Université de Lille

Université de Sherbrooke

Thèse de doctorat

Neuromorphic in-memory learning with analog integrated circuits and nanoscale memristive devices

Apprentissage neuromorphique en mémoire avec des circuits intégrés analogiques et des dispositifs mémoristifs à l'échelle nanométrique

Thèse de doctorat de l'Université de Lille
préparée à UMR CNRS 8520 Institut d'Electronique, de Microélectronique et de Nanotechnologie – IEMN
École doctorale n°632 Sciences de l'Ingénierie et des Systèmes (ENGSYS)
Spécialité de doctorat: Electronique, microélectronique, nanoélectronique et micro-ondes

Thèse de doctorat de l'Université de Sherbrooke
préparée à Laboratoire Nanotechnologies et Nanosystèmes – LN2
Spécialité de doctorat: Génie Electrique

Thèse présentée et soutenue le 5 December 2024, par

NIKHIL GARG

Composition du Jury :

Jean-Michel PORTAL Professeur, Université Aix-Marseille	Président
Sylvian SAIGHI Professeur, Université de Bordeaux	Rapporteur
Elisa VIANELLO Senior Scientist, CEA-Leti	Rapporteur
Sean WOOD Assistant Professor, Université de Sherbrooke	Rapporteur
Laura BEGON-LOURS Assistant Professor, ETH Zurich	Examinateur
Fabien ALIBART Chargé de recherche, CNRS	Directeur de thèse
Dominique DROUIN Professeur, Université de Sherbrooke	Directeur de thèse
Damien QUERLIOZ Directeur de recherche, Université Paris-Saclay	Invité
Yann BEILLIARD Professeur Associé, Université de Sherbrooke	Invité

Centre National de la Recherche Scientifique

iemn Institut d'Electronique, de Microélectronique et de Nanotechnologie UMR CNRS 8520

LN2

RÉSUMÉ

L'intégration de l'intelligence artificielle (IA) dans l'informatique en périphérie (EC) et les dispositifs portables présente des défis importants en raison des contraintes strictes en matière de puissance de calcul et de consommation d'énergie. L'informatique neuromorphique, inspirée par la conception économe en énergie du cerveau et ses capacités d'apprentissage continu, offre une solution prometteuse pour ces applications. Cette thèse propose un cadre flexible de co-conception algorithme-circuit qui aborde à la fois le développement des algorithmes et la conception matérielle, facilitant ainsi le déploiement efficace de l'IA sur du matériel spécialisé à ultra-basse consommation d'énergie.

La première partie se concentre sur le développement d'algorithmes et introduit la plasticité synaptique dépendante de la tension (VDSP), une règle d'apprentissage non supervisée inspirée du cerveau. Le VDSP vise à mettre en œuvre en ligne le mécanisme de plasticité de Hebb en utilisant des synapses memristives à l'échelle nanométrique. Ces dispositifs imitent les synapses biologiques en ajustant leur résistance en fonction de l'activité électrique passée, permettant ainsi un apprentissage en ligne efficace. Le VDSP met à jour la conductance synaptique en fonction du potentiel de membrane du neurone, éliminant ainsi le besoin de mémoire supplémentaire pour stocker les timings des pics d'activité. Cette approche permet un apprentissage en ligne sans les circuits de formage d'impulsions complexes habituellement requis pour la plasticité dépendante du timing des pics (STDP) avec des memristors. Nous montrons comment le VDSP peut être avantageusement adapté à trois types de dispositifs memristifs (synapses à filament d'oxyde métallique et jonctions tunnel ferroélectriques) avec des caractéristiques de commutation analogiques distinctives. Les simulations au niveau du système de réseaux neuronaux à impulsions utilisant ces dispositifs ont validé leurs performances sur des tâches de reconnaissance de motifs sur MNIST, atteignant jusqu'à 90 % de précision avec une meilleure adaptabilité et une réduction du réglage des hyperparamètres par rapport au STDP. De plus, nous avons évalué la variabilité des dispositifs et proposé des stratégies d'atténuation pour améliorer la robustesse.

Dans la deuxième partie, nous implémentons un neurone analogique de type LIF, accompagné d'un régulateur de tension et d'un atténuateur de courant, afin d'interfacer sans heurts les neurones CMOS avec des synapses memristives. La conception du neurone inclut une fuite double, facilitant l'apprentissage local via le VDSP. Nous proposons également un mécanisme d'adaptation configurable qui permet de reconfigurer les neurones LIF adaptatifs en temps réel. Ces circuits polyvalents peuvent s'interfacer avec une gamme de dispositifs synaptiques, permettant ainsi le traitement de signaux avec une variété de dynamiques temporelles. En intégrant ces neurones dans un réseau, nous présentons un bloc de construction neuronal auto-apprenant CMOS-memristor (NBB), composé de circuits analogiques pour la lecture en croix et de neurones LIF, ainsi que de circuits numériques pour basculer entre les modes d'inférence et d'apprentissage. Des réseaux neuronaux compacts, capables de s'adapter eux-mêmes, d'apprendre en temps réel et de traiter des données environnementales, lorsqu'ils sont réalisés sur du matériel à ultra-basse consommation d'énergie, ouvrent de nouvelles perspectives pour l'IA dans l'informatique en périphérie. Les avancées à la fois en matériel (circuits) et en algorithmes (apprentissage en ligne) accéléreront considérablement le déploiement des applications d'IA en exploitant l'informatique analogique et les technologies de mémoire à l'échelle nanométrique.

Mots-clés : Neuromorphic engineering, Synaptic learning, In-memory computing, Memristors, On-chip learning, Spiking neural networks

ABSTRACT

"Somewhere, Something Incredible Is Waiting To Be Known" - Carl Sagan

Integrating artificial intelligence (AI) into edge computing (EC) and portable devices presents significant challenges due to stringent constraints on computational power and energy consumption. Neuromorphic computing, inspired by the brain's energy-efficient design and continuous learning capabilities, offers a promising solution for these applications. This thesis proposes a flexible algorithm-circuit co-design framework that addresses both unsupervised online learning algorithm development and hardware design, facilitating the efficient deployment of AI on specialized, ultra-low-power high-density hardware.

The first part focuses on algorithm development and introduces voltage-dependent synaptic plasticity (VDSP), a brain-inspired unsupervised learning rule. VDSP is aimed at the online implementation of Hebb's plasticity mechanism using nanoscale memristive synapses. These devices mimic biological synapses by adjusting their resistance based on past electrical activity, enabling efficient online learning. VDSP updates synaptic conductance based on the membrane potential of the neuron, eliminating the need for additional memory to store spike timings. This approach allows for online learning without the complex pulse-shaping circuits typically required for spike-timing-dependent plasticity (STDP) with memristors. We show how VDSP can be advantageously adapted to three types of memristive devices (metal-oxide filamentary synapses, and ferroelectric tunnel junctions) with distinctive analog switching characteristics. System-level simulations of spiking neural networks using these devices validated their performance on MNIST pattern recognition tasks, achieving up to 90% accuracy with improved adaptability and reduced hyperparameter tuning compared to STDP. Additionally, we evaluated device variability and proposed mitigation strategies to enhance robustness.

In the second part, we implement an analog leaky integrate-and-fire (LIF) neuron, accompanied by a voltage regulator and current attenuator, to seamlessly interface CMOS neurons with memristive synapses. The neuron design features dual leakage, facilitating local learning through VDSP. We also propose a configurable adaptation mechanism that allows adaptive LIF neurons to be reconfigured in run-time. These versatile circuits can interface with a range of synaptic devices, allowing the processing of signals with a variety of temporal dynamics. Integrating these neurons into a network, we present a CMOS-memristor self-learning neural building block (NBB), consisting of analog circuits for crossbar reading and LIF neurons, along with digital circuits for switching between inference and learning modes. Compact neural networks that can self-adapt, learn in real time, and process environmental data, when realized on ultra-low-power hardware, open new possibilities for AI in edge computing. Advances in both hardware (circuits) and algorithms (online learning) will greatly accelerate the deployment of AI applications by leveraging analog computing and nanoscale memory technologies.

Keywords: Neuromorphic engineering, Synaptic learning, In-memory computing, Memristors, On-chip learning, Spiking neural networks

Dedicated to my parents whom I am indebted to for giving me their love of learning and life

Dédié à mes parents, à qui je suis redevable de m'avoir transmis leur amour de l'apprentissage et de la vie.

ACKNOWLEDGEMENTS

The publication of this thesis marks the culmination of years of dedicated work, and its completion would not have been possible without the support of many individuals who played pivotal roles along the way. For this, I am profoundly grateful. I am especially indebted to my advisor, Prof. Fabien Alibart, who introduced me to this fascinating field and generously shared his time, expertise, and unwavering support at every stage of this journey. I extend my deepest appreciation to him for his guidance and encouragement.

This journey began in 2021 when I embarked on my thesis work in Lille. During my time at IEMN Lille, I had the privilege of working alongside an exceptional group of mentors, colleagues, and friends who supported me in countless ways, from administrative assistance to navigating the complexities of research. I am particularly grateful to Dr. Ankush Kumar for helping me settle in Lille and, along with Dr. Kamila Janzakova, for guiding me through cleanroom experiments at IEMN. The unforgettable experience of working with dendrites will always remain with me. My heartfelt thanks go to Nora and Andy from the administration, who were lifesavers in keeping my paperwork in order and frequently rescuing me with new access cards whenever I misplaced them. Emmanuelle, thank you for managing the reimbursements from my many missions. I am also grateful to Dr. Yannick Coffinier, Dr. Sébastien Pecqueur, Ravi, and Ravinder for their contributions in making my time at IEMN so memorable. Special thanks to Hammouda and Philippe for their warm welcome during my first month in France.

This work would not have been possible without Prof. Damien Querlioz, who graciously hosted me at the C2N lab in Paris. I am also thankful to the C2N team for their camaraderie during our lunches and daily walks to the university restaurant with exciting discussions. Additionally, I extend my sincere thanks to Dr. Laura Begon-Lours for welcoming me to IBM Zurich, introducing me to the neuromorphic device team, and teaching me cleanroom fabrication processes during my two weeks there. I am deeply appreciative of Prof. Jean-Michel Portal for his guidance and hospitality during the UNICO tapeout process in Marseille, where he patiently taught me the fundamentals of IC implementation. I fondly remember our late-night sessions finalizing the design. My gratitude extends to Dr. Eloi Muhr for helping me find housing in Marseille and in the layout of the chip, and to Dr. Mathieu C-Faye, who patiently taught me analog IC design and uplifted my spirits during tough times. A special thank-you to Jean-Pierre for managing the PDK, tools, and libraries, and to Prof. Marc Bocquet for providing critical equipment and training for characterization at the IO lab. Lastly, I must mention Bastien, whose meticulous work in managing delicate chip-to-instrument connections and volunteering to land the probes daily for two weeks was indispensable. I am also grateful to Nicole, Fadi, and Bastien for making my time at IM2NP unforgettable.

The second half of my thesis work took place in Canada at the University of Sherbrooke's 3IT institute. This phase would not have been possible without the mentorship of Prof. Dominique Drouin, who introduced me to imaging and fabrication techniques and graciously hosted a Christmas party at his home. I am profoundly grateful to my co-director, Prof. Yann Beilliard, who closely mentored me for over four years, integrating me deeply into the UNICO project. Collaborating with Ismael, who taught me about SNN simulation in Python and became both a great friend and colleague, was a highlight of this journey. My heartfelt thanks go to the NECOTIS team, including Ahmad, Ilham, Arnaud, Prof. Jean Rouat, and Prof. Sean Wood, for their invaluable collaboration and support. I am equally grateful to my colleagues at 3IT, especially Joao for being a wonderful friend and collaborator, as well as Raphael, Mathieu, Gaspard, Pierre, Dorian, Victor, Patrick, Davide, Tadeas, Javier, and Prof. Serge Ecoffey. I also want to acknowledge the Platforms team, particularly Johnathan, who spent countless hours helping me set up chip probing at LCSM, as well as Joel, Gabriel, and Charles for their technical assistance. My gratitude extends to my friends, housemates and colleagues, including Chander, Adi, Shrivani, and Swayam.

Lastly, I want to thank the reviewers of my thesis and all the jury members for their valuable time and insightful feedback. I am also deeply appreciative of my former teachers and mentors, including Prof. Veeky Baths, Prof. Amalin Prince, and Prof. Abhijit Pethe.

I would be remiss if I did not acknowledge my family and closest friends for their sacrifices and unwavering support. This work would not have been possible without them.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

Acronym	Definition
ADC	Analog to Digital Converter
AER	Address Event Representation
AI	Artificial Intelligence
ANN	Artificial Neural Network
APMU	Analog Pulse Measurement Unit
ART	Adaptive Resonance Theory
ASIC	Application Specific Integrated Circuit
BDSP	Burst-Dependent Synaptic Plasticity
BCM	Bienenstock Cooper Munro
BEOL	Back End of Line
BL	Bit Line
BP	Back-Propagation
BPTT	Back-Propagation Through Time
CA	Current Attenuator
CAM	Content Addressable Memory
C-MPDP	Calcium-based MPDP
CNN	Convolutional Neural Network
CMOS	Complementary Metal-Oxide-Semiconductor
CPU	Central Processing Unit
C-STDP	Calcium-based STDP
DAC	Digital to Analog Converter
DPI	Differential Pair Integrator
DPSS	Dendritic Prediction of Somatic Spiking
DVS	Dynamic Vision Sensor
FDSOI	Fully Depleted Silicon On Insulator
FPGA	Field Programmable Gate Array
GPIO	General-Purpose Input/Output
GPU	Graphics Processing Unit
GUI	Graphical User Interface
HMPDP	Homeostatic MPDP
ICA	Independent Component Analysis
IC	Integrated Circuit
IoT	Internet of Things
LCA	Locally Competitive Algorithm
LDO	Low Dropout Regulator
LIF	Leaky Integrate and Fire
LTD	Long-Term Depression
LTP	Long-Term Potentiation
MPDP	Membrane Potential Dependent Plasticity

Acronym	**Definition**
NBB	Neural Building Block
NEF	Neural Engineering Framework
NLP	Natural Language Processing
OpAmp	Operational Amplifier
OTA	Operational Transconductance Amplifier
PCA	Principal Component Analysis
PCM	Phase Change Memory
PCB	Printed Circuit Board
RAM	Random Access Memory
RBM	Restricted Boltzmann Machine
RDSP	Rate Dependent Synaptic Plasticity
RRAM	Resistive Random Access Memory
RTRL	Real-Time Recurrent Learning
SDSP	Spike-Driven Synaptic Plasticity
SL	Source Line
SNN	Spiking Neural Network
SOA	State-of-the-art
SBCM	Spiking BCM
SR	Shift Register
SRAM	Static Random Access Memory
SRDP	Spike-Rate Dependent Plasticity
STDP	Spike-Timing Dependent Plasticity
TA	Transconductance Amplifier
TGATE	Transmission Gate
TSTDP	Triplet-based STDP
TPU	Tensor Processing Unit
VDSP	Voltage Dependent Synaptic Plasticity
VLSI	Very Large Scale Integration
VMM	Vector Matrix Multiplication
V-STDP	Voltage-based STDP
WL	Word Line
WTA	Winner-Take-All

CHAPTER 1

Introduction

"All truths are easy to understand once they are discovered; the point is to discover them. The challenge in science is not just in understanding, but in finding the right questions to ask. Once the right question is asked, the path to discovery becomes clearer, and what once seemed impossible becomes within reach. " – Galileo Galilei

TABLE OF CONTENTS

1.1 Context

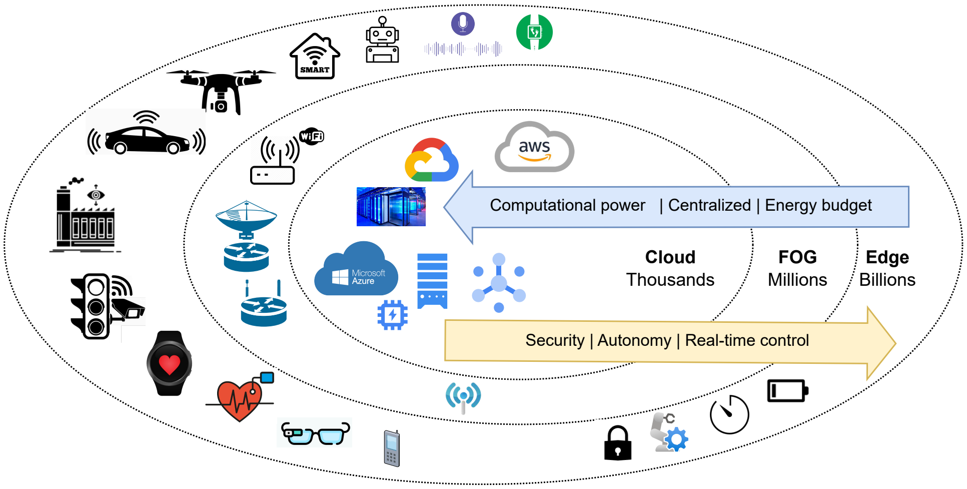

aws
Microsoft Azure
SMART
WiFi
Computational power | Centralized | Energy budget
Cloud
Thousands
FOG
Millions
Edge
Billions
Security | Autonomy | Real-time control

Figure 1.1 Different level of deployment from cloud to end devices. Constraint of security, energy budget, latency, and closed loop optimizations

During the past few decades, the exponential growth in the Internet of Things (IoT) devices, the expansion of memory device storage capacity, and significant advances in computing architectures have collectively paved the way for the development of Artificial Intelligence (AI) models capable of executing highly complex pattern recognition tasks. AI models are increasingly being integrated into **edge computing** (EC) devices, including wearable fitness sensors, autonomous robotic systems, and assistive technologies, to perform pattern recognition and decision-making tasks locally, often in real-time [1]. These applications, however, face critical constraints due to limited power budgets and the need for rapid processing, and each EC application also has its own specific requirements, such as the nature of the data and the core computing tasks involved [2]. In traditional software-based machine learning approaches, these challenges are typically addressed by designing application-specific neural networks. However, this flexibility becomes a major limitation when developing hardware solutions, as each application requires a specialized hardware design. A promising direction is to consider the efficiency of biological systems such as the human brain. For example, while the human brain operates with a power consumption of approximately 20 watts to support around 10^{11} neurons and 10^{15} synapses [3], this efficiency is several orders of magnitude greater than what is achievable with modern computing technologies [4]. As a result, AI hardware must cope with processing unstructured natural data that exhibit strong temporal variability, often with limited training examples.

Developing energy-efficient hardware that can meet these demands remains a significant challenge for edge computing applications.

Neuroscience has profoundly influenced the field of AI, particularly through the development of artificial neural networks, which have driven the evolution of machine intelligence over the past several decades [5]. Early milestones in AI include the creation of the perceptron in 1960 [6], the introduction of layered neural networks in 1987 [7], and the development of deep multilayer networks [8]. These advances were accompanied by significant algorithmic innovations, such as error correction and optimization strategies like backpropagation [9]. One of the earliest applications of these networks was in artificial pattern recognition, demonstrated in the analysis of ECG signals as early as 1962 [10]. Similarly, the progress in AI has been tightly coupled with advancements in computing **hardware**. The ability to train large-scale networks has been substantially improved by parallel computing architectures, such as Graphics Processing Unit (GPU)s and Tensor Processing Unit (TPU)s, which have significantly accelerated model training times and enabled the development of ultra-large-scale models like those used in modern large language models, including ChatGPT. The process of **model training**, which involves optimizing network weights using large datasets, has benefited immensely from increased storage capacity and computing power, enabled by aggressive semiconductor scaling following Moore's Law. However, this exponential growth in computing power has also led to significant energy consumption. For instance, training OpenAI's GPT-3, with 175 billion parameters, required 10,000 GPUs and consumed 936 megawatt-hours of **energy during its training** [11].

This growing energy demand highlights the need to **rethink AI hardware**, taking inspiration from the energy-efficient mechanisms of the human brain [12]. Unlike power-intensive computational paradigms that rely on stacking thousands of processors, the brain operates in a compact, energy-efficient manner, which is crucial for the development of portable, low-power AI systems, especially in **edge computing** applications. Furthermore, reliance on digital memory and processing technology, which serves as the backbone of contemporary computing systems, is increasingly being challenged. As a result, there has been a growing interest in exploring alternative approaches.The emerging field of **neuromorphic computing and engineering** has been an area of active research for more than three decades [13, 14]. This field has revealed how AI algorithms, circuits, and electronic devices can be designed to emulate the architectural and operational principles of the biological brain. A closer examination of these principles indicates that natural intelligence operates differently from modern general-purpose computers. Specifically, the biological brain benefits from the synchronization of instruction operations and the co-location of processing and memory units, which has led to the development of specialized asynchronous and in-memory computing chips in artificial systems.

By mimicking the brain's ability to perform **computation and memory storage in the same location** and processing information in an **event-driven** fashion, neuromorphic engineering aims to develop more energy-efficient and scalable AI systems. A paradigm shift is required not only in operating principles and architecture but also in the **computing substrate**. Emerging nonvolatile memory technologies, including memristive devices, emulate various properties of biological synapses and enable ultradense and energy-efficient neuromorphic hardware utilizing **physical mechanisms** [15]. In addition, Integrated Circuit (IC) chips featuring co-integrated Complementary Metal-Oxide-Semiconductor (CMOS) computing circuits and emerging nanoscale non-volatile memory devices have been recognized as ultra-efficient solutions for executing complex computing tasks [16]. Among these emerging technologies, memristors stand out as particularly promising for the realization of in-memory computing hardware due to their nanoscale footprint, which allows for a high degree of **integration**, their analog **programming** capabilities, and their compatibility with CMOS processes for fabrication and integration.

The integration density, throughput, and speed of these systems are significantly improved by utilizing analog in-memory computing principles with memristors. This enhancement is particularly important for brain-inspired SNN-based AI circuit implementations, which require extensive memory access, parallel processing architectures, and efficient non-linear transformations. The superior power efficiency of SNNs arises from their event-based sensing and computing paradigm. Unlike conventional systems driven by clock cycles, SNNs are activated by changes in environmental conditions, enabling more energy-efficient processing. Additionally, computation within SNNs takes place in the analog domain at the level of neurons, while communication between neurons is facilitated through digital spike events. This hybrid approach combines the best aspects of analog and digital processing, contributing to the overall efficiency of the system.

1.2 Challenges

One of the characteristic features of SNNs is the incorporation of a temporal dimension into their data processing. The timing of spike events is crucial to the information being processed, with system variables evolving dynamically over time. While this temporal aspect makes SNNs particularly well-suited for processing time series data, it also introduces challenges. The added complexity of time-based data representation makes it difficult to directly apply strategies developed for training conventional Artificial Neural Network (ANN)s to Spiking Neural Network (SNN)s.

The realization of **SNNs** through **custom hardware**, leveraging in-memory computing architectures, analog circuits and emerging memristive devices, offers significant improvements in hardware utilization. These improvements are achieved by harnessing the physical laws of computing, leading to more efficient and compact systems. However, this approach also introduces specific constraints in the system's conception. Custom hardware systems for SNNs often have fixed topologies, predefined computational building blocks, and limited resolution of both voltage levels in computational blocks and resistive states in synaptic devices [17, 18]. Although power-efficient operation is a key advantage, achieved through low-voltage operations, this also makes systems more susceptible to noise, an issue that is less significant in digital systems due to quantization. Furthermore, the finite resolution of semiconductor patterning processes introduces variability, reducing the repeatable precision achievable with nanoscale memristive synapses. This variability makes precise programming of memristive synapses challenging, as the stochastic nature of switching behavior leads to variability in switching voltage and conductance range. Although high-precision programming circuits have been proposed [19], the circuit overhead for learning and interfacing must be minimized to ensure that the scalability benefits of such hardware are not compromised.

For **deployment of** AI applications through such custom SNN **hardware**, the typical approach involves labeling a sample of population data and using it to optimize the weights of the SNN topology through gradient-based learning techniques. However, this approach limits the generalization capabilities of the system as it is heavily based on the size and quality of the training data. Moreover, once deployed, such systems often lack the ability to learn from out-of-sample data, which hinders their adaptability in real-world scenarios where environmental conditions and device characteristics may change over time.

To address these limitations, **online learning, lifelong learning, or always on learning** systems are emerging as promising approaches to create AI systems that continuously adapt to new data and device imperfections. However, implementing gradient-based learning in an online setting poses significant challenges. The non-linear nature of SNNs, the requirement for weight transposition, the bidirectional signal propagation, and the temporal credit assignment problem inherent in time-evolving SNNs complicate the deployment of such learning methods. In addition, the energy costs associated with error propagation and credit assignment-based learning are substantial, as these processes require transmitting complex learning signals across different parts of the chip. To fully exploit the area-energy advantages offered by neuromorphic in-memory computing architectures and memristive device technology, online learning must be designed to account for local variables and be implementable with minimal silicon overhead.

Although biologically plausible local Hebbian learning rules have been proposed for SNNs and memristive synapses, **realizing a scalable on-chip implementation** remains an **algorithm-circuit-device engineering** challenge. Developing an integrated system that combines spiking neurons, memristive synapses, and online learning capabilities is essential to create practical AI solutions capable of solving real-world pattern recognition problems.

1.3 Research question and objectives

Neuromorphic computing principles should provide the direction towards building low-power intelligible systems for edge-AI deployment. These principles are derived from centuries of biological evolution and subsequent neuroscience research that have uncovered the unique architectural and operational characteristics that contribute to the efficiency of the human brain. However, electronic systems have also evolved significantly in the past century, leading to breakthrough discoveries such as CMOS-based integrated circuits and nanoscale emerging memory devices. These CMOS and memory devices offer ultra-compact integration density, and with a new logic-memory integrated architecture, the energy gap between the biological brain and electronic systems can be bridged. In this thesis, we propose to bridge this energy gap by leveraging the principles of analog computing hardware and online learning, focusing on the following research question:

How to translate neural learning principles to analog electronic devices and systems?

Neuromorphic learning principles, such as Hebbian plasticity, are appealing for hardware implementation because they offer: (i) online processing, (ii) the ability to operate without supervision or the need for correct labels, and (iii) localized updates based on the state variables of pre- and post-synaptic neurons, avoiding the necessity for global error propagation from the classification layer. Analog electronic devices include a combination of Complementary Metal-Oxide-Semiconductor (CMOS) transistors integrated on silicon and memristive non-volatile nanoscale memory devices co-integrated to CMOS ASIC. The first question we explore is how to adapt neural learning principles for real-time synaptic weight learning in hardware. Specifically, which key events (triggers)—derived from Hebbian learning—initiate the learning process, and which local variables of the neuron (such as spike timing and membrane potential) dictate the polarity and magnitude of the learning?

Thus, the **first objective (OB1)** is to **realize a hardware-friendly local learning algorithm** within the SNN simulation framework, enabling the evaluation of its efficiency for unsupervised pattern classification and benchmarking it against state-of-the-art algorithms like Spike-Timing Dependent Plasticity (STDP).

1.3.1 On device learning

Nanoscale memristive devices hold significant promise due to their nonvolatile memory, programmable conductance levels, and compatibility with CMOS technology, making them ideal candidates for high-density memory blocks in AI hardware. The second set of questions to explore includes how the behavior of memristive devices influences learning, how the programming signal impacts network performance when integrated with synaptic devices operating on different principles such as oxidation or ferroelectricity, and how analog programming can be adjusted to account for known device behaviors like variability.

The **second objective (OB2)** is to develop a **memristive programming strategy** that leverages the analog properties of memristive devices by translating the online learning rule into hardware. Supported by characterization, modeling, and system-level simulations, this strategy aims to benchmark resistive and ferroelectric devices and implement efficient learning for real-world tasks, addressing the non-idealities and accuracy limitations of analog computing.

1.3.2 Circuits

Circuit realization is crucial for the experimental validation of CMOS circuits and hardware-implemented online learning, especially given the complexities of modeling CMOS circuits in the sub-threshold region and memristive devices, where device dimension mismatches arise from the finite resolution of semiconductor patterning. This challenge intensifies when scaling a microscopic physics-based model to an entire spiking neural network, as managing the computational overhead of solving all governing differential equations and their interactions becomes essential. These issues raise the key question: What circuits and functionalities are needed to interface with synaptic devices and generate learning signals? Furthermore, how much flexibility can be achieved to accommodate different synaptic devices, network architectures (scale/application), and the time scales of the signals involved?

The **third objective (OB3)** is to develop **computation circuits** using biomimetic analog neurons fabricated in CMOS technology. Beyond computing, these circuits must **interface with the memristive synapse** to ensure impedance matching and spike transmission without altering the memristive state.

Finally, by integrating analog and digital circuit components, an architecture for real-time learning and prediction is proposed. The **fourth objective (OB4)** is to develop and verify mixed-signal circuits for analog computation, while implementing communication and control through asynchronous digital logic, ultimately achieving a low-power real-time SNN prototype.

1.4 Organization of thesis

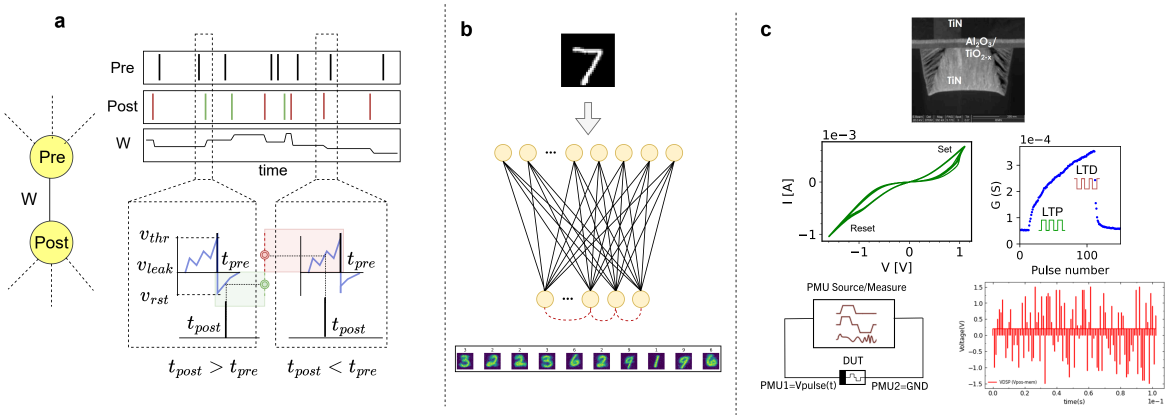


Figure 1.2 Voltage Dependent Synaptic Plasticity (VDSP) for online unsupervised local learning with memristive synapses. **a** Schematic representation of VDSP based learning based on membrane potential of pre-synaptic neuron. **b** The images of handwritten digits were input into an SNN topology comprising two layers of spiking neurons, fully interconnected via memristive synapses (top), and the learned receptive fields post-training with the MNIST dataset (bottom). **c** (top) cross-section of a TiO_2 memristor [20]. Quasi-DC current-voltage (I-V) characteristics of the memristive device and multi-level switching behavior achieved via pulse programming (middle). (bottom) A dedicated pulse-based characterization method is used to validate VDSP-based learning through simulations.

1.4.1 Background

From a brief historical perspective of the development of neuroscience, electronic devices, and computing, chapter 2 introduces the background and state of the art. Specifically, first, neuromorphic principles are introduced, emphasizing the unique representation of time and memory in spiking neural networks. Next, the mechanism of learning or plasticity is examined, particularly highlighting unsupervised, local, and online learning methods, followed by an analysis of circuits to implement such learning in hardware. Third, the hardware implementation of SNNs is elaborated, with the motivations behind choosing analog in-memory computing hardware systems using memristive synapses. Finally, the current engineering challenges and scientific knowledge gaps in the literature in implementing online learning on memristive hardware are presented.

1.4.2 Voltage dependent synaptic plasticity

chapter 3 centers on **OB1** and presents VDSP for hardware-oriented implementation of Hebbian learning. Starting from the challenges associated with hardware implementation of STDP, this chapter introduces the philosophy of estimating a neuron's spike time through its membrane potential. (Figure 1.2a) The above argument is supported by mathematical derivation and the conditions for an accurate or stochastic estimation of the spike time are presented. The learning rule was modeled in the SNN simulation framework, and the learning performance was evaluated using unsupervised handwritten digit recognition (Figure 1.2b). Important hyperparameters such as learning rate are discussed, followed by comparing performances with the ones reported in the literature. Next, advantages with respect to STDP are elaborated through comparative benchmarking. Finally, the impact of noise on SNN performance is evaluated.

1.4.3 Learning with memristive synapses

chapter 4 focuses on **OB2**, examining the interaction between the designed VDSP learning rule and the switching behavior of memristive devices. This learning avoids the complex pulse shaping circuitry required for STDP implementation. We provide the first demonstration of neuron state-based online learning with memristive devices, characterizing and modeling the distinct switching behaviors of two resistive devices based on TiO_2 (Figure 1.2c (top)) and HfO_2 based valence change memory and a ferroelectric tunnel junction based memristive device. Previous studies [20] have evaluated quasi-DC (Figure 1.2c(middle)) and LTP/LTD multi-level programming by applying sequence short pulse of fixed voltage first to induce potentiation in steps followed by depression. The voltage-dependent switching was characterized by a dedicated electrical measurement protocol using random voltage pulses (Figure 1.2c). Subsequently, a parametric model was developed to represent resistive switching dependent on applied voltage magnitude and the device's resistance state, capturing key memristive properties such as non-linearity and switching threshold. The model allows for system-level analysis of learning efficiency across various deviations in the behavior of the devices. The simulation of mismatch allows to fine-tune the learning-circuit parameters to match characteristics of device under consideration like threshold, non-linearity, variability, and resistance range. The study also demonstrates the resilience of the learning efficiency to variations in device parameters, suggesting the potential for engineering efficient circuits and systems with stochastic but scalable memristive devices.

1.4.4 CMOS neuron for memristor integrated neuromorphic circuits

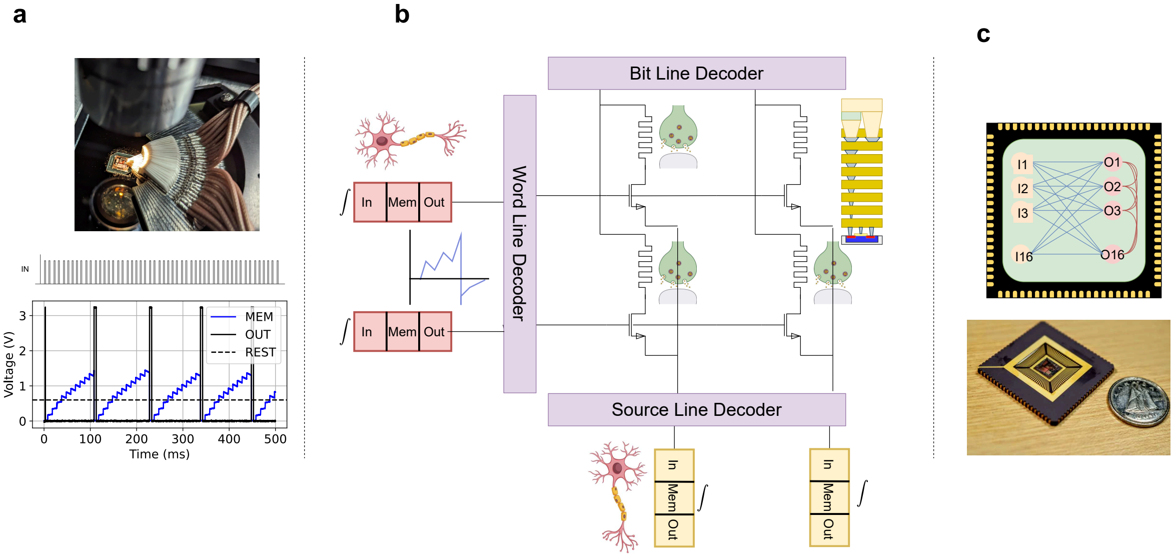


Figure 1.3 Computing circuit and SNN architecture. **a** Probe testing of synaptic reading and on-chip Leaky Integrate and Fire (LIF) neuron (top), measurement results of neuron membrane potential and output spikes in response to periodic excitation through input spikes (bottom).**b** Architecture for on-chip learning managed by specific Bit Line (BL), Word Line (WL), and Source Line (SL) decoders (mixed-signal circuits) within a 1T1R crossbar and LIF neurons. **c** Simplified diagram of the fabricated Application Specific Integrated Circuit (ASIC), comprising two banks of analog spiking neurons interconnected by a memristive crossbar synaptic array (top) and packaged chip through wire-bonding (bottom).

Toward **OB3**, chapter 5, an analog LIF neuron is presented, highlighting the sub-blocks for interaction with the memristive synapse through a voltage regulator and a current attenuator implemented for stable reading of the memristor. The LIF neuron features a dual leak mechanism on the biological time scale to enable implementation of VDSP based online learning. Additionally, configurability through bias voltage is engineered to make the neuron suitable for a wide range of applications. In addition, a novel connection scheme is proposed to dynamically reconfigure the first-order LIF neuron to an adaptive neuron for homeostasis. The circuits were implemented using CMOS technology, and extensive electrical characterization was performed (Figure 1.3a) to validate critical functionalities such as sensitivity to synaptic resistance and modulation of neuron characteristics such as threshold, refractory period, and leak rate.

1.4.5 Neural building block for 3D integrated CMOS-RRAM SNNs

chapter 6 towards **OB4** elaborates the architecture and circuits of the analog and mixed-signal CMOS-Resistive Random Access Memory (RRAM) neural building block for implementing on-chip online learning. The circuits for data-path composed of two layers of spiking neurons interconnected by 1T1R memristive synaptic array are presented. Next, the control architecture (Figure 1.3b) and digital circuit elements composed of the configuration registers are described to switch between the inference and learning phases. In addition, a mechanism for implementing winner-takes-all-based lateral inhibition is presented.

The fabricated ASIC (Figure 1.3c) was packaged and characterized using a custom Field Programmable Gate Array (FPGA) controlled test Printed Circuit Board (PCB). The digital circuits on the chip were measured to demonstrate online learning, and timing diagrams of the measurements are elaborated to validate the control logic. Finally, the results of the characterization of all neurons implemented in the SNN are compared to showcase the impact of device mismatch in analog CMOS circuits.

1.4.6 Conclusion and Perspective

In conclusion, (i) we propose a dedicated online learning rule and demonstrate the programming strategy with memristive devices. (ii) We develop and validate CMOS circuits for computing, interfacing, and learning with memristor. These circuits are analog and digital: for computing, utilizing physical principles of charge integration and Ohm's law and leveraging high-speed, low-power digital circuits and architecture to control and interface analog computing blocks.

CHAPTER 2

Background

"We are like dwarfs sitting on the shoulders of giants. Our glance can thus take in more things and reach farther than theirs. It is not because our sight is sharper nor our height greater than theirs; it is that we are carried and elevated by the high stature of the giants."
– Bernard de Chartres

TABLE OF CONTENTS

2.1 Outline

Subsequent to a historical summary of development in the computing, neuroscience, and electronic domain, this chapter first outlines the key distinctive principles of neuromorphic engineering: computing in the time domain and the memory hierarchy in section 2.3. The process of memory formation or learning is discussed in the second section section 2.4, including theories of unsupervised, online, and local learning. Third, the development in hardware paradigms is discussed, with a special focus on analog domain computation, in-memory processing architectures, and emerging nanoscale memory devices in section 2.5.

2.2 History

Over the past century, the fields of electronic devices, computing systems, and neuroscience have advanced significantly. These three areas form the basis for current computing hardware and cutting-edge artificial intelligence applications, including ChatGPT [21], self-driving cars [22], and wearable healthcare monitoring [23].

Early **computers**, like the ENIAC, were called "fixed-program computers" because they needed physical rewiring for different tasks [24]. In ***1945***, John von Neumann introduced the "stored-program computer" concept, allowing data and instructions to be stored in the same memory, simplifying reprogramming [25]. This was first realized with the EDVAC in ***1949*** [26], while the Manchester Baby, running its first stored program in ***1948***, is recognized as the earliest practical example. The Manchester Mark 1 in ***1949*** further showcased the potential of von Neumann's architecture [27]. Around ***1955***, the development of the Central Processing Unit (CPU) centralized instruction execution, boosting efficiency, and advances in transistor technology led to the first microprocessors, such as the Intel 4004 in ***1971*** [28] and Intel 8080 in ***1974*** [29]. Alongside CPUs, specialized processors like the GPU (1980s), initially for graphics, found broader use with parallel processing capabilities [30]. Google's TPU, introduced in ***2015***, further advanced machine learning tasks [31].

The field of **neuroscience** has advanced through key discoveries, beginning with Cajal's establishment of the neuron doctrine in ***1887***, which identified neurons as discrete units of the nervous system [32]. Significant progress followed in ***1939*** when Hodgkin and Huxley recorded the first action potential [33]. In ***1949***, Donald Hebb introduced the Hebbian theory, explaining synaptic strengthening through simultaneous neuronal activity, fundamental to learning and memory [34]. The 1970s brought further insights into neural plasticity, including the discovery of Long-Term Potentiation (LTP), key for synaptic strengthening, by Bliss and Lomo in ***1973*** [35]. In the late ***1990s***, Bi and Poo uncovered STDP, showing that spike timing determines the direction of synaptic changes [36]. Recent research emphasizes the role of astrocytes in synaptic function and plasticity, adding complexity to our understanding of neural networks [37].

The development of **electronic devices** began with the invention of vacuum tubes in ***1905***, the first components capable of amplifying signals [38]. Electronic current control was proposed in ***1930*** [39] and realized in ***1948*** with the invention of the transistor by Bardeen, Brattain, and Shockley [40], revolutionizing electronics. This led to the development of CMOS technology in ***1959*** [41] and integrated circuits, driving Moore's law, which predicted exponential increases in transistor count. The introduction of FinFET in ***1998*** [42] addressed short-channel effects as transistor sizes shrank. Innovations continued with carbon nanotube transistors in ***2002*** [43], gate-all-around MOSFETs in ***2008*** [44], and the 3D FinFET in ***2012*** [45], further improving performance and efficiency.

2.3 Neuromorphic computing principles

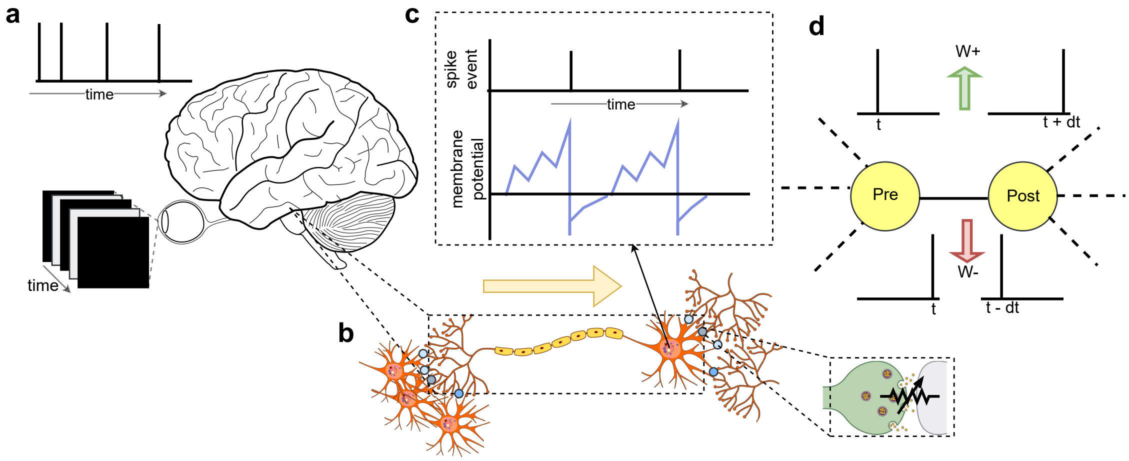


Figure 2.1 Neuromorphic computing: The salient features of the human-brain worth taking inspiration for intelligent machines **a** Event-based sensing. **b** co-location of computation and memory. **c** evolving temporal dynamics. **d** synaptic plasticity based on local variables

Although the term **neuromorphic engineering** was first introduced by Carver Mead in the 1990s to describe the use of CMOS transistors operating in the weak-inversion region to emulate neurons' ion channels [13], the field has expanded significantly over the last three decades. Neuromorphic computing now encompasses a broader range of technologies and concepts inspired by the architecture and function of the brain. The key features of the brain that serve as inspiration are illustrated in Figure 2.1.

First, asynchronous event-based sensing and processing, unlike the traditional clock-based paradigm, processes information only when an event occurs. This approach drastically reduces power consumption, particularly for always-on, sparsely activated applications.

Second, the brain's architecture integrates memory and processing in a co-located manner, with neurons and synapses densely interconnected, in contrast to the separate CPU and RAM in traditional von Neumann architectures. This integration allows for more efficient data processing and storage. Third, the brain's ability to exhibit time-evolving neural dynamics and continuous adaptation in response to environmental stimuli provides a powerful model for developing systems that can learn and adapt in real time. These principles are foundational to the development of neuromorphic computing technologies and could be grouped into time-domain computing and unique memory characteristics.

2.3.1 Time-domain computing

In general-purpose computing processors, a centralized **clock** serves as the primary mechanism for coordinating instruction execution, ensuring that all operations occur in a synchronized manner. However, neuromorphic systems fundamentally differ by relying on the timing of spike events to convey information, closely mimicking biological neural processes. Different processes evolve at vastly different time scales, and the sequence of occurrence of an event is used to encode, process, retain, and transmit information.

At the sensor level, neuromorphic systems utilize an event-based spike encoding paradigm, as seen in Dynamic Vision Sensor (DVS) cameras [46]. Here, the instantaneous intensity value is compared with the last recorded value, and if it exceeds a certain threshold, a spike is emitted. This allows for temporal resolutions on the order of microseconds to milliseconds. Similarly, in dynamic audio sensors [47], which are inspired by the biological cochlea, the input is filtered into different frequency bands, each representing sub-signals evolving at different time scales ranging from 20 kHz to 20 Hz.

Next, the leaky integrate-and-fire neuron model [48] integrates incoming spikes and emits a spike when the accumulated charge surpasses a predefined threshold. The leak rate of such neurons typically operates on the order of hundreds of milliseconds. More biologically accurate neuron models, such as those described by Izhikevich [49], exhibit homeostasis or adaptation mechanisms. These models include additional state variables, analogous to calcium ion concentration in biological neurons, which adjust the spiking threshold to regulate neural activity over time, with dynamics occurring on the scale of seconds. Lastly, learning is carried out with eligibility traces [50], which tag the synapses upon the occurrence of special events and are utilized for updating weights upon the arrival of a reward. These processes occur over a time scale ranging from one to several hundred seconds.

The asynchronous and event-driven nature of neuromorphic systems contrasts sharply with conventional clock-driven processors, where all subsystems operate on the same clock cycle. This approach fails to account for the varied time scales of different neural processes, leading to inefficiencies. In neuromorphic systems, the timing hierarchy—from rapidly responding to input changes, to the slow dynamics of neuron state variables, to the rare communication of spikes, and finally, to the long-term processes of learning and memory—requires each function to operate at its respective speed. This makes deployment of such system on conventional processor inefficient due to the energy overhead associated with unnecessary clock cycles and the constant refreshing of dynamic memory.

2.3.2 Memory

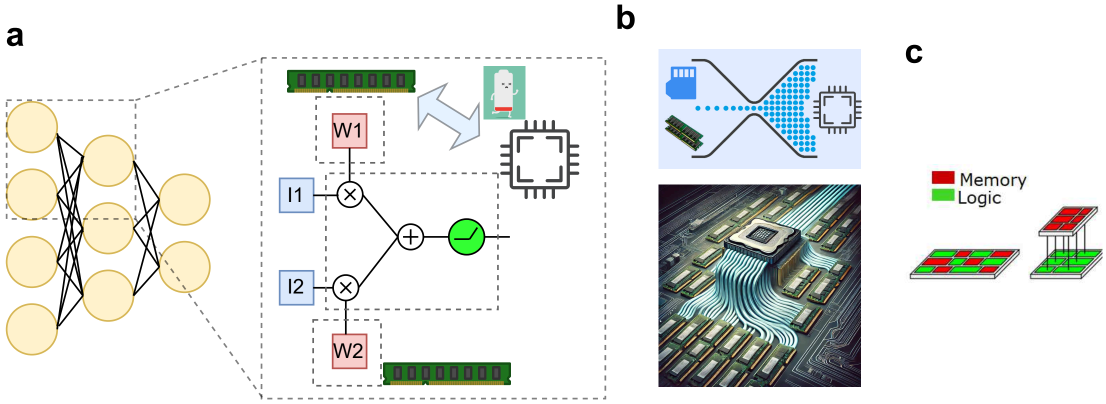


Figure 2.2 Memory transfer bottleneck in neural networks and computing architecture: **a** The weights and bias for inference in a neural network are stored in memory. The computation however takes place in processor requiring massive data transfer. A large fraction of energy is spent in shuttling data between memory and processor. **b** There exists a bottleneck due to the difference between required throughput and data transfer capabilities of communication architecture (top). The bottleneck is illustrated through AI generated image (bottom). **c** 3-D integration of memory and computing logic to overcome the memory transfer bottleneck.

The different temporal processes of neural dynamics translate into a hierarchy of memory creation. In second-generation neural networks, synaptic weights account for the memory overhead. To compute the network's decision, the sensed data are effectively multiplied and accumulated with the stored weights, as depicted in Figure 2.2. Modern computing architectures also feature hierarchical memory, organized based on physical distance from the processor: several levels of caches, dynamic and static Random Access Memory (RAM), and stored memory in hard disks. However, as model sizes have grown, particularly the size of all weights needed for inference, they often exceed the capacity of fast caches and must be fetched from RAM. This memory retrieval process accounts for a significant portion of the latency and energy overhead in modern computing systems.

Spiking neural networks, being stateful machines, exacerbate this memory bottleneck. In these networks, two levels of memory retrieval are required: first, for updating the state of evolving variables (neurons), where the value from the last time step is used; and second, for accessing the synaptic weights used to calculate the weighted sum of dendritic spikes. Additionally, more complex neuron models are multi-compartmental [51], meaning they have several state variables that need to be stored, fetched, and updated. Neurons in these models are highly parameterized [52, 53], with unique parameters such as spiking thresholds and leak rates, which also contribute to the memory overhead. Furthermore, activity traces [54] are stored over extended periods to calculate weight updates, adding another layer of memory requirements.

This combination of frequent state updates, complex parameter management, and long-term storage needs makes memory management in SNNs far more challenging than in traditional computing architectures, highlighting the need for specialized memory systems capable of efficiently handling the dynamic, high-dimensional data typical of spiking neural networks.

2.4 Learning and adaptation

The process of memory formation occurs as the result of processes called plasticity [55] in the biological brain and as network training in artificial neural networks. The following section discusses different approaches for such learning and adaptation.

The minimization of errors through gradient descent and backpropagation [9] in training the weights of deep neural networks [8] has been shown to be highly efficient for learning various types of patterns [56]. However, when applied to SNNs, this approach encounters significant challenges due to the **recurrent nature** of SNNs and the **discontinuity** in their transfer function caused by spike-based representations. The recurrent behavior in SNNs arises because the current state of a neuron depends not only on the present stimuli but also on its previous state, necessitating advanced algorithms like backpropagation through time. Although strategies such as eligibility traces have been proposed to mitigate issues like vanishing gradients, the feasibility of real-time online learning during deployment remains uncertain. Thus, conventional gradient-based optimization does not translate directly to online learning in SNNs.

Furthermore, gradient-based training typically relies on the calculation of **error** signals, which are not readily available in autonomous systems before human annotation, underlying the importance of unsupervised learning. Biological neural systems exhibit homeostasis or adaptation to changing environments, a trait that has been emulated in artificial SNNs to enhance their performance [57]. Local learning rules, which do not require the transmission of error signals across the layers of a deep network, are particularly attractive for hardware implementation. This is especially relevant in memristive in-memory computing architectures, where such transmission could negate the benefits of compute-memory co-location. Thus, three key features are essential for efficient learning in SNNs: **online** adaptation, **unsupervised** learning, and **locality**. The following sections will first explore these aspects from an algorithmic perspective, followed by a discussion of their implications for hardware design and deployment.

2.4.1 Online

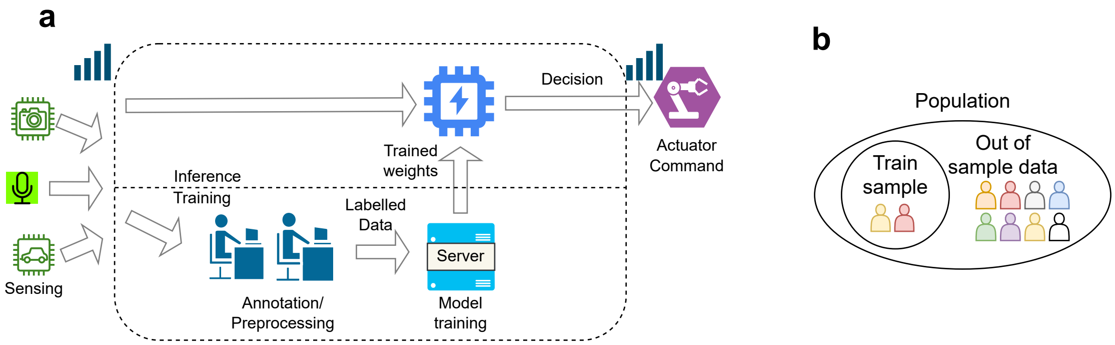


Figure 2.3 Offline learning and cloud computing **a** In the offline learning model, data captured by IoT devices is transmitted to cloud servers for processing. The processed results are then relayed back to the originating devices. As data accumulates from various devices, it is annotated by experts, creating labeled datasets that are subsequently used for supervised training of the network's parameters. **b** Limited generalization capabilities of offline unsupervised learning to the training sample. The venn diagram illustrates the fact that out of the entire available population only a small fraction is used as training data, limiting generalization of gradient-based supervised learning.

Continuous adaptation in response to environmental stimuli and rewards / punishment in response to action is a key feature of natural intelligence. However, machine intelligence deviates significantly from this and is heavily based on offline learning, as shown in Figure 2.3. In the offline learning paradigm, the signals sensed by IoT devices are transferred to cloud computing servers where they are processed, and inference output is sent back to the end device. Upon accumulation of data from all the devices, experts annotate the samples which are, in turn, used for supervised learning of network parameters. However, the quality and quantity of labeled samples strongly influence the network's performance.

On-line learning, on the other hand, circumvents the above limitation by empowering the sensor itself to learn from out-of-training sample data, thus dramatically enhancing the generalization capabilities of the deployed algorithm throughout the agent's lifetime. This mechanism is inspired by natural intelligence and has been demonstrated first byAdaptive Resonance Theory (ART) [58]. Moreover, online learning can also help circumvent the variability associated with analog hardware[59].

2.4.2 Unsupervised

Machine learning algorithms [60] can be broadly classified into supervised, unsupervised [61], and reinforcement learning [62]. Although supervised learning uses annotation of training examples by a human expert to quickly converge the model parameters, it requires annotation of every training example. The dependence on data annotation becomes a critical bottleneck in online learning scenarios where the optimization is performed at the same time as system deployment, as discussed in the previous subsection. Moreover, state-of-the-art neural networks are massive in size, and with the growing amount of training data available, this size is not expected to decrease. There is a need for a self-adaptive AI inference system that does not require human-assisted annotation of all sensed signals.

Unsupervised learning in the machine learning domain attracted interest in clustering in 1998 with the ART1 computer [63]. The K means clustering [64] has been widely used in several Natural Language Processing (NLP) applications. Moreover, dimensionality engineering algorithms such as Principal Component Analysis (PCA) [65, 66] and Independent Component Analysis (ICA) [67] can also be grouped under unsupervised learning and are critical for the extraction of features from raw environmental signals. More recently, unsupervised Locally Competitive Algorithm (LCA) [68] has been shown to be effective for sparse encoding and dictionary learning.

In the context of neural networks, Hopfield Networks [69], inspired by ferromagnetism [70], introduced unsupervised learning by converging to stable patterns as Content Addressable Memory (CAM). Restricted Boltzmann Machine (RBM) [71] expanded on this by sampling neuron states probabilistically, improving unsupervised inference. Moreover, deep belief networks [72], also known as autoencoders, combine several RBMfor unsupervised feature learning. In addition, Helmholtz machines [73], a precursor to the popular variational autoencoders [74], effectively encode input data into probabilistic distributions, refining the ability to infer hidden causes through the unsupervised wake-sleep algorithm by Hilton [75]. Finally, Self-Organizing maps [76] based neural networks have been proven useful for unsupervised feature learning, and have strong influenced recurrent SNNs for sequence learning [77].

However, for implementing such learning in hardware in real-time, one important bottleneck is shuttling the data between different physical locations on the hardware. To this end, the philosophy of local learning is discussed in the following subsection.

2.4.3 Local

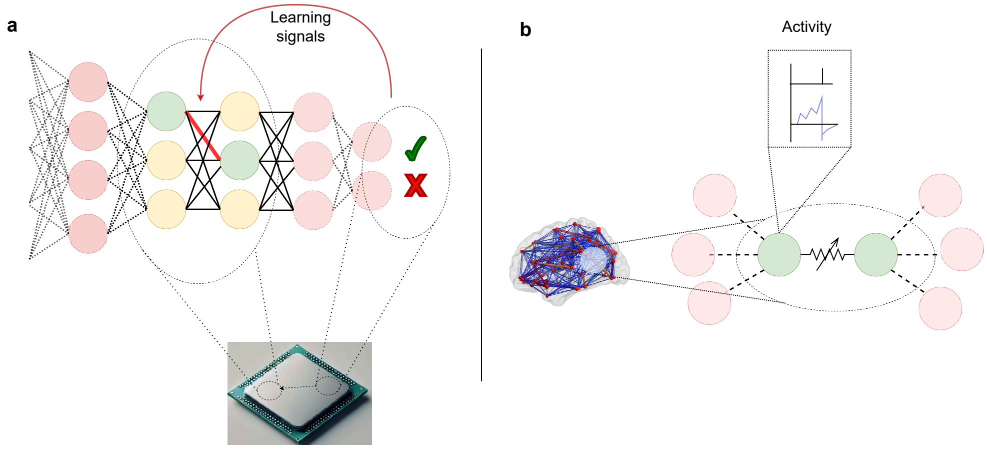


Figure 2.4 Non local vs local learning: **a** Backpropogation of error require transmission of learning signals including correct label, predicted output and weights of subsequent layer to be transmitted to inner layers. On physical chip, this results in transmission of the above signals to far-apart locations. **b** In local learning algorithms, only the local variables: activity (membrane voltage and spike times) of pre-synaptic and post-synaptic neuron are responsible for learning.

For online learning in state-of-the-art deep neural networks, high-dimensional learning signals are needed to be transmitted from the last layer to intermediate layers, as shown in Figure 2.4. Local learning algorithms use locally accessible state variables associated with the immediate neuron, i.e. presynaptic and postsynaptic neuron. The state variable could be, for instance, the spike time or the spike rate, which translates to various STDP and Spike-Rate Dependent Plasticity (SRDP) based learning rules.

Table 2.1 Spike-based local synaptic plasticity rules: comparative table (Reproduced with permission from [78])

Plasticity rule	**Local variables**	**Spikes interaction**	**Update trigger (spike)**		**Synaptic weights**			**Stop-learning**
			LTD	**LTP**	**Type**	**Bistability**	**Bounds**	
STDP [79]	Pre- and post-synaptic spike traces	Nearest spike	Pre	Post	Analog	No	Hard	No
T-STDP [80]	Pre-synaptic spike trace + 2 post-synaptic spike traces (different time constants)	Nearest spike / all-to-all	Pre	Post	Analog	No	Hard	No
SDSP [81]	Post-synaptic membrane voltage + post-synaptic spike trace	All-to-all	Pre		Binary*	Yes	Hard	Yes[1]
V-STDP [82]	Pre-synaptic spike trace + post-synaptic membrane voltage + 2 post-synaptic membrane voltage traces	All-to-all	Pre	Continuous	Analog	No	Hard	Yes[2]
C-STDP [83]	One synaptic spike trace updated by both pre- and post-synaptic spikes	All-to-all	Continuous		Analog	Yes	Soft	Yes[3]
SBCM [84]	Pre- and post-synaptic spike traces	All-to-all	Continuous		Analog	No	Hard	No
MPDP [85]	Pre-synaptic spike trace + post-synaptic membrane voltage	All-to-all	Continuous		Analog	No	Hard	Yes[4]

Continued on next page

Table 2.1 Spike-based local synaptic plasticity rules: comparative table (continued)

Plasticity rule	**Local variables**	**Spikes interaction**	**Update trigger (spike)**		**Synaptic weights**			**Stop-learning**
			LTD	**LTP**	**Type**	**Bistability**	**Bounds**	
DPSS [86]	Pre-synaptic spike trace + post-synaptic dendritic voltage + post-synaptic somatic spike	All-to-all	Continuous		Analog	No	Hard	No
RDSP [87]	Pre-synaptic spike trace	All-to-all	Post		Analog	No	Soft	No
H-MPDP [88]	Pre-synaptic spike trace + post-synaptic membrane voltage	All-to-all	Continuous		Analog	No	Hard	Yes[5]
C-MPDP [89]	Post-synaptic membrane voltage + post-synaptic spike trace	All-to-all	Pre		Analog	No	Hard	No
BDSP [90]	Pre-synaptic spike trace + post-synaptic event trace + post-synaptic burst trace	All-to-all	Post (event)	Post (burst)	Analog	No	Hard	No

[*] Binary with analog internal variable. [1] At low and high activities of post-neuron (post-synaptic spike trace). [2] At low low-pass filtered post-synaptic membrane voltage (post-synaptic membrane voltage trace). [3] At low activity of pre- and post-neurons merged (synaptic spike trace). [4] At medium (between two thresholds) internal update trace. [5] At medium (between two thresholds) post-synaptic membrane voltage.

In context of STDP, there are different variations proposed in past, from the simplest pair-based model which only considers the nearest spike interaction [79] to Triplet-based STDP (T-STDP), which considers triplets of spike proposed in [80] which can also explain the frequency dependence in synaptic plasticity observed in neural cells. Moreover, Calcium-based STDP (C-STDP) [83] also incorporates the calcium variable of the postsynaptic neuron to calculate the weight update to improve learning performance.

These plasticity rules, along with the local variables used, the learning triggers, and the stop learning mechanism, are summarized in Table 2.1. Spike-based local synaptic plasticity rules use various local variables, such as pre- and post-synaptic spike traces and membrane voltages, to govern changes in synaptic strength. Spike interactions can either be nearest-spike or all-to-all, influencing the update mechanism for LTP and Long-Term Depression (LTD), which are typically triggered by the timing of pre- and post-synaptic spikes. These rules differ in how they update synaptic weights, which can be either analog or binary, with some models supporting bistability, allowing the synapse to stabilize in two distinct states. Synaptic weight bounds can be hard or soft, affecting the degree of flexibility in weight changes. Additionally, many models incorporate stop-learning mechanisms that halt further synaptic updates once certain stability conditions are met, allowing the network to consolidate learning effectively.

In addition to spike time, plasticity can depend on the firing rate of the neuron as described in Spiking BCM (SBCM) [84] and Rate Dependent Synaptic Plasticity (RDSP) [87]. Another class of learning rule like Spike-Driven Synaptic Plasticity (SDSP) [81], Membrane Potential Dependent Plasticity (MPDP) [85], Calcium-based MPDP (C-MPDP) [89], Homeostatic MPDP (H-MPDP) [88], and Voltage-based STDP (V-STDP) [82] are based on the membrane potential of the neuron. This learning simplifies the circuit by avoiding the storage of spike time in the form of an activity trace. However, the spike trace of one neuron, referred to as the calcium variable, is still used to calculate the weight update in some rules. More recently, Burst-Dependent Synaptic Plasticity (BDSP) [90] also incorporates high-frequency neuron bursts to enable learning in hirerchal networks, where bursts in pyrimidal neurons coordinate plasticity in lower layers.

Although STDP and other Hebbian local learning rules are biologically plausible and hardware friendly for online learning, they suffer from a scalability challenge when deployed on large-scale multilayer networks to solve complex problems as they do not guarantee error minimization. Moreover, in biological and artificial neural networks, updating the weights on every spike event would be energetically inefficient [91] and there is a strong possibility of a third factor acting as a neuromodulator to trigger Hebbian learning. For example, in Dendritic Prediction of Somatic Spiking (DPSS) [86], dendritic potential serves as the third factor. In biological systems [92, 93], the dopamine, noradrenaline, and acetylcholine molecule regulates synaptic plasticity triggered by novelty (or surprise) or reward(or punishment) [94]. However, in the context of artificial SNNs, such learning signals should essentially assign spatio-temporal credit based on neuron activity and global reward [95]. The added temporal dimension demands a mechanism to assign a reward to individual synapses based on the activity in the past that resulted in the reward.

2.4.4 Circuits for learning

Table 2.2 Neuromorphic circuits for spike-based local synaptic plasticity models. (Produced with permission from Khasef et al. [78])

Rule	Paper	Difference with the model	Implementation
STDP	[96][1]	/	0.6 µm Fabricated
	[97]	All-to-all spike interaction + bistable weights	1.5 µm Fabricated
	[98]	/	0.6 µm Fabricated
	[99]	Anti-STDP + Non-exponential spike trace	0.35 µm Fabricated
	[100]	Bistable weights	1.6 µm Fabricated
	[101][2]	All-to-all interaction + binary weights	0.25 µm Fabricated
	[102]	Soft bounds	0.6 µm Fabricated
	[103]	All-to-all spike interaction + asymmetric bounds (soft lower bound + hard upper bound)	0.35 µm Fabricated
	[104]	/	0.25 µm Fabricated
	[105]	All-to-all spike interaction	0.35 µm Fabricated
	[106]	All-to-all spike interaction + asymmetric bounds (soft lower bound + hard upper bound)	0.35 µm Fabricated
	[107]	/	0.15 µm Simulated
T-STDP	[108]	/	Simulated
	[109]	/	0.35 µm Simulated
	[110]	/	0.35 µm Fabricated
SDSP	[111]	No post-synaptic spike trace + no stop-learning mechanism	1.2 µm Fabricated
	[112]	No post-synaptic spike trace + no stop-learning mechanism	0.6 µm Fabricated
	[113]	No post-synaptic spike trace + no stop-learning mechanism	0.6 µm Fabricated
	[114]	Analog weights	0.35 µm Fabricated
	[115]	Analog weights	0.35 µm Fabricated
	[116]	Analog weights	0.35 µm Fabricated
C-STDP	[117]	Hard bounds	0.18 µm Fabricated
RDSP	[118]	Nearest spike interaction + reset of pre-synaptic spike trace at post-spike + very small soft bounds	2 µm Fabricated
	[119]	Nearest spike interaction + asymmetric bounds (soft lower bound + hard upper bound)	0.35 µm Fabricated

[1] Potentiation and depression triggers done with digital logic gates.

[2] Weight storage in digital SRAM.

In the last two decades, neuromorphic Very Large Scale Integration (VLSI) circuits have been proposed with incremental complexity, focused on addressing the challenges of scalability, efficiency, and functional integration. Table 2.2 summarizes the proposed circuit implementations of unsupervised local learning rules. The proposed circuits try to reproduce a biological variant of the local learning rule (STDP, T-STDP, SDSP, C-STDP, and RDSP), with some modifications implied by the circuit and devices, such as hard/soft bounds, weight resolution (binary/analog), and stop learning mechanisms.

The first work by Hafliger et al. [118] in 1996 introduced RDSP circuit using a 2 µm technology node. This study focused on basic spike processing with a single-neuron circuit, laying the foundation for more complex designs.

Bistability of synapses In 2000, Fusi et al. [111] implemented SDSP in a 1.2 µm node, with synaptic circuit composed of 18 transistors. This study demonstrated that silicon-implied on-line learning is robust to variations in transistors, due to continuous adaptation of binary synaptic weights. Chicca et al. [112] and Bofill et al. [96] in 2001 explored SDSP and STDP using smaller technology nodes (0.6 µm). Chicca's work focused on stable learning models with stochasticity driven by neuron using a test chip composing 21 neurons, while Bofill emphasized precision in temporal asymmetry learning. By 2002, Indiveri [97] demonstrated stable synaptic learning in a 1.5 µm node, focusing on long-term bi-stability properties, essential for engineering synapses with smaller area footprint. Arthur et al. [101] in 2005, impleted learning models with binary weight STDP using a 0.25 µm node, demonstrating learning with more than 1000 on-chip neurons, emphasizing robustness to variability in spike times.

Complexity and Adaptation The work of Chicca et al. [113] on SDSP further integrated long-term memory capabilities into VLSI circuits, employing a recurrent network on a 0.6 µm node. Indiveri et al. [120] in 2006 implemented STDP in a 1.6 µm node, with test chip comprising 32 neurons and 256 synapses. In particular, the neuron was low power, configurable, exhibeted homeostasis through spike frequency adaptation, and the digital Address Event Representation (AER) based communication protocol was implemented.

Power efficiency through non-volatile synapse Liu & Möckel [103] in 2008, introduced floating-gate STDP in a 0.35 µm node, focusing on non-volatility of weight, long time scale operation upto seconds, and conditional weight update circuit which was activated only on detection of corelated spike pairs. Ramakrishnan et al. [119] demonstrated effective STDP and stable long-term learning with floating gate synapses using 0.35 µm nodes with 20,000 synapses.

Scalability and Integration Chicca et al. [121] further advanced the field by integrating cognitive abilities into multifunctional VLSI circuits, supporting 128 neurons and 4096 synapses. Gopalakrishnan et al. [110] introduced T-STDP for robust memory retention, while Huayaney et al. [117] implemented C-STDP using a 0.18 µm node to ensure stable synaptic plasticity.

Applications Cameron et al. [99] demonstrated the learning efficiency for the visual pattern recognition task. Furthermore, Koickal et al. [102] in 2007 expanded the application domain by developing an adaptive mechanism for real-time olfaction response in a sensory array. Tanaka et al. [104] demonstrated STDP for retrival of associative memory using the recurrent Hopfield network topology implemented in 0.25 µm node.

The following section presents entire hardware systems, highlighting the key differences with respect to conventional computing architecture.

2.5 Neuromorphic hardware systems

Modern processors leverage the fast-switching capabilities of advanced, scaled transistors, such as those in the 2nm range, to execute complex tasks with remarkable speed and energy efficiency. By breaking down tasks into multiple instructions that are executed sequentially at GHz clock speeds [122], these digital processors can achieve billions of instructions per second. In contrast, neuromorphic systems, designed to mimic the brain's architecture, solve equations for all neurons and synapses simultaneously and in real-time, providing highly efficient processing for complex tasks. However, the nature of spiking neurons, which operate along a temporal dimension and often exhibit recurrent behavior, challenges the conventional divide-and-conquer approach typical of pipelined hardware [123]. Therefore, a hybrid approach combining serial and parallel processing is crucial: while sparse spikes are effectively handled in a serial manner over time, operations such as multiply and accumulate are more efficiently processed in parallel.

Moreover, SNNs operate fundamentally differently from traditional digital processors, as they rely on event-driven neural spikes rather than the continuous fetch-and-execute cycles of digital computation. This shift requires specialized hardware, as the precision in SNNs is encoded in the timing of spikes rather than in traditional numerical representations. Given that spikes are sparse and occur infrequently, using high-speed clocks typical of digital processors is inefficient. However, to accurately capture temporally clustered spikes, such as bursts [124], it is crucial to avoid missing these events. Consequently, asynchronous operation becomes essential for SNN hardware, allowing the system to respond dynamically to spikes as they occur rather than being tied to a fixed clock rate.

The journey toward neural processors began with Intel's ETANN 80170NX in the 1980s, which used analog circuits for neural functions [125]. This was followed by digital chips like the Nestor/Intel Ni1000 [126] and the exploration of FPGA-based accelerators for neural networks in the 1990s [127] [128].

Table 2.3 Neuromorphic hardware systems using digital circuits

Name	Ref	Year	Node (nm)	Area (mm^2)	Mapped topology	On-chip learning	Task (dataset)	Accuracy (%)	E/sample	Throughput [samples/s]
SpiNNaker	[129]	2013	130	88.4	784-500-500-10	Flexible	MNIST	95	6mJ	50
SpiNN. 2	[130]	2021	22	9	390-256-256-29	Flexible	Keyword spotting	93.80	7.1μJ	1k
Loihi	[131]	2018	14	60	390-256-256-29	Flexible	Keyword spotting	93.80	270 μJ	296
MorphIC	[132]	2019	65	2.86	4x (196-500-10)	Stoch. SDSP	MNIST	95.90	21.8 μJ	250
Chen et al.	[133]	2018	10	1.72	236-20	STDP	MNIST	88	1 μJ	6.25k
Seo et al.	[134]	2011	45	0.78	256-256	Stoch. STDP	Pattern recall	N.A.	N.A.	N.A.
ODIN	[135]	2018	28	0.086	256-256	SDSP	EMG gesture	53.60	7.4μJ	42.5
Knag et al.	[136]	2015	65	3.06	4x64	SAILnet	Custom	N/A	109nJ	62.5k
Kim et al.	[137]	2015	65	1.8	4x64	SGD (last layer)	MNIST	90	27nJ	9.9M
Park et al.	[138]	2019	65	10.1	784-200-200-10	Mod. segr. dendrites	MNIST	97.80	236nJ	100k

One of the first digital neuromorphic systems were the SpiNNaker SNN simulation platform, introduced in 2013 [129], and IBM's TrueNorth processor, released in 2014 [139]. Intel followed with its Loihi chip in 2017 [131]. Each of these systems has since evolved, and successors have emerged in subsequent years. SpiNNaker 2 [130], Loihi 2 [140], and IBM's Northpole processor [141]. It is important to note that the SpiNNaker platform stands apart from the other two chips in its design philosophy, aiming to achieve very large-scale neural simulations without stringent energy constraints, primarily through the use of stacked ARM cores. Other notable designs from various research groups are summarized in Table 2.3, detailing their CMOS technology node, area, mapped topology, and demonstration capabilities.

However, digital systems face challenges due to the **overhead** of **signal domain conversion** and the significant differences with respect to the requirements of neuromorphic signal processing, which is arguably more similar to analog computing.

2.5.1 Analog domain

"The digital computers considered in the last section may be classified amongst the 'discrete state machines'. These are the machines which move by sudden jumps or clicks from one quite definite state to another. These states are sufficiently different for the possibility of confusion between them to be ignored. Strictly speaking there are no such machines. Everything really moves continuously. But there are many kinds of machine which can profitably be thought of as being discrete state machines. For instance in considering the switches for a lighting system it is a convenient fiction that each switch must be definitely on or definitely off."– Alan turing in [142].

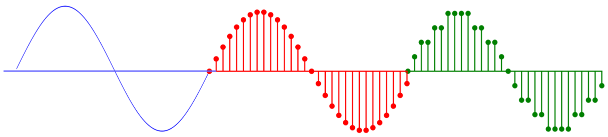


Figure 2.5 Analog to digital domain conversion. Natural signals are analog, and are continuous in amplitude and time. These are captured at regular time intervals to create sampled signals. Thereafter, the amplitude of the signals is quantized to produce digital signals for storage and processing.

Digital circuits primarily use transistors as switches and are often limited by challenges such as charge leakage in DRAM and capacitive charging in inverters, which can constrain ultra-fast instruction execution. Interestingly, neuromorphic systems efficiently exploit these same phenomena for time-based operations, turning what are constraints in digital systems into advantages in analog computing. This is especially beneficial since natural signals are inherently analog, allowing analog computing to avoid the overhead associated with domain conversion. Modern spiking neural networks use binary-like spike signals to reduce noise, thereby enhancing the reliability of inherently noisy analog hardware[143]. These unique properties of neuromorphic engineering help overcome the historical challenges of noise and variability in analog computing [144].

The concept of analog neuromorphic systems dates back to the 1990s, beginning with pioneers such as Carver Mead, Misha Mahowald, and Rodney Douglas [145]. Significant platforms like the wafer-scale BrainScales [146, 147], developed as part of the Human Brain Project, and Stanford's Neurogrid [148] and Braindrop [149] followed. The major analog neuromorphic hardware platforms are tabulated in Table 2.4.

Table 2.4 Overview of analog neuromorphic systems.

Name	Ref	Year	Node (nm)	Area (mm^2)	Mapped topology	On-chip learning	Task (dataset)	Time Scale
DYNAPs	[150]	2017	180	38.6	8-192-3	N/A	EMG Gesture	Real-time
Brink et al.	[151]	2012	350	21.7	300-10	STDP	N.A.	Biological
Mayr et al.	[152]	2015	28	0.36	128-64	SDSP	N.A.	Biological
ROLLS	[153]	2015	180	44	256-256	SDSP	2-class Caltech101	Real-time
HICANN	[146]	2010	180	49	2x (224)-256	STDP	5-class MNIST	Accelerated
HICANN-X	[154]	2022	65	27.9	256-512	Flexible	MNIST	Accelerated
Neurogrid	[148]	2014	180	168	1M	-	-	Real-time
BrainDrop	[149]	2018	28	0.65	4096 neurons	-	-	Real-time

Wafer-scale, accelerated time In 2010, the HICANN (High Input Count Analog Neural Network) chip [146] was launched under the FACETS project for accelerated neural simulations, resulting in wafer-scale analog computing platform: BrainScales. 325 HICANN chips are placed on 20cm wafer, and several of the wafer can be further connected. The succesor, BrainScales2 system made up of the HICANN-X chip [154], was launched in 2019. The system was implemented in 65nm CMOS technology and thus integrated more circuits as compared to the 180nm first version. A dedicated digital co-processor for deployed to supporting plasticity in accelerated neural system, allowing implementation of online learning in complex neuron networks. This is made possible by a high degree of configurability with its programmable synapses and flexible neuron models. However, these accelerated wafer-scale analog computing systems are meant for simulating large-scale network models, and thus depart from the design philosophy of small-scale systems meant for real-time deployment on edge computing devices.

Real-time, modular, analog and mixed-signal In 2014, Neurogrid [148] introduced a large-scale analog and mixed-signal system designed for real-time neural simulations, featuring multi-chip integration. The key innovation was architectures with shared or multiplexed components to allow flexible architectures, and increasing the size of deployable network. The system's multi-chip communication employed a tree-like routing architecture for efficient data transmission, and it explored four different architectures with various levels of multiplexing, incorporating shared axon, dendrite, and synapse cells to optimize neural network simulations. In terms of computation, the system used custom biophysical neuron models that allow real-time simulation of millions of neurons on multiple chips. Neurogrid's hybrid analog and mixed-signal architecture combined the compactness of analog computing with the deterministic communication of digital systems.

The second-generation chip, BrainDrop [149], introduced in 2017, further innovated by offering a higher level of abstraction for easier application deployment. Users define the non-linear dynamic system, which is implemented via the Neural engineering framework (NEF) on analog circuits. This mapping is crucial given the challenges posed by mismatches in scaled transistors operating in the sub-threshold regime, where small bias currents are used to minimize power consumption.

Learning enabled systems Brink et al. [151] in 2012, introduced a learning-enabled neuron array designed for biological realism with high synapse resolution, allowing for detailed simulations of neurons and their connections that closely mimic biological processes. Although the system supported learning mechanisms, its primary focus was not real-time learning or adaptability, but rather on improving biological accuracy. The high synapse resolution provided fine control over neural connections, making it highly suitable for representing complex neural networks. This system was intended for researchers to study biologically realistic neural circuits, focusing on fixed simulations rather than real-time adaptability or dynamic reconfiguration.

In 2015, ROLLS [153] was introduced for real-time on-chip learning with a configuration of 256 neurons and 128K synapses. This system allowed for dynamic reconfiguration, enabling the network to be adapted in real time for different application purposes, making it highly versatile. Online learning in synapses was implemented through SDSP rule [81], and the neurons exhibited homeostasis through Adaptive exponential integrate-and-fire (AdExpIF) neurons. In ROLLS, more than 90% of the silicon area was occupied by synapse and less than 1% by neurons. This points to the fact that significant **optimizations have already been made with respect to the area and energy consumption of the analog neuron**. The **emerging challenge** to engineer large scale system lies in efficient **communication** between neural cores and scaling down the **memory**.

The primary scaling limitation in these systems arises from the complex and costly task of routing spikes to different neurons. Biologically inspired neural networks have a high fan-in, where each neuron receives inputs from thousands of others. Additionally, the neuron models and populations are often hard-wired, making it difficult to deploy different architectures such as CNNs and RNNs. Another significant constraint is the power consumed in fetching weights and transmitting spikes to subsequent layers, a process central to both network operation and online learning. Furthermore, capacitive memories implemented in CMOS occupy the majority of the silicon area and are challenging to scale due to physical limitations. These issues underscore the need for memory-centered architectures, which are discussed in the following section.

2.5.2 Memory centered architectures

Neuromorphic systems require efficient weight-fetch operations for online learning, addressing memory access challenges through near-memory and in-memory processing. These approaches enhance performance and efficiency by minimizing data transfer delays, making efficient memory access critical for realizing large-scale neural systems that operate in real-time with low power consumption. It is important to note that data shuttling between memory and the processor accounts for the majority of the energy budget in embedded AI systems. Innovations in near-memory and in-memory computing have been pivotal in reducing the memory-processor bottleneck, effectively coupling memory and computation similarly to how the brain's neuron synapses operate, thereby improving overall system efficiency. Moreover, for plasticity in such systems, the model weights must be fetched from the memory to processor to calculate the weight update. In-memory learning proposes to solve the latency and energy limitations by updating the weights directly in memory with local learning signals during inference.

Heterogeneous memory structure In 2017, DYNAP-SE [150] introduced low-power, scalable processors for real-time processing using LIF neurons with bio-mimetic dynamics.

The chip featured hybrid mesh and tree routing structures to optimize circuit overheads and latency through two levels of memory: Static Random Access Memory (SRAM) and CAM based on 8-T NOR memory cell. The second generation, DYNAP-SE2 [155], improved upon the original design, which featured 256-neuron cores with basic integrate-and-fire models. SE2 introduced more advanced models like the AdExpIF neuron, enabling biologically realistic behaviors such as spike-frequency adaptation and firing rate homeostasis, supporting dynamic real-time learning. While the first iteration used linear synapses, SE2 enhanced short-term plasticity and added synaptic filters mimicking NMDA and AMPA receptors, improving spiking network simulations. Additionally, SE2's local connectivity architecture evolved into a hierarchical 2D-grid routing system, allowing for low-latency communication across multiple chips, making it ideal for larger, more complex SNNs.

2.5.3 Emerging devices for the post-Moore era

Apart from the architecture point of view, there is a physical limitation for technology node scaling. Semiconductor manufacturing could be broken down into 3 basic processes, deposition, lithography, and etching. In this case, lithography is basically the patterning of computationally designed structures to metal/insulator stacks. The resolution of patterning decides the feature size of different electronic devices such as transistors and resistors. Until the last decade, aggressive scaling of CMOS nodes happened in agreement with Moore's law [156] and Dennar's MOSFET scaling law [157]. However, the latest 2nm node is very close to the thickness of a single layer of atoms; moreover, as transistor sizes continued to decrease, process variations became a significant challenge, particularly for sub-32nm nodes, as detailed in the 2011 study by Kuhn [158], limiting integrated circuit dependability.

To overcome the limitations of CMOS scaling, 3D NAND flash memory has been developed, achieving a projected memory area density of 10 Gb/mm^2 or 1 kb/ µm^2, compared to 1 Gb/mm^2 or 100 b/µm^2 for its 2D counterpart (15nm) in 2016 [159]. For DRAM cells, predictions for 2019 indicated that a 16nm pitch could yield a density of 46 GB/cm^2. More recently, novel device technologies such as emerging non-volatile resistive memory with nanoscale footprints have become promising alternatives to overcome CMOS scaling limitations. Memristive devices, which were first proposed as a fourth circuit element [160] and later realized [161], are particularly noteworthy. These devices can be scaled to achieve memory densities as high as 460 GB/cm^2 [162] and can be fabricated with dimensions as small as 2nm [163]. With analog switching characteristics, memristive devices closely resemble human synaptic models [164], making them highly suitable for synaptic realization in hardware-implemented neural networks [16].

In-memory computing using memristive devices in a memory crossbar architecture can be realized through Ohm's and Kirchhoff's laws to implement Vector Matrix Multiplication (VMM), a key operation in artificial neural networks for computing neuron activations [165, 166]. In the context of SNNs, the vectors represent the spikes transmitted by input neurons, while the weights correspond to the synaptic conductance of individual memory devices. The 3D stackability of memristive devices [167] enables high-density network integration by leveraging physical principles such as redox reactions, ferroelectricity and magnetism, providing energy-efficient computation.

2.5.4 Demonstrations with Crossbar Arrays

Offline Learning Valentian et al. [168] developed a fully integrated spiking neural network with analog neurons and RRAM synapses in a 13.5k array. Their use of offline gradient descent and quantization learning methods, combined with the analog implementation, supported significant accuracy on the MNIST dataset, facilitated by the high density of 1T1R synapses. Wan et al. [169] utilized RRAM in a 256 × 256 array, implementing an Integrate-and-Fire neuron model in CMOS. Their approach achieved good efficiency in offline learning, particularly for probabilistic graphical models (PGMs) in the MNIST reconstruction task.

Gradient-based Online Learning In the work by Burr et al. [170], supervised learning using Phase Change Memory (PCM) with a 500 × 661 array was demonstrated within a multi-layer perceptron topology. Their gradient-descent and backpropagation learning schemes, controlled via software, achieved strong training performance on the MNIST dataset. Furthermore, in optimizing RRAM devices for neuromorphic systems, Wu et al. [171] applied this technology to a 1Kb array with differential weight topology. Their implementation of online gradient descent learning with CMOS-based I&F neurons demonstrated effective learning performance on the YaleFace dataset.

Crossbar in Loop Learning and CNN Integration In [172], demonstration with artificial 4x4 pattern with **STDP** and 1T1R-based synapse has been demonstrated. For simple patterns, the **binary** states of RRAM were sufficient to achieve 100% recognition accuracy. Importantly, the functionality of the neuron is emulated with an **off-chip** microcontroller. The implementation of off-chip neuron and learning requires signal conversion, and Analog to Digital Converter (ADC) and Digital to Analog Converter (DAC) account for more than 90% of the energy expenditure [173]. The study was extended to another **simulation**-based study [174] where the models were fitted from [172] to perform MNIST classification with 784x50000x10 fully connected network topology with synapses between 1st and second layer trained with STDP while the ones between 2nd layer and third layer of neuron with supervised learning. The recognition rate in the test set was 92%, and interestingly, **noise** in the input layer was optimized for STDP learning performance.

In [175], semi-supervised learning was proposed in a hybrid Convolutional Neural Network (CNN)-FC architecture. CNN weights were trained through **supervised learning**, and **STDP** was used for unsupervised training of the fully connected layer. Ten 8x8-size arrays were used to implement CNN kernels. An external ADC was used to read the CNN state and feed spikes to the fully connected network crossbar. CNN is implemented using 20x20 filters in front-end-of-line integrated memories for VMM. The weights of the convolutional layer are trained through gradient descent. The VMM outputs were processed by **FPGA** with neurons and sent to another 8x8 crossbar for classification. STDP-based learning was implemented in the classification layer.

On-Chip SNN and Mixed-Signal Implementation In [176], learning on chip was demonstrated through **pulse overlapping** of rectangular pulses for 2-layer SNN. A dual core chip with LIF neurons and 6T2R synapses was manufactured for a signed multi-bit weight representation. The restricted Boltzmann machine and event-driven **contrastive divergence** were used with **STDP** to perform the MNIST classification.

In [177], RRAM based SNN for MNIST recognition on 256x32 2T2R crossbar, 32 **digital** neurons, routing logic, IV converters, and learning circuit on **2.25mm2** 180nm CMOS integrated circuit was reported with energy consumption of **12.5 pJ/synaptic operation**, static power consumption of **6.4 uW** and accuracy of 98.3%. The synaptic weight update rule is **Precise-spike-driven (PSD)** rule [178, 179]. The weight update occurs at every teacher neuron spike or output neuron spike, the polarity of the weight update is based on the sign of error between the output neuron spike and the teacher spike, and the magnitude is dependent on the membrane potential of the post-synaptic neuron. The neurons are digital and communicate with a crossbar with the peripheral circuit to read analog currents and feed spikes to input layer neurons. The energy consumption of the digital circuit was **14.09mW**, the RRAM crossbar was **6.4 uW** with a write voltage of 0.2V and **12.17 mW** for analog circuits, including ADC, IV converters, and switches. It needs to be clarified whether the results are from post-layout simulations or taped-out chips, as memory integration is not discussed. Interestingly, a phase-based input encoding scheme was used to generate at most one spike per sample.

Table 2.5 Summary of memristive crossbar based physical implementation of neuromorphic systems comparing array size, learning algorithm, and neuron model.

Ref	**Device**	**Array Size**	**Topology**	**Learning**	**Neuron model**	**Neuron Implementation**	**Synapse**	**Spiking**	**Dataset**	**Source of Learning Signal**
[169]	RRAM	256x256	RBM	PGMs, Offline	I&F	CMOS (Analog)	1T1R	Yes	MNIST reconstruction	Hardware (In-hardware)
[168]	RRAM	13.5k	144x10	Offline gradient descent, quantization	LIF	CMOS (Analog)	1T1R	Yes	MNIST	Software
[175]	RRAM	20x20	CNN and FC	Gradient-descent, STDP, Online	SFA	FPGA	1T1R	Yes	CIFAR-10	Software (FPGA)
[176]	PCM	692 k/core	832x832 RBM	STDP, eCD, Online	I&F	CMOS (Analog)	6T2R	Yes	MNIST	Hardware (On-chip)
[180]	PCM	256x256	256x256	STDP, Online	I&F	CMOS (Analog)	2T1R	Yes	Custom patterns	Software
[170]	PCM	500x661	528x250 x125x10	Gradient-descent, Backprop, Offline	Non-spiking	Software	1T1R	No	MNIST	Software
[171]	RRAM	1Kb	320x3	Online Gradient Descent (only sign), Online	Non-spiking	CMOS (Digital)	1T1R Differential	No	YaleFace	Software

Current-Controlled Switching In [180], **PCM**-based synapses are programmed using **STDP** with **pulse overlapping**. Current controlled switching was proposed, and the pulse shape was generated **off-chip**. Custom patterns were used to demonstrate learning. The STDP time scale and neuron leak constant are in the sub millisecond range. In [181], a single memristor was connected to two neurons implemented through CMOS circuits. The neurons are engineered to generate a bi-triangular pulse, and the weight update was performed on a single synapse using the pulse overlapping technique. [182] implements the pulse engineering circuit to implement STDP on memristors in 28nm Fully Depleted Silicon On Insulator (FD-SOI). The area occupied by a pulse generator was **703 µm^2** to generate a pulse between 8 µs and 100 ms.

Table 2.5 provides a summary of the hardware and on-chip crossbar-in-loop learning demonstrations, highlighting aspects such as neuron implementation (software/FPGA/ASIC), synapse architecture, dataset, and learning methods.

2.6 Conclusion

Neuromorphic computing for designing energy-efficient intelligent embedded systems. The brain-inspired computing paradigm shows promise for developing unconventional computing systems. These specialized systems incorporate bio-inspiration in memory and time-domain to create asynchronous parallel computing systems. The temporal dimension is well-suited for real-time computing in devices with limited power budgets. Energy consumption is reduced through analog computing and devices that use the physical phenomenon of electronics to emulate complex non-linear dynamics in real-time. Additionally, the in-memory computing architecture further reduces energy consumption and increases integration density with nanoscale memory devices. Memristive devices can be used for 3D stacking. However, the challenges of 3D stacking and vertical integration, such as the significant heat generated by densely packed transistors, emphasize the need for low-power, analog operation. High-speed digital switching, which involves millions of transistors toggling between ON and OFF states multiple times per millisecond, requires advanced cooling technologies that are difficult to implement in 3D architectures. Therefore, low-power and analog operations are not just optional features, but necessities in modern computing hardware. Important features which makes neuromorphic engineering efficient: distinctions in time representation: **temporal** domain computing, and **memory**.

Memory and learning Memory is a critical component of AI hardware, as neural networks are data-intensive, with millions of weights and streaming input signals. The memory storage process is essentially learning. Online learning is essential for continuous learning during the agent's life cycle. For online learning, unsupervised and local learning is critical because expert annotation is not available in real time. Implementing complex learning physically across chips poses a challenge in meeting the real-time computing-learning requirement. Hebbian learning is an attractive option for online local unsupervised learning, and different model circuits have been proposed.

However, we emphasize that such unsupervised learning algorithms are a new and active area of research for improvement, as digit recognition using gradient descent was first demonstrated in the early 1990s, while a similar counterpart to STDP and SNN only emerged in the 2010s. The increasing use of unannotated raw environmental signals and the growing collection of IoT data in the current decade seem to align very well. Unsupervised online learning holds promise for improving the generalization capabilities of neural networks in real-world deployment and usage of raw signals [183].

Hardware realization Hardware implementation is required due to the previously mentioned differences in memory and time. Digital systems have been proposed in the past, but face challenge due to the energy-delay overhead of signal domain conversion. Analog domain computing has greater similarities with neuromorphic models and utilizes physical principles like Ohm's law and charge integration for performing multiply and accumulate operations.

Architecture and device More recently, memory-centric architecture and specialized devices have been introduced for scalable real-time learning and computing. Memristive devices are nanoscale, nonvolatile memory devices programmed by using physical conductance change mechanisms such as redox reactions, phase change, and ferroelectric domain switching. STDP-based learning has been proposed and validated on various memristive devices, including HfO_2 based memristors [184, 185, 186, 187], nano-composite memristive devices (ON/OFF ratio > 1000)[181], STOx based resistive memories [188], ferroelectric memories [189, 190, 191], FeFET (industrial node 28nm) [192], and STT-MRAM [193]. Most of the proposed approaches first realize the physical implementation of a single synapse and then analyze the learning performance at the system level through simulations with the model fitted from device experiments [192].

Circuit-algorithm co-engineering challenge. Several studies have explored various learning strategies for neural hardware, including offline learning with weight transfer for inference hardware, online learning using gradient descent for ANN implementations, and crossbar-in-loop learning with STDP. Online learning is particularly beneficial for memristive devices as it helps mitigate non-idealities inherent in these systems. However, in simulation-based studies, there is a trade-off between the scalability and model accuracy of CMOS and RRAM models. Moreover, implementing neurons and peripheral circuits on separate silicon chips presents scalability challenges due to the limited number of I/O connections. This separation also leads to significant inefficiencies, as signal conversion and transfer account for more than 90% of the system's power consumption [173]. Thus, while some of the unsupervised learning mechanisms are already demonstrated on discrete memristive elements but require a demonstration at the network level in order to assess their effectiveness.

It is important to emphasize that even the simplest version of Hebbian learning: STDP is not trivial to implement on hardware, as accurate spike times are needed to calculate the weight update, which may not be readily available. Previous studies have used neuron activity traces to address this challenge, necessitating additional hardware overhead, such as capacitors for charge storage, digital registers, and update logic. Developing hardware for spiking neural networks that can perform analog in-memory computing with online learning remains a algorithm-circuit co-design challenge. This thesis aims to tackle this challenge by first creating an unsupervised learning rule specifically for memristive devices. The next step will be to design an analog spiking neuron and then develop a mixed signal computing chip that integrates analog neurons, memristive devices, and a specialized architecture to enable unsupervised online learning.

CHAPTER 3

Voltage-dependent synaptic plasticity: Unsupervised probabilistic Hebbian plasticity rule based on neurons membrane potential

"The brain is the organ of destiny. It holds within its humming mechanism secrets that will determine the future of the human race." — *Wildor Hollingworth*

TABLE OF CONTENTS

3.1 Preface

Contribution to document

The following chapter (journal article) represents the first step towards the objective of this thesis: integrating learning into memristive neuromorphic systems. Hebbian principles are appealing for hardware-based learning as they are unsupervised and rely on local variables. In the widely used Hebbian learning rule, Spike-Timing Dependent Plasticity (STDP), the local variable is the timing of spikes from presynaptic and postsynaptic neurons. However, spike times are not inherently stored, which creates additional overhead in tracking activity traces to compute effective weight updates. To address this, we propose a Voltage Dependent Synaptic Plasticity (VDSP)-based learning approach.

Through mathematical derivation, we showed that it is possible to accurately determine when a neuron fires based on its membrane voltage. This model was implemented in Nengo's Spiking Neural Network (SNN) simulation framework. We evaluated the performance of unsupervised learning by training the network to recognize handwritten digits using SNN simulations.The recognition rate achieved with VDSP learning was similar to previous studies using STDP. However, VDSP simplifies hyperparameter tuning by eliminating the need to adjust the temporal sensitivity window, which is a requirement in STDP. The temporal response of VDSP learning depends on the input neuron's behavior and remains robust to variations in the firing rates of input pixels, whereas STDP performs optimally only when the temporal windows align with the input frequency.

The VDSP learning approach uses the spiking activity of the postsynaptic neuron to **trigger** the weight update, while the **membrane potential** of the presynaptic neuron determines the polarity and magnitude of the update. These trigger and state variables provide the essential framework for implementing the learning rule in circuits, as discussed later in chapter 6. This article lays the basis for simulations based on memristive devices, which will be explored in chapter 4. Furthermore, neuronal characteristics—particularly bidirectional leakage—motivate the design of the neuron circuit described in chapter 5.

Title: Voltage-dependent synaptic plasticity: Unsupervised probabilistic Hebbian plasticity rule based on neurons membrane potential

Title in French: Plasticité synaptique voltage-dépendante : Règle de plasticité hebbienne probabiliste non supervisée basée sur le potentiel de membrane des neurones.

Authors: Nikhil Garg[1,2,3,*], Ismael Balafrej[1,2,4], Terrence C. Stewart[5], Jean-Michel Portal[6], Marc Bocquet[6], Damien Querlioz[7], Dominique Drouin[1,2], Jean Rouat[1,2,4], Yann Beilliard[1,2], Fabien Alibart[1,2,3,*]

Affiliations:

1. Institut Interdisciplinaire d'Innovation Technologique (3IT), Université de Sherbrooke, Sherbrooke, Québec, Canada
2. Laboratoire Nanotechnologies Nanosystémes (LN2) – CNRS UMI-3463, Université de Sherbrooke, Sherbrooke, Québec, Canada
3. Institute of Electronics, Microelectronics and Nanotechnology (IEMN), Université de Lille, Villeneuve-d'Ascq, France
4. NECOTIS Research Lab, Department of Electrical and Computer Engineering, Université de Sherbrooke, Sherbrooke, Québec, Canada
5. National Research Council Canada, University of Waterloo Collaboration Centre, Waterloo, Ontario, Canada
6. Aix-Marseille Université, Université de Toulon, CNRS, IM2NP, Marseille, France
7. Université Paris-Saclay, CNRS, Centre de Nanosciences et de Nanotechnologies, Palaiseau, France

***Corresponding Authors:**

- Fabien Alibart – Fabien.Alibart@Usherbrooke.ca
- Nikhil Garg – Nikhil.Garg@Usherbrooke.ca

Journal: Frontiers in Neuroscience

Date: October 2022 (Publication) [194]

Résumé

Cette étude propose la plasticité synaptique dépendante du voltage (VDSP), une nouvelle règle d'apprentissage local non supervisé inspirée du cerveau pour la mise en œuvre en ligne du mécanisme de plasticité de Hebb sur le matériel neuromorphique. La règle d'apprentissage VDSP proposée met à jour la conductance synaptique sur le pic du neurone postsynaptique uniquement, ce qui réduit d'un facteur deux le nombre de mises à jour par rapport à la plasticité dépendante du timing du pic standard (STDP). Cette mise à jour dépend du potentiel de membrane du neurone présynaptique, qui est facilement disponible dans le cadre de la mise en œuvre du neurone et ne nécessite donc pas de mémoire supplémentaire pour le stockage. De plus, la mise à jour est également régularisée sur le poids synaptique et empêche l'explosion ou la disparition des poids lors de stimulations répétées. Une analyse mathématique rigoureuse est effectuée pour établir une équivalence entre VDSP et STDP. Pour valider les performances au niveau du système de VDSP, nous formons un réseau neuronal à pics monocouche (SNN) pour la reconnaissance des chiffres manuscrits. Nous rapportons une précision de 85,01 ± 0,76 % (moyenne ± écart type) pour un réseau de 100 neurones de sortie sur l'ensemble de données MNIST. Les performances s'améliorent lorsque la taille du réseau est mise à l'échelle (89,93 ± 0,41 % pour 400 neurones de sortie, 90,56 ± 0,27 pour 500 neurones), ce qui valide l'applicabilité de la règle d'apprentissage proposée pour les tâches de reconnaissance de formes spatiales. Les travaux futurs porteront sur des tâches plus complexes. Il est intéressant de noter que la règle d'apprentissage s'adapte mieux que STDP à la fréquence du signal d'entrée et ne nécessite pas de réglage manuel des hyperparamètres.

Abstract

This study proposes voltage-dependent-synaptic plasticity (VDSP), a novel brain-inspired unsupervised local learning rule for the online implementation of Hebb's plasticity mechanism on neuromorphic hardware. The proposed VDSP learning rule updates the synaptic conductance on the spike of the postsynaptic neuron only, which reduces by a factor of two the number of updates with respect to standard spike timing dependent plasticity (STDP). This update is dependent on the membrane potential of the presynaptic neuron, which is readily available as part of neuron implementation and hence does not require additional memory for storage. Moreover, the update is also regularized on synaptic weight and prevents explosion or vanishing of weights on repeated stimulation. Rigorous mathematical analysis is performed to draw an equivalence between VDSP and STDP. To validate the system-level performance of VDSP, we train a single-layer spiking neural network (SNN) for the recognition of handwritten digits. We report 85.01 ± 0.76% (Mean ± SD) accuracy for a network of 100 output neurons on the MNIST dataset. The performance improves when scaling the network size (89.93 ± 0.41% for 400 output neurons, 90.56 ± 0.27 for 500 neurons), which validates the applicability of the proposed learning rule for spatial pattern recognition tasks. Future work will consider more complicated tasks. Interestingly, the learning rule better adapts than STDP to the frequency of input signal and does not require hand-tuning of hyperparameters.

3.2 Introduction

The amount of data generated in our modern society is growing dramatically, and Artificial Intelligence (AI) appears as a highly effective option to process this information. However, AI still faces the major challenge of data labeling: machine learning algorithms associated with supervised learning can bring AI at human-level performance, but they require costly manual labeling of the datasets. A highly desirable alternative would be to deploy unsupervised learning strategies that do not require data pre-processing. Neuromorphic engineering and computing, which aims to replicate bio-realistic circuits and algorithms through a spike-based representation of data, relies heavily on such unsupervised learning strategies. Spike timing dependent plasticity (STDP) is a popular unsupervised learning rule used in this context, where the relative time difference between the pre-and post-synaptic neuron spikes defines synaptic plasticity [81, 195, 196]. STDP is a spiking version of the traditional Hebbian learning concept [34, 197, 36], where a synaptic connection is modified depending only on the local activity correlations between its presynaptic and postsynaptic neurons.

In addition to its intrinsic unsupervised characteristic, STDP is also very attractive due to the locality of its synaptic learning. Such a feature could dramatically reduce hardware constraints of SNN by avoiding complex data exchange at the network level. However, STDP retains a major challenge: it requires precise spike times/traces to be stored in memory and fetched at every update to the processor. In most implementations [79, 198], decaying spike traces are used to compute synaptic weight update, adding extra state variables to store and update. In digital neuromorphic systems [199, 200, 201, 202], implementing STDP comes with an added cost of memory requirement for storing spike times/traces for every neuron and energy expenditure for fetching these variables during weight update. For analog hardware implementation [203, 204, 205, 206], circuit area and power are spent in storing spike traces on capacitors, thus raising design challenges. In-memory computing approaches have been strongly considered for STDP implementation to mitigate memory bandwidth requirements. The utilization of non-volatile memory-based synapses, or memristors, has been primarily considered [207, 208, 209, 182]. The seminal idea is to convert the time distance between pre- post-signals into a voltage applied across a single resistive memory element. The key advantage is to compute the STDP function directly on the memory device and to store the resulting synaptic weight permanently. This approach limits data movement and ensures the compactness of the hardware design (single memristor crosspoints may feature footprints below 100 nm). Further similar hardware propositions for STDP implementation have been discussed in the literature [210, 211]. Nevertheless, in all these approaches, time-to-voltage conversion requires a complex pulse shape (pulse duration should be in the order of STDP window and pulse amplitude should reflect the shape of STDP function), thus requiring complex circuit overhead and limiting the energy benefit of low power memory devices.

Moreover, STDP has the constraint of a fixed time window. As STDP is a function of the spike time difference between a post and a presynaptic neurons, the time window is the region in which the spike time difference must fall to update the weight significantly. The region of the time windows must be optimized to the temporal dynamics of spike-based signals to achieve good performances with STDP. This latter point raises additional issues at both the computational level (i.e., how to choose the appropriate STDP time window) and hardware level (i.e., how to design circuits with this level of flexibility). In other words, the challenge for deploying unsupervised strategies in neuromorphic SNN is two-sided: the concept of STDP needs to be further developed to allow for robust learning performances, and hardware implementations opportunities need to be considered in the meantime to ensure large scale neuromorphic system development.

In this work, we propose Voltage-Dependent Synaptic Plasticity (VDSP), an alternative approach to STDP that addresses these two limitations of STDP: VDSP does not require a fixed scale of spike time difference to update the weights significantly and can be easily integrated on in-memory computing hardware by preserving local computing. Our approach uses the membrane potential of a pre-synaptic neuron instead of its spike timing to evaluate pre/post neurons correlation. For a Leaky Integrate-and-Fire (LIF) neuron [48], membrane potential exhibits exponential decay and captures essential information about the neuron's spike time; intuitively, a high membrane potential could be associated with a neuron that is about to fire while low membrane potential reflects a neuron that has recently fired. A post-synaptic neuron spike event is used to trigger the weight update based on the state of the pre-synaptic neuron. The rule leads to a biologically coherent temporal difference. We validate the applicability of this unsupervised learning mechanism to solve a classic computer vision problem. We tested a network of spiking neurons connected by such synapses to perform recognition of handwritten digits and report similar performance to other single-layer networks trained in unsupervised fashion with the STDP learning rule. Remarkably, we show that the learning rule is resilient to the temporal dynamics of the input signal and eliminates the need to tune the hyperparameters for input signals of different frequency range. This approach could be implemented in neuromorphic hardware with little logic overhead, memory requirement and enable larger networks to be deployed in constrained hardware implementations.

Past studies have investigated the role of membrane potential in the plasticity of the mammalian cortex [212]. The in-vivo voltage dependence of synaptic plasticity has been demonstrated in [213]. In [214], bidirectional connectivity formulation in the cortex has been demonstrated as a resultant of voltage-dependent Hebbian-like plasticity. In [215], a voltage-based Hebbian learning rule was used to program memristive synapses in a recurrent bidirectional network. A presynaptic spike led to a weight update dependent on the membrane potential of postsynaptic neurons. The membrane potential was compared with a threshold voltage. If the membrane potential exceeded this threshold, long-term potentiation (LTP) was applied by applying a fixed voltage pulse on the memristor, while, for low membrane potential, long-term depression (LTD) took place. However, in their case, the weight update is independent of the magnitude of the membrane potential, and hence the effect of precise spike time difference cannot be captured. Lastly, these past studies have never reported handwritten digit recognition and benchmark against STDP counterparts.

In the following sections, we first describe the spiking neuron model and investigate the relation between spike time and neuron membrane potential. Second, we describe the proposed plasticity algorithm, its rationale, and its governing equations. Third, the handwritten digit recognition task is described with SNN topology, neuron parameters and learning procedure. In the results section, we report the network's performance for handwritten digit recognition. Next, we demonstrate the frequency normalization capabilities of VDSP as opposed to STDP by trying widely different firing frequencies for the input neurons in the handwritten digit recognition task without adapting the parameters. Finally, the hyperparameter tuning and scalability of the network are discussed.

3.3 Materials and methods

3.3.1 Neuron modeling

LIF neurons [48] are simplified version of biological neurons, hence easy to simulate in an SNN simulator. This neuron model was used for the pre-synaptic neuron layers. The governing equation is

$$\tau_m \frac{dv}{dt} = -v + I + b \tag{3.1}$$

where τ_m is the membrane leak time constant, v is the membrane potential, which leaks to resting potential (v_{rest}), I is the injected current, and b is a bias. Whenever the membrane potential exceeds a threshold potential (v_{th}), the neuron emits a spike. Then, it becomes insensitive to any input for the refractory period (t_{ref}) and the neuron potential is reset to voltage (v_{reset}).

An adaptation mechanism is added to the post neurons to prevent instability due to excessive firing. In the resulting adaptive leaky integrate-and-fire (ALIF) neuron, a second state variable is added. This state variable n is increased by inc_n whenever a spike occurs, and the value of n is subtracted from the input current. This causes the neuron to reduce its firing rate over time when submitted to strong input currents [216]. The state variable n decays by τ_n :

$$\tau_n \frac{dn}{dt} = -n \tag{3.2}$$

3.3.2 Relation between spike time and membrane potential

Hebbian-based STDP can be defined as the relation between $\Delta w \in \mathbb{R}$, the change in the conductance of a weight, and $\Delta t = t_{\text{post}} - t_{\text{pre}}$, the time interval between a presynaptic spike at time t_{pre} and a postsynaptic spike at time t_{post} with $\Delta t, t_{\text{pre}}, t_{\text{post}} \in \mathbb{R}^+$. This relation can be modeled as

$$\Delta w \propto \begin{cases} \exp\left(\frac{-\Delta t}{\tau^+_{\text{STDP}}}\right), & t_{\text{pre}} < t_{\text{post}} \\ -\exp\left(\frac{\Delta t}{\tau^-_{\text{STDP}}}\right), & \text{otherwise.} \end{cases} \tag{3.3}$$

with τ_{STDP} being the time constants for potentiation (+) and depression (−). This model is commonly computed during both the pre and postsynaptic neuron spikes, e.g., with the two traces model [79]. For VDSP, we seek to compute a similar Δw, but as a function of only $V(t_{\text{post}})$, the membrane potential of a presynaptic neuron at the time of a postsynaptic spike.

Fortunately, when the presynaptic LIF neuron is only fed by a constant positive current $I \in \mathbb{R}^+$, the spiking dynamics can be predicted. Solving the presynaptic LIF neuron's differential equation for the membrane potential with no bias (Equation 3.1 with $b = 0$) during subthreshold behavior yields

$$v(t) = I + c \cdot \exp\left(\frac{-t}{\tau_m}\right), \tag{3.4}$$

where c is the integration constant. Solving Equation 3.4 for t_{pre} and t_{post} allows us to define a new relation for $t_{post} - t_{pre}$:

$$t_{\text{post}} - t_{\text{pre}} = \tau_m \ln\left(\frac{v(t_{\text{pre}}) - I}{v(t_{\text{post}}) - I}\right). \tag{3.5}$$

with $v(t_{\text{pre}})$ and $v(t_{\text{post}})$ equal to the membrane potential of the presynaptic neuron at the moment of a presynaptic spike and postsynaptic spike, respectively. Assuming I is sufficient to make the presynaptic neuron spike in a finite amount of time, i.e., $I > v_{\text{th}}$, then $v(t_{\text{pre}} - \epsilon) = v_{\text{th}}$ and $v(t_{\text{pre}} + \epsilon) = v_{\text{reset}}$, with ϵ representing an infinitesimal number. Conceptually, $v(t_{\text{pre}} - \epsilon)$ represents a spike that is about to happen and $v(t_{\text{pre}} + \epsilon)$ a spike that has happened in the recent past, when there is no refractory period ($t_{\text{ref}} = 0$). Assuming $\epsilon \to 0$, we obtain:

$$\Delta t = \tau_m \ln \left(\frac{v_{\text{th}} - I}{v(t_{\text{post}}) - I} \right) \tag{3.6}$$

if the presynaptic neuron is about to spike or

$$\Delta t = \tau_m \ln \left(\frac{v_{\text{reset}} - I}{v(t_{\text{post}}) - I} \right) \tag{3.7}$$

if the presynaptic neuron recently spiked. To select between one of these values, we must obtain the smallest Δt, as to form a pair of t_{pre} and t_{post} that are closest in time. These two equations can be combined into:

$$|\Delta t| = \tau_m \cdot \min \left\{ \left| \ln \left(\frac{v_{\text{th}} - I}{v(t_{\text{post}}) - I} \right) \right|, \left| \ln \left(\frac{v_{\text{reset}} - I}{v(t_{\text{post}}) - I} \right) \right| \right\} \tag{3.8}$$

By using Δt as a function of $v(t_{\text{post}})$ from Equation 3.8, with $v_{\text{th}} = 1$, $v_{\text{reset}} = -1$ and knowing $v_{\text{reset}} \leq v(t_{\text{post}}) < v_{\text{th}}$, then equation 1 can be rearranged to:

$$\Delta w \propto \begin{cases} \left(\frac{v(t_{\text{post}}) - I}{-1 - I} \right)^{\frac{\tau_m}{\tau^{+}_{\text{STDP}}}}, & I - \sqrt{I^2 - 1} > v(t_{\text{post}}) \\ -\left(\frac{1 - I}{v(t_{\text{post}}) - I} \right)^{\frac{\tau_m}{\tau^{-}_{\text{STDP}}}}, & \text{otherwise.} \end{cases} \tag{3.9}$$

This final result proves that, when the presynaptic neuron is driven by constant current, Hebbian STDP can be precisely modeled using only $v(t_{\text{post}})$, the membrane potential of a presynaptic neuron at the time of a postsynaptic spike. Note that such generalization cannot be done in the case of Poisson-like input signals. Figure 3.1A,B demonstrate experimentally the relation between the membrane potential and $|\Delta t|$ from Equation 3.8. The condition $I - \sqrt{I^2 - 1} > v(t_{\text{post}})$ can be inferred from Equation 3.8, to select the minimal parameter, since

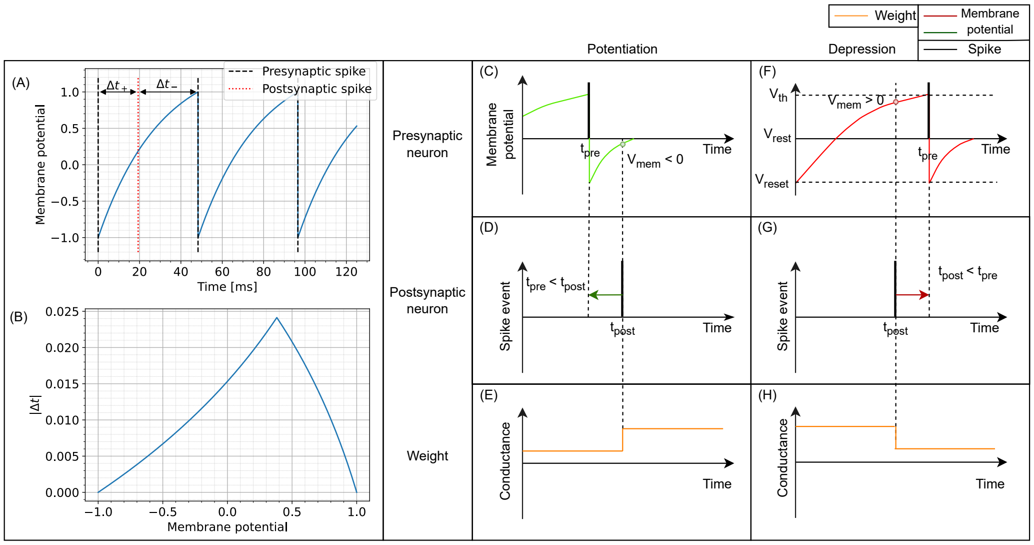


Figure 3.1 Schematic representation of the VDSP learning rule implemented between a pre- and postsynaptic spiking neuron. In (A), the membrane potential of a LIF neuron is shown evolving through time when fed with a constant current. In (B), the absolute time difference between the post and presynaptic spikes is computed analytically as a function of the membrane potential from (A). It is trivial, once the spike time difference is computed, to determine the STDP window as a function of membrane potential. (C,F) Show the spiking event of the presynaptic neuron (vertical black line) along with its membrane potential (colored curve). (D,G) Show the spike event of the postsynaptic neuron. The weight update (E,H) happens whenever the post-synaptic neuron fires. The update is dependent on the membrane potential of pre-synaptic neuron. If the pre-synaptic neuron fired in the recent past ($t_{pre} < t_{post}$), the membrane potential of the presynaptic neuron is lesser than zero, and we observe potentiation of synaptic weight (C–E). Whereas if the pre-synaptic neuron is about to fire ($t_{post} < t_{pre}$), the membrane potential of the pre-synaptic neuron is greater than zero and we observe depression of synaptic weight (F–H).

$$\min\{a, b\} = \begin{cases} a, & \text{if } a \leq b \\ b, & \text{otherwise.} \end{cases} \tag{3.10}$$

Moreover, as Equation 3.8 shows, the neuron parameters, namely the membrane reset and threshold potentials, are implicitly used to calculate the potentiation and depression windows. For example, the condition $I - \sqrt{(I^2 - 1)} > v(t_{\text{post}})$ of Equation 3.9 can be simplified to $v(t_{\text{post}}) < 0$ if $v_{\text{reset}} = \frac{I}{v_{\text{th}} - I}$ instead of -1. Both v_{th} and v_{reset} can be modified to tune the balance between potentiation and depression. Supplementary Figure 3.4 highlights the empirical effect of changing the value of v_{th} and v_{reset} on the $\Delta w = \text{VDSP}(\Delta t)$ window between two neurons with a fixed initial weight $w = 0.5$.

3.3.3 Proposed plasticity algorithm

The proposed implementation of synaptic plasticity depends on the postsynaptic neuron spike time and the presynaptic neuron's membrane potential. This version of Hebbian plasticity in which the weight is updated on either postsynaptic or presynaptic spikes is also known as single spike synaptic plasticity [207]. In real world applications, the presynaptic input current I is often not known and not constant, which would be mandatory for reproducing STDP perfectly as demonstrated in Equation 3.9. The less information is known about the input current, the more our plasticity rule converge into a probabilistic model. A low membrane potential suggests that the presynaptic neuron has fired recently, leading to synaptic potentiation (Figure 3.1C–E). A high presynaptic membrane potential suggests that the pre-synaptic neuron might fire shortly in the future and leads to depression (Figure 3.1F–H). A different resting state potential and reset potential is essential to discriminate inactive neurons and neurons that spiked recently.

Hebbian plasticity mechanisms can be grouped into additive or multiplicative types. In the additive versions of plasticity, the magnitude of weight update is independent of the current weight, but weight clipping must be implemented to restrict the values of weight between bounds [81]. Although the weight is not present in weight change computation equation directly, the present weight must be fetched for applying clipping. In neurophysiology experiments [217], it is also demonstrated that the weight update depends on the current synaptic weight in addition to the temporal correlation of spikes and is responsible for stable learning. The weight dependence is often referred to as multiplicative Hebbian learning as opposed to its additive counterpart and leads to stable learning and log-normal distribution of firing rates which are coherent with biological system recording [218].

VDSP relies on the multiplicative plasticity rule that considers the present weight value for computing the weight update magnitude. During potentiation, the weight update is proportional to $(W_{max} - W)$, and during the depression phase, the weight update magnitude is proportional to W, where W is the current weight, and W_{max} is the maximum weight. Multiplicative weight dependence is a crucial feature of VDSP, and no hardbound is needed as typically used with additive plasticity rules. A detailed discussion is presented in the discussion section and Figure 3.2.

The functional dependence of weight update on the membrane potential of the presynaptic neuron and the current synaptic weight is presented in Figure 3.2A,B. The weight or synaptic conductance varies between zero and one. The weight update is modeled as:

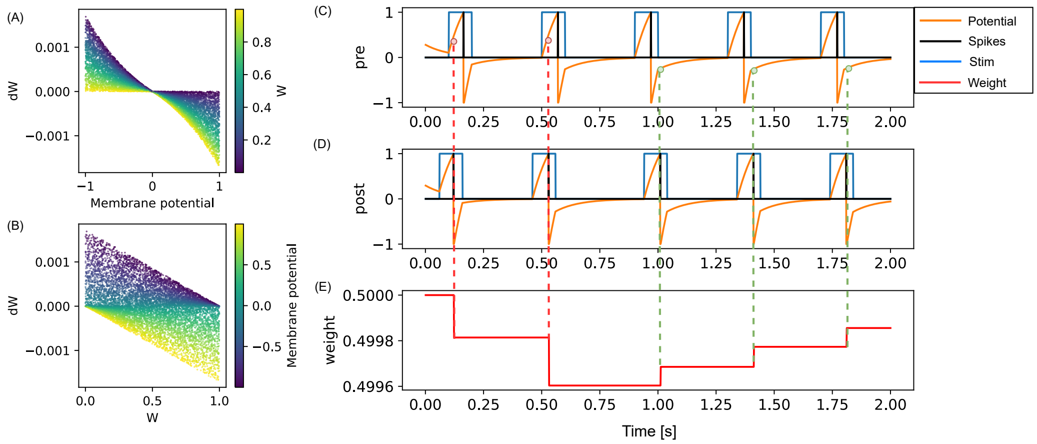


Figure 3.2 (A) The weight update (dW) is plotted as a function of the membrane potential of pre-synaptic neuron, with the color code representing the initial weight. (B) The dW is linearly dependent on (1-W) for potentiation and on (W) for depression. The learning rate is set to 0.001 in both (A,B). (C–E) A pair of pre-synaptic neuron and post-synaptic neuron is simulated along with their synaptic weight evolution. The weight update occurs at every post-synaptic neuron spike event and is negative if the pre-synaptic neuron membrane potential is greater than zero (shown in red dotted lines). The weight update is positive (green dotted lines) if the pre-synaptic neuron voltage is lesser than zero.

where dW is the change in weight, V_{pre} is the membrane potential of the presynaptic neuron, t_{post} is the time of postsynaptic neuron spike event, W is the current weight of the synapse, W_{max} is the maximum weight and is set to one, t is the current time, and lr is the learning rate.

To illustrate the weight update in the SNN simulator, a pair of neurons (Figure 3.2C,D) were connected through a synapse (Figure 3.2E) implementing the VDSP learning rule. The presynaptic and postsynaptic neurons were forced to spike at specific times. To potentiation and depression for $t_{post} > t_{pre}$ and $t_{post} < t_{pre}$ are shown with green and red dotted lines, respectively.

3.3.4 MNIST classification network

To benchmark the learning efficiency of the proposed learning rule for pattern recognition, we perform recognition of handwritten digits. One advantage of this task is that the weights of the trained networks can be interpreted to evaluate the network's learning. We use the modified national institute of standards and technology database (MNIST) dataset [219] for training and evaluation, which is composed of 70,000 (60,000 for training and 10,000 for evaluation) 28×28 grayscale images. The SNNs were simulated using the *Nengo* python simulation tool [220], which provide numerical solutions to the differential equations of both LIF and ALIF neurons. The timestep for simulation was set to 5 ms, which is equal to the chosen refractory period for the neurons.

The input layer is composed of 784 (28×28) LIF neurons (Figure 3.3). The pixel intensity is encoded with frequency coding, where the spiking frequency of the neuron is proportional to the pixel value. It is essential, when using VDSP, to use different v_{rest} and v_{reset} values to discriminate inactive neurons and neurons that spiked recently (Figure 3.2C,D). In our work, v_{rest} is set to zero volt, and v_{reset} is set to -1 V.

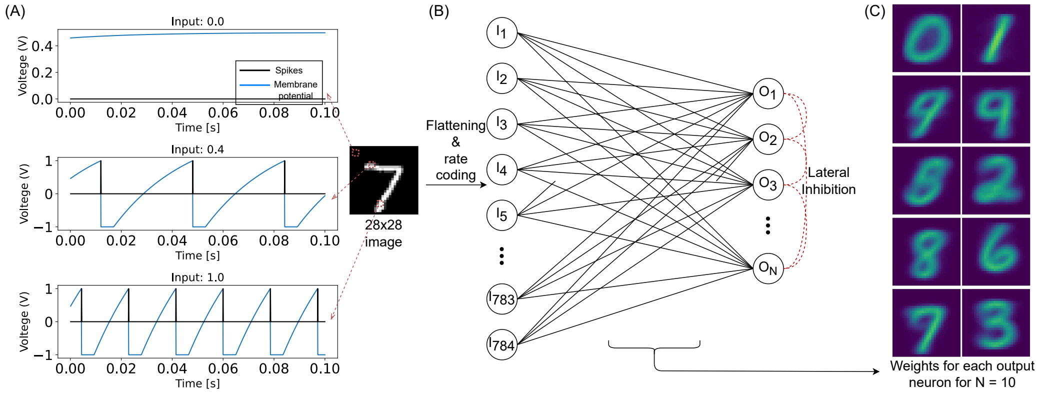


Figure 3.3 Representation of the SNN implementation used in this study to benchmark the VDSP learning rule with the MNIST classification task. (A) The response of the LIF neuron used in this study is plotted for input current of magnitude 0 (black pixel), 0.4 (gray pixel), and 1 (white pixel) for a duration of 100 ms. In (B), 28 × 28 grayscale image is rate encoded with the help of 784 input LIF neurons. Each sample is presented for 350 ms. The input neurons are fully connected to the ALIF output neurons connected in Winner Takes All (WTA) topology for lateral inhibition. (C) The weight matrix for each of the 10 output neurons.

Property	Input layer	Output layer
Refractory period	5 ms	5 ms
Leak time constant	30 ms	30 ms
Reset voltage	-1 V	0 V
Rest voltage	0 V	0 V
Threshold	1 V	1 V
Bias	0.5	0
Adaptation increment	-	0.01
Adaptation leak time constant	-	1 s
WTA time constant	-	10 ms

Table 3.1 In order to reproduce the results of this study, the same can be used in conjunction with proposed equations of the VDSP rule with a learning rate equal to 5×10^{-2}.

The output layer is modeled as ALIF neurons connected in a Winner Takes All (WTA) topology: on any output neuron spike occurrence, the membrane potential of all other neurons is clamped to zero for 10 ms. All the input neurons are connected to all the output neurons through synapses implementing the VDSP learning rule. The initial weights of these synapses were initialized randomly, with a uniform distribution between the minimum (0) and maximum (1) weight values. Each image from the MNIST database was presented for 350 ms with no wait time between images. The neuron parameters of input and output neurons used in this study are summarized in Table 1.

Once trained, the weights were fixed, and the network was presented again with the samples from the training set, and all the output neurons were assigned a class based on activity during the presentation of digits of a different class. The 10,000 images from the test set of the MNIST database were presented to the trained network for testing the network. Based on the class of neuron with the highest number of spikes during sample presentation time, the predicted class was assigned. The accuracy was computed by comparing it with the true class. For larger networks, the cumulative spikes of all the neurons for a particular class were compared to evaluate the network's decision. The above could be easily realized in hardware with simple connections to the output layer neurons. More sophisticated machine learning classifiers like Support Vector Machines (SVMs) or another layer of spiking neurons can also be employed for readout to improve performance [221].

3.4 Results and discussion

On training a network composed of 10 output neurons for a single epoch, with 60,000 training images of the MNIST database, we observe distinct receptive fields for all the ten digits (Figure 3.3C). Note that the true labels are not used in the training procedure with the VDSP learning rule, and hence the learning is unsupervised. We report classification accuracy of $61.4 \pm 0.78\%$ (Mean ± S.D.) based on results obtained from five different initial conditions.

3.4.1 Presynaptic firing frequency dependence of VDSP

As stated previously, the VDSP rule does not use the presynaptic input current to compute Δw. Therefore, as the presynaptic input current changes, e.g., in between the samples of the MNIST dataset, the change in weight conductance, Δw, is affected. Figure 3.4A presents the relation between the presynaptic firing frequency when the input current is changed and the $\Delta w =$ VDSP (Δt) window between two neurons with a fixed initial weight w=0.5. As the current gets larger, the presynaptic firing frequency is increased, and the window shortens. This has a normalizing effect on the learning mechanism of VDSP when subjected to different spiking frequency regimes.

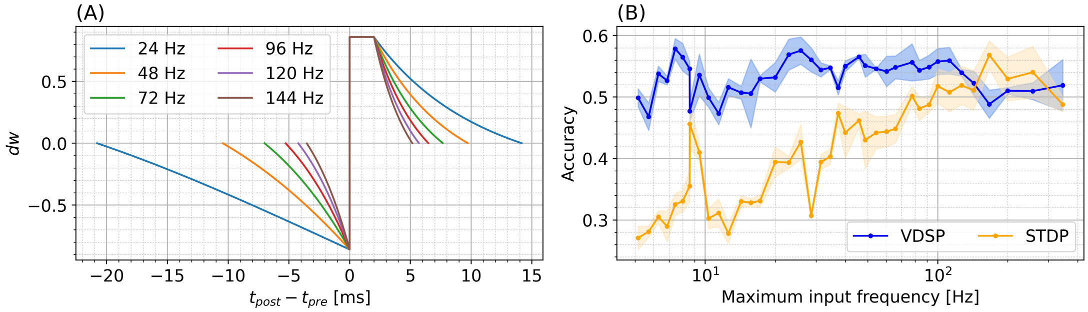


Figure 3.4 Presynaptic firing frequency dependence of VDSP and STDP. Subfigure (A) shows the effect of scaling the presynaptic neuron input current on the VDSP update window for fixed weight w = 0.5 in a two neurons configuration. As the input current changes, the presynaptic neuron fires at various frequencies indicated by the line color. Higher presynaptic spiking frequencies result in smaller time windows. The plateau between Δt $\in$ [0,2] ms is an artifact of the refractory period of 2 ms, where the membrane potential is kept at a reset value throughout. In (B), similar scaling is applied to the values of the pixels being fed to the presynaptic neurons during the MNIST classification task using the WTA architecture. Each point in (B) results from running the task 5 times with different random seeds using 10 output neurons, with standard deviation shown with the light-colored area under the curve. No adaptation mechanism was used for (B) to provide an unbiased comparison between classical STDP and VDSP in different spiking frequency regimes. No frequency-specific optimization was done during these experiments.

In Figure 3.4B, we recreated a simplified version of the MNIST classification task using the WTA presented in the previous sections. Notably, there is no adaptation mechanism in the output layer, and the duration of the images is dynamically computed to have a maximum of ten spikes per pixel per image. These changes were made to specifically show the dependence of the input frequency on the accuracy, but they also affect the maximum reached accuracy in the case of VDSP. We ran the network with ten output neurons for one epoch with both VDSP and STDP with constant parameters. As expected, VDSP is much more resilient to the change in spiking input frequency. This effect is beneficial since the same learning rule can be used in hardware, and the learning can be accelerated by simply scaling the input currents. We note that neither the VDSP nor the STDP's parameters are maximized for absolute performance in this experiment, and we used the same weight normalizing function as [87] for STDP.

3.4.2 Impact of network size and training time on VDSP

To investigate the impact of the number of output neurons and epochs on classification accuracy, the two-layer network for MNIST classification is trained for up to five epochs and five hundred output neurons. The resulting accuracy for the different number of epochs and number of output neurons is shown in Figure 3.5. Note that network hyperparameters were not re-optimized for these experiments (i.e., hyperparameters were optimized for a 50 output neuron topology only). Key performance numbers are tabulated in Table 2 and compared to the state-of-the-art accuracy reported in the literature. We observe equivalent or higher performance than the networks trained with the pair-based STDP in software simulations [87] and hardware-aware simulations [208, 210, 211] for most network sizes. This result validates the efficiency of the VDSP learning rule for solving computer vision pattern recognition tasks.

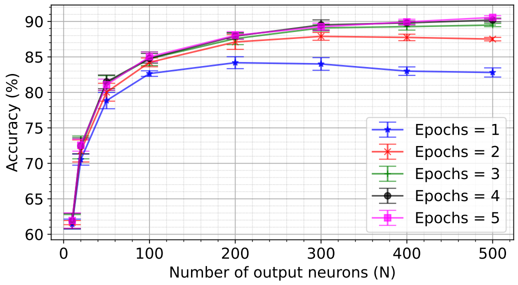


Figure 3.5 A spiking neural network with 784 input neurons and N output neurons was trained on the training set (60,000 images) of the MNIST dataset for different numbers of epochs. The accuracy was computed on the test set (10,000) unseen images of the MNIST dataset. Networks with the number of output neurons ranging from 10 to 500 were trained for the number of epochs ranging from 1 to 5. Each experiment was conducted for five different initial conditions. The mean accuracy for five trials is plotted in the figure, with the error bar indicating the standard deviation.

The performances of the network trained with VDSP are well aligned with hardware aware software simulations (Table 3.2) for simplified STDP and memristor simulation [208], resistive memory-based synapse simulation [211], PCM based synapse simulation [210]. VDSP has lower accuracies with respect to [222] in their 50 and 200 neuron simulations, which can be explained by the different number of learning epoch and encoding strategy of the MNIST digits.

The comparable performance of VDSP with standard STDP can be attributed to the fact that the membrane potential is a good indicator of the history of input received by

This work			Past studies			
Neurons	**Epochs**	**Accuracy (%)** ($\mu \pm \sigma$)	**Neurons**	**Epochs**	**Accuracy (%)**	**Ref.**
10	1	61.4 ± 0.78	10	1	60	[208]
50	1	78.84 ± 1.28	50	1	76.8	[211]
50	3	81.3 ± 1.76	50	3	77.2	[210]
			50	1	78.55	[181]
			50	3	81	[208]
			50	-	83.03	[222]
100	3	84.74 ± 1.08	100	3	82.9	[87]
			100	1	89.15	[181]
			200	17	91.63	[222]
300	3	89.08 ± 0.49	300	3	93.5	[208]
400	3	89.26 ± 0.54	400	3	87	[87]
500	5	90.56 ± 0.27				

Table 3.2 The performance achieved by training SNN with the VDSP rule is tabulated for various network sizes (number of output neurons) and epochs. Each experiment was repeated with five different initial conditions, and the accuracies are reported as (Mean ± S.D.). Compared with the hardware-independent approach of pair based STDP, we achieved 84.74 ± 1.08% for a network of 100 output neurons trained over three epochs. For a network of 400 output neurons trained over three epochs, we achieved 89.26 ± 0.54%.

neurons and not just the last spike. In addition, the weight update in VDSP depends on the current weight, which regularizes the weight update and prevents the explosion or dying of weights. As in Supplementary Figure 3.1, we observe a bimodal distribution of weights and clear receptive fields for a network of 50 output neurons. When this weight dependence is removed and clipping of weights between 0 and 1 is used, most weights become either zero or one, and receptive fields are not clear with current parameters (Supplementary Figure 3.2).

3.4.3 VDSP parameters optimization

Convergence of the VDSP learning was possible with additional parameters optimization. Firstly, clear receptive fields require to decrease the weight of inactive pixels corresponding to the background. To penalize these background pixels, which do not contribute to the firing of the output neuron, we introduce a positive bias voltage in the input neurons of the MNIST classification SNN. This bias leads to a positive membrane potential of background neurons but does not induce firing. Consequently, the weight values are depressed according to the VDSP plasticity rule. Depressing the background neuron weight also balances the potentiation of foreground pixels and keeps in check the total weights contribution of an output neuron, thus preventing single neurons from always "winning" the competition. To validate the above hypothesis, we experimented training with zero bias voltage Supplementary Figure 3.3 and observed poor receptive fields.

The learning rate is a crucial parameter for regulating the granularity of weight updates. To study the impact of learning rate and the number of epochs on the performance, we train networks with learning rates ranging from 10–5 to 1 for up to five epochs. The resulting performance for five different runs is plotted for ten output neurons and 50 output neurons in Figure 6. For a single epoch, we observe the optimal performance for ten output neurons at a learning rate of 5×10–3. For 50 output neurons and a single epoch, the optimal learning rate was 1×10–2. This result is indicative of the fact that the optimal learning rate increases for a greater number of neurons. Conventional STDP, on the other hand, has a minimum of two configurable parameters: learning rate and temporal sensitivity window for potentiation and depression. These are to be optimized to the dynamics of the input signal. VDSP has just one parameter and can be optimized based on the number of output neurons and training data size or the number of epochs, as discussed. There are many additional hyperparameters in a spiking neural network (SNN), such as time constant, thresholds, bias, and gain of the neurons, which can affect network performances. The neuron and simulation parameters tabulated in Table 1 were optimized with grid search performed on a network comprising 50 output neurons trained over a single epoch.

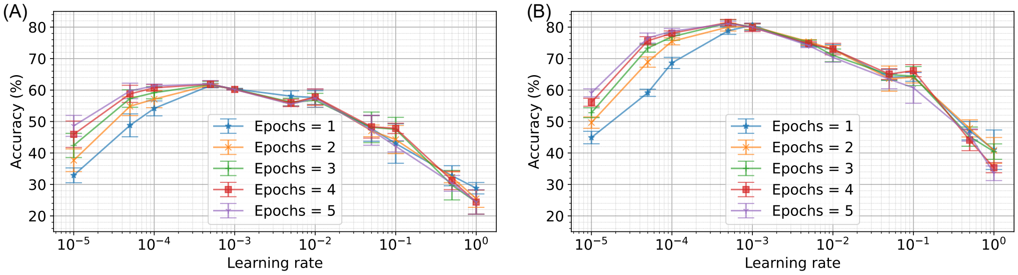


Figure 3.6 Dependence of the performance on learning rate and number of epochs for different network sizes. In (A), a network with 10 output neurons was trained on the MNIST dataset for different numbers of epochs and learning rates. Networks with learning rates ranging from 10^{-5} to 1 were trained for the number of epochs ranging from 1 to 5. Each experiment was conducted for five different initial conditions. The mean accuracy for five trials is plotted in the figure, with the error bar indicating the standard deviation. In (B), the experiments are repeated for 50 output neurons. As depicted, the optimum learning rate for a single epoch and 10 neurons is $5{\times}10^{-4}$. Whereas, for 50 output neurons, the optimum learning rate for a single epoch is 10^{-3}.

3.4.4 Hardware choices for VDSP

In the past, voltage dependent plasticity rules proposed triggering weight update on presynaptic neuron spike (Brader et al., 2007; Diederich et al., 2018). Updating on presynaptic neuron spike is also an intuitive choice considering the forward directional computation graph for SNN. However, in the specific case of the output layer of multi-layer feedforward networks with WTA-based lateral inhibition, at most, one output neuron spikes at a time, and the output spike frequency would be significantly lower than the input spike frequency, reducing the frequency of weight updates required. Moreover, in multi-layer feedforward networks, activity in layers close to the output layer corresponds to the recognition of higher-level features and is a more attractive choice to synchronize the weight update. In addition, in networks for classification tasks, a convergence of layer size occurs from a large number of input neurons (for achieving high spatial resolution in neuromorphic sensors like DVS cameras, for instance) to a few neurons in the output layer. In hardware, a lower weight update frequency would imply lesser power consumption required in learning and a reduction in the learning time, thus providing greater flexibility with bandwidth available for inference.

The locality of the learning rule could be dependent on the hardware architecture. In the specific case of in-memory computing based neuromorphic hardware implementations, the synapse is physically connected to both postsynaptic and presynaptic neurons. State variables like the membrane potential of these neighboring neurons are readily available to the connecting synapse. Moreover, for memristive synapses, the dependence of weight change on initial weight is an inherent property of device switching. The proposed learning rule is attractive for implementing local learning in such systems.

For lateral inhibition in the output layer, the membrane potential of all the other output neurons is clamped to zero for 10 ms upon firing of any output neurons. This choice is inspired by the similar approach employed in [208, 222, 181]. One alternative is using an equal number of inhibitory spiking neurons in the output layer [87]. However, using an equal number of inhibitory output neurons doubles the number of neurons, leading to the consumption of a significant silicon area when implemented on a neuromorphic chip. On the other hand, clamping the membrane potential does not require substantial circuit area and is a more viable option for hardware implementations.

We also evaluated the impact of injected Gaussian noise on neuron response for different input currents and noise distributions (Supplementary Figure 3.5). Gaussian noise centered around zero with different deviations was injected into the input neurons. While the membrane potential is substantially noisy in the case of mid-level noise injection, we do not observe a significant drop in performance. This feature makes VDSP an attractive choice of learning rule to be deployed on noisy analog circuits and nanodevices with high variability.

We also tested the applicability of the method for a network receiving random Poisson-sampled input spike patterns to drive the input layer. To elucidate this, a network of 10 output neurons was trained by feeding Poisson sampled spike trains to the input neuron with the frequency being proportional to the pixel value. The plots of membrane potential and neuron spike for different input values are presented in Supplementary Figure 3.6A–C. The network was trained for one epoch and recognition accuracy of 58% was obtained on the test set. The resulting weight plots are shown in Supplementary Figure 3.6D. Stable learning is observed and a small performance drop of 3% occurred as compared to constant input current.

3.5 Conclusion and future scope

In this work, we presented a novel learning rule for unsupervised learning in SNNs. VDSP is solving some of the limitations of STDP for future deployment of unsupervised learning in SNN. Firstly, as plasticity is derived from the membrane potential of the pre-synaptic neuron, VDSP on hardware would reduce memory requirement for storing spike traces for STDP based learning. Hence, larger and more complex networks can be deployed on neuromorphic hardware. Secondly, we observe that the temporal window adapts to the input spike frequencies. This property solves the complexity of STDP implementation, which requires STDP time window adjustment to the spiking frequency. This intrinsic time window adjustment of VDSP could be exploited to build hierarchical neural networks with adaptive temporal receptive fields [223, 224]. Thirdly, the frequency of weight update is significantly lower than the STDP, as we do not perform weight updates on both presynaptic and postsynaptic neuron spike events. This decrease in weight updates frequency by a factor of two is of direct interest for increasing the learning speed of SNN simulation and operation. Furthermore, this improvement is obtained without trading off classification performances on the MNIST dataset, thus validating the applicability of VDSP rule in pattern recognition. The impact of hyperparameters (learning rate, network size, and the number of epochs) is discussed in detail with the help of simulation results.

In the future, we will investigate the implementation of VDSP in neuromorphic hardware based on emerging memories. Also, future work should consider investigating the proposed learning rule for multi-layer feed-forward networks and advanced network topologies like Convolutional Neural Networks (CNNs) [225, 196] and Recurrent Neural Networks (RNNs)

[226]. Finally, using this unsupervised learning rule in conjunction with gradient-based supervised learning is an appealing aspect to be explored in future works.

Data availability statement

Publicly available datasets were analyzed in this study.

This data can be found at http://yann.lecun.com/exdb/mnist/.

Author contributions

JR, YB, FA, J-MP, and DD contributed to formulating the study. NG and IB designed and performed the experiments and derived the models. NG, IB, FA, and YB analyzed the data. TS contributed to realizing the plasticity rule in Nengo. All authors provided critical feedback and helped shape the research, analysis, and manuscript.

Funding

We acknowledged financial supports from the EU: ERC-2017-COG project IONOS (# GA 773228) and CHIST-ERA UNICO project. This work was also supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) (No. 559,730) and Fond de Recherche du Québec Nature et Technologies (FRQNT).

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

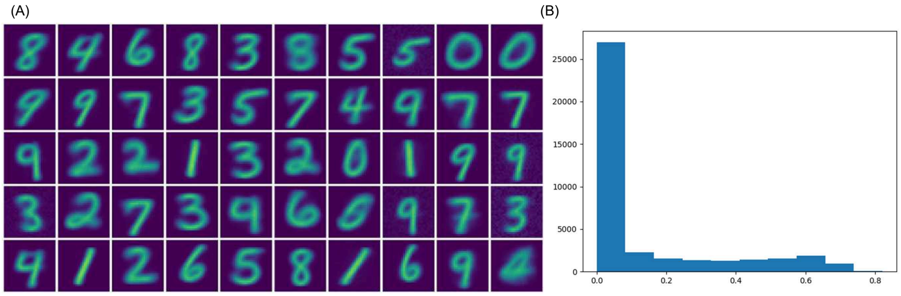
Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

3.6 Supplementary material

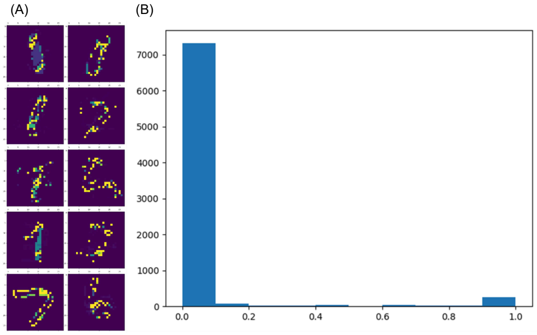
Receptive fields for 50 output neurons

Increasing the number of neurons in the output layer makes multiple neurons learn different representations of each class.



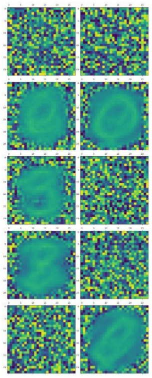
Supplementary Fig. 3.1 A network of 784 input neurons and 50 output neurons was trained with 60,000 images from the training subset of the MNIST dataset over three epochs. The weight map from each of the 50 neurons is shown in (A) where each image represents 784 synaptic weights for each output neuron. The histogram of the synaptic weights is plotted in (B). A bimodal distribution can be observed.

Importance of weight dependence of weight update function



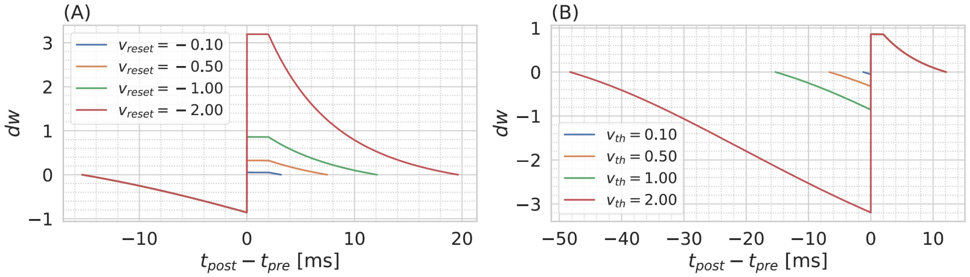
Supplementary Fig. 3.2 Weight plots and histogram for additive VDSP. The change in weight (dW) is independent of the current weight (W). After training a network of 10 output neurons with 60,000 training images, the obtained weights for each of the output neuron is plotted in (A). The histogram of all the network weights is plotted in (B). It can be observed that the weights are set to either zero or one as in the histogram.

Importance of penalization of background pixels with bias



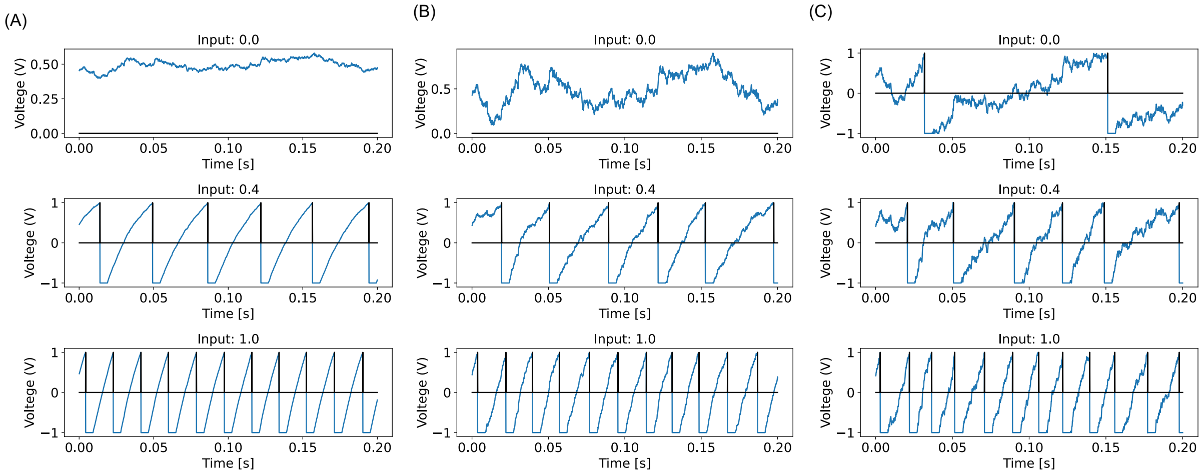
Supplementary Fig. 3.3 Weights of the network when the bias of input neuron was set to zero. A bias of zero leads to membrane potential of input neurons representing background pixel to remain at zero. Hence, the weights of input neurons that were inactive were not depotentiated. The neuron to fire first after presentation of one image has a higher probability of firing even for other digits as some pixels overlap.

Sensitivity of the temporal VDSP window on LIF neuron's parameters



Supplementary Fig. 3.4 Impact of the presynaptic LIF neuron's parameters on the shape of VDSP. In (A), the presynaptic neuron's reset potential is changed between -0.10 and -2, as indicated by the line colour with a fixed presynaptic neuron potential threshold of 1. This change impacts the potentiation part of the window ($tpost - tpre > 0$). In (B), the presynaptic neuron's potential threshold is changed between 0.10 and 2.0 as indicated by the line colour, with a fixed presynaptic neuron reset value of -1. This change impacts the depression part of the window ($tpost - tpre < 0$). Modifying these two values allows the tuning of the VDSP learning rule to a desired balance between potentiation and depression. I.e., for more potentiation, one should decrease the value of $vreset$ and for more depression, one should increase the value of vth.

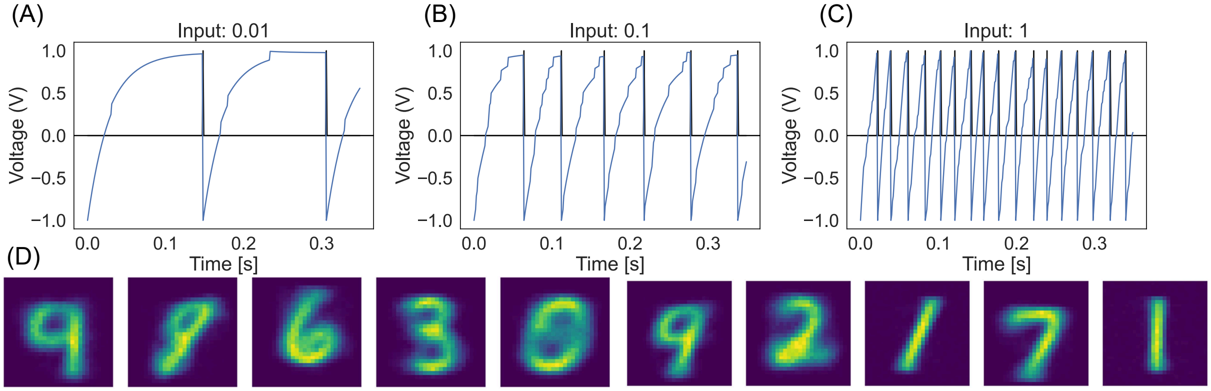
Impact of additive gaussian noise on network performance for 50 output neurons



	Low noise	Mid noise	High noise
Accuracy$(\mu \pm \sigma)$	81.25 ± 0.65	80.75 ± 0.86	65.32 ± 1.31

Supplementary Fig. 3.5 In (A), the neuron is excited by the constant input of magnitude 0, 0.4, and 1 to the input neuron of the MNIST classification network. Low magnitude noise of gaussian distribution centred around zero is injected to the input neuron in (A) for a network composed of 50 output neurons. In (B), the noise of mid-intensity is injected into the input neurons. Similarly, in (C), the noise of high intensity is injected into the input neurons. All accuracies are in format Mean ± S.D. resulting from five trials.

Impact of Poisson sampled input current on network performance for 10 output neurons



Supplementary Fig. 3.6 In (A-C), a LIF neuron was stimulated by Poisson-sampled spike trains of 0.01, 0.1, and 1 kHz, with constant weight of 0.1 and bias b. (D) Poisson spikes with a frequency proportional to the pixel intensity of MNIST images and bias to penalize background pixels were fed to input neurons of SNN. The network with 10 output neurons was trained with 60,000 training images from the MNIST database with a maximum input spike frequency of 1kHz corresponding to a white pixel (intensity of 256). In (D), the weight of the individual output neuron is plotted to visualize the receptive fields at the end of training with Poisson spikes.

CHAPTER 4

Learning with memristive synapses

"All models are wrong, but some are useful" – George Box

TABLE OF CONTENTS

4.1 Preface

Contribution to document

The objective of this chapter is to translate the plasticity rule outlined in chapter 3 into a practical synaptic device programming strategy. Memristive devices are particularly appealing as physical synaptic elements due to their non-volatile memory and scalability. Their scalability is further enhanced when a single device can store multiple resistance states or weights. Recent advances in memristive technologies, such as valence-change mechanisms (e.g., Ti and Hf oxide switching layers) and ferroelectric tunneling mechanisms, have enabled the demonstration of analog conductance programming in these devices, allowing for finer control of synaptic weights.

Different switching mechanisms lead to distinct electrical behaviors, such as threshold, asymmetry, non-linearities, and variability. In this chapter, we present a characterization and modeling technique to measure programming voltage and state-dependent switching in response to fixed-width (sub-microsecond) programming pulses. This technique provides a detailed understanding of how memristive states evolve under rapid electrical stimuli, which is critical for accurate simulation models. A phenomenological memristor model is proposed and fitted to enable system-level simulations incorporating the switching characteristics observed in electrical measurements. Using the model and system-level simulations that account for variability and parametric analysis, we assess key factors for learning, particularly the learning rate in SNNs trained with Hebbian learning. This learning process faces unique challenges due to the interaction of parameters such as spike timing, pulse amplitude, width, and the frequency of weight updates. To address these complexities, we propose a scaling factor-based mapping from simulation to hardware. Additionally, we examine the trade-off between gradual learning, achieved with a low scaling factor (resulting in smaller updates per sample) and learning with higher threshold mismatch, highlighting the balance between precision and robustness.

The findings of this chapter, particularly the role of the scaling factor and its impact on the effective learning rate of the network, form the basis for the circuit design of the VDSP amplifier presented in chapter 6. When taking into account device-to-device variations in memristive parameters, the ideal characteristics and programming conditions for synaptic devices can vary. In the following chapter, we aim to address the key question of how to fine-tune the parameters of computing and programming circuits based on the modeled characteristics of memristive devices, such as switching thresholds and asymmetry.

Title: Unsupervised local learning based on voltage-dependent synaptic plasticity for resistive and ferroelectric synapses.

Title in French: Apprentissage local non supervisé basé sur la plasticité synaptique dépendante de la tension pour les synapses résistives et ferroélectriques.

Date: October 2024 (Submission)

Status: Submitted for peer-review.

Journal: Nature Communication Materials

Authors: Nikhil Garg[1,2,3,*], Ismael Balafrej[4], Joao Henrique Quintino Palhares[1,2,5,6], Laura Bégon-Lours[7], Davide Florini[1,2], Donato Francesco Falcone[7], Tommaso Stecconi[7], Valeria Bragaglia[7], Bert Jan Offrein[7], Jean-Michel Portal[8], Damien Querlioz[9], Yann Beilliard[1,2], Dominique Drouin[1,2], Fabien Alibart[1,2,3,*]

Affiliations:

1. Institut Interdisciplinaire d'Innovation Technologique (3IT), Université de Sherbrooke, Sherbrooke, Québec, Canada
2. Laboratoire Nanotechnologies Nanosystèmes (LN2) – CNRS, Université de Sherbrooke, Québec, Canada
3. Institute of Electronics, Microelectronics and Nanotechnology (IEMN), Université de Lille, France
4. NECOTIS Research Lab, Electrical and Computer Engineering Dep., Université de Sherbrooke, Quebec, Canada
5. STMicroelectronics, Crolles, France
6. Univ. Grenoble Alpes, CEA, CNRS, Grenoble INP, SPINTEC, Grenoble, France
7. IBM Research GmbH - Zurich Research Laboratory, Ruschlikon, Switzerland
8. Aix Marseille Univ, Université de Toulon, CNRS, IM2NP, Marseille, France
9. Université Paris-Saclay, CNRS, Centre de Nanosciences et de Nanotechnologies, Palaiseau, France

***Corresponding Authors:**

- Fabien Alibart – Fabien.Alibart@Usherbrooke.ca
- Nikhil Garg – Nikhil.Garg@Usherbrooke.ca

Résumé

Dans cette étude, nous présentons la plasticité synaptique dépendante du voltage (VDSP) comme une approche efficace pour l'apprentissage non supervisé et local dans les synapses memristives basée sur les principes hebbiens. Cette méthode permet l'apprentissage en ligne sans nécessiter de circuits de mise en forme d'impulsions complexes généralement nécessaires pour la plasticité dépendante du timing des pics (STDP). Nous montrons comment la VDSP peut être avantageusement adaptée à trois types de dispositifs memristifs (synapses filamentaires à base d'oxyde métallique à base de TiO2, HfO2 et jonctions tunnel ferroélectriques (FTJ) à base de HfZrO4) avec des caractéristiques de commutation distinctives. Des simulations au niveau système de réseaux neuronaux à pics incorporant ces dispositifs ont été réalisées pour valider l'apprentissage non supervisé sur des tâches de reconnaissance de formes basées sur MNIST, obtenant des performances de pointe. Les résultats ont démontré une précision de plus de 83% sur tous les dispositifs utilisant 200 neurones. De plus, nous avons évalué l'impact de la variabilité des appareils, tels que les seuils de commutation et les niveaux HRS/LRS, et proposé des stratégies d'atténuation pour améliorer la robustesse.

Abstract

In this study, we introduce voltage-dependent synaptic plasticity (VDSP) as an efficient approach for unsupervised and local learning in memristive synapses based on Hebbian principles. This method enables online learning without requiring complex pulse-shaping circuits typically necessary for spike-timing-dependent plasticity (STDP). We show how VDSP can be advantageously adapted to three types of memristive devices (TiO_2,HfO2-based metal-oxide filamentary synapses, and $HfZrO_4$-based ferroelectric tunnel junctions (FTJ)) with disctinctive switching characteristics—. System-level simulations of spiking neural networks incorporating these devices were conducted to validate unsupervised learning on MNIST-based pattern recognition tasks, achieving state-of-the-art performance. The results demonstrated over 83% accuracy across all devices using 200 neurons. Additionally, we assessed the impact of device variability, such as switching thresholds and HRS/LRS levels, and proposed mitigation strategies to enhance robustness.

Keywords: Neuromorphic, Memristor, Learning, Unsupervised, In-memory computing, Pattern recognition.

4.2 Introduction

Deploying artificial intelligence (AI) applications on edge computing devices is raising the challenge of implementing intelligent algorithms with sever constraints on energy consumption, challenge that cannot be fulfilled by conventional technologies such as modern GPU. One strategy toward this goal is relying on the neuromorphic engineering and computing framework, which proposes the physical implementation of algorithms with specialized hardware designed to emulate the human brain's structure and function. Among the different propositions of neuromorphic devices and circuits, memristors [160, 161] —resistive devices with programmable conductance— have been considered as emerging memory devices used to create electronic synapses [227] in AI hardware systems. In neuromorphic architectures, memristors enable the realization of the Vector Matrix Multiplication function physically through Ohm's and Kirchoff's laws, which reduces significantly energy consumption and latency of the intensive VMM operation. In addition, memristor have generated a large interest to implement physically the different learning algorithms required during training of neuromorphic systems. Learning algorithms used for conventional artificial neural networks (ANNs) mostly rely on backpropagation and have been widely considered in the context of in-memory computing with memristor with severe constraints on memristors' accuracy (i.e. number of states available during programming), linearity (i.e. linear change of conductance on the entire resistive range) and variability [228, 229, 230, 170]. In neuromorphic approaches, and in particular in spiking neural networks (SNNs) [231], learning algorithms rely deeply on bio-inspiration and memristors could offer the additional advantage of local learning by implementing Hebbian principles. For instance, Spike timing-dependent plasticity (STDP) [36] have been found to be responsible for plasticity in biological synapses and worth adapting to hardware systems [101]. In such learning algorithms, change of states of the synaptic conductance depends only on the pre and post neuron activity and doesn't require the propagation of global error signals across the entire network, thus limiting data movement and the associated energy consumption. Various STDP implementations have considered memristor to implement online learning in SNN but (i) translation of pre- and post-neuron activity into actual signals promoting the change of resistance (i.e. learning) is impacting the overall network performances and (ii) the impact of memristive devices non-idealities is not straightforward to extract and still largely unexplored with respect to the large variety of memristive devices that could be considered based on physical mechanisms such as heating [232], oxidation [233, 234], phase change [235, 236], and ferroelectric domain [237] switching.

In this paper, we propose (i) to adapt a recent extension of STDP to memristive devices and to evaluate its performances on a relevant classification task. We consider for this study Voltage Dependent Synaptic Plasticity (VDSP), which uses spike timing and neuron membrane potential as a representation of pre and post neuron activities. Such local learning algorithm has been proposed recently as an interesting solution to solve the strong dependency of STDP to the range of frequency in SNNs and to ease the physical implementation by reducing the number of local parameters to be stored. (ii) We further analyse the impact of memristive devices properties on the performances and resilience of VDSP. We consider three distinctive technologies with different switching dynamics associated to different physical mechanisms originating the change of resistance. TiO_2 and CMO-HfO_2 are representative of valence change memory devices where oxygen vacancies are responsible for resistive switching and HZO represents the more recent class of ferroelectric tunnel junctions where ferroelectric domains are defining the resistive states. These choice of the device stack was motivated by their CMOS-compatible fabrication process [20, 238, 239] for back end of line (BEOL) integration. All three device can present analog change of resistance that could be advantageously used for online learning implementation, but with different relationship in between driving signals and devices' response (i.e. voltage driven resistive change in our case). We show in this paper how VDSP parameters can be adjusted to each technology by combining electrical characterization of the different technologies and electrical modeling. While individual technologies evaluation is often reported and can limit the generalization of the impact of device non-idealities on learning, we show here for three technologies how important switching characteristics such as variation in switching threshold and range of resistance changes can affect the performances of VDSP.

Several works have proposed implementation of STDP with memristive devices [240]. In these approaches, temporal correlation in between pre- and post- neuron events can be advantageously translated into voltage with pulse overlapping technique [207]. This technique enables the implementation of online learning in various memristive device technologies and has been proven efficient for unsupervised pattern learning and circumventing known issues of variability and stochasticity in memristive devices [208, 241]. Nevertheless, such pulse-overlapping method for temporal correlation detection faces challenges since the duration of the programming pulse is directly related to the time window for detecting the correlated spikes. When implemented into circuits, long pulses come with trade-offs such as reduced analog control, increased energy consumption for programming, and lower throughput. Additionally, if complex pulses (i.e. with exponential decaying function) can translate time distance in between the pre and post events into a large range of voltages, such pulses engineering is associated to complex voltage sources design that need to be adapted to each different learning algorithm and memristive technology, thus preventing a general circuit design approach.

From a system level perspective, conventional STDP requires to store locally at the pre and post neuron level the timing of the last emitted spike (note that pulse overlapping techniques are circumventing this issue by storing the pulse timing into the synaptic programming voltage). Strategies to reduce the local memory requirement have been proposed with neuron-state-dependent synaptic plasticity for CMOS synapses [115]. These learning rules, known as spike-driven synaptic plasticity (SDSP) [242], or single-spike STDP, use the membrane potential of the post-synaptic neuron and an extra calcium concentration variable associated with the rate of spike of the post-neuron to modulate the synaptic weight during a pre-synaptic spike, thus reproducing Hebbian learning concept. VDSP

[194, 243] employ a similar strategy by extracting the probability of pre-neuron firing from the neuron membrane potential. This approach further reduces the memory requirement by removing the calcium concentration memory block, while still maintaining a high-performing network, as demonstrated through recognition rates of handwritten digits obtained through unsupervised learning.

In the following, we first introduce the theoretical framework for implementing VDSP in analog memristive synapses. We propose a device characterization approach that models the programming behavior as a function of the applied voltage and the current device state. This behavior is used to fit a standard memristor model, allowing us to quantify key switching properties such as the switching threshold and nonlinear characteristics. Next, we discuss the mapping from device characteristics to simulation and outline the SNN simulation setup. Performance is incrementally evaluated based on key factors, including the number of output neurons, training samples, and learning epochs. Finally, we examine the impact of device variation and present hardware strategies to mitigate performance degradation, showing improvements by a significant margin.

4.3 Results

4.3.1 Voltage-dependent switching of memristors

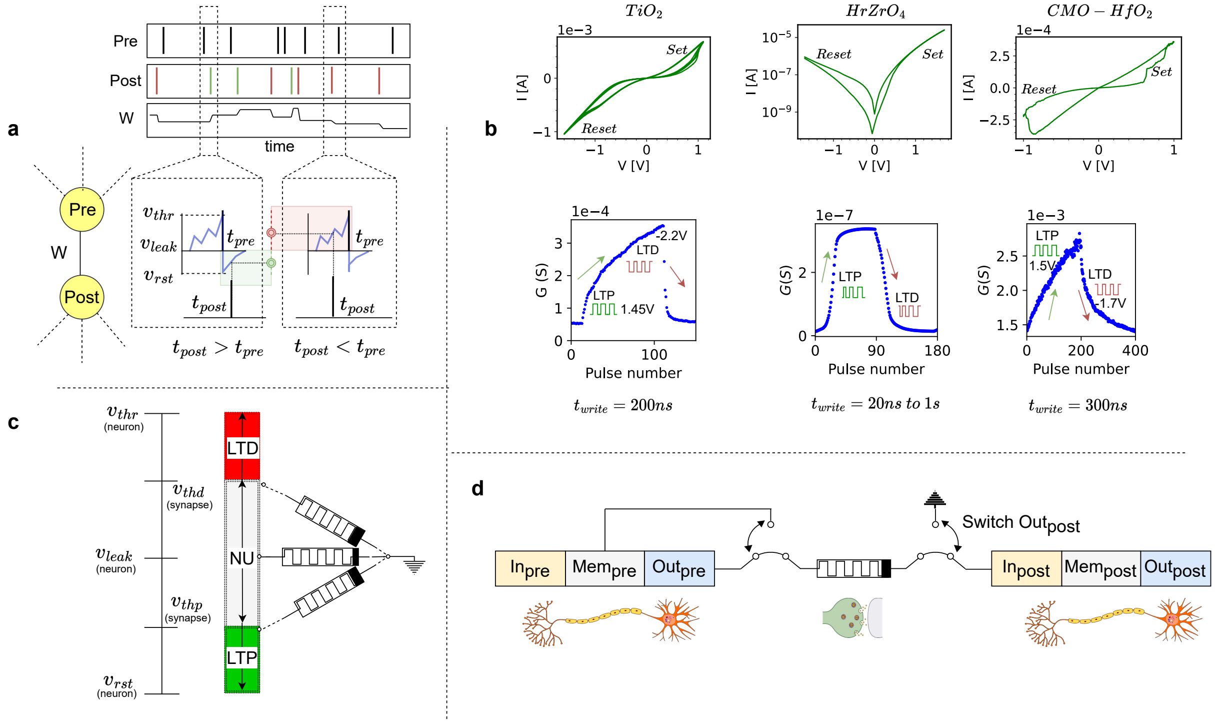


Figure 4.1 Schematic representation of the VDSP learning rule implemented in a memristive synapse between a pre- and postsynaptic spiking neuron. **a** Theoretical framework of VDSP in determining the causality of pre- and post-neuron spike events. The synapse is potentiated when the post-neuron spikes after the pre-neuron, and is depressed when the pre-neuron spikes following the post-neuron spike event. **b** Cumulative long term potentiation/depression (LTP/LTD) plot obtained with the response by pulses of positive polarity to show LTP and fixed magnitude followed by ones with negative polarity to induce LTD. DC sweeps (I-V) characteristics loops for all three devices is shown below. **c** Regions for long-term potentiation (LTP), long-term depression (LTD), and no update (NU) are based on voltage-regulated, threshold-dependent memristor switching. Schematic representation of VDSP for memristor programming. **d** Circuit-level depiction of VDSP on a single memristive synapse. The output from the post-synaptic neuron controls the switch between the inference and weight update phases. In, Mem, and Out represent the input terminal, membrane potential, and output terminal respectively for pre- and post-synaptic neuron.

Synaptic memory, based on Hebbian learning principles, captures the history of causal and anti-causal spike pairs between the pre-neuron and post-neuron. Using voltage-dependent synaptic plasticity (VDSP), the recent activity of the pre-neuron can be inferred from the neuron's membrane potential. A low membrane potential corresponds to a recent firing event, while a high membrane potential signals an imminent spike, as illustrated in Figure 4.1a. This learning mechanism is translated into the programming strategy for memristors, where synaptic conductance modulation is stored as a function of the history of applied voltages during successive spiking events.

In memristive devices, switching is primarily controlled by the programming voltage amplitude and duration. The distinctive current-voltage (I-V) characteristics (pinched hysteresis loop) of the three devices are shown in Figure 4.1b (top panels). In bipolar memristive devices, the hysteresis loop evidences the High Resistance State (HRS) and Low Resistance State (LRS) achieved after applying a voltage of opposite polarity. HRS and LRS define the switching range of the device in its binary regime. All three devices show very distinctive I-V hysteresis signatures that are linked to the physical mechanisms involved during switching. In oxide-based memristive devices, such behavior is intimately linked to the balance between drift and diffusion [244] and results in different voltage-resistance dependencies. In ferroelectric devices, switching dependency is associated with nucleation mechanisms in ferroelectric domains that govern the resistance states of the tunnel junction [237, 245].

Figure 4.1b (bottom) illustrates how switching can be controlled in an analog way when the programming voltage is applied as a sequence of pulses. The gradual change of conductance during the increase of resistance (LTD) and decrease of resistance (LTP) can be advantageously used to implement online learning. In this example, LTP and LTD are obtained with constant amplitude pulses, and the transitions evidence the cumulative effects that can be obtained by adjusting only the pulse duration. This scenario (i.e., gradual switching through cumulative effects of identical pulses) is the most straightforward way to implement learning in ANNs and SNNs, as each learning event can be associated with the application of a single pulse without adapting the pulse shape, and actual learning results from the repetition of the same learning signal. More complex situations occur when both pulse amplitude and pulse duration can be modulated during the learning signal, with the benefit of a larger dynamic range of programming and a higher number of states available [19].

From Figure 4.1b (bottom), all three technologies present different signatures in their analog regimes that can be further analyzed in terms of the number of states available, linearity of the transition, and min/max resistance states. For instance, implementing online learning in ANNs would favor the highest number of states between the HRS and LRS, the most linear transition, and the least variability between the HRS and LRS of different devices to map the backpropagated error signal with the highest accuracy. In the context of SNNs, the impact of these parameters is less clear and needs to be evaluated for each learning scenario. For example, conventional STDP could translate the magnitude of the LTP/LTD into different pulse durations [246] or a combination of pulse duration and pulse amplitude [247].

VDSP relies on the internal membrane voltage parameters as a probability of a spike being correlated or anti-correlated. The magnitude of the pre-neuron membrane voltage potential is directly associated with the magnitude of the learning signal when a post-synaptic spike is emitted. Figure 4.1c illustrates how the membrane voltage potential of the neuron can be translated into a programming voltage of the memristor. The neuron's membrane potential, which lies between the threshold (V_{th}) and reset potential (V_{rst}), can be mapped to the min/max voltage applied to the memristor, respectively. Such mapping results in a non-switching region for small voltages (when events are not strongly correlated) and LTD/LTP events when the voltage is above/below the switching threshold of the memristor.

This solution greatly simplifies the programming scenario and offers two main advantages for online learning implementation: (i) The membrane potential of the neuron can be converted into sub-microsecond pulses for memristor programming while capturing spike coincidences over millisecond-long windows based on the membrane potential time constant. Using short pulses allows for more precise analog control of memristor conductance switching and reduces energy consumption. (ii) Since the neuron's membrane potential provides a range of programming voltages, it can fully utilize the entire conductance range of the devices. This contrasts with fixed voltage pulse programming, where the ON/OFF ratio is often compromised to achieve gradual conductance modulation.

4.3.2 Electrical characterization

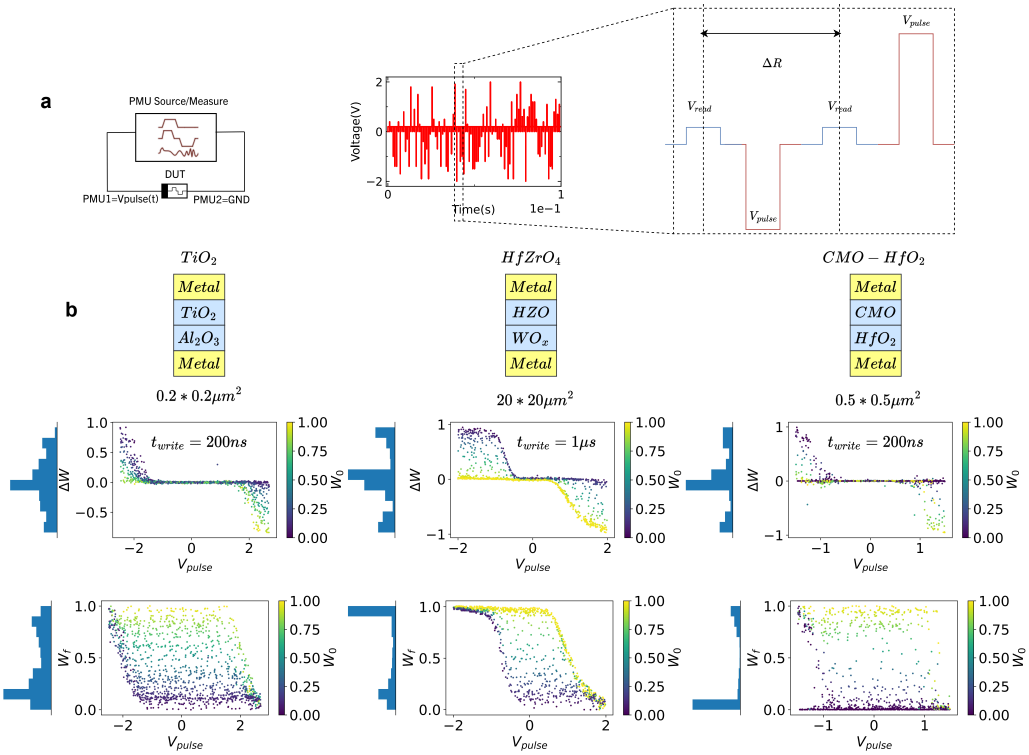


Figure 4.2 Device characterization protocol, stack, and switching behavior. **a** The characterization protocol involves applying write pulses of random amplitude with a constant pulse width, followed by read pulses. **b** The device stack for the three characterized devices includes metal top and bottom electrodes and switching oxide layers in between. **c** Weight change (ΔW) in relation to the applied programming pulse (V_{pulse}) and the initial weight (W_0) for the three device stacks: TiO_2, HZO, and CMO-HfO$_2$, from left to right. **d** Final weight (W_f) as a function of the applied voltage (V_{pulse}) and initial weight across the investigated device stacks. Histograms in (c) and (d) represent the change of weight and final weight, respectively, for the entire pulse protocol.

A specialized electrical characterization protocol was developed to characterize and model the voltage-dependent switching behavior. The dynamics of this switching were evaluated by applying short pulses (200ns or 1μs) at different voltage levels randomly distributed between V_{min} and V_{max}. Between each write pulse, the device's resistance was measured with a low-magnitude reading pulse. Using a random sequence helps to explore simultaneously the contribution of voltage amplitude and the impact of the initial state, as initial states are, in principle, randomly generated during the overall sequence. The weight change (ΔW) is plotted as a function of the applied voltage (V_{pulse}) and the initial weight (W_o) for the three device stacks (Figure 4.2c). The weight (w) represents the normalized conductance of the device and is calculated as follows:

$$w = \frac{g - g_{\min}}{g_{\max} - g_{\min}} \tag{4.1}$$

Where g_{min} and g_{max} represent the conductance in the HRS and LRS, respectively. Histograms in Figure 4.2c show the cumulative count of ΔW over the entire experiment. For the three devices, the highest probability of ΔW is centered around 0, which can be attributed to the absence of switching when voltages are below the SET and RESET threshold voltages (i.e., dead zone). Here SET refers to the process of switching the memristor to a low-resistance state (w=1), while RESET switches it to a high-resistance state (w=0). TiO_2 memories exhibit a more uniform distribution of weight change values compared to HZO and CMO-HfO_2, highlighting that this technology presents a more gradual switching for the given pulse amplitude and pulse duration chosen during the protocol (i.e., HZO and CMO-HfO_2 could present a more gradual switching if these parameters are modified, as evident in Figure 4.1b). In this work, we fixed the protocol for the three technologies to favor different device responses, which will be evaluated using the VDSP learning algorithm.

Asymmetries in the histograms also reveal that LTP and LTD are not equivalent, and that HZO shows a more gradual RESET transition while CMO-HfO_2 shows a more gradual SET transition. This effect is captured by the steepness of the transition below/above the negative/positive threshold in the heat map representation.

Figure 4.2d presents the final weight reached after a write pulse programming event, and the histograms present the cumulative count of the final weight. TiO_2 shows an overall homogeneous distribution of weight achievable during the protocol, while HZO and CMO-HfO_2 show a more asymmetric distribution. HZO tends to reach the LRS more often, while CMO-HfO_2 tends to reach the HRS more frequently. This is consistent with the asymmetry in ΔW observed in Figure 4.2c.

4.3.3 Device modeling

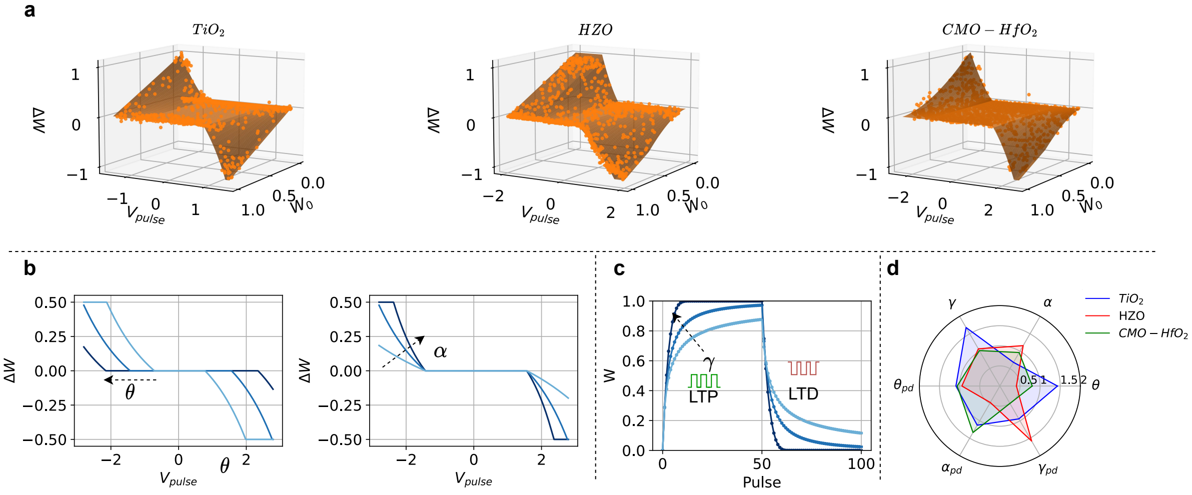


Figure 4.3 Model fitting. **a** 3D visualization of the modeled ΔW is shown as a surface plot against the applied voltage (V_{pulse}) and the initial weight (W_0). Characterization points for the TiO_2, HZO, and CMO-HfO$_2$ devices are also presented. **b** The impact of fitting parameters θ (left) and α (right) illustrates the variation in memristive switching threshold and curvature, respectively. Both plots show the change in weight as a result of a single voltage pulse, with $W_0 = 0.5$. **c** Cumulative potentiation/depression plot obtained from 50 pulses of positive polarity (LTP) followed by pulses of negative polarity (LTD). The resultant curve of final weight with respect to three values of γ is shown to illustrate the non-linearity fitting. **d** Fitted model parameters for the three device stacks.

We subsequently model the voltage-driven modification of the memristor's weight for various initial conditions and programming pulse magnitude. The alteration in weight (ΔW) is represented as the product of a switching rate function f dependent on the applied voltage v and a window function g dependent on the weight W:

$$\Delta W = f(v) \cdot g(W) \tag{4.2}$$

$$f(v) = \begin{cases} e^{-\alpha_p \cdot (v-\theta_p)} - 1 & \text{if } v < \theta_p \\ e^{\alpha_d \cdot (v-\theta_d)} - 1 & \text{if } v > \theta_d \end{cases} \tag{4.3}$$

Where α_p and α_d are exponential curvature fitting parameters for potentiation and depression, v is the applied voltage, and θ_p and θ_d are the threshold voltages for memristive device switching for potentiation and depression, respectively. The window function Equation 4.4

describes the dependence of the weight change on the initial state (w) and is responsible for the multiplicative effect during cumulative switching events.

$$g(W) = \begin{cases} (1-w)^{\gamma_p} & \text{if } v < \theta_p \\ w^{\gamma_d} & \text{if } v > \theta_d \end{cases} \tag{4.4}$$

Where γ_p and γ_d are the non-linear fitting parameters for potentiation and depression. This non-linearity implies that a particular voltage pulse has a diminished impact on the device's conductance when applied multiple times.

The model parameters for different resistive and ferroelectric memristive devices are compared in Figure 4.3d and presented in Table 4.1. The 3D representation with data points from the characterization and model is compared in Figure 4.3a to depict the model's accuracy (a 2D representation of model behavior on characterization points is shown in Supplementary Figure 4.1). The model effectively captures device behavior and allows for quantitative metrics such as threshold, curvature, and state dependence. The effect of θ and α on ΔW is shown for $W_o = 0.5$ in Figure 4.4b.

Next, to examine the state dependence, programming was performed using 50 LTP pulses of +1V followed by LTD pulses of -1V, as illustrated for three values of γ. Note that all the center lines in the three plots correspond to the fitted parameters for TiO_2. In Figure 4.4d and Table 4.1, all parameters (θ, α, γ) are compared for the TiO_2, HZO, and CMO-HfO_2 devices.

Device	α_p	α_d	v_{thp}	v_{thd}	γ_p	γ_d	**HRS (Ω)**	**LRS (Ω)**	**RMSE**	**sc**$_{pd}$
TiO_2	0.678	0.762	1.432	1.563	1.68	1.583	15k	2k	0.047	1.057
HZO	1.159	0.549	0.411	0.387	1.067	1.684	45M	17M	0.041	1.2
CMO-HfO_2	0.96	1.27	0.8	0.85	1.017	0.5	4k	1k	0.0141	1

Table 4.1 Model parameters for TiO_2, HZO, and CMO-HfO_2 devices along with the high resistance state (HRS) and low resistance state (LRS). Additionally, the table presents the root mean square error (RMSE) of ΔW, comparing characterization data with model predictions.

In particular, the TiO_2 device has the highest θ, implying a high switching threshold. This suggests a wide dead zone and a high SET/RESET voltage requirement. The α_p is, however, the lowest, implying a gradual dependence on weight change by increasing the SET voltage. The α_d is lower, pointing toward the fact that RESET is better controlled by voltage than SET. Finally, the γ, or state-dependent non-linearity, is highest in these devices. Previous studies have reported these behaviors [248], and they can also be observed in Figure 4.1b.

HZO exhibits the lowest α_d, implying that LTD is most gradual with respect to voltage (slightest curvature). The α_p is, however, twice the value of α_d, signifying a strong asymmetry in voltage-dependent switching. Finally, γ_d shows a strong asymmetry: state-dependent non-linearity is more evident in LTD. Such behaviors can also be observed in the IV sweep and LTP/LTD plots shown in Figure 4.1b, highlighting that the memristive device model strongly correlates with the known device behaviors. For CMO-HfO_2 devices, γ is the lowest compared to the other two devices, signifying linear multi-level programming, which is also evident in Figure 4.1b. The α has a higher LTD value than LTP, similar to TiO_2: gradual weight change occurs with increasing reset voltage. γ_d is smaller than γ_p, indicating that RESET is more linear than SET. However, it is important to note in this parametric model that different sets of parameters could explain the same switching response for a device. For instance, as per Equation 4.3, α and θ have an inverse relationship, where a higher value of α can still fit the data points with a low error rate if θ is lowered accordingly.

4.3.4 SNN benchmark

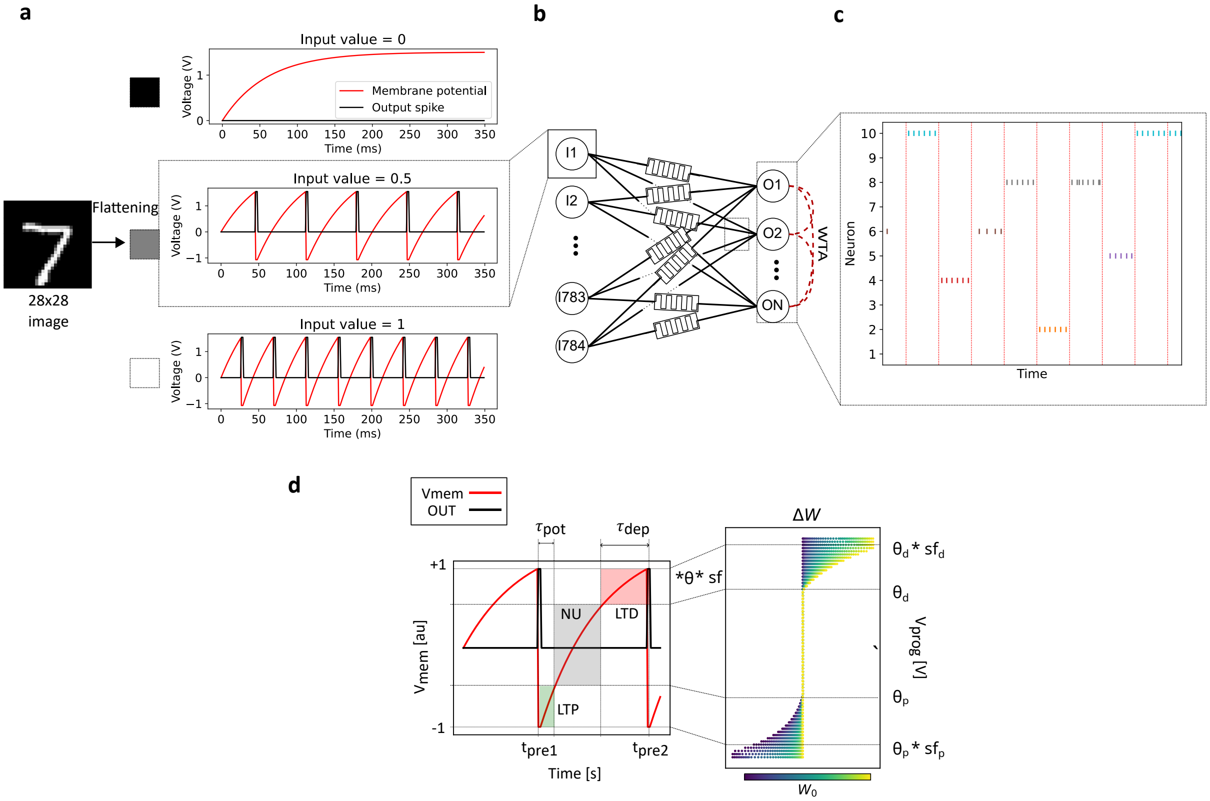


Figure 4.4 MNIST benchmark and simulation-device mapping in SNN. **a** The input LIF neurons integrate a constant input current based on pixel values to implement rate-encoding. **b** Each neuron receives input from the individual pixels of a 28x28 image and is fully connected to N output neurons through memristive synapses. **c** A raster plot for a network comprising 10 neurons illustrates the response to 10 sample inputs. The red-dotted vertical line distinguishes between different samples. **d** (left) The LIF neuron's membrane potential and spikes under constant stimulation input. (right) Weight change (ΔW) versus applied voltage. The scaling factor (sf) and switching thresholds (θ) are used to map the neuron membrane potential to the memristive device programming window. Long-term potentiation (LTP), long-term depression (LTD), and no update (NU) can be adapted to the memristive device requirements.

In order to assess learning capabilities, we conducted training on a Spiking Neural Network (SNN) using the MNIST dataset to recognize handwritten digits. In this method, we supplied the input pixels as constant currents to the encoding layer (refer to Figure 4.4a), where the Leaky Integrate-and-Fire (LIF) neurons convert the image's pixels into spike trains with frequencies proportional to their respective intensities. Additionally, we introduced Gaussian noise into the input pixels to induce stochastic sampling of the membrane potential (and thus programming voltage) during weight update events. This means that pixels receiving active input undergo different degrees of weight update magnitude. This enhances the realism of the evaluation by accounting for environmental noise and fluctuations in process temperature in encoding circuits, and also replicates the stochasticity observed in biological systems (Poisson distributed spikes).

These spikes are weighted by the memristive synapse (Figure 4.4b) and integrated by neurons in the output layer. The neurons in the output layer are connected through a winner-take-all (WTA) topology, which leads to only one active output neuron at a given instance, corresponding to the network's decision, as shown in Figure 4.4c. A detailed description of the neuron model, training/evaluation procedure, and hyper-parameters is provided in the Methods section.

A scaling factor (sf) is used to tune the actual voltage applied to the memristor so that it matches the operational range of the memristive device. This factor essentially translates the computing signals (in the form of membrane potential) into the appropriate voltage levels that a memristor requires for switching. The relationship between these parameters can be expressed as follows:

$$V_{prog} = Vmem \cdot sf \cdot \theta \tag{4.5}$$

In this equation, the programming voltage (V_{prog}) is the product of the neuron membrane potential (V_{mem}), the memristor's fitted threshold value (θ), and the scaling factor (sf). Figure 4.4d captures the impact of the memristive threshold and scaling factor on the temporal response sensitivity of VDSP, i.e., the time window in which a post-synaptic spike could occur for the synaptic weight to change. In particular, if a postsynaptic spike occurs between t_{pre1} and $t_{pre1} + \tau_{pot}$, the synapse will experience potentiation. On the other hand, a postsynaptic spike that occurs between t_{pre2} and $t_{pre2} - \tau_{dep}$ will lead to depression. These areas of LTP and LTD depend on whether the programming voltage would be able to surpass the memristive switching threshold. Since the programming voltage depends on the membrane potential, θ, and scaling factor (shown in Equation 4.5), the scaling factor directly influences the regions of LTP and LTD, thereby affecting the temporal sensitivity window of VDSP. For instance, a lower scaling factor would necessitate a higher membrane potential for the programming voltage to exceed the memristive switching threshold. As a result, the pre-neuron must be closer to firing, leading to a shorter time difference required between the pre- and post-neuron spikes to induce a weight change.

In the case of VDSP, the degree of change in weight in response to a single training sample depends on (i) the number of update instances, which depends on the spiking frequency of the output neuron, (ii) how many update instances fall into the LTP/LTD regions, and

(iii) the magnitude of weight change defined by the fitted parameters of the memristive model. The first is tunable by changing sample presentation duration, leak rate, and the threshold of output neurons, i.e., the network hyper-parameters. The second and third depend on the scaling factor and the fitted parameters or physical switching characteristics of the memristive devices. Thus, it is essential to adjust the scaling factor for both the memristive device and the classification problem.

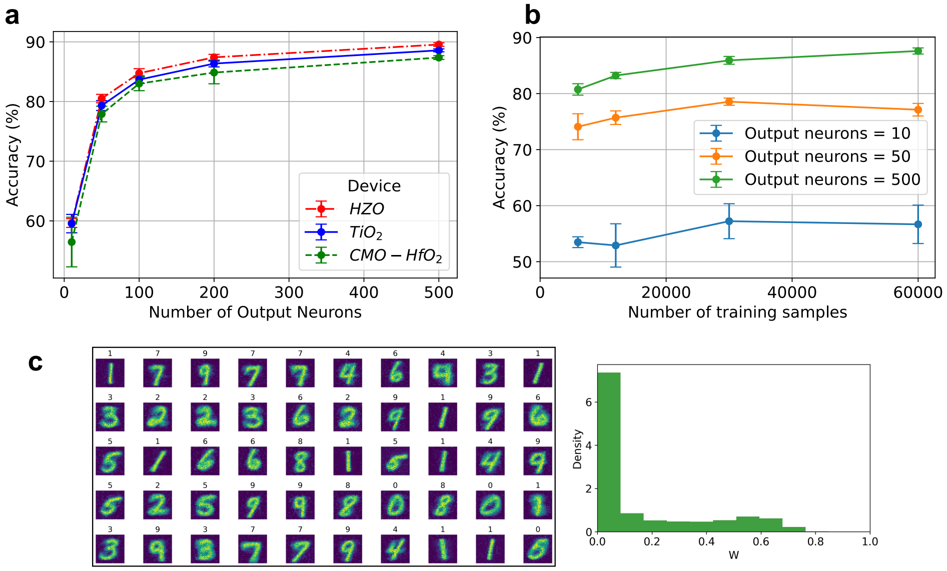


Figure 4.5 MNIST benchmark results. **a** The test accuracy is shown as a function of the network size, indicated by the number of output neurons, for all three devices. The network was trained with three epochs of the MNIST dataset, and the average and standard deviation from five experiments with different initial weights are represented as error bars. **b** Performance evolution of the network based on the number of training samples used, for networks with 10, 50, and 500 neurons. The results correspond to the TiO_2 fitted model. **c** The weight plots and histogram after training are displayed for a network comprising 50 neurons.

The training subset of the MNIST database was used to perform unsupervised learning and evaluation for SNN for up to three epochs. Figure 4.5a illustrates the resulting performance for networks with 10 to 500 output neurons, trained with three epochs of 60,000 MNIST samples. For a network of 10 output neurons, the recognition rate was 60%, which increased to more than 88% for a network of 500 neurons. In addition, to evaluate incremental learning, a network of 10, 50, and 200 output neurons was trained up to a single epoch, and the labels were assigned by presenting 10,000 unseen digits from the remaining dataset. The experiments were carried out for five different initial conditions (weights), and the average and standard deviation are shown as error bars in Figure 4.5b.

A similar analysis for incremental learning with HZO and CMO-HfO_2 device parameters is illustrated in Supplementary Figure 4.4 and Supplementary Figure 4.5.

Larger networks allow for the learning of receptive field areas (inspired by neuron selectivity in neural cells [249]) or distinct features for each class [87]. In other words, non-overlapping representations are essential to distinguish numbers with overlapping characteristics, such as the vertical line of one and nine. For example, the learned weights of a network of 50 output neurons are shown in Figure 4.5c, where complementary representations of each class can be seen. Competitive learning [250] was implemented using the winner-take-all (WTA) mechanism in the neurons of the output layer to learn such complementary features. In addition, the histogram of the network's weights at the end of training shows a bimodal distribution (Figure 4.5c), resulting from soft clipping due to the state-dependent multiplicative component of the VDSP update function. This clipping, a known non-linearity [251] of the memristive transition during repeated LTP/LTD programming, as shown in Figure 4.1b and is beneficial for online learning as it promotes stable learning without forgetting previously learned patterns [252].

Table 4.2 compares the performance with previously reported works, showing that HZO exhibits the highest performance, followed by TiO_2 and CMO-HfO_2. In sum, the learning efficiency of VDSP for all three devices is equivalent to or superior to their STDP counterparts. All the network parameters, such as the degree of noise, leak rate of input neurons, threshold, and output neurons, were optimized through the TiO_2 device model for a network of 10 output neurons. Only the scaling factor for LTP and LTD was optimized through grid search (sf_p, sf_{pd}) for the three devices and number of samples in the incremental training experiment, with the optimized values plotted in Supplementary Figure 4.2. The sf plays a similar role to the learning rate in ANNs as it regulates the degree of weight change.

It is essential to highlight that, unlike classical machine learning optimizations and gradient-based learning in ANNs, the notion of learning rate is not trivial in SNNs, specifically in cases of unsupervised local online learning where there is no batch processing. The magnitude of the weight change depends on the difference in spike times between the presynaptic and postsynaptic neurons. The choice of learning rate is critical to avoid local minima or continuous oscillations, as a sub-optimal rate slows learning and hinders convergence. On the other hand, a higher learning rate can cause instability in weights, lead to forgetting previously learned information, and make convergence difficult.

4.3.5 Impact of device variations

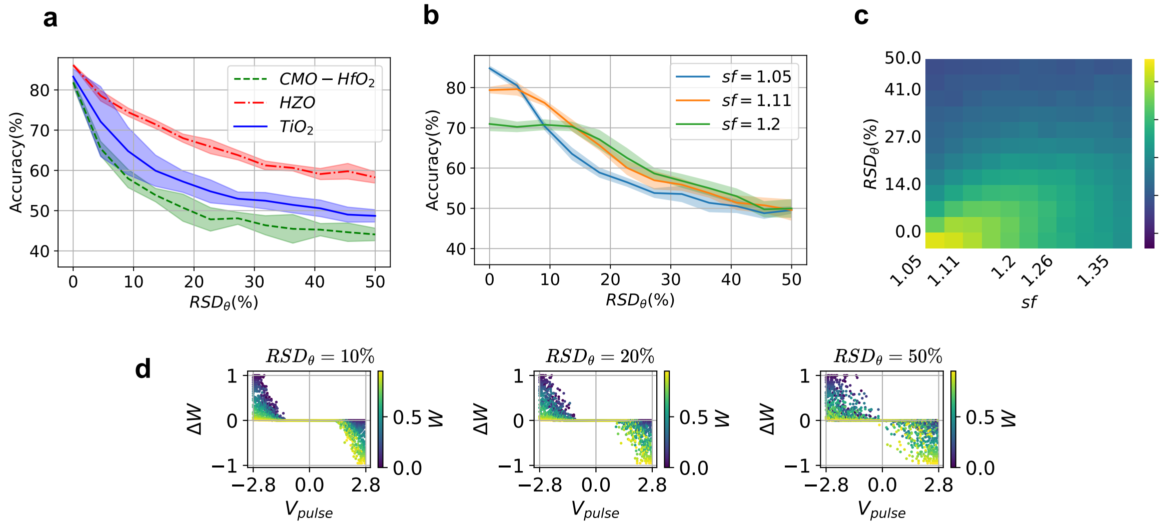


Figure 4.6 Impact of device-to-device variability in switching threshold. **a** The switching thresholds (θ) for the 784x200 synapses are sampled from a normal distribution centered on the fitted model parameter. The relative standard dispersion, represented as $\frac{\sigma}{\mu}$, varies and is noted as RSD_θ. **b** The resulting accuracy is shown for three distinct values of sf for TiO_2 device parameters. (Note that the other two devices follow a similar trend, as illustrated in Supplementary Figure 4.4 and Supplementary Figure 4.5. **c** A detailed grid search of sf and RSD_θ was conducted, and the resulting average accuracy over ten experiments is displayed as a 2-D heatmap. **d** Plots of device characteristics depicting ΔW versus V_{pulse} to demonstrate the variability effects for three $\frac{\sigma}{\mu}$ levels: 0.1, 0.2, and 0.5 (arranged from left to right).

The SET/RESET voltage or the switching threshold of memristive devices varies from device to device, particularly in the case of analog switching. The corresponding memristor model parameter (θ) is sampled from a normal distribution, with the mean centered around the fitted parameter and a relative standard deviation (RSD) that was incrementally changed to study its impact on the device's behavior. RSD is defined as the ratio of the standard deviation to the arithmetic mean of a normal distribution ($\frac{\sigma}{\mu}$). A histogram of sampled threshold distribution for different values of RSD is shown in Supplementary Figure 4.3. Figure 4.6a illustrates how network performance is affected by different degrees of threshold variability for a neural network with 200 output neurons using model parameters from three different devices. As the RSD increases, the performance of the network degrades. For the TiO$_2$ device, with 20% variability in the switching threshold, the performance of 82% drops to 56%. The corresponding fitted parameter (θ) threshold of these devices was 1.4V and 1.5V for LTP and LTD, respectively. A similar analysis was conducted for the HZO device (see Supplementary Figure 4.4), and its performance dropped to 68%. These FTJs have the lowest thresholds, around 0.4V, but the highest α or curvature. The achievable conductance range in ferroelectric devices is strongly correlated with the magnitude of the programming voltage. Therefore, a greater scaling factor could be used, as detailed in Supplementary Figure 4.2. The low switching threshold makes STDP implementation with pulse overlapping challenging. The pulse overlapping technique requires transmitting spikes through sub-threshold pulses for non-destructive reading; thus, the maximum programming voltage for LTP/LTD is limited to twice this threshold.

The scaling factor is a crucial parameter that determines the probability of switching in memory devices. A baseline value of 1.05 prevents devices with a higher threshold from switching, limiting the number of devices that can synapse to learn. To accommodate threshold variations and increase the number of such devices, a larger scaling factor should be used. In Figure 4.6b, it is demonstrated that a high scaling factor of 1.2 keeps performance stable, with only a slight decrease from 71% to 68% when the variability is 20%. On the other hand, a scaling factor of 1.05 achieves a performance of over 82% without considering device mismatch. However, when variability is introduced, the performance drops to less than 60%. This highlights the importance of selecting the right scaling factor to maintain strong performance, especially in the presence of variability.

The use of a high scaling factor can help reduce performance drops caused by variability, but it can also hinder generalized learning that utilizes the entire dataset. This creates a trade-off between stability and learning precision. A high scaling factor increases the magnitude of the programming voltage, leading to a greater weight change magnitude for each update. However, when training with a dataset like MNIST, the goal is to generalize over all the training samples and learn incrementally from each sample. Therefore, it is crucial to regulate the impact of each training sample on the weights, which can be achieved by reducing the scaling factor. A parametric sweep between RSD and sf, shown in Figure 4.6c, further illustrates how different scaling factors impact device performance. Additionally, Figure 4.5d plots the influence of threshold mismatch on the switching function of the devices for different variabilities in switching parameters. These results are based on the TiO_2 model parameters, with similar trends observed for the other two device stacks (CMO-HfO_2 and HZO), as shown in Supplementary Figure 4.4 and Supplementary Figure 4.5. This consistency across different materials highlights a generalizable behavior of the scaling factor for accommodating the threshold mismatches across multiple resistive memory devices. In summary, even with 50% variability, the performance across all three devices remains above 45%.

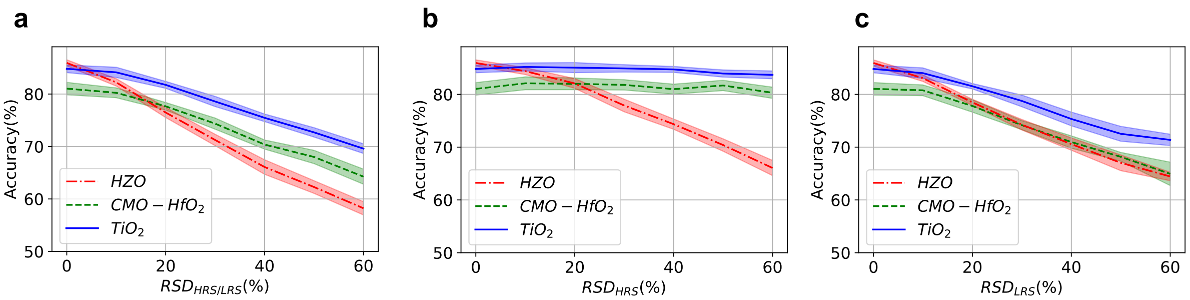


Figure 4.7 Impact of device-to-device variability in HRS and LRS levels. **a** The HRS and LRS for the 784x200 synapses are sampled from a normal distribution, centered on the fitted model parameter. The relative standard dispersion, represented as $\frac{\sigma}{\mu}$, varies and is noted as RSD_θ. The resulting accuracy is shown for three devices. **b** The impact of HRS is isolated and evaluated. **c** The impact of LRS is isolated and evaluated. Each experiment was repeated with ten different initial conditions, with the lines and shades depicting the mean and standard deviation of measured accuracy.

The impact of variations in both the low resistance state (LRS) and high resistance state (HRS) on a 784x200 synaptic network was analyzed (Figure 4.7). Device-to-device variability was simulated by sampling the HRS and LRS of each synapse from a normal distribution centered around the measured device values. The standard deviation ($\mathrm{RSD_{HRS}}$ or $\mathrm{RSD_{LRS}}$) of these distributions was adjusted to investigate its effect on the network's recognition accuracy. Both HRS and LRS were sampled simultaneously from their respective distributions (Figure 4.7a). To further differentiate the effects of HRS and LRS individually, Figure 4.7b presents the impact of variability in HRS, while Figure 4.7c presents the effects of LRS mismatch ($\mathrm{RSD_{LRS}}$) on the network's performance.

TiO_2-based memories demonstrated the highest resilience to variations in both high and low resistance states, as illustrated in Figure 4.7a, followed by CMO-HfO_2 and HZO. This resilience trend aligns with their measured resistance ranges (Table 4.1), where TiO_2 exhibits the highest ON/OFF ratio of 7, followed by CMO-HfO_2 and HZO, with their corresponding ratios being 4 and 3, respectively. TiO_2 and CMO-HfO_2 appear unaffected by HRS variations (Figure 4.7b), likely due to their larger ON/OFF ratios. In these devices, the HRS states (with weight parameter $w = 0$) have minimal influence on the activity of the subsequent layer, which reduces performance sensitivity. However, when the HRS approaches the LRS, the HRS can affect the output neuron's activity, a critical factor in weight updates for VDSP. As such, significant HRS variability can degrade performance in devices with smaller resistance ranges. Lastly, Figure 4.7c reveals that all three devices are equally affected by variations in LRS. However, the maximum accuracy drop with 20% variation in LRS is less than 10%, underscoring the system's robustness against device mismatch. This demonstrates the relative insensitivity of LRS variations compared to HRS, particularly in devices with larger resistance ranges.

Ref	Device	Circuit	Architecture	Plasticity	Accuracy
[210]	PCM	8-R Multicell	784x50	STDP	70%
[253]	PCM	2-R Differential	784x350x10	Supervised	80%
[254]	MTJ	1-R	784x[100]100	Stochastic STDP	70%
[211]	HfOx/TaOy	1T1R	784x50	STDP	75%
[255]	2D h-BN	1R	784x500	STDP	68%
[256]	PCM	6T2R	724x500	STDP	73.6%
[257]	Ag/Si ECM	1R	784x50	Simplified STDP	80%
This work	TiO_2	1R	784x50	VDSP	**79%**
This work	HZO	1R	784x50	VDSP	**81%**
This work	CMO-HfO_2	1R	784x50	VDSP	**78%**

Table 4.2 Comparison of the current study with previous memristive-based online learning benchmarks with MNIST. Different device technologies, including Phase change (PCM), Magnetic tunnel junction (MTJ), and electrochemical metallization (ECM) with circuit configurations, are tabulated for classic, stochastic, and simplified versions of STDP with respective network architectures.

4.4 Discussion

4.4.1 Device-Specific Switching Dynamics

There are two types of switching that we observe. The first type involves a pulse of the same magnitude and pulse width that causes the device to transition between different conductance states. This is also known as cumulative switching, and it is exploited by STDP for online learning. The transition is non-linear, and the magnitude of weight change reduces when moving to the boundary. The second mechanism shows that the conductance state strongly depends on the switching voltage or pulse width. By adjusting the programming voltage level, we can expand the boundaries of switching voltage. The three devices exhibit different degrees of cumulative or voltage-dependent switching in LTP or LTD, depending on the dynamics of underlying mechanisms like oxidation, filament rupture, or ferroelectric domain switching. Some processes are self-limiting, allowing robust control, while others are not and lead to abrupt behaviors. For instance, (i) in TiO_2, the SET process involves slower drift of oxygen vacancies [258] and is more gradual or cumulative in comparison to the RESET process which involves conductive filament melting. These resistive devices often experience noisy switching due to variability in oxygen vacancy migration [259, 260]. They have a high ON/OFF ratio, so the learning is resilient to variations in the ratio. (ii) In HZO-based FTJs, the SET process tends to be more abrupt compared to the RESET process, which is typically gradual and exhibits better linearity [261]. (iii) CMO-HfO$_2$ devices often display variability in resistance states (HRS and LRS) due to random oxygen vacancy movements, and exhibits more granularity in potentiation over depression [244]. VDSP-based learning effectively mitigates differences in device parameters, leading to similar MNIST test-set recognition rates despite the distinct parameter combinations exhibited by the devices due to different underlying physical mechanisms of switching. Scaling factors and the asymmetry in the scaling factor can be tuned with respect to the device threshold and asymmetry in switching dynamics. This scaling factor is a important determinant of network's learning rate. There exists a trade-off between gradual weight updates with a low scaling factor and abrupt updates with a higher scaling factor. While high scaling factor improve resilience to variations in the device's switching threshold, gradual updates help the model generalize better by learning incrementally from each sample.

4.4.2 Programming Strategy and Variability

Resistive memories exhibit significant variability when scaled down due to the resolution limits of semiconductor patterning techniques and variations in fabrication parameters. This variability causes each device to exhibit slight differences in performance. These variations pose particular challenges in analog computing, complicating precise and accurate programming. In the case of SNNs with memristive weights, this variability directly affects the synaptic learning process, as different devices may exhibit different switching thresholds and ON/OFF states. This inconsistency makes it difficult to reliably program the desired resistance states across all devices, ultimately impacting the overall performance and accuracy of the network. One approach is to map the device's switching characteristics onto the learning algorithm, including its threshold voltages and asymmetric response to programming pulses. This requires selecting an appropriate learning rate to adjust synaptic weight updates, compensating for device variations. Fine-tuning network hyperparameters, such as the scaling factor that links neuron membrane potential to memristive programming voltage, ensures robust learning. By adapting these scaling factors to account for different switching thresholds, we show that the network can maintain reliable performance despite device variability. This robustness is driven by two key factors: the application of online learning and the advantages provided by VDSP. Firstly, it is well-established that the challenges posed by variability and noise in memristive devices can be mitigated through online learning strategies [241]. Through continual adaptation, online learning enables the system to overcome fluctuations and inconsistencies inherent to the hardware. Device behaviors like gradual switching, which allows for fine-grained adjustments to synaptic weights, can mitigate the effects of variability. Additionally, devices with a broader range of conductance states can support more precise weight changes, reducing the impact of any single variation. For instance, memristors with cumulative switching, where the conductance increases gradually with repeated pulses, tend to be more resilient to slight variations in pulse width or amplitude, providing a built-in mechanism for error correction. Secondly, and perhaps more crucially, VDSP principles enhance resilience against device mismatch. In traditional STDP implementations that use pulse overlapping techniques, the amplitude and width of the programming pulses must be carefully adjusted based on the characteristics of the synaptic device. Furthermore, the programming pulse width is tailored to meet the specific requirements of the Hebbian learning window, ensuring that spike correlations occur within the window to induce synaptic changes. However, VDSP simplifies this process by utilizing a continuous physical quantity—namely, the neuron membrane potential—amplified and directly applied for synaptic programming. A higher-than-necessary amplification factor ensures that even mismatched memristors with high switching thresholds are programmed successfully. Additionally, noise originating from the input layer—whether from the encoding circuit or environmental sources—contributes to the membrane potential and, consequently, the programming voltage. This added noise, combined with variability among memristors, introduces a stochastic element into the learning process. This randomness allows for finer weight adjustments, which can smooth out inconsistencies due to device mismatch and improve learning accuracy.

4.4.3 Conclusion

This article addresses designing and tuning computing circuits based on memristive device characteristics and demonstrates neuron state-based online learning with VDSP. This method is local in space and time, as it requires the application of the instantaneous analog membrane potential of the pre-synaptic neuron. The learning process does not require complex pulse-shaping circuitry and models the classical exponential dependence of weight change on the difference in spike time between pre- and post-synaptic neurons. The slow neuron dynamics enable the memristive device to adapt its conductance based on the frequency and timing of spikes, responding to changes in membrane potential. This coupling is essential for enabling long-term potentiation and depression (LTP and LTD) in a bio-realistic manner. By combining the slow dynamics of neurons (voltage over time) with the voltage-dependent switching of memristors, we can achieve learning with a bio-realistic time scale using sub-microsecond programming pulses. Using short pulses saves power, increases endurance, and improves learning due to controlled switching. Limiting the pulse width reduces the energy consumed during each programming event, particularly in large-scale neuromorphic systems where energy efficiency is critical. Shorter pulses also decrease wear on the devices, enhance their endurance, and offer more precise control over weight updates. We characterized and modeled two resistive devices and a ferroelectric memristive device, each exhibiting distinctive switching behaviors. The characterization of voltage-dependent switching behavior was achieved by applying pulses of random voltage magnitude. Different devices exhibit properties such as switching threshold, non-linearity, and variability. A generalized model was proposed to describe resistive switching as dependent on the magnitude of the applied voltage and the resistance state of the device, with the fitted model closely resembling the characteristics of the tested devices. The model parameters accounted for fundamental memristive properties such as threshold, non-linearity, and asymmetry, enabling the validation of learning efficiency across a spectrum of device behavior deviations. Moreover, immunity to deviations in model parameters was observed by sampling the device model parameters from a distribution with varying spreads. Importantly, performance deterioration due to variability can be mitigated by tuning the scaling factor.

The proposed plasticity, implemented with simple CMOS circuits [262], allows large-scale systems with limited energy and space constraints to carry out online learning for real-world pattern recognition applications. In future work, we plan to expand this approach to more complex patterns by using current-limited switching through 1T1R devices and investigating the relationship between the device's material stack and the resulting intermediate conductance states. The VDSP-based programming provides high-resolution memristive learning, effectively linking neuron dynamics with memristive properties to detect and respond to long-term and temporal patterns. However, because model parameters are heavily influenced by programming conditions, attributing performance solely to the device stack is speculative, and further research is required to fully understand these interactions.

4.5 Methods

4.5.1 Device Fabrication

The fabrication recipe for the TiO_2 devices used in this study can be found in [20], and the same for the HZO and $CMO - HfO_2$ devices is detailed in [238] and [239], respectively.

4.5.2 Characterization setup

Electrical measurements were performed on an Agilent B1500A semiconductor analyzer and with a B1530A waveform generator/fast measurement unit (WGFMU). Write pulses were generated by a remote-sense and switch unit (RSU) module close to the probe and applied to the top electrode while the bottom electrode was grounded. The resistance of the device was measured at V = +/-100 mV with a high resolution source measurement unit (SMU) on the top electrode, while the bottom electrode was grounded.

4.5.3 Model fitting

The Levenberg-Marquardt least squares fitting algorithm (lmfit) [263] was used to fit the model parameters to the characterization data.

4.5.4 SNN simulations

Leaky Integrate-and-Fire (LIF) neurons [48], are simplified models of biological neurons, making them efficient to simulate within a SNN simulator. This neuron model was used for the presynaptic neuron layers. The corresponding equation is

$$\tau_m \frac{dv}{dt} = -v + I + b \tag{4.6}$$

where τ_m denotes the membrane leak time constant, v represents the membrane potential, which decays to the resting potential (v_{rest}), I is the injected current, and b is a bias term. When the membrane potential exceeds a threshold level (v_{th}), the neuron fires a spike. Subsequently, it becomes unresponsive to any input during the refractory period (t_{ref}) and the neuron potential is reset to the voltage (v_{reset}).

In the output layer, an adaptive leaky integrate and fire (ALIF) neuron model was used, which has an additional state variable n, which increases by inc_n with each spike, and its value is deduced from the input current. This leads the neuron to decrease its firing rate over time when exposed to high input currents [216]. The state variable n decays with a time constant τ_n :

$$\tau_n \frac{dn}{dt} = -n \tag{4.7}$$

Gaussian noise was used to induce stochasticity in the input layer, which creates a jitter in pixels and samples the learned features at each output spike. Bias current was applied to background pixels to penalize inactive pixels and is essential for regularization and preventing one neuron not to learn all digits (equal amount of de-potentiation of nonactive pixels for learning distinguishing features). Each image was presented for 40 ms, resulting in at most 3 spikes in the input layer per sample. Hard Winner takes all based lateral inhibition was applied in output layer, in which, all neurons were inhibited for a period of τ_{wta} on firing event of any neuron. The network parameters were tuned using genetic search with Optuna [264] package for 1000 experiments for the parameters of the TiO2 device. Afterwards, all network parameters were the same for all network sizes, epochs, and three devices. The training was performed without using labels, and at the end of training, the weights were fixed, and the last 10,000 samples were used for assigning the class to each output neuron. Subsequently, the samples from the MNIST dataset's test set were used to evaluate the recognition rate.

Acknowledgments

We want to acknowledge engineers from 3IT, Sherbrooke, for TiO_2 device fabrication. The authors acknowledge the staff of Binnig and Rohrer Nanotechnology Center (BRNC), Zurich, for HZO and CMO-HfO_2 device fabrication.

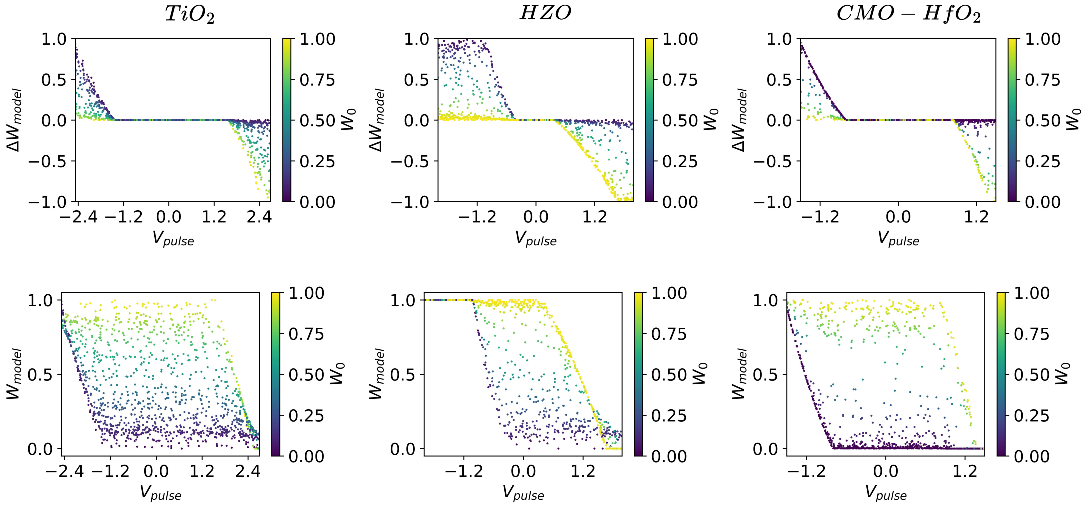
Funding

We acknowledged financial supports from the EU: ERC-2017-COG project IONOS (# GA 773228) and CHIST-ERA UNICO project. This work was also supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) (No. 559730) and Fond de Recherche du Québec Nature et Technologies (FRQNT). FA thanks the MEIE for support through the chair in neuromorphic engineering.

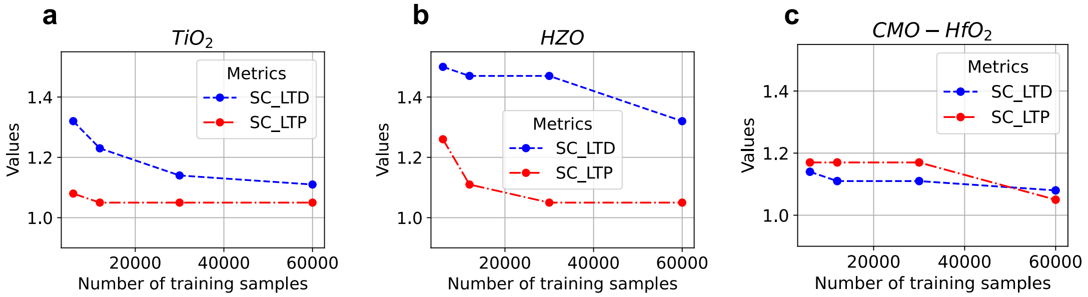
Author Contributions Statement

J.H., N.G. formulated characterization protocol. J.H., L.B.L., and D.F. performed device characterizations. N.G. devised and fitted the model. N.G. and I.B. implemented and performed the SNN simulations. L.B., V.B., T.S., and D.F.F. performed device fabrication. F.A., D.D., Y.B., D.Q., J.M.P., and B.J.O. supervised and administered the current study. N.G. and F.A. wrote the initial draft of the manuscript. All authors provided critical feedback and helped shape the research, analysis, and manuscript.

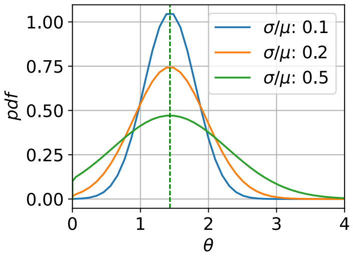
4.6 Supplementary Information



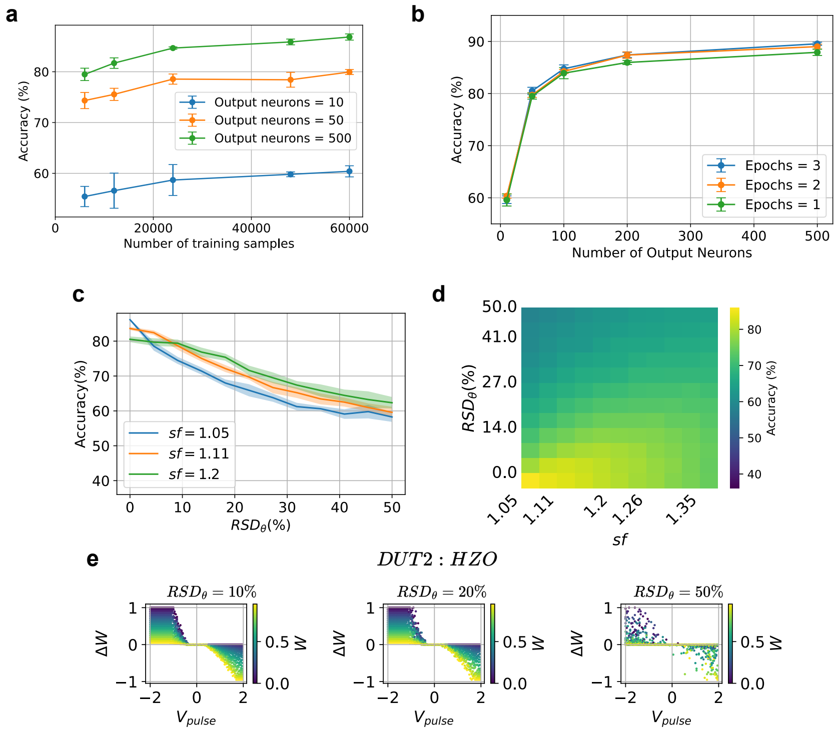
Supplementary Fig. 4.1 The fitted model's prediction on characterization data points displays the weight change (ΔW) and final weight (W_f) in relation to the applied voltage (V_{pulse}) and initial weight (W_0) for TiO_2, HZO, and CMO-HfO$_2$ measurement points (from left to right).



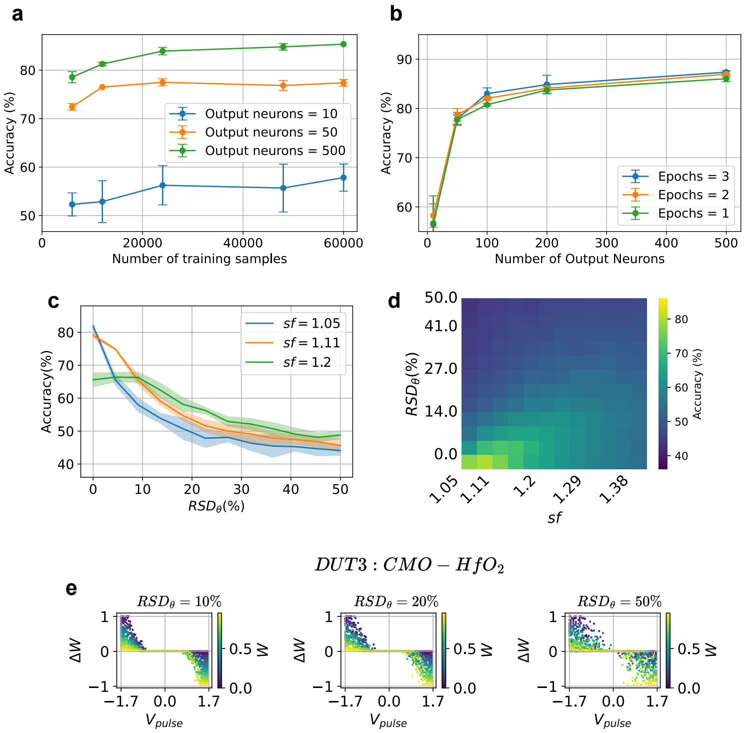
Supplementary Fig. 4.2 Scaling factor based on the number of training samples. The optimal scaling factors for LTP (sf_p) and LTD (sf_d) are depicted as a function of the number of training samples for a network consisting of 500 output neurons. The plots are presented for three devices: TiO_2, HZO, and CMO-HfO$_2$ device parameters (from left to right).



Supplementary Fig. 4.3 Probability distribution function of θ for TiO_2, showing variability in the form of relative standard dispersion $(\frac{\sigma}{\mu})$.



Supplementary Fig. 4.4 MNIST benchmark results for HZO device. **a** Evolution of test performance in response to iterative training through examples from the MNIST dataset for 10, 50, and 500 output neurons. **b** Dependence of testing accuracy on the number of output neurons and training epochs. Each experiment was repeated five times with different initial weights, and the error bars show the standard deviation. **c** Impact of variability in switching threshold (θ) plotted for three different scaling factors. **d** A detailed grid search of sf and RSD_θ was conducted, and the resulting average accuracy over ten experiments is displayed as a 2-D heatmap. **e** Impact of varying degrees of variation in θ on the device switching characteristics.



Supplementary Fig. 4.5 MNIST benchmark results for CMO-HfO$_2$ device. **a** Test performance progression observed during iterative training using MNIST dataset examples with 10, 50, and 500 output neurons. **b** Relationship between testing accuracy and the number of output neurons and training epochs. Each experiment was repeated five times with different initial weights, with error bars indicating the standard deviation. **c** The effect of variability in the switching threshold (θ) plotted for three distinct scaling factors. **d** A detailed grid search of sf and RSD_θ was conducted, and the resulting average accuracy over ten experiments is displayed as a 2-D heatmap. **e** Effect of varying the degree of θ variation on device switching characteristics.

CHAPTER 5

Versatile CMOS Analog LIF Neuron for Memristor-Integrated Neuromorphic Circuits

"Listen to the technology; find out what it's telling you." — Carver Mead

TABLE OF CONTENTS

5.1 Preface

Contribution to document

Toward our objective of implementing learning with analog memristive circuits, we have so far outlined the learning rule, Voltage Dependent Synaptic Plasticity (VDSP) in chapter 3, and in chapter 4, benchmarked it with memristive synaptic models. In the previous chapter, we emphasized the importance of adapting the network parameters of input and output neurons to accommodate a range of synaptic behaviors resulting from various gradual and abrupt SET/RESET transitions. It is crucial that the resting state and the reset potential of a neuron are distinct to accurately estimate the spike times from its membrane voltage. This distinction enables us to differentiate between neurons that have just fired and those that have been recently inactive. However, these differences in reset and resting potentials cause upward leakage (from reset to resting state) after a spike. Moreover, the rate of this membrane potential leakage directly influences the temporal window for VDSP-based learning. Thus, a configurable and bidirectional leak mechanism is essential for LIF neurons when interfacing with memristive synapses to support VDSP-based learning.

The following chapter (conference paper) introduces the first set of analog circuits aimed at building the self-learning Neural Building Block (NBB). This includes (1) a Low Dropout regulator (LDO) for stable reading of memristive resistance, (2) a Current Attenuator (CA) to downscale the read current for integration on a small, low-footprint capacitance in the neuron, and (3) an analog Leaky Integrate and Fire (LIF) neuron with a configurable bi-directional leak rate. The circuits were designed using a 130nm CMOS technology node and subsequently fabricated. The test structures, including the signal chain, were integrated into the primary chip, with 25 scribe lines placed on the last metal layer for probe testing. Synaptic reading was measured across a wide range of input resistances, from 1kΩ to 1MΩ, to establish the compatibility range for integrating memristive devices in the full NBB presented in chapter 6. Additionally, the tunability of neuron behavior was demonstrated by adjusting bias voltages to modulate key parameters such as pulse width, threshold, and leak rates, allowing for flexible control over the neuron's response characteristics.

Title: Versatile CMOS Analog LIF Neuron for Memristor-Integrated Neuromorphic Circuits

Title in French: Neurone LIF analogique CMOS polyvalent pour circuits neuromorphiques intégrés à memristor

Date: Dec 2024 (Publication) [262]

Status: Published in conference proceeding.

Conference: ACM/IEEE International Conference on Neuromorphic Systems (ICONS) 2024

Authors: Nikhil Garg[1,2,3,*], Davide Florini[1,2], Patrick Dufour[1,2], Eloir Muhr[4], Mathieu C. Faye[4], Marc Bocquet[4], Damien Querlioz[5], Yann Beilliard[1,2], Dominique Drouin[1,2], Fabien Alibart[1,2,3], Jean-Michel Portal[4]

Affiliations:

1. Institut Interdisciplinaire d'Innovation Technologique (3IT), Université de Sherbrooke, Sherbrooke, Québec, Canada
2. Laboratoire Nanotechnologies Nanosystèmes (LN2) – CNRS, Université de Sherbrooke, Québec, Canada
3. Institute of Electronics, Microelectronics and Nanotechnology (IEMN-CNRS), Université de Lille, Lille, France
4. Aix-Marseille Université, Université de Toulon, CNRS, IM2NP, Marseille, France
5. Université Paris-Saclay, CNRS, Centre de Nanosciences et de Nanotechnologies, Paris, France

***Corresponding Author:**

– Nikhil Garg – Nikhil.Garg@Usherbrooke.ca

Résumé

Les systèmes hétérogènes avec des circuits CMOS analogiques intégrés à des dispositifs memristifs à l'échelle nanométrique permettent un déploiement efficace de réseaux neuronaux sur du matériel neuromorphique. Les neurones CMOS à faible encombrement peuvent émuler une dynamique temporelle lente en fonctionnant avec des niveaux de courant extrêmement faibles. Néanmoins, le courant lu à partir des synapses memristives peut être supérieur de plusieurs ordres de grandeur, et il est obligatoire d'effectuer une adaptation d'impédance entre les neurones et les synapses. Dans cet article, nous mettons en œuvre un neurone analogique à fuite intégrée et à déclenchement (LIF) avec un régulateur de tension et un atténuateur de courant pour interfacer les neurones CMOS avec les synapses memristives. De plus, la conception du neurone propose une double fuite qui pourrait permettre la mise en œuvre de règles d'apprentissage locales telles que la plasticité synaptique dépendante de la tension. Nous proposons également un schéma de connexion pour mettre en œuvre des neurones LIF adaptatifs basés sur l'interaction à deux neurones. Les circuits proposés peuvent être utilisés pour s'interfacer avec une variété de dispositifs synaptiques et traiter des signaux de dynamiques temporelles diverses.

Abstract

Heterogeneous systems with analog CMOS circuits integrated with nanoscale memristive devices enable efficient deployment of neural networks on neuromorphic hardware. CMOS Neuron with low footprint can emulate slow temporal dynamics by operating with extremely low current levels. Nevertheless, the current read from the memristive synapses can be higher by several orders of magnitude, and performing impedance matching between neurons and synapses is mandatory. In this paper, we implement an analog leaky integrate and fire (LIF) neuron with a voltage regulator and current attenuator for interfacing CMOS neurons with memristive synapses. In addition, the neuron design proposes a dual leakage that could enable the implementation of local learning rules such as voltage-dependent synaptic plasticity. We also propose a connection scheme to implement adaptive LIF neurons based on two-neuron interaction. The proposed circuits can be used to interface with a variety of synaptic devices and process signals of diverse temporal dynamics.

Keywords: Neuromorphic computing, Analog circuits, In-memory computing, memristors, ASIC

5.2 Introduction

Analog circuits and devices can efficiently emulate neural dynamics in real-time by utilizing physical mechanisms such as Kirchoff's law through memristive synapses for signal transmission and capacitive charging on CMOS devices for temporal integration of current [265]. Such strategies are inherited from the seminal work of C. Mead [266] to implement silicon neurons with rich temporal dynamics at low power. More recently, nanoscale non-volatile memristive devices have been considered for synaptic function implementation and offer several interesting features, such as non-volatile synaptic weight storage, analog programming, and low-power operation. Such devices are also CMOS compatible and could allow 3D integration [167] for high-density synaptic arrays to implement in-memory computing architectures [267]. The development of spiking neural networks based on these technologies still needs to address several challenges, such as synapses/neuron impedance matching and operation stability validation.

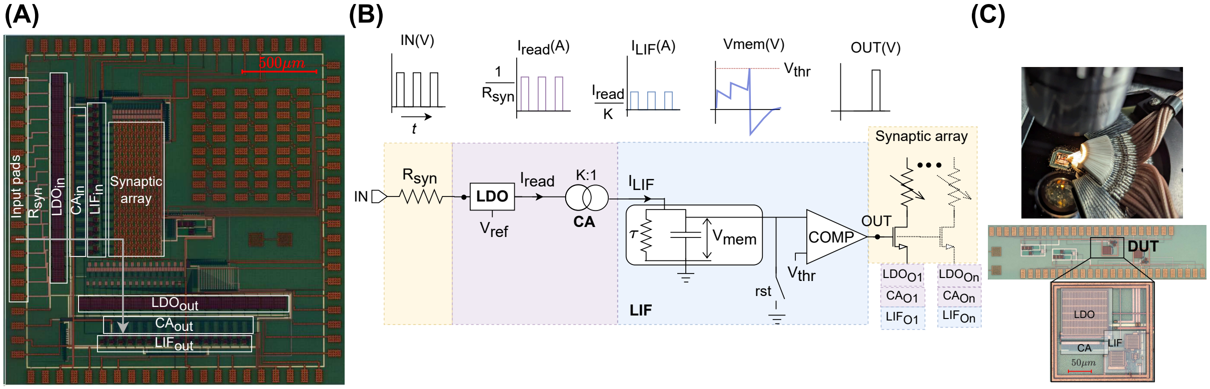


Figure 5.1 Proposed architectures and approach. (A) Optical micrograph of the realized "UNICO" ASIC with an arrow denoting the signal chain from IO pad to neurons of the input layer (LIF_{in}) through a resistive synaptic column and then to output layer (LIF_{out}) through a memristive synaptic array. (B) The signal chain for spike transmission consists of a synaptic resistance, a low dropout regulator (LDO) for regulating the voltage at other synaptic terminals, and a current attenuator for down-scaling the current by a factor of 'K'. The leaky integrate and fire (LIF) neuron integrates the read current. The neuron membrane potential tracks the neuron state and is reset when crossing a threshold voltage level. Consequently, a spike is transmitted to the next layer of neurons through the synaptic array. (C) Test structures with connection pads probed for electrical characterization of implemented circuit blocks.

This study presents an architecture and circuits that enable stable reading of the memristive synapse into an analog ‘spiking neuron circuit. These circuit blocks, including a low-dropout regulator (LDO) and current attenuator, were implemented alongside a LIF neuron in a 130nm CMOS integrated circuit (IC). The measurement results demonstrate the sensitivity of the neuron's activity for a range of synaptic resistance, excitation voltage levels, and pulse widths representative of the memristive synapse operation. The modulation of threshold, leak rate, and refractory period showcase the configurability of neuron dynamics that could be adapted to match different synaptic array sizes, device conductance range, and the time scale of the deployed application. Finally, an architecture to configure generic memristive LIF neurons to an adaptive variant is proposed.

Neurons with slow leak rates and long-time-scale dynamics can be critical for emulating bio-realistic dynamics in real time. Specialized low-pass filtering circuits ([116], [268]) are often used to implement such slow dynamics with small capacitance. We implemented such an integrator circuit with modifications to realize controllable bi-directional leakage for local learning with voltage-dependent synaptic plasticity (VDSP) [194, 243]. In contrast to previous studies [269, 270], the interfacing between the integrator operating at low-current levels and the memristive device was enabled by a current attenuator implemented in the signal chain. VDSP-based synaptic plasticity can further exploit such dynamics to detect spike patterns in longer time windows and enable efficient local learning.

The manuscript is organized as follows. The Methods section details the realized ASIC architecture with key circuit blocks to illustrate the signal chain. The results section presents the electrical characterization of the neuron subject to various input synaptic currents. The membrane threshold level, and refractory period impact on spiking activity is further evaluated. Next, the characterization results in response to spike-based (pulsed) stimuli are presented with an analysis of leak rate modulation. In the end, a connection topology for the dynamic configuration of generic LIF neurons to adaptive LIF neurons is presented, and hardware overheads for different circuit functionalities are discussed.

5.3 Materials and Methods

5.3.1 Implemented Design

The application-specific integrated circuit (ASIC) was realized for hardware implementation of SNN on a monolithic chip through CMOS analog neurons and back end of the line (BEOL) integrated memristive [20] synaptic devices (Figure 5.1(A). The memristive devices are integrated in this design in a 1T1R configuration. In such a configuration, the output terminal of the pre-neuron sends voltage spikes (pulses) to the gate of the 1T1R downstream synaptic array that will enable input current to the post-neurons. The signal chain of the synapse and neuron is illustrated in Figure 5.1(B). For neuron characterization, the presynaptic input is a voltage applied to the resistor's first terminal (IN), emulating the memristive synapse. LDO clamps the second terminal at V_{ref}, and the I_{syn} current weighted by the resistance R_{syn} is scaled by a factor of K by the current attenuator. The LIF neuron membrane voltage performs temporal integration of the current and resets to V_{reset} when V_{mem} is higher than the Vth threshold voltage. Test structures with connection pads implemented on the ASIC were probed for testing (Figure 5.1C). The analog bias voltages and input signals were generated through an FPGA-controlled PCB, and signals from ASIC were measured through an oscilloscope.

The full SNN integrates two layers of neurons in which the output spike is transmitted to the gates of all the downstream synapses of the memristive array. The output of the synaptic array is connected to the second layer of LIF neurons, which present again the successive blocks of LDO and current attenuator.

5.3.2 Circuits

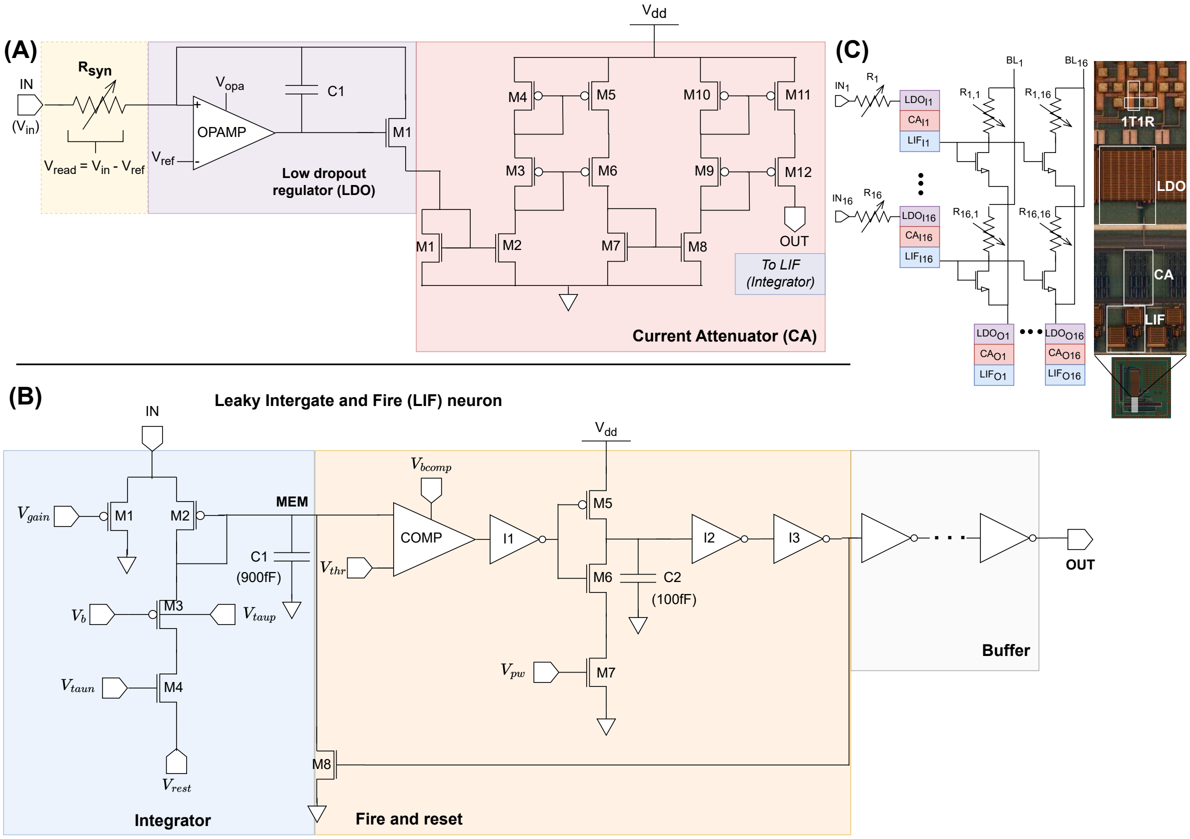


Figure 5.2 Key circuit blocks. (A) This schematic shows the synaptic resistance, Low Dropout Regulator (LDO), and current attenuator arranged to read the synaptic current to the neuron. (B) The LIF neuron is depicted as comprising a leaky integrator, fire and reset block, and buffer. (C) The schematic displays the synaptic array architecture on the left, and on the right, a micrograph of the realized chip is zoomed in around the signal chain from the synaptic array to the neuron.

The circuit schematics of LDO and current mirror are shown in Figure 5.2(A). External Input voltage (V_{in}) is applied to the first terminal of the synaptic resistance (R_{syn}). The other terminal of the synapse is connected to the (+) input terminal of the operational amplifier (OPAMP) in LDO. The other terminal (-) of OPAMP is connected to V_{ref}. Whenever a voltage greater than V_{ref} is applied on IN, the output of OPAMP rises, and transistor M1 is turned on. The current read from R_{syn} is transmitted via the feedback branch, and the voltage at the positive input of OPAMP is pulled to V_{ref}. The output current of LDO is the accumulation of weighted current from all upstream synapses, which is fed to the downstream current attenuator.

The current attenuator is implemented with a cascoded current mirror. This topology is beneficial for a wide swing [271] across the resistance range of upstream memristive synapse. The attenuation factor was tuned by adjusting the W/L ratio of each transistor pair (e.g., M1 and M2, M3 and M4, etc.). Additionally, the sizing was performed to ensure a constant attenuation factor of 500 throughout the read current range. Such topology of LDO and current mirror is also referred to as active current mirror in previous studies [272].

The LIF neuron (Figure 5.2B) performs temporal integration of signals from the current attenuator and consists of a low pass filtering based on difference pair integrator (DPI) circuit [268] for implementing the integration of synaptic current on the capacitor ($C1$). The resting state potential bias voltage controls the level to which the membrane potential leaks (V_{rest}) and was set higher than the reset potential (0V). A transistor (M3) was added to the DPI circuit to realize bi-directional leakage of membrane potential, and the leak rate is controlled by V_{taup} and V_{taun} bias voltages.

The fire and reset block is composed of a comparator for membrane voltage threshold crossing detection with an externally supplied threshold level (V_{thr}). This block is biased by $V_{biascomp}$. The neuron's pulse width and refractory period are configured by modulating the discharging rate of capacitance $C2$ through bias voltage V_{pw}. A reset transistor (M8) discharges the membrane capacitor to the ground on a spike event, and the generated spike is transmitted to the downstream synapse through a buffer. As the reset transistor is activated throughout the spike generation period, V_{pw} also controls the neuron's refractory period.

The architecture of the memristive synaptic array is shown in Figure 5.2(C)(left). The neurons in the input layer (LIF_{I1} to LIF_{I16}) transmit a generated spike to the next layer by applying the spike output to the transistor's gate of the 1T1R synaptic cell. The spike amplitude is set to V_{dd} to minimize the voltage drop access transistor during spike transmission. All the synaptic cells in a column are connected to the corresponding output neuron through LDO and the current attenuator. The circuits were implemented in 130nm CMOS technology, and a picture from a microscope is shown to visualize the layout of the signal chain in Figure 5.2(C)(right). The typical values and short descriptions of bias voltages are summarized in Table 5.1.

Table 5.1 Bias voltages used in circuits

Name	Typical (V)	Purpose
V_{dd}	3.3	Power supply
V_{ref}	2.4	LDO reference
V_{opa}	2.4	OPAMP bias (LDO)
V_{gain}	2.1	Gain modulation
V_{taun}	1.2	Leak rate (Down)
V_{taup}	1.2	Leak rate (UP)
V_{rest}	0.6	Resting potential
V_{thr}	1.2	Neuron threshold level
V_{bcomp}	2.4	Bias for COMP (LIF)
V_{pw}	1	Pulse width modulation

5.4 Results and Discussion

5.4.1 Synaptic resistance reading

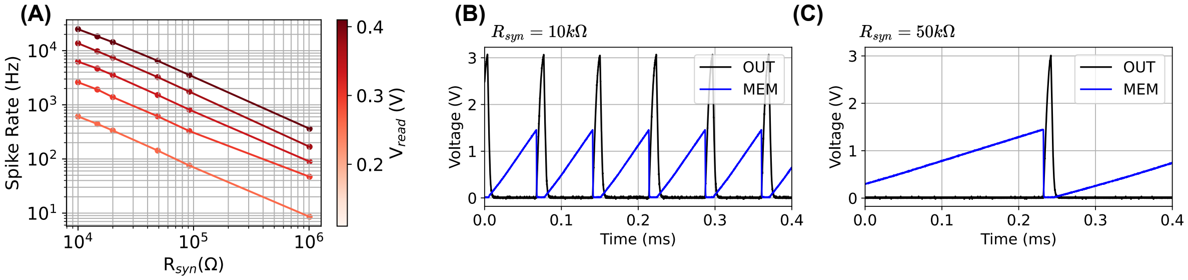


Figure 5.3 Characterization of neuron's sensitivity to synaptic resistance. (A) Measured output spike rate of neuron with respect to R_{syn} for different V_{read} levels between 100mV and 400mV. The respective V_{read} was applied for 1s for each experiment, and the neuron's response was recorded. Comparison of neuron's membrane voltage and output response is shown for R_{syn} of 10kΩ (B) and 50kΩ (C) with read voltage (V_{read}) of 250mV.

For characterizing the synaptic resistance reading, an external resistor (R_{ext}) was connected in series with the on-chip resistor of 10kΩ (R_{IC}) to emulate a synaptic resistance, resulting in read current $I_{syn} = (V_{in} - V_{ref})/(R_{ext} + R_{IC})$, where V_{ref}, the reference voltage supplied to LDO, was set to 1V, and $R_{ext} + R_{IC}$ represents the net synaptic resistance (R_{syn}). The relationship between the measured output spike rate and input resistance (R_{syn}) is shown in Figure 5.3(A). The input resistance was varied from 10kΩ to 1MΩ, and experiments were repeated for five different read voltages ranging between 100mV and 400mV. Such resistance range matches the one observed in our memristive devices.

The applied voltage is also compatible with the read voltage range from the memristive synapses (i.e., read without weight disturbance).

To illustrate the temporal response of the neuron, the measured membrane voltage and output of the neuron are compared in Figure 5.3(B-C) for synaptic resistance of 10kΩ and 50kΩ and read voltage (V_{read}) of 250mV.

Sensitivity to the large range of synaptic resistance makes the neuron suitable for various memristive technologies. In the above experiments, the output spike rate was observed to vary between 8Hz and 25kHz. The ability to read and differentiate between low resistances (with high firing rates) can enable reading from several presynaptic resistances in parallel.

5.4.2 Neuron transfer characteristics

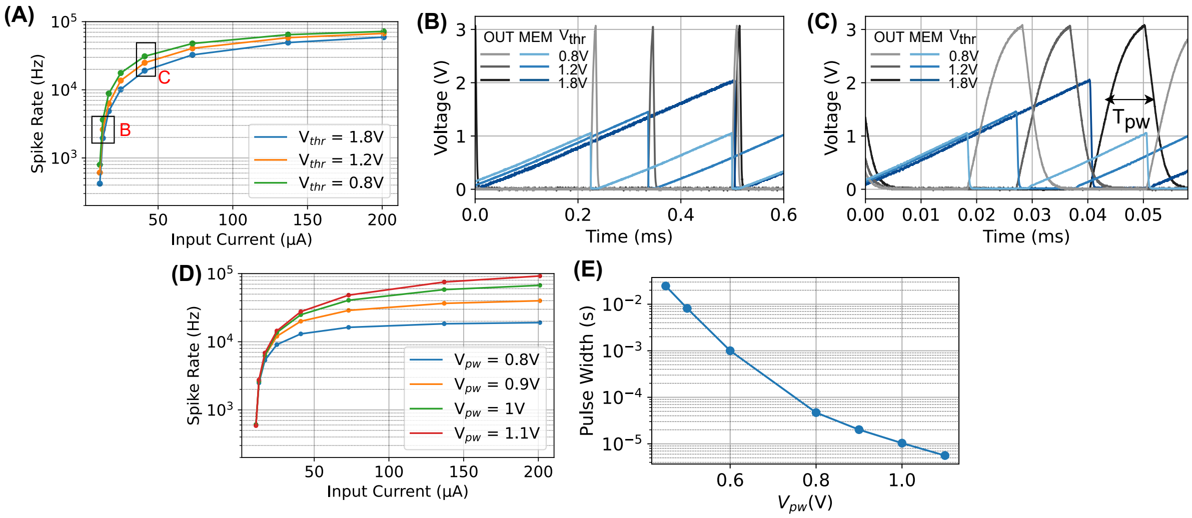


Figure 5.4 Characterization of neuron's transfer function with DC excitation. (A) Measured spike rate of neuron with respect to input current (I_{syn}) and spiking threshold bias level (V_{thr}). Comparison of the measured temporal response of membrane voltage and output for different threshold levels obtained with I_{syn} of 10μA (B) and 40μA (C). (D) Neuron's spike rate with respect to input current for different V_{pw} bias levels. (E) Pulse width (T_{pw}) of the generated output spike plotted with respect to V_{pw} bias voltage.

To characterize the excitation of neurons for a range of synaptic currents and bias voltages, the excitation current was varied from 10μA to 200μA. The read voltage was applied through an on-chip resistor of 10kΩ for a long duration (100ms). The rate of output spikes generated by the neuron is plotted with respect to the injected current, and the neuron's threshold voltage level (V_{thr}) is shown in Figure 5.4(A). For the threshold level of 1.8V, the spike rate was 419Hz and 59kHz for input current of 10μA and 200μA, respectively.

Similarly, for a threshold level of 0.8V, the spike rate varied between 800Hz and 68kHz. The measured membrane voltage and output spike are compared for different threshold voltages and input currents of 10μA (B) and 40μA (C).

The threshold impacts the range of firing rate (activity) that could be observed with respect to input current. A lower threshold could increase the sensitivity to low current (high synaptic resistance), but the spike rate saturate early. Conversely, a high threshold led to lower firing rates for small excitation currents, and a variation in spike rate could be observed even for high input currents. The maximum spike rate of the neuron is limited by the output pulse width or refractory period controlled by V_{pw} bias level. The neuron's response(rate) to different (V_{pw}) is shown in Figure 5.4(D). The maximum firing rate was 20kHz and 92kHz for (V_{pw}) of 0.8V and 1.1V, respectively. The V_{pw} bias voltage modulates the output spike's pulse width and the neuron's refractory period by controlling the discharging rate of the $C2$ capacitor in Figure 5.2(C). The pulse width was measured as the time difference between the crossing of OUT and $V_{dd}/2$ and is plotted for V_{pw} in Figure 5.4(E). The resultant pulse width of the output spike varied between 20ms and 10μs for V_{pw} of 0.45V and 1V, respectively.

5.4.3 Temporal dynamics

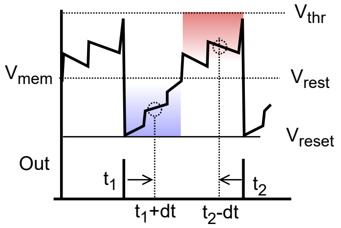


Figure 5.5 Illustrative plot to show the behavior of a neuron with bi-directional leakage. The downward and upward leakage occurs when membrane voltage is greater or less than resting state potential (V_{rest}), respectively. A leak in the upward direction occurs when the neuron is reset after the spike event (t_1) and enables estimation of time elapsed (in blue). Similarly, the occurrence of a spike event in the near future (t_2) can be predicted through a high value of membrane voltage (in red).

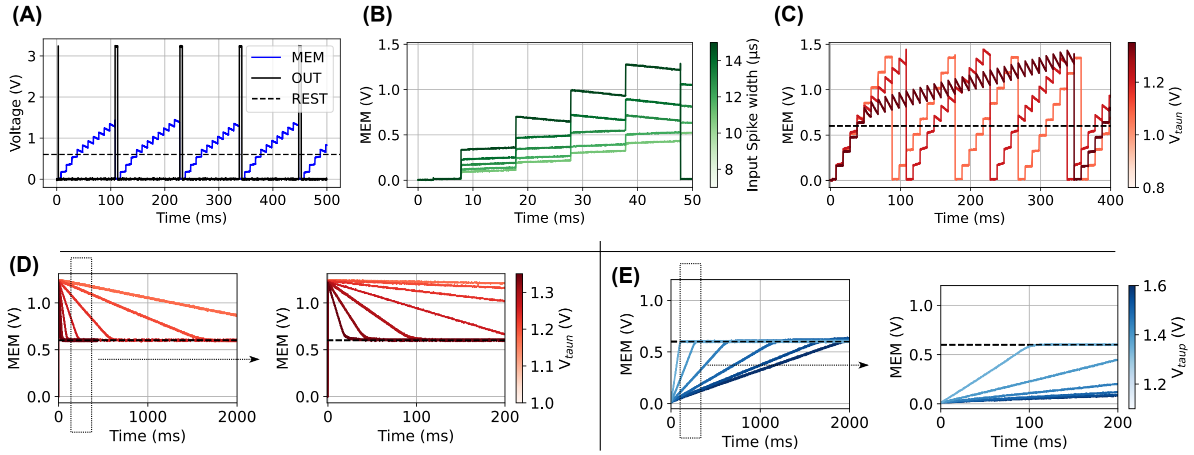


Figure 5.6 Characterization of temporal dynamics. (A) Input pulses of width 10μs were applied at 100Hz, with the neuron's membrane potential and output shown across time. (B) The input pulse width was varied between experiments (7μs and 15μs), comparing the charging events of membrane voltage. (C) The experiment varied V_{taun} across experiments to compare the neuron's membrane voltage evolution across time. In (A-C), the I_{syn} for each pulse was of magnitude 40μA. (D) A single input pulse was applied to charge the membrane voltage close to the neuron's threshold, and the downward leak rate is controlled by V_{taun} bias level and varied across experiments. (E) The magnitude of the applied spike was increased to induce neuron firing. The leakage in the upward direction (from V_{reset} to V_{rest}) is compared by varying V_{taup} bias level between experiments.

The neuron was characterized by applying short pulses at frequent intervals to access charging with respect to input pulse width and dynamics in the absence of excitation. Spikes of magnitude 40μA and width 10μs were fed to the neuron by applying voltage pulses (V_{read}=400mV) across the 10kΩ on-chip resistor at the rate of 100Hz. The measured response of the neuron membrane voltage and output spikes is plotted in Figure 5.6(A) and results in a neuron spike rate of 8Hz. The above experiment was repeated for different pulse widths of input spikes between 5μs to 15μs. The measured response of the neuron's membrane voltage across time is plotted in Figure 5.6(B) to compare the magnitude of the increment in membrane voltage for every charging event. A pulse of 7μs led to a small increment in membrane voltage at every spike event, resulting in a firing rate of 2Hz. Whereas a 15μs input pulse could fully charge the neuron with 5 input spikes, resulting in an output spike rate of 18Hz. Since the pulse width influences the number of spikes, the postsynaptic neuron integrates before firing the pulse width of the presynaptic neuron spikes, which can be adapted for the crossbar size. For example, suppose a neuron in the postsynaptic layer is expected to integrate signals from four times input neurons in parallel. In that case, the pulse width of input neurons should be accordingly reduced by increasing the V_{pw} bias level to maintain similar temporal dynamics in the output layer. A shorter pulse also leads to a granular increment in membrane voltage, allowing down-sizing of the membrane capacitance. In the absence of input current, the membrane voltage leaks to a resting state potential (V_{rest}). Since the reset level was set to 0V, the neuron exhibits leakage in upward direction after the refractory phase to V_{rest} (set to 600mV). This bi-directional mechanism was implemented to differentiate between idle neurons and those in the refractory period, as shown in Figure 5.5, and is beneficial for implementing local synaptic learning.

An important aspect of memristive synapses / CMOS neuron co-integration for SNN is to allow local learning based on the activity of the adjacent neurons [208]. In local learning rules such as Spike Timing Dependent Plasticity (STDP), the spike time difference between the pre and post-neurons is converted into a programming voltage by overlapping of slow decaying voltage pulses [273]; however, this approach can consume a significant fraction of area-energy budget [209] and implementing pulses with complex shapes can become challenging. More recently, Voltage-dependent synaptic plasticity (VDSP) [194, 243] was proposed to implement learning efficiently in hardware without requiring pulse shaping circuits. In this approach, the recent pre neuron's activity can be estimated through the neuron membrane potential (i.e., low membrane potential being associated with a recent firing event and high membrane potential associated with an imminent spiking event). Mapping of this concept to memristive devices programming requires the implementation of more complex membrane dynamics. In this study, we present a neuron circuit with bi-directional leakage to enable learning of memristive weights through VDSP.

For characterizing the tunability of leak rate in the downward direction, spikes of magnitude 40μA and width 10μs were fed to the neuron at 100Hz, and V_{taun} was varied between experiments. The measured membrane voltage response is compared in Figure 5.6(C). For V_{taun} of 1.2V (typical), the output spike rate was 8Hz, which could be increased to 12Hz by lowering the leak rate. Conversely, increasing the leak rate could decrease the output spike rate to 2Hz.

For evaluation of V_{taun}, the neuron was excited by a single input pulse of higher magnitude to charge the membrane voltage to just below the spiking threshold, and the response was recorded for 2s. The membrane voltage is compared for values of V_{taun} between 1V and 1.35V in Figure 5.6(D) for time intervals of 2s and 200ms. The neuron could fully leak from the threshold level to resting state potential in 8 ms for a high leak rate. Lowering the leak rate could modulate this duration to more than 2s.

The leak rate impacts the neuron's memory window and can be adjusted to match the dynamics of the input signals. For instance, in scenarios where the neuron receives sparse spikes at a low frequency, reducing the leak rate can help preserve the neuron's memory over a large temporal window. Conversely, increasing the leak rate when processing high-frequency input signals can regulate the neuron's output firing rate.

Next, we characterized the upward modulation of the leak rate. The neuron was stimulated with a single high-magnitude spike to trigger an output spike event, and its membrane potential subsequently leaked from V_{reset} to V_{rest}, as illustrated in Figure 5.6(E). The biasing voltage (V_{taup}) was adjusted between 1.1V and 1.6V. We monitored the membrane voltage over 2-second periods to assess the upward leak rate. The ability to tune the upward leak rate and the resting state voltage is beneficial for adjusting the learning dynamics by modifying the probabilities of potentiation and depression.

5.4.4 Configurability to adaptive neuron

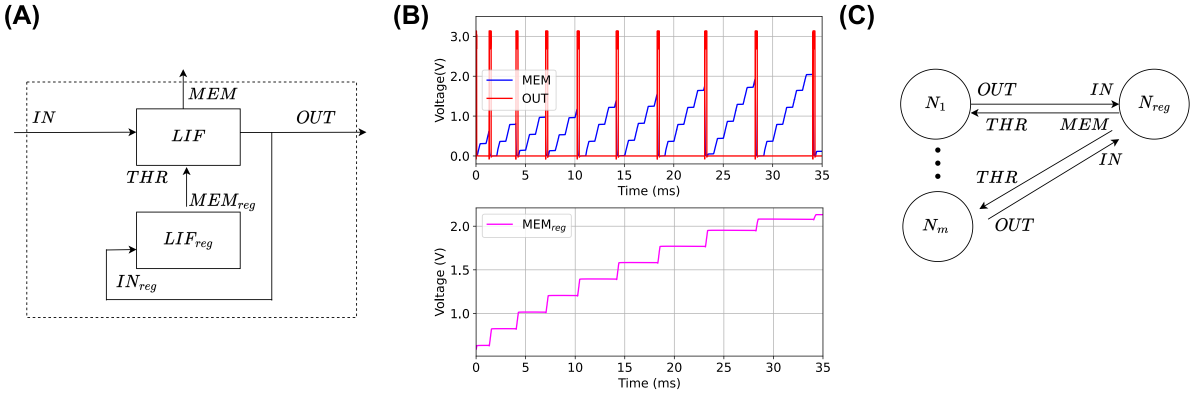


Figure 5.7 (A) Connection scheme to realize an adaptive LIF neuron with a pair of LIF neurons. The regulator neuron integrates the output signals of the primary neuron, and its membrane voltage (MEM_{reg}) is used as the threshold of the primary neuron. (B) The primary neuron was stimulated at a 1kHz rate and connected to a regulator neuron. The membrane potentials of both neurons and the output of the primary neuron are plotted using results from a SPICE circuit simulation. (C) Single regulator neuron (N_{reg}) shared by a sub-population of generic LIF neurons ($N_1, ..., N_m$).

The spiking neurons' information processing and memory capacity can be further enhanced through threshold adaptation [274, 275, 276]. The designed neuron block can be used as a regulator neuron to monitor the spiking activity of a primary neuron. As depicted in Figure 5.7(A), the output of the primary neuron (OUT) is connected to the input terminal of the regulator neuron (IN_{reg}). This connection tracks the neuron activity through the membrane potential of the regulator neuron (MEM_{reg}), which in turn sets the threshold (THR) for the primary neuron.

The designed circuit blocks, connected as described in Figure 5.7, were simulated in SPICE. The primary neuron, excited by 1 kHz pulses, and the corresponding changes in membrane potential for both the primary and regulator neurons are depicted in Figure 5.7(B). During this simulation, the threshold voltage of the primary neuron (MEM_{reg}) increased from 600mV to 2V, resulting in the inter-spike interval of the output spikes increasing from 2ms to 5ms. A chain consisting of a 10kΩ synaptic resistance, followed by an LDO and a current attenuator, was used to transmit output voltage pulses from the primary to the regulator neuron. Alternatively, this synaptic resistance can be replaced with a memristor, which can be programmed to introduce an additional level of plasticity at the neuron level.

Additionally, a sub population of neurons (N_1, N_2, ..., N_m) could share a common regulator neuron (N_{reg}), as illustrated in Figure 5.7(C). Having a pool of LIF with the same regulator neuron can improve the system's scalability by sharing resources. This architecture can configure a fully analog LIF neuron chip to an arbitrary population of LIF and ALIF neurons in runtime. Such a heterogeneous population has been shown to enhance the learning capabilities of SNNs [277, 57, 278]. Furthermore, monitoring neuron activity over time through traces can also enhance local learning [115], providing a dynamic feedback mechanism to improve performance and adaptability.

5.4.5 Overhead of different functionalities

The power consumption of the implemented LDO and current attenuator was estimated through SPICE simulations of implemented circuits. The static power consumption, measured when no signal was present at the input, was 10µW for the LDO and 10pW for the current attenuator. During the operation of reading from the resistive synapse, the dynamic power consumption was estimated to be 18µW for the LDO and 20 µW for the current attenuator. The static power dissipation of LDO can be accounted for leakage due to the high biasing current used to obtain fast response time and high output current levels (up to 200µA). This quick activation is necessary when using short-reading pulses (neuron output spikes). As discussed in the earlier section, short pulses lead to gradual charging of (post-synaptic) neurons, thus helpful for scaling up crossbar size or the number of pre-synaptic neurons. Moreover, power dissipation through the memristive device during reading can be lower with short-read pulses. As these circuits are shared by synapses in a row or column, energy utilization can be balanced.

The LIF neuron's static energy and energy per spike were estimated to be 5µW and 200 pJ/spike. The high static power can be attributed to leakage due to biasing the comparator. With a lower bias current, the static power can be reduced to 17nW, but the energy per spike increases to 7nJ. This trade-off occurs because a slow comparator can lead to high leakage during the firing phase. Feedback mechanisms [279] can potentially reduce the energy per spike of the neuron with a low comparator bias current. Overheads associated with reading memristive states are also present in hardware Artificial Neural Network (ANN) implementations, where trans-impedance amplifiers and analog-to-digital converters (ADCs) sense the current from the synaptic column for subsequent computational processing and account for a major fraction of energy consumption. However, analog neurons are more suitable for interacting with analog memristive synapses. SNNs benefit from events' sparse activity, and gating the power supply [280] of synapse reading blocks around spike events can reduce static power dissipation through leakage.

5.5 Conclusion

We propose a versatile CMOS circuit to integrate memristive synapses into the signal chain of analog neuromorphic circuits. This integration pathway includes a Low-Dropout Regulator (LDO), a current attenuator, and an analog Leaky Integrate-and-Fire (LIF) neuron. The circuit blocks were implemented on a 130nm CMOS ASIC. The chip features multiple instances of these blocks alongside a synaptic array composed of 1T1R cells, facilitating the integration of memristive synaptic devices. The circuit building blocks of the signal chain in the implemented ASIC were characterized. We demonstrated the neuron's sensitivity to the synaptic state by testing resistances ranging from 10kΩ to 1MΩ. As a result, the neuron's firing rate was observed to vary between 8 Hz and 25 kHz. Additionally, the neuron's activity was characterized across a range of applied read voltages. By increasing the spiking threshold and pulse width of the neuron, the neuron can be fine-tuned to lower and higher firing rates, respectively. Excitation with a sparse train of pulses was used to measure the neuron's temporal response. The neuron's charging was controllable by changing the input pulse width. Consequently, the presynaptic neuron spike's width can be adjusted per the postsynaptic neuron's fan-in characteristics. In the absence of an input signal, the neuron retains its memory through leakage, with an adjustable leak time constant. The bi-directional leakage enables the estimation of neurons' recent activity and should benefit online learning through local synaptic learning.

We experimentally validated the modulation of the leak rate across short and long-term intervals, which can be helpful in processing signals with various temporal dynamics. The topology of pair neurons for configuration with adaptive neurons could further enhance the capabilities of the implemented neuron. Future work will detail the architecture and characterization of the integrated CMOS-RRAM ASIC with in-situ learning.

Author Contributions

N.G. designed the circuits with contributions from E.M. and M.C.F., under the direction of J.M.P.; D.F. performed post-processing of the chip to expose the pads for probing. N.G. performed the on-chip experimental measurements, with contributions from P.D.; N.G. wrote the initial version of the manuscript. J.M.P., F.A., D.D., Y.B., D.Q., and M.B. directed the project and edited the manuscript. All authors discussed the results and reviewed the manuscript.

Acknowledgment

We acknowledge the support from CMC Microsystems for fabrication services using the 130-nanometre CMOS technology from TSMC. The authors thank Bastien Imbert for assisting in the characterization setup.

CHAPTER 6

Neural building block

"The brain is imagination and that was exciting to me. I wanted to build a chip that could imagine something" – Misha Mahowald

TABLE OF CONTENTS

6.1 Résumé

L'intégration de l'apprentissage en ligne avec une synapse mémristive peut aider à atténuer la variabilité et le bruit, permettant ainsi au système de s'adapter en fonction de ses actions. Les circuits analogiques CMOS sont essentiels pour des calculs à faible consommation d'énergie et à haut débit en tirant parti des similitudes entre les composants. La mise en œuvre de l'apprentissage en ligne dans un tel système nécessite une architecture dédiée capable de passer sans interruption entre les phases d'inférence et de mise à jour des poids. Ce chapitre présente la conception d'un bloc de construction neuronal CMOS-RRAM (NBB), qui intègre des neurones analogiques, conçus avec des circuits CMOS, et des réseaux mémristifs Back-End-of-Line (BEOL) contrôlés par des circuits numériques. L'architecture proposée garantit la compatibilité entre les circuits analogiques, discutés précédemment, et la communication numérique asynchrone, jetant ainsi les bases de systèmes neuromorphiques évolutifs. Dans des travaux futurs, la conception proposée sera utilisée pour démontrer l'apprentissage en ligne avec le réseau synaptique mémristif intégré en BEOL, montrant la capacité du système à s'adapter continuellement en réponse à de nouvelles données.

6.2 Abstract

Incorporating online learning with memristive synapses can mitigate variability and noise, enabling systems to adapt dynamically based on their interactions. Analog CMOS circuits play a vital role in achieving low-energy, high-throughput computation by capitalizing on the inherent similarities of their components. Implementing online learning in such systems requires a specialized architecture that can seamlessly alternate between inference and weight update phases. This chapter presents the design of a CMOS-RRAM Neural Building Block (NBB), which integrates analog neurons, realized through CMOS circuits, with Back-End-of-Line (BEOL) memristive arrays controlled by digital circuits. The proposed architecture ensures compatibility between analog circuits and asynchronous digital communication, forming the foundation for scalable neuromorphic systems. Future work will demonstrate online learning using the BEOL-integrated memristive synaptic array, showcasing the system's continuous adaptability to new data inputs.

6.3 Introduction

This chapter introduces the digital logic required for scanning out spikes generated by the LIF neuron, as detailed in chapter 5. The crossbar reading circuit, which includes the LDO, CA, and LIF, is integrated into the chip for each row and column of the spiking neural network (SNN) with a 16x16 configuration. This results in 32 neurons (16 rows

+ 16 columns) connected by 256 (16x16) 1T1R memory cells. Given the large number of memory cells, assigning a dedicated I/O pad to each cell is impractical. To address this, addressing logic was implemented to route selected cells to the limited I/O pads on the chip. This addressing logic is user-configurable through a hardware-software design running on the Zynq System on a chip (SoC) platform, referred to as *Lotus* throughout this chapter.

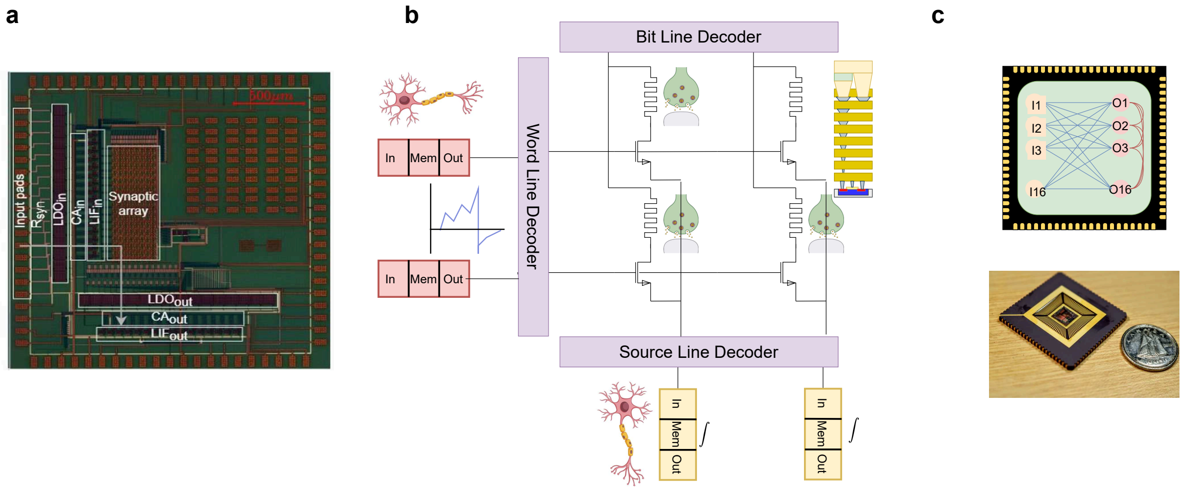


Figure 6.1 NBB Overview **a** Micrograph of Neural Building Block (NBB) with annotated signal path from input pads to output neuron bank. **b** Architecture of neuron and synaptic array illustrated through a 2x2 example. The memristive synapse is implemented in the Back end of the line (BEOL). **c** The full chip features 84 I/O connections and implements a 16x16 fully connected network (top). Packaged sample of a single NBB mounted to a carrier (bottom).

The NBB comprises 32 neurons and 256 synapses. The neurons emulate a biological spiking neuron model, specifically the LIF neuron, and are implemented using CMOS transistors operating in the sub-threshold region through analog Differential Pair Integrator (DPI) circuits [268]. These neurons process input signals by integrating them over time, generating spikes when the membrane potential crosses a certain threshold. The synaptic weights are realized with 1T1R BEOL integrated non-volatile memory devices, which offer multiple stable states, enabling flexible control over the strength of connections between neurons.

The layout and architecture of the NBB are depicted in Figure 6.1a, which outlines the chip's signal path from input pads to output neuron banks. The architecture is further illustrated in Figure 6.1b with a 2x2 neuron and synapse array example, where memristive synapses are implemented in the Back end of the line (BEOL) layer. This configuration allows the NBB to dynamically adjust synaptic weights via VDSP, an analog conductance programming technique, facilitating real-time learning. Figure 6.1c highlights the full chip's 84 I/O connections, demonstrating its implementation of a 16x16 fully connected network (top), with a packaged sample of a single NBB mounted to a carrier (bottom).

To enable real-time configuration and spike data scanning, the *Lotus* PCB integrates the Zynq SoC, which contains both Analog to Digital Converter (ADC) and Digital to Analog Converter (DAC) components. These components provide bias voltages and excitation signals to the neurons. The Zynq SoC combines an FPGA for digital circuit execution and an ARM core for managing real-time configurations, including triggering the FPGA circuits. The system is controlled via a custom Graphical user interface (GUI), which allows the user to configure various operational modes, such as (i) serial electro-forming, reading, and writing of individual synaptic devices, (ii) implementing Winner-Take-All (WTA) in neurons, and (iii) scanning membrane potentials and output spikes. These operations are efficiently managed through shift registers to optimize data handling.

Characterization of the neurons was conducted to assess the impact of process-voltage-temperature (PVT) variations, as the DPI circuit operates in the sub-threshold region, making it sensitive to even small variations. To mitigate these effects, on-chip amplification of the membrane potential was implemented to generate accurate programming voltages for VDSP-based learning. A mixed-signal circuit was also designed to compute the polarity and magnitude of the programming voltage, ensuring proper functionality of the memristive devices during weight updates. This precise control over synaptic plasticity is crucial for enabling real-time learning in the neuromorphic system.

6.4 Materials and Methods

6.4.1 Data path

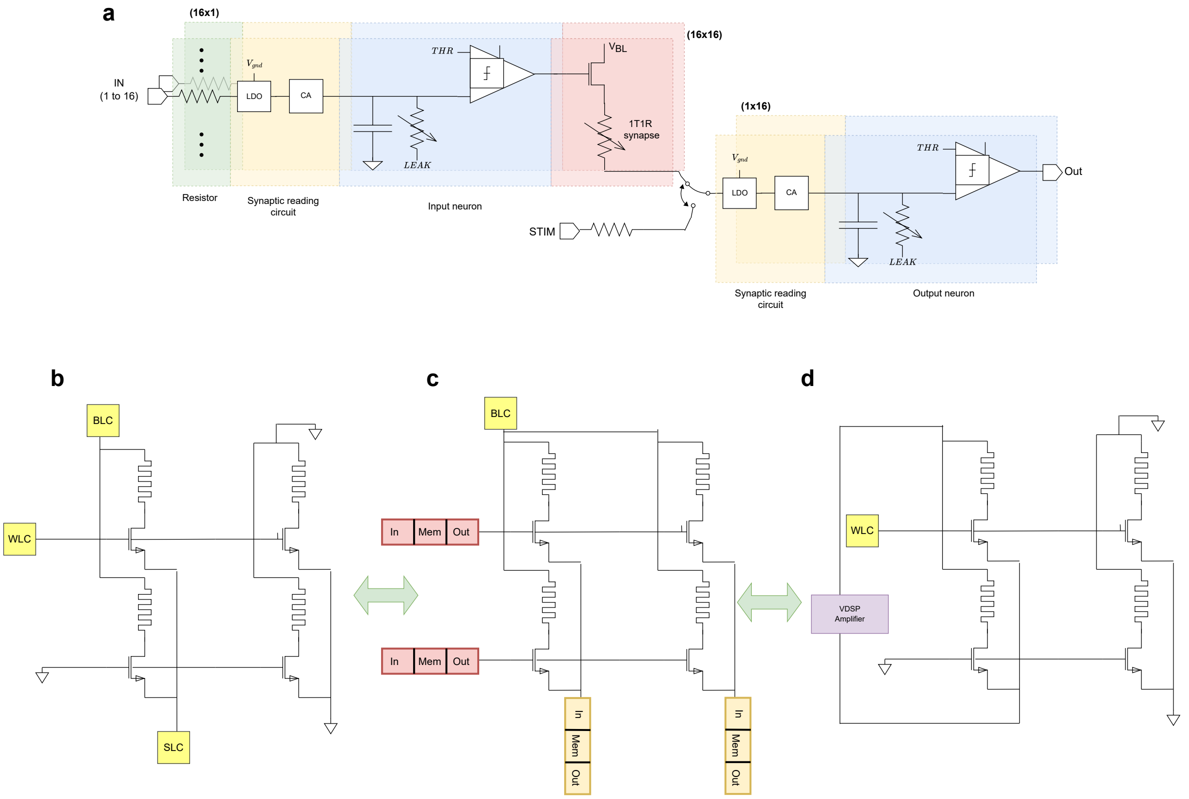


Figure 6.2 Data path of NBB and modes of operation. **a** The signal path starts from the input pads (IN 1 to 16), followed by the synaptic reading circuit, which includes the LDO and CA, and the input LIF neurons. The signal then passes through the 16x16 1T1R synaptic array, leading to a second layer of synaptic reading circuits and neurons. **b** In characterization mode, the IO pads (BLC/WLC/SLC) address the selected crossbar cell for forming, reading, and writing operations. **c** In inference mode, the output spike from the presynaptic LIF neuron is fed to the gate (WL) of the synaptic cell. The cell's source is connected to the resistance reading and LIF blocks in the postsynaptic neuron bank. **d** In learning mode, a single synaptic cell is programmed by connecting the BL and SL to the on-chip amplifier, while the corresponding row is connected to the WLC IO pad to provide compliance current during the programming operation (externally).

The signal chain or data path (Figure 6.2a) follows a similar architecture to the one presented in the previous chapter, extending through the first layer of LIF neurons. The spikes from this layer are transmitted to the second layer, where they are weighted by a 16x16 1T1R synaptic array. Additionally, an external stimulation mechanism has been integrated for output neurons to support online learning.

The crossbar operates in three distinct modes: characterization, inference, and learning. In the `characterization` phase (Figure 6.2b), a single 1T1R cell is selected and accessed via the bit line, word line, and source line characterization pads (`BLC/WLC/SLC`). This configuration is crucial for memristor forming, programming, and reading the state of individual cells.

In the `inference` mode (Figure 6.2c), the synapses and neurons are interconnected to compute the network's decision in response to inputs provided to the presynaptic neuron. The output spikes from the presynaptic neurons are applied to the gate of the respective 1T1R cells, while the source line is connected to the postsynaptic neuron input terminal.

Lastly, during the `learning` phase (Figure 6.2d), the on-chip amplifier supplies the programming voltage to the bit line (`BL`) and source line (`SL`). The compliance current for programming is set externally by applying voltage to the word line characterization pad (`WLC`).

The architecture is designed with two key motivations:

1. **Neuron-gate connection to the 1T1R cell**: In this architecture, the output of the neuron is connected to the gate of the 1T1R cell. When a neuron spike event occurs, it activates all memory cells in that row. The neuron only draws a small amount of current from each connected memory cell, while most of the read current is pulled from the bit line. This configuration eases the load on the neuron's output block, enhancing the system's scalability.
2. **Efficiency in charging**: The bit line (`BL`) remains continuously charged, and only the word line (`WL`) is charged during spike transmission. This provides both energy and latency benefits, improving overall system efficiency.

6.4.2 Crossbar block

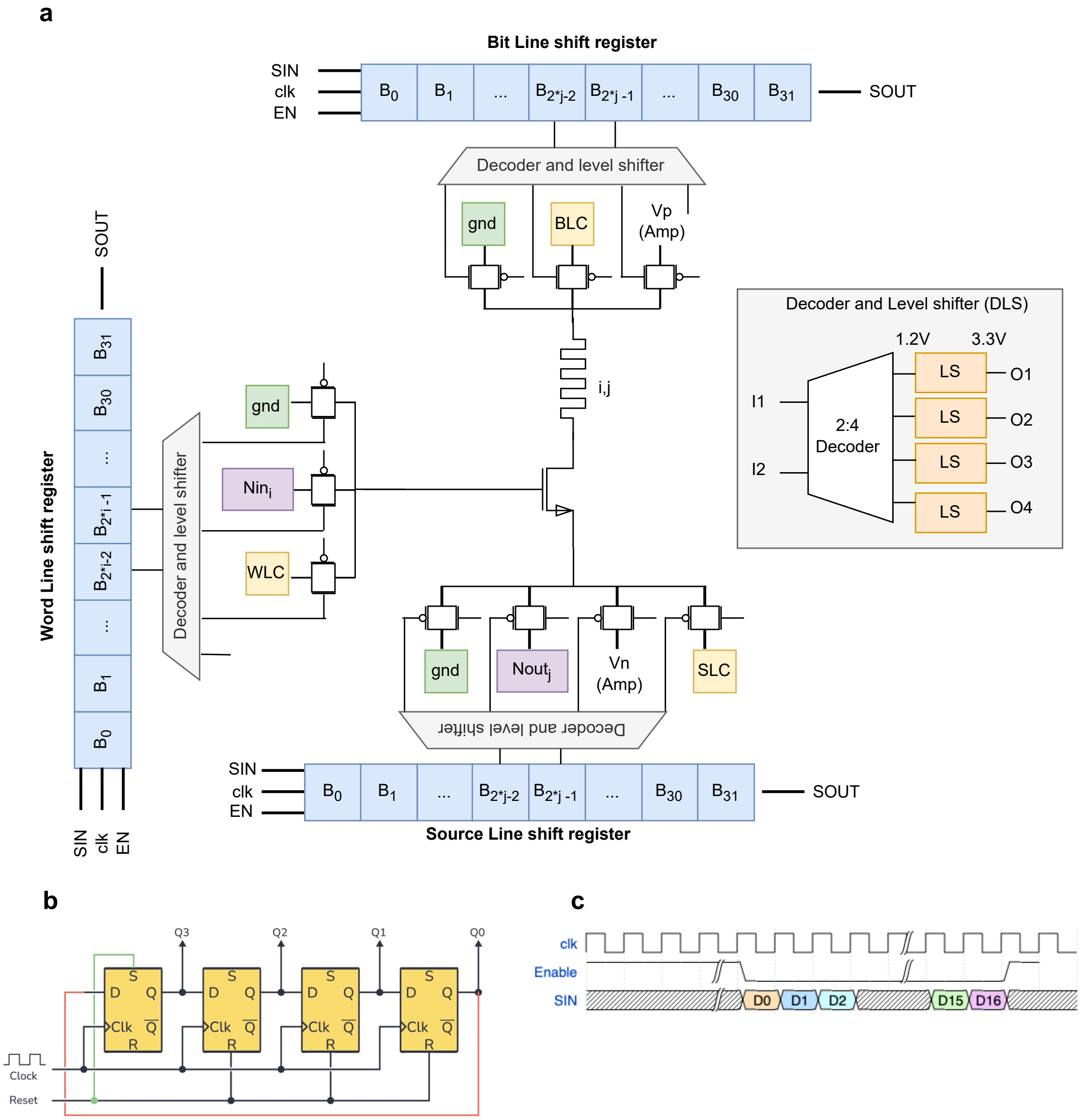


Figure 6.3 Crossbar addressing circuits and logic. **a** A single cell of the crossbar block with controlling logic, which includes a 32-bit shift register for addressing the BL, SL, and WL. Each row or column is controlled by two bits from the shift register, which are decoded into four bits by the decoder and level shifter (DLS) block. **b** D-Q flip-flop-based serial-in-parallel-out shift register. **c** Configuration of on-chip shift registers.

Figure 6.3a illustrates the circuit for controlling a single cell in the crossbar block. The design incorporates three 32-bit shift registers to address the bit, word, and source line.

Each row or column is controlled by two dedicated configuration bits, which are decoded through a 2:4 decoder to generate four output bits, with only one bit enabled at any given time. A level shifter also converts the 1.2V digital signals to 3.3V. These decoded bits control the transmission gates, which then switch the corresponding row or column to connect in one of three modes: inference, characterization, or learning. The array of transmission gates behave like an analog multiplexer for selecting and routing specific memory cells within the matrix. These gates are controlled by a digital shift register, as illustrated in Figure 6.3b.

The serial-in-parallel-out shift registers allow for greater control over analog circuits despite having a limited number of external IO pads. In the NBB, a total of five 32-bit configuration bits and two 16-bit bits are used to establish the addresses for the characterization IO pads (WLC/BLC/SLC), neuron input pads (`IN1` to `IN16`), and to enable switching between different configurations. These registers are constructed using DQ flip-flops, which are sequential logic devices that store a single bit of data. They operate by capturing the value of the input (D) at the rising edge of the clock signal and holding it until the next clock cycle. The circuit diagram illustrating four of these flip-flops connected to form a shift register is shown in Figure 6.3b. The value stored by each flip-flop (Q1 to Q4) can be used to drive switches Transmission Gate (TGATE) used to configure the chip. A level shifter was implemented on the chip to convert the 1.2V external programming signals to the 3.3V level required to control the transmission gates. This ensures proper signal compatibility between digital circuits or programming inputs and on-chip analog circuitry operating at a higher voltage.

The shift registers are programmed externally through serial input (`SIN`) and enable (`EN`) signals, in conjunction with shared `clock` and `reset` signals to ensure synchronized control (see Figure 6.3c). This setup allows for a precise and efficient configuration of the crossbar by serially transmitting data to control the selection of memory cells. The combination of analog and digital control mechanisms offers a flexible and scalable solution, ensuring seamless integration and adaptability within the neuromorphic system.

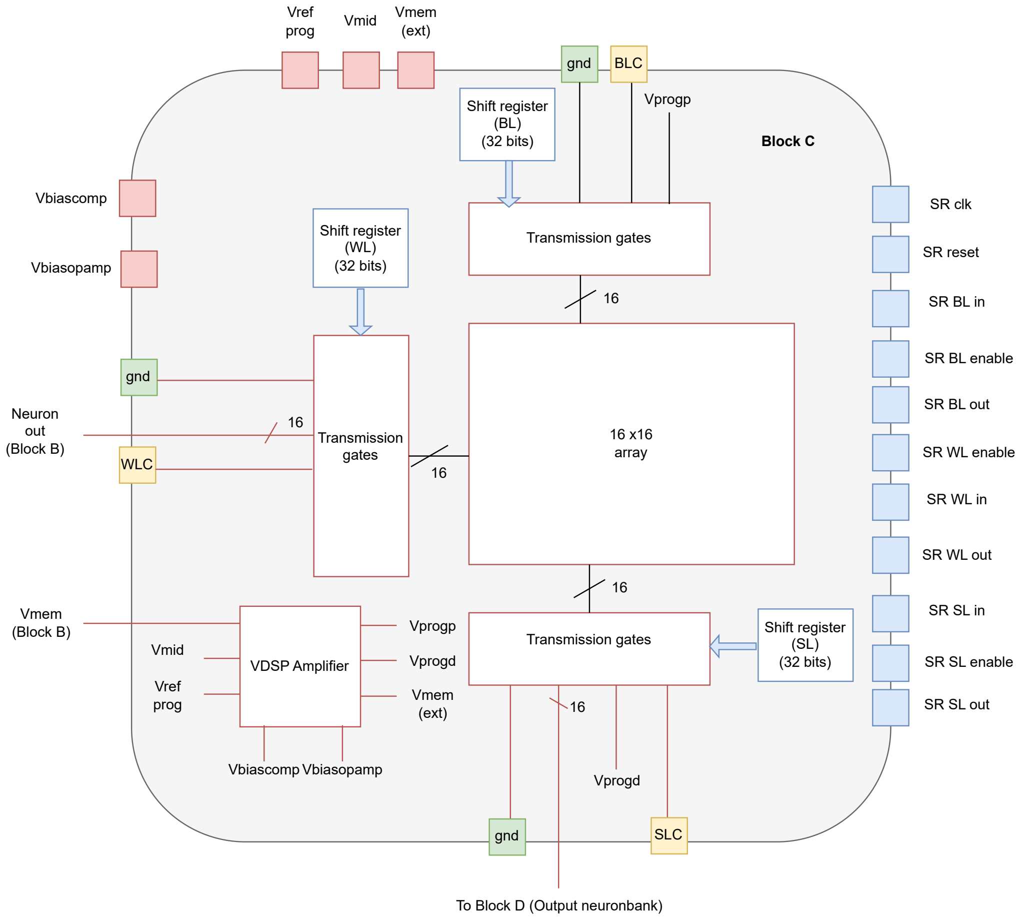


Figure 6.4 Top Schematic Diagram of Block C (Crossbar Block). Around the 16x16 1T1R array, banks of transmission gates are serially configured by shift registers. This configuration is controlled via digital I/O pads located on the right side of the diagram (SR WL/BL/SL). Each row and column can be connected to the neuron bank, characterization pad, ground plane, or the VDSP amplifier, depending on the configuration. The VDSP amplifier receives bias voltages supplied through external analog I/O pads, which are highlighted in red.

The block diagram of the crossbar block, which comprises the 1T1R array and its control/communication logic, is shown in Figure 6.4. Each shift register has dedicated pads for the `IN`, `EN`, and `Out` signals, while the `clk` and `reset` signals are shared across all registers to ensure synchronized operation throughout the system. The `BLC`, `WLC`, and `SLC` pads serve as the characterization I/O for the NBB, while external bias voltages, such as `Vbiascomp`, `Vbiasopamp`, `Vrefprog`, and `Vmid`, are supplied to the VDSP amplifier block through analog I/O pads. The output from the crossbar block, consisting of 16 source lines, is then routed to the output neuron bank (Block D).

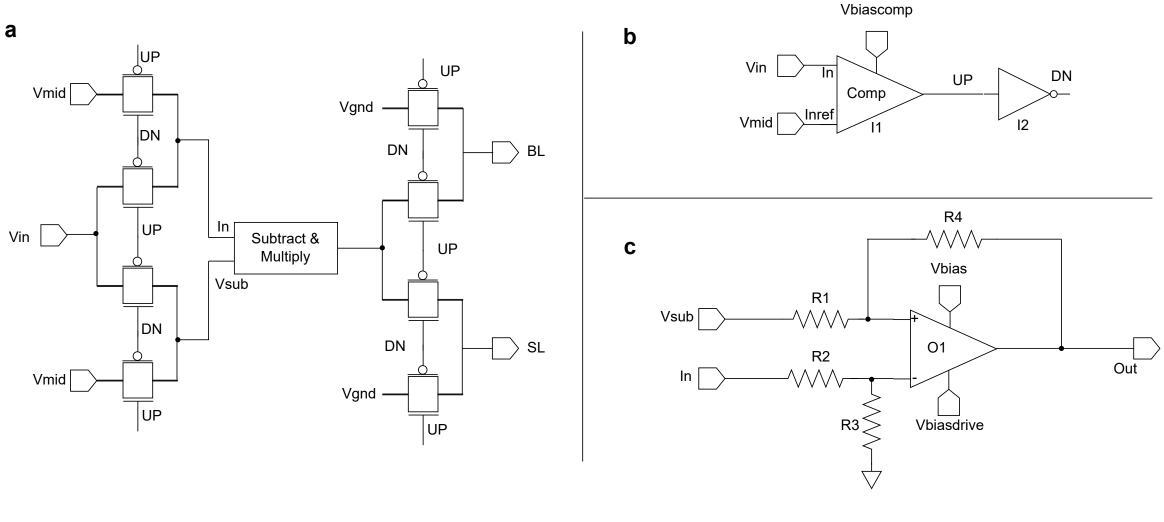


Figure 6.5 Circuits in the programming block (VDSP amplifier). **a** Top-level schematic of the programming block, consisting of a subtract-and-multiply block along with multiple transmission gates (TGATEs). **b** On-chip comparator circuit. **c** The subtract-and-multiply block.

The programming voltage generator block, shown in Figure 6.5, generates the `BL` and `SL` programming voltages based on the neuron membrane potential. The unipolar nature of the membrane potential determines whether potentiation or depression occurs, depending on whether the membrane potential is greater than or less than `VMID`. The `VMID` value corresponds to the resting state potential of the LIF neuron with bidirectional leakage. Figure 6.5a illustrates the overall circuit, which includes the subtract-and-multiply block. This block is composed of an operational amplifier and resistors, as detailed in Figure 6.5c. The transfer function for the VDSP amplifier block is:

$$V_{\mathrm{BL}} = \begin{cases} (V_{\mathrm{mid}} - V_{\mathrm{mem}}) \times 3 & \text{if } V_{\mathrm{mem}} < V_{\mathrm{mid}} \\ 0 & \text{otherwise} \end{cases} \tag{6.1}$$

$$V_{\mathrm{SL}} = \begin{cases} (V_{\mathrm{mem}} - V_{\mathrm{mid}}) \times 3 & \text{if } V_{\mathrm{mem}} > V_{\mathrm{mid}} \\ 0 & \text{otherwise} \end{cases} \tag{6.2}$$

This topology utilizes a differential operational amplifier to subtract voltages, as described in [281]. The amplification factor is set to 3, determined by the ratio of the feedback resistances (R4/R1). This amplification factor is crucial in defining the scaling factor discussed in chapter 4.

For increased configurability, the resistances R1 and R4 can be replaced with a memristive block, allowing dynamic adjustment of the amplification factor. Additionally, multiple parallel fixed shunt resistances could be incorporated to fine-tune the scaling factor with higher precision.

6.4.3 Input block

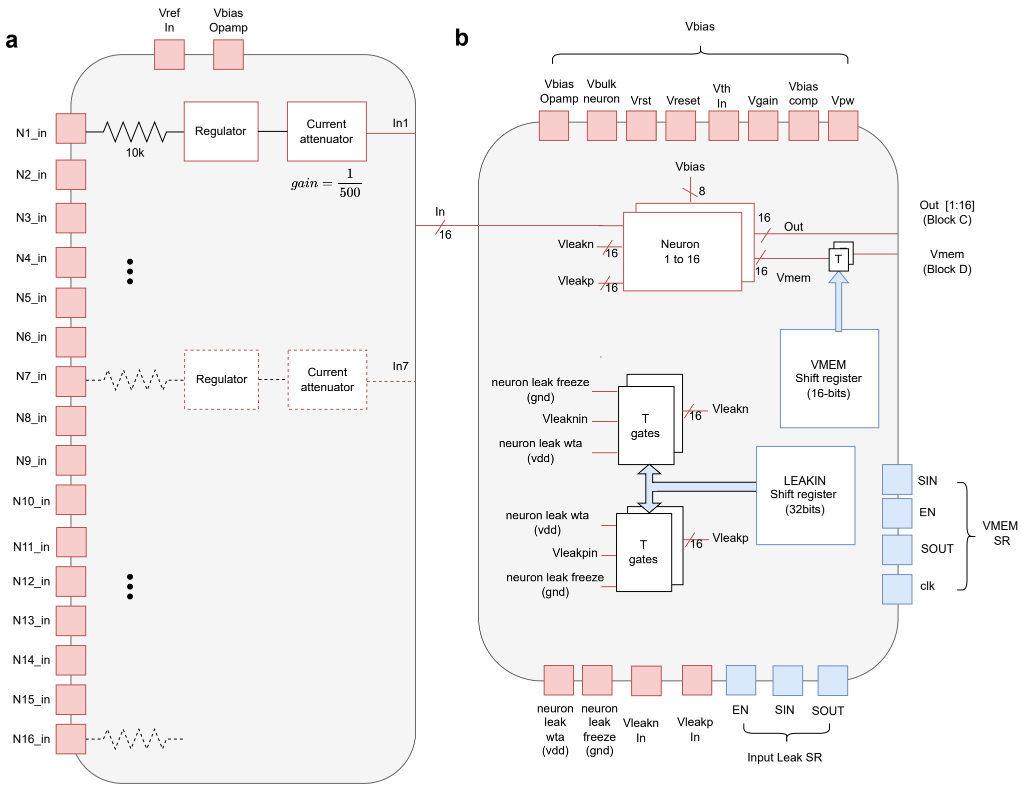


Figure 6.6 Schematic block diagram of Block (A-B). **a** This block comprises 16 instances of (i) fixed on-chip resistances, (ii) Low Dropout regulator (LDO), and (iii) Current Attenuator (CA). **b** The input neuron bank consists of 16 neurons, along with two Shift Register (SR)s for configuring the leak mode (freeze/typical/WTA) and addressing neurons to probe the membrane potential via the VMEM I/O pad and the VDSP amplifier.

Figure 6.6 represents the input block comprising Blocks A and B. This block consists of fixed resistances, an LDO (low-dropout regulator), a current attenuator, and an input neuron bank. The neuron bank is composed of 16 neurons and two Shift Register (SR)s: LEAKIN and VMEM. The 32-bit LEAKIN shift register controls the selection of the bias voltages `Vleakp` (upward leakage) and `Vleakn` (downward leakage), which are connected to the transistor gates in the DPI block (chapter 5). These voltages regulate leakage in three operational modes: typical, freeze (no leakage), and WTA (winner-take-all, maximum leakage).

In typical mode, `Vleakp` and `Vleakn` are connected to the `Vleakp In` and `Vleakn In` analog I/O pads on the NBB, respectively. In freeze mode, `Vleakn` is connected to the `NeuronLeakGnd` pad, tied to `gnd`, while `Vleakp` is routed to the `NeuronLeakVdd` pad, which is supplied with a 3.3V bias, disabling both upward and downward leakage. In WTA mode, the connections are reversed: `Vleakn` is tied to 3.3V, and `Vleakp` is connected to `gnd`.

The second shift register in Block B is `VMEM`, a 16-bit register where only one bit is set high, representing the index of the input neuron whose membrane potential is connected to the `VMEM` signal. This `VMEM` signal is routed to the programming block in Block C and can be monitored via the NBB's `VMEM_ext` analog I/O pad using the on-chip voltage follower circuit.

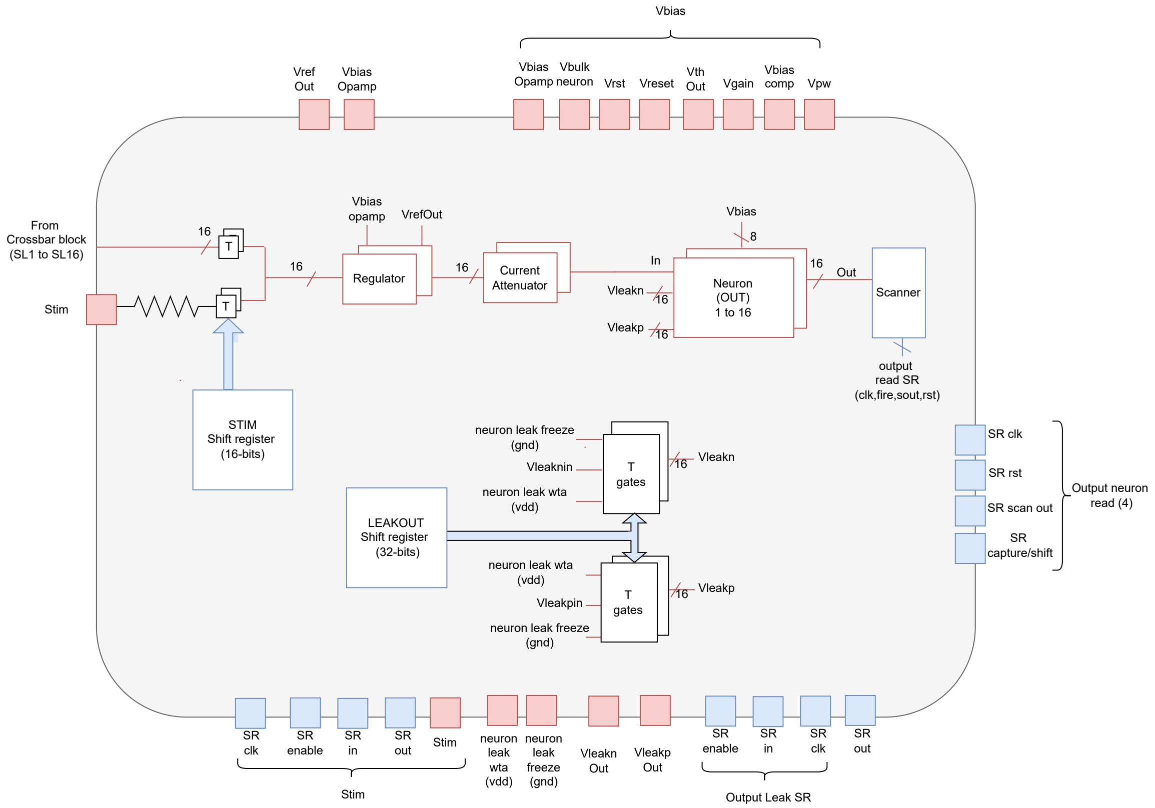


Figure 6.7 Schematic diagram of Block D: Output neuron bank, consisting of 16 LIF neurons and shift registers for (i) configuring the leak rate, (ii) selecting the output neuron index for stimulation, and (iii) asynchronously scanning out neuron activity.

The output neuron bank, or Block D, is shown in Figure 6.7. It has two inputs: `LEAKOUT` (32-bit) and `STIM` (16-bit), along with an output connected to scanner shift registers. The `LEAKOUT` shift register operates similarly to the `LEAKIN` register in the input neuron bank, allowing each neuron to be configured in one of three leak modes: freeze (minimum leakage), inference (typical leakage), and WTA (maximum leakage). This configuration provides independent control over each neuron's behavior across different operational modes.

The `STIM` shift register determines whether the respective source line is connected to the common `STIM` analog I/O pad, which interfaces with the downstream signal chain, including the LDO, current attenuator, and neurons. The `STIM` pad is connected through a 10kΩ fixed resistor, implemented in CMOS. This stimulation mechanism aids the VDSP learning process by externally exciting the correct output neuron, based on the label, to facilitate learning. This teaching mechanism is particularly useful for initializing the learning process from a subset of labeled samples, helping to establish preliminary receptive fields. These receptive fields can later be refined through fully unsupervised learning.

The final register, the scanner output register, is responsible for capturing the output generated by the neuron bank and will be described in more detail in the next section.

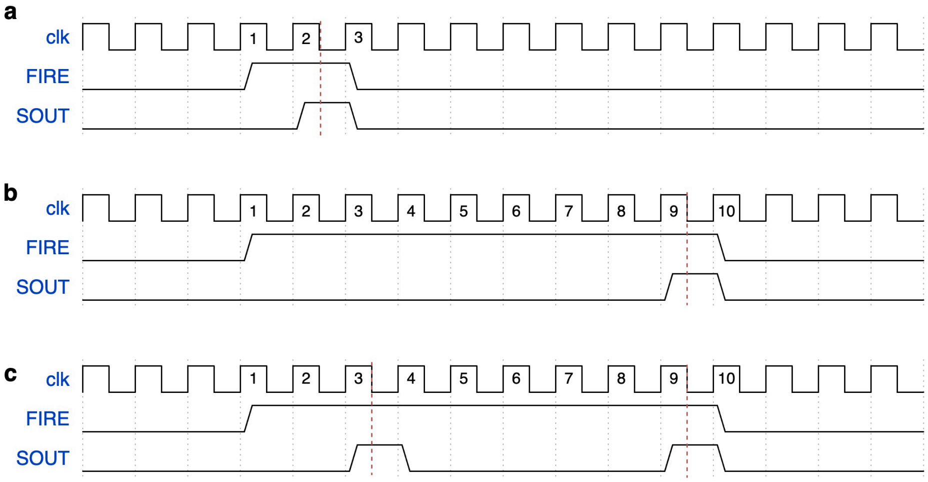


Figure 6.8 Output block timing diagram illustrating the scanning logic for `clk`, `FIRE`, and `SOUT` signals. **a** Timing when neuron 2 fires. **b** Timing when neuron 9 fires. **c** Timing when neurons 3 and 9 fire simultaneously.

The spikes from the output neuron bank are transmitted through asynchronous digital logic. When any of the 16 neurons generates a spike, the outputs of all neurons are captured in flip-flops, and the `FIRE` signal is activated, as shown in Figure 6.8. Once the spikes are captured, the 16 bits are serially shifted out through the `SOUT` pad on each falling edge of the externally supplied clock signal, continuing until all active bits are transmitted. For example, Figure 6.8a illustrates the output waveform when the second neuron fires, and Figure 6.8b shows the waveform for the ninth neuron. When all active bits have been shifted out, the `FIRE` signal is reset to zero, allowing the digital logic to resume monitoring the output neuron bank. In the case where multiple neurons fire simultaneously, the respective bits are shifted out, as demonstrated in Figure 6.8c.

The architecture implements a handshaking protocol to synchronize the capture and transfer of data between input signals and internal registers. The process begins with the activation of the `capture` signal. When `capture = '1'` and a rising clock edge (`clk_edge = '1'`) is detected, the incoming data bit (`din`) is stored in the `input_value` register at the position indexed by the `counter`. After each data capture, the `counter` is decremented to ensure proper bit ordering.

Once the `capture` signal transitions from '1' to '0', the system acknowledges the completion of data capture by setting the `done` signal to '1'. At this point, the captured data is transferred to the `value` register, and both the `input_value` register and the `counter` are reset to their initial states. The entire process is synchronized to the clock, ensuring reliable data transfer. The `done` signal is cleared as soon as the clock edge is no longer active.

6.4.4 Top Architecture

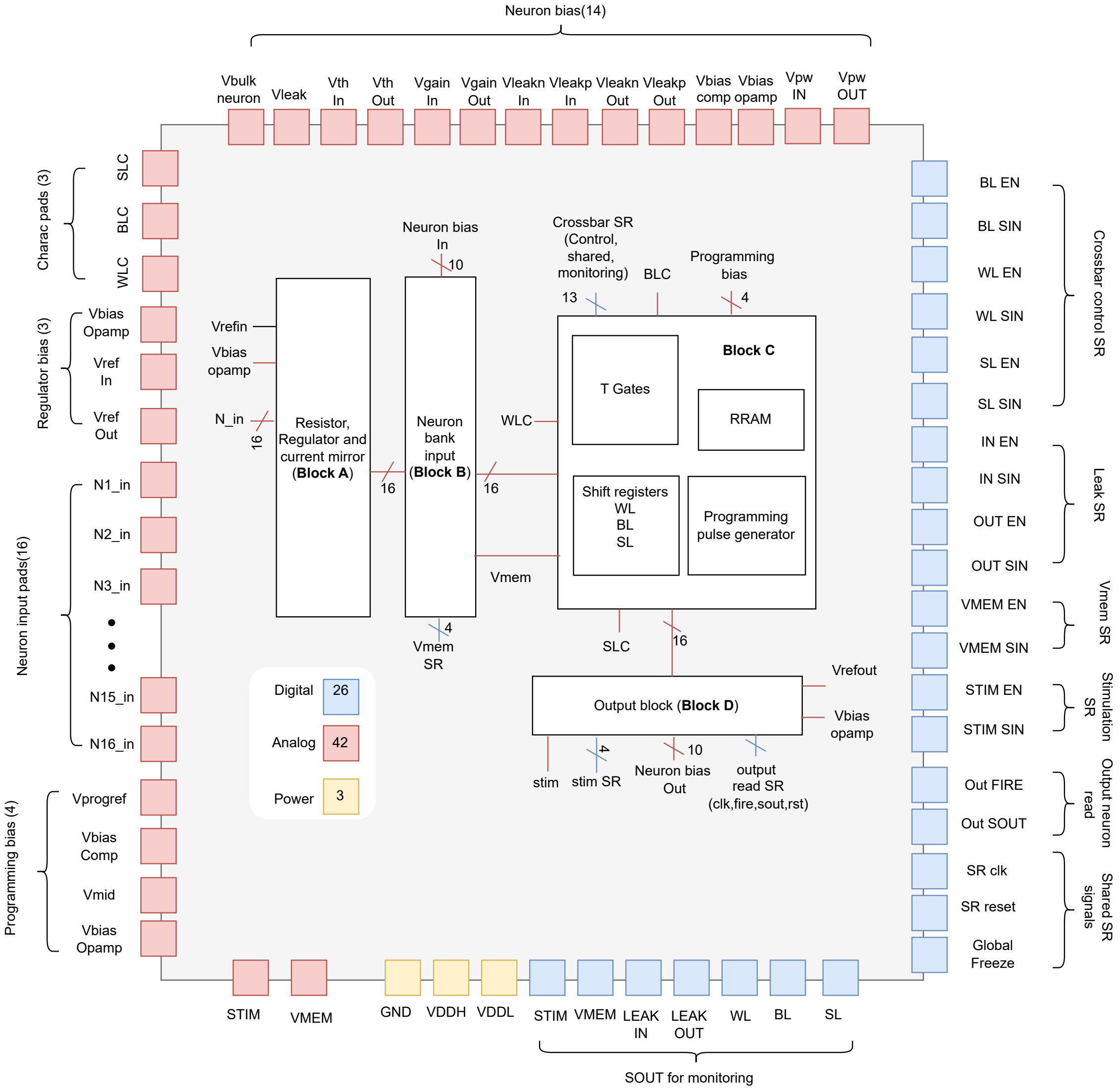


Figure 6.9 Schematic block diagram of NBB. The diagram consists of four modules: Block (A-D). Analog I/O pads are highlighted in red, and digital I/O pads are shown in blue. Additionally, power supply pads (in yellow) are provided for GND, VDDL, and VDDH to supply power to various circuits. The analog I/O pads (in red) include crossbar characterization pads and bias voltages for (i) regulators (LDO), (ii) input neurons, (iii) output neurons, and (iv) the programming block. VMEM and STIM pads are used for monitoring membrane potential and stimulating the input and output neuron banks, respectively.

The top schematic diagram is shown in Figure 6.9. The design operates at two power levels: `VDDH` (3.3V) and `VDDL` (1.2V). A higher voltage of up to 3.3V is necessary to form memristive devices, which is why the addressing and interface circuits are powered by VDDH. In contrast, the digital control logic is powered by VDDL to minimize energy consumption during communication, reduce programming power consumption, and lower latency.

Shift Register	**Bit Width**
BL (Bit Line)	32-bit
WL (Word Line)	32-bit
SL (Source Line)	32-bit
STIM	32-bit
VMEM (Membrane Voltage)	16-bit
LEAKIN	32-bit
LEAKOUT	32-bit

Table 6.1 Configuration Shift Registers and their Bit Widths in NBB

In total, there are seven configuration shift registers in the NBB, as shown in Table 6.1. The 32-bit registers assign two bits per row or column, managing the configuration of the respective row or column. In contrast, the 16-bit registers, such as STIM and VMEM, have only a single bit corresponding to the input neuron or output neuron index, respectively.

Register	**Mode**	**Control code (2-bit)**
LEAK	Freeze	00
	Inference	01
	Winner-Take-All	11
BL	Grounded	00
	Inference	01
	Characterization	01
SL	Grounded	00
	Inference	10
	Characterization	11
WL	Grounded	00
	Inference	01
	Characterization	11

Table 6.2 Configuration codes and Descriptions for various operating modes

The two-bit control code for each row or column is listed in Table 6.2. The 32-bit value used to program the register consists of 16 concatenated control codes, which are programmed serially through the shift registers.

Name	IO (Analog) ID	Name	IO (Analog/Digital) ID
BLC	A4	Neuron_IN_6	A28
WLC	A5	Neuron_IN_7	A29
SLC	A6	Neuron_IN_8	A30
VMEM	A7	Neuron_IN_9	A31
STIM	A8	Neuron_IN_10	A32
Neuron_IN_1	A23	Neuron_IN_11	D18
Neuron_IN_2	A24	Neuron_IN_12	D19
Neuron_IN_3	A25	Neuron_IN_13	D20
Neuron_IN_4	A26	Neuron_IN_14	D21
Neuron_IN_5	A27	Neuron_IN_15	D22
		Neuron_IN_16	D23

Table 6.3 Input and output pads with Analog and Digital Input/Output (IO) signals assigned to IDs of Lotus for IN1 to IN16, BLC, WLC, SLC, VMEM, and STIM.

Table 6.3 presents the assignment of analog input and monitoring pads, along with their corresponding Analog or Digital Input/Output (IO) identifiers (IDs), for key signals in the NBB data path. The BLC pad provides access to the crossbar's bit line during individual read and write operations, while WLC and SLC represent the word line and source line characterization pads for memory access.

The VMEM pad is used to access the membrane voltage of one selected neuron out of the 16 input neurons. A voltage follower ensures that probing this signal does not compromise the integrity of the data path. The neuron is addressed through the 16-bit VMEM shift register. The STIM pad is used for neuron stimulation, providing input to one or more output neuron banks. Since there is a single analog input signal, it is shared among multiple neurons, and multiple bits in the 16-bit STIM shift register can be set to '1' to stimulate several neurons simultaneously.

The signals Neuron_IN_1 to Neuron_IN_16 represent the input channels for the neuron data path, allowing the system to receive data from external sources. The first ten input channels are assigned to a dedicated Analog pulse measurement unit (APMU), while the remaining six are routed through digital General-Purpose Input/Output (GPIO) pins on the Lotus platform.

Bias Voltage	IO (Analog) ID	Inference Mode (V)	Learning /freeze Mode (V)
`Vref_in`	A1	0.8	0.8
`Vref_out`	A2	1	1
`Vbiasopamp`	A3	2.4	2.4
`Vbiasopamp_shared`	A9	2.4	2.4
`Vbiascomp`	A10	1	2.4
`Vbulk_neuron`	A11	1.2	3.3
`Vleak_neuron`	A12	0.6	0.6
`Vth_input`	A13	1.2	1.2
`Vgain_input`	A14	2.1	2.1
`Vtaun_input`	A15	1.2	0
`Vbtaup_input`	A16	0	0
`Vpw_input`	A17	1	1
`Vth_output`	A18	1.2	1.2
`Vgain_output`	A19	2.1	2.1
`Vtaun_output`	A20	1.2	0
`Vbtaup_output`	A21	0	0
`Vpw_output`	A22	1	1

Table 6.4 Bias Voltages (Names in schematics) with identification and Analog and Digital IO signals for Inference and Freeze (Learning) Modes.

Table 6.4 lists the analog bias voltages and their corresponding APMU IDs for both inference and learning modes. These signals provide flexibility in modulating the behavior of the neuron and synaptic reading circuits.

- `Vref_in` and `Vref_out`: Supply the reference input and output voltages for the 10k fixed synaptic input resistance or the respective column.
- `Vbiasopamp` and `Vbiasopamp_shared`: Set the bias for the operational amplifiers used in the LDO and voltage follower for VMEM monitoring.
- `Vbiascomp`: Controls the bias of the neuron comparator, shifting from 1V in inference mode to 2.4V in freeze mode to prevent spike generation during crossbar programming.
- `Vbulk_neuron`: Modulates the bulk voltage for the neuron bank, adjusting the upward leak rate. It is set to 1.2V for the typical leak rate in inference mode and 3.3V in freeze mode for online learning.
- `Vleak_neuron`: Manages the neuron's resting state potential, which remains at 0.6V across both modes.
- `Vth_input` and `Vgain_input`: Control the threshold and gain of the input neuron bank, fixed at 1.2V and 2.1V, respectively, in both modes.

- `Vtaun_input`: Defines the time constant for downward (negative) leakage, set to 1.2V in inference mode and 0V in freeze mode.
- `Vbtaup_input`: Maintains a bias level of 0V across both modes.
- `Vpw_input`: Controls the pulse width for the input.

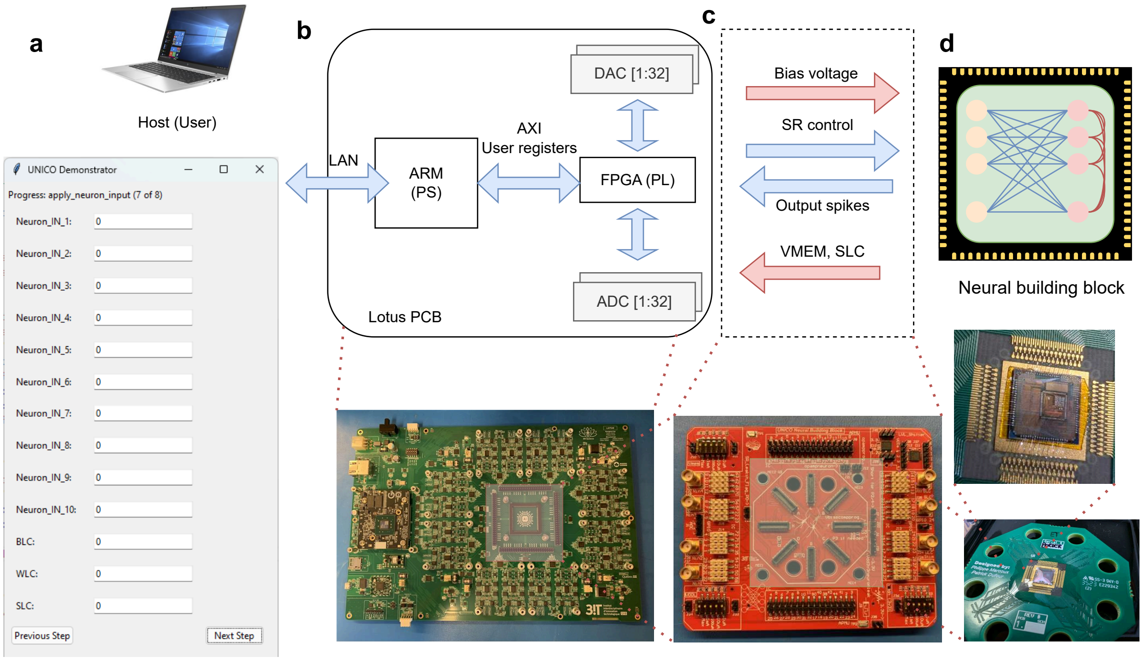
For the output neuron bank:

- `Vth_output` and `Vgain_output`: Regulate the threshold and gain, independent of the input neuron bank.
- `Vtaun_output` and `Vbtaup_output`: Behave similarly to the input neurons, controlling time constants.
- `Vpw_output`: Sets the pulse width for the output neuron bank.

It is important to individually configure the input/output pulse width levels, as the spikes generated by the input neurons are applied to the gates of the synaptic cells, while the output spikes are captured by the activity-sensing logic in the output neuron bank, which uses a 16-bit shift register for scanning.

6.5 Results

6.5.1 Configuring the chip



Progress: apply_neuron_input (7 of 8)

Neuron_IN_1: 0

Neuron_IN_2: 0

Neuron_IN_3: 0

Neuron_IN_4: 0

Neuron_IN_5: 0

Neuron_IN_6: 0

Neuron_IN_7: 0

Neuron_IN_8: 0

Neuron_IN_9: 0

Neuron_IN_10: 0

BLC: 0

WLC: 0

SLC: 0

Figure 6.10 Communication and control path from the PC to the UNICO chip via the Lotus system **a** The host communicates with Lotus through a LAN interface and a Graphical user interface (GUI). **b** Simplified schematic (top) and photograph (bottom) of the Lotus platform comprising the Zynq SoC, ADCs, and DACs. **c** (top) The digital signals (blue) and analog signals (red) are sent/received from Lotus to the ASIC. (bottom) The daughter board for routing IO signals. **d** (top) Schematic representation of the packaged chip with 84 IO pads (Digital/Analog/Power). Wire-bonded die (middle) to the carrier PCB (bottom).

Figure 6.10 outlines the user interface to the NBB through LAN and the Lotus board. The programming was performed via a custom HDL module implemented in the programmable logic (PL) of the Zynq SoC, located on the Lotus characterization board, which supports 32 channels of analog and 32 channels of digital I/O. The signals are routed from the *Lotus* PCB to the *UNICO* chip through a mezzanine routing PCB, as shown in Figure 6.10. The mezzanine board connects the digital and analog signals to the ASIC. The chip is wire-bonded to the carrier PCB, facilitating the proper routing of I/O signals between the UNICO chip and the Lotus system for testing and characterization.

Shift Register	**Digital Pin**	**IO (Digital)**
BL	SIN	D9
	EN	D10
WL	SIN	D11
	EN	D12
SL	SIN	D13
	EN	D14
LEAKIN	SIN	D3
	EN	D4
LEAKOUT	SIN	D5
	EN	D6
STIM	SIN	D15
	EN	D16
VMEM	SIN	D7
	EN	D8
clk	Clock Signal	D2
reset	Reset Signal	D1
GF	Global Freeze	D17

Table 6.5 Digital IO Pin Assignments for Each Shift Register and Control Signals

Table 6.5 lists the digital pin assignments for each shift register along with the corresponding GPIO connections. Each shift register—BL, WL, SL, LEAKIN, LEAKOUT, STIM, and VMEM—has associated signals SIN and EN, which are connected to specific GPIO pins. For example, BL shift register uses GPIO pin 9 for SIN and pin 10 for EN, while WL uses pins 11 and 12, respectively. The table also specifies global control signals, including the clock signal (clk) assigned to GPIO pin 2, the reset signal (reset) to pin 1, and the global freeze (GF) to pin 17. These assignments allow for the control and serial programming of the on-chip shift registers through the digital pins and GPIO connections of the FPGA in the Lotus PCB.

ID	Name	Width	Description
0	`clock_div`	16-bit	Set the clock divider
1	`reset`	1-bit	Set the reset signal
2	`BL` (Bit Line)	32-bit	Set the Bit Line (BL) value
3	`WL` (Word Line)	32-bit	Set the Word Line (WL) value
4	`SL` (Source Line)	32-bit	Set the Source Line (SL) value
5	`LEAKIN`	32-bit	Set the LEAKIN value
6	`LEAKOUT`	32-bit	Set the LEAKOUT value
7	`VMEM` (Membrane Voltage)	16-bit	Set the Membrane Voltage (VMEM) value
8	`STIM`	16-bit	Set the STIM value
9	`program_triggers`	4-bit	Set the program trigger signals (`CB`, `LEAK`, `VMEM`, `STIM`)
10	`freeze_value`	1-bit	Read the freeze value (write zero to reset)
11	`freeze_APMU`	32-bit	Set which APMU trigger is forced to zero when freeze signal is active
12	`freeze_enable`	1-bit	Enable freezing
26	`program_done`	5-bit	Read the program done signal (`CB`, `LEAK`, `VMEM`, `STIM`, `program_done`)
28	`readreg_output`	16-bit	Read the output data from readreg
50	`fifo_spikes`	32-bit	Read the number of spikes in the FIFO
51	`fifo_timestamp_sec`	32-bit	Read the next value in the FIFO (timestamp seconds)
52	`fifo_timestamp_frac`	32-bit	Read the next value in the FIFO (timestamp fractional part and neuron number) (28-bit fractional, 4-bit neuron number)

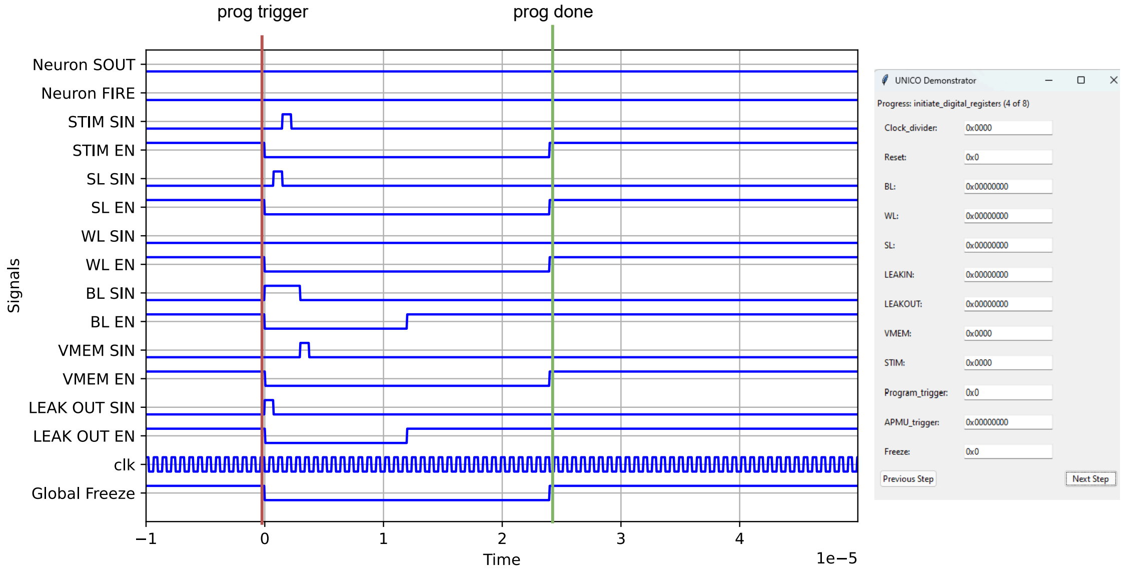
Table 6.6 AXI Slave Registers: ID, Name, Width, and Description

The AXI registers in Zynq SoC facilitate the communication between programmable system (PS) composed of ARM CPU core and programmable logic (PL) implemented through FPGA. Table 6.6 briefly describes these AXI registers in user space. The user (linux host on PS) can interact or modify these registers which set and trigger the on-chip shift registers of NBB. Each register is identified by its `ID`, and the table details the `Name`, `Width`, and `Description` of the corresponding registers. For instance, `clock_div` (ID 0) is a 16-bit register used to set the clock divider, while the `reset` register (ID 1) is a 1-bit register for controlling the reset signal. Registers like `BL` (Bit Line), `WL` (Word Line), and `SL` (Source Line) are 32-bit wide and allow users to set values for their respective shift registers. Other registers include `VMEM` (ID 7) and `STIM` (ID 8), both 16-bit registers, which control the membrane voltage and stimulation shift registers, respectively.

The `program_triggers` register (ID 9) is 4-bit wide and is used to set program trigger signals for different components such as `CB`, `LEAK`, `VMEM`, and `STIM`.

Upon detecting the program trigger from the PS, the FPGA module serially transmits the value stored in the respective user register. The global freeze (GF) is set to low throughout the configuration operation to ensure all the signals are connected to ground during shifting of bits. Upon successful serialization of all 16/32 bits, the FPGA module generates a `program_done` signal. This `program_done` in-turn de-freeze the chip by setting GF to one, enabling all the level shifters on chip, and thus applying the configuration.

The table also includes registers for handling freeze control, such as `freeze_value` (ID 10), `freeze_APMU` (ID 11), and `freeze_enable` (ID 12), which manage freezing mechanisms in the system. User registers also exist for interfacing with the FIFO, capturing output neuron bank activity as tuples of neuron index and timestamp. For instance, `fifo_spikes` (ID 50) reads the spike count in the FIFO, while `fifo_timestamp_sec` (ID 51) and `fifo_timestamp_frac` (ID 52) stores timestamps corresponding to neuron numbers.



Progress: initiate_digital_registers (4 of 8)

Clock_divider: 0x0000

Reset: 0x0

BL: 0x00000000

WL: 0x00000000

SL: 0x00000000

LEAKIN: 0x00000000

LEAKOUT: 0x00000000

VMEM: 0x0000

STIM: 0x0000

Program_trigger: 0x0

APMU_trigger: 0x00000000

Freeze: 0x0

Figure 6.11 Configuring on-chip shift registers. (left) Graphical user interface (GUI) for custom configuration by user.

The shift registers on the chip were programmed using a custom value entered by the user in a dedicated GUI, as shown in Figure 6.11.

The clock_divider argument sets the frequency of the clock signal that is generated in the FPGA to control the chip. By default, this frequency is set to 1 MHz (see Figure 6.11). Additionally, the reset signal is an active-high signal, provided from the FPGA to the chip for resetting the system.

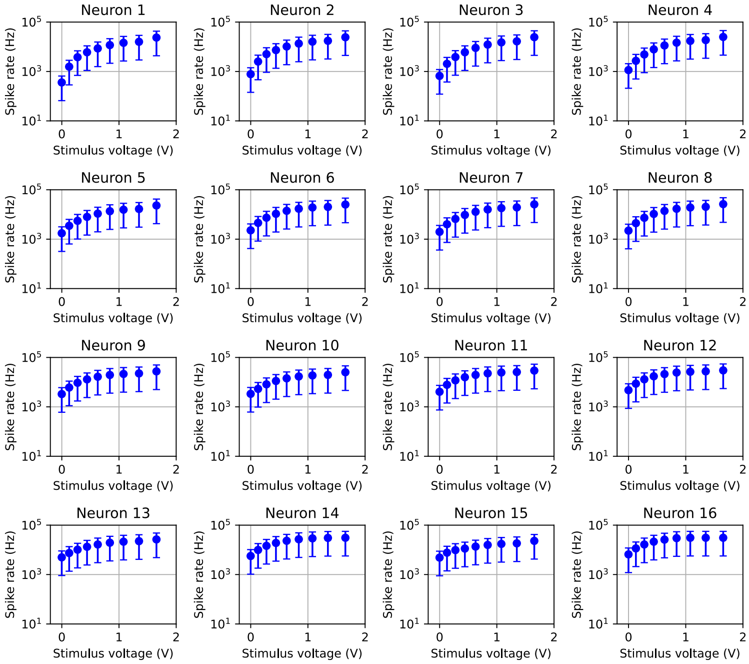
Once the next step button is pressed on the GUI (Figure 6.11(right)), the program trigger signal is set high from the PS. The HDL module on the FPGA, upon receiving the trigger, reads all the respective user registers corresponding to values to be programmed (entered via the GUI) and shifts them out through serial input (SIN) and enable (EN) signals, as shown in Figure 6.11(left). The EN signal remains low for 16 clock cycles (for STIM and VMEM) or 32 clock cycles (for WL, BL, SL, LEAKIN, LEAKOUT), depending on the width of the shift register.

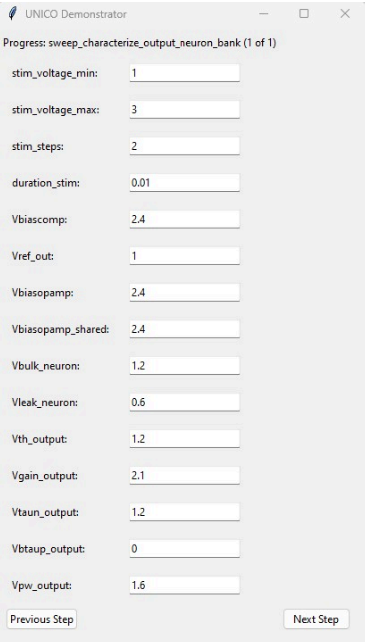
The global freeze (GF) signal remains low throughout the programming operation. The GF signal is used to enable the level shifters, which convert the shift register bits to 3.3V to enable the selected transmission gates, as shown in Figure 6.3. During the programming operation, the register values are not stable as they are shifting with each clock cycle. Therefore, the shift register bits are only applied once all EN signals for SR programming are set high (indicating that programming of all registers is complete).

In freeze mode (when the GF signal is low), a 0 (low) is applied to all transmission gates, selecting the first configuration, which connects all the crossbar lines to the ground plane.

Neuron_SOUT and FIRE are signals from the chip, read by the FPGA. These correspond to signals from the on-chip neuron activity scanner, which will be discussed in the following sections.

6.5.2 Neuron bank characterization





Progress: sweep_characterize_output_neuron_bank (1 of 1)

stim_voltage_min: 1
stim_voltage_max: 3
stim_steps: 2
duration_stim: 0.01
Vbiascomp: 2.4
Vref_out: 1
Vbiasopamp: 2.4
Vbiasopamp_shared: 2.4
Vbulk_neuron: 1.2
Vleak_neuron: 0.6
Vth_output: 1.2
Vgain_output: 2.1
Vtaun_output: 1.2
Vbtaup_output: 0
Vpw_output: 1.6

Figure 6.12 Characterization results of the output neuron bank (left) and GUI parameters for the parametric sweep of the output neuron bank (right).

The 16 output neurons on the chip were characterized by individually stimulating them via the `STIM` pad, which connects to each neuron through a fixed 10kΩ on-chip resistor. The activity of the output neuron bank was measured using asynchronous digital logic implemented through a shift register. When an output neuron becomes active, the states of all 16 output neurons are captured in flip-flops. The `FIRE` signal is then triggered, and the captured bits are transmitted serially with each clock cycle until all active neuron states have been transmitted.

Variations in the firing rate of each neuron were observed (see Figure 6.12), primarily due to delays introduced by the handshaking logic. This occurs because the ARM core on the Zynq FPGA manages several functions, leading to variability in handling the interrupts generated by the FPGA. This variability is represented by error bars in the plot.

A second source of variation arises from transistor dimension mismatches, caused by the finite resolution of the semiconductor patterning processes. Additionally, the resistance of metal interconnect lines becomes significant when operating at very low excitation currents, resulting in a trend of increasing minimum firing rates from neurons 1 through 16.

6.5.3 VDSP based learning

Algorithm 1: Memory Programming and Inference Process

```
Data: Neuron index (8-bit), digital control signals, analog input signals
Result: Memory programmed and system returns to inference mode
STIM register (8-bit) is configured with the neuron index. The LEAK SR register (digital) is set to
  inference mode;
STIM signal (analog) is applied through APMU, triggering a neuron firing event that activates the FIRE
  signal (digital);
Upon FIRE activation, the GF signal (digital) is set to low, and LEAK SR is configured to freeze all
  input neurons;
The SOUT signal (digital) is sampled at each clock cycle to detect the active neuron index ($i$, 8-bit);
Lateral inhibition is activated by configuring the LEAKOUT SR (digital) of all other neurons to WTA
  mode for 10 ms;
The BL and SL SR registers (digital) are shifted to set column $i$ to programming mode, grounding all
  other columns (GND);
for each row from 1 to 16 do
    Set the WL and VMEM SR registers (digital) to activate the current row;
    The membrane voltage (analog) sets the programming voltage for BL and SL (analog);
    A 200 ns pulse is applied through the WLC signal (analog) to set the compliance current for
      programming;
After programming the 16th row, deactivate the freeze mode and return the GF signal (digital) to
  inference mode;
```

The algorithm summarized above outlines the memory programming process in the NBB system. It involves configuring the shift registers, stimulating a neuron via the STIM signal, and performing lateral inhibition in conjunction with programming the synaptic columns. The synapses of each row are updated serially. Once the programming is complete, the system returns to inference mode by resetting the global freeze signal.

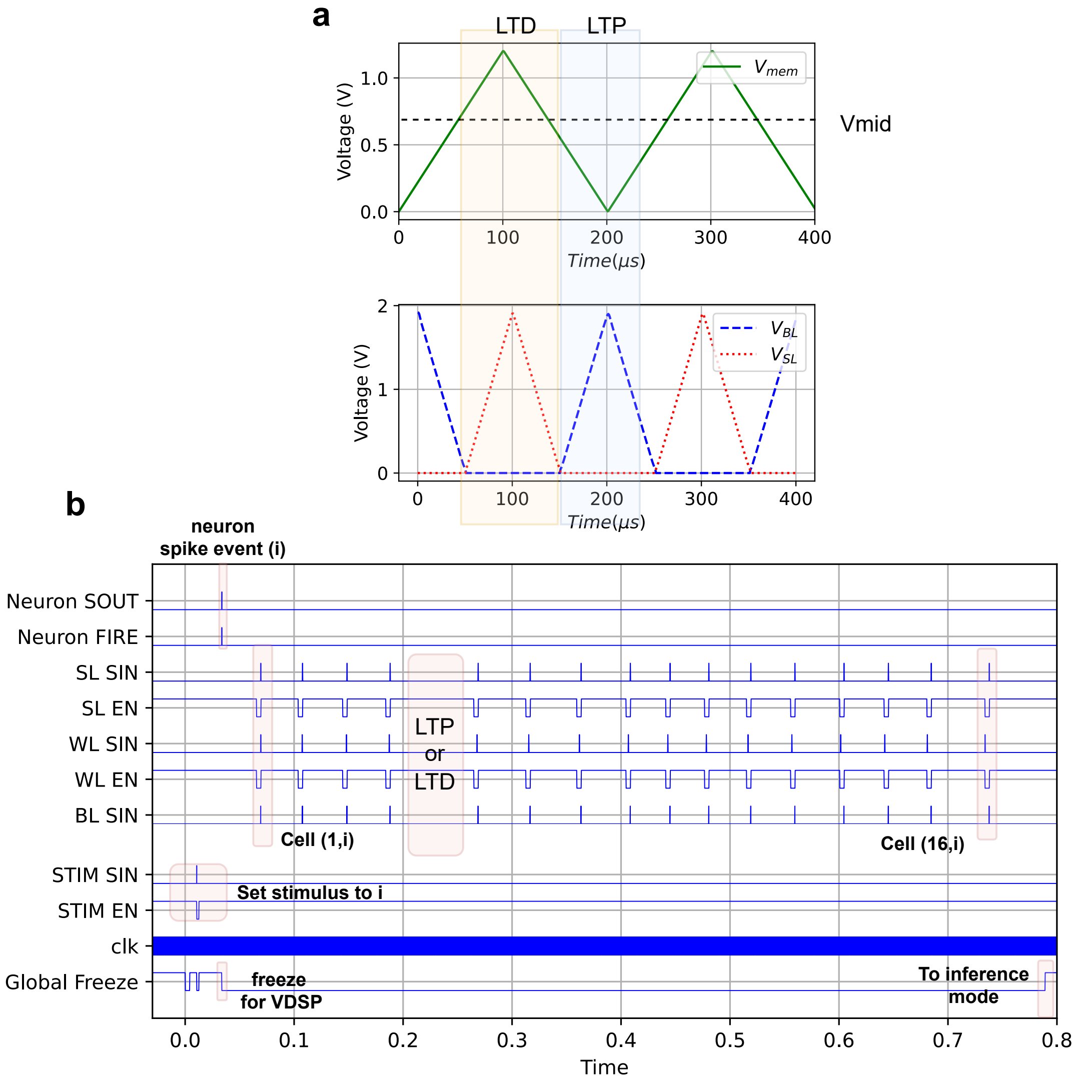


Figure 6.13 VDSP programming results. **a** Response of the VDSP amplifier circuit in the programming block (obtained through SPICE simulations). The `Vmem` and `Vmid` (threshold for LTP/LTD) signals are applied as inputs, generating the `VBL` and `VSL` signals. **b** Timing diagram obtained by characterizing the learning operation. A neuron spike event triggers the freeze (`GF=0`), followed by the serial update sequence of each row. The addressing of each row is highlighted, and between the addressing operations, LTP/LTD occurs by applying the output of the programming amplifier to the crossbar cell.

The SPICE simulation output of the LTD amplifier block is shown in Figure 6.13a. However, direct probing of V_{LTP} and V_{LTD} is not possible, as no dedicated I/O pads are available for these signals. In the event of circuit malfunction, the Analog to Digital Converter (ADC), Digital to Analog Converter (DAC), and a user program running on the processing system (PS) of the *Zynq* platform can be used in conjunction with the Lotus board to create a custom scaling factor. This scaling factor can then be provided as a parameter in the GUI, enabling users to fine-tune the amplification factor based on the behavior of the integrated memristive devices in future implementations.

Furthermore, Figure 6.13b shows the measurement results that illustrate the VDSP programming sequence. The diagram shows the sequence of events triggered by a spike in the output neuron bank, followed by the freeze operation and serial update of each synaptic row. In addition, it highlights how learning operations, such as LTP and LTD, are applied through the programming amplifier during the row addressing phase, completing the synaptic update cycle.

Name	**Value**	**Name**	**Value**
`stim_index`	1	`amp_threshold`	0.6 (V)
`stim_voltage`	1.1 (V)	`VBL_read`	1.2 (V)
`duration_read`	1 (s)	`VSL_read`	1 (V)
`duration_WTA`	1e-4 (s)	`VWL_read`	3.3 (V)
`amp_factor_ltp`	3	`VWL_LTP`	3.3 (V)
`amp_factor_ltd`	3	`VWL_LTD`	3.3 (V)
`Vbiascomp`	2.4 (V)	`Vref_out`	1 (V)
`Vbiasopamp`	2.4 (V)	`Vbiasopamp_shared`	2.4 (V)
`Vbulk_neuron`	1.2 (V)	`Vleak_neuron`	0.6 (V)
`Vth_output`	1.2 (V)	`Vgain_output`	2.1 (V)
`Vtaun_output`	1.2 (V)	`Vbtaup_output`	0 (V)
`Vpw_output`	1.6 (V)		

Table 6.7 Characterization of VDSP learning experiment: GUI parameters for stimulating a single neuron (crossbar column) to trigger the update sequence.

Table 6.7 summarizes key parameters for characterizing the learning process, with some voltages previously defined in Table 6.4. The parameter `stim_index` refers to the stimulus index, set to 1, and `stim_voltage` defines the applied stimulus voltage, set at 1.1V. The durations of the read and Winner-Takes-All (WTA) phases are denoted by `duration_read` and `duration_WTA`, with values of 1 second and 1×10^{-4} seconds, respectively.

The amplification factors for Long-Term Potentiation (LTP) and Long-Term Depression (LTD) are controlled by `amp_factor_ltp` and `amp_factor_ltd`, both set to 3, with a threshold voltage `amp_threshold` of 0.6V. The bit line, source line, and word line read voltages (VBL_read, VSL_read, and VWL_read) are 1.2V, 1V, and 3.3V, respectively. Similarly, VWL_LTP and VWL_LTD remain constant at 3.3V during LTP and LTD operations.

Additional parameters such as Vbiascomp, Vref_out, Vbiasopamp, Vbiasopamp_shared, and neuron-specific voltages like Vbulk_neuron and Vleak_neuron retain the values previously listed in Table 6.4.

6.6 Discussion

6.6.1 Back end of the line (BEOL) Integration

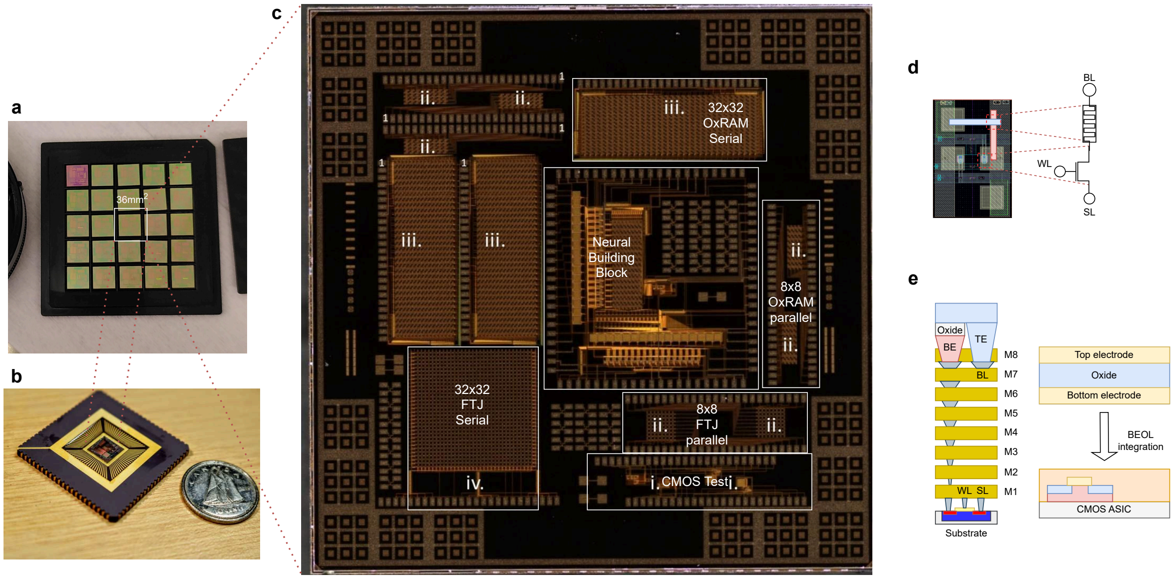


Figure 6.14 UNICO ASIC for Back end of the line (BEOL) integration of synaptic memories through 1T1R architecture **a** Bare silicon dies received from the foundry. **b** A single die wire-bonded to a carrier (package). **c** Annotated micrograph of the ASIC comprising (i) CMOS test blocks, (ii) 8x8 parallel access Ferroelectric Tunnel Junction (FTJ) and Resistive Random Access Memory (RRAM), (iii) 32x32 RRAM serial access, (iv) 32x32 serial access FTJ, and a neural building block in the center. **d** Stack (side view) of the integrated chip, comprising a CMOS substrate, followed by 8 levels of metal interconnects, and Bottom Electrode (BE)/Oxide/Top Electrode (TE) of the BEOL-integrated memory. **e** Placement of vias on the last metal layer (top view) for connecting the bottom and top metal electrodes of the memristor.

The entire UNICO chip consists of three main blocks: the NBB (Neural Building Block), the MCC (Memory/RRAM Characterization Cell), and the Neuron Test Block & CMOS Test Structures. The Neural Building Block is the core of the design, serving as the modular, self-sufficient, and lifelong learning component of the CMOS-RRAM integrated neuromorphic system-on-chip. This block comprises neurons implemented with CMOS transistors, synapses realized as BEOL-integrated memristive devices, logic for plasticity, and peripheral circuits to manage the input response and readout system.

For the integration of TiO_2 and HfO_2-based memristors in the back-end-of-line, vias were placed on the eighth (last) metal layer of the fabricated CMOS ASIC. In addition to the NBB, the full chip includes several 8x8 and 32x32 1T1R arrays for extensive characterization. Beyond the TiO_2 and HfO_2 devices used in the NBB, HZO-based ferroelectric devices will also be integrated into the CMOS ASIC. An annotated micrograph of the entire chip is shown in Figure 6.14. More details on pin mapping of all characterization arrays and NBB are provided in Appendix A. Although the fabrication process and memristor measurements are beyond the scope of this thesis, detailed information on the fabrication recipe and measurement results will be published by the group in future work.

6.6.2 Demonstration of learning

In the future, following the BEOL integration of synaptic devices, the following demonstrations can be conducted to showcase learning with 2x1, 9x9, and 16x16 networks.

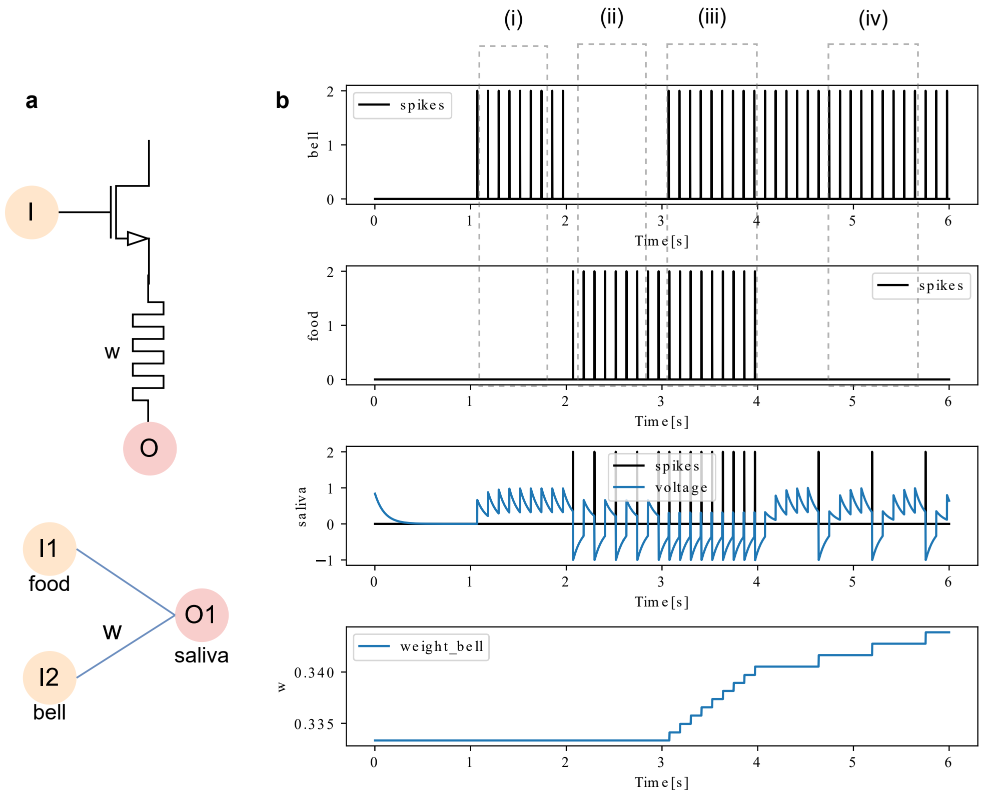


Figure 6.15 Associative learning demonstration. **a** Schematic of a 2x1 network. The input neuron feeds into the gate of the 1T1R synapse, and the output neuron is connected to the bottom electrode of the memristor. **b** In the initial phase (i), the bell is stimulated, but there is no activity in the saliva output neuron, as the initial synaptic weight is set to 0 (high resistance state, HRS). (ii) In the next phase, the food is activated, causing spikes in the output neuron. (iii) Next, both food and bell are activated together, leading to LTP in the weight between bell and saliva. Finally, in (iv), only the bell is activated, causing the saliva neuron to fire, confirming successful association. (Results correspond to simulations of VDSP.)

A simple demonstration of associative learning can be performed using a classical conditioning experiment (simulation shown in Figure 6.15). In this setup, two input neurons represent food (`I1`) and a bell (`I2`), while one output neuron (`O1`) represents saliva. The synaptic weight between the food and saliva neurons is fixed at 1 (low resistance state, LRS), while the synapse between the bell and saliva is trained using the VDSP learning rule. When I1 and I2 are stimulated simultaneously, the weight between the bell and saliva neurons undergoes potentiation. As a result, when only the bell neuron is activated, the output neuron responds, demonstrating successful associative learning.

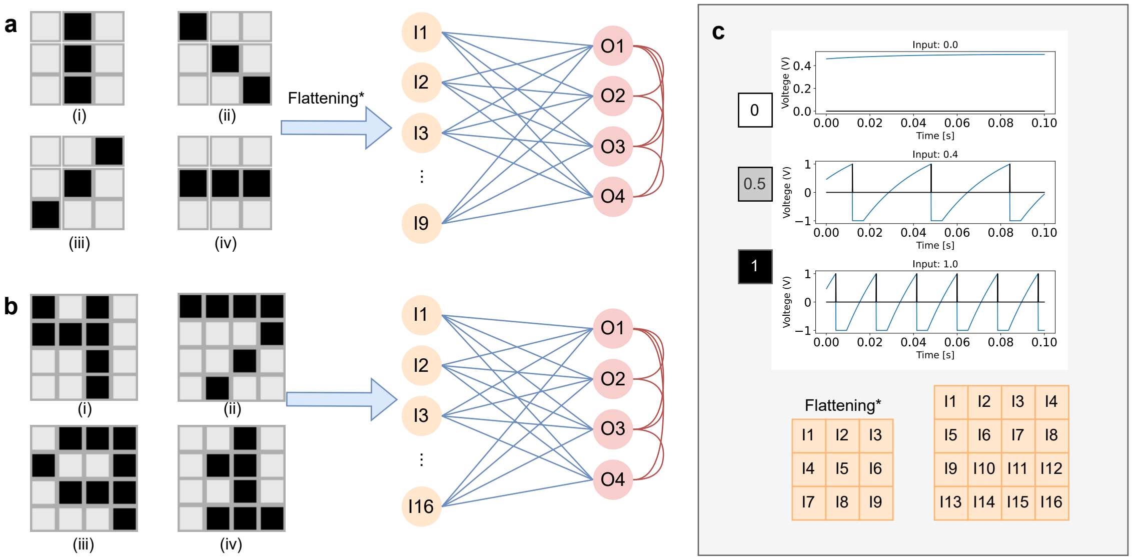


Figure 6.16 Pattern learning demonstration **a** 4 3x3 patterns (i-iv), processed by 9x4 network. **b** 4x4 patterns, to be processed by 16x4 network. **c** Rate coding and flattening.

As a next step, synthetic patterns will be used to evaluate learning in hardware, as depicted in Figure 6.16. Initially, 3x3 patterns (see Figure 6.16), consisting of 4 distinct classes, will be used. Each pixel in the pattern is flattened and mapped to the corresponding input neuron. These patterns consist of black and white pixels, where white represents a value of 0 (no excitation to the neuron, resulting in no spike), and black represents maximum excitation, corresponding to the highest spike frequency (see Figure 6.16c). This setup enables rate encoding, where the intensity of the pixels is converted to spike frequency, simulating neural coding.

After validating the learning behavior with 3x3 patterns, more complex 4x4 patterns will be introduced to assess the system's ability to learn and recognize digits (see Figure 6.16b). Following this, further experiments will be conducted to evaluate the system's long-term learning and adaptation capabilities. This involves continuously training the system over extended periods to observe how well it adapts to changes in input patterns, as well as its ability to retain memory and adjust to new patterns without catastrophic forgetting. Such tests will help determine how well the system can handle dynamic environments, further validating its learning flexibility and robustness.

Name	**Value**	**Name**	**Value**
`index_active_neurons`	[1, 2, 3]	`amp_factor_ltp`	3
`excitation_voltage_min`	1.2	`amp_factor_ltd`	3
`excitation_voltage_max`	3.3	`amp_threshold`	0.6
`excitation_voltage_off`	1.1	`VBL_read`	1.2
`VBL_inference`	1.1	`VSL_read`	1
`duration_neuron_excitation`	10	`VWL_read`	3.3
`duration_monitoring`	1	`VWL_LTP`	3.3
`vmem_index`	1	`VWL_LTD`	3.3
`enable_learning`	`True`	`duration_read`	1e-6
`duration_WTA`	1e-3	`duration_write`	1e-6
`bank`	`OUT`	`duration_gap`	1e-6
`duration_read_vmem`	1e-6		

Table 6.8 User parameters for performing demonstration of online learning.

Several parameters can be configured by the user through the GUI. Table 6.8 outlines the parameters for conducting a learning demonstration, with various voltages and settings already defined in Table 6.4 and Table 6.7.

The `index_active_neurons` parameter specifies the indices of the active neurons, which are set to [1, 2, 3]. The excitation voltage is controlled by `excitation_voltage_min` and `excitation_voltage_max`, set to 1.2V and 3.3V, respectively, while `excitation_voltage_off` defines the voltage when excitation is disabled, set to 1.1V.

During inference, `VBL_inference` is set to 1.1V. The neuron excitation duration, represented by `duration_neuron_excitation`, is set to 10 units, while the monitoring duration, `duration_monitoring`, is set to 1 unit.

The `vmem_index` parameter specifies the memory index used for reading the membrane voltage, with a value of 1. Learning is enabled through the `enable_learning` parameter, which is set to `True`. The duration of the Winner-Takes-All (WTA) phase is 1×10^{-3} seconds, as defined by `duration_WTA`.

The amplification factors for LTP and LTD are controlled by `amp_factor_ltp` and `amp_factor_ltd`, both set to 3, with an amplification threshold `amp_threshold` of 0.6V. The read voltages for the bit line, source line, and word line (`VBL_read`, `VSL_read`, and `VWL_read`) are set to 1.2V, 1V, and 3.3V, respectively, as described in Table 6.7.

During LTP and LTD operations, the word line voltages (VWL_LTP and VWL_LTD) remain at 3.3V. The timing parameters for the read, write, and gap phases (duration_read, duration_write, and duration_gap) are all set to 1×10^{-6}, as is the read duration for the membrane voltage (duration_read_vmem).

6.6.3 Conclusion

This chapter consolidates the algorithms, circuits, and device considerations from the previous sections to develop a mixed-signal, self-learning neural building block. The VDSP algorithm from chapter 3, the scaling factor for mapping memristive device characteristics from chapter 4, and the neuron circuit from chapter 5 are integrated to form the core of the architecture, which supports configurable modes for inference, learning, and device characterization. The analog circuitry consists of input and output neuron banks, interfaced with synaptic reading circuits via voltage regulation and current attenuation. Between these neuron layers is a 16x16 1T1R synaptic crossbar array with BEOL integration, featuring an addressing circuit that uses analog switches and configuration registers to connect individual bit, word, and source lines to either the neuron bank, the VDSP amplifier, or analog I/O pads. The VDSP amplifier transforms the membrane potential of the neurons, applying it to the bit or source line to induce potentiation or depression, thereby enabling efficient synaptic learning.

The full chip consists of the NBB and memory characterization cells, which include TiO_2, HfO_2-based VCM, and HZO-based FTJs. Several 8x8 parallel access and 32x32 (1024) serial access cells enable large-scale demonstrations, statistical modeling, and system simulations through extensive measurements. After BEOL integration of the memories, three experiments are proposed to validate the learning capabilities of the NBB, focusing on associative learning and unsupervised learning with 3x3 and 4x4 pattern sets. In these experiments, pixel intensities are converted into spike rates by the input layer of LIF neurons, while the output neuron bank implements a winner-take-all (WTA) mechanism to make network decisions.

Realizing this hardware implementation of a self-learning building block is essential because many aspects of the system—such as device variability, noise, and interfacing challenges between computing and memory—cannot be fully captured through simulations alone. Accurate modeling of electronic components at scale presents significant challenges, and addressing the inherent variations and issues that arise in real-world scenarios requires physical hardware validation. This step is necessary to ensure reliable performance and functionality in practical applications.

CHAPTER 7

Conclusion

"The important thing is not to stop questioning. Curiosity has its own reason for existing."
– Albert Einstein

TABLE OF CONTENTS

With the proliferation of IoT devices and the rise of big data, vast amounts of data are available for training machine learning algorithms. However, much of this data is unlabeled, limiting the effectiveness of state-of-the-art supervised learning methods. Continual learning through new experiences is arguably a fundamental element behind the superiority of natural intelligence. In this thesis, we investigate the following question: **How can neuromorphic learning principles be translated into analog electronic devices and systems?** We propose that **online learning** is a key component in the overall process of physically implementing artificial intelligence on specialized electronic hardware. Such hardware is necessary for the deployment of efficient AI algorithms on edge devices that operate with limited power budgets and computational resources.

The objectives outlined in chapter 1 are briefly reviewed in relation to the key outcomes of this project in the following sub-sections.

1. To implement a **hardware-friendly local learning algorithm** within the SNN simulation framework, enabling the evaluation of its efficiency for unsupervised pattern classification and benchmarking it against state-of-the-art algorithms like STDP.
2. Outline a **memristive programming strategy** that leverages the analog properties of memristive devices by translating the online learning rule into hardware, supported by **characterization, modeling, and system-level simulations** to benchmark different device technologies, including resistive and ferroelectric devices.
3. Implement **computation circuits** using biomimetic analog **neurons** fabricated in CMOS technology, while also ensuring **interface with the memristive synapse** for impedance matching and spike transmission without altering the memristive state.
4. Develop and validate **mixed-signal circuits** for analog computation, while implementing communication and control via asynchronous digital logic, to achieve a **low-power, real-time SNN prototype**.

7.1 Local learning algorithm

As described in chapter 1, we set out to address key questions about modifying neural learning principles for real-time synaptic weight learning on the hardware. Specifically, our goal was to identify key events (triggers) that initiate the learning process and determine which local variables of the neuron, such as spike timing and membrane potential, define the polarity and magnitude of learning.

Although Hebbian-based approaches like Spike-Timing-Dependent Plasticity (STDP) have demonstrated strong performance, the deployment of them on neuromorphic hardware remains challenging. These challenges are due to the overhead of storing precise spike-timing information or activity traces, as well as the substantial burden posed by peripheral circuitry. To overcome these limitations, we proposed a local unsupervised learning rule: Voltage Dependent Synaptic Plasticity (VDSP), introduced in chapter 3. The algorithm focuses on simplifying hardware implementation while retaining the core principles of unsupervised Hebbian learning. The key to this simplification is to take advantage of the membrane voltage to estimate the timing of the spike, reducing the need to store precise timing information. Through rigorous mathematical analysis and simulations, we demonstrated that the proposed learning rule aligns with Hebb's plasticity principles. This simplification reduces the complexity of on-chip learning circuits, making real-time learning on neuromorphic hardware a more viable option. In [194], we showed that unsupervised learning with the VDSP rule significantly improves recognition rates in pattern classification tasks using simple SNNs, achieving greater accuracy than 90% in handwritten digit recognition. Furthermore, the learning rule demonstrated robustness against injected noise, making it suitable for analog and digital neuromorphic hardware.

This efficiency was further validated in convolutional neural networks (CNN) [243], where the rule proved to be effective for visual and audio pattern learning. The ability to utilize unannotated raw data for training AI algorithms is particularly advantageous, especially given the exponential growth in IoT devices and the recorded signals they generate. As such, VDSP offers a compelling solution for deploying unsupervised learning in state-of-the-art SNN topologies, enabling efficient training on unlabeled data in real-world applications.

7.2 Learning with memristive synapses

Memristors are excellent synaptic devices because of their nanoscale footprint and compatibility with the CMOS process integration. Recently, devices exhibiting multi-level switching within a single memory cell have been proposed. However, two key challenges remain: (i) Translating a learning algorithm into a practical programming strategy requires converting learning signals, such as spike timing in STDP or neuron membrane potential in VDSP, into voltage pulses through circuits; (ii) Unique switching characteristics of memristors, such as non-linearity, asymmetry, and variability, must be accounted for.

In chapter 4, the VDSP learning algorithm was translated into a memristor programming strategy. To achieve this, three devices were examined: TiO2 and HfO2-based Valence Change Memory (VCM) and HfZrO4-based Ferroelectric Tunnel Junction (FTJ). The voltage-dependent switching behavior of these devices was characterized using a dedicated electrical measurement protocol, which was then used to fit a simplified memristor model for system-level simulations. Our results demonstrate the effectiveness of VDSP-driven online learning in both resistive and ferroelectric memristive devices. The learning algorithm showed resilience to variations in key device parameters, such as ON and OFF resistance and switching thresholds. In addition, we proposed strategies to adapt the programming approach based on the known degree of variability, enabling the effective use of stochastic nanoscale devices. Overall, online learning presents a promising method for adapting these nanoscale, ultra-scalable devices to practical neuromorphic applications.

Moreover, unsupervised learning based on VDSP leads to the generation of explainable receptive fields. This method not only furthers the goal of explainable AI, but also improves the robustness of learning against adversarial attacks and the quantization limitations imposed by the restricted resolution of synaptic devices. Our initial results in [282] demonstrate the resilience of this learning method to drift in PCM-based memristors.

7.3 Analog circuits for computing with memristor

We revisit the questions outlined in chapter 1: What circuits and functionalities are required to interact with synaptic devices and generate learning signals, and how much flexibility can be achieved to support various synaptic devices, network architectures (scale/application), and signal time scales?

Integrating memristors into analog neuromorphic circuits presents two key challenges: (i) Interfacing nanodevices exhibiting non-trivial properties, such as non-linearity and variability, with low-power CMOS transistors. The analog nature of information transfer increases the importance of signal integrity in this context. (ii) Implementing circuits that support local learning through optimized circuit elements. The simplicity of this additional circuit block is essential to achieve the scalability promised by memristive technology.

To address these challenges, in chapter 5, we present a versatile CMOS circuit designed to integrate memristive synapses into the signal processing chain of analog neuromorphic systems. A biomimetic LIF neuron was developed, fabricated, and tested, validating three key characteristics: compatibility with memristive synapses, long-term memory retention, and configurability.

First, the neuron demonstrated sensitivity to a wide range of memristive conductance values, confirming its applicability across various memristive technologies and device dimensions. Secondly, a dedicated regulator and current attenuator circuit enabled a reduction in neuron capacitance while maintaining biologically realistic time scales on the order of seconds. This advancement bridges the fields of analog neuromorphic electronics and memristive in-memory computing. Third, the neuron exhibited extreme configurability in parameters such as leak rate and pulse width, covering an order of magnitude. This flexibility allows the proposed circuits to support various SNN topologies and application scenarios. Lastly, we introduced a simple yet innovative connection scheme that enables real-time reconfiguration of an arbitrary subset of LIF neurons into an adaptive variant, maximizing hardware utilization and leveraging established homeostasis (learning) models.

7.4 Mixed-signal in-memory computing and learning architecture

In chapter 6, we outlay the architecture of mixed-signal neural building block for SNN implementation on analog CMOS-RRAM hardware. Digital circuit blocks on the chip enabled configuring between operating modes for (i) serial (row/column) electrical measurement (form/read/write) of 1T1R crossbar cell, (ii) inference through connecting the ouput terminals of the first layer of on-chip LIF neurons to the synaptic reading signal chain of the second layer (composed of Low Dropout regulator (LDO), Current Attenuator (CA), and LIF neurons), and learning mode to implement VDSP based programming through the membrane voltage of the neuron in first layer. The activity of the second layer was scanned through digital shift registers, which operated asynchronously and serially shifted the spike events.

The digital control circuit also enabled modulating the leak rate of analog LIF neurons on chip (16 input + 16 output) to one of three states to (i) freeze the membrane leakage during weight update or memory programming operations, (ii) typical leak rate set by the analog IO bias voltage, or (iii) maximum leak to selectively inhibit set of neurons to implement Winner-Take-All (WTA). Finally to assist learning, stimulation mechanism is built in the output neuron bank, through which the correct output neuron (corresponding to labeled class of presented sample) can be excited externally to accelerate initial learning.

The 1T1R array architecture along with analog switches, and digital registers enables transitioning between different configurations to implement online learning through VDSP. Analog circuit block for amplifying the neuron membrane voltage to levels above the switching threshold is presented. Moreover, mixed-signal circuit for computing polarity of weight update is implemented, and thus the amplified voltage is applied to either top or bottom electrode for potentiation or depression.

A hardware-software design was needed for controlling the NBB. The FPGA and ARM core of Zynq Soc based measurement system was programmed through HDL and firmware modules. The user, through GUI can carry different experiment through modifying different parameters. The Zynq SoC also controls DACs/ADCs on Lotus PCB to supply bias voltages and scan signals from NBB.

Through electrical measurement, the digital control logic on the chip was simulated to verify the chip's configurability. In addition, the transfer characteristics of all 16 neurons were analyzed, highlighting the impact of transistor mismatch and line resistance.

The back-end-of-line integration of the memristor is currently underway, with future work set to demonstrate online learning with fully integrated synapses. This hardware implementation of online learning is an important milestone to reach for the future development of AI applications based on SNNs.

7.5 Summary

Through this thesis, we propose a flexible, algorithm-circuit-based hardware solution for the deployment of small-scale neural networks in EC environments. (i) At the **hardware** level, we developed various circuit blocks with key components that are suitable for memristor-based in-memory computing architectures and neuromorphic computing principles. Ultimately, this work culminates in a mixed-signal CMOS-RRAM neural building block for spiking neural networks (SNNs), which can be interconnected through advanced packaging technologies to form larger networks. (ii) At the **software** level, we created hardware-aware behavioral models of memristive devices and spiking neurons, along with a framework for evaluating the learning efficiency of unsupervised SNNs. This framework was crucial in validating custom online learning rules and memristor programming strategies. Broadly, the objective of this project was to design ultra-low-power hardware for edge computing applications, with the aim of creating a versatile system that could be embedded in a wide range of applications.

The software, models, and firmware developed to reproduce the results would be made available at:

- https://github.com/nikhil-garg
- https://github.com/3it-inpaqt

7.6 Publications

The original contributions to the scientific literature are outlined below.

7.6.1 Primary contributions

1. **N. Garg**, I. Balafrej, T. Stewart, J.-M. Portal, M. Bocquet, D. Querlioz, D. Drouin, J. Rouat, Y. Beilliard, F. Alibart, "Voltage-dependent synaptic plasticity: Unsupervised probabilistic Hebbian plasticity rule based on neurons membrane potential," *Frontiers in Neuroscience*, vol. 16, pp. 983950 (2022) [194].
2. **N. Garg**, I. Balafrej, J. H. Quintino Palhares, L. Bégon-Lours, D. Florini, D. F. Falcone, T. Stecconi, R. Dangel, V. Bragaglia, B. Offrein, J.-M. Portal, D. Querlioz, Y. Beilliard, D. Drouin, F. Alibart, "Unsupervised local learning based on voltage-dependent synaptic plasticity for resistive and ferroelectric synapses," (Submitted to *Nature Communication Materials*).
3. **N. Garg**, D. Florini, P. Dufour, E. Muhr, M. Faye, M. Bocquet, D. Querlioz, Y. Beilliard, D. Drouin, F. Alibart, J.-M. Portal "Versatile CMOS Analog LIF Neuron for Memristor-Integrated Neuromorphic Circuits," *International Conference on Neuromorphic Systems (ICONS)*, 2024 [262].

7.6.2 Related Collaborative Works

1. G. Goupy, A. Juneau-Fecteau, **N. Garg**, I. Balafrej, F. Alibart, L. Frechette, D. Drouin, Y. Beilliard, "Unsupervised and efficient learning in sparsely activated convolutional spiking neural networks enabled by voltage-dependent synaptic plasticity," *Neuromorphic Computing and Engineering*, vol. 3, no. 1, pp. 14001 (2023) [243].
2. J. H. Quintino Palhares, **N. Garg**, P.-A. Mouny, Y. Beilliard, J. Sandrini, F. Arnaud, L. Anghel, F. Alibart, D. Drouin, P. Galy, "28 nm FDSOI embedded PCM exhibiting near zero drift at 12 K for cryogenic SNNs," *npj Unconventional Computing*, vol. 1, no. 1, pp. 8 (2024) [282].
3. K. Janzakova, I. Balafrej, A. Kumar, **N. Garg**, C. Scholaert, J. Rouat, D. Drouin, Y. Coffinier, S. Pecqueur, F. Alibart, "Structural plasticity for neuromorphic networks with electropolymerized dendritic PEDOT connections," *Nature Communications*, vol. 14, no. 1, pp. 8143 (2023) [283].
4. M. Ghazal, A. Kumar, **N. Garg**, S. Pecqueur, F. Alibart, "Neuromorphic Signal Classification using Organic Electrochemical Transistor Array and Spiking Neural Simulations," *IEEE Sensors Journal*, (2024) [284].

7.7 Future works

7.7.1 Multi-core architecture

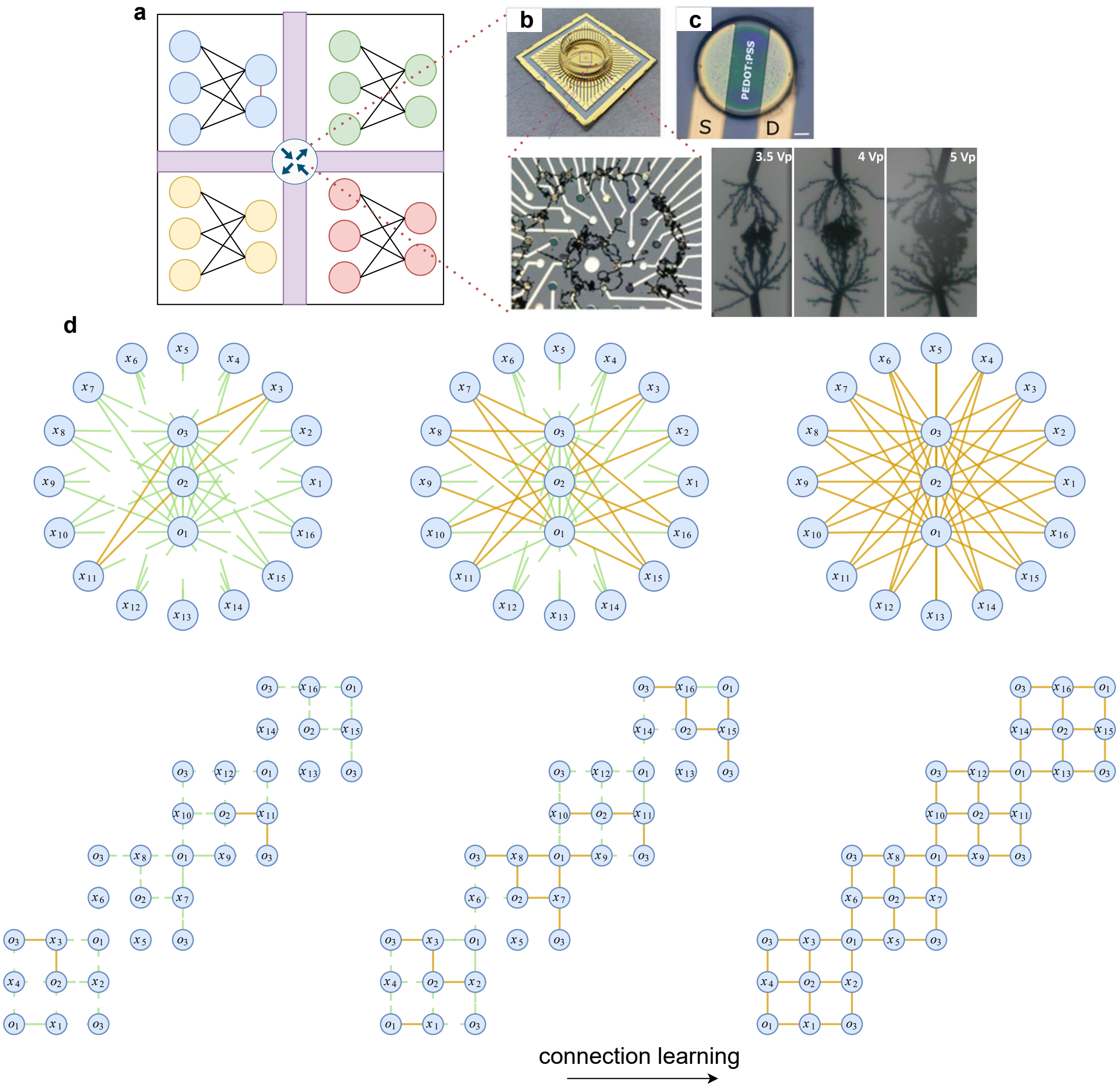


Figure 7.1 Multi-core architecture with plastic interconnects. (Adapted with permission from [283]) **a** Multicore architecture composed of multiple CMOS-RRAM neural building blocks, interconnected by a programmable router. **b** The programmable router consists of an array of Organic Electrochemical Transistor (OECT)s (top). 2D microelectrode arrays with dendrititic connections (bottom). **c** Between the source and drain, there is a PEDOT:PSS polymer (top), which transforms into conducting interconnects upon electrical stimulation (bottom). **d** In a hardware-mapped 3D network, through learning connections with structural plasticity, each node connects to its nearest neighbors.

The first promising avenue for future work is scaling the NBB to multi-core platform. For instance, in Figure 7.1a, a multi-core architecture composed of multiple neural building blocks is illustrated. The blocks are integrated with a programmable router which can potentially be realized using an array of Organic Electrochemical Transistor (OECT) [285] acting as reservoirs (see Figure 7.1b). A key feature of this system is its use of wet-computing with PEDOT polymer-based interconnects, where the strength of the connections is modulated by the history of applied electrical signals (Figure 7.1c). These programmable interconnects enable the system to dynamically adjust through an on-line learning mechanism, where connections between nodes (such as OECTs) can "grow" in response to specific application demands. This flexibility is further enhanced by hardware-programmable interconnects that take advantage of dendritic computing and structural plasticity, allowing the creation of adaptive routing mechanisms [283].

This approach enables the system to independently refine its routing framework in response to learning demands, significantly improving scalability and adaptability across various tasks. Furthermore, memristor-based routers, as investigated in previous studies [286], offer a complementary solution to develop programmable routing networks. These routers further enhance scalability and adaptability, expanding the system's ability to handle diverse tasks. However, the key challenge remains to design an effective routing architecture that seamlessly integrates these technologies and fully leverages their potential.

Our approach involves creating a flexible toolbox of elementary SNNs or neural building blocks, which are designed as a modular Lego-inspired CMOS-RRAM system, to be assembled into application-specific systems. By integrating these NBBs with digital circuits on configurable FPGA logic, the system can scale through advanced packaging techniques such as die-wafer bonding and chiplet-based architectures [287]. These approaches are essential for achieving higher density and performance in neuromorphic systems. Flip-chip integration onto an interposer with high-density interconnects provides a solution to the growing need for compact, scalable hardware platforms that can handle the increasing complexity of applications. However, challenges remain in interconnect design as the systems scale up. The 3D co-integration of different technologies, such as CMOS, memristors, and organic materials, poses significant difficulties. This requires optimizing the process flow to ensure signal integrity and compatibility with standard CMOS manufacturing [287].

7.7.2 Neuron circuit

The current design prioritizes compatibility with memristive synapses and flexibility in parameters such as threshold and leak rate. However, a key issue for future designs is the area and energy overhead of the neuron circuit. To mitigate these concerns, the following optimizations are recommended:

- Integrating a low power comparator circuit within the neuron to reduce dynamic energy expenditure (energy per spike)
- Migrating the design from the 130nm technology node to a more advanced node, such as 7nm.
- Adjusting the power supply level from 3.3V to lower conventional levels, such as 1.2V, to achieve greater energy efficiency

In terms of area overhead, the primary limiting factor is the size of the membrane capacitance, which integrates inputs over time. This capacitor occupies the majority of the neuron area, as a large capacitance is necessary to support slow leakage and bio-plausible temporal dynamics. To address this, future designs could explore replacing the large capacitance with nanodevices such as:

- Materials such as Mott insulators and NbO_2/VO_2 devices exhibit volatile memory that inherently diminishes over time. Such properties enable the substitution of significant capacitors in neuronal circuits, thereby decreasing spatial demands [288, 289, 290].
- Non-volatile resistive [291] or capacitive [292] elements.

The neuron circuit proposed in chapter 5 relies on several DC bias voltages to set its operating parameters. One promising approach for future designs is the use of memristive circuits in a voltage divider configuration to generate these bias voltages [293], eliminating the need for external bias generators. Neuron state variables are also used to generate learning signals for synaptic learning. Additionally, neuron dynamics can be modulated by adjusting parameters such as the leak time constant and threshold. In pure CMOS circuit implementations, this modulation is achieved by altering the bias voltage or current supplied to neurons. However, this strategy faces scalability challenges, as it requires precise voltage control for each neuron. To address this, previous designs have incorporated memristors, which offer extreme scalability and reduce area and power constraints.

Although memristive devices often exhibit variability, this characteristic can be advantageous for capturing a range of spatiotemporal patterns. In a hybrid CMOS-RRAM neuron, the memristor's conductance can be programmed to manage both learning and variability. Known as meta-plasticity, this flexibility permits neuron dynamics to directly shape synaptic learning. Additionally, there is potential to embed plasticity within the homeostasis and adaptation mechanisms of neuron thresholds, as elaborated earlier in chapter 5. The proposed framework redefines the memristive synapse and LIF neuron to function as a regulator, facilitating real-time adjustments to the thresholds or homeostatic properties of other neurons or groups. By employing a memristor to regulate the synaptic resistance of this control neuron, the network can adjust and learn over time, enabling more adaptable runtime modifications. However, incorporating these advanced oxide-based devices into CMOS technology presents a challenging yet rewarding endeavor.

7.7.3 Device engineering and learning models

The UNICO chip, with its more than 4k memories, serves as a test vehicle for extensive device characterization. A key avenue for future work involves material optimization with emerging memristive device concepts, such as HZO ferroelectric tunnel junctions [294], which hold the potential to overcome current limitations in size and energy consumption. However, challenges related to integration and scaling remain, largely due to low current density, and ongoing material research aims to address these issues [295, 296, 297]. The material stack of oxides can be co-optimized through system-level simulations of learning scenarios, with optimization depending on factors like network scale, the nature of the classification problem, and the learning rule. While the first step may involve optimizing device dimensions, more profound improvements may come from rethinking the material stack itself. The UNICO chip offers a unique opportunity to investigate interactions with the copper interconnect layer, going beyond previous work on passive devices. Furthermore, integrating transistors for compliance current control allows exploration of advanced programming strategies, which are crucial for improving device endurance.

In addition to material optimization, the UNICO chip's 1T1R arrays can be leveraged for on-device learning demonstrations with algorithms beyond VDSP. As discussed in chapter 2, various Hebbian learning approaches that have not yet been explored with memristive devices offer promising opportunities. This is crucial because accurately modeling memristive devices, especially at the array scale, is inherently challenging. Although single devices can be effectively modeled with physics-based approaches, scaling these models to evaluate performance with large datasets remains difficult. The proposed VDSP learning rule represents an initial step towards simplifying Hebbian learning for memristive synapses. Building on this fully unsupervised model, a promising next step would be the introduction of a third factor [94]. This third factor, in the form of a reward or surprise signal, could help the system better leverage labeled datasets for faster optimization. It would also enable more efficient learning in complex neural network topologies, such as multi-layer and recurrent networks.

Furthermore, the scaling factor described in chapter 4 is crucial for tuning the programming circuit based on the characterized response of the target memristive device. In chapter 6, the programming circuit employs the ratio of two resistances connected via an Operational Amplifier (OpAmp). These resistances can be replaced with memristor-based programmable devices, which allows for dynamic adjustments. By programming the memristor in the circuit, the VDSP-based learning can be fine-tuned through a learning mechanism, enabling the system to adapt the learning rate in real-time.

7.7.4 Interfacing with biology

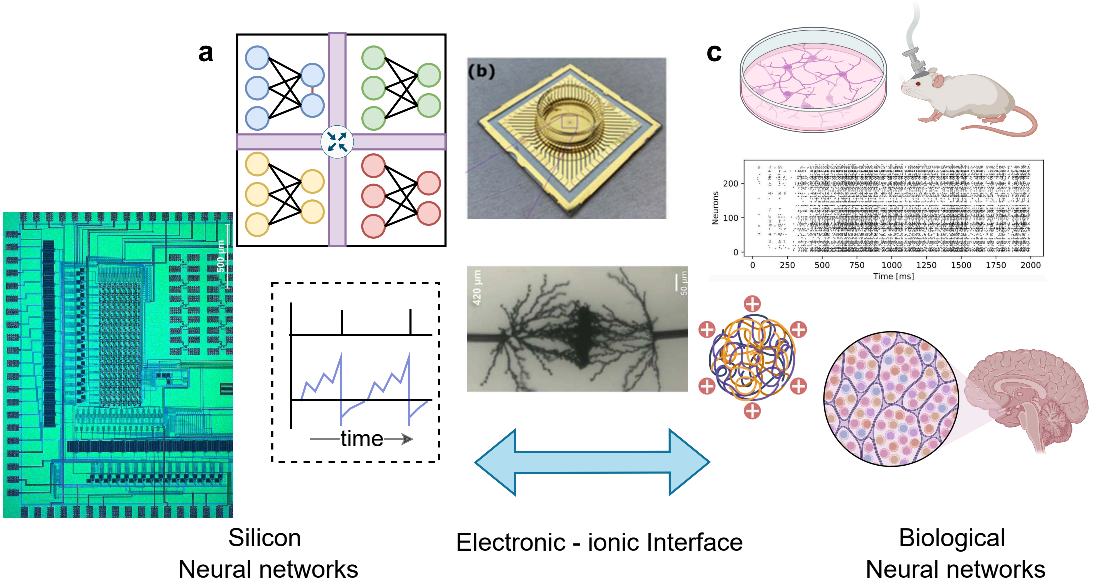


Figure 7.2 Interfacing silicon and biological neural networks. **a** The multi-core architecture, made of CMOS-RRAM NBB, functions as electronic hardware that structurally and functionally emulates bio-mimetic neural networks. **b** Multi-electrode array of Organic Electrochemical Transistor (OECT)s (top) and dendritic polymer for interconnects between nodes in the MEA array (bottom). (Reproduced with permission from [284]) **c** Network of neural cells in a petri dish transmitting information through ion channels. Spike raster illustrating binary activation of neurons. The network could be derived from rodent (rat) or human cells, either in-vivo or in-vitro.

An exciting avenue for further research involves using the created hardware to process signals and identify patterns from biological neurons, with the output of silicon-implemented SNNs potentially serving as stimuli for neural cells and facilitating the exploration of neuromodulation. The hardware developed during this project, as illustrated in Figure 7.2a, offers significant benefits for this type of interface because it closely resembles neural cells.

For example, it mirrors the time scale in processing and the creation of discrete spike events. This interface introduces new opportunities beyond those available in current technologies.

In a recent study, we utilized OECTs [285] as reservoirs [284] to process spike-encoded EMG signals [298]. Future work may involve linking the OECT reservoir to the UNICO chip, facilitating seamless integration for biosignal processing. In fact, OECTs can be co-integrated to develop a unified sensing and processing system, as shown in Figure 7.2b. Additionally, the hardware-friendly VDSP rule can be explored within the framework of structural plasticity to acquire sparse connections [283]. This system could potentially communicate with networks of neural cells, either in-vivo or in-vitro (Figure 7.2c), for applications such as neural prosthetics and brain-computer interfaces. This could potentially improve the adaptability of neuromorphic systems in biological signal processing. Naturally, this involves various challenges related to micro-fabrication and integration of electronic while maintaining compatibility with both semiconductor fabrication processes and neural cells.

A compelling recent study titled "Neuronal Cultures Playing Pong: Initial Steps Toward Advanced Screening and Biological Computing" [299] demonstrated early signs of intelligence in organoids (in vitro neuron cultures). In the long term, integrating biosignal analysis into compact, ultra-low power hardware could revolutionize neural interfaces by enabling localized computing near the recording site. This would minimize heat dissipation, reduce data transfer, and significantly lower power consumption, making the technology more efficient for real-time applications [300]. The relationship between neuroscience and computing fosters a cycle of innovation: brain-inspired computing drives advancements in hardware, while computing-driven brain research uncovers deeper insights into neural processes. Recent breakthroughs in computing hardware have enabled the simulation of large-scale brain models, contributing to our understanding of brain dynamics, degenerative diseases, and potential treatments. Imaging technologies such as Electroencephalography (EEG) and Electrocorticography (ECoG) continue to play a critical role in uncovering the mechanisms of human cognition. These insights, in turn, inform the development of new computing hardware and artificial intelligence, creating a closed feedback loop of symbiotic innovation.

7.7.5 Summary

Our approach focuses on the development of plastic building blocks with self-adaptive capabilities, which address key challenges in edge computing applications. Using unsupervised learning through brain-inspired SNNs, we can efficiently extract features from temporal data, significantly reducing the dependency on large training datasets. Furthermore, our work advances the field of neural networks by adapting plasticity mechanisms and threshold modulation techniques to meet hardware constraints, while fostering new advancements in analog SNNs for next-generation electronic devices. These innovations form the basis for future research on the deployment of SNNs in EC environments. The hybrid hardware/software framework developed through this effort will act as a crucial tool to drive further application-focused advancements and innovation. Below is a summary of potential avenues for future research:

1. Our results are encouraging and should be validated through a larger hardware implementation, ideally with a multi-core architecture. We propose structural plasticity using organic transistors and polyimide-based sparse 3D interconnects as a potential strategy for bottom-up scaling of the proposed neural building block.
2. Future research should explore the integration of memristors directly within the neuron circuit to add a layer of plasticity to key neuron parameters such as the threshold and leak rate. These weights would be non-volatile, reducing the need for analog bias signals for each NBB and thereby improving scalability.
3. More research is required to assess the impact of mismatch and variability in memristive devices, focusing on array-level characterization and robust statistical modeling. Additionally, the integration of innovative material stacks presents a challenging yet promising avenue for future research, offering the potential to improve key performance metrics such as device footprint, current range, and CMOS-compatible programming voltages.
4. An important open question for future research is how to enhance the efficiency of fully unsupervised plasticity, allowing the system to learn and adapt to complex patterns more quickly and accurately. One promising approach could be the incorporation of a third factor, such as a reward or reinforcement signal, to guide the plasticity process. This additional signal could help the network prioritize relevant features, improve the synaptic update mechanism, and ultimately accelerate convergence.
5. The developed hardware mimics the structural and operational principles of biological neural systems, enhancing its capabilities for real-time neural sensing and modulation. Future low-power AI hardware should aim to integrate computing directly into sensors, pushing the boundaries of efficiency.

7.8 Perspective

As AI becomes increasingly integrated into our daily lives, significant challenges arise, particularly in the area of energy efficiency. The deployment of AI at the edge, where devices must operate with minimal power, underscores energy as a critical bottleneck. This problem extends beyond mere technical challenges and has become a worldwide issue, as the carbon emissions from AI datacenters are on par with those of entire countries. Although centralized processing has enabled the wide-scale deployment of AI, it raises serious sustainability concerns. Moreover, the high energy consumption in AI systems leads to heating in semiconductors, which poses challenges to the 3D stackability of electronic devices, a crucial factor for the future of electronics.

The future of computing is unlikely to rely on a single technology. The limitations of purely CMOS-based systems highlight the need for integration of new materials, such as graphene-based 2D materials [301, 302] and organic materials [303] for bio-interfacing. This shift calls for a multidisciplinary approach that combines fabrication, modeling, characterization, simulation, and design to develop the next generation of computing systems. In this context, heterogeneous architectures are expected to drive innovation. Technologies like analog computing, memristors, and FPGAs, long overshadowed by the deterministic scaling of Moore's law, are anticipated to gain prominence as we move beyond this era. Flexible and tailored computing solutions, such as the idea of Lego-like chiplets [304, 305], illustrate this movement.

One promising direction is neuromorphic technology, which utilizes analog circuits and memristive devices. While it remains uncertain whether neuromorphic technology comprising a spiking neural network implemented with analog CMOS and integrated memristive synapses will lead to a breakthrough in machine intelligence, the approach of algorithm-circuit co-design presents a significant opportunity. This approach opens new avenues for emerging fields like quantum computing, photonics, and nanotechnology, all of which have the potential to revolutionize our comprehension and application of artificial intelligence.

A notable advancement in this field of neuromorphic engineering is the concept of online learning, which provides significant benefits. Online learning improves the generalizability of AI systems, enabling them to perform beyond the limitations of their initial training data. Similar to the human brain, these systems are capable of continuously acquiring knowledge and adjusting to new experiences, thus developing with each encountered challenge. Adaptability is especially important for addressing the inherent constraints of hardware, such as fixed topology, limited resolution, analog computing noise, and variability in memristive devices. In the end, I would like to quote Alan Turing, one of the first visionaries to propose the concept of artificial intelligence.

"Instead of trying to produce a programme to simulate the adult mind, why not rather try to produce one which simulates the child's? If this were then subjected to an appropriate course of education one would obtain the adult brain. Presumably the child-brain is something like a note-book as one buys it from the stationers. Rather little mechanism, and lots of blank sheets." - Alan Turing

CHAPTER 8

Conclusion en français

Avec la prolifération des dispositifs IoT et l'essor du big data, d'énormes quantités de données sont disponibles pour l'entraînement des algorithmes d'apprentissage automatique. Cependant, une grande partie de ces données n'est pas étiquetée, limitant l'efficacité des méthodes d'apprentissage supervisé les plus avancées. L'apprentissage continu à travers de nouvelles expériences est sans doute un élément fondamental derrière la supériorité de l'intelligence naturelle. Dans cette thèse, nous examinons la question suivante : **Comment les principes d'apprentissage neuromorphique peuvent-ils être traduits en dispositifs et systèmes électroniques analogiques ?** Nous proposons que l'**apprentissage en ligne** est un composant clé dans le processus global de mise en œuvre physique de l'intelligence artificielle sur un matériel électronique spécialisé. Un tel matériel est nécessaire pour le déploiement d'algorithmes IA efficaces sur des dispositifs périphériques fonctionnant avec des budgets d'énergie et des ressources informatiques limités.

Les objectifs décrits dans chapter 1 sont brièvement révisés en relation avec les résultats clés de ce projet dans les sous-sections suivantes.

1. Mettre en œuvre un **algorithme d'apprentissage local adapté au matériel** dans le cadre de simulation SNN, permettant l'évaluation de son efficacité pour la classification de motifs non supervisés et le comparant à des algorithmes de pointe comme le STDP.
2. Décrire une **stratégie de programmation memristive** qui tire parti des propri étés analogiques des dispositifs memristifs en traduisant la règle d'apprentissage en ligne en matériel, soutenue par **la caractérisation, la modélisation et les simulations au niveau système** pour évaluer différentes technologies de dispositifs, y compris les dispositifs résistifs et ferroélectriques.
3. Implémenter des **circuits de calcul** utilisant des **neurones** biomimétiques analogiques fabriqués en technologie CMOS, tout en assurant également **l'interface avec la synapse memristive** pour une adaptation d'impédance et une transmission de spike sans altérer l'état memristif.

4. Développer et valider des **circuits mixtes** pour le calcul analogique, tout en mettant en œuvre la communication et le contrôle via une logique numérique asynchrone, pour atteindre un prototype de SNN en temps réel à **faible puissance**.

À travers cette thèse, nous proposons une solution matérielle flexible, basée sur des algorithmes-circuits, pour le déploiement de réseaux neuronaux à petite échelle dans des environnements EC. (i) Au niveau du **matériel**, nous avons développé divers blocs de circuits avec des composants clés adaptés aux architectures de calcul en mémoire basées sur les memristors et aux principes de calcul neuromorphique. Finalement, ce travail aboutit à un bloc de construction neural CMOS-RRAM mixte pour les réseaux neuronaux à impulsions (SNNs), qui peut être interconnecté grâce à des technologies d'encapsulation avancées pour former des réseaux plus grands. (ii) Au niveau du **logiciel**, nous avons créé des modèles comportementaux conscients du matériel de dispositifs memristifs et de neurones à impulsions, accompagnés d'un cadre pour évaluer l'efficacité d'apprentissage des SNNs non supervisés. Ce cadre était crucial pour valider les règles d'apprentissage en ligne personnalisées et les stratégies de programmation des memristors. Largement, l'objectif de ce projet était de concevoir un matériel ultra-basse consommation pour des applications de calcul en périphérie, avec l'ambition de créer un système polyvalent pouvant être intégré dans une large gamme d'applications.

8.1 Travaux futurs

Notre approche se concentre sur le développement de blocs de construction plastiques avec des capacités auto-adaptatives, qui répondent aux défis clés dans les applications de calcul en périphérie. En utilisant un apprentissage non supervisé à travers des SNNs inspirés du cerveau, nous pouvons extraire efficacement des caractéristiques à partir de données temporelles, réduisant considérablement la dépendance à de grands ensembles de données d'entraînement. De plus, notre travail fait avancer le domaine des réseaux neuronaux en adaptant les mécanismes de plasticité et les techniques de modulation de seuil pour répondre aux contraintes matérielles, tout en favorisant de nouvelles avancées dans les SNNs analogiques pour les dispositifs électroniques de nouvelle génération. Ces innovations constituent la base de futures recherches sur le déploiement de SNNs dans des environnements EC. Le cadre matériel/logiciel hybride développé à travers cet effort agira comme un outil crucial pour stimuler davantage de progrès orientés vers des applications et des innovations.

1. Nos résultats sont encourageants et devraient être validés à travers une mise en œuvre matérielle plus large, idéalement avec une architecture multicœur. Nous proposons

la plasticité structurelle en utilisant des transistors organiques et des interconnexions 3D sparses basées sur des nanopolyimides comme stratégie potentielle pour le dimensionnement ascendant du bloc de construction neuronal proposé.

2. Les recherches futures devraient explorer l'intégration de memristors directement au sein du circuit neuronique pour ajouter une couche de plasticité aux paramètres clés du neurone tels que le seuil et le taux de fuite. Ces poids seraient non-volatils, réduisant la nécessité de signaux de polarisation analogiques pour chaque NBB et améliorant ainsi l'évolutivité.
3. Il est nécessaire de mener davantage de recherches pour évaluer l'impact des discordances et de la variabilité dans les dispositifs memristifs, en se concentrant sur la caractérisation au niveau des réseaux et la modélisation statistique robuste. De plus, l'intégration de piles de matériaux innovantes présente une voie difficile mais prometteuse pour de futures recherches, offrant le potentiel d'améliorer des métriques de performance clés telles que l'empreinte du dispositif, la plage de courant et les tensions de programmation compatibles avec les CMOS.
4. Une question importante pour les recherches futures est comment améliorer l'efficacité de la plasticité totalement non supervisée, permettant au système d'apprendre et de s'adapter à des motifs complexes plus rapidement et avec plus de précision. Une approche prometteuse pourrait être l'incorporation d'un troisième facteur, tel qu'un signal de récompense ou de renforcement, pour guider le processus de plasticité. Ce signal supplémentaire pourrait aider le réseau à prioriser les caractéristiques pertinentes, améliorer le mécanisme de mise à jour synaptique, et accélérer ultimement la convergence.
5. Le matériel développé imite les principes structurels et opérationnels des systèmes neuronaux biologiques, améliorant ses capacités pour la détection et la modulation neurales en temps réel. Le futur matériel IA à faible puissance devrait viser à intégrer le calcul directement dans les capteurs, repoussant les limites de l'efficacité.

APPENDIX A
UNICO ASIC

Characterization cells

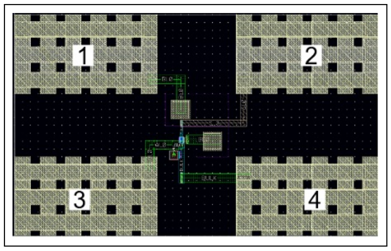


Figure A.1 Single 1T1R cell composed of 4 characterization pads: gate, source, top-electrode, and bulk of transistor.

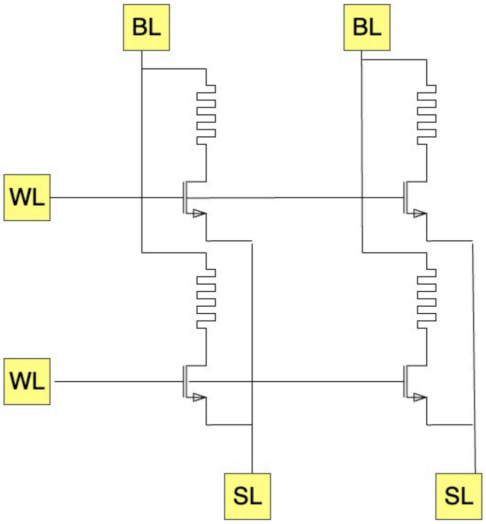


Figure A.2 2x2 representation of 8x8 parallel memory characterization array

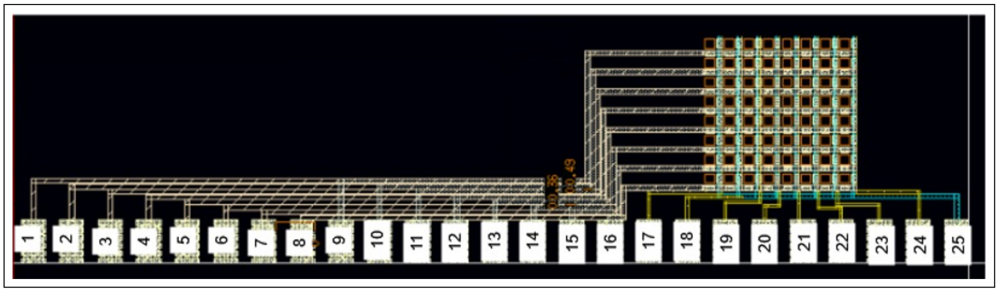


Figure A.3 8x8 parallel memory characterization cell with parallel access of BL, WL, SL through 25 scribes.

Pin No.	Name	Pin No.	Name
1	BL<7>	14	SL<2>
2	BL<6>	15	SL<1>
3	BL<5>	16	SL<0>
4	BL<4>	17	WL<0>
5	BL<3>	18	WL<1>
6	BL<2>	19	WL<2>
7	BL<1>	20	WL<3>
8	BL<0>	21	WL<4>
9	SL<7>	22	WL<5>
10	SL<6>	23	WL<6>
11	SL<5>	24	WL<7>
12	SL<4>	25	gnd
13	SL<3>		

Table A.1 8x8 Parallel characterization cell: Pin Number and Name.

Table A.1 provides the pin assignments for the parallel interface, listing the bit line (BL), source line (SL), and word line (WL) connections. Pins 1 through 8 are associated with the bit lines BL<0> to BL<7>, while pins 9 through 16 correspond to the source lines SL<0> to SL<7>. The word line connections are covered by pins 17 through 24, which represent WL<0> to WL<7>.

Pin 25 for ground (gnd), shared by the bulk terminals of all transistors of the 8x8 1T1R array.

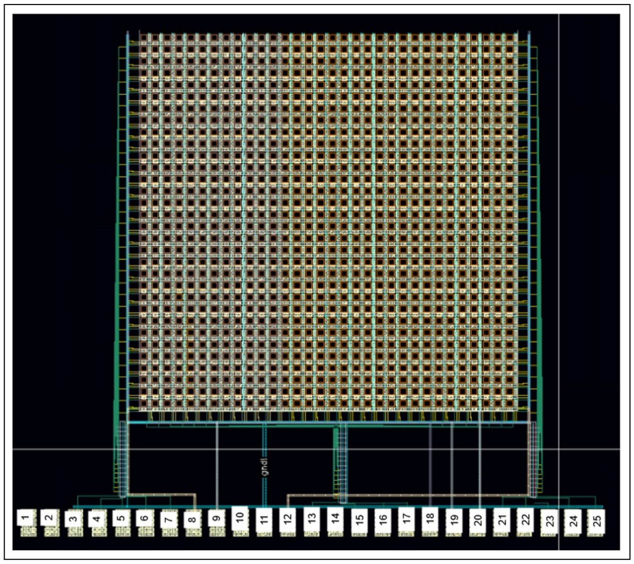


Figure A.4 32x32 memory characterization cell with serial acess through 25 analog and digital IO signals.

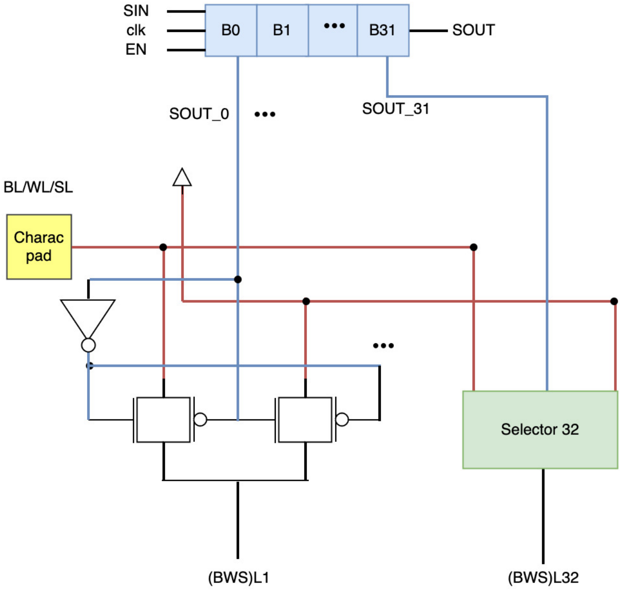


Figure A.5 Shift register for addressing a single row or column of 32x32 crossbar array

Pin No.	**Name**	**A/D/P**	**I/O**
1	-	-	-
2	-	-	-
3	IN_SR_LS_BL<31>	D	O
4	SIN_BL	D	I
5	EN_SR_BL	D	I
6	RESET_BL	D	I
7	CLK_BL	D	I
8	VBL	A	IO
9	EN_LS	D	I
10	-		-
11	gnd	P	I
12	VSL	A	IO
13	IN_SR_LS_WL<31>	D	O
14	SIN_WL	D	I
15	EN_SR_WL	D	I
16	RESET_WL	D	I
17	CLK_WL	D	I
18	VDDL	P	I
19	VWL	A	IO
20	VDDH	P	I
21	CLK_SL	D	I
22	RESET_SL	D	I
23	SIN_SL	D	I
24	EN_SR_SL	D	I
25	IN_SR_LS_SL<31>	D	IO

Table A.2 HZO Serial: Pin Number, Name, A/D/P, and I/O

Table A.2 details the pin assignments for the HZO serial interface, specifying the pin numbers, names, analog/digital/power (`A/D/P`) classification, and input/output (`I/O`) functionality.

Most of the pins are used for digital signal control, such as `SIN_BL`, `EN_SR_BL`, `RESET_BL`, and `CLK_BL` for bit line management, and similarly for the word line and source line control with corresponding pins like `SIN_WL`, `SIN_SL`, and their associated enable (`EN_SR`), reset (`RESET`), and clock (`CLK`) signals.

Pin 8 (`VBL`), pin 12 (`VSL`), and pin 19 (`VWL`) are designated for analog input/output, controlling the bit line, source line, and word line voltages, respectively. Pins 11 (`gnd`), 18 (`VDDL`), and 20 (`VDDH`) are power-related and are used to supply ground and operating voltages.

Several pins, such as pin 3 (`IN_SR_LS_BL<31>`) and pin 25 (`IN_SR_LS_SL<31>`), handle data output for the least significant bit of the shift registers associated with the bit line and source line, respectively. Pins 1, 2, and 10 are unused and marked with a dash.

LIST OF REFERENCES

[1] Joel Hartmann, Paolo Cappelletti, Nitin Chawla, Franck Arnaud, and Andreia Cathelin. Artificial Intelligence: Why moving it to the Edge. *European Solid-State Device Research Conference*, 2021-September:1–6, 2021.

[2] Weisong Shi, Jie Cao, Quan Zhang, Youhuizi Li, and Lanyu Xu. Edge computing: Vision and challenges. *IEEE Internet of Things Journal*, 3(5):637–646, 2016.

[3] Peter Lennie. The cost of cortical computation. *Current biology*, 13(6):493–497, 2003.

[4] Mark Horowitz. Computing's energy problem (and what we can do about it). In *Digest of Technical Papers - IEEE International Solid-State Circuits Conference*, volume 57, pages 10–14. IEEE, 2014.

[5] Bernard Widrow and Rodney Winter. Neural nets for adaptive filtering and adaptive pattern recognition. *Computer*, 21(3):25–39, 1988.

[6] Frank Rosenblatt. Principles of neurodynamics. perceptrons and the theory of brain mechanisms. Technical report, Cornell Aeronautical Lab Inc Buffalo NY, 1961.

[7] Bernard Widrow, RG Winter, and Robert A Baxter. Learning phenomena in layered neural networks. In *Proceedings of the IEEE First International Conference on Neural Networks*, volume 2, pages 411–430, 1987.

[8] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[9] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

[10] Lawrence Stark, Mitsuharu Okajima, and Gerald H Whipple. Computer pattern recognition techniques: electrocardiographic diagnosis. *Communications of the ACM*, 5(10):527–531, 1962.

[11] Emma Strubell, Ananya Ganesh, and Andrew McCallum. Energy and policy considerations for modern deep learning research. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34-09, pages 13693–13696, 2020.

[12] David Mytton and Masaō Ashtine. Sources of data center energy estimates: A comprehensive review. *Joule*, 6(9):2032–2056, sep 2022.

[13] Carver Mead. Neuromorphic electronic systems. *Proceedings of the IEEE*, 78(10):1629–1636, 1990.

[14] Carver Mead. How we created neuromorphic engineering. *Nature Electronics*, 3(7):434–435, 2020.

[15] Danijela Marković, Alice Mizrahi, Damien Querlioz, and Julie Grollier. Physics for neuromorphic computing. *Nature Reviews Physics*, 2(9):499–510, 2020.

[16] Mario Lanza, Abu Sebastian, Wei D. Lu, Manuel Le Gallo, Meng-Fan Chang, Deji Akinwande, Francesco M. Puglisi, Husam N. Alshareef, Ming Liu, and Juan B. Roldan. Memristive technologies for data storage, computation, encryption, and radio-frequency communication. *Science*, 376(6597), jun 2022.

[17] Indranil Chakraborty, Mustafa Ali, Aayush Ankit, Shubham Jain, Sourjya Roy, Shrihari Sridharan, Amogh Agrawal, Anand Raghunathan, and Kaushik Roy. Resistive Crossbars as Approximate Hardware Building Blocks for Machine Learning: Opportunities and Challenges. *Proceedings of the IEEE*, 108(12):2276–2310, 2020.

[18] Saion K. Roy, Ameya Patil, and Naresh R. Shanbhag. Fundamental Limits on the Computational Accuracy of Resistive Crossbar-based In-memory Architectures. *Proceedings - IEEE International Symposium on Circuits and Systems*, 2022-May:384–388, 2022.

[19] Fabien Alibart, Ligang Gao, Brian D Hoskins, and Dmitri B Strukov. High precision tuning of state for memristive devices by adaptable variation-tolerant algorithm. *Nanotechnology*, 23(7):075201, 2012.

[20] Abdelouadoud El Mesoudy, Gwénaëlle Lamri, Raphaël Dawant, Javier Arias-Zapata, Pierre Gliech, Yann Beilliard, Serge Ecoffey, Andreas Ruediger, Fabien Alibart, and Dominique Drouin. Fully cmos-compatible passive tio2-based memristor crossbars for in-memory computing. *Microelectronic Engineering*, 255:111706, 2022.

[21] Luciano Floridi and Massimo Chiriatti. Gpt-3: Its nature, scope, limits, and consequences. *Minds and Machines*, 30:681–694, 2020.

[22] Claudine Badue, Rânik Guidolini, Raphael Vivacqua Carneiro, Pedro Azevedo, Vinicius B Cardoso, Avelino Forechi, Luan Jesus, Rodrigo Berriel, Thiago M Paixao, Filipe Mutz, et al. Self-driving cars: A survey. *Expert systems with applications*, 165:113816, 2021.

[23] Laurent Fiorina, Pascale Chemaly, Joffrey Cellier, Mina Ait Said, Charlène Coquard, Salem Younsi, Fiorella Salerno, Jérôme Horvilleur, Jérôme Lacotte, Vladimir Manenti, et al. Artificial intelligence-based ecg analysis improves atrial arrhythmia detection from a smartwatch ecg. *European Heart Journal-Digital Health*, page ztae047, 2024.

[24] Herman H Goldstine and Adele Goldstine. The electronic numerical integrator and computer (eniac). In *The Origins of Digital Computers: Selected Papers*, pages 359–373. Springer, 1946.

[25] John Von Neumann. First draft of a report on the edvac. *IEEE Annals of the History of Computing*, 15(4):27–75, 1993.

[26] SE Gluck. The electronic discrete variable computer. *Electrical Engineering*, 72(2):159–162, 1953.

[27] Simon H. Lavington. The manchester mark i and atlas: a historical perspective. *Communications of the ACM*, 21(1):4–12, 1978.

[28] William Aspray. The intel 4004 microprocessor: What constituted invention? *IEEE Annals of the History of Computing*, 19(3):4–15, 1997.

[29] Siegfried Wendt. Functional description of the integrated processor circuit intel 8080. *Euromicro Newsletter*, 2(1):30–37, 1976.

[30] F Robert A Hopgood, Roger J Hubbold, and David Duce. *Advances in computer graphics II*. Springer Science & Business Media, 1986.

[31] Norman P Jouppi, Cliff Young, Nishant Patil, David Patterson, Gaurav Agrawal, Raminder Bajwa, Sarah Bates, Suresh Bhatia, Nan Boden, Al Borchers, et al. In-

datacenter performance analysis of a tensor processing unit. In *Proceedings of the 44th annual international symposium on computer architecture*, pages 1–12, 2017.

[32] Juan A De Carlos and José Borrell. A historical reflection of the contributions of cajal and golgi to the foundations of neuroscience. *Brain research reviews*, 55(1):8–16, 2007.

[33] Alan L Hodgkin and Andrew F Huxley. Action potentials recorded from inside a nerve fibre. *Nature*, 144(3651):710–711, 1939.

[34] Donald Olding Hebb. *The organization of behavior: A neuropsychological theory*. Psychology press, 2005.

[35] Tim VP Bliss and Terje Lømo. Long-lasting potentiation of synaptic transmission in the dentate area of the anaesthetized rabbit following stimulation of the perforant path. *The Journal of physiology*, 232(2):331–356, 1973.

[36] Guo-qiang Bi and Mu-ming Poo. Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and postsynaptic cell type. *Journal of neuroscience*, 18(24):10464–10472, 1998.

[37] Flora Vasile, Elena Dossi, and Nathalie Rouach. Human astrocytes: structure and functions in the healthy brain. *Brain Structure and Function*, 222(5):2017–2029, 2017.

[38] John Ambrose Fleming. Instrument for converting alternating electric currents into continuous currents., November 7 1905. US Patent 803,684.

[39] Lilienfeld Julius Edgar. Method and apparatus for controlling electric currents, January 28 1930. US Patent 1,745,175.

[40] John Bardeen and Walter Hauser Brattain. The transistor, a semi-conductor triode. *Physical Review*, 74(2):230, 1948.

[41] Martin M Atalla, Eileen Tannenbaum, and EJ Scheibner. Stabilization of silicon surfaces by thermally grown oxides. *Bell System Technical Journal*, 38(3):749–783, 1959.

[42] Digh Hisamoto, Wen-Chin Lee, Jakub Kedzierski, Erik Anderson, Hideki Takeuchi, Kazuya Asano, Tsu-Jae King, Jeffrey Bokor, and Chenming Hu. A folded-channel mosfet for deep-sub-tenth micron era. *IEDM Tech. Dig*, 1998:1032–1034, 1998.

[43] H-SP Wong. Beyond the conventional transistor. *IBM Journal of Research and Development*, 46(2.3):133–168, 2002.

[44] B Yang, KD Buddharaju, SHG Teo, N Singh, GQ Lo, and DL Kwong. Vertical silicon-nanowire formation and gate-all-around mosfet. *IEEE Electron Device Letters*, 29(7):791–794, 2008.

[45] Chris Auth, C Allen, A Blattner, D Bergstrom, M Brazier, M Bost, M Buehler, V Chikarmane, T Ghani, T Glassman, et al. A 22nm high performance and low-power cmos technology featuring fully-depleted tri-gate transistors, self-aligned contacts and high density mim capacitors. In *2012 symposium on VLSI technology (VLSIT)*, pages 131–132. IEEE, 2012.

[46] Patrick Lichtsteiner, Christoph Posch, and Tobi Delbruck. A 128× 128 120 db 15 μs latency asynchronous temporal contrast vision sensor. *IEEE Journal of Solid-State Circuits*, 43(2):566–576, 2008.

[47] Shih-Chii Liu, André van Schaik, Bradley A. Minch, and Tobi Delbruck. Asynchronous binaural spatial audition sensor with $2 \times 64 \times 4$ channel output. *IEEE Transactions on Biomedical Circuits and Systems*, 8(4):453–464, 2014.

[48] Larry F Abbott. Lapicque's introduction of the integrate-and-fire model neuron (1907). *Brain research bulletin*, 50(5-6):303–304, 1999.

[49] Eugene M Izhikevich. Which model to use for cortical spiking neurons? *IEEE transactions on neural networks*, 15(5):1063–1070, 2004.

[50] Guillaume Bellec, Franz Scherr, Anand Subramoney, Elias Hajek, Darjan Salaj, Robert Legenstein, and Wolfgang Maass. A solution to the learning dilemma for recurrent networks of spiking neurons. *Nature communications*, 11(1):1–15, 2020.

[51] James M Bower and David Beeman. *The book of GENESIS: exploring realistic neural models with the GEneral NEural SImulation System.* Springer Science & Business Media, 2012.

[52] Eoin P Lynch and Conor J Houghton. Parameter estimation of neuron models using in-vitro and in-vivo electrophysiological data. *Frontiers in neuroinformatics*, 9:10, 2015.

[53] Cyrille Rossant, Dan FM Goodman, Jonathan Platkiewicz, and Romain Brette. Automatic fitting of spiking neuron models to electrophysiological recordings. *Frontiers in neuroinformatics*, 4:1273, 2010.

[54] Wulfram Gerstner, Marco Lehmann, Vasiliki Liakoni, Dane Corneil, and Johanni Brea. Eligibility traces and plasticity on behavioral time scales: experimental support of neohebbian three-factor learning rules. *Frontiers in neural circuits*, 12:53, 2018.

[55] Mark F Bear. A synaptic basis for memory storage in the cerebral cortex. *Proceedings of the National Academy of Sciences*, 93(24):13453–13459, 1996.

[56] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal processing magazine*, 29(6):82–97, 2012.

[57] Darjan Salaj, Anand Subramoney, Ceca Kraisnikovic, Guillaume Bellec, Robert Legenstein, and Wolfgang Maass. Spike frequency adaptation supports network computations on temporally dispersed information. *eLife*, 10:e65459, jul 2021.

[58] Stephen Grossberg. Adaptive resonance theory: How a brain learns to consciously attend, learn, and recognize a changing world. *Neural networks*, 37:1–47, 2013.

[59] Stefano Ambrogio, Pritish Narayanan, Hsinyu Tsai, Robert M Shelby, Irem Boybat, Carmelo Di Nolfo, Severin Sidler, Massimo Giordano, Martina Bodini, Nathan CP Farinha, et al. Equivalent-accuracy accelerated neural-network training using analogue memory. *Nature*, 558(7708):60–67, 2018.

[60] Stuart J Russell and Peter Norvig. *Artificial intelligence: a modern approach.* Pearson, 2016.

[61] Geoffrey Hinton and Terrence J Sejnowski. *Unsupervised learning: foundations of neural computation.* MIT press, 1999.

[62] Leslie Pack Kaelbling, Michael L Littman, and Andrew W Moore. Reinforcement learning: A survey. *Journal of artificial intelligence research*, 4:237–285, 1996.

[63] Thomas Frank, K-F Kraiss, and Torsten Kuhlen. Comparative analysis of fuzzy art and art-2a network clustering performance. *IEEE Transactions on Neural networks*, 9(3):544–559, 1998.

[64] Tapas Kanungo, David M Mount, Nathan S Netanyahu, Christine D Piatko, Ruth Silverman, and Angela Y Wu. An efficient k-means clustering algorithm: Analysis and implementation. *IEEE transactions on pattern analysis and machine intelligence*, 24(7):881–892, 2002.

[65] Harold Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, 24(6):417, 1933.

[66] Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52, 1987.

[67] Pierre Comon. Independent component analysis, a new concept? *Signal processing*, 36(3):287–314, 1994.

[68] Christopher J Rozell, Don H Johnson, Richard G Baraniuk, and Bruno A Olshausen. Sparse coding via thresholding and local competition in neural circuits. *Neural computation*, 20(10):2526–2563, 2008.

[69] John J Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, 79(8):2554–2558, 1982.

[70] Stephen G Brush. History of the lenz-ising model. *Reviews of modern physics*, 39(4):883, 1967.

[71] David H Ackley, Geoffrey E Hinton, and Terrence J Sejnowski. A learning algorithm for boltzmann machines. *Cognitive science*, 9(1):147–169, 1985.

[72] Yoshua Bengio, Pascal Lamblin, Dan Popovici, and Hugo Larochelle. Greedy layer-wise training of deep networks. *Advances in neural information processing systems*, 19, 2006.

[73] Peter Dayan, Geoffrey E Hinton, Radford M Neal, and Richard S Zemel. The helmholtz machine. *Neural computation*, 7(5):889–904, 1995.

[74] DP Kingma. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[75] Geoffrey E Hinton, Peter Dayan, Brendan J Frey, and Radford M Neal. The" wake-sleep" algorithm for unsupervised neural networks. *Science*, 268(5214):1158–1161, 1995.

[76] Teuvo Kohonen. Self-organized formation of topologically correct feature maps. *Biological cybernetics*, 43(1):59–69, 1982.

[77] Melika Payvand, Filippo Moro, Kumiko Nomura, Thomas Dalgaty, Elisa Vianello, Yoshifumi Nishi, and Giacomo Indiveri. Self-organization of an inhomogeneous memristive hardware for sequence learning. *Nature communications*, 13(1):1–12, 2022.

[78] Lyes Khacef, Philipp Klein, Matteo Cartiglia, Arianna Rubino, Giacomo Indiveri, and Elisabetta Chicca. Spike-based local synaptic plasticity: A survey of computational models and neuromorphic circuits. *arXiv preprint arXiv:2209.15536*, 2022.

[79] Sen Song, Kenneth D Miller, and Larry F Abbott. Competitive hebbian learning through spike-timing-dependent synaptic plasticity. *Nature neuroscience*, 3(9):919–926, 2000.

[80] Jean-Pascal Pfister and Wulfram Gerstner. Triplets of spikes in a model of spike timing-dependent plasticity. *Journal of Neuroscience*, 26(38):9673–9682, 2006.

[81] Joseph M Brader, Walter Senn, and Stefano Fusi. Learning real-world stimuli in a neural network with spike-driven synaptic dynamics. *Neural computation*, 19(11):2881–2912, 2007.

[82] Claudia Clopath, Lars Büsing, Eleni Vasilaki, and Wulfram Gerstner. Connectivity reflects coding: a model of voltage-based stdp with homeostasis. *Nature neuroscience*, 13(3):344–352, 2010.

[83] Michael Graupner and Nicolas Brunel. Calcium-based plasticity model explains sensitivity of synaptic changes to spike pattern, rate, and dendritic location. *Proceedings of the National Academy of Sciences*, 109(10):3991–3996, 2012.

[84] Trevor Bekolay, Carter Kolbeck, and Chris Eliasmith. Simultaneous unsupervised and supervised learning of cognitive functions in biologically plausible spiking neural networks. In *Proceedings of the annual meeting of the cognitive science society*, volume 35, 2013.

[85] Pierre Yger and Kenneth D Harris. The convallis rule for unsupervised learning in cortical networks. *PLoS Computational Biology*, 9(10):e1003272, 2013.

[86] Robert Urbanczik and Walter Senn. Learning by the dendritic prediction of somatic spiking. *Neuron*, 81(3):521–528, 2014.

[87] Peter U Diehl and Matthew Cook. Unsupervised learning of digit recognition using spike-timing-dependent plasticity. *Frontiers in computational neuroscience*, 9:99, 2015.

[88] Christian Albers, Maren Westkott, and Klaus Pawelzik. Learning of precise spike times with homeostatic membrane potential dependent synaptic plasticity. *PloS one*, 11(2):e0148948, 2016.

[89] Sadique Sheik, Somnath Paul, Charles Augustine, and Gert Cauwenberghs. Membrane-dependent neuromorphic learning rule for unsupervised spike pattern detection. In *2016 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, pages 164–167. IEEE, 2016.

[90] Alexandre Payeur, Jordan Guerguiev, Friedemann Zenke, Blake A Richards, and Richard Naud. Burst-dependent synaptic plasticity can coordinate learning in hierarchical circuits. *Nature neuroscience*, 24(7):1010–1019, 2021.

[91] Verena Pawlak, Jeffery R Wickens, Alfredo Kirkwood, and Jason ND Kerr. Timing is not everything: neuromodulation opens the stdp gate. *Frontiers in synaptic neuroscience*, 2:146, 2010.

[92] Sara Zannone, Zuzanna Brzosko, Ole Paulsen, and Claudia Clopath. Acetylcholine-modulated plasticity in reward-driven navigation: a computational study. *Scientific reports*, 8(1):1–20, 2018.

[93] Matthijs B Verhoog and Huibert D Mansvelder. Presynaptic ionotropic receptors controlling and modulating the rules for spike timing-dependent plasticity. *Neural plasticity*, 2011, 2011.

[94] Nicolas Frémaux and Wulfram Gerstner. Neuromodulated spike-timing-dependent plasticity, and theory of three-factor learning rules. *Frontiers in neural circuits*, 9:85, 2016.

[95] Nicolas Frémaux, Henning Sprekeler, and Wulfram Gerstner. Functional requirements for reward-modulated spike-timing-dependent plasticity. *Journal of Neuroscience*, 30(40):13326–13337, 2010.

[96] Adria Bofill, D Thompson, and Alan Murray. Citcuits for vlsi implementation of temporally asymmetric hebbian learning. *Advances in Neural Information processing systems*, 14, 2001.

[97] Giacomo Indiveri. Neuromorphic bisable vlsi synapses with spike-timing-dependent plasticity. *Advances in neural information processing systems*, 15, 2002.

[98] Adria Bofill-i Petit and Alan F Murray. Synchrony detection and amplification by silicon neurons with stdp synapses. *IEEE Transactions on neural networks*, 15(5):1296–1304, 2004.

[99] Katherine Cameron, Vasin Boonsobhak, Alan Murray, and David Renshaw. Spike timing dependent plasticity (stdp) can ameliorate process variations in neuromorphic vlsi. *IEEE Transactions on Neural Networks*, 16(6):1626–1637, 2005.

[100] G. Indiveri, E. Chicca, and R. Douglas. A vlsi array of low-power spiking neurons and bistable synapses with spike-timing dependent plasticity. *IEEE Transactions on Neural Networks*, 17(1):211–221, 2006.

[101] John V Arthur and Kwabena Boahen. Learning in silicon: Timing is everything. *Advances in neural information processing systems*, 18, 2005.

[102] Thomas Jacob Koickal, Alister Hamilton, Su Lim Tan, James A Covington, Julian W Gardner, and Tim C Pearce. Analog vlsi circuit implementation of an adaptive neuromorphic olfaction chip. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 54(1):60–73, 2007.

[103] Shih-Chii Liu and Rico Mockel. Temporally learning floating-gate vlsi synapses. In *2008 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 2154–2157. IEEE, 2008.

[104] Hideki Tanaka, Takashi Morie, and Kazuyuki Aihara. A cmos spiking neural network circuit with symmetric/asymmetric stdp function. *IEICE transactions on fundamentals of electronics, communications and computer sciences*, 92(7):1690–1698, 2009.

[105] Simeon A Bamford, Alan F Murray, and David J Willshaw. Spike-timing-dependent plasticity with weight dependence evoked from physical constraints. *IEEE Transactions on Biomedical Circuits and Systems*, 6(4):385–398, 2012.

[106] Roshan Gopalakrishnan and Arindam Basu. Robust doublet stdp in a floating-gate synapse. In *2014 International Joint Conference on Neural Networks (IJCNN)*, pages 4296–4301. IEEE, 2014.

[107] Michele Mastella, Fabio Toso, Giuseppe Sciortino, Enrico Prati, and Giorgio Ferrari. Tunneling-based cmos floating gate synapse for low power spike timing dependent plasticity. In *2020 2nd IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS)*, pages 213–217. IEEE, 2020.

[108] Christian Mayr, Marko Noack, Johannes Partzsch, and René Schüffny. Replicating experimental spike and rate based neural learning in cmos. In *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*, pages 105–108. IEEE, 2010.

[109] Mostafa Rahimi Azghadi, Said Al-Sarawi, Derek Abbott, and Nicolangelo Iannella. A neuromorphic vlsi design for spike timing and rate based synaptic plasticity. *Neural Networks*, 45:70–82, 2013.

[110] Roshan Gopalakrishnan and Arindam Basu. Triplet spike time-dependent plasticity in a floating-gate synapse. *IEEE Transactions on Neural Networks and Learning Systems*, 28(4):778–790, 2015.

[111] Stefano Fusi, Mario Annunziato, Davide Badoni, Andrea Salamon, and Daniel J Amit. Spike-driven synaptic plasticity: theory, simulation, vlsi implementation. *Neural computation*, 12(10):2227–2258, 2000.

[112] Elisabetta Chicca and S Fusi. Stochastic synaptic plasticity in deterministic avlsi networks of spiking neurons. In *Proceedings of the World Congress on Neuroinformatics*, pages 468–477. Citeseer, 2001.

[113] Elisabetta Chicca, Davide Badoni, Vittorio Dante, Massimo D'Andreagiovanni, Gaetano Salina, Luciana Carota, Stefano Fusi, and Paolo Del Giudice. A vlsi recurrent network of integrate-and-fire neurons connected by plastic synapses with long-term memory. *IEEE Transactions on neural networks*, 14(5):1297–1307, 2003.

[114] Massimiliano Giulioni, Patrick Camilleri, Vittorio Dante, Davide Badoni, Giacomo Indiveri, Jochen Braun, and Paolo Del Giudice. A vlsi network of spiking neurons with plastic fully configurable "stop-learning" synapses. In *2008 15th IEEE International Conference on Electronics, Circuits and Systems*, pages 678–681. IEEE, 2008.

[115] Srinjoy Mitra, Stefano Fusi, and Giacomo Indiveri. Real-time classification of complex patterns using spike-based learning in neuromorphic vlsi. *IEEE transactions on biomedical circuits and systems*, 3(1):32–42, 2008.

[116] E. Chicca, Fabio Stefanini, C. Bartolozzi, and G. Indiveri. Neuromorphic electronic circuits for building autonomous cognitive systems. *Proceedings of the IEEE*, 102:1367–1388, 2014.

[117] Frank L Maldonado Huayaney, Stephen Nease, and Elisabetta Chicca. Learning in silicon beyond stdp: a neuromorphic implementation of multi-factor synaptic plasticity with calcium-based dynamics. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 63(12):2189–2199, 2016.

[118] Philipp Häfliger, Misha Mahowald, and Lloyd Watts. A spike based learning neuron in analog vlsi. *Advances in neural information processing systems*, 9, 1996.

[119] Shubha Ramakrishnan, Paul E Hasler, and Christal Gordon. Floating gate synapses with spike-time-dependent plasticity. *IEEE Transactions on Biomedical Circuits and Systems*, 5(3):244–252, 2011.

[120] Giacomo Indiveri, Elisabetta Chicca, and Rodney Douglas. A vlsi array of low-power spiking neurons and bistable synapses with spike-timing dependent plasticity. *IEEE transactions on neural networks*, 17(1):211–221, 2006.

[121] Elisabetta Chicca, Fabio Stefanini, Chiara Bartolozzi, and Giacomo Indiveri. Neuromorphic electronic circuits for building autonomous cognitive systems. *Proceedings of the IEEE*, 102(9):1367–1388, 2014.

[122] Eby G Friedman and JH Mulligan. Clock frequency and latency in synchronous digital systems. *IEEE Transactions on Signal Processing*, 39(4):930–934, 1991.

[123] EBY G FRIEDMAN and JH Mulligan Jr. Pipelining of high performance synchronous digital systems. *International Journal of Electronics Theoretical and Experimental*, 70(5):917–935, 1991.

[124] Fleur Zeldenrust, Wytse J Wadman, and Bernhard Englitz. Neural coding with bursts—current state and future perspectives. *Frontiers in computational neuroscience*, 12:48, 2018.

[125] LR Kern. Design and development of a real-time neural processor using the intel 80170nx etann. In *[Proceedings 1992] IJCNN International Joint Conference on Neural Networks*, volume 2, pages 684–689. IEEE, 1992.

[126] Michael Perrone and Leon Cooper. The ni1000: High speed parallel vlsi for implementing multilayer perceptrons. *Advances in Neural Information Processing Systems*, 7, 1994.

[127] Michael Gschwind, Valentina Salapura, and Oliver Maischberger. Space efficient neural net implementation. In *Proc. of the Second International ACM/SIGDA Workshop on Field Programmable Gate Arrays*, 1994.

[128] Michael Gschwind, Valentina Salapura, and Oliver Maischberger. A generic building block for hopfield neural networks with on-chip learning. In *IEEE International Symposium on Circuits and Systems, Atlanta, GA*, 1996.

[129] Eustace Painkras, Luis A Plana, Jim Garside, Steve Temple, Francesco Galluppi, Cameron Patterson, David R Lester, Andrew D Brown, and Steve B Furber. Spinnaker: A 1-w 18-core system-on-chip for massively-parallel neural network simulation. *IEEE Journal of Solid-State Circuits*, 48(8):1943–1953, 2013.

[130] Sebastian Höppner, Yexin Yan, Andreas Dixius, Stefan Scholze, Johannes Partzsch, Marco Stolba, Florian Kelber, Bernhard Vogginger, Felix Neumärker, Georg Ellguth, et al. The spinnaker 2 processing element architecture for hybrid digital neuromorphic computing. *arXiv preprint arXiv:2103.08392*, 2021.

[131] Mike Davies, Narayan Srinivasa, Tsung-Han Lin, Gautham Chinya, Yongqiang Cao, Sri Harsha Choday, Georgios Dimou, Prasad Joshi, Nabil Imam, Shweta Jain, et al. Loihi: A neuromorphic manycore processor with on-chip learning. *Ieee Micro*, 38(1):82–99, 2018.

[132] Charlotte Frenkel, Jean-Didier Legat, and David Bol. Morphic: A 65-nm 738k-synapse/mm^2 quad-core binary-weight digital neuromorphic processor with stochastic spike-driven online learning. *IEEE Transactions on Biomedical Circuits and Systems*, 13(5):999–1010, 2019.

[133] Gregory K Chen, Raghavan Kumar, H Ekin Sumbul, Phil C Knag, and Ram K Krishnamurthy. A 4096-neuron 1m-synapse 3.8-pj/sop spiking neural network with on-chip stdp learning and sparse weights in 10-nm finfet cmos. *IEEE Journal of Solid-State Circuits*, 54(4):992–1002, 2018.

[134] Jae-sun Seo, Bernard Brezzo, Yong Liu, Benjamin D Parker, Steven K Esser, Robert K Montoye, Bipin Rajendran, José A Tierno, Leland Chang, Dharmendra S Modha, et al. A 45nm cmos neuromorphic chip with a scalable architecture for learning in networks of spiking neurons. In *2011 IEEE Custom Integrated Circuits Conference (CICC)*, pages 1–4. IEEE, 2011.

[135] Charlotte Frenkel, Martin Lefebvre, Jean-Didier Legat, and David Bol. A 0.086-mm^2 12.7-pj/sop 64k-synapse 256-neuron online-learning digital spiking neuromorphic processor in 28-nm cmos. *IEEE Transactions on Biomedical Circuits and Systems*, 13(1):145–158, 2019.

[136] Phil Knag, Jung Kuk Kim, Thomas Chen, and Zhengya Zhang. A sparse coding neural network asic with on-chip learning for feature extraction and encoding. *IEEE Journal of Solid-State Circuits*, 50(4):1070–1079, 2015.

[137] Jung Kuk Kim, Phil Knag, Thomas Chen, and Zhengya Zhang. A 640m pixel/s 3.65 mw sparse event-driven neuromorphic object recognition processor with on-chip learning. In *2015 Symposium on VLSI Circuits (VLSI Circuits)*, pages C50–C51. IEEE, 2015.

[138] Jeongwoo Park, Juyun Lee, and Dongsuk Jeon. A 65-nm neuromorphic image classification processor with energy-efficient training through direct spike-only feedback. *IEEE Journal of Solid-State Circuits*, 55(1):108–119, 2019.

[139] Paul A Merolla, John V Arthur, Rodrigo Alvarez-Icaza, Andrew S Cassidy, Jun Sawada, Filipp Akopyan, Bryan L Jackson, Nabil Imam, Chen Guo, Yutaka Nakamura, et al. A million spiking-neuron integrated circuit with a scalable communication network and interface. *Science*, 345(6197):668–673, 2014.

[140] Garrick Orchard, E Paxon Frady, Daniel Ben Dayan Rubin, Sophia Sanborn, Sumit Bam Shrestha, Friedrich T Sommer, and Mike Davies. Efficient neuromorphic signal processing with loihi 2. In *2021 IEEE Workshop on Signal Processing Systems (SiPS)*, pages 254–259. IEEE, 2021.

[141] Dharmendra S Modha, Filipp Akopyan, Alexander Andreopoulos, Rathinakumar Appuswamy, John V Arthur, Andrew S Cassidy, Pallab Datta, Michael V DeBole, Steven K Esser, Carlos Ortega Otero, et al. Neural inference at the frontier of energy, space, and time. *Science*, 382(6668):329–335, 2023.

[142] A. M. Turing. Computing machinery and intelligence. *Mind*, 59(236):433–460, 1950.

[143] Suyog Gupta, Ankur Agrawal, Kailash Gopalakrishnan, and Pritish Narayanan. Deep learning with limited numerical precision. In *International conference on machine learning*, pages 1737–1746. PMLR, 2015.

[144] Mirembe Musisi-Nkambwe, Sahra Afshari, Hugh Barnaby, Michael Kozicki, and Ivan Sanchez Esqueda. The viability of analog-based accelerators for neuromorphic computing: A survey. *Neuromorphic Computing and Engineering*, 1(1):012001, 2021.

[145] Rodney Douglas, Misha Mahowald, and Carver Mead. Neuromorphic analogue vlsi. *Annual review of neuroscience*, 18:255–281, 1995.

[146] Johannes Schemmel, Daniel Brüderle, Andreas Grübl, Matthias Hock, Karlheinz Meier, and Sebastian Millner. A wafer-scale neuromorphic hardware system for

large-scale neural modeling. In *2010 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1947–1950. IEEE, 2010.

[147] Christian Pehle, Sebastian Billaudelle, Benjamin Cramer, Jakob Kaiser, Korbinian Schreiber, Yannik Stradmann, Johannes Weis, Aron Leibfried, Eric Müller, and Johannes Schemmel. The brainscales-2 accelerated neuromorphic system with hybrid plasticity. *Frontiers in Neuroscience*, 16, 2022.

[148] Ben Varkey Benjamin, Peiran Gao, Emmett McQuinn, Swadesh Choudhary, Anand R Chandrasekaran, Jean-Marie Bussat, Rodrigo Alvarez-Icaza, John V Arthur, Paul A Merolla, and Kwabena Boahen. Neurogrid: A mixed-analog-digital multichip system for large-scale neural simulations. *Proceedings of the IEEE*, 102(5):699–716, 2014.

[149] Alexander Neckar, Sam Fok, Ben V Benjamin, Terrence C Stewart, Nick N Oza, Aaron R Voelker, Chris Eliasmith, Rajit Manohar, and Kwabena Boahen. Braindrop: A mixed-signal neuromorphic architecture with a dynamical systems-based programming model. *Proceedings of the IEEE*, 107(1):144–164, 2018.

[150] Saber Moradi, Ning Qiao, Fabio Stefanini, and Giacomo Indiveri. A scalable multicore architecture with heterogeneous memory structures for dynamic neuromorphic asynchronous processors (dynaps). *IEEE transactions on biomedical circuits and systems*, 12(1):106–122, 2017.

[151] Stephen Brink, Stephen Nease, Paul Hasler, Shubha Ramakrishnan, Richard Wunderlich, Arindam Basu, and Brian Degnan. A learning-enabled neuron array ic based upon transistor channel models of biological phenomena. *IEEE Transactions on Biomedical Circuits and Systems*, 7(1):71–81, 2012.

[152] Christian Mayr, Johannes Partzsch, Marko Noack, Stefan Hänzsche, Stefan Scholze, Sebastian Höppner, Georg Ellguth, and Rene Schüffny. A biological-realtime neuromorphic system in 28 nm cmos using low-leakage switched capacitor circuits. *IEEE transactions on biomedical circuits and systems*, 10(1):243–254, 2015.

[153] Ning Qiao, Hesham Mostafa, Federico Corradi, Marc Osswald, Fabio Stefanini, Dora Sumislawska, and Giacomo Indiveri. A reconfigurable on-line learning spiking neuromorphic processor comprising 256 neurons and 128k synapses. *Frontiers in neuroscience*, 9:141, 2015.

[154] Johannes Schemmel, Sebastian Billaudelle, Philipp Dauer, and Johannes Weis. Accelerated analog neuromorphic computing. In *Analog Circuits for Machine Learning, Current/Voltage/Temperature Sensors, and High-speed Communication*, pages 83–102. Springer, 2022.

[155] Ole Richter, Chenxi Wu, Adrian M Whatley, German Köstinger, Carsten Nielsen, Ning Qiao, and Giacomo Indiveri. Dynap-se2: a scalable multi-core dynamic neuromorphic asynchronous spiking neural network processor. *Neuromorphic Computing and Engineering*, 4(1):014003, 2024.

[156] Robert R Schaller. Moore's law: past, present and future. *IEEE spectrum*, 34(6):52–59, 1997.

[157] Mark Bohr. A 30 year retrospective on dennard's mosfet scaling paper. *IEEE Solid-State Circuits Society Newsletter*, 12(1):11–13, 2007.

[158] Kelin J Kuhn, Martin D Giles, David Becher, Pramod Kolar, Avner Kornfeld, Roza Kotlyar, Sean T Ma, Atul Maheshwari, and Sivakumar Mudanai. Process technology variation. *IEEE Transactions on Electron Devices*, 58(8):2197–2208, 2011.

[159] K Parat and A Goda. Scaling trends in nand flash. In *2018 IEEE International Electron Devices Meeting (IEDM)*, pages 2–1. IEEE, 2018.

[160] Leon Chua. Memristor-the missing circuit element. *IEEE Transactions on circuit theory*, 18(5):507–519, 1971.

[161] Dmitri B Strukov, Gregory S Snider, Duncan R Stewart, and R Stanley Williams. The missing memristor found. *nature*, 453(7191):80–83, 2008.

[162] R Stanley Williams. How we found the missing memristor. *IEEE spectrum*, 45(12):28–35, 2008.

[163] Shuang Pi, Can Li, Hao Jiang, Weiwei Xia, Huolin Xin, J Joshua Yang, and Qiangfei Xia. Memristor crossbars with 4.5 terabits-per-inch-square density and two nanometer dimension. *arXiv preprint arXiv:1804.09848*, 2018.

[164] Navnidhi K Upadhyay, Hao Jiang, Zhongrui Wang, Shiva Asapu, Qiangfei Xia, and J Joshua Yang. Emerging memory devices for neuromorphic computing. *Advanced Materials Technologies*, 4(4):1800589, 2019.

[165] Amirali Amirsoleimani, Fabien Alibart, Victor Yon, Jianxiong Xu, Mohammad Reza Pazhouhandeh, Serge Ecoffey, Yann Beilliard, Roman Genov, and Dominique Drouin. In-memory vector-matrix multiplication in monolithic complementary metal–oxide–semiconductor-memristor integrated circuits: Design choices, challenges, and perspectives. *Advanced Intelligent Systems*, 2, 2020.

[166] Daniele Ielmini and H. S.Philip Wong. In-memory computing with resistive switching devices. *Nature Electronics*, 1(6):333–343, 2018.

[167] Max M. Shulaker, Gage Hills, Rebecca S. Park, Roger T. Howe, Krishna Saraswat, H. S.Philip Wong, and Subhasish Mitra. Three-dimensional integration of nanotechnologies for computing and data storage on a single chip. *Nature*, 547(7661):74–78, 2017.

[168] A Valentian, F Rummens, E Vianello, T Mesquida, C Lecat-Mathieu de Boissac, O Bichler, and C Reita. Fully integrated spiking neural network with analog neurons and rram synapses. In *2019 IEEE International Electron Devices Meeting (IEDM)*, pages 14–3. IEEE, 2019.

[169] Weier Wan, Rajkumar Kubendran, S Burc Eryilmaz, Wenqiang Zhang, Yan Liao, Dabin Wu, Stephen Deiss, Bin Gao, Priyanka Raina, Siddharth Joshi, et al. 33.1 a 74 tmacs/w cmos-rram neurosynaptic core with dynamically reconfigurable dataflow and in-situ transposable weights for probabilistic graphical models. In *2020 IEEE International Solid-State Circuits Conference-(ISSCC)*, pages 498–500. IEEE, 2020.

[170] Geoffrey W Burr, Robert M Shelby, Severin Sidler, Carmelo Di Nolfo, Junwoo Jang, Irem Boybat, Rohit S Shenoy, Pritish Narayanan, Kumar Virwani, Emanuele U Giacometti, et al. Experimental demonstration and tolerancing of a large-scale neural network (165 000 synapses) using phase-change memory as the synaptic weight element. *IEEE Transactions on Electron Devices*, 62(11):3498–3507, 2015.

[171] Huaqiang Wu, Peng Yao, Bin Gao, Wei Wu, Qingtian Zhang, Wenqiang Zhang, Ning Deng, Dong Wu, H-S Philip Wong, Shimeng Yu, et al. Device and circuit

optimization of rram for neuromorphic computing. In *2017 IEEE International Electron Devices Meeting (IEDM)*, pages 11–5. IEEE, 2017.

[172] G Pedretti, V Milo, S Ambrogio, R Carboni, S Bianchi, A Calderoni, N Ramaswamy, AS Spinelli, and D Ielmini. Memristive neural network for on-line learning and tracking with brain-inspired spike timing dependent plasticity. *Scientific reports*, 7(1):1–10, 2017.

[173] Injune Yeo, Sang-Gyun Gi, Gunuk Wang, and Byung-Geun Lee. A hardware and energy-efficient online learning neural network with an rram crossbar array and stochastic neurons. *IEEE Transactions on Industrial Electronics*, 68(11):11554–11564, 2020.

[174] G Pedretti, S Bianchi, V Milo, A Calderoni, N Ramaswamy, and D Ielmini. Modeling-based design of brain-inspired spiking neural networks with rram learning synapses. In *2017 IEEE International Electron Devices Meeting (IEDM)*, pages 28–1. IEEE, 2017.

[175] I Muñoz-Martin, S Bianchi, E Covi, G Piccolboni, A Bricalli, A Regev, JF Nodin, E Nowak, G Molas, and D Ielmini. A siox rram-based hardware with spike frequency adaptation for power-saving continual learning in convolutional neural networks. In *2020 IEEE Symposium on VLSI Technology*, pages 1–2. IEEE, 2020.

[176] M Ishii, S Kim, S Lewis, A Okazaki, J Okazawa, M Ito, M Rasch, W Kim, A Nomura, U Shin, et al. On-chip trainable 1.4 m 6t2r pcm synaptic array with 1.6 k stochastic lif neurons for spiking rbm. In *2019 IEEE International Electron Devices Meeting (IEDM)*, pages 14–2. IEEE, 2019.

[177] Jikai Lu, Jinsong Wei, Junjie An, Chenggao Zhang, Tuo Shi, and Qi Liu. Rram-based analog-weight spiking neural network accelerator with in-situ learning for iot applications. In *2021 IEEE 14th International Conference on ASIC (ASICON)*, pages 1–4. IEEE, 2021.

[178] Qiang Yu, Huajin Tang, Kay Chen Tan, and Haizhou Li. Precise-spike-driven synaptic plasticity: Learning hetero-association of spatiotemporal spike patterns. *Plos one*, 8(11):e78318, 2013.

[179] Bernard Widrow and Michael A Lehr. 30 years of adaptive neural networks: perceptron, madaline, and backpropagation. *Proceedings of the IEEE*, 78(9):1415–1442, 1990.

[180] Sangbum Kim, M Ishii, S Lewis, T Perri, M BrightSky, W Kim, R Jordan, Geoffrey W Burr, Norma Sosa, A Ray, et al. Nvm neuromorphic core with 64k-cell (256-by-256) phase change memory synaptic array with on-chip neuron circuits for continuous in-situ learning. In *2015 IEEE international electron devices meeting (IEDM)*, pages 17–1. IEEE, 2015.

[181] Vyacheslav A Demin, Dmitry V Nekhaev, Igor A Surazhevsky, Kristina E Nikiruy, Andrey V Emelyanov, Sergey N Nikolaev, Vladimir V Rylkov, and Mikhail V Kovalchuk. Necessary conditions for stdp-based pattern recognition learning in a memristive spiking neural network. *Neural Networks*, 134:64–75, 2021.

[182] Luis A Camuñas-Mesa, Bernabé Linares-Barranco, and Teresa Serrano-Gotarredona. Implementation of a tunable spiking neuron for stdp with memristors in fdsoi 28nm.

In *2020 2nd IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS)*, pages 94–98. IEEE, 2020.

[183] C Lee Giles and Tom Maxwell. Learning, invariance, and generalization in high-order neural networks. *Applied optics*, 26(23):4972–4978, 1987.

[184] Md Musabbir Adnan, Sagarvarma Sayyaparaju, Garrett S Rose, Catherine D Schuman, Bon Woong Ku, and Sung Kyu Lim. A twin memristor synapse for spike timing dependent learning in neuromorphic systems. In *2018 31st IEEE International System-on-Chip Conference (SOCC)*, pages 37–42. IEEE, 2018.

[185] Sagarvarma Sayyaparaju, Gangotree Chakma, Sherif Amer, and Garrett S Rose. Circuit techniques for online learning of memristive synapses in cmos-memristor neuromorphic systems. In *Proceedings of the on Great Lakes Symposium on VLSI 2017*, pages 479–482, 2017.

[186] Karsten Beckmann, Josh Holt, Harika Manem, Joseph Van Nostrand, and Nathaniel C Cady. Nanoscale hafnium oxide rram devices exhibit pulse dependent behavior and multi-level resistance capability. *Mrs Advances*, 1(49):3355–3360, 2016.

[187] Ryan Weiss, Hritom Das, Nishith N Chakraborty, and Garrett S Rose. Stdp based online learning for a current-controlled memristive synapse. In *2022 IEEE 65th International Midwest Symposium on Circuits and Systems (MWSCAS)*, pages 1–4. IEEE, 2022.

[188] Taimur Ahmed, Sumeet Walia, Edwin LH Mayes, Rajesh Ramanathan, Vipul Bansal, Madhu Bhaskaran, Sharath Sriram, and Omid Kavehei. Time and rate dependent synaptic learning in neuro-mimicking resistive memories. *Scientific reports*, 9(1):1–11, 2019.

[189] Gwendal Lecerf, Jean Tomas, and Sylvain Saïghi. Excitatory and inhibitory memristive synapses for spiking neural networks. In *2013 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1616–1619. IEEE, 2013.

[190] Sören Boyn, Julie Grollier, Gwendal Lecerf, Bin Xu, Nicolas Locatelli, Stéphane Fusil, Stéphanie Girod, Cécile Carrétéro, Karin Garcia, Stéphane Xavier, et al. Learning through ferroelectric domain dynamics in solid-state synapses. *Nature communications*, 8(1):1–7, 2017.

[191] Benjamin Max, Michael Hoffmann, Halid Mulaosmanovic, Stefan Slesazeck, and Thomas Mikolajick. Hafnia-based double-layer ferroelectric tunnel junctions as artificial synapses for neuromorphic computing. *ACS Applied Electronic Materials*, 2(12):4023–4033, 2020.

[192] Arnob Saha, ANM Nafiul Islam, Zijian Zhao, Shan Deng, Kai Ni, and Abhronil Sengupta. Intrinsic synaptic plasticity of ferroelectric field effect transistors for online learning. *Applied Physics Letters*, 119(13):133701, 2021.

[193] Peng Zhou, Julie A Smith, Laura Deremo, Stephen K Heinrich-Barna, and Joseph S Friedman. Synchronous unsupervised stdp learning with stochastic stt-mram switching. *arXiv preprint arXiv:2112.05707*, 2021.

[194] Nikhil Garg, Ismael Balafrej, Terrence C Stewart, Jean-Michel Portal, Marc Bocquet, Damien Querlioz, Dominique Drouin, Jean Rouat, Yann Beilliard, and Fabien

Alibart. Voltage-dependent synaptic plasticity: Unsupervised probabilistic hebbian plasticity rule based on neurons membrane potential. *Frontiers in Neuroscience*, 16:983950, 2022.

[195] Timothée Masquelier and Simon J Thorpe. Unsupervised learning of visual features through spike timing dependent plasticity. *PLoS computational biology*, 3(2):e31, 2007.

[196] Chankyu Lee, Priyadarshini Panda, Gopalakrishnan Srinivasan, and Kaushik Roy. Training deep spiking convolutional neural networks with stdp-based unsupervised pre-training followed by supervised fine-tuning. *Frontiers in neuroscience*, 12:435, 2018.

[197] Tim VP Bliss and Graham L Collingridge. A synaptic model of memory: long-term potentiation in the hippocampus. *Nature*, 361(6407):31–39, 1993.

[198] Abigail Morrison, Markus Diesmann, and Wulfram Gerstner. Phenomenological models of synaptic plasticity based on spike timing. *Biological cybernetics*, 98:459–478, 2008.

[199] Peter U Diehl and Matthew Cook. Efficient implementation of stdp rules on spinnaker neuromorphic hardware. In *2014 International Joint Conference on Neural Networks (IJCNN)*, pages 4288–4295. IEEE, 2014.

[200] Amirreza Yousefzadeh, Timothée Masquelier, Teresa Serrano-Gotarredona, and Bernabé Linares-Barranco. Hardware implementation of convolutional stdp for on-line visual feature learning. In *2017 IEEE international symposium on circuits and systems (ISCAS)*, pages 1–4. IEEE, 2017.

[201] Corey Lammie, Tara Julia Hamilton, André van Schaik, and Mostafa Rahimi Azghadi. Efficient fpga implementations of pair and triplet-based stdp for neuromorphic architectures. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 66(4):1558–1570, 2018.

[202] Amrutha Manoharan, Gadamsetty Muralidhar, and Binsu J Kailath. A novel method to implement stdp learning rule in verilog. In *2020 IEEE Region 10 Symposium (TENSYMP)*, pages 1779–1782. IEEE, 2020.

[203] Simon Friedmann, Johannes Schemmel, Andreas Grübl, Andreas Hartel, Matthias Hock, and Karlheinz Meier. Demonstrating hybrid learning in a flexible neuromorphic hardware system. *IEEE transactions on biomedical circuits and systems*, 11(1):128–142, 2016.

[204] Govind Narasimman, Subhrajit Roy, Xuanyao Fong, Kaushik Roy, Chip-Hong Chang, and Arindam Basu. A low-voltage, low power stdp synapse implementation using domain-wall magnets for spiking neural networks. In *2016 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 914–917. IEEE, 2016.

[205] Andreas Grübl, Sebastian Billaudelle, Benjamin Cramer, Vitali Karasenko, and Johannes Schemmel. Verification and design methods for the brainscales neuromorphic hardware system. *Journal of Signal Processing Systems*, 92:1277–1292, 2020.

[206] Satoshi Moriya, Tatsuki Kato, Daisuke Oguchi, Hideaki Yamamoto, Shigeo Sato, Yasushi Yuminaka, Yoshihiko Horio, and Jordi Madrenas. Analog-circuit implementation of multiplicative spike-timing-dependent plasticity with linear decay. *Nonlinear Theory and Its Applications, IEICE*, 12(4):685–694, 2021.

[207] Teresa Serrano-Gotarredona, Timothée Masquelier, Themistoklis Prodromakis, Giacomo Indiveri, and Bernabe Linares-Barranco. Stdp and stdp variations with memristors for spiking neuromorphic learning systems. *Frontiers in neuroscience*, 7:2, 2013.

[208] Damien Querlioz, Olivier Bichler, Philippe Dollfus, and Christian Gamrat. Immunity to device variations in a spiking neural network with memristive nanodevices. *IEEE transactions on nanotechnology*, 12(3):288–295, 2013.

[209] Stefano Ambrogio, Nicola Ciocchini, Mario Laudato, Valerio Milo, Agostino Pirovano, Paolo Fantini, and Daniele Ielmini. Unsupervised learning by spike timing dependent plasticity in phase change memory (pcm) synapses. *Frontiers in neuroscience*, 10:56, 2016.

[210] Irem Boybat, Manuel Le Gallo, SR Nandakumar, Timoleon Moraitis, Thomas Parnell, Tomas Tuma, Bipin Rajendran, Yusuf Leblebici, Abu Sebastian, and Evangelos Eleftheriou. Neuromorphic computing with multi-memristive synapses. *Nature communications*, 9(1):2514, 2018.

[211] Yilong Guo, Huaqiang Wu, Bin Gao, and He Qian. Unsupervised learning on resistive memory array based spiking neural networks. *Frontiers in neuroscience*, 13:812, 2019.

[212] Alain Artola, S Bröcher, and Wolf Singer. Different voltage-dependent thresholds for inducing long-term depression and long-term potentiation in slices of rat visual cortex. *Nature*, 347(6288):69–72, 1990.

[213] Peter Jedlicka, Lubica Benuskova, and Wickliffe C Abraham. A voltage-based stdp rule combined with fast bcm-like metaplasticity accounts for ltp and concurrent "heterosynaptic" ltd in the dentate gyrus in vivo. *PLoS computational biology*, 11(11):e1004588, 2015.

[214] Claudia Clopath, Lars Büsing, Eleni Vasilaki, and Wulfram Gerstner. Connectivity reflects coding: A model of voltage-based spike-timing-dependent-plasticity with homeostasis. *Nature Precedings*, pages 1–1, 2009.

[215] Nick Diederich, Thorsten Bartsch, Hermann Kohlstedt, and Martin Ziegler. A memristive plasticity model of voltage-based stdp suitable for recurrent bidirectional neural networks in the hippocampus. *Scientific reports*, 8(1):9367, 2018.

[216] Giancarlo La Camera, Alexander Rauch, Hans-R Lüscher, Walter Senn, and Stefano Fusi. Minimal models of adapted neuronal response to in vivo–like input currents. *Neural computation*, 16(10):2101–2124, 2004.

[217] Mark CW Van Rossum, Guo Qiang Bi, and Gina G Turrigiano. Stable hebbian learning from spike timing-dependent plasticity. *Journal of neuroscience*, 20(23):8812–8821, 2000.

[218] Jun-nosuke Teramae and Tomoki Fukai. Computational implications of lognormally distributed synaptic weights. *Proceedings of the IEEE*, 102(4):500–512, 2014.

[219] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[220] Trevor Bekolay, James Bergstra, Eric Hunsberger, Travis DeWolf, Terrence C Stewart, Daniel Rasmussen, Xuan Choo, Aaron Russell Voelker, and Chris Eliasmith.

Nengo: a python tool for building large-scale functional brain models. *Frontiers in neuroinformatics*, 7:48, 2014.

[221] Damien Querlioz, WS Zhao, Philippe Dollfus, J-O Klein, Olivier Bichler, and Christian Gamrat. Bioinspired networks with nanoscale memristive devices that combine the unsupervised and supervised learning approaches. In *Proceedings of the 2012 IEEE/ACM International Symposium on Nanoscale Architectures*, pages 203–210, 2012.

[222] Seongbin Oh, Chul-Heung Kim, Soochang Lee, Jang Saeng Kim, and Jong-Ho Lee. Unsupervised online learning of temporal information in spiking neural network using thin-film transistor-type nor flash memory devices. *Nanotechnology*, 30(43):435206, 2019.

[223] Federico Paredes-Vallés, Kirk YW Scheper, and Guido CHE De Croon. Unsupervised learning of a hierarchical spiking neural network for optical flow estimation: From events to global motion perception. *IEEE transactions on pattern analysis and machine intelligence*, 42(8):2051–2064, 2019.

[224] Amadeus Maes, Mauricio Barahona, and Claudia Clopath. Learning compositional sequences with multiple time scales through a hierarchical network of spiking neurons. *PLoS Computational Biology*, 17(3):e1008866, 2021.

[225] Saeed Reza Kheradpisheh, Mohammad Ganjtabesh, Simon J Thorpe, and Timothée Masquelier. Stdp-based spiking deep convolutional neural networks for object recognition. *Neural Networks*, 99:56–67, 2018.

[226] Matthieu Gilson, Anthony Burkitt, and J Leo van Hemmen. Stdp in recurrent neuronal networks. *Frontiers in computational neuroscience*, 4:23, 2010.

[227] Sung Hyun Jo, Ting Chang, Idongesit Ebong, Bhavitavya B Bhadviya, Pinaki Mazumder, and Wei Lu. Nanoscale memristor device as synapse in neuromorphic systems. *Nano letters*, 10(4):1297–1301, 2010.

[228] Raqibul Hasan, Tarek M Taha, and Chris Yakopcic. On-chip training of memristor based deep neural networks. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 3527–3534. IEEE, 2017.

[229] Fabien Alibart, Elham Zamanidoost, and Dmitri B Strukov. Pattern classification by memristive crossbar circuits using ex situ and in situ training. *Nature communications*, 4(1):2072, 2013.

[230] Gregory S Snider. Self-organized computation with unreliable, memristive nanodevices. *Nanotechnology*, 18(36):365202, 2007.

[231] Wolfgang Maass. Networks of spiking neurons: the third generation of neural network models. *Neural networks*, 10(9):1659–1671, 1997.

[232] AA Fursina, RGS Sofin, IV Shvets, and D Natelson. Origin of hysteresis in resistive switching in magnetite is joule heating. *Physical Review B—Condensed Matter and Materials Physics*, 79(24):245131, 2009.

[233] Rainer Waser, Regina Dittmann, Georgi Staikov, and Kristof Szot. Redox-based resistive switching memories-nanoionic mechanisms, prospects, and challenges. *Advanced Materials (Deerfield Beach, Fla.)*, 21(25-26):2632–2663, 2009.

[234] YB Nian, J Strozier, NJ Wu, X Chen, and A Ignatiev. Evidence for an oxygen diffusion model for the electric pulse induced resistance change effect in transition-metal oxides. *Physical review letters*, 98(14):146403, 2007.

[235] Geoffrey W Burr, Matthew J Breitwisch, Michele Franceschini, Davide Garetto, Kailash Gopalakrishnan, Bryan Jackson, Bülent Kurdi, Chung Lam, Luis A Lastras, Alvaro Padilla, et al. Phase change memory technology. *Journal of Vacuum Science & Technology B*, 28(2):223–262, 2010.

[236] Matthias Wuttig and Noboru Yamada. Phase-change materials for rewriteable data storage. *Nature materials*, 6(11):824–832, 2007.

[237] André Chanthbouala, Vincent Garcia, Ryan O Cherifi, Karim Bouzehouane, Stéphane Fusil, Xavier Moya, Stéphane Xavier, Hiroyuki Yamada, Cyrile Deranlot, Neil D Mathur, et al. A ferroelectric memristor. *Nature materials*, 11(10):860–864, 2012.

[238] Laura Bégon-Lours, Mattia Halter, Francesco Maria Puglisi, Lorenzo Benatti, Donato Francesco Falcone, Youri Popoff, Diana Dávila Pineda, Marilyne Sousa, and Bert Jan Offrein. Scaled, ferroelectric memristive synapse for back-end-of-line integration with neuromorphic hardware. *Advanced Electronic Materials*, page 2101395, 2022.

[239] Donato Francesco Falcone, Stephan Menzel, Tommaso Stecconi, Antonio La Porta, Ludovico Carraria-Martinotti, Bert Jan Offrein, and Valeria Bragaglia. Physical modeling and design rules of analog conductive metal oxide-hfo 2 reram. In *2023 IEEE International Memory Workshop (IMW)*, pages 1–4. IEEE, 2023.

[240] Daniel E Feldman. The spike-timing dependence of plasticity. *Neuron*, 75(4):556–571, 2012.

[241] Shimeng Yu, Ximeng Guan, and H-S Philip Wong. On the stochastic nature of resistive switching in metal oxide rram: Physical modeling, monte carlo simulation, and experimental characterization. In *2011 International Electron Devices Meeting*, pages 17–3. IEEE, 2011.

[242] Charlotte Frenkel, Giacomo Indiveri, Jean-Didier Legat, and David Bol. A fully-synthesized 20-gate digital spike-based synapse with embedded online learning. In *2017 IEEE international symposium on circuits and systems (ISCAS)*, pages 1–4. IEEE, 2017.

[243] Gaspard Goupy, Alexandre Juneau-Fecteau, Nikhil Garg, Ismael Balafrej, Fabien Alibart, Luc Frechette, Dominique Drouin, and Yann Beilliard. Unsupervised and efficient learning in sparsely activated convolutional spiking neural networks enabled by voltage-dependent synaptic plasticity. *Neuromorphic Computing and Engineering*, 3(1):014001, 2023.

[244] Gilbert Sassine, Selina La Barbera, Nabil Najjari, Marie Minvielle, Catherine Dubourdieu, and Fabien Alibart. Interfacial versus filamentary resistive switching in tio2 and hfo2 devices. *Journal of Vacuum Science & Technology B*, 34(1), 2016.

[245] André Chanthbouala, Arnaud Crassous, Vincent Garcia, Karim Bouzehouane, Stéphane Fusil, Xavier Moya, Julie Allibe, Bruno Dlubak, Julie Grollier, Stephane Xavier, et al. Solid-state memories based on ferroelectric tunnel junctions. *Nature nanotechnology*, 7(2):101–104, 2012.

[246] Nasir Ilyas, Dongyang Li, Chunmei Li, Xiangdong Jiang, Yadong Jiang, and Wei Li. Analog switching and artificial synaptic behavior of ag/sio x: Ag/tio x/p++-si memristor device. *Nanoscale research letters*, 15:1–11, 2020.

[247] Kristy A Campbell, Kolton T Drake, and Elisa H Barney Smith. Pulse shape and timing dependence on the spike-timing dependent plasticity response of ion-conducting memristors as synapses. *Frontiers in bioengineering and biotechnology*, 4:97, 2016.

[248] Kyungah Seo, Insung Kim, Seungjae Jung, Minseok Jo, Sangsu Park, Jubong Park, Jungho Shin, Kuyyadi P Biju, Jaemin Kong, Kwanghee Lee, et al. Analog memory and spike-timing-dependent plasticity characteristics of a nanoscale titanium oxide bilayer resistive switching device. *Nanotechnology*, 22(25):254023, 2011.

[249] Elie L Bienenstock, Leon N Cooper, and Paul W Munro. Theory for the development of neuron selectivity: orientation specificity and binocular interaction in visual cortex. *Journal of Neuroscience*, 2(1):32–48, 1982.

[250] David E Rumelhart and David Zipser. Feature discovery by competitive learning. *Cognitive science*, 9(1):75–112, 1985.

[251] Ke Yang, J Joshua Yang, Ru Huang, and Yuchao Yang. Nonlinearity in memristors for neuromorphic dynamic systems. *Small Science*, 2(1):2100049, 2022.

[252] Erika Covi, Stefano Brivio, Alexander Serb, Themis Prodromakis, Marco Fanciulli, and Sabina Spiga. Analog memristive synapse in spiking networks implementing unsupervised learning. *Frontiers in neuroscience*, 10:482, 2016.

[253] SR Nandakumar, Manuel Le Gallo, Irem Boybat, Bipin Rajendran, Abu Sebastian, and Evangelos Eleftheriou. A phase-change memory model for neuromorphic computing. *Journal of Applied Physics*, 124(15), 2018.

[254] Gopalakrishnan Srinivasan, Abhronil Sengupta, and Kaushik Roy. Magnetic tunnel junction based long-term short-term stochastic synapse for a spiking neural network with on-chip stdp learning. *Scientific reports*, 6(1):29545, 2016.

[255] Juan B Roldan, David Maldonado, Cristina Aguilera-Pedregosa, Enrique Moreno, Fernando Aguirre, Rocío Romero-Zaliz, Angel M García-Vico, Yaqing Shen, and Mario Lanza. Spiking neural networks based on two-dimensional materials. *npj 2D Materials and Applications*, 6(1):63, 2022.

[256] Uicheol Shin, Masatoshi Ishii, Atsuya Okazaki, Megumi Ito, Malte J Rasch, Wanki Kim, Akiyo Nomura, Wonseok Choi, Dooyong Koh, Kohji Hosokawa, et al. Pattern training, inference, and regeneration demonstration using on-chip trainable neuromorphic chips for spiking restricted boltzmann machine. *Advanced Intelligent Systems*, 4(8):2200034, 2022.

[257] D. Querlioz, O. Bichler, P. Dollfus, and C. Gamrat. Immunity to device variations in a spiking neural network with memristive nanodevices. *IEEE Transactions on Nanotechnology*, 12:288–295, 2013.

[258] Liang Zhao, Jinyu Zhang, Yu He, Ximeng Guan, He Qian, and Zhiping Yu. Dynamic modeling and atomistic simulations of set and reset operations in TiO_2-based unipolar resistive memory. *IEEE Electron Device Letters*, 32(5):677–679, 2011.

[259] Ji-Ho Ryu and Sungjun Kim. Artificial synaptic characteristics of tio2/hfo2 memristor with self-rectifying switching for brain-inspired computing. *Chaos, Solitons & Fractals*, 140:110236, 2020.

[260] Doosung Lee, Yonghun Sung, Hyunchul Sohn, Dae-Hong Ko, and Mann-Ho Cho. Change of resistive-switching in tio 2 films with additional hfo 2 thin layer. *Journal of the Korean Physical Society*, 60:1313–1316, 2012.

[261] Hojoon Ryu, Haonan Wu, Fubo Rao, and Wenjuan Zhu. Ferroelectric tunneling junctions based on aluminum oxide/zirconium-doped hafnium oxide for neuromorphic computing. *Scientific reports*, 9(1):20383, 2019.

[262] Nikhil Garg, Davide Florini, Patrick Dufour, Eloir Muhr, Mathieu C Faye, Marc Bocquet, Damien Querlioz, Yann Beilliard, Dominique Drouin, Fabien Alibart, et al. Versatile cmos analog lif neuron for memristor-integrated neuromorphic circuits. In *2024 International Conference on Neuromorphic Systems (ICONS)*, pages 185–192. IEEE, 2024.

[263] Matthew Newville, Till Stensitzki, Daniel B. Allen, and Antonino Ingargiola. LMFIT: Non-Linear Least-Square Minimization and Curve-Fitting for Python, September 2014.

[264] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2623–2631, 2019.

[265] Antoine Joubert, Bilel Belhadj, Olivier Temam, and Rodolphe Héliot. Hardware spiking neurons design: Analog or digital? In *The 2012 International Joint Conference on Neural Networks (IJCNN)*, pages 1–5. IEEE, 2012.

[266] C. Mead. Neuromorphic electronic systems. *Proceedings of the IEEE*, 78(10):1629–1636, 1990.

[267] Abu Sebastian, Manuel Le Gallo, Riduan Khaddam-Aljameh, and Evangelos Eleftheriou. Memory devices and applications for in-memory computing. *Nature nanotechnology*, 15(7):529–544, 2020.

[268] G. Indiveri. A low-power adaptive integrate-and-fire neuron circuit. In *Proceedings of the 2003 International Symposium on Circuits and Systems, 2003. ISCAS '03.*, volume 4, pages IV–IV, 2003.

[269] X. Wu, V. Saxena, K. Zhu, and S. Balagopal. A cmos spiking neuron for brain-inspired neural networks with resistive synapses and in situ learning. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 62(11):1088–1092, 2015.

[270] G. Lecerf, J. Tomas, S. Boyn, S. Girod, A. Mangalore, J. Grollier, and S. Saïghi. Silicon neuron dedicated to memristive spiking neural networks. In *2014 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1568–1571, 2014.

[271] Erik Bruun and Peter Shah. Dynamic range of low-voltage cascode current mirrors. In *Proceedings of ISCAS'95-International Symposium on Circuits and Systems*, volume 2, pages 1328–1331. IEEE, 1995.

[272] Matias Miguez, Joel Gak, Alejandro Oliva, and Alfredo Arnaud. Active current mirrors for low-voltage analog circuit design. *Circuits, Systems, and Signal Processing*, 36(12):4869–4885, 2017.

[273] Sylvain Saïghi, Christian G. Mayr, Teresa Serrano-Gotarredona, Heidemarie Schmidt, Gwendal Lecerf, Jean Tomas, Julie Grollier, Sören Boyn, Adrien F. Vincent, Damien Querlioz, Selina La Barbera, Fabien Alibart, Dominique Vuillaume, Olivier Bichler, Christian Gamrat, and Bernabé Linares-Barranco. Plasticity in memristive devices for spiking neural networks. *Frontiers in Neuroscience*, 9:51, 2015.

[274] Simon Laughlin. A simple coding procedure enhances a neuron's information capacity. *Zeitschrift für Naturforschung c*, 36(9-10):910–912, 1981.

[275] Richard Naud, Nicolas Marcille, Claudia Clopath, and Wulfram Gerstner. Firing patterns in the adaptive exponential integrate-and-fire model. *Biological cybernetics*, 99:335–347, 2008.

[276] Jan Benda and Andreas V. M. Herz. A universal model for spike-frequency adaptation. *Neural Computation*, 15:2523–2564, 2003.

[277] Gabrielle J Gutierrez and Sophie Denève. Population adaptation in efficient balanced networks. *ELife*, 8:e46926, 2019.

[278] Carlyn A Patterson, Stephanie C Wissig, and Adam Kohn. Distinct effects of brief and prolonged adaptation on orientation tuning in primary visual cortex. *Journal of Neuroscience*, 33(2):532–543, 2013.

[279] Manu V Nair and Giacomo Indiveri. An ultra-low power sigma-delta neuron circuit. In *2019 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–5. IEEE, 2019.

[280] Pavan Kumar Chundi, Dewei Wang, Sung Justin Kim, Minhao Yang, Joao Pedro Cerqueira, Joonsung Kang, Seungchul Jung, Sangjoon Kim, and Mingoo Seok. Always-on sub-microwatt spiking neural network based on spike-driven clock-and power-gating for an ultra-low-power intelligent device. *Frontiers in Neuroscience*, 15:684113, 2021.

[281] Behzad Razavi. *Design of analog CMOS integrated circuits*. McGraw-Hill, Inc., 2005.

[282] Joao Henrique Quintino Palhares, Nikhil Garg, Pierre-Antoine Mouny, Yann Beilliard, Jury Sandrini, Franck Arnaud, Lorena Anghel, Fabien Alibart, Dominique Drouin, and Philippe Galy. 28 nm fd-soi embedded phase change memory exhibiting near-zero drift at 12 k for cryogenic spiking neural networks (snns). *npj Unconventional Computing*, 2024.

[283] Kamila Janzakova, Ismael Balafrej, Ankush Kumar, Nikhil Garg, Corentin Scholaert, Jean Rouat, Dominique Drouin, Yannick Coffinier, Sébastien Pecqueur, and Fabien Alibart. Structural plasticity for neuromorphic networks with electropolymerized dendritic pedot connections. *Nature Communications*, 14(1):8143, 2023.

[284] Mahdi Ghazal, Ankush Kumar, Nikhil Garg, Sébastien Pecqueur, and Fabien Alibart. Neuromorphic signal classification using organic electrochemical transistor array and spiking neural simulations. *IEEE Sensors Journal*, 2024.

[285] Ariana Villarroel Marquez, N. McEvoy, and A. Pakdel. Organic electrochemical transistors (oects) toward flexible and wearable bioelectronics. *Molecules*, 25, 2020.

[286] Junren Chen, Siyao Yang, Huaqiang Wu, Giacomo Indiveri, and Melika Payvand. Scaling limits of memristor-based routers for asynchronous neuromorphic systems. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 2023.

[287] Jieming Yin, Zhifeng Lin, Onur Kayiran, Matthew Poremba, Muhammad Shoaib Bin Altaf, Natalie D. Enright Jerger, and G. Loh. Modular routing design for chiplet-based systems. *2018 ACM/IEEE 45th Annual International Symposium on Computer Architecture (ISCA)*, pages 726–738, 2018.

[288] Suhas Kumar, R Stanley Williams, and Ziwen Wang. Third-order nanocircuit elements for neuromorphic engineering. *Nature*, 585(7826):518–523, 2020.

[289] Matthew D Pickett, Gilberto Medeiros-Ribeiro, and R Stanley Williams. A scalable neuristor built with mott memristors. *Nature materials*, 12(2):114–117, 2013.

[290] Wei Yi, Kenneth K Tsang, Stephen K Lam, Xiwei Bai, Jack A Crowell, and Elias A Flores. Biological plausibility and stochasticity in scalable vo2 active memristor neurons. *Nature communications*, 9(1):4661, 2018.

[291] Dongseok Kwon, Sung Yun Woo, Jong-Ho Bae, Suhwan Lim, Byung-Gook Park, and Jong-Ho Lee. Hardware-based spiking neural networks using capacitor-less positive feedback neuron devices. *IEEE Transactions on Electron Devices*, 68(9):4766–4772, 2021.

[292] Jin Luo, Liutao Yu, Tianyi Liu, Mengxuan Yang, Zhiyuan Fu, Zhongxin Liang, Liang Chen, Cheng Chen, Shuhan Liu, Si Wu, et al. Capacitor-less stochastic leaky-fefet neuron of both excitatory and inhibitory connections for snn with reduced hardware cost. In *2019 IEEE International Electron Devices Meeting (IEDM)*, pages 6–4. IEEE, 2019.

[293] Thomas Dalgaty, Melika Payvand, Barbara De Salvo, Jerome Casas, Giusy Lama, Etienne Nowak, Giacomo Indiveri, and Elisa Vianello. Hybrid cmos-rram neurons with intrinsic plasticity. In *2019 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–5. IEEE, 2019.

[294] José PB Silva, Ruben Alcala, Uygar E Avci, Nick Barrett, Laura Bégon-Lours, Mattias Borg, Seungyong Byun, Sou-Chi Chang, Sang-Wook Cheong, Duk-Hyun Choe, et al. Roadmap on ferroelectric hafnia-and zirconia-based materials and devices. *APL Materials*, 11(8), 2023.

[295] Y. Goh, J. Hwang, Minki Kim, Yongsung Lee, Minhyun Jung, and S. Jeon. Selector-less ferroelectric tunnel junctions by stress engineering and an imprinting effect for high-density cross-point synapse arrays. *ACS applied materials & interfaces*, 2021.

[296] F. Ambriz-Vargas, G. Kolhatkar, M. Broyer, Azza Hadj-Youssef, R. Nouar, A. Sarkissian, Reji Thomas, C. Gómez-Yáñez, M. Gauthier, and A. Ruediger. A complementary metal oxide semiconductor process-compatible ferroelectric tunnel junction. *ACS applied materials & interfaces*, 9 15:13262–13268, 2017.

[297] Tzu-Yun Wu, Tian-Sheuan Chang, Heng-Yuan Lee, S. Sheu, W. Lo, T. Hou, Hsin-Hui Huang, Y. Chu, Chih-Cheng Chang, Ming-Hung Wu, Chien-Hua Hsu, C. Wu, Min-Ci Wu, and Wen-Wei Wu. Sub-na low-current hzo ferroelectric tunnel junction for high-performance and accurate deep learning acceleration. *2019 IEEE International Electron Devices Meeting (IEDM)*, pages 6.3.1–6.3.4, 2019.

[298] Nikhil Garg, Ismael Balafrej, Yann Beilliard, Dominique Drouin, Fabien Alibart, and Jean Rouat. Signals to spikes for neuromorphic regulated reservoir computing and emg hand gesture recognition. In *International Conference on Neuromorphic Systems 2021*, pages 1–8, 2021.

[299] Lena Smirnova and Thomas Hartung. Neuronal cultures playing pong: first steps toward advanced screening and biological computing. *Neuron*, 110(23):3855–3856, 2022.

[300] Rahul Sarpeshkar, Woradorn Wattanapanitch, Scott K Arfin, Benjamin I Rapoport, Soumyajit Mandal, Michael W Baker, Michale S Fee, Sam Musallam, and Richard A Andersen. Low-power circuits for brain–machine interfaces. *IEEE Transactions on Biomedical Circuits and Systems*, 2(3):173–183, 2008.

[301] Sang Jin Kim, Kyoungjun Choi, Bora Lee, Yuna Kim, and B. Hong. Materials for flexible, stretchable electronics: Graphene and 2d materials. *Annual Review of Materials Research*, 45:63–84, 2015.

[302] Chen-Yu Wang, Cong Wang, Fanhao Meng, Pengfei Wang, Shuang Wang, S. Liang, and F. Miao. 2d layered materials for memristive and neuromorphic applications. *Advanced Electronic Materials*, 6, 2019.

[303] Kanghong Liao, Peixian Lei, Meilin Tu, Songwen Luo, Ting Jiang, W. Jie, and J. Hao. Memristor based on inorganic and organic two-dimensional materials: Mechanisms, performance, and synaptic applications. *ACS applied materials & interfaces*, 2021.

[304] G. Loh, S. Naffziger, and Kevin M. Lepak. Understanding chiplets today to anticipate future integration opportunities and limits. *2021 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 142–145, 2021.

[305] P. Vivet, E. Guthmuller, Y. Thonnart, G. Pillonnet, César Fuguet, I. Miro-Panadès, G. Moritz, J. Durupt, C. Bernard, D. Varreau, Julian J. H. Pontes, S. Thuries, David Coriat, M. Harrand, D. Dutoit, D. Lattard, L. Arnaud, J. Charbonnier, P. Coudrain, A. Garnier, F. Berger, A. Gueugnot, A. Greiner, Quentin L. Meunier, A. Farcy, A. Arriordaz, S. Chéramy, and F. Clermidy. Intact: A 96-core processor with six chiplets 3d-stacked on an active interposer with distributed interconnects and integrated power management. *IEEE Journal of Solid-State Circuits*, 56:79–97, 2021.