Design of an ultra low-power, Memristor based, Neural network micro-architecture

Yashvardhan Biyani

5396859

ET4300 - Master Thesis

TU Delft Delft University of Technology

Design of an ultra low-power, Memristor based, Neural network micro-architecture

ET4300 - Master Thesis

by

Yashvardhan Biyani

Student number: 5396859

Thesis Committee:

Dr. Ir. Rajendra Bishnoi; CE, TU Delft *(M.Sc. Supervisor)*

Prof. Dr. Ir. Said Hamdioui; CE, TU Delft

Abhairaj Singh; CE, TU Delft

Dr. Charlotte Frenkel; EI, TU Delft

September 20, 2022

Department of Quantum & Computer Engineering (QCE)
Faculty of EEMCS
Delft University of Technology


TU Delft
Delft
University of
Technology

Acknowledgement

I would like to express my gratitude to Professor Rajendra Bishnoi for being an outstanding mentor throughout my research process. Not only did he assist me in getting the necessary resources, but his regular support and direction were crucial to maintaining my motivation during this process. His detailed insights and feedback assisted me in steering my thesis in the right direction and expanding my understanding of in-memory computing. Prof. Said Hamdioui is one of the most supportive and dynamic professors I've known, for which I would like to express my gratitude. The fact that he is always accessible to his students and possessed a helpful personality, greatly facilitated in carrying out this research. I would also want to thank Abhairaj Singh and Sumit Diware for their invaluable guidance and assistance on countless occasions. I would also like to thank the Computer Engineering Department for their overall support and cooperation during my thesis.

I am indebted to my parents and family for their unconditional love, support, and advice, as well as for providing me with the chance to study here and develop as a person. I am fortunate to be a part of this journey where I have not only met some amazing people but also had the opportunity to work on some cutting-edge concepts that will help me in my future life and career.

Abstract

The advent of Artificial Intelligence (AI) and Internet-of-things (IoT) has led to a significant demand for edge computing and enabling Neural Network Implementation on edge devices. However, due to large MAC operations involved in the implementation of Neural networks, the traditional digital hardware based on the von-Neumann architecture is not well suited for an edge device. Computation-in-Memory (CIM) is an attractive alternative in mitigating the challenges involved with traditional hardware by directly processing the data within the memory. It utilizes emerging memory devices such as Resistive Random Access Memory (RRAM) to perform in-place computations in the analogue domain, thereby, eliminating the bottlenecks associated with the constant movement of data in the von-Neumann architecture. However, the standard implementation of CIM comes with several challenges, with the primary being high power consumption, which debate its implementation on an edge device in accelerating Neural network computation. Thus, this work proposes a novel CIM crossbar that has the potential to alleviate the challenges associated with the standard CIM crossbar. Subsequently, an ultra-low power micro-architecture design is proposed, based on the novel CIM crossbar, that can accelerate Binary Neural Networks (BNN) and Spiking Neural Networks (SNN) with high power efficiency. The benchmark results obtained over the implementation of custom-developed BNN and SNN, trained over the MNIST dataset, indicate a power reduction of 13x and 26x respectively for the proposed micro-architecture, compared to its standard CIM crossbar counterpart. In addition, the proposed micro-architecture exhibits energy savings of around 4-5x over both BNN and SNN, making it a promising alternative for accelerating neural network computation over edge devices.

Contents

List of Figures

List of Tables

Introduction

1.1 Motivation

Recent breakthroughs in Artificial Intelligence (AI) make them an increasingly compelling alternative for powering many real-world applications that were previously considered unachievable by explicit programming [6][7][8]. It is paving the way to a new realm of possibilities such as self-driving automobiles [9], Natural language processing (NLP) [10], and text interpretation, which otherwise cannot be realised solely by analytical techniques.
Neural networks (NN) are the current state-of-the-art contender for AI applications and have multiple variants such as Artificial Neural Network (ANN), Deep neural networks (DNN), Convolution neural networks (CNN) and, more recently, Spiking Neural networks (SNN). These networks are largely trained and implemented on powerful workstations or data centres with virtually unlimited resources. The advent of Internet-of-Things (IoT) [11] has led to a significant demand in edge AI [12], where the computations associated with the AI application are required to be locally carried out at the device level without engaging cloud computing. However, deploying these networks on edge devices, that are generally resource constrained, presents a major challenge.

The core computation in any neural network is a Multiply and Accumulate (MAC) Operation between the inputs and the trained weights, constituting about 70-80 % of the entire computation [13]. These operations are typically implemented on classic CMOS-based digital hardware such as CPU, GPU [14], AI oriented ASIC's (TPU [15]) and FPGA installed in workstations or data centers. While it is true that the aforementioned computing systems are being tailored to accelerate AI applications by incorporating high memory bandwidth and parallelism, their core function is still based on the von-Neumann architecture, where the memory and computational units are separate entities as depicted in figure 1-1.
Consequently, vector-vector multiplication in such digital systems is performed by sequentially fetching each pair of vector elements from the memory and multiplying them in a separate processing unit. This is followed by an addition operation in which the result of the multiplication is added to the previously stored data, again necessitating retrieval of data from memory and storage of the accumulated result. In the case of matrix multiplications, the entire method has to be repeated until each vector of the first matrix is multiplied by every vector of the second matrix to generate the final matrix.

Due to several data transactions between the processor and memory for each MAC operation, this entire procedure is a highly compute-intensive activity with a time complexity of at least

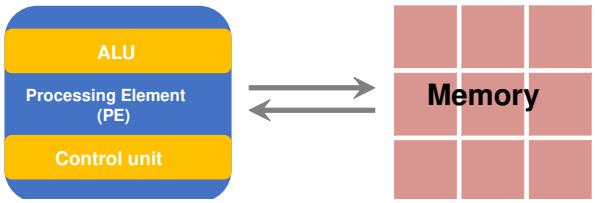


Figure 1-1: Simplified illustration of von-Neumann architecture

$O(n^2)$ and constitutes a major bottleneck in terms of latency and energy usage. Notably, the latency and energy cost associated to such data transactions can be orders of magnitude higher than that of the processing element [16]. This is the also the reason for the ever-widening gap between the processor and memory speeds, known as the memory wall [17], which limits the capabilities of modern digital processors, that otherwise can have exceptionally high clock frequencies. In addition, they can be pipeline enabled, have multiple cores, and use multi-level cache to accelerate computations. Subsequently, the load/store overhead of loading each pair of items from memory and storing the result back into memory is responsible for longer overall latencies, while the energy consumed during these transactions contributes to a much greater total energy consumption, rendering matrix multiplication on traditional digital hardware slow and inefficient.

Moreover, with the increase in the size and density of neural networks, this bottleneck is further aggravated because of massive data transfers within the system [18]. Therefore, it makes it impractical to deploy the same neural network on a purely digital edge device having limited power, energy and/or area budget. This encourages the development of an alternate computer architecture which is capable of not only accelerating MAC operations but also carrying them out with high energy/power efficiency. It is essential for alleviating the memory-processor (von Neumann) bottleneck and make neural network implementation practical for edge computing.

1.2 Overview of Computation-in-Memory (CIM)

One of the upcoming computing technologies which has the potential to mitigate the challenges involved with classic computing architecture is Computation-in-Memory (CIM). As the name suggests, it utilises emerging memory technologies, such as Resistive Random Access Memory (RRAM), to perform in-place computations and eliminate the bottlenecks associated to data movement. Figure 1-2 depicts a typical implementation of MAC operation over CIM, where a 2D crossbar of memristors is subjected to a set of input voltages. Since the memristors retain data in the form of resistance states, using simple circuit laws, it can be shown that the output currents generated in the crossbar are analogous to a single VMM operation between the input voltages and the matrix represented by the weights of memristors (refer section 2.1)

However, the conventional CIM crossbar as depicted in the figure 1-2, has issues with scalability owing to the parallel configuration of all the memristors. Although, the currents through

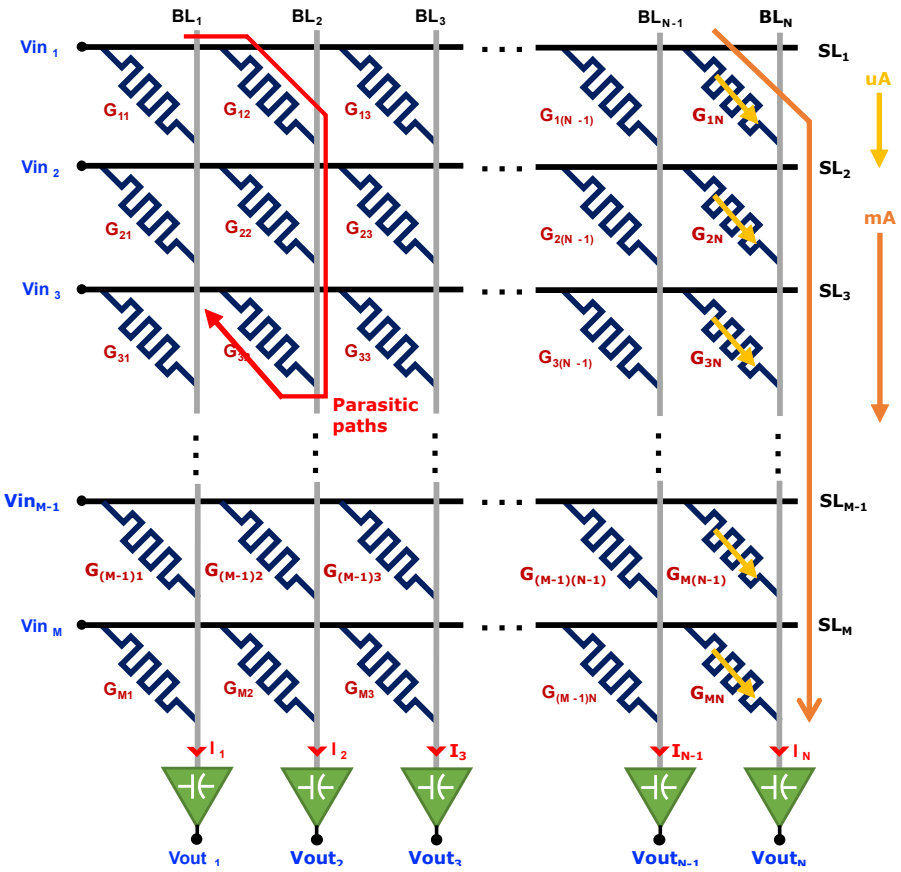


Figure 1-2: Typical implementation of a MAC operation over CIM architecture

a single memristor is in the orders of sub μA, the total currents through the crossbar can easily accumulate to the orders of mA if given enough size. This results in high power consumption that is unsuitable for edge devices expecting power consumption in the order of μW.

Several strategies have been proposed for reducing the power consumption by decreasing either the read voltages or the absolute conductances of memristors. For instance, SRIF [16] propose an additional circuit at the periphery that utilises an NMOS, coupled with a differential amplifier, to maintain the bitline at a higher voltage and effectively reduce the read voltage across RRAM to 0.1V, which is low compared to read voltages used in other works [19][20]. This limits the magnitude of induced currents which, in turn, reduce the power consumption. However, reducing the read voltages with this technique has its own challenges as the functionality of the crossbar is highly dependent on the performance of the differential amplifier. Moreover, owing to the limits of the differential amplifier, the read voltages cannot be reduced beyond a certain point with this technique. Furthermore, for the conductance values employed in this study, which vary from 50 μS to 500 μS, currents may still accumulate to the mA range for a crossbar of size 64×64.

On the other hand, PUMA [20] use RRAM at very low conductances ranging from 1 μS

to 10 μS to limit the magnitude of resulting currents. Extremely low conductance levels are undesirable because they are susceptible to device-to-device and cycle-to-cycle variations (refer section 2.2).Additionally, lower conductances or higher resistances induce more noise into the system, which may significantly impact the performance of analogue components.
Not only the currents, but the read voltages in standard CIM crossbar also pose a problem, which may reach as high as 500 mV as seen in PUMA. Usage of such high read voltages can cause significant read disturb in memristors (refer section 2.2) and gets worse with increasing conductance of memristors. This phenomena not only lead to the deviation of the stored weights from their initial value, but may also cause complete flipping of the resistance states overtime, inducing undesirable errors. In addition, the intrinsic architecture of conventional CIM crossbar makes it prone to flow of parasitic currents, as illustrated in figure 1-2, owing to the indirect coupling of columns via Source lines. By recycling the background parasitic/sneak current, attempts have been made [21] to reduce its impact on the performance of the crossbar. Nonetheless, the effect of parasitic currents cannot be fully eliminated since it mostly occurs due to mismatch in components, and may further degrade the accuracy as well as the efficiency of the system.

1.3 Problem Statement

Thus, a new approach is required in the design of CIM crossbar which can alleviate the challenges associated with the standard CIM crossbar. In addition to being compatible with the implementation of Neural networks, the novel approach must be able to limit the read currents in the range of nanoamperes (nA), thus reducing the power consumption as well as the phenomenon of read disturb.
Moreover, the columns in the this innovative approach must be isolated from each other to eliminate the existence of parasitic paths within the crossbar. Based on the above discussion, a novel CIM crossbar is proposed in this work which, to the author's knowledge, has not been done previously. This is proceeded by the realisation of a low power micro-architecture, based on the proposed novel CIM crossbar, to accelerate neural network implementation with high power efficiency.

1.4 Contribution

In this work, we present a low power micro-architecure based on the novel CIM crossbar discussed previously with the aim of accelerating neural network computations on edge devices. The micro-architecture is designed to accelerate binarised Artificial neural networks (ANN) and Spiking neural networks (SNN) as explained in section 2.3. The contributions in this work are as follows:

1. We propose and design a RRAM based novel CIM crossbar, which includes
    - A unique crossbar design using 2T1R configuration
    - 100 nA constant current source, capable of generating upto 1 uA of fixed current with a step of 100 nA
    - Usage of charge pump to reduce latency
    - Linear voltage to current converter also known as V/I converter

- Leaky-Integrate and Fire circuit equipped with refractory behaviour

2. Development of custom Binary Neural Network as well as Spiking Neural Network over MNIST dataset [22], compatible with the aforementioned micro-architecture, for the purpose of bench-marking the proposed hardware against current state-of-the-art
3. Chip tapeout of the proposed micro-architecture

The benchmark results obtained on the implementation of custom-developed Binary Neural Network (BNN) as well as Spiking Neural Network (SNN) exhibits promising performance of the proposed architecture. When compared to current state-of-the-art with reasonable overheads, the proposed architecture has a reduced power consumption by 13× and 26× on BNN and SNN, respectively. Although the proposed architecture has higher latency, it still consumes lower energy with respect to its standard counterpart by 4.75× for BNN and 3.94× for SNN. Finally, the worst case scenario calculations demonstrate up to two orders of magnitude reduction in power for the proposed architecture.

1.4.1 Outline

Following an overview of memristor technology and neural networks in Chapter 2, the proposed architecture is described in depth in Chapter 3. Chapter 4 covers the results of bench-marking this micro-architecture against two custom-built neural networks. The tapeout procedure and the design of the chip created for fabrication are discussed in Chapter 5. The Chapter 6 provides concluding remarks and recommendations for future phases of this project.

Background

In recent years, several neuromorphic computing architectures have been presented in an effort to accelerate neural network computations by digitally emulating functions of neuron. Spinnaker [23] and Loihi [24] present custom CMOS-based integrated circuits with multiple dedicated processing cores for performing Neural network computations with a high degree of parallelism. However, the underlying functionality of these works is still based on the von Neumann architecture, making them inferior in comparison to the previously described memristor crossbar technology. In an attempt to surpass the bottlenecks associated with the von Neumann architecture and push the limits of digital hardware, Truenorth et al.[25] propose a novel design based on Near-memory computing architecture [26].

Here, each processing core is equipped with its own SRAM block, to keep the memory close to the processing element, and thus minimise the bottlenecks associated with data movement. Despite having better performance to its von Neumann counterparts, CMOS technology is struggling with excessive sub-threshold leakage [27] and also, scalability [28].

In addition, the volatile nature of CMOS-based memory such as SRAM results in excessive power consumption. Several state-of-the-art works [19][20][29][30] based on CIM architecture have been presented in recent years that utilize the Resistive Random Access memory (RRAM) based standard CIM crossbar to accelerate neural network computation. Resistive Random Access memory (RRAM) has been the most preferred memristor technology (as explained in 2.2) for the crossbar because of its non-volatility, low read/write voltages, fast access speeds, and compact size.

2.1 Compute-in-Memory (CIM)

In-Memory Computing or Compute-in-Memory (CIM) is a trending computer architecture that shows promise in mitigating the issues associated with conventional computer architecture [31] [32]. As the name indicates, it seeks to directly process data stored in the memory, eliminating the requirement for data transportation and, subsequently, the von-Neumann bottleneck associated with conventional digital hardware. This is facilitated by utilizing a special kind of memory device, known as memristor [33], which stores data in the form of discrete resistance states. This data may be immediately processed in the analogue domain with the aid of simple circuit laws, resulting in a new computing paradigm that combines digital and analogue computing.

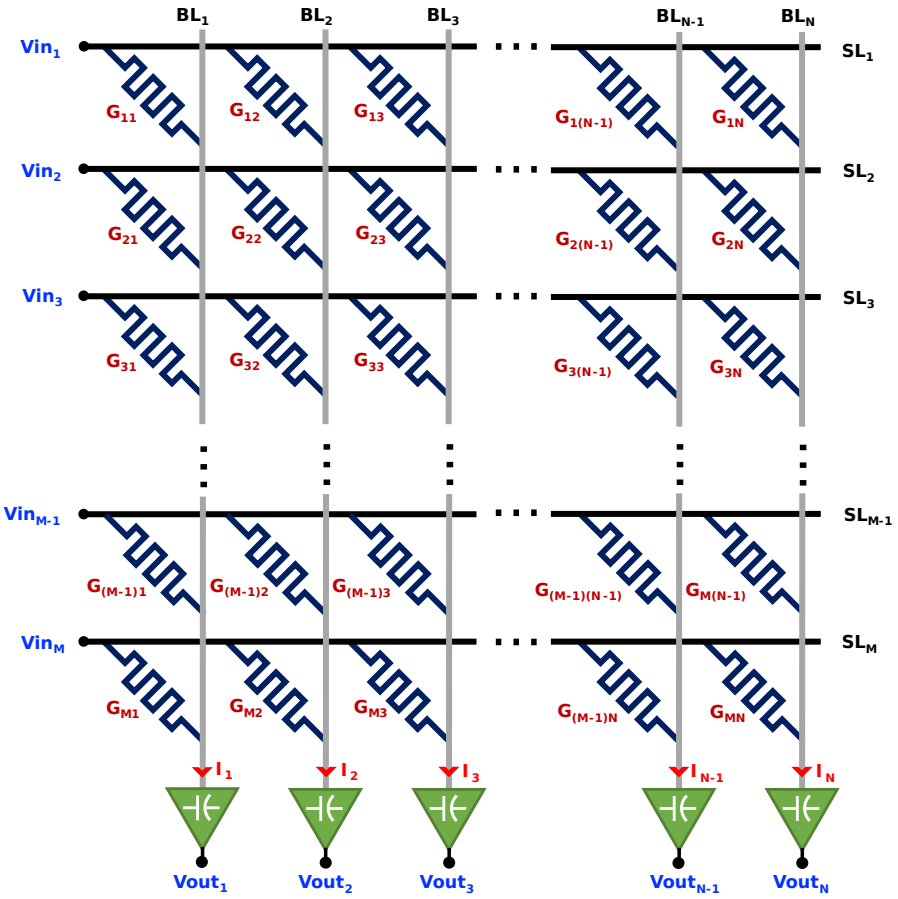


Figure 2-1: Illustration of a standard CIM crossbar depicting typical implementation of a VMM operation

Figure 2-1 depicts the standard implementation of a MAC operation based on CIM architecture, where M×N memresistive components are arranged in a 2D crossbar array and programmed to certain conductance states. Each memristor is connected across its respective Source line (SL) and Bit line (BL), which are the corresponding inputs and outputs of the crossbar. As seen in figure 2-1, a set of analogue voltages are applied to the SL (Row), while the BL (Column) are maintained at a fixed voltage, in this instance 0V. Now, utilizing Ohm's law, the currents across each memristor (i_{mn}) may be determined as follows:

$$i_{mn} = V_m.G_{mn} \tag{2-1}$$

From the equation 2-1, it can be observed that the magnitude of memristor currents constitute one multiply operation, where the input voltage (V_m) is the first operand and memristor conductance (G_{mn}) is the second operand. Thus, a memristor crossbar of size M×N can perform M×N multiply operations simultaneously. Moreover, according to Kirchoff's Current law, the memristor currents (i_{mn}) accumulate at their respective n^{th} BL giving rise to final column currents (I_n), as depicted in equation 2-2.

$$I_1 = i_{11} + i_{21} \dots i_{M1} \tag{2-2a}$$
$$I_2 = i_{21} + i_{22} \dots i_{M2} \tag{2-2b}$$
$$=$$
$$I_N = i_{1N} + i_{2N} \dots i_{MN} \tag{2-2c}$$

The magnitude of each column current (I_n) constitutes M addition operations, between M operands, i.e., the M memristor currents in n^{th} column. Subsequently, a crossbar of size M×N can perform N addition operations, of M operands each in parallel. By substituting equation 2-1 in 2-2, it can be rewritten as equation 2-3, to illustrate that the column current (I_n) is analogous to a single MAC operation between the input voltages and the memristor conductances in n^{th} column. Accordingly, a crossbar of size M×N can perform N MAC operations simultaneously.

$$I_1 = V_1G_{11} + V_2G_{21} \dots V_MG_{M1} \tag{2-3a}$$
$$I_2 = V_1G_{21} + V_2G_{22} \dots V_MG_{M2} \tag{2-3b}$$
$$=$$
$$I_N = V_1G_{1n} + V_2G_{2N} \dots V_MG_{MN} \tag{2-3c}$$

$$\implies \begin{bmatrix} I_1 & I_2 & \dots & I_N \end{bmatrix} = \begin{bmatrix} V_1 & V_2 & \dots & V_N \end{bmatrix} \begin{bmatrix} G_{11} & G_{12} & \dots & G_{1N} \\ G_{21} & G_{22} & \dots & G_{2N} \\ . & . & . & . \\ . & . & . & . \\ . & . & . & . \\ G_{M1} & G_{M2} & \dots & G_{MN} \end{bmatrix} \tag{2-4}$$

$$I_{1\times N} = V_{1\times M}.G_{M\times N} \tag{2-5}$$

Consider now that the set of applied input voltages represent a vector 'V' of size 1×M and the conductance values of the memristors constitute a matrix 'G' of size M×N. It can be shown from equation 2-4 and 2-5 that the magnitude of resulting currents across N columns is equivalent to the final vector 'I' of size 1×N, obtained on the multiplication of vector 'V' and matrix 'G'.

Therefore a standard CIM memristor crossbar of size M×N is capable of performing one Vector-Matrix Multiplication (VMM) operation between a vector of size M and a matrix of size M×N with complete parallelism, finishing the operation in one cycle. This is highly promising as the time complexity of performing a VMM operation has substantially lowered

down from O(n^2) to O(1), accelerating the computation by about M×N times.

Additionally, the in-place computations save time and energy that would otherwise be lost on the constant movement of data between the memory and processing unit. To convert digital values into suitable analogue voltages and vice versa, additional peripheral components such as analogue-to-digital converters (ADCs) and digital-to-analogue converters (DACs) may be required since the core operations occur in the analogue domain. This may lead to increased power, latency, and area overhead that are not otherwise required in conventional digital hardware.

2.2 Memristor

As prospective successors, developing memory technologies such as phase-change random access memory (PCRAM), magnetic RAM (MRAM), ferroelectric RAM (FeRAM), and resistive RAM (RRAM), are being investigated[31][33]. Table 2-1 depicts a detailed comparison among the aforementioned memristor technologies.

Memory type	**Flash**	**PCRAM**	**MRAM**	**FeRAM**	**RRAM**
Cell	1T	1T1R	1T1R	1T1C	1T1R
R/W time (nsec)	60×10^3/ 2×10^6	76/ 20×10^3	12	200/134	8.5/10
R/W energy ($\frac{nJ}{bit}$)	-	15.3	0.9/1.3	9.77	-
Endurance	10^5	10^7	10^{16}	10^{13}	10^8
Retention	>10 Years	>10 Years	>10 Years	>10 Years	>10 Years
Density ($\frac{Mb}{mm^2}$)	555	15.7	0.35	0.93	6.66
Tech Node (nm)	34	58	90	130	180

Table 2-1: Detailed Comparision of Different Memristor Technologies[5]

Since each technology has its own advantages and disadvantages, it is difficult to find the most appropriate replacement for CMOS technology. For instance, PCRAM is the most energy-intensive due to its resistive switching behavior, while FeRAM suffers from signal degradation during scaling. Although MRAM has high endurance, it not only consumes a significant amount of power during programming, owing to its large write-currents, but also has a low magneto-resistance ratio (R_{off}/R_{on}) leading to lower accuracies. Thus, we focus on RRAM technology due to its compact structure, fast switching operation, and ease in scalability. Memristors are generally used with an access transistor connected in series, also known as one-transistor one-memristor (1T1R) configuration, in order to provide accessibility by individually selecting each memristor in a crossbar via word lines (WL) for the purpose of programming.

2.2.1 Resistive Random Access Memory (RRAM)

Resistive Random Access Memory (RRAM) has drawn considerable attention as a CIM memory element because of its many overall benefits over other technologies. A RRAM device is

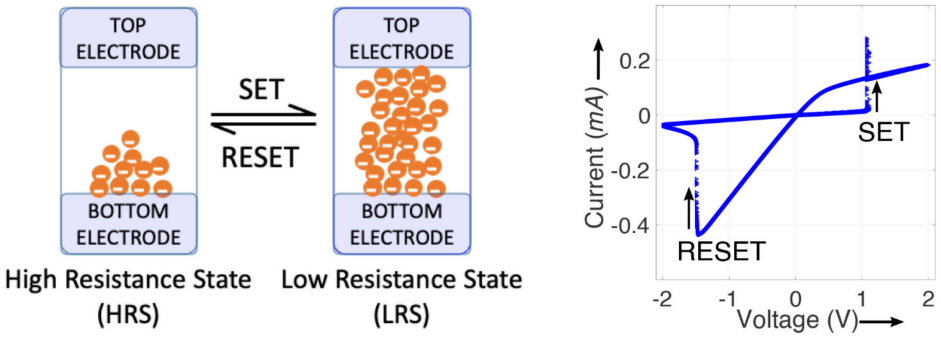


Figure 2-2: RRAM and its typical I/V characterisitcs[1]

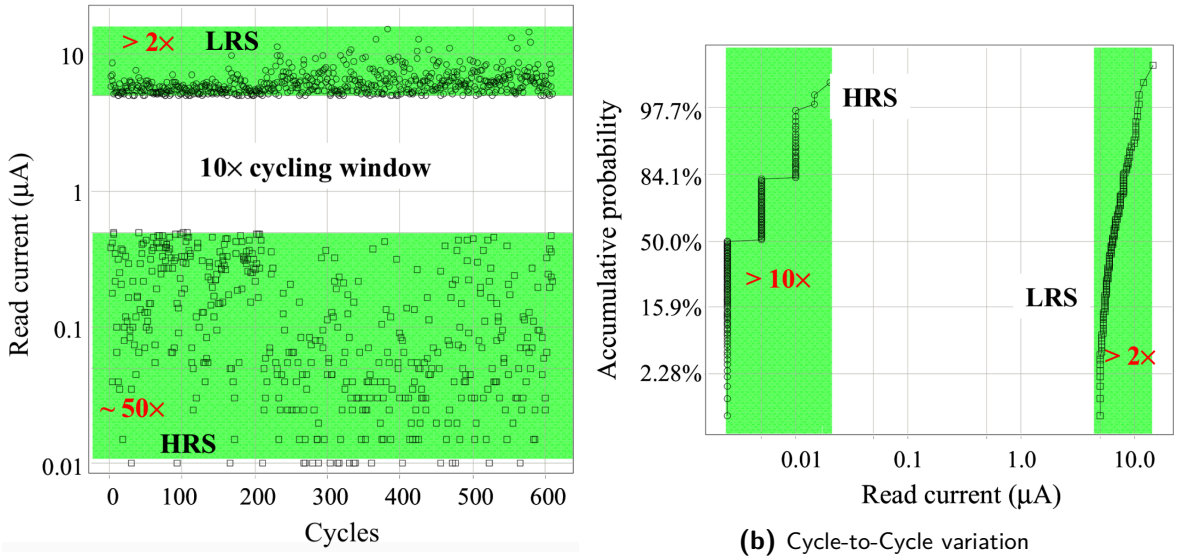
composed of an oxide material sandwiched between two metal electrodes. By modifying the oxygen vacancies induced in this layer, the RRAM's resistance may be altered and utilized to store data. RRAMs typically exhibit a high-resistance state (HRS) and a low-resistance state (LRS) which is used to represent and store binary data as '0' and '1', respectively. The switching from HRS to LRS is known as "SET," while the switching from LRS to HRS is known as "RESET", as depiceted in figure 2-2.

On applying the Set Voltage (V_{SET}) across RRAM, a conductive path, known as the filament, begins to form, which enhances the conductivity of the oxide layer. Once the formation of the filament is completed, there is a sudden change in the channel resistance of RRAM, leading to LRS. Similarly, the application of the Reset Voltage (V_{RESET}) across the RRAM, causes the rupture of the conductive filament, reducing the conductivity of the oxide layer and leading to HRS. This behavior can be verified by the hysterical I/V characteristics of RRAM as shown in figure 2-2.

In addition, it is possible to achieve multiple intermediate resistance states between the LRS and HRS by controlling the extent of filament creation or rupture, also referred to as Multi-Level Cell (MLC) operation. This allows multiple bits of data to be stored in a single RRAM device. However, current RRAM technology still struggles to achieve these intermediate states reliably, therefore they are normally employed to store one-bit data, as is the case in this work.

The underlying physics and fabrication process of a resistive memory device may result in a number of non-idealities that deviate from its ideal performance as a programmable resistor. Among the most major non-idealities are:

- **Device-to-Device Variations** Due to fabrication errors, different resistive memory devices exhibit varying resistance properties under identical programming conditions [32][2]. These variations further aggravate with increase in resistance of memory devices, as seen in figure 2-3a, thus discouraging the use of very high resistance states in order to maintain reasonable accuracy.
- **Cycle-to-Cycle Variations** Similarly, due to the stochastic nature of the underlying physics (like the filament formation/rupture in RRAM, crystallization/amorphization in PCM), the same resistive memory device exhibits various resistance characteristics at different points in time under identical programming conditions [32] [2]. Since an



(a) Device-to-Device variation

(b) Cycle-to-Cycle variation

Figure 2-3: Non idealities in RRAM technology [2]

increase in resistance makes stochastic processes more random, the extent of variations is greater for memresistors programmed to higher resistance states, as shown in Figure 2-3b,

- **Resistance Drift** With every read operation, the currents flowing through the memristor cause minute formation or rupture of the filament. Overtime, it causes the resistance of the memristor to deviate from its initial value, inducing errors. The higher the read currents, the higher will be the impact. Thus, it is important to keep the read currents in check in order to ensure the integrity of the stored data over an extended period of operation.

2.3 Neural Network implementation using CIM

2.3.1 Artificial Neural Network (ANN)

Presently, Artificial Neural Networks (ANN) are the most prevalant approach to artificial intelligence, which are inspired by the functionality of a basic perceptron structure [34]. Each layer of an ANN is comprised of a certain number of real-valued inputs and output neurons coupled through weighted paths. Consequently, a layer's input values are scaled by the weight of each forward path and accumulate at the output neuron. This is followed by the activation function built inside the neuron, which executes a defined set of computations on the accumulated result to generate the final output of that layer. These outputs are then fed as inputs to the subsequent layer, and this process continues until the neurons in the last layer generate final outputs. This process is also known as Forward pass/Inference, by which a neural network attempts to produce a prediction based on a collection of input features and a trained set of weights stored in the paths established between two successive layers.
This entire operation may be mapped to a CIM crossbar as illustrated in figure 2-4, where a

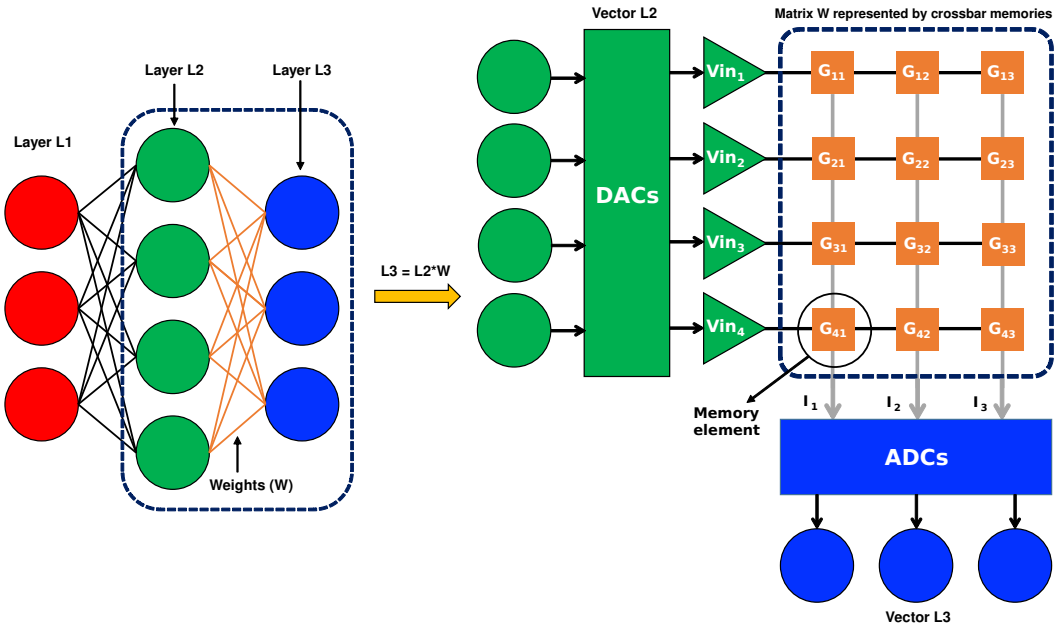


Figure 2-4: Implementation of Artifical Neural network using CIM

fully connected layer (Layer 2-3) with 4 inputs and 3 outputs constitutes a VMM operation between the 4×1 input vector and 4×3 weight matrix. The outputs are then converted into digital values using digital-to-analogue converters (DACs) at the periphery in order to perform out further computations relating to the activation function used by the specific layer. In specific cases, this computation can also be realised in the analogue domain with the usage of customized analogue circuits. For instance, the activation of a threshold may be readily emulated using a comparator, minimizing the necessity of power-hungry DACs.
In this work, we confine ourselves to binarised neural networks (BNN), in which both inputs and weights are binary, in other words, they can only take on two values. In addition, the outputs produced by the VMM operation between inputs and weights are subjected to a threshold activation that binarises these outputs against a threshold value.
This can be easily mapped on both the standard and the novel CIM crossbar, proposed later in this work, where the inputs are a fixed, high or low voltage and the RRAMs, representing the weights, can be reliably programmed to either its Set or Reset state. Additionally, the outputs may be directly binarised in the analogue domain using a comparator. Therefore, the implementation of Binary neural networks does not necessitate the use of additional circuits, such as ADCs and DACs, to convert digital values to analogue signals and vice versa. Compatible with both crossbars, BNNs are ideally suited for hardware comparisons between the conventional and novel CIM crossbars, as described in the chapter 4.

2.3.2 Spiking Neural Network (SNN)

Spiking Neural Networks (SNN) are a new category of low-power neural networks that draw their inspiration from the way in which the human brain functions. Considered one of the most advanced biological computers, the fundamental computation unit of a human brain is the neuron, the human brain consists of an extraordinarily dense network of neurons (com-

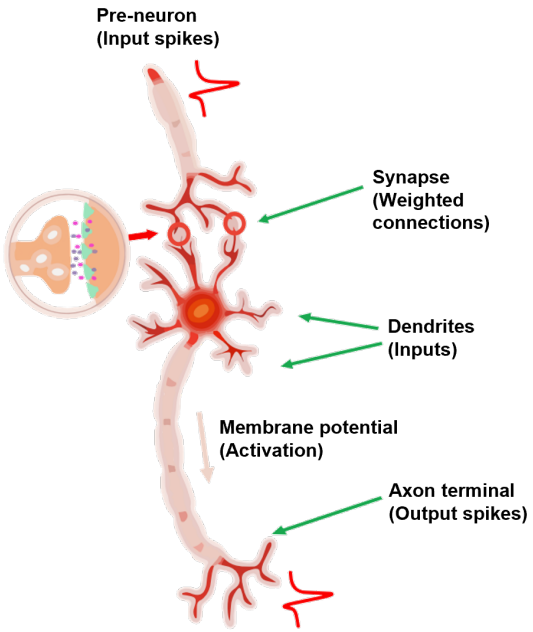


Figure 2-5: Schematic of a biological neuron[3]

putational units) interconnected by synapses, which are the weighted connections between neurons and represent memory inside the brain. A neuron consists of four major components, the soma being the main body of the neuron that conducts computations depending on dendritic inputs. The axon terminals, in the axon, are responsible for transmitting output from the soma to other neurons, as seen in the figure 2-5.
Neurons communicate utilizing binary spikes that are temporally encoded and are received or transmitted through a vast network of synapses that modulate the signals depending on their weights. Charges induced by these modulated input spikes accumulate in the neuron's body, hence altering its membrane potential. Leakage of charges from the neuron influences the membrane potential as well. Thus, if a neuron receives a sufficient number of spikes in a short period of time, the membrane potential may surpass the threshold of the neuron, resulting in the firing of an output spike from the neuron.
The human brain may contain up to 10^{11} neurons connected via 10^{15} spikes while requiring only twenty fJ of energy per operation [3]. Therefore, Spiking neural networks have the potential to enable ultra low-power neuromorphic computing, since they are inspired by the functioning and efficiency of the human brain.

SNN, in contrast to ANN, executes an inference in several time steps and receives a varied number of spikes at the input in each time step depending on the input encoding scheme. It is necessary to transform the inputs into binary spikes which get modulated through the network of weighted synpases they travel through. These modulated spikes then accumulate further to perform computations. Several encoding schemes, as depicted in figure 2-6, are proposed to convert the rea-valued input into relevant spikes over a defined number of time steps. The rate encoding scheme and the temporal encoding scheme are the two most common variants.
The rate encoding technique is the most widely utilized encoding scheme due to its simplicity

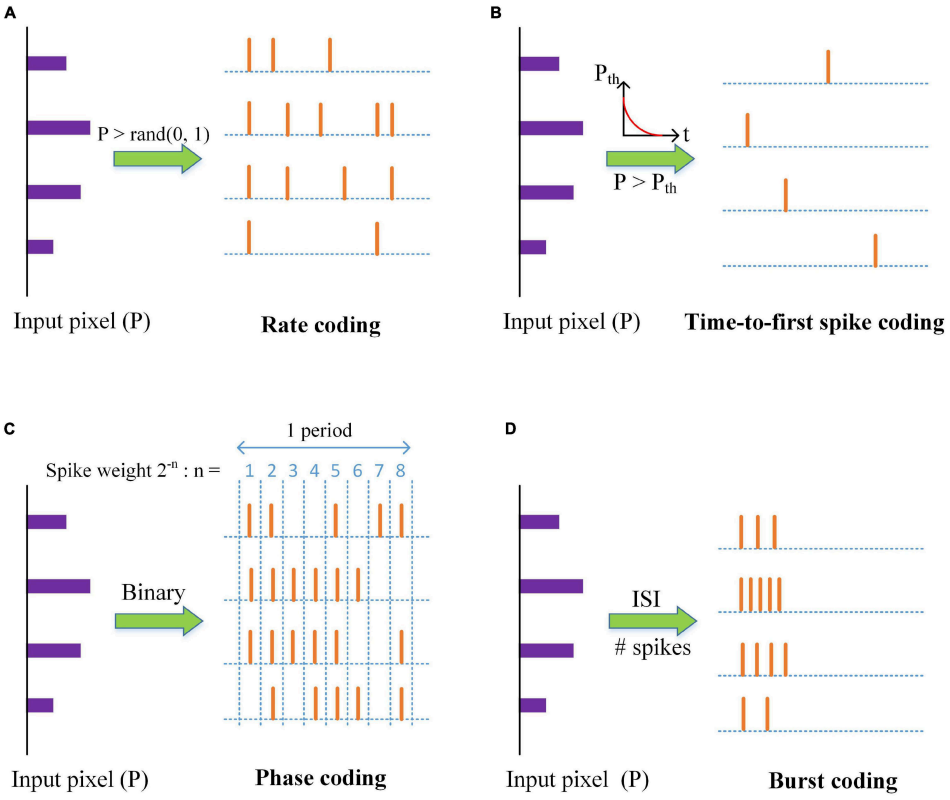


Figure 2-6: Input encoding schemes for Spiking neural network [4]

of implementation and noise resistance. In this scheme, the input values are first normalised before being considered as the probability values of the firing rate. These probability values are then used to convert the inputs into a Poseidon spike train, spread over the total number of time steps of inference. Consequently, we use a rate encoding scheme in this work for the sake of training and benchmarking.

The distinction between a Spiking neural network and a Binary neural network is in their activation dynamics. In recent years, several models have been presented that mathematically represent the dynamics of a biological neuron with varying degrees of complexity. The traditional Hodgkin-Huxley model [35] is a fourth-order biophysical model that represents the behavior of the currents flowing into the neuron ion channels in a manner that is biologically plausible. Due to its complexity, however, other second-order simplified models have been proposed, such as the FitzHugh and Nagumo model [36][37] and the Morris-Lecar model [38], among others.

In recent years, the Izhikevich model [39] and the Adaptive Exponential Integrate and Fire (AdEx) model [40] have gained a great deal of popularity due to their ability to mimic a wide range of spiking regimes observed in biological neurons by varying a reduced number of model parameters. While comprehensive biophysical models may accurately recreate the dynamics of biological neurons, they are computationally difficult to realise and is currently inconsistent with hardware implementations. Owing to these factors, simple first-order models, such as the Leaky Integrate and Fire model (LIF)[41][42], are presently the most popular choice for implementing Spiking Network Networks, which is also the case in this work.

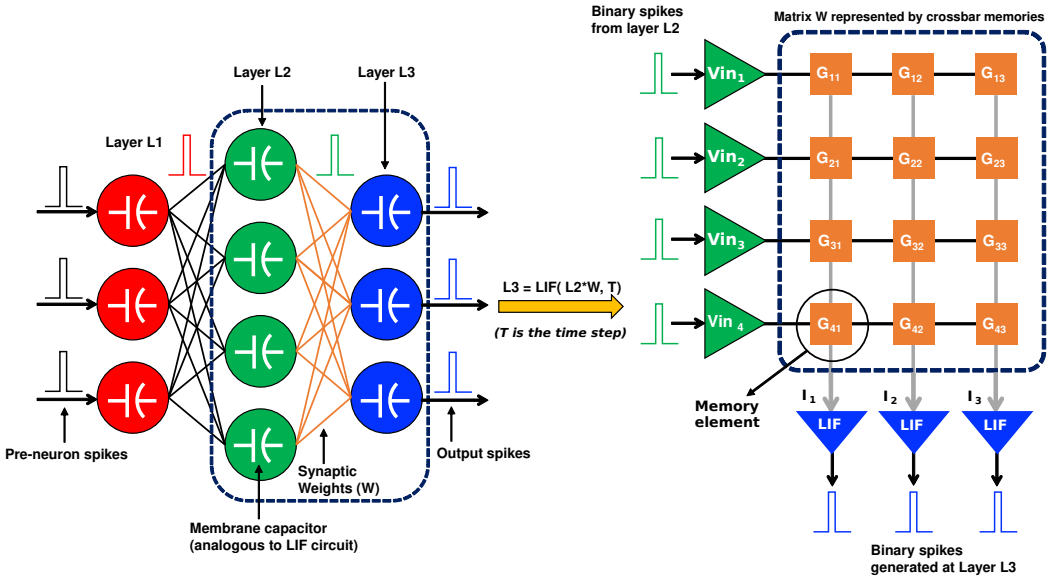
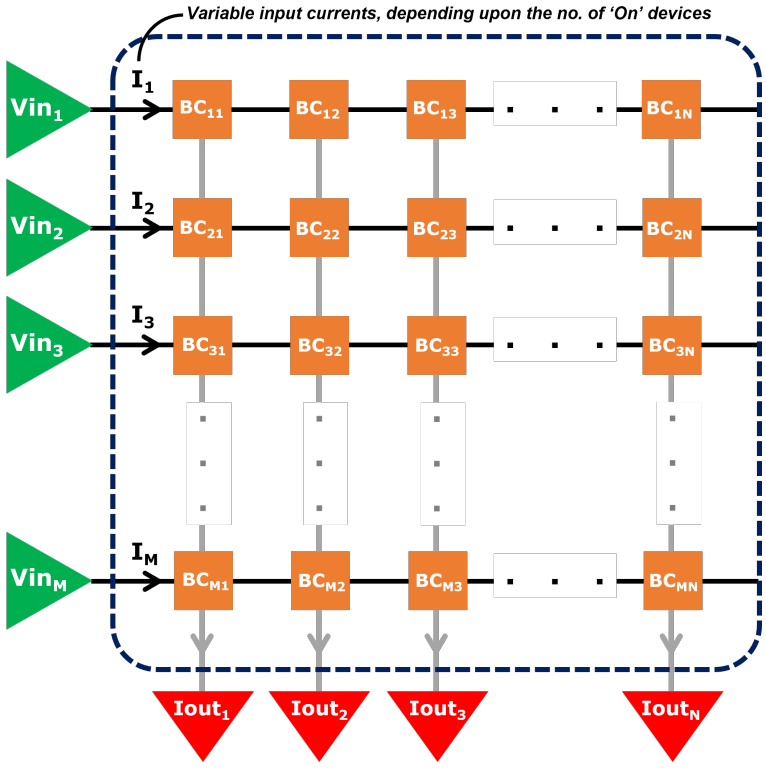


Figure 2-7: Implementation of Spiking Neural network using CIM

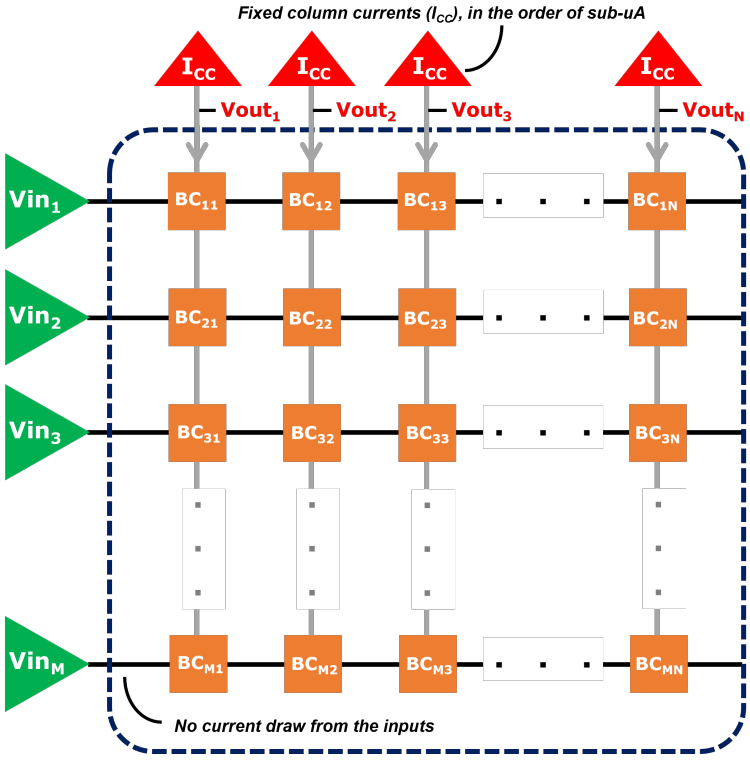
Owing to the binary structure of the inputs and outputs, SNNs may be mapped to a CIM crossbar in a manner similar to that of Binary Neural Networks. Likewise, the synaptic weights between two layers can be represented in the same way by the memristor conductances in the crossbar. The main difference lies in the activation dynamics of SNN, which employs the LIF model, to accumulate the induced charges at each time step of the inference cycle and fire an output spike once the membrane potential exceeds its threshold.
This can be emulated using the custom LIF circuit, proposed later in Chapter 3, which, as depicted in figure 2-7, is paired up with each crossbar column at its periphery. The current generated in a column, that is the result of a MAC operation between the input spikes and synaptic weights across that column, is integrated at each time step in the inbuilt capacitor of that column's LIF circuit. A comparator (part of the LIF circuit) is then used to give an output binary pulse or spike as soon as the voltage across the inbuilt capacitor exceeds the set threshold. Thus, each column, combined with its inputs, weights and LIF circuit, represents an artificial neuron. The column, within the final layer, which fires the maximum number of times in a given inference cycle, is considered to be the predicted output.

Proposed Design

Figure 3-1 illustrates the fundamental distinction between the standard approach and the novel approach to CIM proposed in this study. In contrast to the conventional crossbar, where the input voltages are the first operand and the output is in the form of current, the novel approach utilizes a fixed current established in the column as the first operand and obtains analogue voltages at the top of the column as the output.

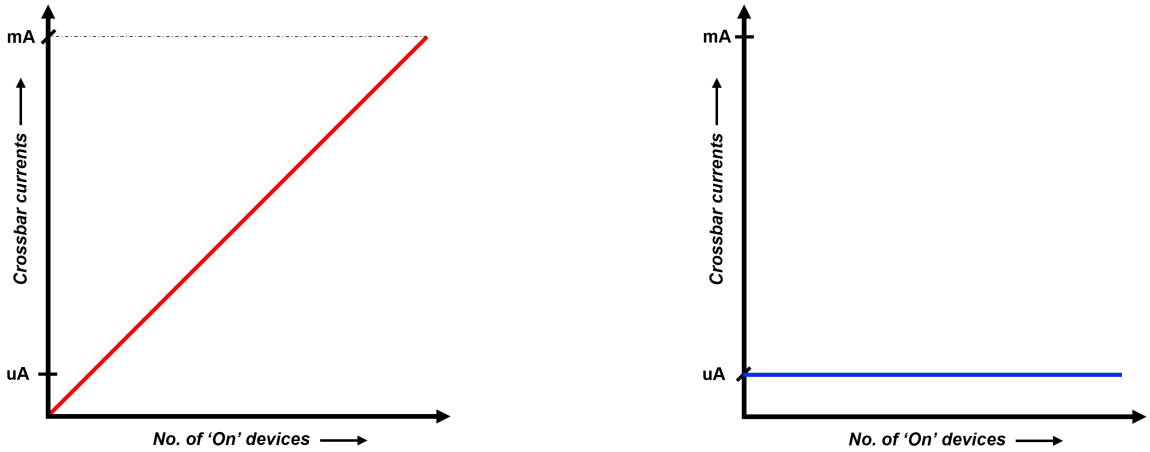


(a) Standard approach



(b) Novel approach

Figure 3-1: Illustration of the standard and novel approach to CIM



(a) Standard approach **(b)** Novel approach

Figure 3-2: Comparison of the crossbar currents in standard as well as novel approach to CIM

As seen in figure 3-2, the crossbar currents remain constant regardless of the number of 'on' devices, as opposed to the standard implementation, in which the currents increase as the number of 'on' devices increases. By maintaining the column currents in the sub-μA range, the currents in the novel crossbar can be limited in the μA range. This exhibits great potential in reducing the power consumption by orders of magnitude compared to the standard crossbar, where the crossbar currents are highly dependent on the input voltages and can accumulate in the range of mA.

3.1 Novel approach

Figure 3-3 illustrates the proposed novel CIM crossbar, in which each memristor is coupled with an NMOS to produce a single bit-cell. In addition, these bit-cells in a column are connected in series configuration rather than parallel configuration as in a standard CIM crossbar. Each column is driven by its constant current source, such that the current 'I', set by the current source, is the same across the entire column.
Assuming the NMOS is operating in the triode region and its "on" resistance is negligible in comparison to that of the memristors, the current through a bit cell either flows via the memristor or the NMOS, depending on the state of the NMOS. Considering a binary input , if the NMOS in a bit-cell is "off" ($V_{in(m)} = 0$), the current 'I' flows through its corresponding memristor, causing a voltage drop (v_{mn}) which, depending on its resistance state (R_{mn}), may be given using Ohm's Law as 3-1,

$$v_{mn} = I.R_{mn} \tag{3-1}$$

In contrast, if V_{in} is high ($V_{in(m)} = V_{DD}$) or the NMOS is ON, the current in the bit-cell would follow the path with least resistance and flow via the NMOS, resulting in a negligible voltage drop. This constitutes one multiply operation in which, depending on the state of the

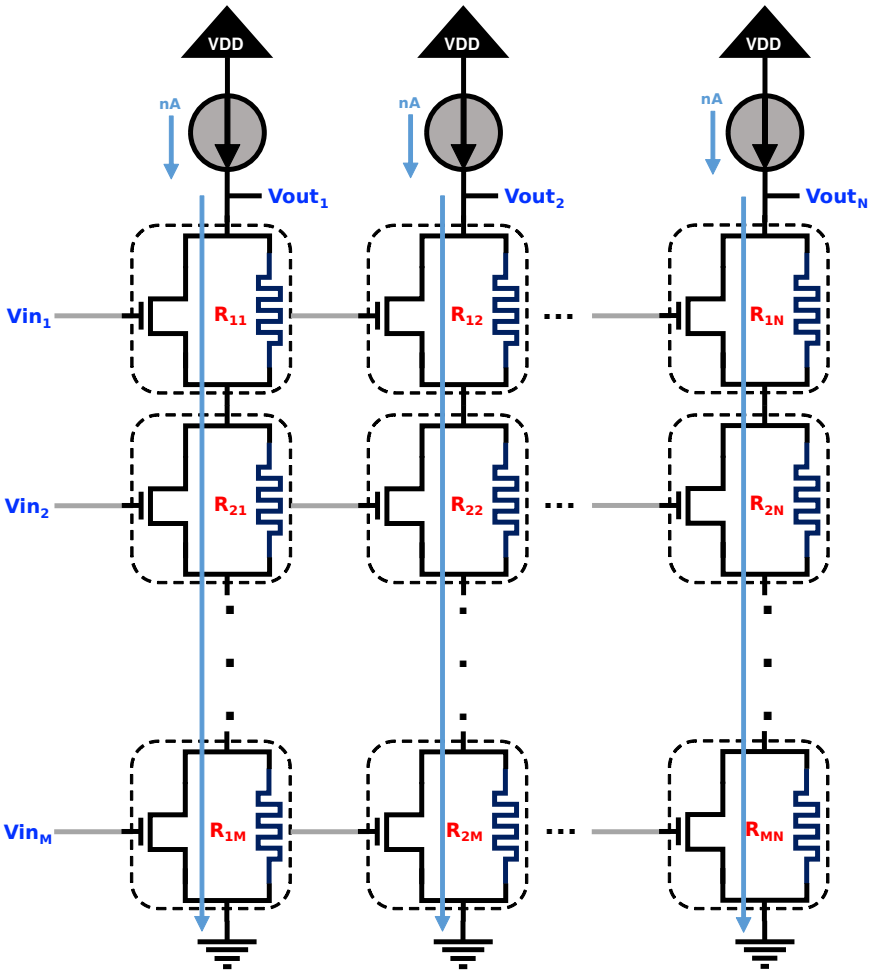


Figure 3-3: Illustration of the novel CIM crossbar depicting typical implementation of a VMM operation

input, the column current serves as the first operand, and the memristor's resistance serves as the second operand.

Since all of these bit cells are connected in series, the voltage drop across each bit-cell accumulates as a result, constituting an addition operation of M operands. As shown in equation 3-2, the final voltage at the output (V_n) of the n^{th} column is the result equivalent to one MAC operation between the inputs and the memristor resistances in the column. Consequently, this crossbar can perform a single VMM operation in the same manner as the standard CIM crossbar, assuming the inputs are binary.

$$V_n = (I.V_{in1}).R_{1n} + (I.V_{in2}).R_{2n} + \ \ldots \ + (I.V_{mn2}).R_{mn} \tag{3-2}$$

As observed in figure 3-3, if the column currents are maintained within the order of nA, the overall current in the crossbar may be confined within the range of μA, resulting in a power

reduction by one-to-two orders. Secondly, if the resistances of the memristor are maintained on the order of kilo ohm, the read voltages across the memristors may be decreased to the μV range, considerably reducing the phenomenon of read disturb and boosting the overall reliability. In addition, since the inputs are applied at the gate of the NMOS and each column is driven by its own current source, there is no coupling between any two column and the problem of parasitic currents is eliminated.

There are drawbacks associated with this method of implementation, which must be trade-off against the benefits received in power and reliability. Firstly, the massive chain of RC networks formed by several bit-cells arranged in series within a column increases the operation's latency and is also affected by the magnitude of the column current.
Secondly, this kind of crossbar requires an extra NMOS in addition to the memristor to perform the required computation, hence increasing the overall area of the circuit. Thirdly, it can only accept binary inputs since the signals are applied at the gate of an NMOS operating in the triode/linear region as a switch.
Finally, additional circuits may be required to convert the output voltage into suitable currents for driving subsequent stages, such as in the case of Spiking Neural Networks.

3.2 Design Overview

Figure 3-4: Overview of the proposed micro-architecture

As shown in 3-4, the design consists of three stages that, when integrated, provide the micro-architecture for Spiking Neural Networks (SNN) or Binarised Artificial Neural Networks (BNN).

- The first stage, known as the MAC unit, is the novel memristor crossbar that performs the MAC (Multiply-and-Accumulate) operation between the input spikes/binary inputs (first operand) and the synaptic weights (second operand) and outputs the result as an analogue voltage.
- The second stage, known as the V/I converter, senses the MAC unit's output voltage and transforms it into a linearly proportional current, which is then fed to the third stage.
- The third and final stage is the Leaky-Integrate and Fire (LIF) circuit, which accumulates the current generated by the V/I converter inside its membrane capacitor and generates an output spike when the threshold is met. In the case of Binary Neural Networks, the same can be modified to sample and hold as well as threshold the output voltages of a layer.

The entire operation constitutes a single inference cycle, and the total latency depends on the duration of each stage's settling time. Each stage's inputs and outputs are expected to have

a linear relationship, such that the final output is proportional to the number of "on" devices. "On" devices refer to bit-cells with column current flowing through their memresistive path.

3.3 Stage I: MAC unit

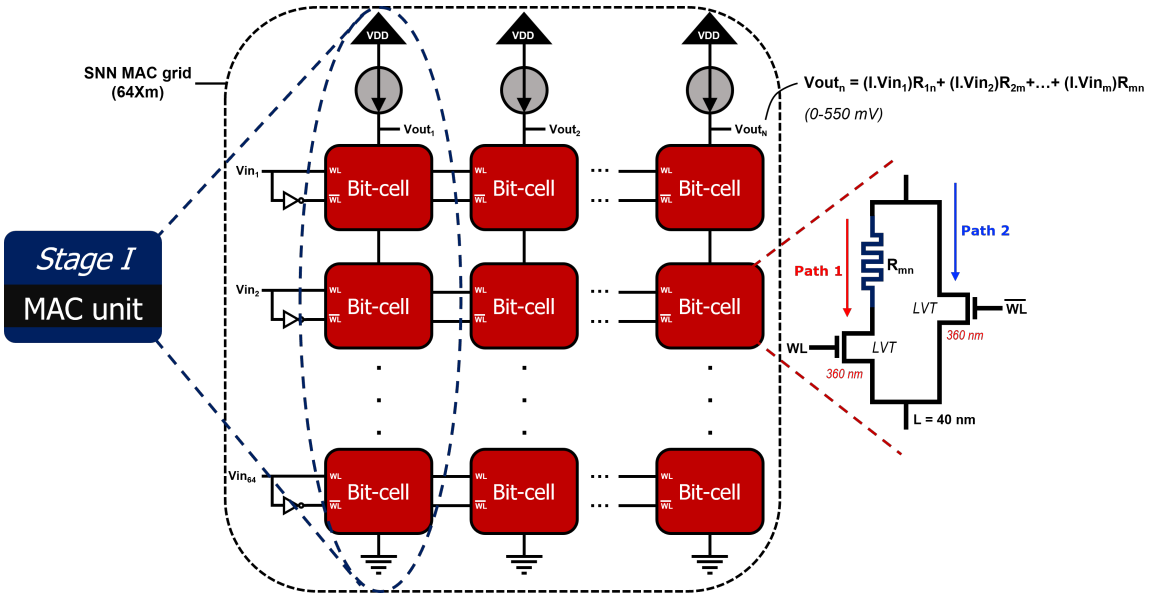


Figure 3-5: 1^{st} stage: MAC unit

The MAC unit is the novel crossbar column (refer 3.1), which is composed of 64 bit-cells stacked in series to represent 64 synaptic weights. As depicted in figure 3-5, the bit-cells assume a 2T1R configuration in which an RRAM is in series with an access transistor (NMOS), and their combination is in parallel configuration with another identical access transistor (NMOS). Since bit-cells receive complementary input signals applied at their NMOS gates, only one path in a bit-cell is active at a given time. When the bit-cells/devices receive a "high" signal at WL or the weighted (RRAM) path is active, they are considered to be "on," and vice versa.

The primary objective of employing the 2T1R configuration in a bit-cell is to counteract the non-negligible "on" resistances of the access transistors (NMOS). The presence of identical NMOS in both paths fixes the number of active access transistors in a column at any given time, since one of the paths is always active in a bit cell. Subsequently, 64 NMOS are always active in the proposed column, regardless of the input states. Consequently, the combined resistance posed by the active NMOS in the column remains nearly constant across the entire operating range and can be treated as a constant offset at the output. This helps maintain the linear output response of this unit with respect to the number of "on" devices at the expense of some area overhead. To further maintain the minimal "on" resistance of the access transistors, low V^t (high-speed) NMOS bit-cells are developed.

For the purpose of this work, we utilize the HfO_x RRAM model provided by JART (Jülich Aachen Resistive Switching Tools), which are currently the providers of one of the most accurate open source verilog-a models for RRAM [43]. Since the weights are binary for the scope of our work, the RRAM's are considered to only exist in one of its two stable high (20 k Ω) or low (2 k Ω) resistance state with a R_{off}/R_{on} ratio of 10.

3.3.1 Constant current source

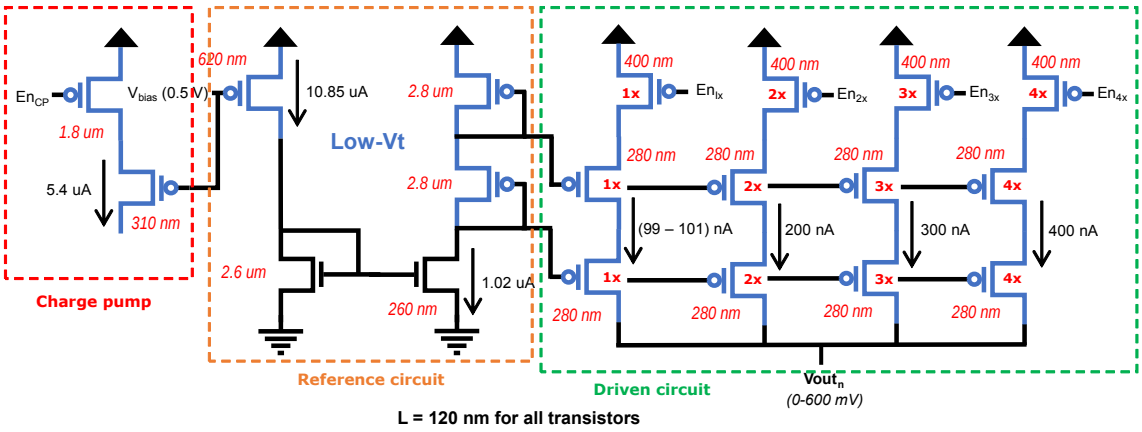


Figure 3-6: Schematic of the constant current source

As discussed earlier, it is necessary to establish constant currents in each column for this crossbar to function. Figure 3-6 depicts a constant current source that is capable of establishing a minimum fixed current of 100 nA through the column. It is divided into three main sections, with the reference circuit designed to generate a 10 uA of reference current from a bias voltage of 0.5 V, while the driver circuit operates in a sub-threshold regime to appropriately scale down the established reference current by a factor of 100, resulting in 100 nA of current. The driver circuit is repeated for each column, while the reference circuit is only repeated per crossbar size of 64×64. The current source is intentionally designed to operate in the sub-threshold regime to not only generate very low currents, 100 nA in this case, but also to provide maximum voltage headroom at the output, as the PMOS channel currents in this mode of operation are in an inversely exponential relation with the source-to-drain voltage (V_{SD}) across it. Given that the source voltage of the PMOS pair is fixed at 1.1 V, virtually no change in its drain voltage, which is also the output voltage of each column, affects the established channel currents. So long as the PMOS does not drive into cut-off mode, it will continue to maintain the column current over a wide range of output voltages (0-600 mV in this case).

In order to provide controllability, the driven circuit includes additional sources of current to step up the column current if the crossbar is not performing as intended at lower currents. The extra current sources are sized in the ratio 2:3:4 in comparison to the minimum sized current source in order to generate 200 nA, 300 nA, and 400 nA of current, respectively, when one of the extra current sources is activated via its switch. Thus, the current can be varied from 0 to 1 uA with a minimum step of 100 nA by appropriately activating the correct combination

of these sources. Note that the transistor lengths here are 120 nm, as opposed to the 40 nm technology node used in this work. It is done to reduce the impact of the channel length modulation effect in the reference circuit and to produce a more stable reference current.

3.4 Stage II: V/I converter

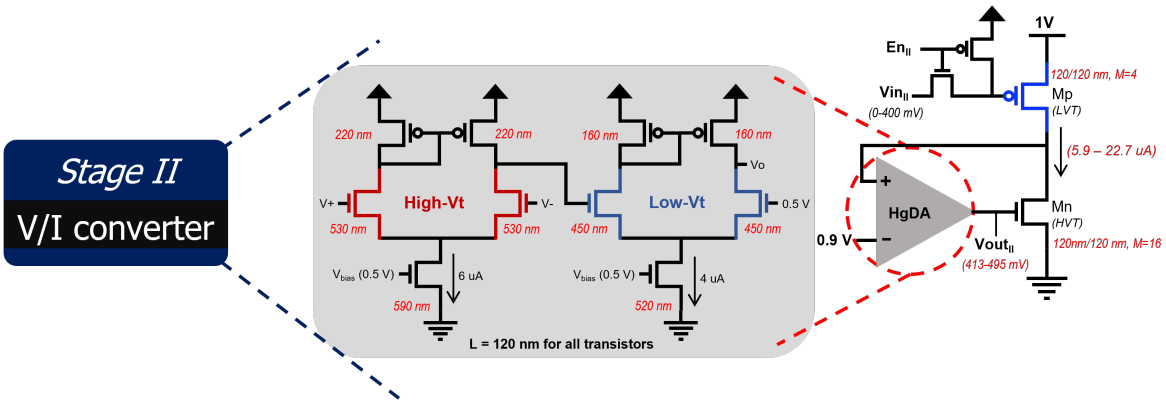


Figure 3-7: Schematic of the V/I converter

As the name suggests, the V/I converter is a linear voltage to current converter. Attributed to the fact that the newly proposed crossbar outputs the results of a MAC operation as an analogue voltage, it is necessary to convert this voltage output into a proportional current required to drive subsequent stages. The triode/linear region of the MOSFET is utilized to achieve the desired behavior and is a crucial component for the MAC unit. A PMOS source-to-drain current can be derived as follows:

$$I_D = \frac{\mu_p C_{Ox}}{2} M \left(\frac{W}{L}\right) (V_{SG} - |V_{Thp}|)^2 (1 + \lambda_p V_{SD}) \tag{3-3}$$

If the V_{SD} across the MOSFET is kept constant in some way, then the channel current (I_{DS}) is linearly proportional to the applied V_{SG}, as observed by the equation 3-3. Since the input voltage can range from 0 to 400 mV, PMOS is a suitable candidate for this application as it can be maintained in the triode/linear region with the given range of input voltage applied to its gate while its source node is connected to a fixed high voltage, in this case 1V. Now, the objective is to stabilize the drain voltage of the PMOS (M_p) at a fixed voltage, which results in a constant V_{SD} across M_p that is low enough to maintain its operation within the triode region for the given range of input voltage applied to its gate.

As seen in figure 3-7, this is accomplished by combining the input PMOS with an NMOS (M_n) in a common source configuration, where the M_n is driven via its gate by a high gain differential amplifier while its source is connected to GND. The amplifier operates in a feedback configuration, where its positive input continuously monitors the drain voltage of M_p. Due to the high open loop gain of differential amplifiers, it biases the NMOS such that its positive input, and in effect, the drain voltage of M_p follows its negative input and in turn, the

reference voltage (V_{ref}). Since a reference voltage of 0.9V is used in this case, it effectively results in a constant V_{SD} of 0.1 across M_p. In order to have a minimal offset between the inputs, the differential amplifiers employed in this application must have a high open loop gain.

This stage's operation is highly dependent on the functionality of the differential amplifier, which is designed to stabilise the drain voltage of M_p at 0.9V by appropriately biasing the control NMOS (M_n). However, NMOS based differential amplifiers are typically inefficient at an input bias voltage of 0.9, which results in a lower gain that is undesirable in this instance. A lower gain may result in a high offset between the feedback voltage and the reference voltage, thereby diminishing the functionality of the V/I converter. Thus, a two-stage differential amplifier is employed, with the first stage serving primarily to reduce the input bias from 0.9 V to 0.5 V at the expense of a lower gain, while the second stage is responsible for the main amplification. Utilizing high V_t NMOS at the inputs, the initial stage achieves this behavior. In contrast, the second stage maximizes gain by utilizing low V_t NMOS at the inputs. Subsequently, we achieve an overall gain of 87, which is sufficient for our application.

3.5 Stage III: LIF circuit

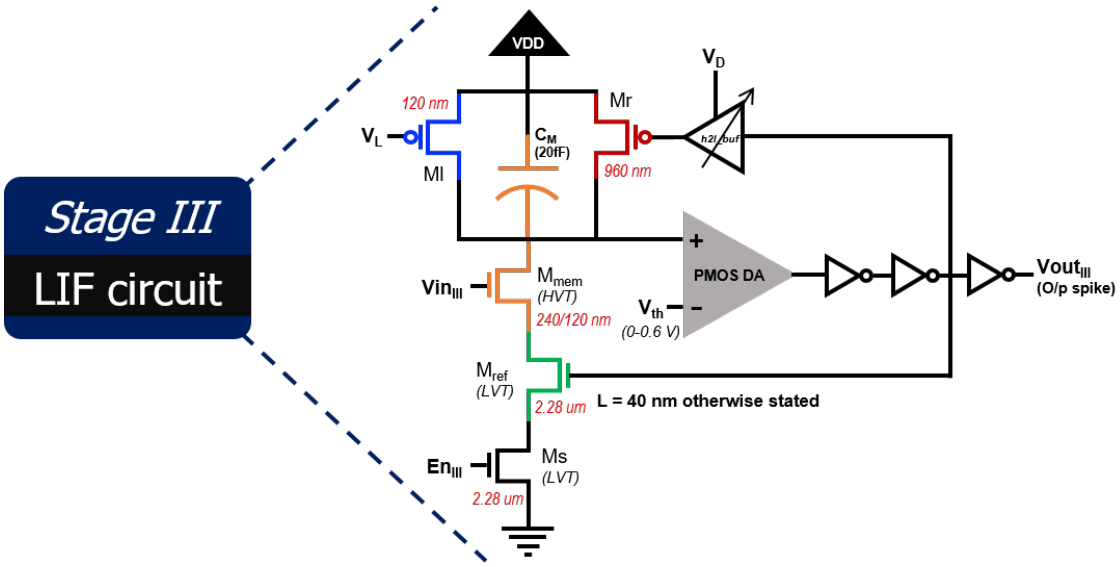


Figure 3-8: Schematic of the LIF circuit

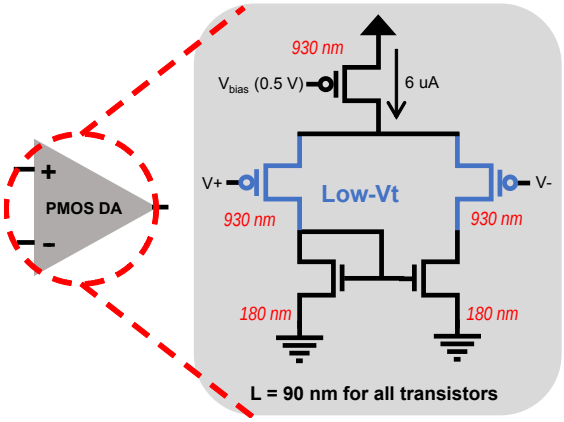


Figure 3-9: Schematic of the PMOS based DA used as a comparator

The Leaky-Integrate and Fire (LIF) circuit, as shown in figure 3-8, is a standard circuit for SNN that stores the computed MAC result as charge on a "membrane capacitor" and fires an output spike when the voltage across the capacitor reaches a predetermined threshold. Consequently, the LIF circuit requires a constant current input whose magnitude is proportional to the result of the MAC operation, which is then integrated over the capacitor for a fixed time interval. When a spike is fired, an internal feedback loop in the circuit triggers the reset switch after a certain refractory time, resetting the capacitor to its initial voltage. Variable refractory period is achieved by a ring oscillator with alternate inverters that have tunable V_{SS} which can be tuned to achieve a particular propagation time and, consequently, refractory delay. It also employs a refractory switch that isolates the LIF circuit for that particular time interval from the preceding stages, preventing the accumulation of incoming currents on the membrane capacitor. A modified D latch monitors the circuit at each time

step and stores the spike in the event of firing. This spike is retained and read until the beginning of a new cycle.

3.6 Additional circuitry

3.6.1 Charge pump

A large RC network comprised of multiple RRAMs and MOSFETS in series increases the latency and energy consumption of the proposed crossbar, causing it to settle to the desired voltage with considerable delay. This can be circumvented by employing a charge pump, as shown in figure 3-6 that emits very short pulses of high current (2.5-5 μA). These high current pulses enable the crossbar to rapidly converge to the expected value, thereby drastically reducing the latency.

3.6.2 Bidirectionality

This is a key feature of this architecture, since it allows columns to be read in either direction to address conductance variation due to read disturb. Conventionally, the columns in a crossbar are coupled to each other, which not only might result in the flow of parasitic currents in the event of an component mismatch, but also renders the circuit unidirectional, i.e., the direction of inputs and outputs cannot be altered. Since inputs are applied to the Word Line (WL) rather than the Source Line (SL), column operations are independent of one another. Moreover, since each column is driven by its own constant current source, it is simple to interchange the direction of column operation or column current flow by applying the same constant current from the opposite direction. Additionally, since the columns are isolated from one another, the issue of parasitic currents is mitigated.

Results

4.1 Simulation setup

Parameters	**Specifications**
RRAM Device	HfO_x[43]
R_{off}/R_{on}	20KΩ/2KΩ
Read Current	100 nA
Number of devices per column	64
CMOS Technology	40 nm TSMC
Process Corner	Typical
Temperature	27°C

Table 4-1: Simulation setup

Table 4-1 lists the design specifications used for our analysis. Simulations are performed using the HfO_x based RRAM model, provided by Jülich Aachen Resistive Switching Tools (RWTH Aachen University)[43] and assembled in 2T1R configuration. For our analysis, the RRAM model only needed to store a single bit of data and was modified accordingly to exist either at 2 KΩ (low resistive state representing binary '0') or 20KΩ (high resistive state representing binary '1') with a $R_{off}/R_{on} = 10$. The simulations were performed using the TSMC 40nm technology node, where each crossbar column consists of 64 devices.

4.2 Simulation results

4.2.1 Stage I: MAC unit

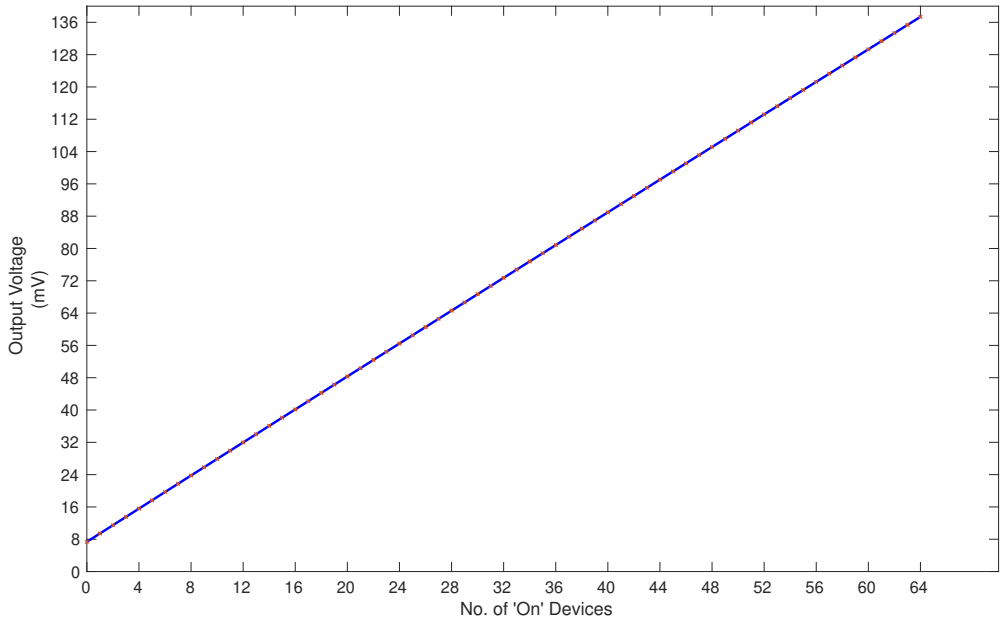


Figure 4-1: DC operating point plot of V_{out} (Output Voltage) vs Number of 'On' devices (bit-cells) in a MAC unit column

As seen in figure 4-1, the output voltage of the MAC unit is proportional to the number of 'On' devices, as anticipated. Since each bit-cell has an NMOS in both paths, the cumulative voltage drop due to the access transistors in the column remains nearly constant for any state of the input. Thus, the 2T1R configuration helps in maintaining the linearity of the crossbar output and can be treated as a constant offset error.

Figure 4-2 represents the transient behaviour of the MAC unit for a different number of 'On' devices. As observed, the output voltages for every state of the input settle by 500 ns, suggesting a latency of 500 ns for the MAC unit column without including a charge pump.

Figure 4-3 represents the transient behaviour of a MAC unit column using a charge pump. Initially, two pulses of high current (2.5 uA) were applied, with a pulse width of 5ns and duty cycle of 12%. Consequently, the latency of the MAC unit column is reduced by order of magnitude to 50 ns indicating the significance of the charge pump. The settling of output voltages at a different number of 'On' devices can be seen distinctively in figure 4-4 where the transient behaviour of the MAC unit column is shown from 45 ns to 50 ns. As expected, the voltages safely settle by 50 ns, indicating a latency of 50 ns for the MAC unit column with a charge pump.

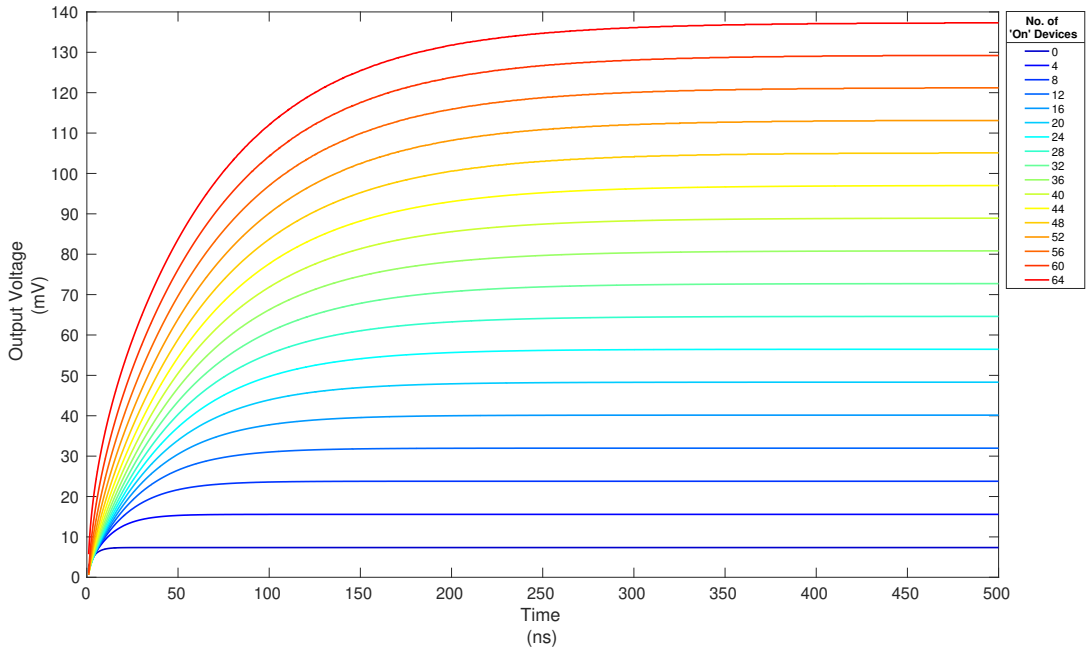


Figure 4-2: Transient plot of V_{out} (Output voltage) vs time for different **Number of 'On' devices** (bit-cells) in a MAC unit column without the usage of the charge pump

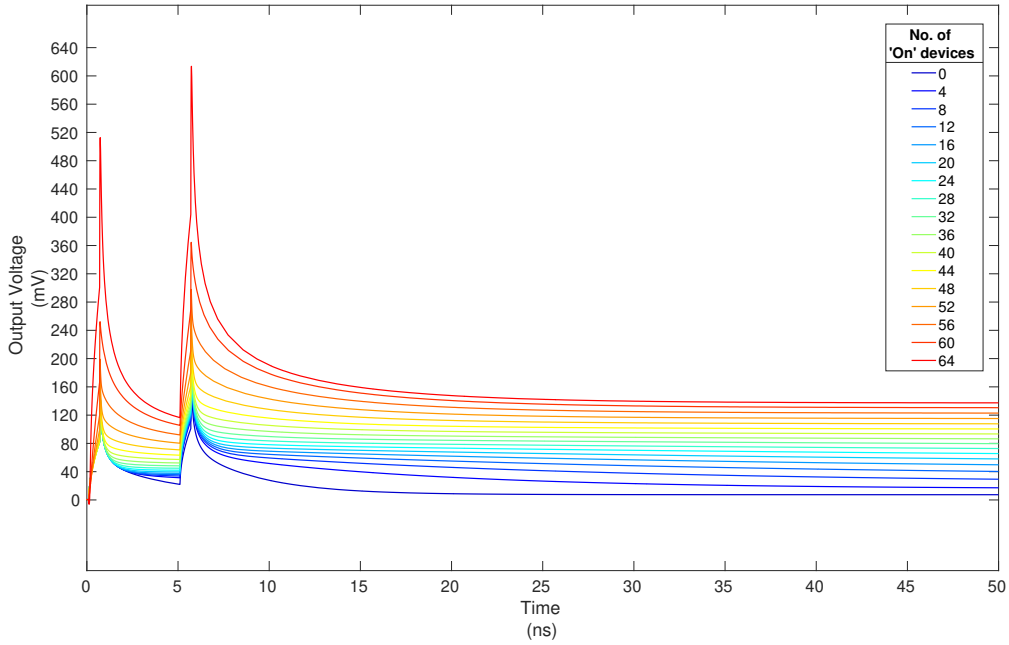


Figure 4-3: Transient plot of V_{out} (Output voltage) vs time for different **Number of 'On' devices** (bit-cells) in a MAC unit column with the usage of the charge pump

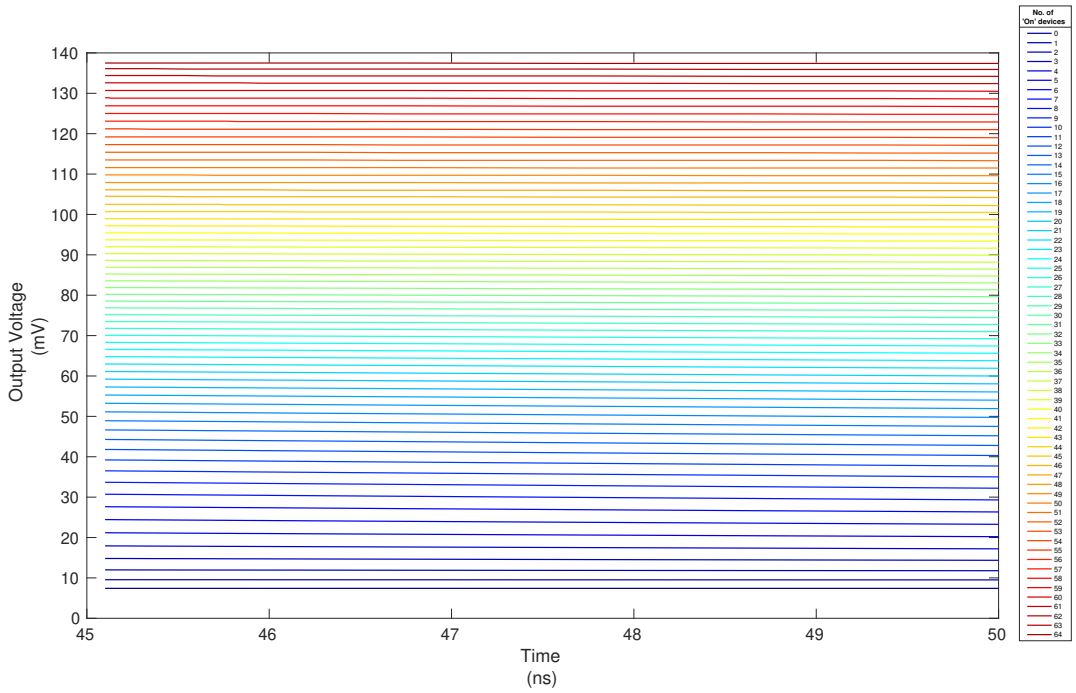


Figure 4-4: Detailed transient plot of V_{out} (Output voltage) vs time from 45 ns to 50 ns for different **Number of 'On' devices** (bit-cells) in a MAC unit column with the usage of the charge pump

4.2.2 Stage II: V/I converter

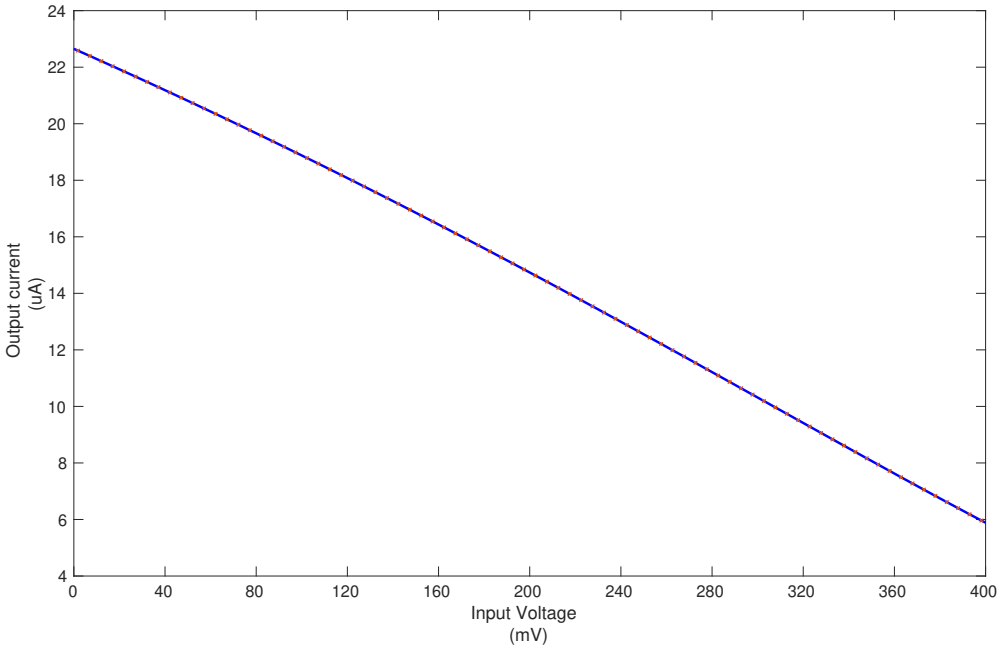


Figure 4-5: DC operating point plot of I_{out} (Output current) vs Vin_{II} (input voltage) in a V/I Converter

Figure 4-5 represents the curve of the final output current (I_{out}) with respect to the input voltage (Vin_{II}) applied at the gate of the PMOS (M_p) of the V/I Converter. As expected, the output current is quite linear to the applied input voltage, which is crucial to converting the output voltage of the 1^{st} stage (MAC unit) into a proportional current that is further required to drive the third stage.

Figure 4-6 represents transient behavior of V/I converter at different input voltages. As seen, all the output currents settle by 10 ns and hence, the latency of the V/I converter can be considered to be 10 ns. It is more clearly visible in figure 4-7, which depicts the generated output currents from 10 ns to 14 ns. As expected, the output currents remain stable for the particular duration verifying the claimed latency of 10 ns.

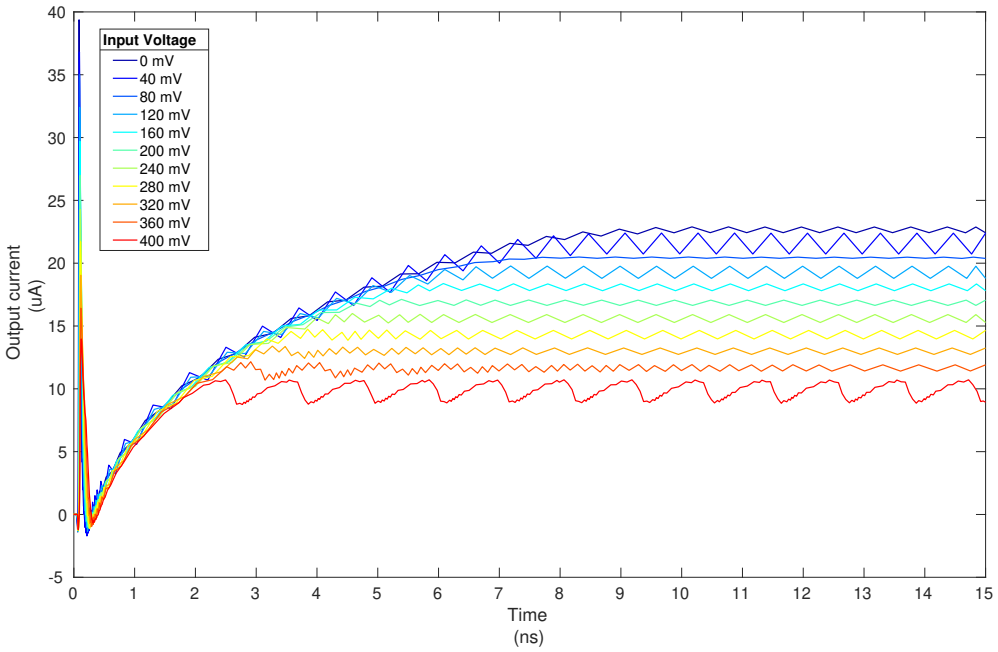


Figure 4-6: Transient plot of I_{out} (Output current) vs time at different input voltages (V_{in}) in a V/I Converter

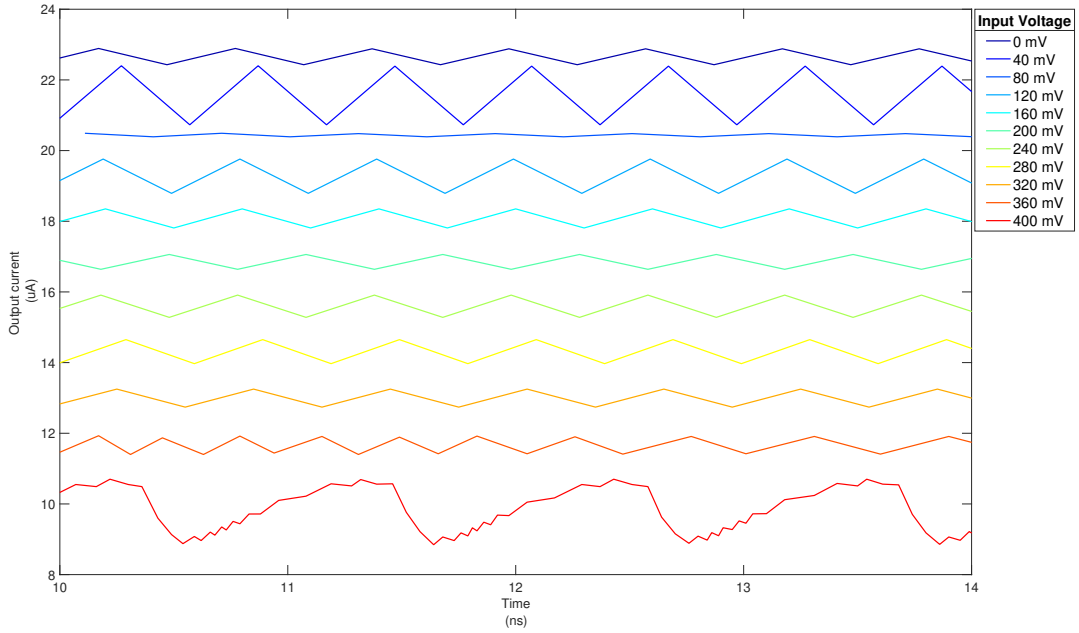


Figure 4-7: Detailed transient plot of I_{out} (Output current) vs time from 11 ns to 15 ns at different input voltages (Vin_{II}) in a V/I Converter

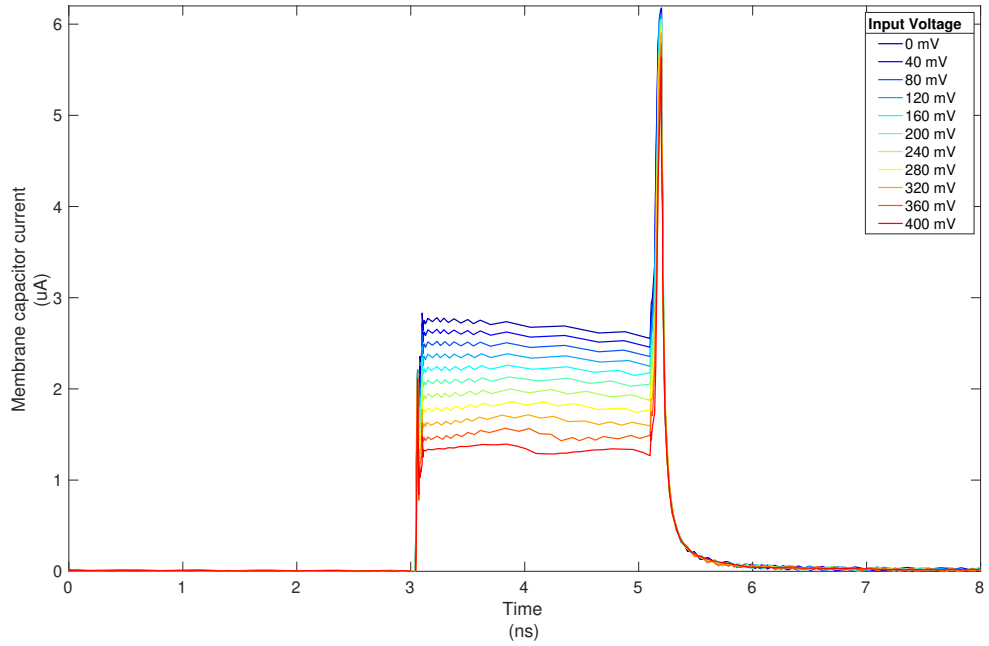


Figure 4-8: Transient plot of I_c (Membrane capacitor current) vs time at different input voltages (Vin_{II})

The currents generated by the V/I converter are copied to the LIF circuit via the current mirroring technique, and the resulting behaviour can be seen in figure 4-8. As expected, the discharge currents generated for the membrane capacitor are almost constant for the duration it is active, and thus the capacitor is discharged at a constant rate. This can be further confirmed by figure 4-9, which depicts the transient voltage across the membrane capacitor, discharged at different input currents for a fixed time interval. As seen, the voltage reduces linearly as expected and stabilises at a particular voltage level after the discharge stops, depending upon the magnitude of the current, which in turn, is governed by the input voltage to the V/I converter.

Figure 4-11 and 4-10 depicts the curve of the Control voltage ($Vout_{II}$) and Feedback voltage (Voltage at the drain of M_p) of the V/I converter respectively. As anticipated, the differential amplifier applies a bias on the NMOS (M_n) so that the drain voltage of M_p/M_n remains stable at 0.9 V, irrespective of the input voltage applied to the V/I converter. This is crucial for the operation of the V/I converter in order to keep the input PMOS (M_p) biased to the triode region and ensure a linear variation of channel currents (I_D) w.r.t to its gate voltage.

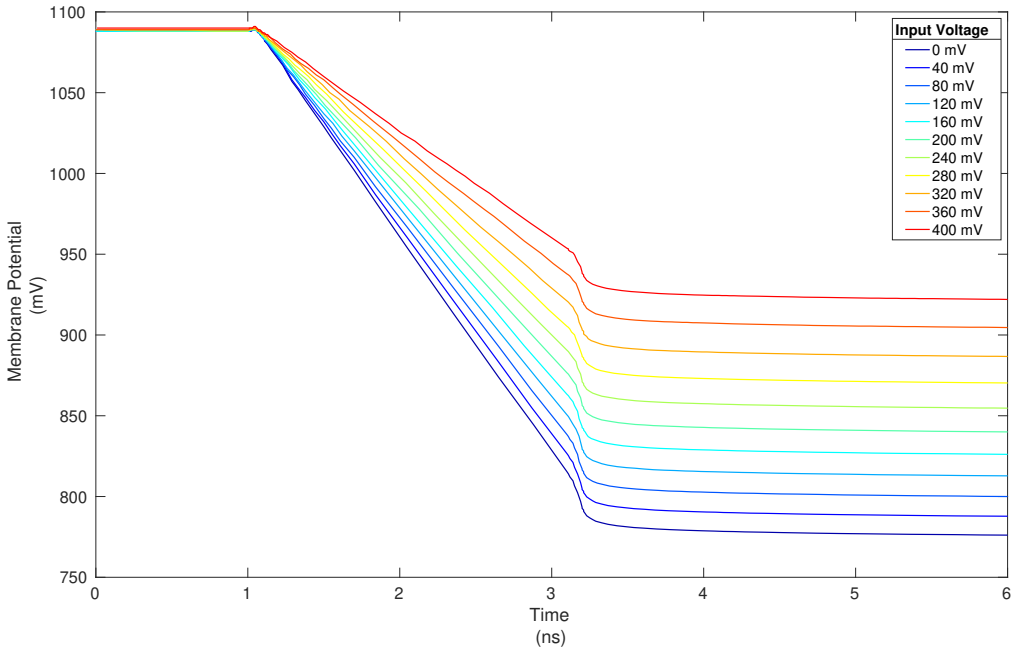


Figure 4-9: Transient plot of V_c (Membrane potential) vs time at different input voltages (V_{in})

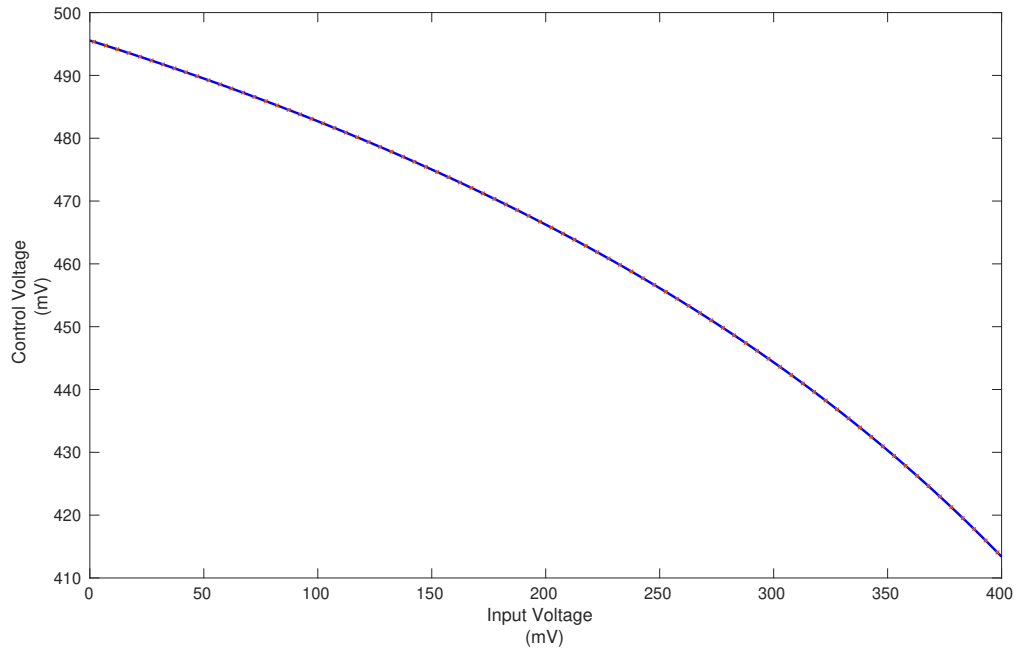


Figure 4-10: DC operating plot of V_{amp} (Control voltage) vs V_{in} (input voltage) in a V/I Converter

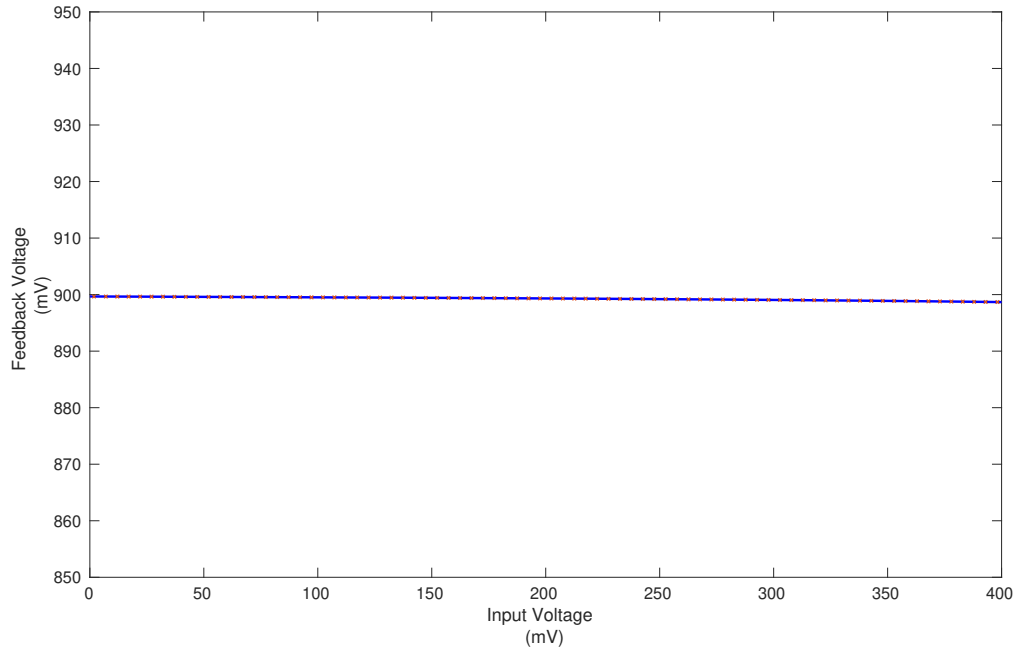
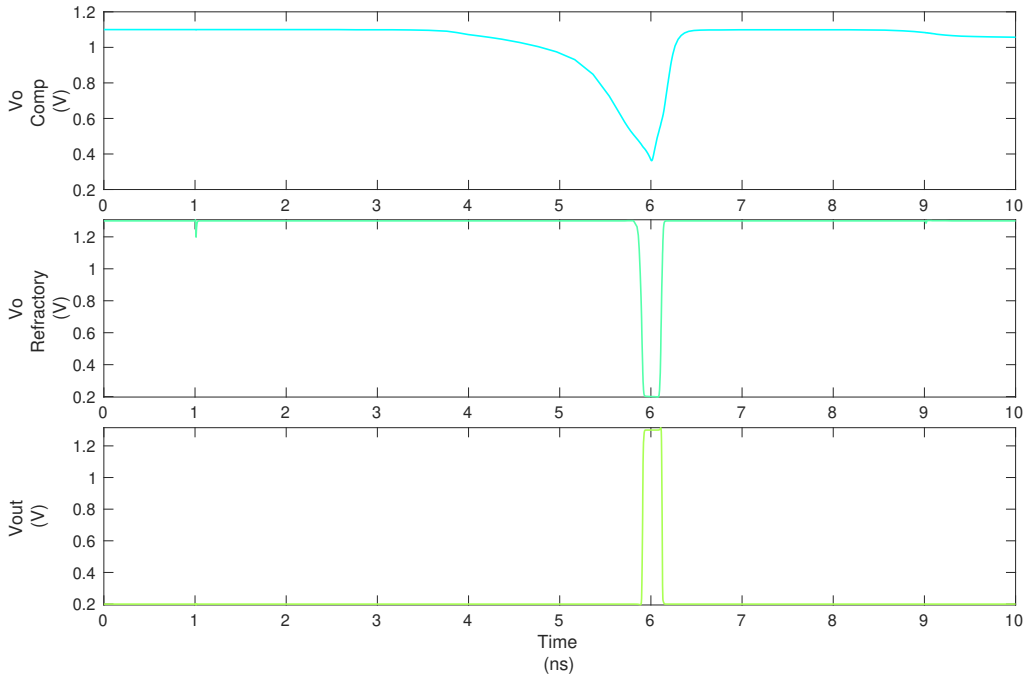


Figure 4-11: DC operating plot of V_f (Feedback voltage) vs V_{in} (input voltage) in a V/I Converter

4.2.3 Stage III: LIF circuit



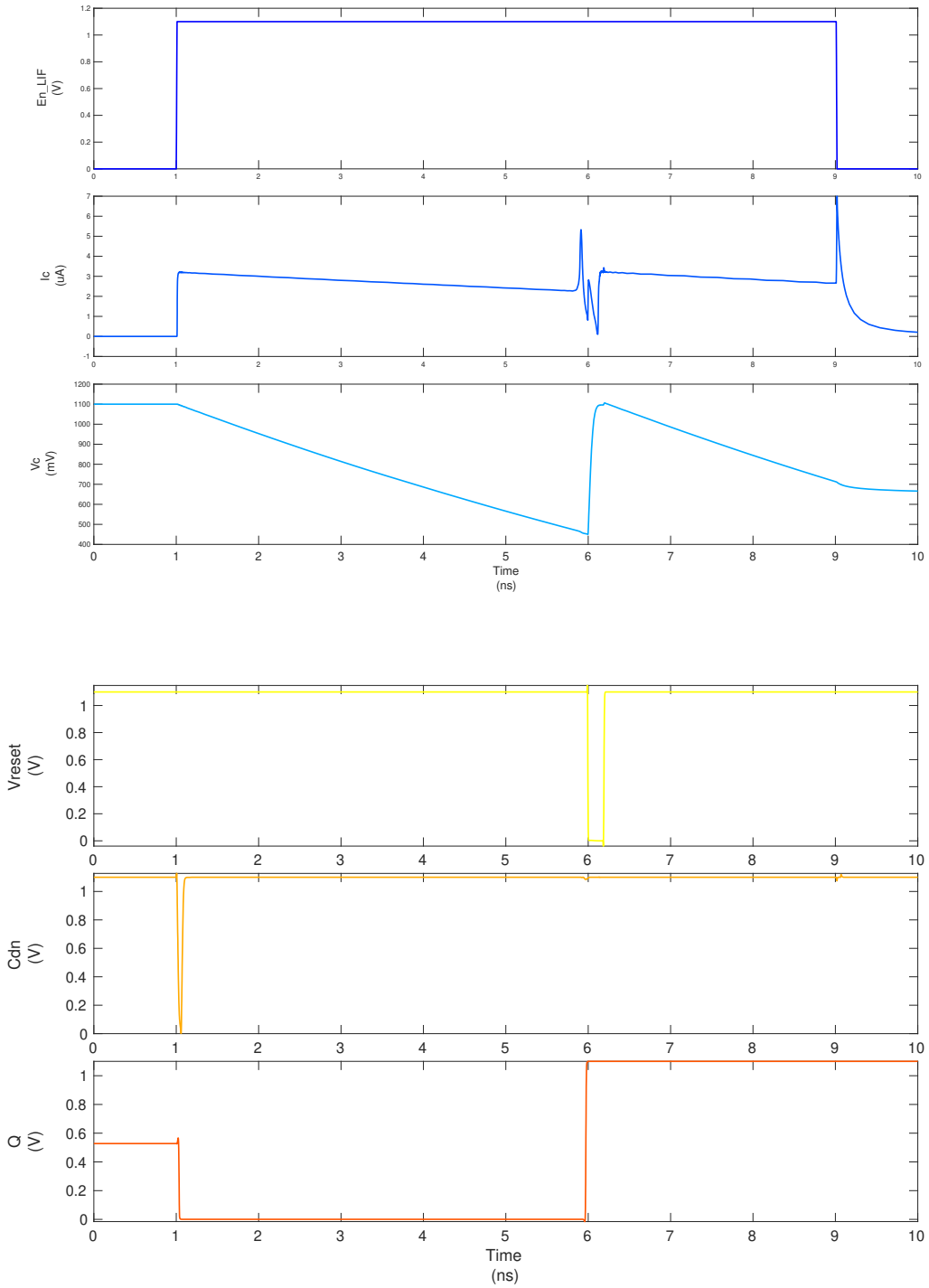
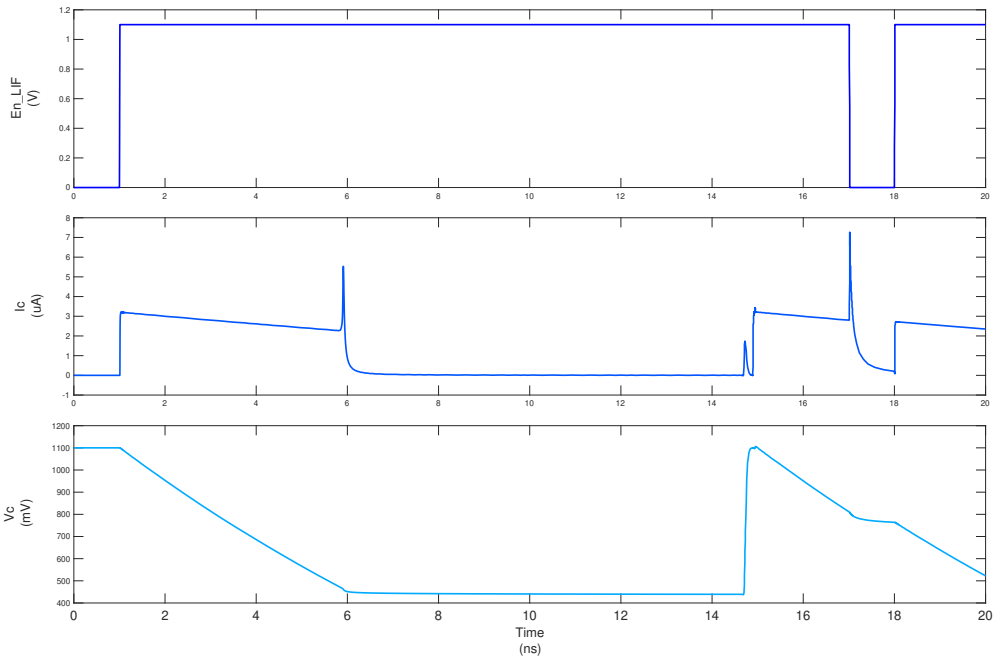


Figure 4-14: Comprehensive transient plots of LIF circuit vs time with no refractory delay i.e. $V_D = 1.1V$

Figure 4-14 depicts the flow of the LIF circuit starting from the Enable signal (En_{III}) to the latching of the output spike (Q) in the event of firing. As seen when the enable signal switches to high at one ns, a constant current (I_C) is generated across the membrane capacitor

depending upon the input voltage to the V/I converter. Consequently, the membrane capacitor (C_m), precharged to V_{DD} (1.1V), begins discharging at a constant rate and continues to discharge until it reaches the set threshold voltage of 0.5 V. On reaching the threshold voltage, the output of the comparator (Vo_{comp}), initially at a high state, transitions to a low state and this signal is further strengthened by the chain of inverters at the output of the comparator.

The output of the second buffer ($Vo_{refractory}$) transitions from a high to a low state, which not only disables the membrane cap discharge but also triggers the reset PMOS (V_{reset}) to charge/reset the membrane cap back to V_{DD}. Moreover, the third inverter (V_{out}) transitions from a low to a high state, which represents the generated output spike, as a result of a firing event, and is latched into the custom D latch (Q). After the cap voltage is reset to V_{DD}, the output of the comparator transitions back to the high state, re-enabling the cap current (I_C), if any, and hence, normal LIF operation while the output goes back to the low state.



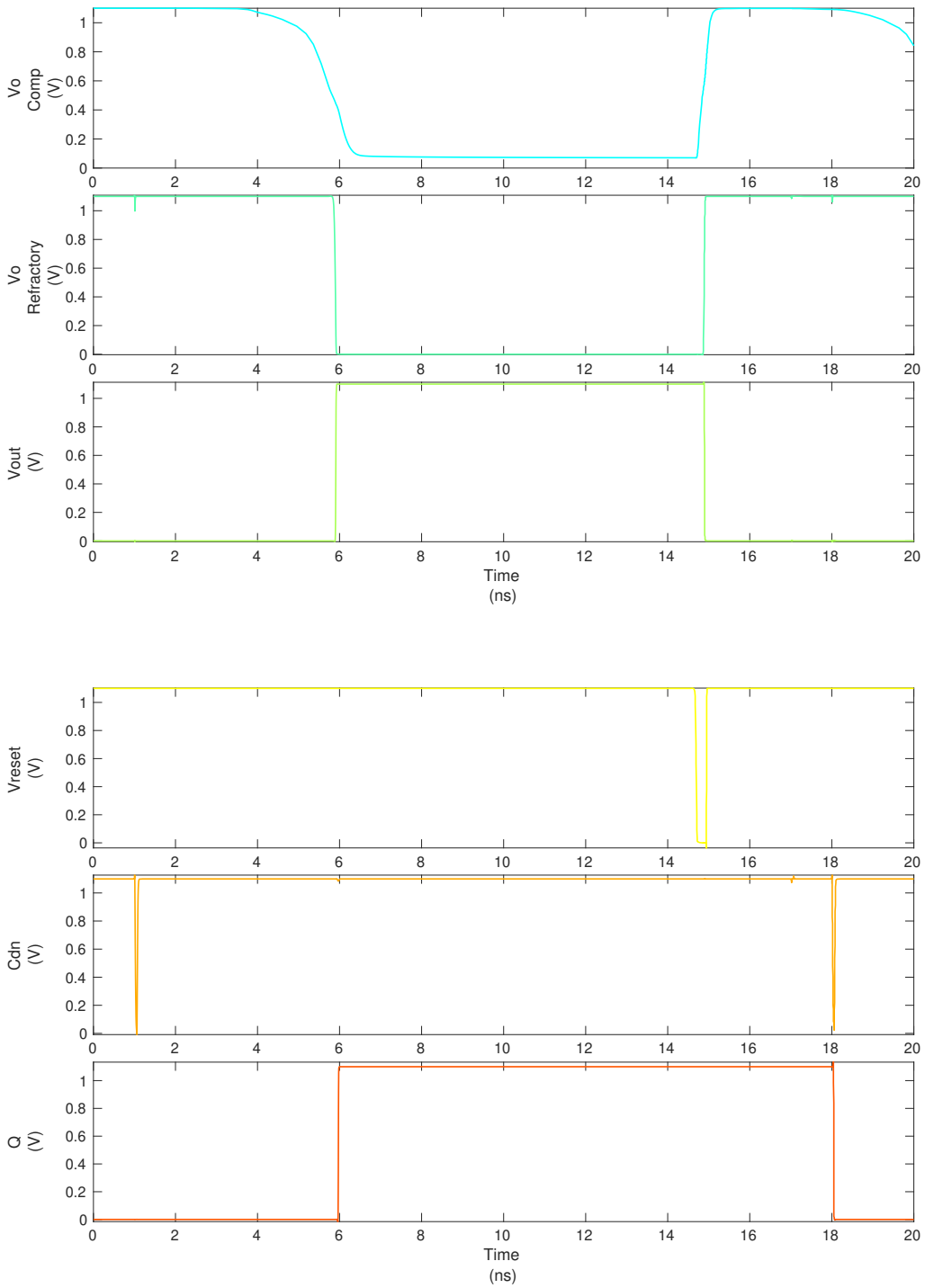


Figure 4-17: Comprehensive transient plots of LIF circuit vs time with refractory delay of about 8.5 ns i.e. $V_D = 0.6V$

Figure 4-17 depicts the flow of the LIF circuit with **refractory behaviour** starting from the onset of Enable signal (En_{III}) to the latching of the output spike (Q). When the enable signal switches to high at 1ns, a constant current (I_C) is generated across the membrane capacitor

depending upon the input voltage to the V/I converter. Consequently, the membrane capacitor (V_C), precharged to V_{DD} (1.1V), begins discharging at a constant rate and continues to discharge until it reaches the set threshold voltage of 0.5 V. On reaching the threshold voltage, the output of the comparator (Vo_{comp}), initially at a high state, transitions to a low state, and this signal is further strengthened by the chain of inverters at the output. The output of the first buffer ($Vo_{refractory}$) transitions from a high to a low state, which not only disables the membrane cap discharge but also goes on to trigger the reset PMOS (V_{reset}) to recharge the membrane capacitor back to V_{DD}.

However, in this case, the high-to-low trigger signal from the output of the second inverter ($Vo_{refractory}$) reaches the gate of the reset PMOS (V_{reset}) with a delay of almost 8.5 ns, set by changing the V_D to 0.6 V. Thus, the circuit remains disabled until the reset mechanism is triggered, preventing any accumulation of charges for the time being. On the other hand, the third inverter (V_{out}) transitions from low to high state and retain its state until the refractory period is over. However, since this output signal is latched at its rising edge, the length of the spike does not affect the operation of the custom D latch (Q). After the end of the refractory period, the capacitor voltage is reset to V_{DD} causing the output of the comparator to transition back to the high state, re-enabling the cap current (I_C), if any, and hence, normal LIF operation.

4.3 Comparison

4.3.1 Software setup

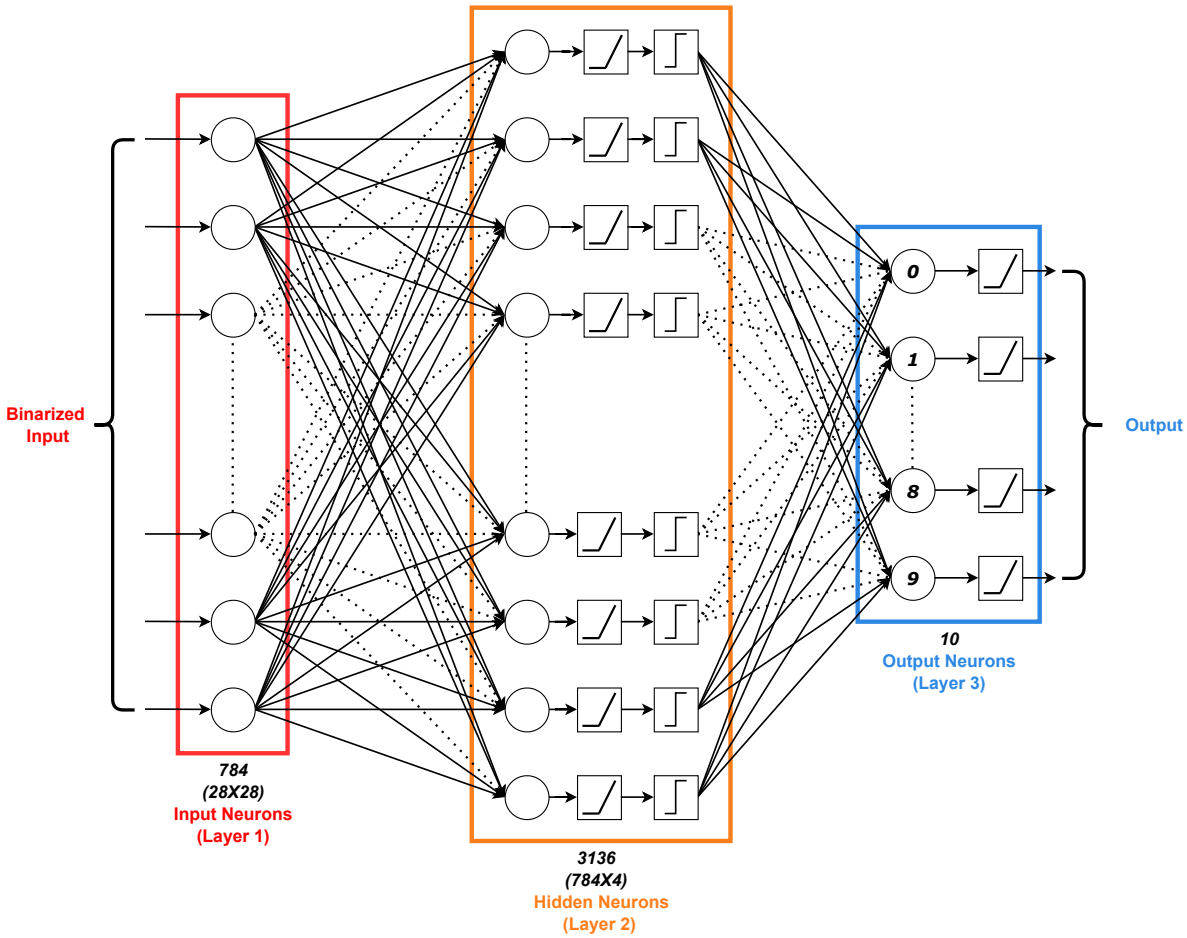


Figure 4-18: Proposed ANN model for comparison

Using the PyTorch library [44], A binary Artificial Neural Network (ANN) is developed over MNIST dataset [22] in order to compare the state-of-the-art and proposed hardware for ANN implementation. As seen in Figure 4-18, the network has three fully connected layers where,

- The first layer, i.e., the input layer, consists of 784 inputs, representing the binarised pixel values of an MNIST image of size 28×28.
- The hidden layer consists of 3136 neurons, and the output of this layer passes through the RELU function. The resulting outputs are further binarised using a custom threshold function, and the binarised outputs will then be used as input to the third layer, also the output layer.
- The output layer consists of 10 neurons representing ten output classes in the MNIST dataset pertaining to 10 digits from 0-9. Similar to the previous layer, the final layer also passes through the RELU function, where the neuron achieving maximum output is the predicted output class/digit.

The network is trained over binary weights (0.02/0.002) with a high-to-low ratio of 10. This is done to mimic 1-bit RRAM and keep the system compatible with the presently available RRAM technology. Since the inputs, weights, and outputs are binary, the threshold function is approximated as a hard-sigmoid function during training to ensure the availability of valid gradients. The network is trained over the MNIST training dataset, divided into 128 batches using Adam Optimiser.

After training for three epochs, the test accuracy achieved by the network is approximately 88.4%. The main reason for training this binary neural network is to keep it compatible with the proposed crossbar so that it can be used later to draw comparisons.

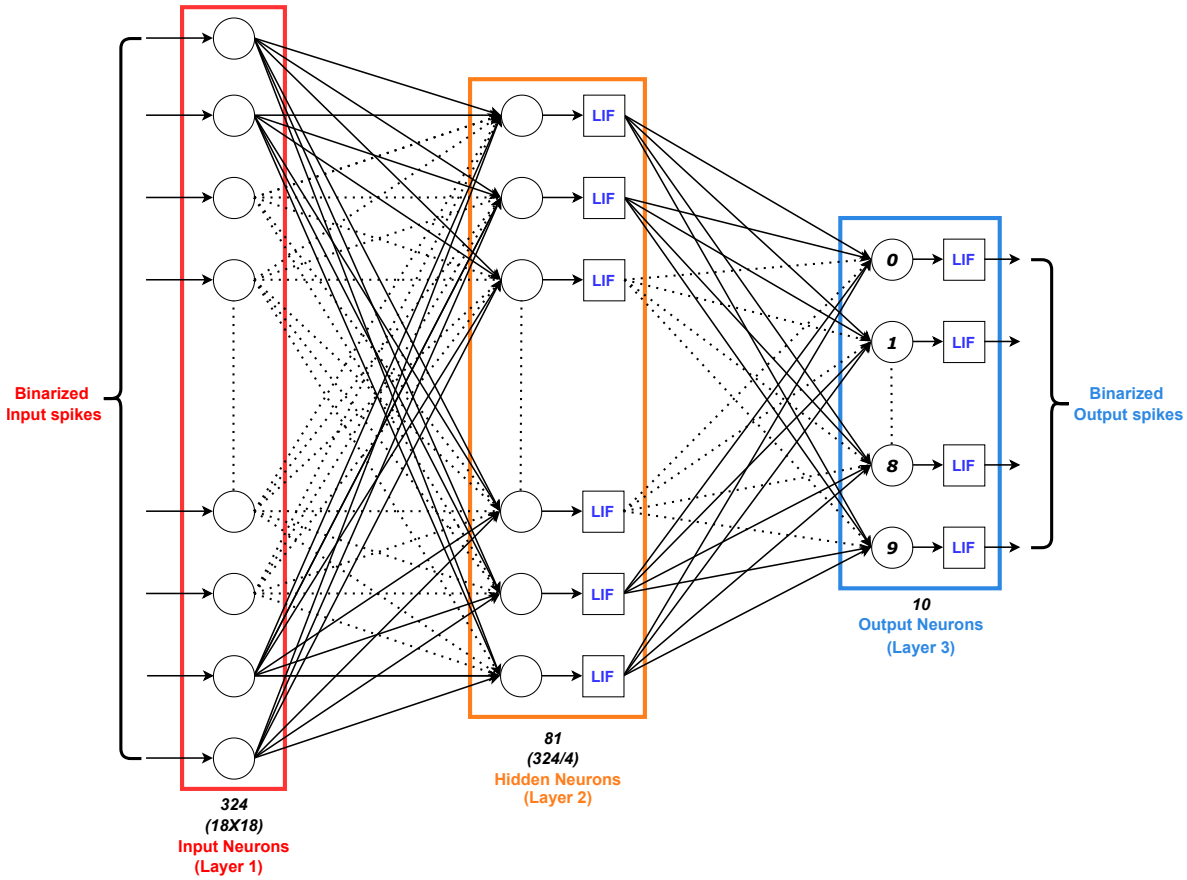


Figure 4-19: Proposed SNN model for comparison

A Spiking Neural Network (SNN) is constructed over the MNIST dataset [22] using the Snntorch library [45] in order to compare the state-of-the-art and proposed hardware for SNN implementation. As seen in Figure 4-19, the network has three fully connected layers where,

- The first layer, also known as the input layer, consists of 324 inputs, representing an MNIST image of size 18×18.
- The hidden layer consists of 81 neurons followed by the Leaky-Integrate-and-Fire (LIF) activation, which retains the layer's output at each time step and fires a binary spike

on reaching the set threshold, which in this case is 0.6.
- Similarly, the third (output) layer consists of 10 neurons, and the LIF activation has a firing threshold of 0.5, representing the ten output classes corresponding to MNIST digits.

In order to make the input image compatible with the proposed Spiking Neural Network, each pixel of the image must be converted into a relevant train of spikes that spans over the total inference time steps. This is accomplished using a rate encoded scheme (refer section 2.3), while the neuron in the output layer firing maximum times over the course of inference is considered to be the predicted class.

While trained on the same binary weights (0.02/0.002) as before to mimic 1-bit RRAM and keep it compatible with the presently available RRAM technology, the network is set to carry out inference in 60-time steps. To ensure the availability of valid gradients during the training process, the thresholding operation in LIF activation is approximated by a surrogate fast sigmoid function [45] while the thresholding function for weights employs the same hard-sigmoid function as used in ANN training. The network is trained over the MNIST training dataset, divided into 128 batches using Adam Optimiser. After training the network for three epochs, the test accuracy achieved by the network is approximately 87.2%.

4.3.2 Hardware mapping

The trained neural network models are mapped to various hardware systems that use memristor crossbars at the core to carry out MAC operations. The common blocks (colored blue) are taken from [19] and their relevant parameters are mentioned in the table 4-2.
The comparisons are made between the state-of-the-art conventional crossbar proposed in [16] and the novel crossbar proposed in this work. Unlike the novel crossbar where the inputs are given at the word line (WL), the inputs in the conventional crossbar proposed in [16] are given at the source line (SL), which, in combination with the RRAM weights, leads to a flow of current in the bit line (BL) of each column that is analogous to a vector-vector multiplication.

However, the work in SRIF [16] modifies the operation of a standard crossbar to enable low-power operation. It proposes a custom Sample+Hold (S+H) circuit that stabilises the BL voltage to 0.6 V. In comparison, 0.7 V is used as binary high at the SL, effectively minimising the read voltage to 100 mV across each RRAM instead of V_{DD} (1.1 V) in the standard implementation. This makes it suitable for comparison with this work, as both works target low power and low energy implementation.

Component	Params	Spec	Power (μW)	Latency (ns)	Area (μm^2)
Input Register (IR) [19]	Size	256B	230	25	770
Sample and Hold (S+H)[19]	Unit	1	0.0097	0.833	0.039
ADC [46]	No. of Output Bits	8	3060	0.833	1500
SRIF based Standard Crossbar [16]	Size	64×64	SNN:(L1-L2) - 12700 SNN:(L2-L3) - 3900 ANN:(L1-L2) - 22900 ANN:(L2-L3) - 11500	4.5	136.67
SRIF S+H [16]	Unit	1	6.6	4.5	30.22
MAC Unit (based on novel crossbar)	Size	64×64	19.14	50	367.56
V/I Converter	Unit	1	26.69	10	29.78
LIF Circuit	Unit	1	6.6	2	86.8

Table 4-2: List of component specifications

Artificial Neural Network(ANN)

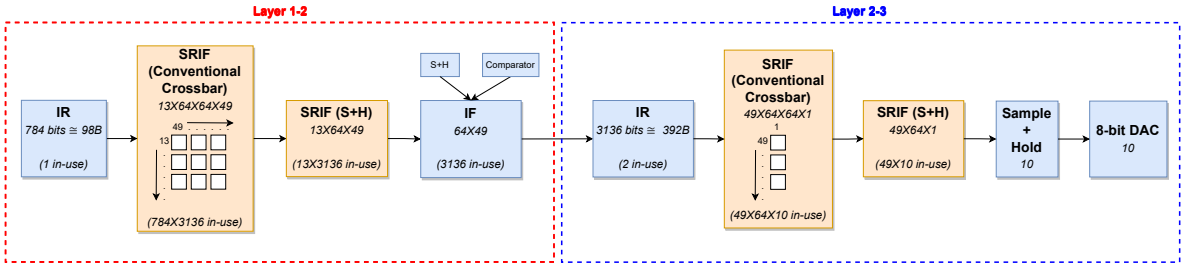


Figure 4-20: Proposed hardware setup for implementing the ANN model using conventional crossbar

As seen in figure 4-20 the input bits are stored in a standard input register (IR), as used in ISAAC [19]. These bits are parallelly transferred to a grid of conventional crossbars [16], each being 64×64 in size, having their cumulative weights programmed according to the learnt weights after training the model shown in 4-18. The grid size depends upon the total number of inputs and outputs for the particular layer and has been sized accordingly, as mentioned in the figure 4-20. The resulting currents are then sampled and held using the proposed LIF

circuit that is modified to prevent the leaky behaviour. Using the in-built comparator, the sampled voltage is further binarised against the learnt threshold voltage. This completes one cycle from layer 1 to layer 2.

The binary outputs from the 1^{st} cycle is now stored in a separate IR for the next cycle. Again, the bits stored in the input registers are parallelly transferred to a grid of a standard crossbar proposed in [16], and the resultant currents are sampled and held using a custom Sample and Hold circuit in [19]. Finally, the resulting voltages are converted into 8-bit digital values using an 8-bit DAC, proposed in [46], which are further available for post-processing. This completes one cycle from layer 2 to layer 3.

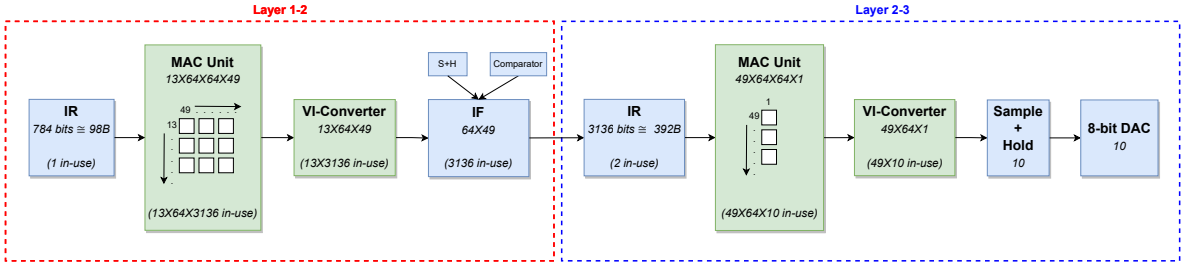


Figure 4-21: Proposed hardware setup for implementing the ANN model using novel crossbar

In the setup shown in figure 4-21, the flow of implementation remains the same, with several blocks (coloured blue) being reused from the previous setup. However, the standard crossbar is now replaced by the novel crossbar (MAC unit) proposed in this work (coloured green), which is illustrated in figure 4-21. The same is followed by a VI converter (also coloured green) which converts the output MAC voltage from each crossbar into proportionate currents to carry out the inference further.

Spiking Neural Network(SNN)

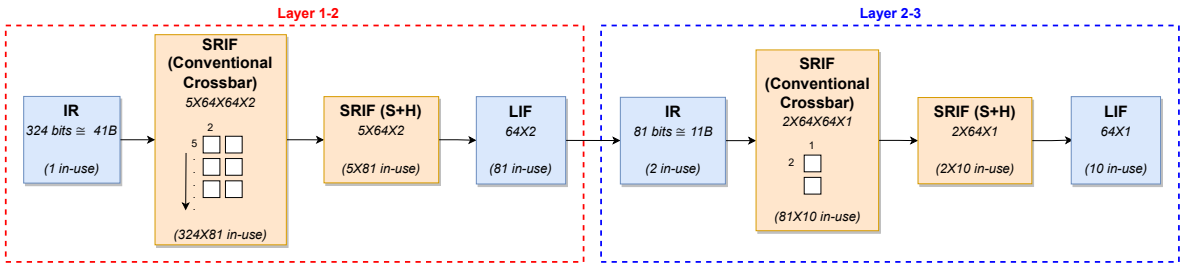


Figure 4-22: Proposed hardware setup for implementing the SNN model using conventional crossbar

As seen in figure 4-22 the input spikes are stored as bits in a standard input register (IR), as used in [19]. Every time step, these bits/spikes are parallelly transferred to a grid of conventional crossbars, each being 64×64 in size, and having their cumulative weights programmed according to the learnt weights after training the model shown in 4-19. The grid size depends

upon the total number of inputs and outputs for the particular neural network layer and has been chosen accordingly, as mentioned in the figure 4-22. The resulting currents are then sampled and integrated using the proposed LIF circuit within its in-built membrane capacitor. On reaching the threshold set for that layer, the LIF circuit fires an output spike, which is stored as a bit in a separate standard IR. With this, one cycle is completed from layer 1 to layer 2.

Similarly, the binary outputs or spikes generated in the previous cycle are stored as bits in a separate IR, fed as input to an appropriately sized grid of the standard crossbar that carries out computations similarly. This constitutes another cycle from layer 2 to layer 3. The process repeats for 60-time steps, and the neuron in the output layer firing maximum times is considered the predicted class.

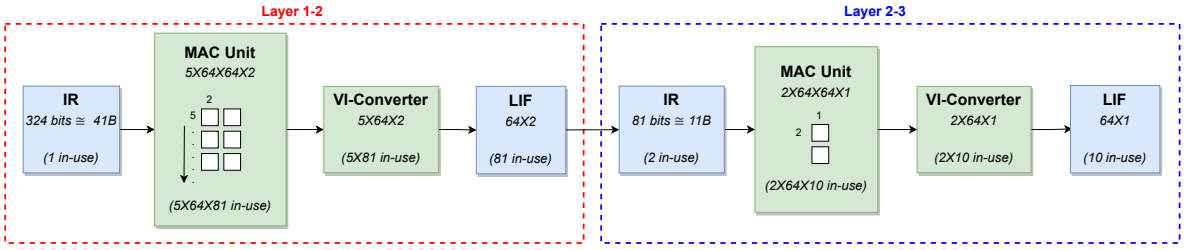


Figure 4-23: Proposed hardware setup for implementing the SNN model using novel crossbar

In the setup shown in figure 4-23, the flow of implementation remains the same, with several blocks (coloured blue) being reused from the previous setup. However, as depicted in figure 4-23, the conventional crossbar is now replaced by the novel crossbar (MAC unit) proposed in this work (coloured green). The same is followed by a V/I converter (also coloured green), which converts the output MAC voltage from each crossbar into proportionate currents to carry out inference further.

Table 4-3 represents the absolute results per inference for each aforementioned hardware setup. These values are further used to generate the performance comparison charts as depicted in figures 4-24 and 4-25 in order to make performance comparisons between the proposed and the state-of-the-art hardware on implementation of both the proposed neural networks.

Implementation	No. of cycles per inference	Energy (nJ)	Latency (μs)	Average Power (mW)	Worst Case Power (mW)	Area (mm^2)
ANN: Standard Crossbar	1	69.48	0.0627	1108.3	9840.5	1.71
ANN: Novel Crossbar	1	14.62	0.1752	83.45	108.4	1.85
SNN: Standard Crossbar	60	44.2	1.22	36	328.3	0.047
SNN: Novel Crossbar	60	11.24	7.94	1.41	1.82	0.05

Table 4-3: Performance results (per inference) of the implementation of custom developed neural networks over standard and novel crossbars

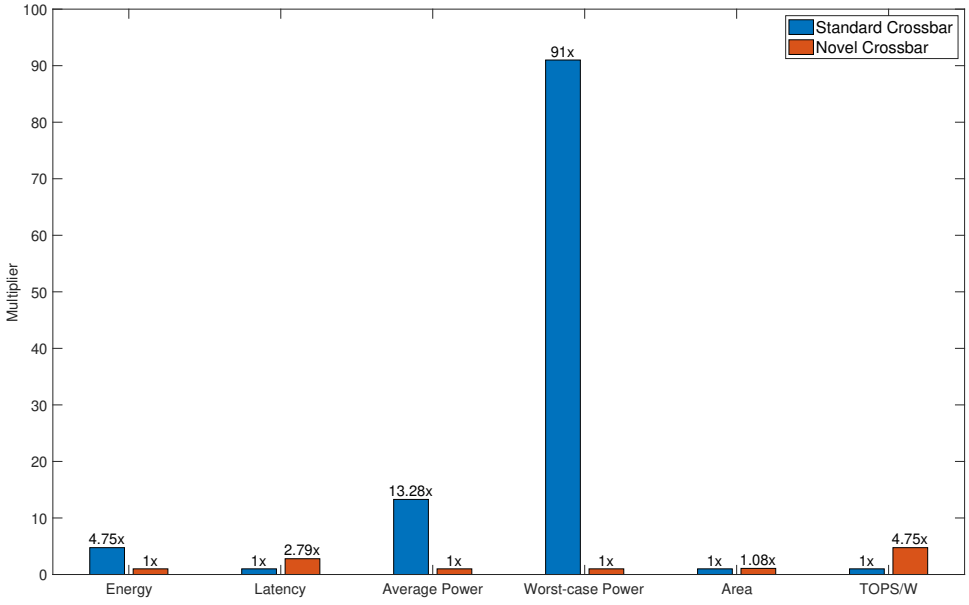


Figure 4-24: Performance comparison chart on the proposed ANN model

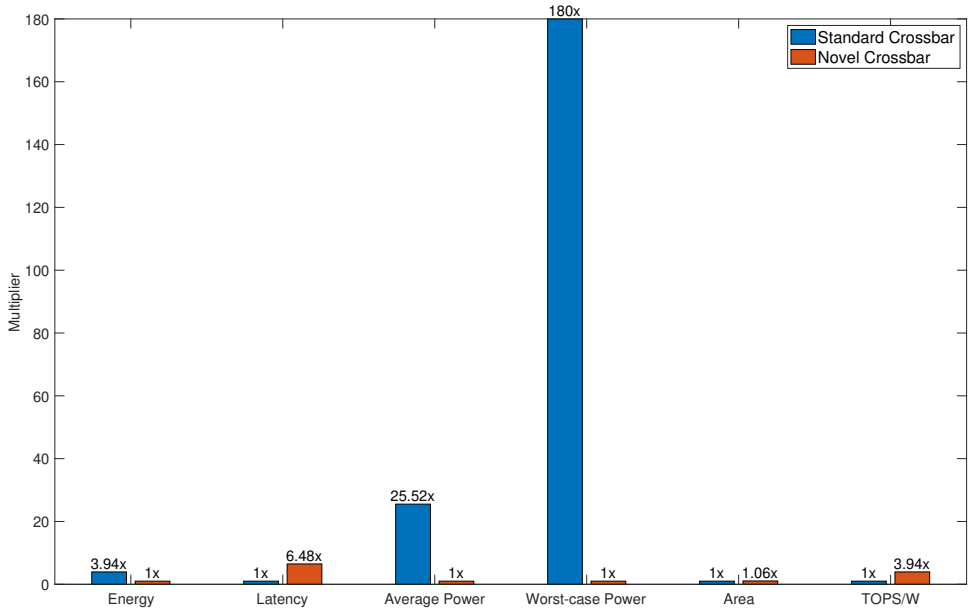


Figure 4-25: Performance comparison chart on the proposed SNN model

As observed, there is a considerable reduction in average power consumption and energy consumption for implementing neural networks on the novel crossbar. After achieving almost 13× and 26× reduction in power with the implementation of ANN and SNN, respectively, the novel crossbar exhibits promising performance over its standard crossbar counterpart with minimal area overhead. Even though the latency of the novel crossbar is higher than the state-of-the-art, the former's overall energy consumption is still lower by around 4-5× on ANN as well as SNN, indicating the potential use of the novel crossbar in designing neuromorphic micro-architectures with high power and energy efficiency.

4.3.3 Read disturb

Figures 4-27 and 4-26 represent the extent of read disturb in RRAMs as part of the standard as well as the novel crossbar. These curves are generated over the JART RRAM model [43], initially programmed to either 2KΩ (Set state) or 20KΩ (Reset state). Table 4-4 the simulation parameters for each case.

Initial state	Crossbar	Spec
Set (2KΩ)	Standard	V_{Read} =0.4V
	Novel	I_{Read} =400nA
Reset (20KΩ)	Standard	V_{Read} =0.2V
	Novel	I_{Read} =200nA

Table 4-4: Simulation setup for Read Disturb

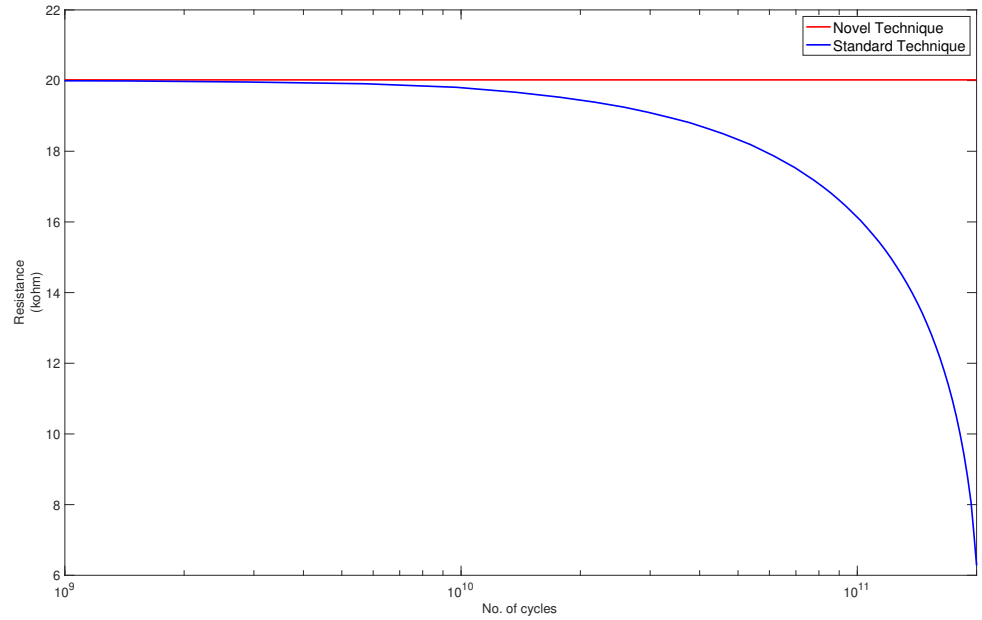


Figure 4-26: Comparison of extend of read disturb when programmed to 'Set' state

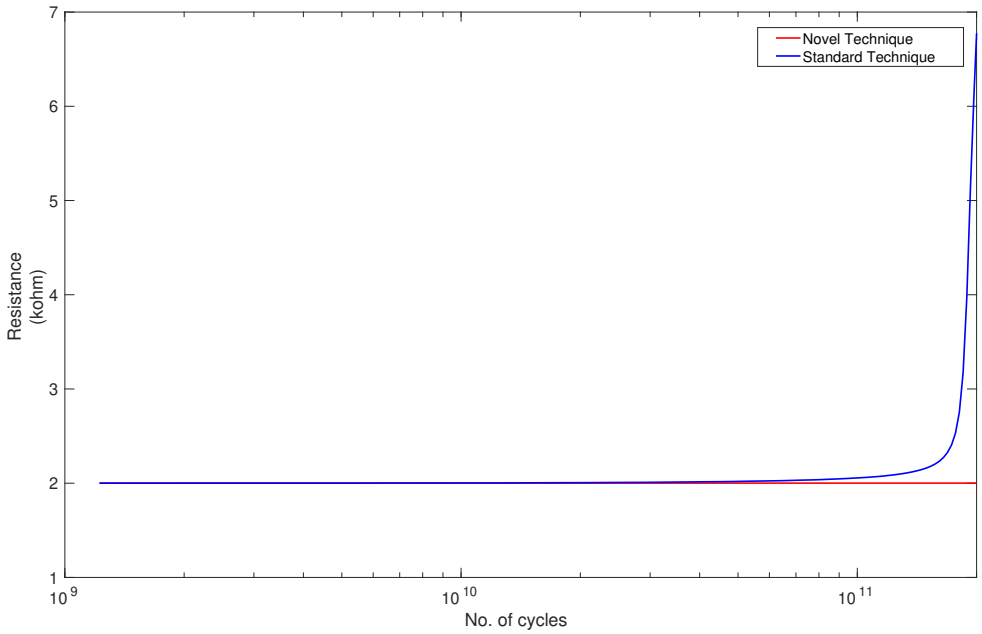


Figure 4-27: Comparison of extend of read disturb when programmed to 'Reset' state

As predicted, the RRAMs used in a standard crossbar experience a considerable read disturb where their resistance deviates by almost 230% in both cases in the given number of cycles. On

the other hand, the resistance of the RRAM used in the novel crossbar experiences virtually no deviation from its initial state. This is important in ensuring the robustness of the system for prolonged periods.

Tapeout

5.1 Overview of Chip

To evaluate the efficacy of the proposed microarchitecture, it was decided to continue with the tapeout of the design to validate the functionality of the design over silicon. Figure 5-1 shows an overview of the chip, which is divided into seven main zones, with various pins, each of which is explained in the table 5-1.

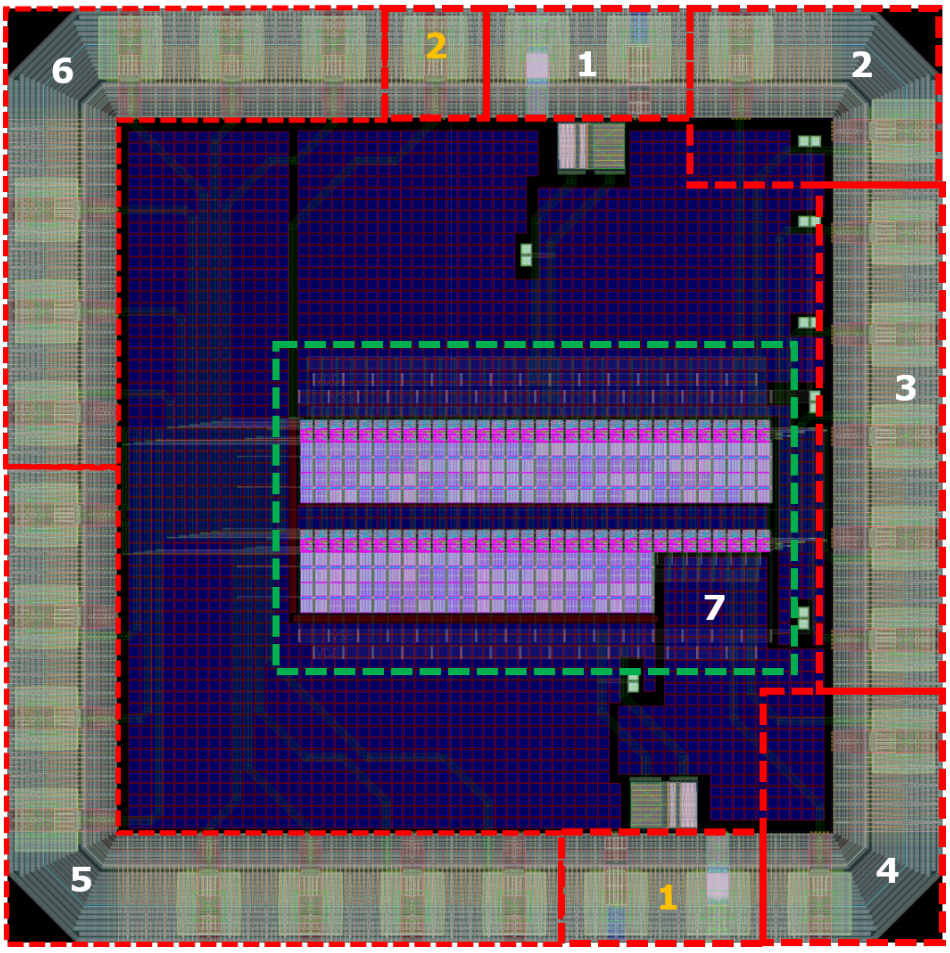


Figure 5-1: Overview of the designed chip

Zone	List of Pins	Remarks
1	V_{DD},V_{SS}	Power rails; 2 pairs available for better power distribution
2	V_{out1},I_{out2},Z_{out3}	Analogue Output Pins
3	V_b,V_{b2},V_{th},V_L,V_D	Analogue Bias Pins
4	R_{in1},V_{in2}	Analogue Input Pins
5	En_{LIF},En_{VI},En_{CP}, En_{4x},En_{3x},En_{2x},En_{1x}	Digital Control Pins
6	S_0-S_5	Digital Address Pins
7	-	Core of the chip, containing novel crossbar of size 64×64

Table 5-1: Sectional Overview of the Chip (where the pins are listed in a clockwise direction)

The chip was developed utilizing TSMC's 40nm technology node, and it has a surface area of 1mm×1mm = $1mm^2$. Within its core, it consists of 64 novel crossbar columns, each of which contain 64 hard-coded poly resistors, representing 64 synapses. This renders a novel crossbar of dimension 64 × 64 where each column is equipped with its own V/I converter and LIF circuit. The primary objective of the chip design was to have controllability and observability in order to perform measurements. Thus, pertinent circuitry has been developed to first select the required column and retrieve the analogue signals generated at each stage from the pins in zone two.

Additionally, certain columns in the crossbar are neither equipped with synapses nor are intended to perform any computation, since their individual stages are disconnected from each other. The principal objective of these columns is to independently verify the functionality of each stage, and they are designed such that pins in zone four may be used to provide manual inputs for each stage. Multiple columns are maintained in this manner in order to ensure the chip has redundancy in the event that a column or stage fails due to a fabrication or design setback, as discussed in the section 5.2.

The synapses have been hard-coded and there is no mechanism for reprogramming in order to keep the chip design within the scope of this work. Moreover, due to the unavailability of RRAM fabrication technology on the specified technological node, the synapses were simulated using poly-resistors with a set low or high resistance state of 2k and 20k, respectively.

Furthermore, due to area constraints, there was insufficient space to individually control the crossbar's inputs. Consequently, all inputs to the crossbar are connected and controlled by a single pin (WL), resulting in the simultaneous switching of all 64 devices. In order to counter this, the columns are grouped into pairs of two and the synapses of each pair are assigned a unique combination of resistances. The predicted outcome from each combination of resistances may then be utilized to perform the measurements. The purpose of keeping two sets of synapses for each combination is to have a backup column in the event that one of the columns experiences a fabrication setback.

5.2 Design Challenges

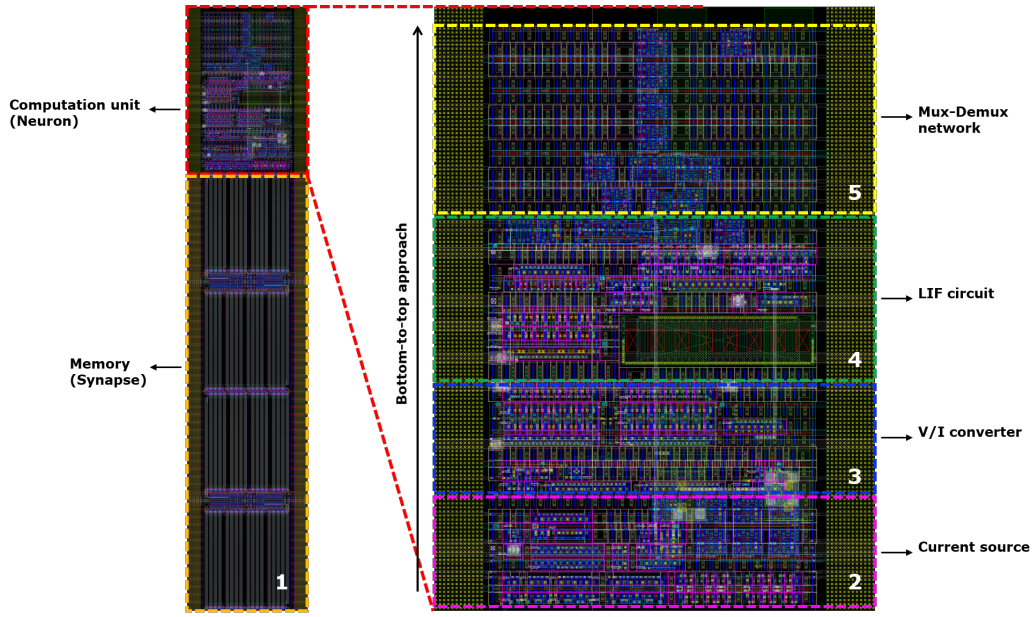


Figure 5-2: Layout of one novel crossbar column

The figure reffig:chip col depicts a crossbar column with five sections. Collectively, sections 1 and 2 represent the first stage (MAC Unit), where section 1 is the synapses and section 2 is the current source that can generate 100nA-1 μA of current in 100 nA increments. Section 3 represents the V/I converter, followed by section 4 which contains the LIF circuit. Section 5 displays a portion of the 6-bit Mux-Demux Network, which was developed to select each column independently. Therefore, the phases are arranged in a bottom-to-top order, and the challenges encountered in the design of these stages are elaborated upon.

5.2.1 MAC unit

The current source is a critical element for driving the 1^{st} stage (MAC unit) and, based on simulations, must be able to generate near accurate currents in order to assure accurate functionality. Nevertheless, mismatches caused by process variation in the PMOS pair, which constitutes the last stages of the current source, might lead to inaccurate currents. In order to make them relatively resistant to process variations, the PMOS were divided into their minimum sizes and arranged using the 1-2-2-1 device mathcing technique.

5.2.2 V/I Converter

Similarly, the NMOS-based differential amplifier used in the 2^{nd} stage (V/I converter) is an essential component to its operation. Consequently, the input NMOS and head PMOS are matched using the 1-2-2-1 device matching technique to make them resistant to process variations. Furthermore, the huge tail NMOS is split into minimum sizes and uniformly arranged in order to limit the variations in its cumulative dimensions.

5.2.3 LIF circuit

The LIF circuit is composed of analogue and digital circuits that work in tandem to execute relevant computations. Assuming that the digital circuits are sufficiently resistant to process variations, the analogue circuits, such as the comparator (PMOS-based differential amplifier) and the membrane capacitor, must be reliably fabricated to ensure proper functionality. Thus, the differential amplifier based on PMOS is designed employing the same techniques as in the previous stage. As MOSFET gates are very susceptible to process variations and voltages across the them, conventional TSMC MOM (Metal-on-metal) capacitor was used to design the membrane capacitor instead. In addition, utilising TSMC's standard digital library, a custom D latch circuit is designed to monitor and record the spiking event in each time step. Due to the restricted availability of pins, this latch is designed to self-clear whenever the LIF circuit is enabled from its disabled state. This saves one pin that would have been required to transmit a 'Clear' signal to latch at the onset of every new cycle.

5.2.4 Mux-Demux network

Using the TSMC standard digital library, a 5-bit MUX-DEMUX network is designed and placed at the top of each row of 32 crossbar columns, spanning from the left-most to the right-most column. Two of these networks are used to manage the two rows of 32 columns, and they create a 6 bit Mux-Demux network when combined. This network is equipped with buffers at certain intervals in order to regenerate the digital signals flowing across the networks.
In a comparable way, each stage is equipped with buffers to regenerate common digital signals for the next column after receiving them from the preceding column. The bias lines are made very thick so that the analogue voltages applied from the right of the chip (Zone 3) is able to span the length of the core unit (Zone 7) with minimal variation. Additionally, the bias lines are also equipped with decoupling capacitors, to make the analogue signals resistant to external noise.

Conclusion

6.1 Concluding Remarks

In conclusion, a novel low-power CIM microarchitecture has been proposed to accelerate binary as well as spiking neural networks, which shows promising characteristics to be suitable for implementation in edge devices. Two custom-trained neural networks are demonstrated, one of which is binary and the second spiking neural network, both developed over the MNIST dataset with a test accuracy of 88.4% and 87.2%, respectively. Developed to benchmark the microarchitecture, the results indicate up to 13x and 26x reduction in average power in the proposed CIM microarchitecture as compared with a standard crossbar on binary neural networks and spiking neural networks, respectively. Moreover, approximately a 4x-5x reduction in energy consumption was achieved with the implementation of BNN and SNN, respectively. Owing to the promising results, the CIM microarchitecture has been fabricated on TSMC 40nm technology node to validate its efficacy on silicon against the simulated results.

6.2 Recommendations for Future Works

Some recommendations have been proposed below

- Using actual RRAM fabrication technology to design the synapses in the chip instead of poly-resistors.
- Exploring the capability of bi-directionality.
- Exploring RRAM technology which have lower absolute resistances to design the synapses of the crossbar. This will not only help in reducing the latency of the crossbar but the column size can also be made bigger.
- Working on an appropriate writing mechanism for the proposed novel CIM crossbar.
- Exploring the scope of online training

Bibliography

[1] S. Diware, A. Singh, A. Gebregiorgis, R. V. Joshi, S. Hamdioui, and R. Bishnoi, "Accurate and Energy-Efficient Bit-Slicing for RRAM-Based Neural Networks," *IEEE Transactions on Emerging Topics in Computational Intelligence*, pp. 1–14, 2022. [Online]. Available: https://ieeexplore.ieee.org/document/9840507/

[2] A. Chen and M. R. Lin, "Variability of resistive switching memories and its impact on crossbar array performance," in *IEEE International Reliability Physics Symposium Proceedings*, 2011.

[3] Q. Duan, Z. Jing, X. Zou, Y. Wang, K. Yang, T. Zhang, S. Wu, R. Huang, and Y. Yang, "Spiking neurons with spatiotemporal dynamics and gain modulation for monolithically integrated memristive neural networks." [Online]. Available: https://doi.org/10.1038/s41467-020-17215-3

[4] W. Guo, M. E. Fouda, A. M. Eltawil, and K. N. Salama, "Neural Coding in Spiking Neural Networks: A Comparative Study for Robust Neuromorphic Systems," *Frontiers in Neuroscience*, vol. 15, p. 212, 3 2021.

[5] M. Zangeneh and A. Joshi, "Design and Optimization of Nonvolatile Multibit 1T1R Resistive RAM," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 22, no. 8, pp. 1815–1828, 8 2014. [Online]. Available: http://ieeexplore.ieee.org/document/6595151/

[6] T. B. Brown, B. Mann *et al.*, "Language models are few-shot learners," 2020.

[7] D. Silver, J. Schrittwieser *et al.*, "Mastering the game of go without human knowledge," *Nature 2017 550:7676*, vol. 550, pp. 354–359, 10 2017. [Online]. Available: https://www.nature.com/articles/nature24270https://www.nature.com/articles/nature24270/

[8] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks." [Online]. Available: http://code.google.com/p/cuda-convnet/

[9] "Artificial intelligence & autopilot | tesla." [Online]. Available: https://www.tesla.com/AI

[10] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever, "Zero-shot text-to-image generation." [Online]. Available: https://github.com/openai/DALL-E

[11] S. Nižetić, P. Šolić, D. López-de-Ipiña González-de Artaza, and L. Patrono, "Internet of Things (IoT): Opportunities, issues and challenges towards a smart and sustainable future," *Journal of Cleaner Production*, vol. 274, p. 122877, 11 2020. [Online]. Available: /pmc/articles/PMC7368922//pmc/articles/PMC7368922/?report=abstracthttps://www.ncbi.nlm.nih.gov/pmc/articles/PMC7368922/

[12] "What Is Edge AI and How Does It Work? | NVIDIA Blog." [Online]. Available: https://blogs.nvidia.com/blog/2022/02/17/what-is-edge-ai/

[13] S. Shukla, B. Fleischer, M. Ziegler, J. Silberman, J. Oh, V. Srinivasan, J. Choi, S. Mueller, A. Agrawal, T. Babinsky, N. Cao, C. Y. Chen, P. Chuang, T. Fox, G. Gristede, M. Guillorn, H. Haynie, M. Klaiber, D. Lee, S. H. Lo, G. Maier, M. Scheuermann, S. Venkataramani, C. Vezyrtzis, N. Wang, F. Yee, C. Zhou, P. F. Lu, B. Curran, L. Chang, and K. Gopalakrishnan, "A Scalable Multi-TeraOPS Core for AI Training and Inference," *IEEE Solid-State Circuits Letters*, vol. 1, no. 12, pp. 217–220, 12 2018.

[14] "Graphics Cards with Turing GPU Architecture | NVIDIA." [Online]. Available: https://www.nvidia.com/en-in/geforce/turing/

[15] N. P. Jouppi *et al.*, "In-datacenter performance analysis of a tensor processing unit," *Proceedings - International Symposium on Computer Architecture*, vol. Part F128643, pp. 1–12, 6 2017.

[16] A. Singh, M. A. Lebdeh, A. Gebregiorgis, R. Bishnoi, R. V. Joshi, and S. Hamdioui, "SRIF: Scalable and reliable integrate and fire circuit ADC for memristor-based CIM architectures," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 68, no. 5, pp. 1917–1930, 5 2021.

[17] S. A. McKee and R. W. Wisniewski, *Memory Wall*. Boston, MA: Springer US, 2011, pp. 1110–1116. [Online]. Available: https://doi.org/10.1007/978-0-387-09766-4_234

[18] A. Mehonic and A. J. Kenyon, "Brain-inspired computing needs a master plan," *Nature*, vol. 604, 2022. [Online]. Available: https://doi.org/10.1038/s41586-021-04362-w

[19] A. Shafiee, A. Nag, N. Muralimanohar, R. Balasubramonian, J. P. Strachan, M. Hu, R. S. Williams, and V. Srikumar, "ISAAC: A Convolutional Neural Network Accelerator with In-Situ Analog Arithmetic in Crossbars," in *2016 ACM/IEEE 43rd Annual International Symposium on Computer Architecture (ISCA)*. IEEE, 6 2016, pp. 14–26. [Online]. Available: http://ieeexplore.ieee.org/document/7551379/

[20] A. Ankit, I. El Hajj *et al.*, "PUMA: A Programmable Ultra-efficient Memristor-based Accelerator for Machine Learning Inference." [Online]. Available: https://doi.org/10.1145/3297858.3304049

[21] M. Shevgoor, N. Muralimanohar, R. Balasubramonian, and Y. Jeon, "Improving memristor memory with sneak current sharing," *Proceedings of the 33rd IEEE International Conference on Computer Design, ICCD 2015*, pp. 549–556, 12 2015.

[22] "MNIST handwritten digit database, Yann LeCun, Corinna Cortes and Chris Burges." [Online]. Available: http://yann.lecun.com/exdb/mnist/

[23] S. B. Furber, F. Galluppi, S. Temple, and L. A. Plana, "The SpiNNaker project," *Proceedings of the IEEE*, vol. 102, no. 5, pp. 652–665, 2014.

[24] M. Davies, N. Srinivasa *et al.*, "Loihi: A Neuromorphic Manycore Processor with On-Chip Learning," *IEEE Micro*, vol. 38, no. 1, pp. 82–99, 1 2018.

[25] F. Akopyan, J. Sawada *et al.*, "TrueNorth: Design and Tool Flow of a 65 mW 1 Million Neuron Programmable Neurosynaptic Chip," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 34, no. 10, pp. 1537–1557, 10 2015.

[26] G. Singh, L. Chelini, S. Corda, A. Javed Awan, S. Stuijk, R. Jordans, H. Corporaal, and A. J. Boonstra, "A review of near-memory computing architectures: Opportunities and challenges," *Proceedings - 21st Euromicro Conference on Digital System Design, DSD 2018*, pp. 608–617, 10 2018.

[27] S. Borkar, T. Karnik, and V. De, "Design and reliability challenges in nanometer technologies," *Proceedings - Design Automation Conference*, p. 75, 2004.

[28] N. Z. Haron and S. Hamdioui, "Why is CMOS scaling coming to an END?" *Proceedings - 2008 3rd International Design and Test Workshop, IDT 2008*, pp. 98–103, 2008.

[29] P. Chi, S. Li, C. Xu, T. Zhang, J. Zhao, Y. Liu, Y. Wang, and Y. Xie, "PRIME: A Novel Processing-in-Memory Architecture for Neural Network Computation in ReRAM-Based Main Memory," *Proceedings - 2016 43rd International Symposium on Computer Architecture, ISCA 2016*, pp. 27–39, 8 2016.

[30] B. V. Benjamin, P. Gao, E. Mcquinn, S. Choudhary, A. R. Chandrasekaran, J.-M. Bussat, R. Alvarez-Icaza, J. V. Arthur, P. A. Merolla, and K. Boahen, "Neurogrid: A Mixed-Analog-Digital Multichip System for Large-Scale Neural Simulations." [Online]. Available: http://www.ieee.org/publications_standards/publications/rights/index.html

[31] D. Ielmini and H.-S. P. Wong, "In-memory computing with resistive switching devices," *Nature Electronics*, vol. 1, no. 6, p. 333343, 2018.

[32] A. Singh, S. Diware, A. Gebregiorgis, R. Bishnoi, F. Catthoor, R. V. Joshi, and S. Hamdioui, "Low-power memristor-based computing for edge-ai applications," in *2021 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2021, pp. 1–5.

[33] N. K. Upadhyay, H. Jiang, Z. Wang, S. Asapu, Q. Xia, and J. Joshua Yang, "Emerging Memory Devices for Neuromorphic Computing," *Advanced Materials Technologies*, vol. 4, no. 4, p. 1800589, 4 2019. [Online]. Available: https://onlinelibrary.wiley.com/doi/full/10.1002/admt.201800589https://onlinelibrary.wiley.com/doi/abs/10.1002/admt.201800589https://onlinelibrary.wiley.com/doi/10.1002/admt.201800589

[34] "History of the Perceptron." [Online]. Available: https://home.csulb.edu/~cwallis/artificialn/History.htm

[35] A. L. Hodgkin and A. F. Huxley, "Currents carried by sodium and potassium ions through the membrane of the giant axon of Loligo," *The Journal of Physiology*, vol. 116, no. 4, p. 449, 4 1952. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1392213/

[36] R. FitzHugh, "Impulses and Physiological States in Theoretical Models of Nerve Membrane," *Biophysical Journal*, vol. 1, no. 6, pp. 445–466, 7 1961.

[37] J. Nagumo, S. Arimoto, and S. Yoshizawa, "An Active Pulse Transmission Line Simulating Nerve Axon*," *Proceedings of the IRE*, vol. 50, no. 10, pp. 2061–2070, 1962.

[38] C. Morris and H. Lecar, "Voltage oscillations in the barnacle giant muscle fiber," *Biophysical journal*, vol. 35, no. 1, pp. 193–213, 1981. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/7260316/

[39] E. M. Izhikevich, "Simple model of spiking neurons," *IEEE Transactions on Neural Networks*, vol. 14, no. 6, pp. 1569–1572, 11 2003.

[40] R. Brette and W. Gerstner, "Adaptive exponential integrate-and-fire model as an effective description of neuronal activity," *Journal of Neurophysiology*, vol. 94, no. 5, pp. 3637–3642, 11 2005. [Online]. Available: https://journals.physiology.org/doi/10.1152/jn.00686.2005

[41] H. Ene Paugam-Moisy and S. Bohte, "Computing with Spiking Neuron Networks."

[42] M. E. Fouda, F. Kurdahi, A. Eltawil, and E. Neftci, "Spiking Neural Networks for Inference and Learning: A Memristor-based Design Perspective," *Memristive Devices for Brain-Inspired Computing: From Materials, Devices, and Circuits to Applications - Computational Memory, Deep Learning, and Spiking Neural Networks*, pp. 499–530, 9 2019. [Online]. Available: https://arxiv.org/abs/1909.01771v2

[43] "EMRL." [Online]. Available: https://emrl.de/JART.html

[44] "PyTorch." [Online]. Available: https://pytorch.org/

[45] J. K. Eshraghian, M. Ward, E. Neftci, X. Wang, G. Lenz, G. Dwivedi, M. Bennamoun, D. S. Jeong, and W. D. Lu, "Training Spiking Neural Networks Using Lessons From Deep Learning," 9 2021. [Online]. Available: http://arxiv.org/abs/2109.12894

[46] L. Kull, T. Toifl, M. Schmatz, P. A. Francese, C. Menolfi, M. Brändli, M. Kossel, T. Morf, T. M. Andersen, and Y. Leblebici, "A 3.1 mW 8b 1.2 GS/s single-Channel asynchronous SAR ADC with alternate comparators for enhanced speed in 32 nm digital SOI CMOS," *IEEE Journal of Solid-State Circuits*, vol. 48, no. 12, pp. 3049–3058, 12 2013.