.21 : Taux de reconnaissance d'une simulation de notre réseau de neurones en fonction de trois différentes règles d'apprentissage et de l'accélération de la N-MNIST.

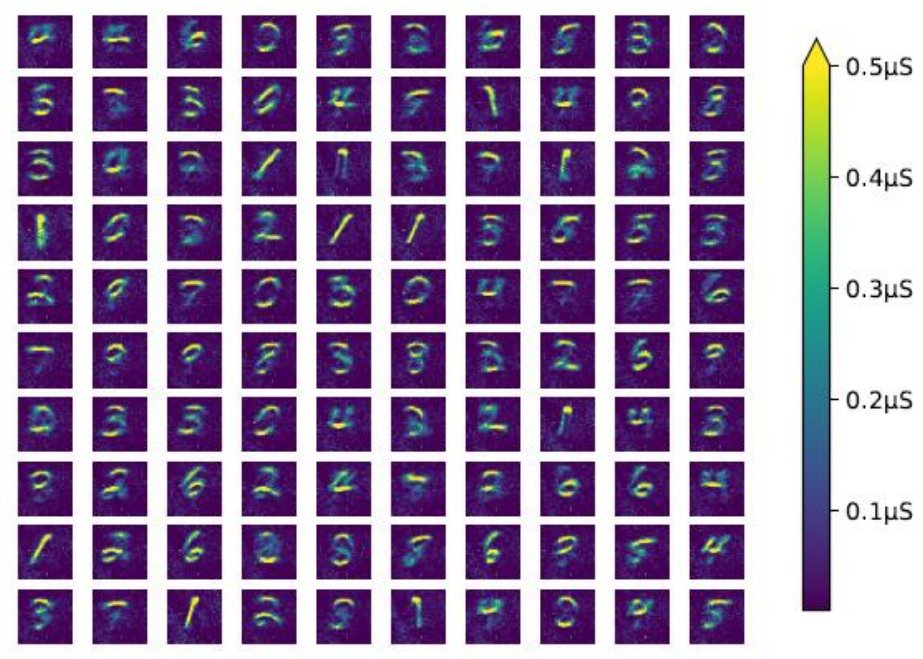


Figure 4.22 : Valeurs des synapses pour chaque neurone post-synaptique après l'apprentissage.

Simulation avec les disparités physiques des composants

Une autre série de simulations a été faite en prenant en compte les variations physiques du système dans le but de vérifier si une variabilité dans les paramètres physiques peut être nuisible ou non à l'apprentissage [LEWDEN P. 2019]. Dans ces simulations seuls les chiffres 5,6 et 9 de la N-MNIST seront pris en compte pour des raisons de vitesse de calcul, mais également pour se comparer au résultat de la littérature. Ces simulations ont été réalisées durant la conception du circuit, les pulses « up » et « down » sont réunis dans un unique signal présenté (figure 4.23). La technique d'apprentissage est sensiblement la même que celle que l'on utilise aujourd'hui, on utilise la 0P1D et la variation des poids est identique que précédemment et suit l'équation ().

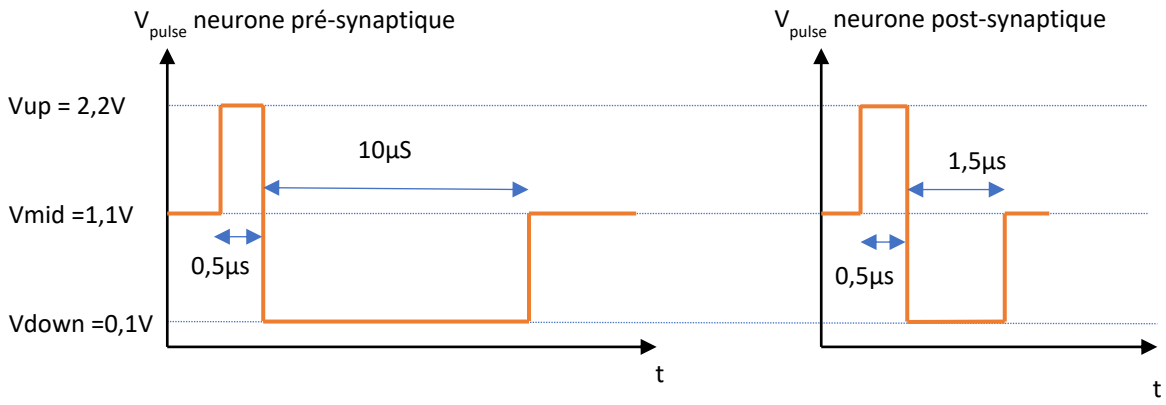


Figure 4.23 : Pulse pré et post-synaptique utilisée pour la simulation.

Les variations prises en compte pour cette simulation sont la variation de la capacité de membrane et la variation de l'offset du CCII. La capacité de membrane est de 1pF, elle sera dans la simulation comprise entre 0,8pF et 1,2pF répartie aléatoirement de manière uniforme sur l'ensemble des neurones post-synaptiques. L'offset du convoyeur de courant est également réparti aléatoirement de manière uniforme de 0mV à 4mV (figure 4.24). En a) on peut voir le taux d'apprentissage en fonction de la variation de l'offset et en b) le pourcentage de neurones non actifs en fonction de l'offset du convoyeur de courant. Les résultats sont le fruit d'une moyenne de 5 simulations pour une valeur d'offset maximale donnée. Il y a donc 10 simulations qui ont été faites pour chaque valeur d'offset, 5 pour une répartition aléatoire de la capacité de membrane sur chaque neurone et 5 pour une capacité de membrane égale à un pico Farad pour tous les neurones post-synaptiques. Sans variabilité de capacité de membrane, en moyenne le taux de reconnaissance y est plus important et vaut 92,4% pour un offset de 0mV, ce taux de reconnaissance tombe à 87,3% pour un offset de 4mV. Avec la variation de la valeur de la capacité de membrane que l'on a choisie, les résultats d'apprentissage sont en moyenne moins bons que les résultats de simulations sans variation, mais restent très proches. Quand on augmente la dispersion de l'offset, on observe que l'on utilise de moins en moins de neurones, on a entre 70% et 80% des neurones qui sont inutilisés quand l'offset devient égal à 4mV. Ce résultat est dû au fait que pour un pulse généré par un neurone pré-synaptique et pour un poids synaptique identique, moins de courant chargera la capacité de membrane du neurone post-synaptique, voire pas du tout quand l'offset devient trop grand.

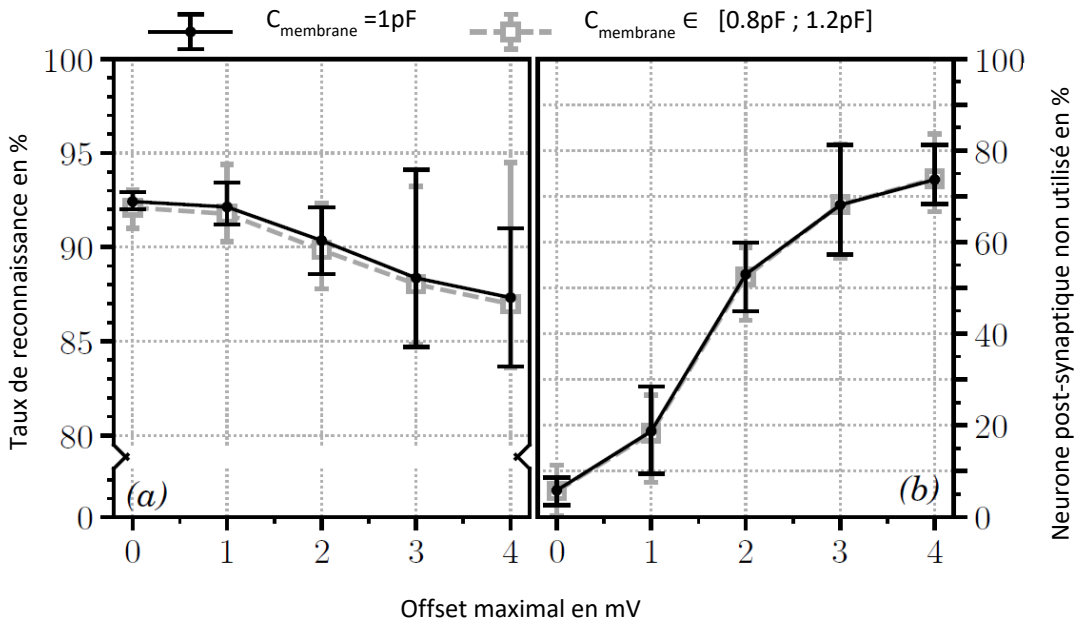


Figure 4.24 : a) Résultat d'apprentissage en fonction de la variation maximale de l'offset des convoyeurs de courant des neurones post-synaptiques. b) Taux de neurones non utilisés en fonction de la variation maximale de l'offset

La variation de l'offset est un point critique dont il faut tenir compte, dans la simulation précédente seul un offset de 4mV au maximum a été simulé avec uniquement trois classes de chiffres le « 5 », le « 6 » et le « 9 ». Si l'on ne fait rien, l'offset du convoyeur de courant sera bien plus important. Une simulation Monte-Carlo de 500 échantillons montre la distribution de l'offset du neurone post-synaptique sur la tension de l'entrée X du convoyeur de courant (Vx) en (figure 4.25). La valeur minimum et maximum des offsets est respectivement de -32mV et de +36mV. La dispersion des offsets si on ne faisait rien est bien trop importante pour pouvoir réaliser un apprentissage. De plus le neurone dont le CCII a un offset négatif charge sa capacité de membrane en permanence et donc déclenchera sans raison, nuisant à l'apprentissage du réseau. La correction d'offset permet de supprimer ce problème en configurant l'offset au plus proche de 0V, mais en restant toujours au-dessus. Sur la (figure 4.26) est montré le résultat de simulation Monte-Carlo de la répartition des offsets après avoir fait la correction sur un total de 500 neurones. Sur ces 500 simulations différentes, seules 2 valeurs d'offsets n'ont pas pu être amenées au-dessus de 0V, et une seule de 9mV qui était à l'origine trop grand pour être ramenée le plus proche de zéro possible. Dans l'optique de mieux voir la répartition des offsets, les trois résultats extrêmes ne sont pas présents sur la (figure 4.26). Les neurones dont le CCII a un offset négatif après la correction seront inhibés et ne pourront donc plus fonctionner. Les neurones dont leur offset est important, mais positif ne posent pas de problème non plus, si l'offset est vraiment important, on peut considérer que le neurone en question est inhibé de fait.

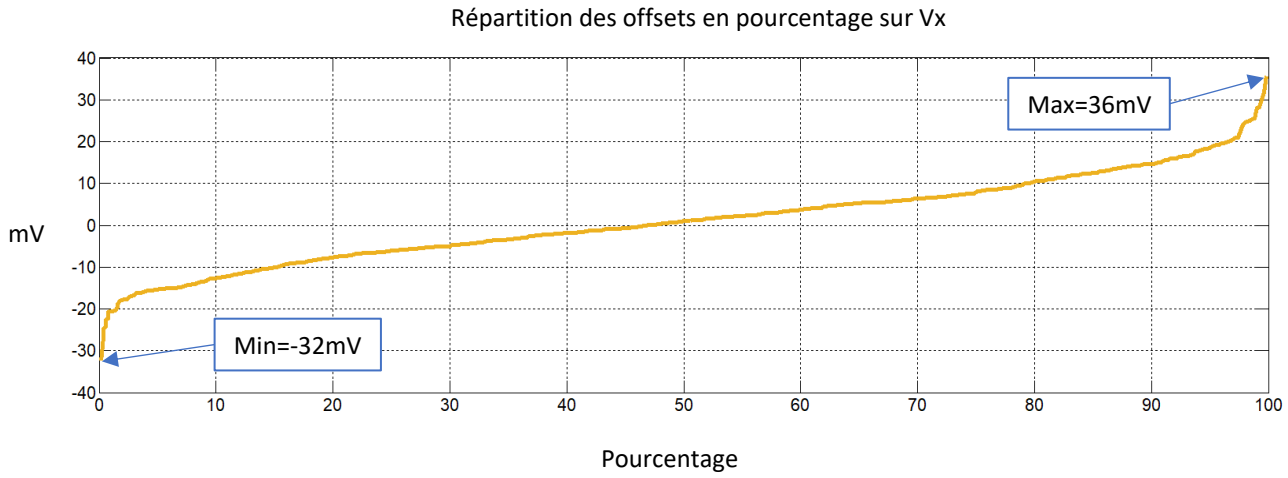


Figure 4.25 : Répartition des offsets du convoyeur de courant sur l'entrée X par rapport à la tension de commande Vmid = 1,1V après une simulation Monte-Carlo avec un total de 500 échantillons et avant correction de l'offset.

On peut remarquer qu'après la correction d'offset présentée à la (figure 4.26), il n'y a pas d'offset corrigé entre 0V et 1,5mV, et 82% des offsets sont entre 1,5mV et 2,5mV. Cette particularité permet aux réseaux de neurones d'être plus résistants au bruit, cela évite qu'à la moindre variation de tension un courant ne

vienne charger la capacité de membrane et permet d'éviter ainsi le déclenchement de neurones de manière intempestive, ce qui risquerait de dégrader les performances d'apprentissage.

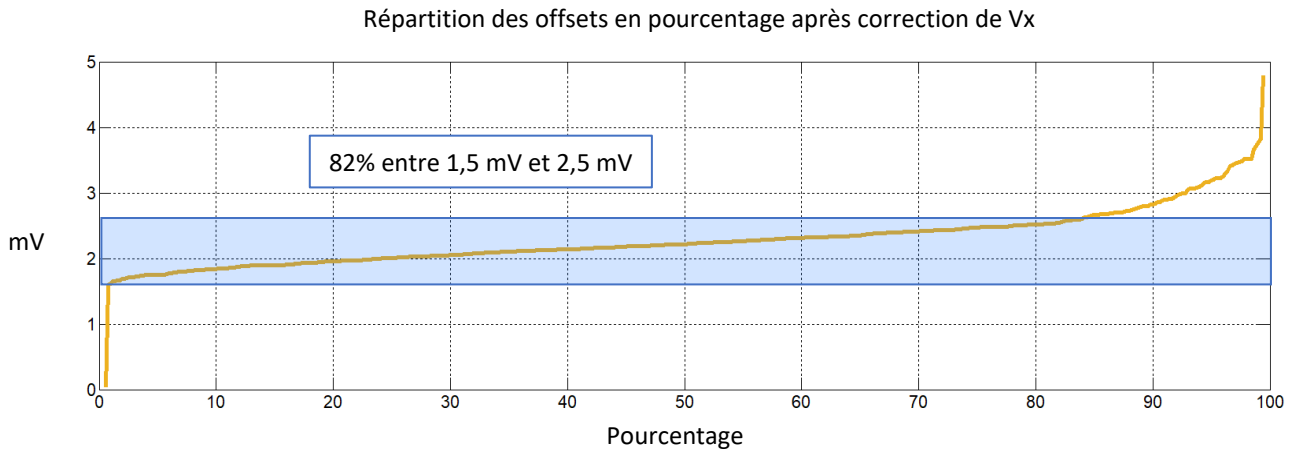


Figure 4.26 : Répartition des offsets du convoyeur de courant sur l'entrée X par rapport à la tension de commande Vmid = 1,1V, après correction sur une simulation Monte-Carlo avec un total de 500 échantillons.

4.6 Consommation des circuits

Le projet ULPEC a pour objectif de créer un réseau de neurones sur puce ultra basse consommation. Nous sommes qu'aux prémices de ces travaux, le but de cette thèse est avant tout de montrer la faisabilité de cette tâche. La consommation de ce réseau pourra être améliorée à l'avenir.

La consommation au repos d'un neurone pré-synaptique est de 243µA, celle d'un neurone post-synaptique est de 110µA. Avec 784 neurones pré-synaptiques et 100 neurones post-synaptiques sous une alimentation de 3,3V, la puissance totale au repos est de 0,66W. L'énergie d'un signal « up », « down » ou post-synaptique est relativement faible et vaux respectivement 472pJ, 8,81nJ et 6,7nJ, pour une résistance équivalente du réseau de neurones valant 10kΩ et étant connectée à Vmid du côté des neurones post-synaptiques. Si l'on prenait uniquement l'énergie nécessaire pour générer un signal dont tout le courant irait uniquement dans la résistance, on aurait : E_{pulse_up} = 182pJ E_{pulse_down} = 3,3nJ, E_{pulse_post} = 841pJ.

Les neurones pré-synaptiques étant largement surdimensionnés, l'électronique des neurones n'a pas été conçue dans le but premier d'être basse consommation, mais avant tous de valider le principe de fonctionnement des neurones évènementiels avec des synapses memristives sur puce. À titre de comparaison voici un tableau concernant l'énergie nécessaire pour générer un évènement en fonction de ces différentes plateformes neuromorphique [XINJIANG Z. 2018] (voir Tableau 4.2). L'énergie dépensée par notre réseau de neurones pour un évènement est de l'ordre de grandeur des plateformes SpiNNaker et Darwin.

Tableau 4.2 : Tableau montrant l'énergie dépensée par évènement en fonction des différentes plateformes

Platform	Neurogrid	brainScaleS	TrueNorth	SpiNNaker	Darwin	Cerveaux humains
Énergie par évènement	100pJ	100pJ	25pJ	10nJ	10nJ	10fJ

4.7 Conclusion

Bien que chacune des parties ait été simulée pour vérifier leur bon comportement de nombreuse fois, il était nécessaire de vérifier notamment aux interfaces des sous-systèmes si tout était connecté comme il se doit. La puce a été simulée en entier par Prophesee pour valider le fonctionnement du système en entier avant d'envoyer la puce en fonderie. La simulation permet de vérifier l'enchainement du déclenchement du pixel de la caméra, décodage de l'adresse par le DCB, déclenchement des neurones pré-synaptiques correspondant qui entraine une charge des neurones post-synaptiques en fonction des poids synaptiques. Elle permet de vérifier aussi, qu'une fois qu'un neurone post-synaptique a déclenché, que les commandes envoyées aux différents neurones par le contrôleur numérique se déroulent correctement.

Nous avons vu dans ce chapitre les contraintes physiques liées au crossbar et à la caméra, les stratégies d'apprentissages implémentées et les résultats de simulation de ces stratégies. Les problèmes que l'on rencontre sur les crossbars de memristors de grandes tailles ne sont pas rédhibitoires et peuvent être atténués. Pour pouvoir répondre aux contraintes de la caméra et gérer nos stratégies d'apprentissage, nous avons dû concevoir un contrôleur numérique. Les résultats de simulations montrent des résultats probants sur notre réseau avec les stratégies d'apprentissage implantées. Les simulations avec les variations physiques montrent l'importance de corriger l'offset tandis que les variations portant sur le temps de charges des neurones n'ont que très peu d'influence.

Hermes sera prochainement testé conjointement par une équipe de Prophesee et de l'IMS. On testera l'apprentissage de son réseau de neurones en utilisant la caméra évènementielle intégrée ou en utilisant un FPGA pour contrôler le déclenchement individuel de chaque neurone pré-synaptique. Prophesee maitrise la conception et le teste de leur caméra, il revendra donc à eux de concevoir la carte permettant le test de Hermes.

CONCLUSION & PERSPECTIVE

Nous avons aujourd'hui un problème de consommation énergétique au niveau mondial concernant les appareils électroniques et de l'IA en particulier. Avec l'avènement des réseaux de neurones et de l'intelligence artificielle, il est urgent d'augmenter les performances énergétiques de ces systèmes. L'ordinateur avec sa classique architecture de von Neumann n'est pas efficace en termes de consommation énergétique et de performance pour effectuer des calculs sur des réseaux de neurones. En s'affranchissant de cette architecture, de nombreuses recherches ont été faites dans le but de réduire la dépense énergétique ou d'augmenter la performance. Les réseaux de neurones à bases de memristors sont une voie de recherche prometteuse parmi d'autres. Cette thèse a permis de mettre en avant le potentiel de cette technologie mais également les problèmes afférents. Ces recherches avec l'étude des réseaux de neurones sur crossbar de memristors de grande taille serviront de point de départ à une étude plus approfondie de ce type de réseaux.

Synthèse

Les memristors que nous utilisons sont de type ferroélectrique. Ils ont la particularité d'avoir une grande résistance à l'état passant, de l'ordre du mégohm, et une résistance à l'état fermé de l'ordre de la centaine de mégohm. Pour des réseaux de neurones dont le but est la basse consommation, de grandes résistances sont un avantage. De plus, les tensions qui permettent les transitions sont de l'ordre du volt, ce qui est compatible avec les tensions mises en jeu dans les circuits intégrés. Les memristors ferroélectriques sont donc de sérieux candidats au développement de solutions alternatives pour le calcul par réseau de neurones.

Les neurones pré et post-synaptiques permettent de contrôler l'apprentissage en modifiant le poids de leurs synapses après avoir récupéré la valeur des memristors via la valeur des courants injectés dans les capacités de membrane des neurones post-synaptiques. De nombreux problèmes ont été soulevés lors de la conception de l'électronique pour le contrôle du crossbar de grande taille. Ils ont été en partie résolus. Nous devions nous assurer qu'il n'y ait pas de courant chargeant en continu la capacité de membrane d'un neurone post-synaptique, ce qui entrainerait des déclenchements intempestifs du neurone post-synaptique. L'offset des convoyeurs de courant sur ce type de crossbar s'il n'est pas corrigé, peut empêcher l'apprentissage du réseau. Les mesures réalisées sur la puce Orion montrent que notre électronique a les performances attendues et résolvent les problèmes soulevés. Cependant la copie en courant n'était pas celle espérée. C'est un défaut mineur qui a été corrigé dans la puce Hermes.

Les contraintes permettant d'assembler le crossbar la caméra et le réseau de neurones, mais également de réaliser l'apprentissage sont important sur cette large puce. Les résistances parasites d'accès aux memristors sur le crossbar bien que petites peuvent potentiellement poser des problèmes de chutes de tension et donc d'apprentissage, plus le crossbar est large plus les chutes de tension seront importantes. Dans notre cas, ces chutes de tension ne sont pas si importantes et peuvent être compensées. Un contrôleur numérique a été conçu pour traiter les évènements venant de la caméra de manière asynchrone et permet de déclencher les neurones correspondants. Il permet également d'implanter et de choisir les stratégies d'apprentissage que nous souhaitons. Les stratégies d'apprentissage que nous avons implantées ont été testées par un simulateur reproduisant le comportement de notre convoyeur de courant. Les résultats que nous avons obtenus montrent un apprentissage correct par rapport à l'état de l'art pour un petit réseau de neurones de 100 neurones de sortie sans couche cachée. La consommation de notre réseau a l'heure actuelle n'est pas optimale mais l'énergie nécessaire pour générer un évènement est comparable aux plateformes SpiNNaker ou Darwin.

Perspective

Les memristors ferroélectriques avec leur grande résistance à l'état passant et leur grand rapport ON/OFF, ont un réel potentiel dans les réseaux de neurones hardwares basse consommation. Des memristors plus résistifs pourront encore améliorer la consommation et diminuer l'effet des résistances parasites des crossbars. Cependant l'intégration de ce type de memristor est encore difficile, il faudrait pouvoir en intégrer davantage pour pouvoir faire un réseau de neurones ayant suffisamment de synapses pour des applications plus complexes.

Les réseaux de neurone hardware seront limités par la taille de leur crossbar. Un crossbar trop grand aura pour conséquence que les résistances d'accès auront un fort impact sur la chute de tension des memristors. Il faudra également porter un soin particulier à l'offset des convoyeurs de courant. Une autre configuration des synapses pourrait être envisagée en utilisant plusieurs petits crossbars afin d'éviter la connexion all-to-all.

Concernant la consommation énergétique des circuits électroniques, cette puce est un prototype où les memristors ont changé de caractéristique au court de la conception plusieurs fois, nous avons mis notre priorité sur la fonctionnalité du système. À cause des changements de caractéristiques des memristors, les neurones pré et post-synaptiques sont surdimensionnés par rapport au besoin actuel, de ce fait la consommation statique de notre puce est importante et pourra être fortement réduite.

Depuis l'invention du memristor en 2008, il est aujourd'hui possible de faire des crossbars d'une dizaine de milliers de memristors. Il reste encore un grand défi technologique à l'intégration sur substrat de silicium et de pouvoir les implanter au plus près des transistors afin qu'ils soient utilisés dans les réseaux

de neurones de manière optimale. Pour des applications diverses, il sera nécessaire d'avoir une grande quantité de synapses afin que le réseau soit capable d'apprendre une grande variété de tâches différentes. Si l'on souhaite une émergence de ces réseaux de neurones, il sera nécessaire d'augmenter l'intégrabilité des memristors afin que l'on augmente le nombre de synapses et de se passer des crossbars pour éviter toutes les contraintes qui y sont liées.

La puce de test Orion nous a permis de confirmer les performances de l'électronique développée durant ces travaux de thèse. La seconde puce Hermes, celle contenant l'ensemble du réseau de neurones, est en cours de fabrication et sera testée courant juin 2021. Ces mesures permettront de confirmer la possibilité de réaliser des réseaux de neurones évènementiels à base de memristor, ou éventuellement de révéler les problèmes à résoudre et les défis à relever.

ANNEXE: MODELE VERILOG A DU MEMRISTOR FERROELECTRIQUE

```
//////////////////////////////////////////////////////////////////////
//
//    Memristor Model
//
//    Author           : Charly Meyer
//                       IMS / Université de Bordeaux
//    Creation   : 22/02/18
//
//
//    Descritpion :   Equations to determine resistance of memistor (R)
from domain (S) :
//                          R=1/( S*( 1/Roff-1/Ron ) + 1/Ron )
//                          S=(1/R-1/Ron)/(1/Roff -1/Ron)
//
//    Version    : V3.3
//
//
//
//////////////////////////////////////////////////////////////////////

`include "constants.vams"
`include "disciplines.vams"

// model of memristior in Verilog A

module Memristor(n,p,R,Sout);
      inout n;   // n negative input
      inout p;   // p positive input
      output R,Sout;
      electrical n,p,R,Sout;
      parameter real PI=3.14159265;
      parameter real timStep=10n; //10ns
      parameter integer para_noise=1; // 1 with noise // 0 without noise
```

```
      // Pamemeter to make Monte Carlo simulation //
      (*cds_inherited_parameter*) parameter real mc_seed = 0;
      localparam integer SEED= (1+mc_seed)*1000000;

      integer seed; // seed need to have integer number
      parameter real Roff_moy=50e6;    // Rmax 50Mega ohm
      parameter real Ron_moy=300e3;    // Rmin = 300k ohm
      real R_init;          // inital value of Resistance
      parameter real S_init=0.1;
      real Roff;
      real Ron;
      real S;
      real S1;
      real oldS;
      real Rmem ;
      real oldR;
      real Rplus;
      real R_interim;
      real R_high_init;
      real Ron_eq;

//  temporary variable

      real a0;
      real z1 ;
      real Gamma;  // gamma, z1 needed to calculate Rmem variation
      real t0;
      real z0;
      real Vr;// V(p,n)
      real Vsens;// Vsens=-1 si vd négatif =1 sinon
      real init; // pour l'initialisation des resistance
      integer cpt;
      integer sd;
      integer test;
      integer ok; // valide la distribution
      integer val_Ron;
      integer val_origin;
```

```
      real Ron_norm[1:13];
      real Ron_val[1:13];
      real Roff_norm[1:18];
      real Roff_val[1:18];
      real Rcalc;
      real rnd; real xrnd;
      real Smem; //

analog
      begin
      if (init < 1)
            begin
                  seed=SEED;
                  val_Ron=1;
                  a0=1;
                  //R_init=9e6+abs($random(seed)%6000)*1e3; //
                  //S=(1/R_init-1/Ron)/(1/Roff-1/Ron);
                  S=S_init;
                  oldS=S;
                  oldR=1/( oldS*( 1/Roff-1/Ron ) + 1/Ron);
                  Vsens=1;
                  init=1;
                  cpt=1;
                  rnd=$rdist_normal(seed,0,1);
                  Ron_val[1]=89.12k;          // minimal value of Ron
                  Ron_val[13]=1.05e6;         // maximal value of Ron




            ////////////////     Ron chosen       /////////////////

            ok=0;
            while ((ok==0))
                  begin
                  rnd= $rdist_uniform(seed,0,1);
                  if (rnd<0.65)
                        xrnd= $rdist_normal(seed,4.1,1.7);
```

```
//$rdist_normal(seed,3.29,2.05)            else
                    xrnd= $rdist_normal(seed,7.77,2);
//$rdist_normal(seed,6.77,2.18)


                Ron=Ron_val[1]*exp(abs(xrnd)*ln(1.25));
                    rnd= $rdist_uniform(seed,0,1);

                if ( ((xrnd<0)||(xrnd>12)) )
                    ok=0;
                else
                    ok=1;

            end

            //////////////    Ron     /////////////////

            rnd=$rdist_normal(seed,0,1);

            Roff_val[1]=3.55e6;                  // minimal value of Roff

            /////////////// Roff chosen  /////////////////

            ok=0;
            while ((ok==0))
                begin
                rnd= $rdist_uniform(seed,0,1);
                if (rnd<0.6)
                    xrnd= $rdist_normal(seed,9.83,0.55);
                else
                    xrnd= $rdist_normal(seed,10.2,4);

                Roff=Roff_val[1]*exp(abs(xrnd)*ln(1.259));
                rnd= $rdist_uniform(seed,0,1);

                if ( ((xrnd<0)||(xrnd>17)) )
                    ok=0;
                else
                    ok=1;

            end

            ////////////----       Roff      ------/////////////////
```

```
      //Ron=400e3; // Ron  forcing
      //Roff=100e6;// Roff forcing

      Rmem=oldR + (1/( S*( 1/Roff-1/Ron ) + 1/Ron) - oldR) ;
      R_interim=Rmem;
      Ron_eq=Ron;
      //R_init=500e3+abs($random(seed)%150)*1e3; //
      S=(1/R_init-1/Ron)/(1/Roff-1/Ron);
      S1=S;

    end

      //////////    end initialisation ////////////////////

    Vr = V(p,n);
    I(p,n) <+  V(p,n)/Rmem ;
    V(R)   <+  Rmem;
    V(Sout)<+  1;
    oldS=S;
    test=0;

///////////////* Noise *//////////////////
//
//   Noise on Ron to be close at the mesurment
//
////////////////////////////////////////

      if (para_noise==1)
           begin
           Rcalc=1/( S*( 1/Roff-1/Ron )+1/Ron);

           if ((Vr<-0.5))
                begin
                if (val_Ron==1)
                     begin
                     val_Ron=0;
                     val_origin=1;
                     R_high_init=Rmem;
                     Ron_eq=0;
                     if (Rmem<2e6)
                          Ron_eq=Ron;
```

```
                    while((Ron_eq<Ron))
                         begin
                         Ron_eq=Ron*(1+R_high_init/12e6) +
(R_high_init/45) *$rdist_normal(seed,0,1);
                    end
                    S=( 1/Rmem - 1/Ron_eq )/( 1/Roff-1/Ron_eq ) ;
                    oldS= ( 1/Rmem - 1/Ron_eq )/( 1/Roff-1/Ron_eq ) ;

               end
          end
          if(Vr>-0.5)
          val_Ron=1;
          if (Vr>0 & val_origin==1)
               begin
               val_origin=0;
               val_Ron=1;
               R_interim=1/( S*( 1/Roff-1/Ron_eq ) + 1/Ron_eq);
               S=(1/R_interim-1/Ron)/(1/Roff-1/Ron);
               oldS=S;
               S1=(1/Rmem-1/Ron)/(1/Roff-1/Ron);
               Ron_eq=Ron;
          end
     end
///////////////* Noise */////////////////
////////////////////////////////////////////////////////////////////////////
////////
//////////////////////*   timer each timeStep
*////////////////////////////////
    @(timer(timStep*cpt) )
     begin
     if (Vr<-2.174) // Vr<-2.174 !
```

```
          S=0;
     else
          begin  //Vr>-2.174 !
          if (abs(Vr)<0.1)//0
               S=oldS;
          else //0
               begin
               if (Vr>0 ) //1  case 'm31-32'
                    begin
                    z1 = -27.55*Vr*Vr*Vr+109.8*Vr*Vr-151.75*Vr+64.5 +
1.548* $rdist_normal(seed,0,1)*para_noise;
                    Gamma       = 7.75*exp(-1.825*Vr);

               end

               else
                    begin
                    Gamma =  0.0021*exp(0.74*Vr)-0.42e-3;
                    z1  =  123.397435896761e+000*pow(Vr,5)
                              +854.530885776750e+000*pow(Vr,4)
                                   +2.35718968530472e+003*pow(Vr,3)
                                        +3.24204676572246e3*pow(Vr,2)
                                             +2.22773645104209e3*Vr

     +609.000203961136e+000
          +3.5e-6*pow($rdist_normal(seed,0,1),1)*para_noise;

               end

               if (((S<0.001)*(Vr>0)) + ((S>1-0.001)*(Vr<0))) //start 2
                    t0=0;

               else if (((S<0.001) * (Vr<0)) + ((S>1-0.001) * (Vr>0)))
                                   t0 = exp(z1+10*Gamma);

               else if (Vr>=0)
                    t0 = pow( 10,(Gamma * tan( PI * (oldS - 0.5)) + z1)
);
```

```
                else
                      t0 = pow( 10,(Gamma * tan(-PI * (oldS - 0.5)) + z1)
);

                z0=log(t0+timStep);

                if (abs(z0)>35)// infinite if z0>37 over
                      S=oldS;

                else
                      begin // non infinite
                      if (Vr<0) //3
                            begin
                            S=1-1/PI*(PI/2-atan((z1-z0)/Gamma)) ;
                            end
                      else
                            S=1/PI*(PI/2-atan((z1-z0)/Gamma)) ;


                end //infinit

                      //variation

                S=S*1;
                if ( (oldS> S && Vr>0) || (oldS< S && Vr<0) )
                      S=oldS;//S=oldS;

                if (S<0)
                      S=0;

                if(S>0.9999)
                      S=0.9999;

          end// else 0

          cpt=cpt+1;

     end //          // S is finiched to be calculated

     Rmem = 1/( S*( 1/Roff-1/Ron_eq ) + 1/Ron_eq);

     end

     ////////////  end timer  /////////////////
```

```
    if (Vr<0)
      Vsens=-1;
    else
      Vsens=1;
end  // end analog
endmodule
```

REFERENCES BIBLIOGRAPHIQUES

[AAMIR S. A. 2018] AAMIR S. A., STRADMANN Y., MÜLLER P., PEHLE C., HARTEL A., GRÜBL A., SCHEMMEL J. MIER K. (2018) An Accelerated LIF Neuronal Network Array for a Large-Scale Mixed-Signal Neuromorphic Architecture. IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 65, no 12, p. 4299-4312.

[ABBOTT L. 1999] ABBOTT L. (1999) Lapicque's introduction of the integrate-and-fire model neuron Brain research bulletin, vol. 50, no 5-6, p. 303-304.

[AFSHAR S. 2014] AFSHAR S., GEORGE L., TAPSON J., VAN SCHAIK A., HAMILTON T. J. (2014) Racing to learn: statistical inference and learning in a single spiking neuron with adaptive kernels. Frontiers in neuroscience, vol. 8, p. 377.

[AKOPYAN F. 2015] AKOPYAN F., SAWADA J., CASSIDY A., ALVAREZ-ICAZA R., ARTHUR J., MEROLLA P., IMAM N., NAKAMURA Y., DATTA P., NAM G. J., BRIAN TABA B., BEAKES M., BREZZO B., KUANG J. B., MANOHAR R., RISK W. P., JACKSON B., DMODHA D. S. (2015) TrueNorth: Design and Tool Flow of a 65 mW 1 Million Neuron Programmable Neurosynaptic Chip IEEE transactions on computer-aided design of integrated circuits and systems, vol. 34, no 10, p. 1537-1557.

[AMIR A. 2017] AMIR A., TABA B., BERG D., MELANO T., MCKINSTRY J., DI NOLFO C., NAYAK T., ANDREOPOULOS A., GARREAU G., MENDOZA M., KUSNITZ J., DEBOLE M., ESSER S., DELBRUCK T., FLICKNER M., MODHA D. (2017) A Low Power, Fully Event-Based Gesture Recognition System. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. p. 7243-7252.

[ARBET D. 2012] ARBET D., NAGY G., GYEPES G., STOPJAKOVÁ V. (2012) Design of Rail-to-Rail Operational Amplifier with Offset Cancelation in 90nm technology. In: 2012 International Conference on Applied Electronics. IEEE, 2012. p. 17-20.

[BADOUAL M. 2006] BADOUAL M., ZOU Q., DAVISON A. P., RUDOLPH M., BAL T., FREGNAC Y., DESTEXHE A. (2006) Biophysical and phenomenological models of multiple spike interactions in spike-timing dependent plasticity. International journal of neural systems, vol. 16, no 02, p. 79-97.

[BI G. Q. 1997] BI G.-Q., POO M.-M. (1997) Synaptic Modifications in Cultured Hippocampal Neurons: Dependence on Spike Timing, Synaptic. Journal of neuroscience, vol. 18, no 24, p. 10464-10472.

[BILL J. 2014] BILL J., LEGENSTEIN R. (2014) A compound memristive synapse model for statistical learning through STDP in spiking neural networks. Frontiers in neuroscience, vol. 8, p. 412.

[BKER J. R. 2010] BKER J. R. (2010) CMOD circuit design, layout, and simulation third edition. John Wiley & Sons.

[BOHR M. T. 2017] BOHR M. T., YOUNG I. A. (2017) CMOS scaling trends and beyond. IEEE Micro, vol. 37, no 6, p. 20-29

[BOUVIER M. 2019] M. BOUVIER, A. VALENTIAN, T. MESQUIDA, F. RUMMENS, M. REYBOZ, E. VIANELLO, E. BEIGNE, (2019) Spiking Neural Networks Hardware Implementations and Challenges: vol. 15, no 2, p. 1-35.

[BOYN S. 2016] BOYN S. (2016). Ferroelectric tunnel junctions: memristors for neuromorphic computing. (Doctoral dissertation, Université Paris-Saclay (ComUE)).

[BOYN S. 2017] BOYN S., GROLLIER J., LECERF G., XU B., LOCATELLI N., FUSIL S., GIROD S., CARRETERO C., GARCIA K., XAVIER S., TOMAS J., BELLAICHE L., BIBES M., BARTHELEMY A., SAIGHI S., GARCIA V. (2017) Learning through ferroelectric domain dynamics in solid-state synapses. Nature communications, vol. 8, no 1, p. 1-7

[BUENO J. 2018] BUENO J., MAKTOOBI S., FROEHLY L., FISCHER I., JACQUOT M., LARGER L., BRUNNER D. (2018). Reinforcement learning in a large-scale photonic recurrent neural network. Optica, vol. 5, no 6, p. 756-760

[BURR G. 2017] BURR G. W., SHELBY R. M., SEBASTIAN A., KIM S., KIM S., SIDLER S., VIRWANI K., ISHII M., NARAYANAN P., FUMAROLA A., SANCHES L. L., BOYBAT I., LE GALLO M., MOON K., WOO J., HWANG H., LEBLEBICI Y. (2017) Neuromorphic computing using non-volatile memory. Advances in Physics: X, vol. 2, no 1, p. 89-124.

[CAPRA M. 2019] CAPRA M., PELOSO R., MASERA G., RUO ROCH M., MARTINA M. (2019) Edge Computing: A Survey On the Hardware Requirements in the Internet of Things World. Future Internet, vol. 11, no 4, p. 100.

[CHAMBOULA A. 2012] CHAMBOULA A, GARCIA V., CHERIFI R. O., BOUZEHOUANE K., FUSIL S., MOYA X., XAVIER S., YAMADA H., DERANLOT C., MATHUR N. D., BIBES M., BARTHELEMY A., GROLLIER J. A ferroelectric memristor. Nature materials, vol. 11, no 10, p. 860-864.

[CHANDRA P. 2007] CHANDRA P. & LITTLEWOOD P.B. 2007. A Landau primer for ferroelectrics. Physics of ferroelectrics, p. 69-116.

[CHI P. 2016] CHI P., LI S., CHENG Y., LU Y., KANG S. H., XIE Y. (2016) Architecture Design with STT-RAM: Opportunities and Challenges. 21st Asia and South Pacific Design Automation Conference (ASP-DAC). IEEE. p. 109-114.

[CHUA L. 1971] CHUA L. O. (1971) Memristor-The Missing Circuit Element, Transactions on circuit theory, vol. 18, no 5, p. 507-519.

[COHEN R. 1992] COHEN R. E., KRAKAUER H., (1992) Electronic structure studies of the differences in ferroelectric behaviour of BaTio3 and PbTiO3 Ferroelectrics, vol. 136, no 1, p. 65-83.

[CRUZ-ALBRECHT J. M. 2012] J. M. CRUZ-ALBRECHT, M. W. YUNG, N. SRINIVASA (2012) Energy-efficient neuron, synapse and STDP integrated circuits. IEEE transactions on biomedical circuits and systems, vol. 6, no 3, p. 246-256.

[DAWBER M. 2005] DAWBER M., RABE K. M., SCOTT J. F. (2005). Physics of thin-film ferroelectric oxides Reviews of modern physics, vol. 77, no 4, p. 1083.

[DOO S. 2016] DOO S. J., KYUNG M. K., SUNGHO K., BYUNG J. C., CHEOL S. H. (2016). Memristors for Energy-Efficient New Computing Paradigms. Advanced Electronic Materials, vol. 2, no 9, p. 1600090.

[ESHRATIFAR A. E 2018] ESHRATIFAR A. E., PEDRAM M. (2018) Energy and performance efficient computation offloading for deep neural networks in a mobile cloud computing environment. Proceedings of the 2018 on Great Lakes Symposium on VLSI. p. 111-116.

[FERRE P. 2018] FERRÉ P., MAMALET F., THORPE S. J. (2018) Unsupervised feature learning with winner-takes-all based STDP. Frontiers in computational neuroscience, vol. 12, p. 24.

[FERRI G. 2004] FERRI G., GUERRINI C. N. (2004) Low-voltage low-power CMOS current conveyor. Springer Science & Business Media.

[FURBER S. B. 2014] FURBER S. B., GALLUPPI F., TEMPLE S., PLANA L. A. (2014) The SpiNNaker Project. Proceedings of the IEEE, vol. 102, no 5, p. 652-665.

[GALLEGO G. 2019] GALLEGO G., DELBR¨UCK T., ORCHARD G., BARTOLOZZI C., TABA B., CENSI, STEFAN LEUTENEGGER A., DAVISON A., CONRADT J., DANIILIDIS K., SCARAMUZZA D. (2019). Event-based Vision: A Survey. arXiv preprint arXiv:1904.08405, 2019.

[GARCIA V. 2009] GARCIA V., FUSIL S., BOUZEHOUANE K., ENOUZ-VEDRENNE S., MATHUR N. D., BARTHÉLÉMY A., BIBES M. (2009) Giant tunnel electroresistance for non-destructive readout of ferroelectric states. Nature, vol. 460, no 7251, p. 81-84.

[GARCIA V. 2014] GARCIA V., BIBES M. (2014) Ferroelectric tunnel junctions for information storage and processing. Nature communications, vol. 5, no 1, p. 1-12.

[GOPALAKRISHNAN R. 2015] GOPALAKRISHNAN R., BASU A., (2015) Triplet Spike Time-Dependent Plasticity in a Floating-Gate Synapse. IEEE transactions on neural networks and learning systems, vol. 28, no 4, p. 778-790.

[GUIFARD B. 1999] GUIFARD B. (1999) Elaboration et caractérisation de céramique ferroélectrique de type PZT floué. Thèse de doctorat. Lyon, INSA.

[GUPTA A. 2009] GUPTA A., LYLE N. Long (2009) Hebbian learning with winner take all for spiking neural networks. International Joint Conference on Neural Networks. IEEE. p. 1054-1060.

[HALAWANI Y. 2015] HALAWANI Y., MOHAMMAD B., HOMOUZ D., AL-QUTAYRI M., SALEH H. (2015) Modeling and Optimization of Memristor and STT-RAM-Based Memory for Low-Power Applications. I EEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 24, no 3, p. 1003-1014.

[HAO L. 2017] HAO L., ZHANG Z., LIU Z. (2016) Application of Artificial Neural Networks for Catalysis: A Review. Catalysts, vol. 7, no 10, p. 306.

[HODGKIN L. 1952] HODGKIN L., A. F. HUXLEY (1952) A quantitative description of membrane current and its application to conduction and excitation in nerve. The Journal of physiology, vol. 117, no 4, p. 500-544.

[HUIJSING J. 2017] HUIJSING J. (2017) Operational amplifiers theory and design third edition Springer Netherlands

[JONES N. 2018] JONES N. (2018) How to stop data centres from gobbling up the world's electricity. Nature, vol. 561, no 7722, p. 163-167.

[KANG Y. 2017] KANG Y., HAUSWALD J., GAO C., ROVINSKI A., MUDGE T., JASON MARS J., TANG L. (2017) Neurosurgeon: Collaborative Intelligence Between the Cloud and Mobile Edge. ACM SIGARCH Computer Architecture News, vol. 45, no 1, p. 615-629.

[KAPPEL D. 2014] DAVID KAPPEL, BERNHARD NESSLER, WOLFGANG MAASS (2014) STDP Installs in winner-take-all circuits an online approximation to hidden markov model learning. PLoS Comput Biol, vol. 10, no 3, p. e1003511.

[KASABOV, N. 2019] KASABOV N. K. (2019) Time-space, spiking neural networks and brain-inspired artificial intelligence. Berlin, Heidelberg: Springer Berlin Heidelberg,

[KO U. 2000] KO U., BALSARA P. T. (2000) High-Performance Energy-Efficient D-Flip-Flop Circuits. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 8, no 1, p. 94-98.

[KORNIJCUK V. 2016] KORNIJCUK V., LIM H., SEOK J. Y., GUHYUN KIM, KIM S. K., KIM I., CHOI B. J., JEONG D. S. (2016) Leaky integrate-and-fire neuron circuit based on floating-gate integrator. Frontiers in neuroscience, 2016, vol. 10, p. 212.

[KRESTINSKAYA O. 2019] KRESTINSKAYA O., PAPPACHEN JAMES A., ONG CHUA L. (2017) Neuromemristive Circuits for Edge Computing: A Review. IEEE transactions on neural networks and learning systems, vol. 31, no 1, p. 4-23.

[KURASHINA T. 1998] KURASHINA T., OGAWA S., WATANABE K. (1998) A High Performance Class AB Current Conveyor. EEE International Conference on Electronics, Circuits and Systems. Surfing the Waves of Science and Technology (Cat. No. 98EX196). IEEE. p. 143-146.

[LECERF G. 2015.] LECERF G. (2015) Développement d'un réseau de neurones impulsionnels sur silicium à synapses memristives. Thèse de doctorat. Bordeaux.

[LECUN Y. 1998] LECUN Y., BOTTOU L., BENGIO Y., HAFFNER P. (1998) Gradient-based learning applied to document recognition. Proceedings of the IEEE, vol. 86, no 11, p. 2278-2324.

[LEWDEN P. 2019] LEWDEN P., VINCENT A. F., MEYER C., TOMAS J., SIAMI S., SAIGHI S. (2019) Hardware Spiking Neural Networks: Slow Tasks Resilient Learning with Longer Term-Memory Bits. IEEE Biomedical Circuits and Systems Conference (BioCAS) IEEE. p. 1-4.

[LEWDEN P. 2020] LEWDEN P., VINCENT A. F., MEYER C., TOMAS J., SAIGHI S. (2020) Toward Hardware Spiking Neural Networks with Mixed-Signal Event-Based Learning Rules. International Joint Conference on Neural Networks (IJCNN). IEEE. p. 1-8.

[MARTIN L. 2016] MARTIN L. W., RAPPE A. M. (2016) Thin-film ferroelectric materials and their applications. Nature Reviews Materials, 2016, vol. 2, no 2, p. 1-14.

[MCCULLOCH W. S 1943] [MCCULLOCH W. S, PITTS W. (1943) logical calculus of the ideas immanent in nervous actv ivity. The bulletin of mathematical biophysics, vol. 5, no 4, p. 115-133.

[MEROLLA P. A. 2014] MEROLLA P. A., ARTHUR J. V., ALVAREZ-ICAZA R., CASSIDY A. S., SAWADA J., AKOPYAN F., JACKSON B. L., IMAM B., GUO C., NAKAMURA Y., BREZZO B., VO I., ESSER S. K., APPUSWAMY, BRIAN R., AMIR A., FLICKNER M. D., RISK W. P., MANOHAR R., MODHA D. S. (2014) A million spiking-neuron integrated circuit with a scalable communication network and interface Science, vol. 345, no 6197, p. 668-673.

[MEYER C. 2018] MEYER C., CHANTHBOUALA A., BOYN S., TOMAS J., GARCIA V., BIBES M., FUSIL S., GROLLIER J., SAÏGHI S. (2018) Verilog-A model of ferroelectric memristors dedicated to neuromorphic designers. 25th IEEE International Conference on Electronics, Circuits and Systems (ICECS). IEEE. p. 557-560.

[MOHAN C. 2017] MOHAN C., CAMUÑAS-MESA L.A., VIANELLO E., PERNIOLA L., REITA C., DE LA ROSA J.M., SERRANO-GOTARREDONA T., LINARES-BARRANCO B. (2017) Bulk-based DC offset calibration for

low-power memristor array read-out system. 32nd Conference on Design of Circuits and Integrated Systems (DCIS). IEEE. p. 1-5.

[MOHMOODI D. 2018] MAHMOODI, M. R. STRUKOV D. (2018) Breaking POps/J barrier with analog multiplier circuits based on nonvolatile memories. In: *Proceedings of the International Symposium on Low Power Electronics and Design*. p. 1-6.

[PELAEZ E. 1990] PELAEZ E. (1990) Parallelism and the Crisis of von Neumann Computing. Technology in Society, vol. 12, no 1, p. 65-77.

[PFEIFFER M. 2018] PFEIFFER M., FFEIL T. (2018) Deep Learning With spiking neurons opportunities and challenges. Frontiers in neuroscience, vol. 12, p. 774.

[PIERANGELO C. 1985] PIERANGELO C., CAPONAGO G., MARTIGNONE R., (1985) Differential amplifier circuit with common mode output voltage. U.S. Patent No 6,940,348, 6 sept.

[POSCH C. 2014] POSCH C., SERRANO-GOTARREDONA T., LINARES-BARRANCO B., DELBRUCK T., (2014) Retinomorphic Event-Based Vision Sensors: Bioinspired CamerasWith Spiking Output. Proceedings of the IEEE, vol. 102, no 10, p. 1470-1484.

[PRIETO A. 2016] PRIETO A., PRIETO B., MARTINEZ ORTIGOSA E., ROS E., PELAYO F., ORTEGA J., ROJAS I. (2016) Neura lnetworks: An overview of early research, current frameworks and new challenges. Neurocomputing, vol. 214, p. 242-268.

[PURVE D. 2014] PURVES D., AUGUSTINE G. J., FITZPATRICK D., HALL W. C., LAMANTIA A.-S., WHITE L.E. (2014) Neurociance. De Boeck, Sinauer, Sunderland, Mass, 2014, vol. 15, p. 16.

[REBECQ H. 2019] REBECQ, H. RANFTL R., KOLTUN V., SCARAMUZZA D. (2019) High Speed and High Dynamic Range Video with an Event Camera. IEEE transactions on pattern analysis and machine intelligence.

[SAKAYORI K. 1995] SAKAYORI K. I., MATSUI Y., ABE H., NAKAMURA E., KENMOKU M., HARA T., ISHIKAWA D., KOKUBU A., HIROTA K.I., IKEDA T. (1995) Curie Temperature of BaTiO3. Japanese journal of applied physics, vol. 34, no 9S, p. 5443.

[SCHRAUWEN B. 2004] SCHRAUWEN B., VAN CAMPENHOUT J. (2004) Extending spikeprop. IEEE International Joint Conference on Neural Networks (IEEE Cat. No. 04CH37541). IEEE. p. 471-475.

[SCHUMAN C. D. 2017] SCHUMAN C. D., POTOK T. E., PATTON R. M., BIRDWELL J. D., DEAN M. E., GARRETT S. R., PLANK J. S. (2017) A Survey of Neuromorphic Computing and Neural Networks in Hardware. *arXiv preprint arXiv:1705.06963*.

[SCHWARTZ E. L. 1997] SCHWARTZ E. L. (1997) Computing with the Leaky Integrate-and-Fire Neuron: Logarithmic Computation and Multiplication. Neural computation, vol. 9, no 2, p. 305-318.

[SEDRA A. 1989] SEDRA A. (1989) The current conveyor: history and progress. IEEE International Symposium on Circuits and Systems. IEEE. p. 1567-1571.

[SEDRA A.S 1990] SEDRA A.S, ROBERTS G.W., GOOHH F. (1990) The current conveyor: history, progress and new results. IEE Proceedings G-Circuits, Devices and Systems, vol. 137, no 2, p. 78-87.

[SEIFFERT U. 2001] SEIFFERT U. (2001) Layer Perceptron Training Using Genetic Algorithms. ESANN. 2001. p. 159-164

[SERRANO-GOTARREDONA T. 2013] SERRANO-GOTARREDONA T., MASQUELIER T., PRODROMAKIS T., INDIVERI G., LINARES-BARRANCO B., (2013) STDP and STDP variations with memristors for spiking neuromorphic learning systems. Frontiers in neuroscience, vol. 7, p. 2.

[SHABBIR J. 2018] SHABBIR J., ANWER T. (2018) Artificial intelligence and its role in near future. arXiv preprint arXiv:1804.01396.

[SHAKIB S. 2017] SHAKIB S., PANDA P., ROY K. (2017) Gabor Filter assisted energy efficient fast learning convolutional neural networks. IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED). IEEE. p. 1-6.

[SHAMSI J. 2018] SHAMSI J. MOHAMMADI K., SHOKOUHI S. B. (2018) A Hardware Architecture for Columnar-Organized Memory Based on CMOS Neuron and Memristor Crossbar Arrays IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 26, no 12, p. 2795-2805.

[SHIBATA, K 1999] SHIBATA K., ITO K. (1999) Gauss-Sigmoid Neural Network. IEEE. p. 1203-1208.

[SOLTIC S. 2010] SOLTIC S., KASABOV N. (2010) Knowledge extraction from evolving spiking neural networks with rank order population coding. International Journal of Neural Systems, vol. 20, no 06, p. 437-445.

[STATHOPOULOS S. 2017] STATHOPOULOS S., ALI KHIAT A., MARIA TRAPATSELI M., SIMONE CORTESE S., ALEXANTROU SERB A., ILIA VALOV I., THEMIS PRODROMAKIS T. (2017) Multibit memory operation of metal-oxide bi-layer memristors. Scientific reports, vol. 7, no 1, p. 1-7.

[STROMATIAS E. 2017] STROMATIAS E., SOTO M., SERRANO-GOTARREDONA T., LINARES-BARRANCO B. (2017) An event-driven classifier for spiking neural networks fed with synthetic or dynamic vision sensor data. Frontiers in neuroscience, vol. 11, p. 350.

[TALATI N. 2020] TALATI N., BEN-HUR R., WALD N., HAJ-ALI A., REUBEN J., KVATINSKY S. (2020) A Real Processing-in-Memory Architecture to Combat the von Neumann Bottleneck. Applications of emerging memory technology. Springer, Singapore. p. 191-213

[TETZLA R. 2014] TETZLA R. (2014) Memristors and Memristive Systems. Springer Science & Business Media.

[TRUONG S. N. 2019] TRUONG S. N. (2019) Truong A Parasitic Resistance-Adapted Programming Scheme for Memristor Crossbar-Based Neuromorphic Computing Systems. Materials, vol. 12, no 24, p. 4097.

[TURING A. M. 1950] TURING A. M., (1950) "Computing machinery and intelligence," Mind, 1950, vol. 59, no 236, p. 433.

[UCHINO K. 2010] UCHINO K. (2010) Ferroelectric devices second. CRC press.

[VARGHESE B 2016] VARGHESE B., WANG N., BARBHUIYA S., KILPATRICK P., NIKOLOPOULOS D. S. (2016) Challenges and Opportunities in Edge Computing. IEEE International Conference on Smart Cloud (SmartCloud). IEEE. p. 20-26.

[VARKEY B. 2014] VARKEY B., GAO P., MCQUINN E., CHOUDHARY S., CHANDRASEKARAN A. R., BUSSAT J. M., ALVAREZ-ICAZA R., ARTHUR J. V., MEROLLA P. A., BOAHEN K. (2014) Neurogrid: AMixed-Analog-DigitalMultichip System for Large-Scale Neural Simulations. *Proceedings of the IEEE*, vol. 102, no 5, p. 699-716.

[VON KÜGELGEN J. 2017] VON KÜGELGEN, J. (2017) On Artificial Spiking Neural Networks: Principles, Limitations and Potential. *June*, vol. 18, p.

[WANG J. 2019] WANG J., ZHUGE F., (2019) Memristive Synapses for Brain-Inspired Computing. *Advanced Materials Technologies*, vol. 4, no 3, p. 1800544.

[WEIS J. 2020] WEIS J., PHILIPP SPILGER P., BILLAUDELLE S., STRADMANN Y., EMMEL A., MÜLLER E., BREITWIESER O., GRÜBL A., ILMBERGER J., KARASENKO V., KLEIDER M., MAUCH C., SCHREIBER K., SCHEMMEL J. (2020) Inference with Artificial Neural Networks on Analog Neuromorphic Hardware. IoT Streams for Data-Driven Predictive Maintenance and IoT, Edge, and Mobile for Embedded Machine Learning. Springer, Cham. p. 201-212.

[WIDROW B. 1990] WIDROW B., LEHR M. A. (1990) 30 Years of Adaptive Neural Networks: Perceptron, madaline, and backpropagation. Proceedings of the IEEE, 1990, vol. 78, no 9, p. 1415-1442.

[WU P.C 2013] WU P.C., LIU B.D., TENG Y.C., YEH C.Y., TSENG S.H., TSAI H.H., JUANG Y.Z. (2013) Novel Automatic Offset Cancelation Approach for Capacitive CMOS MEMS Accelerometers. IEEE International Symposium on Consumer Electronics (ISCE). IEEE. p. 147-148.

[XIA Q. 2019] XIA Q., YANG J.J. (2019) Memristive crossbar arrays for brain-inspired computing. Nature materials, vol. 18, no 4, p. 309-323.

[XINJIANG Z. 2018] Z. XINJIANG, H. ANPING, H. QI, X. ZHISONG, P. K. CHU (2018) Neuromorphic Computing with Memristor Crossbar. Physica status solidi (a), vol. 215, no 13, p. 1700875.

[YOUNG A. R. 2019] YOUNG A. R., M. E. DEAN, J S. PLANK, G. S. ROSE, (2019) A Review of Spiking Neuromorphic Hardware Communication Systems. IEEE Access, vol. 7, p. 135606-135620.

[YU W. 2017] YU W., LIANG F., HEY X., HATCHER W. G., LU G., LINY J., YANG X. (2017) A Survey on the Edge Computing for the Internet of Things. IEEE Internet of Things Journal, vol. 5, no 2, p. 1275-1284.

[ZHENG N. 2018] ZHENG N., MAZUMDE P. (2018) Learning in Memristor Crossbar-Based Spiking Neural Networks Through Modulation of Weight-Dependent Spike-Timing-Dependent Plasticity. IEEE Transactions on Nanotechnology, vol. 17, no 3, p. 520-532.

[ZHI M. 2017] ZHI M., Xin L., LIU H., ZHENG F., GAO H., CHEN Z., CHEN H. (2017) Ferroelectric phase transition of BaTiO3 single crystal based on a tenth order Landau-Devonshire potential. Computational Materials Science, vol. 135, p. 109-118.