Towards Energy-Efficient Spiking Neural Networks: A Robust Hybrid CMOS-Memristive Accelerator

FABIHA NOWSHIN, Virginia Tech, USA
HONGYU AN, Michigan Tech, USA
YANG YI, Virginia Tech, USA

Spiking Neural Networks (SNNs) are energy-efficient artificial neural network models that can carry out data-intensive applications. Energy consumption, latency, and memory bottleneck are some of the major issues that arise in machine learning applications due to their data-demanding nature. Memristor-enabled Computing-In-Memory (CIM) architectures have been able to tackle the memory wall issue, eliminating the energy and time-consuming movement of data. In this work we develop a scalable CIM-based SNN architecture with our fabricated two-layer memristor crossbar array. In addition to having an enhanced heat dissipation capability, our memristor exhibits substantial enhancement of 10% to 66% in design area, power and latency compared to state-of-the-art memristors. This design incorporates an inter-spike interval (ISI) encoding scheme due to its high information density to convert the incoming input signals into spikes. Furthermore, we include a time-to-first-spike (TTFS) based output processing stage for its energy-efficiency to carry out the final classification. With the combination of ISI, CIM and TTFS, this network has a competitive inference speed of 2μs/image and can successfully classify handwritten digits with 2.9mW of power and 2.51pJ energy per spike. The proposed architecture with the ISI encoding scheme can achieve ~10% higher accuracy than those of other encoding schemes in the MNIST dataset.

CCS Concepts: • **Hardware → Emerging architectures**;

Additional Key Words and Phrases: Memristor, spiking neural network, computing-in-memory, spatiotemporal architecture, LIF neuron

ACM Reference format:
Fabiha Nowshin, Hongyu An, and Yang Yi. 2024. Towards Energy-Efficient Spiking Neural Networks: A Robust Hybrid CMOS-Memristive Accelerator. *ACM J. Emerg. Technol. Comput. Syst.* 20, 1, Article 5 (January 2024), 20 pages.
https://doi.org/10.1145/3635165

This work was supported in part by the U.S. National Science Foundation (NSF) under Grant CCF-1750450, Grant ECCS-1731928, and Grant CCF-1937487.
Author's address: F. Nowshin and Y. Yi, Virginia Tech, The Bradley Department of Electrical and Computer Engineering, 1185 Perry Street, Blacksburg, VA 24060, USA; e-mails: {fabiha27, yangyi8}@vt.edu; H. An, Michigan Tech, Electrical and Computer Engineering, 1400 Townsend Dr., Houghton, MI 49931, USA; e-mail: hongyua@mtu.edu.

1550-4832/2024/1-ART5 $15.00
https://doi.org/10.1145/3635165

1 INTRODUCTION

In recent years, machine learning has shown successful results in artificial intelligence applications such as computer vision, natural language processing, pattern classification and speech recognition [1]. The rapid expansion of data volume and neural network training durations over the past decade has posed significant challenges for CPUs and GPUs in terms of energy efficiency [2]. In response, specialized **artificial intelligence (AI)** accelerators such as Eyeriss and Google TPUs have been devised to address these concerns [3]. However, the constant shuttling of data back and forth from the accelerator and memory is often limited by the memory bandwidth and contributes to significant power and energy consumption.

Computing-in-memory (CIM) emerged as a promising paradigm for AI accelerators where the computations are done inside the memory [4]. The conventional CMOS memory technologies encompass **dynamic random access memory (DRAM), static random access memory (SRAM)** and flash that rely on the charge storage phenomenon. However, compared to these traditional technologies, the **emerging non-volatile memory technologies (eNVMs)** are more suited to CIM operations in the field of neuromorphic hardware because of their capability in closely imitating the neurons and synapses in biological systems [5]. The memristor, a specific eNVM, has several advantageous properties of low power consumption, scalability, compatibility with CMOS technology, and analog conductance modulation making it a viable replacement for traditional CMOS memory technologies [5]. These characteristics of the memristor allows it to be used in building CIM architectures, supplanting the usage of analog-to-digital and digital-to-analog converters, resulting in substantial reductions in area and power consumption. Previously there have been several memristor-based AI accelerators built on **Convolutional Neural Networks (CNNs)**, including PRIME, ISAAC, PipeLayer and AtomLayer that have achieved significant improvements in energy and power compared to CMOS-based accelerators [6–9].

The integration of **Spiking Neural Networks (SNNs)** in memristor-based machine learning can be further beneficial in improving the efficiency of these neural networks. SNNs have the ability to emulate biological neurons more closely via the transmission of spiking signals. The information in SNNs is encoded in both the timing and the ordering of the spikes, allowing them to have a spatiotemporal information processing capability [10]. While the spatial aspect arises from the neurons localized to each other, the temporal aspect stems from the timing characteristics of SNNs [11]. Once a threshold is exceeded, a spike is fired, capturing the information in a binary format, which further aids in the energy and power efficiency of these neural networks in hardware implementations. This makes SNNs superior to ANNs and is therefore depicted as the third generation of neural networks [12]. In memristor-based neural networks, the integration of SNNs also ensures improvements in noise margins and increased tolerance in the variation of the devices [13]. There have been several implementations of memristor-based SNNs over the past years. An energy-efficient and reconfigurable architecture built with Memristor Crossbar on deep SNNs, RESPARC, was developed that uses a hierarchical reconfigurable design to add data-flow patterns on SNNs [14]. Zhao et al. [15] fabricated a memristor-based SNN made with **Leaky-integrate and fire (LIF)** neurons capable of performing **spike timing dependent plasticity (STDP)** learning. However, there is no evaluation of the network against any image or pattern recognition tasks. In 2021, a multilayer memristor-based SNN was developed with a temporal order information encoding and STDP for weight updating [16]. Although this work can achieve a relatively high accuracy with the MNIST dataset, the LIF neurons used in [16] incorporate multiple operational amplifiers, consuming significant area and power. A fully hardware implementation of memristor-based SNN was also developed in [17] that realizes the LIF neuron functions with threshold switching. However, the utilization of digital circuitry in this work leads to a notable increase in the hardware overhead.

In our work, we aim to build a low power, area and energy-efficient CIM-based SNN architecture. To achieve our goal, we combine the SNN with our fabricated robust two-layer memristor crossbar array that has an enhanced heat dissipation capability [18]. Two major problems with memristor devices are resistance variation and leakage currents [19]. With the added heat dissipation layer, our device is able to reduce the resistance variation and increase the inference accuracy by ~30% [20]. The issue of the leakage current is minimized by the high on and off ratio of our device, making it reach stable high and low states [20]. One concern with the one-layer crossbar is that the entire structure suffers from high latency, large area, and a substantial amount of power consumption [21]. By combining the monolithic three-dimensional integration technology, two crossbars are stacked on top of each other that can reduce the area, power consumption and latency by 2×, 1.48× and 1.58×, respectively [18]. In addition to the improvements mentioned, the enhanced heat dissipation capability of our memristor crossbar array improves the robustness of the architecture. By effectively dissipating heat, the impact of temperature fluctuations is minimized, ensuring the reliability and stability of the device during prolonged operation. This robustness is crucial in maintaining the accuracy and performance of SNN.

To convert our data into spikes, we investigate and implement a type of temporal encoding scheme known as the **inter-spike interval (ISI)** encoding scheme. Rate and temporal encoding schemes stand out as the two prominent neural coding schemes in literature. While data is encoded in the frequency of the spikes in rate encoding, temporal encoding utilizes the timing aspect of the signal and encodes information at precise times of the spikes [22, 23]. Due to the fewer spike generation in temporal encoding, it is superior to its rate encoding counterpart in terms of power and energy efficiency, especially in hardware implementations [23]. Among the temporal encoding schemes, the widely known ones are phase, burst and **time-to-first-spike (TTFS)** encoding [24–26]. But the major advantage of our developed ISI encoding scheme lies in its high information density property which stems from the fact that information is encoded in both the distance and the timing of the spikes, influenced by the strength of the incoming signal.

We propose an energy-efficient SNN by combining the merits of our ISI encoding scheme and the two layer memristor. The input and output processing units of the memristor crossbar are discussed in detail in our work as well as the construction of the hidden units demonstrating the scalability of the network. A small-scale hardware model is proposed using a three-layer SNN architecture and the accuracy is shown using pixelated handwritten images of digits as inputs. To demonstrate the potential of our design, a large-scale three-layer CIM-based SNN model is then developed in PyTorch to test this network with the MNIST dataset which has shown to have a very high accuracy compared to its TTFS and rate counterpart.

The key contributions of our work are listed below:

(1) We design and optimize our SNN with an ISI encoding scheme for high information density and spatiotemporal information capability, memristor crossbars for CIM operations and TTFS-based classification scheme in the output layer for energy efficiency.
(2) Both positive and negative weight matrices are implemented using our two-layer memristor crossbar for improvements in latency, area and power consumption.
(3) We successfully fabricated our robust memristor crossbar array with an enhanced heat dissipation capability and high on/off ratio to be used for CIM-based vector-matrix multiplication operations.
(4) We demonstrate the classification of handwritten digits using a TTFS classification scheme on hardware with pixelated images of digits 0–9 while consuming merely 2.9mW of power with an inference speed of 2μs/image.

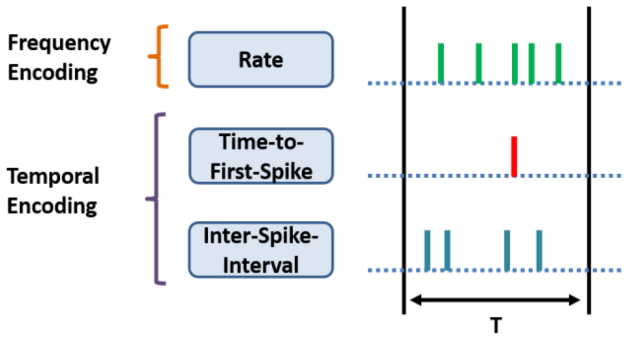


Fig. 1. Overview of different encoding schemes.

(5) We evaluate our large-scale CIM-based SNN through the MNIST dataset on PyTorch and our results provide a ~10% increase in accuracy with the ISI encoding scheme, compared to the rate and TTFS encoding schemes.

2 BACKGROUND AND RELATED WORK

2.1 Encoding Schemes

SNNs operate using spikes which are discrete events at certain times rather than continuous values. SNNs emulate biological neurons more closely where the input spikes are applied to the neurons, and once they exceed a threshold, a spike is fired [27]. Some commonly used spiking neuron models are Izhikevich [28], Hodgkin-Huxley [29] and LIF [30]. The LIF neuron demonstrates a closer resemblance to biological neurons by emitting a spike when a certain threshold is surpassed and subsequently resetting [30]. This neuron remains active only during the spike emission and reception process, leading to a significant reduction in energy and power consumption within SNNs. To apply the data to the SNNs, the information needs to be encoded into a spike train. The performance of the neural network greatly depends on the type of encoding scheme which justifies the attention neural encoding receives in literature. Figure 1 shows the overview of the encoding schemes.

The two more popular encoding schemes are rate and temporal encoding schemes [31–33]. In rate encoding the information is encoded in the frequency of the spikes over a time window T [34]. From Figure 1, the firing rate of the neuron is given by Equation (1)

$$n = \frac{N}{T}, \tag{1}$$

where N is the number of spikes. Latency issues as well as significant energy consumption arising from the large number of spike generation from rate encoding overrides its robustness and simple implementation characteristics [35, 36]. In contrast, temporal encoding schemes have shown superior characteristics over rate encoding schemes by not only solving the energy consumption issue by reducing the number of generated spikes but also capturing the temporal nature of the information [32, 33]. Furthermore, temporal encoding schemes like burst and phase encoding have shown much higher efficiency than **Deep Neural Networks (DNNs)** when used in deep SNNs [25, 26]. However, despite their greater performance, the temporal encoding schemes in [25, 26, 32] and [33] still suffer from the problem of a large amount of spike generation.

The TTFS encoding scheme is a good fit to address this issue of spike generation. As portrayed in Figure 1, the TTFS encoding scheme is the most power and energy-efficient scheme since it generates only one spike for the entire time period [24]. However, this scheme is less robust since

it is heavily reliant on the onset of the sampling window, resulting in the need for an external reference. Therefore, to encode our input information into spikes, we adopt a type of temporal encoding scheme, the ISI encoding scheme, in our design, depicted in Figure 1. Information is encoded in both the timing of the spikes as well as the time between the spikes, giving it the characteristic of two-dimensional information storage [22, 37, 38]. From the ISI encoding scheme developed in [22], our design is a unique version that reduces the number of spikes significantly by generating only two spikes per sample. Our ISI module is modified to include input preprocessing as well as capturing the distance between the spikes in form of pulses to be applied to the memristor crossbar. In the output layer, we integrate the TTFS encoding scheme for classification purposes, leveraging its power and energy efficient characteristics.

2.2 In-Memory Computing

The implementations in neuromorphic hardware require the use of a device or circuit that is capable of imitating the synaptic behavior. The property of the memristor device enables it to be implemented in neuromorphic computing. A key operation in deep learning applications is vector-matrix multiplication, which requires frequent storage and retrieval of weights to and from memory. Due to the large amount of data shuttling, it is challenging for machine learning applications to run on Von Neumann computing architectures [39]. Moreover, the performance of the system degrades due to this constant transfer of data. As technology scales down further, the performance will also be affected by higher energy consumption stemming from the interconnect parasitics [40].

This issue can be resolved by using memristor crossbars to build a CIM structure that has the capability of computing large amounts of vector-matrix multiplications inside the memory while reducing the power and area significantly [39]. A memristor or a "memory resistor" is a two-terminal eNVM technology that has been immensely studied in recent years and has shown to have the capability to emulate neurons and synapses to build neuromorphic hardware [19]. The traditional memory technologies include SRAM, DRAM and Flash that rely on the charge storage phenomenon [41]. However, as technology scales further, these stored charges tend to be lost, exacerbating the performance and introducing noise and reliability issues. On the other hand, memristors have a metal-insulator-metal structure, and one of the ways they can change their resistance state is through the formation and rupture of a conductive filament between the two metal electrodes [19]. They typically have two states, a **high resistance state (HRS)** and a **low resistance state (LRS)**. The switch from HRS to LRS is known as the set process and the switch from LRS to HRS is the reset process. Memristors can be used to build large crossbar structures where each device is located at the crosspoint of the two nanowires [42]. In the memristor crossbar, the horizontal rows are the wordlines while the vertical columns are the bitlines. If an input voltage vector of V is applied to the ith row of a crossbar, the accumulated current from jth column of the crossbar can be calculated as:

$$I_j = \sum_{i=0}^{n} G_{ij} V_i, \tag{2}$$

where G is the conductance stored in the memristor located at the ith row and jth column. Due to their nanoscale structures, memristor crossbars can be used to employ neural networks in a large-scale, consuming significantly less power while operating under a low voltage [42].

3 CIM-BASED SNN ARCHITECTURE

We design the SNN architecture by combining the advantages of the spatiotemporal nature of SNNs and the energy-efficiency of CIMs. As demonstrated in Figure 2, the designed architecture is assembled with four key modules, an ISI encoded input layer, our fabricated memristor crossbar

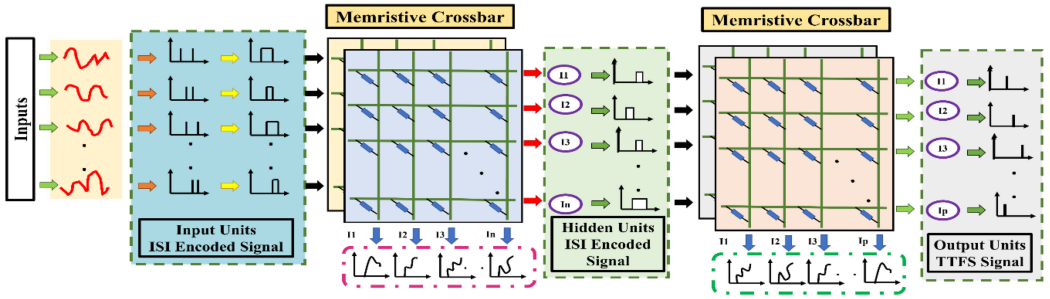


Fig. 2. Architecture of the designed CIM-based SNN.

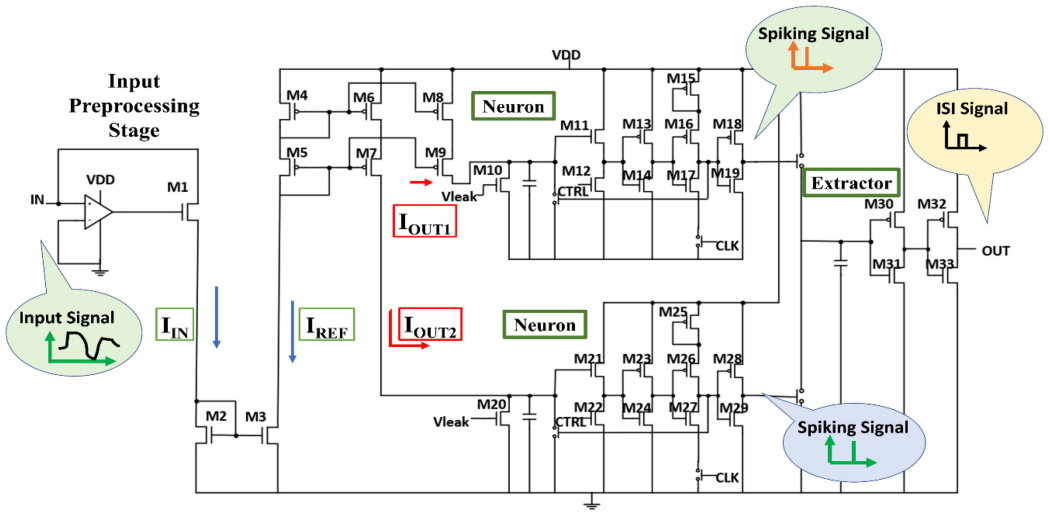


Fig. 3. Input processing unit.

array for the CIM operations, a hidden layer for efficient signal transmission and as a post-processing stage from the crossbar, and a TTFS-based output decoder for the final classification layer. The ISI encoded input layer realizes the spatiotemporal aspect of SNNs by efficiently converting the input signals into spikes while maintaining a high input information density through the generation of only two spikes. The crossbar layer on the other hand can effectively implement positive and negative weight storage mechanism to carry out the vector-matrix multiplication operations and its added heat dissipation layer contributes to reduced resistance variation and robustness. The final TTFS decoder further contributes to energy and power efficiency by generating only one spike per clock cycle since it is only used for classification. The detailed circuit design of each module is discussed in detail in this section.

3.1 Input Layer

Although the ISI encoding scheme has a higher information density, it suffers from the major issues of temporal encoding scheme, latency, and increased spike generation [43]. To overcome these two issues, we develop a novel ISI encoding scheme where only two spikes are generated in the same time frame and the time between them is captured to be applied to the next stage. The circuit design of one input processing unit to produce the ISI output is depicted in Figure 3. The three

key stages used to design the input unit are the input processing stage, LIF neuron stage and the extractor stage. A conventional transconductance amplifier is used to develop our precise current-to-current conversion method in the input preprocessing stage. When an analog input current I_{in} is applied to this unit, the high input impedance of the amplifier routes the signal to the drain of the transistor M_1. With the help of the matched transistors M_2 and M_3, the reference current I_{REF} is generated, effectively matching the input current. The variation of I_{in} linearly produces the two scaled current outputs using the corresponding current mirror stage. We exploit the use of a diode-connected pMOS load using the transistors (M_5, M_7, and M_9) in our current mirror stage to counteract the effects of channel length modulation which would otherwise cause the variation of load voltage to affect the load current. This would prevent the linear generation of scaled current values.

Once the current values are scaled, I_{out1} and I_{out2} are applied to the subsequent stage consisting of two LIF neurons. At this stage we develop the spatiotemporal nature of our design by utilizing the LIF neuron to generate the spikes [38, 44]. V_{leak} represents the leakage voltage, emulating the behavior of biological neurons. This particular LIF neuron uses a capacitor-sensing methodology to monitor the incoming current signal. When the capacitor acquires sufficient charge to surpass the threshold voltage set by the *CTRL* signal, the cascaded inverters ($M_{13,\,14}$ and $M_{18,\,19}$) and ($M_{23,\,24}$ and $M_{28,\,29}$) are used to fire the two output spikes. In the meantime, the frequency of the spikes generated is regulated by an external clock signal to our system, *CLK*. Following the generation of the spikes, the positive feedback loop from the transistors ($M_{16,\,17}$ and $M_{26,\,27}$) resets the switch and allows the capacitor to discharge.

With the generation of two spikes per clock cycle, the values are applied to the extractor unit in the following stage. The spike associated with I_{out1} from the LIF neuron is generated first, enabling the capacitor to charge up to its maximum potential when the signal is applied to it. Upon the arrival of the second spike, the capacitor is reset and the cascaded inverters ($M_{30,\,31}$ and $M_{32,\,33}$) generate a pulse signal, capturing the time between the spikes effectively. The equation for the time difference between the spikes can be expressed as:

$$\tau_1 - \tau_2 = \int C \cdot dV \left[\frac{1}{m \cdot I_{in} - I_{leak}} - \frac{1}{n \cdot I_{in} - I_{leak}} \right] \tag{3}$$

where C and V are the membrane capacitance and voltage and I_{leak} is the leakage current arising from the leakage voltage while m and n are scaling factors for the current. The pulse signal can then be applied to the CIM stage built with the memristor crossbars. Using our ISI approach, only two spikes are generated per clock cycle, which is able to address the issue of latency and energy consumption through significantly reduced spike generations, suffered by other temporal encoding schemes [45]. Our ISI encoding schemes captures information in higher dimensions based on the duration of the pulse and the timing of the spikes, where a shorter pulse corresponds to a higher intensity input. Although the relationship between $(\tau_1 - \tau_2)$ and the input current is inversely proportional, in the preceding layers of our network of the memristor crossbars, we adjust the weight values to achieve high accuracy and reflect the higher intensity of the training data. This ensures that the network learns and adapts effectively to input signal intensity variations, resulting in improved performance and accuracy in classification tasks.

3.2 Memristor Crossbar Layer

We use our fabricated memristor device and crossbar array to carry out the vector-matrix multiplications for the CIM operations. Memristors suffer from low reliability and accuracy which stems from the resistance variations inside the devices. During the switching of memristors, the completion of conductive filaments is facilitated by the electromigration of active metal ions.

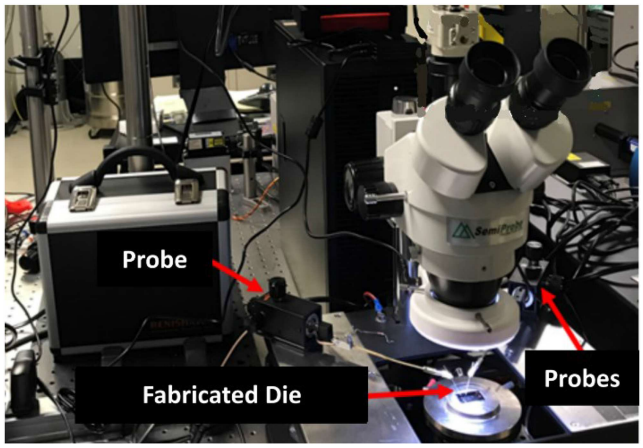
Table 1. Variation Comparison of Memristors with Different Materials

Memristive Device	R_{on}	Thermal conductivity
Cu/TaOx/Rh/Cr	500 ± 5 Ω	Rh: 150; Cr: 94
Cu/TaOx/Rh/Ti	225 – 750 Ω	Rh: 150; Ti: 22
Cu/TaOx/Pt/Cr	331 – 1000 Ω	Pt: 72; Cr: 94
Cu/TaOx/Pt/Ti	230 – 1000 Ω	Pt: 72: Ti: 22

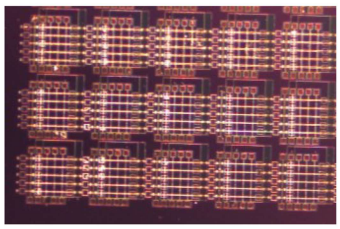
However, due to the stochastic nature of ion migration and atom diffusion, the on-resistance variation (R_{on}) is relatively high, particularly at high temperatures. As the switching process progresses, the temperature gradually increases as a result of the movement of oxygen atoms and ions within the metal oxide, leading to heat accumulation within the memristor. This elevated temperature further amplifies the metal diffusion effect. From a thorough analysis in [18] and [46] it has been concluded that the heat dispersed during the formation and rupture of the conductive filament is an essential factor in influencing the resistance variation of memristors. A significant amount of current flows during the set and reset process of the memristors which results in a substantial amount of heat dissipation. The higher the thermal conductivity of the electrodes and the oxide, the faster is the heat removal process from the conductive filament and the lesser the on-state resistance variation. Our memristor device therefore incorporates an additional heat dissipation layer to allow for rapid heat removal which reduces the resistance variation by ~30%.

The device follows the typical ReRAM structure; it has an oxide layer sandwiched between two electrodes. For our benchmark device, we use the memristor configuration of Cu/TaOx/Pt. The memristor has been fabricated in a thermally oxidized silicon wafer. Due to its medium activation energy and rapid ionization characteristic, we use Copper as the active anode. Several inert electrode configurations have been fabricated on a thermally oxidized silicon wafer, including Pt/Ti, Pt/Cr, Rh/Cr, Rh/Ti, Ir/Ti and Ir/Cr, and they were then compared with our benchmark device to find the most effective material for heat dissipation. These layers have been deposited by e-beam PVD in a Kurt Lesker PVD-250 chamber. The TaOx layer was deposited by evaporating the Ta_2O_5 pellets in the evaporation chamber but without the injection of oxygen. The inert electrodes of Pt, Ir and Rh have poor adhesion properties with the SiO_2 which is why a Ti glue layer needs to be used. The measurements revealed a positive correlation between the cycle-to-cycle resistance variation (R_{on}) and the thermal conductivity of the heat dissipation layer as shown in Table 1. However, Cr has a higher heat conductivity than Ti and is also a good adhesion layer, making it more suited to be used in our memristor device.

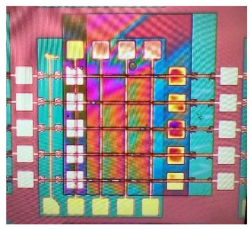
Besides being compatible with CMOS technology, the Rh-Cu combination leads to an insignificant amount of solid solubility between Rhodium and Copper and therefore we incorporate the use of Rhodium for the inert cathode. In comparison with other inert electrodes, Rhodium holds a higher heat conductivity to allow for faster heat removal. Furthermore, replacing the Platinum layer with Rhodium results in a cost reduction by 45 times for our memristor. We perform an endurance test, and based on our results, the configuration of Rh/Cr produces the largest number of switching cycles compared to its counterparts. Therefore, our fabricated memristor has the configuration of Cu/TaOx/Rh/Cr. A layer of TaO_x is incorporated before the top layer is added above the bottom layer. This is carried out by evaporating Ta_2O_5 pellets onto e-beam evaporation system with oxygen injection into the chamber. The stoichiometry of the TaO_x layer is improved through the oxygen injection. This oxide layer is less defective and ensures there is electrical insulation between the top and bottom layers. Figure 4 demonstrates our memristor measurement setup and our fabricated two-layer crossbar. From Figure 4(c), the left and bottom pads are used to access



(a)



(b)



(c)

Fig. 4. (a) Memristor wafer with attached probes. (b) Fabricated memristor wafer. (c) Zoomed-in view of our two-layer 5x5 memristor crossbar.

the top layer while the right and top pads are used to access the bottom layer. Compared to the one-layer, the resistance and capacitances are significantly reduced in the two-layer structure leading to improvements in latency and power consumption.

The I-V switching curve for the memristor is demonstrated in Figure 5. A positive voltage is swept at 0.2(V/s) from the top electrode of the memristor and once the set voltage is exceeded, the conductive filament is formed, producing the I-V characteristic curve. These measurements were performed several times to analyze the switching capability of our memristor. The deviation from the mean value allows us to calculate a cycle-to-cycle variation of only 4% with our memristor configuration, a ~30% increase compared to the Pt/Ti configuration which produces the highest variation of ~43%. This reduced variation gives a much higher accuracy demonstrated in our results section.

The high on/off ratio, 1MΩ to ~940MΩ, comes from the compliance current setting of the device. The relationship between R_{on} and the compliance current can be given by the following equation:

$$R_{on} = \frac{K}{I_{cc}^{n}}, \tag{4}$$

where I_{cc} is the compliance current and n and K are the fitting parameters for the I-V characteristic curve. There is a negative correlation between R_{on} and I_{cc}. For our architecture design we use a smaller compliance current of 1μA which also increases the endurance of our memristor to 1,000

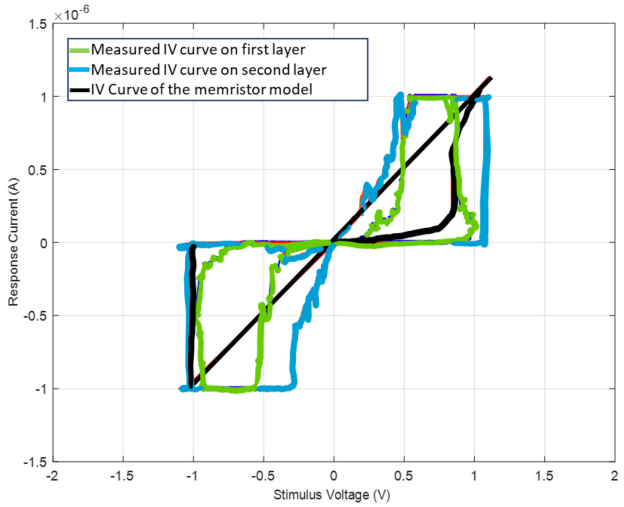


Fig. 5. I-V characteristic curve of our fabricated two-layer memristor with the Cu/TaOx/Rh/Cr configuration.

switching cycles compared to 150 switching cycles at 10μA. This significantly high on and off ratio of our fabricated memristor crossbar can avoid the sneak path current issue.

3.3 CIM-based Vector-Matrix Operations

The weights can be mapped into the memristor crossbar by applying write pulses of the same amplitude (1V, 10ns). Memristors can only hold positive conductance values and therefore, we develop a method to allow for both positive and negative weight implementations via our memristor crossbars. Traditionally, there has been a lack of sufficient implementation of positive and negative weight mapping within crossbars, which is a crucial feature for performing vector-matrix multiplications. In [47] additional circuitry is employed to execute positive and negative weights in the crossbar. Unfortunately, this approach suffers from two drawbacks, increased area and power consumption due to the additional components involved. To overcome these limitations and avoid the need for extra circuitry, we adopt a weight mapping mechanism similar to [48], which involves applying a voltage with a similar amplitude but opposite polarity to two neighboring rows. This approach allows for the representation of both positive and negative weight values. However, it does not address the issue of how to supply a voltage of different polarity to the adjacent rows, considering that the wordlines are interconnected in a crossbar configuration.

As demonstrated in Figure 6, we use our stacked crossbar fabricated with the monolithic 3D technology discussed in the preceding section, with a voltage of the same amplitude but positive polarity applied to one and negative polarity applied to another. The left and bottom pads are used to access the top layer and the right and top pads are used to access the bottom layer. The total outgoing current from each memristor pair can be given by the following equation:

$$I_{total} = \sum_{0}^{i} [(V_i) \cdot \{G_{+ij} - G_{-ij}\}] \tag{5}$$

$$I_{sum} = I_{total} + \alpha, \tag{6}$$

where the conductance term inside the parenthesis $\{G_{+ij} - G_{-ij}\}$ represents an individual weight value. The positive part of our weight value, G_{+ij} is mapped onto one crossbar and the magnitude of the negative part of the weight value, G_{-ij} is mapped onto another. To achieve the negative weights,

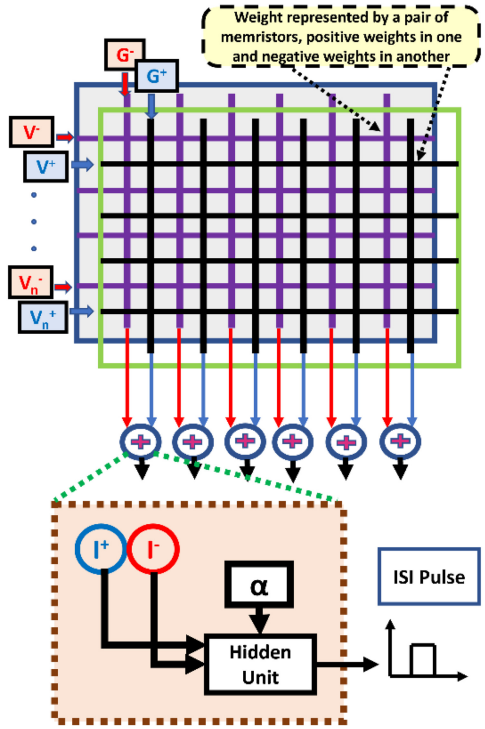


Fig. 6. CIM-based vector matrix operation with positive and negative weight implementation.

a voltage of opposite polarity is applied to the negative crossbar and hence we can effectively realize negative weight values within the overall weight matrix.

Our designed architecture is scalable and therefore makes way for several hidden layers to be added. The positive and negative currents are combined in one node and applied to each hidden unit. Our hidden units follow the similar circuit design approach from Figure 3, where the two currents are amplified and split into two values to allow for spike conversion. However, to adjust for the negative valued currents that arise from our negative weight implementation technique, we incorporate a constant current of α, depicted in Equation (8) and Figure 6. This allows for all the negative valued currents to be transferred to the positive domain which is necessary for the LIF neurons in the hidden stage to function. In our neural network setup, the memristor crossbars are followed by a series of LIF neurons, where each column is connected to each neuron. It is necessary to take all currents into account in order for the subsequent stage of LIF neurons to function correctly. By introducing the α factor, we ensure that all currents, regardless of their polarity, are scaled into the positive range. This scaling operation is crucial for the proper functioning of the LIF neurons, as these neurons rely on positive polarity currents for their activation and subsequent processing. The inclusion of α allows us to avoid complications that could arise from negative current handling within the network.

As plotted in Figure 7, a linear response of our output current can be obtained from our applied positive and negative voltages. In the hidden units these current outputs are then converted to the ISI encoded pulse signals following the same mechanism as Figure 3. Our approach introduces scalability to the network and ensures rapid and power-efficient transmission of signals by converting them to spikes between the CIM stages.

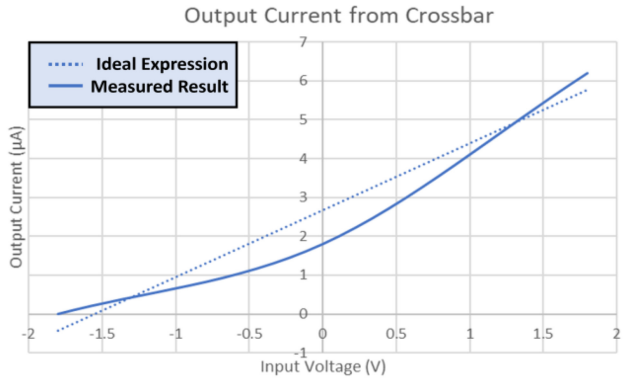


Fig. 7. Measured characteristics of positive and negative weight implementation technique.

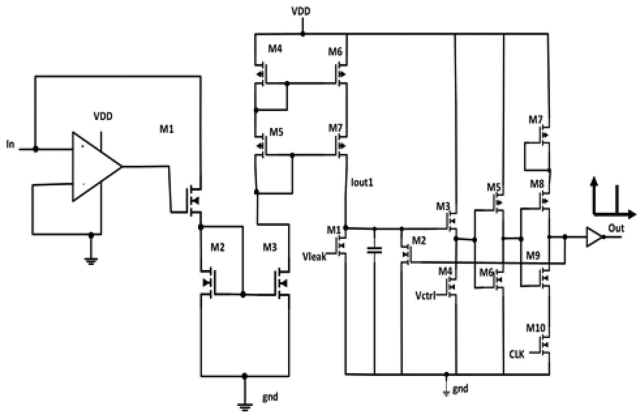


Fig. 8. One output processing unit.

3.4 Output Layer

As discussed in the preceding section, we use a TTFS classification scheme in our output layer to separate the different classes. Since only one spike is generated per clock cycle, it is deemed as the most power and energy-efficient encoding scheme, and we therefore deploy this method for classification [24]. Our TTFS decoder design is demonstrated in Figure 8. The output current from the columns of our memristor crossbar is applied to the precision current-to-current converter stage, where due to its high input impedance it allows for the current to be transmitted to the current mirror stage and replicated with the desired amplification. The amplified value from the current mirror can also be used as an additional weight tuning mechanism for the architecture. Once this current is applied to a single LIF neuron, the capacitor sensing technique allows for a spike to be generated. This neuron is controlled by an external clock that is the same as the clock of our entire architecture. Using one output processing unit for each class, we therefore develop our final output stage.

4 PERFORMANCE ANALYSIS

Our performance analysis is divided into four parts in this design. (1) We use the circuit-level macro model, NeuroSim, to estimate the performance of our memristor and compare it with the benchmark devices [49]. (2) The neural network model shown in Figure 9 is used to carry out image

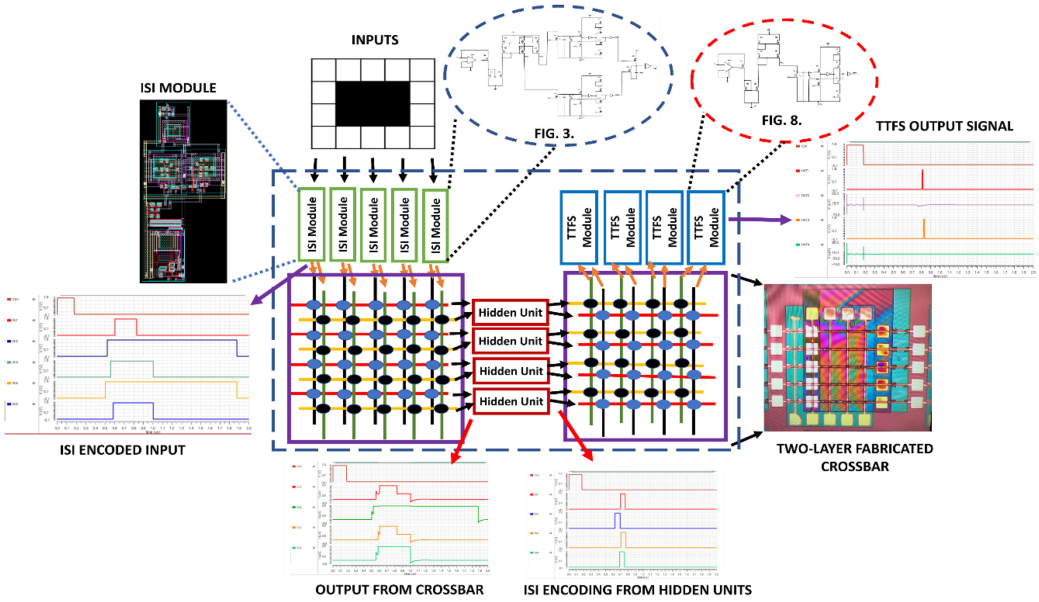


Fig. 9. Hardware simulation setup for our architecture.

Table 2. Comparison of the Memristor with Benchmark Devices[a]

Properties	[50]	[51]	[52]	This memristor	6-bit SRAM
R_{ON}	5MΩ	23MΩ	16.9kΩ	1MΩ	—
ON/OFF Ratio	2	6.84	4.43	~1000	—
Cycle-to-cycle variation	<1%	<1%	5%	4%	—
Online learning accuracy	~10%	~10%	~41%	~80%	94%

[a]The benchmark devices in this table are from the NeuroSim Simulator [49].

classification to demonstrate the accuracy on hardware using our developed architecture. (3) We use circuit analysis in Cadence Virtuoso to determine the area and power consumption of our architecture. (4) We further develop the large-scale model of our neural network on software for the purpose of comparing our ISI encoding scheme with various encoding schemes in the MNIST dataset.

4.1 Comparison of the Memristor with Benchmark Devices

The comparison between the memristor model and the benchmark devices is shown in Table 2. We use the NeuroSim Simulator which has the two-layer **multilayer perceptron (MLP)** neural network with the properties of the analog eNVM devices included in the weights to evaluate the online learning accuracy with the MNIST dataset [49]. We compare our memristor with the state-of-the-art devices used in the NeuroSim simulator [49]. The MLP in NeuroSim can evaluate learning accuracy and circuit-level performance specifically for the synaptic array during learning. When used in online learning, the MLP simulator imitates the hardware parameters in order to train the network with images picked randomly from the training dataset. The input images are converted to black and white 1 bit data to reduce complexity for hardware training and testing. The MLP simulator operates hierarchically, encompassing algorithm-level to device-level considerations. It takes into account detailed properties of synaptic arrays and realistic device behavior.

When benchmarking the devices using the NeuroSim simulator, we observed relatively low accuracies ranging from approximately 10% to 41%. One of the major factors contributing to these low accuracies is the limited on and off ratio of the fabricated memristor [49]. From Table 2 it can be observed that due to the high on and off ratio of our memristor model as well as its heat dissipation capability, it can achieve a very high accuracy of 80% compared to the benchmark devices. The high on and off ratio allows for a better differentiating capability between the signal states and further reduces the likelihood of errors between the different data states. Furthermore, it helps to minimize the noise interference during the readout process, allowing us to make a clear distinction between the noise and signal levels. It also allows us to achieve more precise control over synaptic weights, especially in SNNs. With enhanced weight precision, more accurate neural network behavior can be achieved. It also has a relatively competitive cycle-to-cycle variation that was measured using our setup from Figure 4.

4.2 Training and Classification of 5 × 4 Images

For our hardware simulations and evaluation of the accuracy of our architecture, we develop a three-layer neural network demonstrated in Figure 9. On Cadence Virtuoso, we combine the SPICE model of our memristor, and our circuit modules to create our architecture to evaluate it using handwritten images of digits from 0 to 9. Our output layer consists of a 4 × 4 memristor crossbar and therefore we use offline training to determine the weights to classify 4 digits, from 0 to 3, one for each class. For every input image, an entire row of pixel is processed at a time for one clock cycle. Our demonstration shows the results from applying the pixels from the image of the digit 0 in Figure 9.

The pixels of the input image 0 are applied to the input processing units where the signal passes on to the precision current to current converter where it is split into two values. After that it is passed on to the two LIF neurons to generate two different spikes at different times and they are then combined to produce a pulse signal. The pulse signal for the image 0 that is produced from the ISI encoding unit is shown in Figure 9 as the ISI encoded input. These pulse signals are then applied to the memristor crossbar containing the trained weights stored as conductance values. After carrying out the vector-matrix multiplication computations, the output column currents from the crossbar are applied to the intermediate stage where signal is postprocessed and ISI encoded into pulse signals to be applied to a 4 × 4 output crossbar stage. In the final stage, the output column currents are applied to the TTFS decoder where the currents are postprocessed before being applied to a single LIF neuron to generate an output spike signal. The output spike results from the image 0 is shown in Figure 9 as the TTFS output signal, where the weights are trained such that the first neuron spikes first.

Our design uses a clock frequency of 0.5MHz and a supply voltage of 1.8V. To demonstrate its spatiotemporal information processing capability each image is processed in 2μs. The images from 0 to 9 were used as inputs to the designed SNN structure to generate a TTFS-based classification output to classify them. The results from this data are summarized in Table 3 which shows the spike time and the neuron that spikes. From Table 3 we can observe that for the number 1 the neuron from column 2 spikes first at 1.50μs, for the number 2 the neuron from column 3 spikes first at 1.25μs and for the number 3 the neuron from column 4 spikes first at 993ns. Although our 4 × 4 output crossbar is only able to classify 4 digits, our results from Table 3 show that the spiking pattern from each output neuron is different and thus can successfully classify the images.

4.3 Power and Area Analysis

We design our circuits and architecture in Cadence Virtuoso to evaluate the area and power of the major components in our design. Figure 10 depicts the power and area breakdown of the

Table 3. TTFS Output from Each Column

Number	Column 1	Column 2	Column 3	Column 4
0	805ns	0	825ns	0
1	0	1.50μs	1.51μs	0
2	1.39μs	1.25μs	1.24μs	1.4μs
3	1.02μs	1.47μs	1.10μs	993ns
4	1.02μs	925ns	1.02μs	985ns
5	1.1μs	0	0	0
6	1.19μs	0	1.2μs	0
7	0	0	1.12μs	0
8	1.17μs	0	1.2μs	0
9	1.4μs	1.3μs	1.4μs	1.36μs

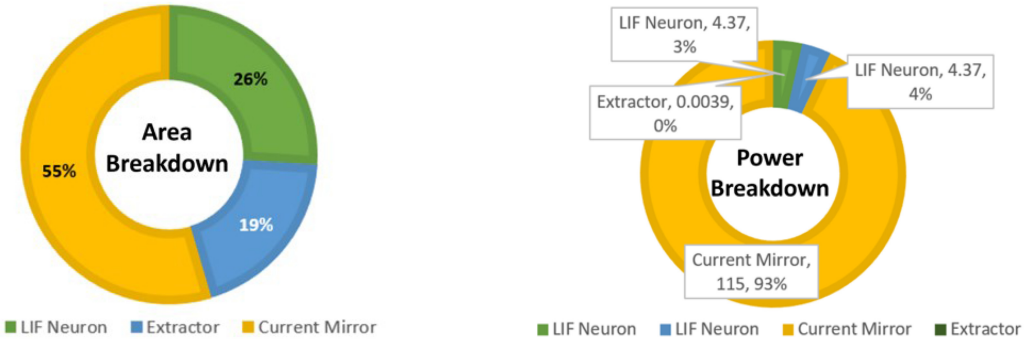


Fig. 10. Area and power breakdown.

components. The power distribution, as shown in Figure 10, demonstrates that for each processing unit the LIF neuron only consumes 4.37μW of power. The current mirror stage consumes the maximum amount of power of 115μW, arising from the use of the opamp. The least amount of power is consumed by the extractor stage which is almost insignificant. The combined power consumption of all modules in the input processing unit is 123μW. Our complete CIM-based SNN architecture consumes only 2.9mW when classifying handwritten digits. The area breakdown from Figure 10 shows that due to the opamp, the major portion of the area is consumed by the current mirror stage of 55%. Each LIF neuron consumes 26% of the area while the extractor unit consumes 19% of the area.

In Table 4 we summarize the comparison of our designed architecture with the state-of-the-art memristor-based neural network designs. For our classification application, the use of binary weights proved to be sufficient in capturing the necessary information and performing the required computations. We found that the precision provided by 1-bit weights yielded satisfactory accuracy for our specific task. Furthermore, employing 1-bit weights brings notable advantages in terms of hardware implementation and power consumption. The simplified representation of weights, using only 1 bit, reduces the complexity of the circuits, making them easier to design and fabricate. We compare our work with [16] and [53] that are memristor-based SNN architectures developed in the SPICE simulator. Our work is also evaluated against memristor-based **Restricted Boltzmann Machine (RBM)** [54] and MLP [55]. Compared to the other designs, our architecture has an extremely competitive power consumption and low latency of only 2.9mW and 334ns.

Table 4. Performance Comparison with the State-of-the-art Memristor-based Neural Network Designs

	[16]	[53]	[54]	[55]	This work
Technology	—	—	130nm	130nm	180nm
Algorithm	SNN	SNN	RBM	MLP	SNN
Memory Cell	ReRAM	ReRAM	ReRAM	ReRAM	ReRAM
Memory Mode	CIM	CIM	CIM	CIM	CIM
Weight Precision	—	—	1-bit signed	3-bit signed	1-bit
Neuron Type	LIF	LIF	IF	—	LIF
Supply Voltage	—	—	1.8V	5V	1.8V
Latency	5μs	~5μs	—	51.1ns	334ns
Power Consumption	16.71mW	—	2.2mW	—	2.9mW

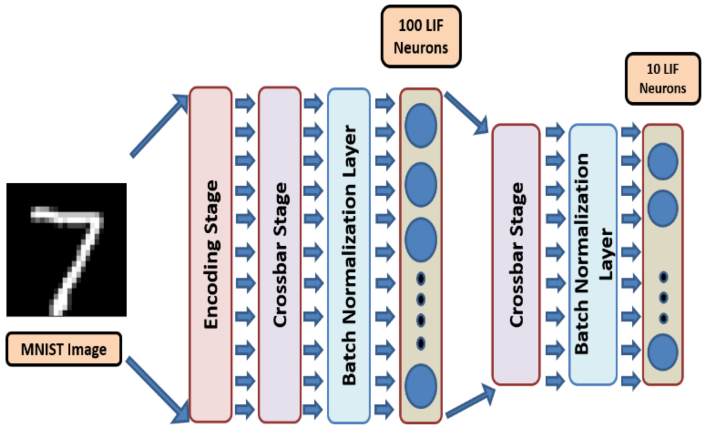


Fig. 11. Architecture of the CIM-based SNN for MNIST classification.

Furthermore, our design only consumes 2.51pJ of energy per synaptic connection, making it suitable to be used in AI accelerators.

4.4 Performance Comparison with Various Encoding Schemes

We developed a three-layer neural network to test our large-scale network. Our design incorporates the ISI encoding scheme as well as the memristor crossbar structure. Using this model, we compare the accuracy of our network against the MNIST dataset in terms of the different encoding schemes of rate, TTFS and ISI encoding. The detailed structure of the model is shown in Figure 11.

Modified from the neural network in [56], we develop our large-scale three-layer SNN model where we incorporate the encoding stage to initially code the inputs into ISI, TTFS or rate. To perform the CIM-based vector-matrix multiplication operations, we then apply the inputs to the memristor crossbar layer. In this stage, minibatches of 128×784 images are multiplied with the crossbar matrix of 784×100. We utilize the ADAM optimizer to train the memristor weights. By applying a clipping mechanism, where weight values above a certain threshold are set to a maximum value, the weights can be constrained within a specific range. This clipping helps prevent the weight values from growing too large and potentially causing instability during training. Additionally, scaling the weight values by an appropriate factor helps bring them closer to the magnitude of the gradients, making the two more compatible.

Table 5. Performance Comparison with Different Encoding Schemes for MNIST Classification

Encoding Scheme	Accuracy	Epochs
Rate	83% [56]	10
TTFS	85%	10
ISI	93%	10

To imitate the current mirror stage and decrease the number of training of epochs, we integrate a Batch Normalization layer in the next step [57]. Here, the output column currents are amplified and transformed into voltages and clamped between two pre-determined values. Clamping involves limiting the output values from the memristor crossbar within predetermined upper and lower bounds. By constraining the output values within a desirable range, extreme values that could disrupt training are avoided. Our final layer consists of a set of LIF neurons for the transmission of spikes and to allow the control of the dynamics from the first layer.

During the training process, careful hyperparameter tuning is performed to determine suitable upper and lower bounds for weight values, as well as an appropriate scaling factor. This tuning ensures that the weight values and gradients are in a reasonable range for effective learning. By finding the right balance, stability during training is maintained, and the updates to the gradients have a substantial impact relative to the weight values. Moreover, the learning rate, which determines the step size of weight updates, is adjusted to strike a balance between training accuracy and stability.

Following this layer, the second layer is placed similar to the former one, where the inputs are applied to the crossbar stage containing a weight matrix of 100×10, after which the resulting output current is converted to voltage using the Batch Normalization layer and clamped. The first and the second layer contain 100 and 10 LIF neurons, respectively. Since the MNIST image contains 10 different classes of images from 0–9, the output layer contains 10 LIF neurons. The accuracy results from each type of encoding scheme are summarized in Table 5. From our results, it was concluded that our design with the ISI encoding scheme is able to achieve the highest accuracy of 93%, compared to 83% in rate encoding and 85% in the TTFS encoding scheme.

5 CONCLUSION

SNNs are energy and power efficient biologically realistic models of neural networks due to their spatiotemporal information processing capability. In this paper we investigated and designed a novel ISI neural coding scheme to convert the incoming data into spikes. Our chosen scheme has a higher information density by encoding information in the times between the spikes and thus outperforms its TTFS and rate counterparts by ∼10%. A memristor crossbar is used for in-memory computation operations to carry out the vector-matrix multiplication process for feature extraction. Our memristor device that was used integrates a novel heat dissipation capability that significantly reduces the resistance variation, a key challenge in memristors. The incorporation of an additional heat dissipation layer in our memristor device contributes to its robustness by facilitating rapid heat removal and reducing resistance variation and enhances the device's stability, reliability, and performance. Our CIM-based SNN architecture discusses the detailed circuit design of each stage, including the input layer, hidden layers, and output layer, as well as the memristor crossbar stage. We further integrate a positive and negative weight mapping mechanism for the vector-matrix multiplication operations. The entire network also has a very low power consumption of only 2.9mW and our simulation results verify that our design can attain a better performance compared to other works of similar architecture.

REFERENCES

[1] Weibo Liu, Zidong Wang, Xiaohui Liu, Nianyin Zeng, Yurong Liu, and Fuad E. Alsaadi. 2017. A survey of deep neural network architectures and their applications. *Neurocomputing* 234 (2017), 11–26.

[2] Neil C. Thompson, Kristjan Greenewald, Keeheon Lee, and Gabriel F. Manso. 2020. The computational limits of deep learning. arxiv:2007.05558.

[3] Yiran Chen, Yuan Xie, Linghao Song, Fan Chen, and Tianqi Tang. 2020. A survey of accelerator architectures for deep neural networks. *Engineering* 6 (2020), 264–274.

[4] Abu Sebastian, Manuel Le Gallo, Riduan Khaddam-Aljameh, and Evangelos Eleftheriou. 2020. Memory devices and applications for in-memory computing. *Nature Nanotechnology* 15 (2020), 529–544.

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

[6] Ping Chi, Shuangchen Li, Cong Xu, Tao Zhang, Jishen Zhao, Yongpan Liu, Yu Wang, and Yuan Xie. 2016. PRIME: A novel processing-in-memory architecture for neural network computation in ReRAM-based main memory. *ACM SIGARCH Computer Architecture News* 44 (2016), 27–39.

[7] Ali Shafiee, Anirban Nag, Naveen Muralimanohar, Rajeev Balasubramonian, John Paul Strachan, Miao Hu, R. Stanley Williams, and Vivek Srikumar. 2016. ISAAC: A convolutional neural network accelerator with in-situ analog arithmetic in crossbars. *ACM SIGARCH Computer Architecture News* 44 (2016), 14–26.

[8] Linghao Song, Xuehai Qian, Hai Li, and Yiran Chen. 2017. PipeLayer: A pipelined ReRAM-based accelerator for deep learning. In *2017 IEEE International Symposium on High Performance Computer Architecture (HPCA)*. IEEE, 541–552.

[9] Ximing Qiao, Xiong Cao, Huanrui Yang, Linghao Song, and Hai Li. 2018. AtomLayer: A universal ReRAM-based CNN accelerator with atomic layer computation. In *Proceedings of the 55th Annual Design Automation Conference*, 1–6.

[10] Samanwoy Ghosh-Dastidar and Hojjat Adeli. 2009. Spiking neural networks. *International Journal of Neural Systems* 19 (2009), 295–308.

[11] Yujie Wu, Lei Deng, Guoqi Li, Jun Zhu, and Luping Shi. 2018. Spatio-temporal backpropagation for training high-performance spiking neural networks. *Frontiers in Neuroscience* 12 (2018), 331.

[12] Wolfgang Maass. 1997. Networks of spiking neurons: The third generation of neural network models. *Neural Networks* 10 (1997), 1659–1671.

[13] Jason K. Eshraghian, Xinxin Wang, and Wei D. Lu. 2022. Memristor-based binarized spiking neural networks: Challenges and applications. *IEEE Nanotechnology Magazine* 16 (2022), 14–23.

[14] Aayush Ankit, Abhronil Sengupta, Priyadarshini Panda, and Kaushik Roy. 2017. RESPARC: A reconfigurable and energy-efficient architecture with memristive crossbars for deep spiking neural networks. In *Proceedings of the 54th Annual Design Automation Conference 2017*, 1–6.

[15] Liang Zhao, Qinghui Hong, and Xiaoping Wang. 2018. Novel designs of spiking neuron circuit and STDP learning circuit based on memristor. *Neurocomputing* 314 (2018), 207–214.

[16] Jiwei Li, Hui Xu, Sheng-Yang Sun, Nan Li, Qingjiang Li, Zhiwei Li, and Haijun Liu. 2021. In-situ learning in hardware compatible multi-layer memristive spiking neural network. *IEEE Transactions on Cognitive and Developmental Systems* (2021).

[17] Xumeng Zhang, Jian Lu, Zhongrui Wang, Rui Wang, Jinsong Wei, Tuo Shi, Chunmeng Dou, Zuheng Wu, Jiaxue Zhu, and Dashan Shang. 2021. Hybrid memristor-CMOS neurons for in-situ learning in fully hardware memristive spiking neural networks. *Science Bulletin* 66 (2021), 1624–1633.

[18] Hongyu An, Mohammad Shah Al-Mamun, Marius K. Orlowski, Lingjia Liu, and Yang Yi. 2021. Three-dimensional neuromorphic computing system with two-layer and low-variation memristive synapses. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 41 (2021), 400–409.

[19] H.-S. Philip Wong, Heng-Yuan Lee, Shimeng Yu, Yu-Sheng Chen, Yi Wu, Pang-Shiu Chen, Byoungil Lee, Frederick T. Chen, and Ming-Jinn Tsai. 2012. Metal–oxide RRAM. *Proceedings of the IEEE* 100 (2012), 1951–1970.

[20] Hongyu An, Mohammad Shah Al-Mamun, Marius K. Orlowski, Lingjia Liu, and Yang Yi. 2020. Robust deep reservoir computing through reliable memristor with improved heat dissipation capability. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 40 (2020), 574–583.

[21] Hongyu An, Kangjun Bai, and Yang Yi. 2018. The roadmap to realize memristive three-dimensional neuromorphic computing system. *Advances in Memristor Neural Networks-Modeling and Applications* (2018), 25–44.

[22] Chenyuan Zhao, Bryant T. Wysocki, Yifang Liu, Clare D. Thiem, Nathan R. McDonald, and Yang Yi. 2015. Spike-time-dependent encoding for neuromorphic processors. *ACM Journal on Emerging Technologies in Computing Systems (JETC)* 12 (2015), 1–21.

[23] Qiang Yu, Huajin Tang, Kay Chen Tan, and Haoyong Yu. 2014. A brain-inspired spiking neural network model with temporal encoding and learning. *Neurocomputing* 138 (2014), 3–13.

[24] Seongsik Park, Seijoon Kim, Byunggook Na, and Sungroh Yoon. 2020. T2FSNN: Deep spiking neural networks with time-to-first-spike coding. In *2020 57th ACM/IEEE Design Automation Conference (DAC)*. IEEE, 1–6.

[25] Jaehyun Kim, Heesu Kim, Subin Huh, Jinho Lee, and Kiyoung Choi. 2018. Deep neural networks with weighted spikes. *Neurocomputing* 311 (2018), 373–386.
[26] Seongsik Park, Seijoon Kim, Hyeokjun Choe, and Sungroh Yoon. 2019. Fast and efficient information transmission with burst spikes in deep spiking neural networks. In *2019 56th ACM/IEEE Design Automation Conference (DAC)*. IEEE, 1–6.
[27] Jesus L. Lobo, Javier Del Ser, Albert Bifet, and Nikola Kasabov. 2020. Spiking neural networks and online learning: An overview and perspectives. *Neural Networks* 121 (2020), 88–100.
[28] Eugene M. Izhikevich. 2003. Simple model of spiking neurons. *IEEE Transactions on Neural Networks* 14 (2003), 1569–1572.
[29] Alan L. Hodgkin and Andrew F. Huxley. 1952. A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of Physiology* 117 (1952), 500.
[30] Roberto A. Vazquez and Aleister Cachón. 2010. Integrate and fire neurons and their application in pattern recognition. In *2010 7th International Conference on Electrical Engineering Computing Science and Automatic Control*. IEEE, 424–428.
[31] Edgar D. Adrian. 1926. The impulses produced by sensory nerve endings: Part I. *The Journal of Physiology* 61 (1926), 49.
[32] Daniel A. Butts, Chong Weng, Jianzhong Jin, Chun-I. Yeh, Nicholas A. Lesica, Jose-Manuel Alonso, and Garrett B. Stanley. 2007. Temporal precision in the neural code and the timescales of natural vision. *Nature* 449 (2007), 92–95.
[33] Marcelo A. Montemurro, Malte J. Rasch, Yusuke Murayama, Nikos K. Logothetis, and Stefano Panzeri. 2008. Phase-of-firing coding of natural visual stimuli in primary visual cortex. *Current Biology* 18 (2008), 375–380.
[34] Bing Han, Gopalakrishnan Srinivasan, and Kaushik Roy. 2020. RMP-SNN: Residual membrane potential neuron for enabling deeper high-accuracy and low-latency spiking neural network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 13558–13567.
[35] Peter U. Diehl, Daniel Neil, Jonathan Binas, Matthew Cook, Shih-Chii Liu, and Michael Pfeiffer. 2015. Fast-classifying, high-accuracy spiking deep networks through weight and threshold balancing. In *2015 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 1–8.
[36] Bodo Rueckauer, Iulia-Alexandra Lungu, Yuhuang Hu, Michael Pfeiffer, and Shih-Chii Liu. 2017. Conversion of continuous-valued deep networks to efficient event-driven networks for image classification. *Frontiers in Neuroscience* 11 (2017), 682.
[37] Alejandro Linares-Barranco, D. Cascado, Gabriel Jimenez, Antón Civit, Matthias Oster, and Bernabé Linares-Barranco. 2006. Poisson AER generator: Inter-spike-intervals analysis. In *2006 IEEE International Symposium on Circuits and Systems*. IEEE, 4 (2006), 3152.
[38] Chenyuan Zhao, Yang Yi, Jialing Li, Xin Fu, and Lingjia Liu. 2017. Interspike-interval-based analog spike-time-dependent encoder for neuromorphic processors. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 25 (2017), 2193–2205.
[39] Daniele Ielmini and H.-S. Philip Wong. 2018. In-memory computing with resistive switching devices. *Nature Electronics* 1 (2018), 333–343.
[40] Victor Zhirnov, Ralph Cavin, and Luca Gammaitoni. 2014. Minimum energy of computing, fundamental considerations. In *ICT-Energy-Concepts Towards Zero-Power Information and Communication Technology IntechOpen*.
[41] Jagan Singh Meena, Simon Min Sze, Umesh Chand, and Tseung-Yuen Tseng. 2014. Overview of emerging nonvolatile memory technologies. *Nanoscale Research Letters* 9 (2014), 1–33.
[42] Anping Huang, Xinjiang Zhang, Runmiao Li, and Yu Chi. 2018. Memristor neural network design. *Memristor and Memristive Neural Networks*, 1–35.
[43] Wenzhe Guo, Mohammed E. Fouda, Ahmed M. Eltawil, and Khaled Nabil Salama. 2021. Neural coding in spiking neural networks: A comparative study for robust neuromorphic systems. *Frontiers in Neuroscience* 15 (2021), 638474.
[44] Fabiha Nowshin, Lingjia Liu, and Yang Yi. 2020. Energy efficient and adaptive analog IC design for delay-based reservoir computing. In *2020 IEEE 63rd International Midwest Symposium on Circuits and Systems (MWSCAS)*. IEEE, 592–595.
[45] Fabiha Nowshin and Yang Yi. 2022. Memristor-based deep spiking neural network with a computing-in-memory architecture. In *2022 23rd International Symposium on Quality Electronic Design (ISQED)*. IEEE, 1–6.
[46] Mohammad Al-Mamun, Sean W. King, Srilekhya Meda, and Marius K. Orlowski. 2018. Impact of the heat conductivity of the inert electrode on ReRAM performance and endurance. *ECS Transactions* 85 (2018), 207.
[47] Kangjun Bai, Lingjia Liu, and Yang Yi. 2021. Spatial-temporal hybrid neural network with computing-in-memory architecture. *IEEE Transactions on Circuits and Systems I: Regular Papers* 68 (2021), 2850–2862.
[48] Can Li, Miao Hu, Yunning Li, Hao Jiang, Ning Ge, Eric Montgomery, Jiaming Zhang, Wenhao Song, Noraica Dávila, and Catherine E. Graves. 2018. Analogue signal and image processing with large memristor crossbars. *Nature Electronics* 1 (2018), 52–59.

[49] Pai-Yu Chen and Shimeng Yu. 2018. Technological benchmark of analog synaptic devices for neuroinspired architectures. *IEEE Design & Test* 36 (2018), 31–38.

[50] Ligang Gao, I-Ting Wang, Pai-Yu Chen, Sarma Vrudhula, Jae-sun Seo, Yu Cao, Tuo-Hung Hou, and Shimeng Yu. 2015. Fully parallel write/read in resistive synaptic array for accelerating on-chip learning. *Nanotechnology* 26 (2015), 455204.

[51] S. Park, A. Sheri, J. Kim, J. Noh, J. Jang, M. Jeon, B. Lee, B. R. Lee, B. H. Lee, and H. Hwang. 2013. Neuromorphic speech systems using advanced ReRAM-based synapse. In *2013 IEEE International Electron Devices Meeting*. IEEE, 25.26. 21-25.26. 24.

[52] Jiyong Woo, Kibong Moon, Jeonghwan Song, Sangheon Lee, Myounghun Kwak, Jaesung Park, and Hyunsang Hwang. 2016. Improved synaptic behavior under identical pulses using AlO x/HfO 2 bilayer RRAM array for neuromorphic systems. *IEEE Electron Device Letters* 37 (2016), 994–997.

[53] Yaozhong Zhang, Mingxuan Jiang, Xiaoping Wang, and Zhigang Zeng. 2022. A novel two-layer memristive spiking neural network with spatio-temporal backpropagation. In *2022 14th International Conference on Advanced Computational Intelligence (ICACI)*. IEEE, 160–165.

[54] Weier Wan, Rajkumar Kubendran, S. Burc Eryilmaz, Wenqiang Zhang, Yan Liao, Dabin Wu, Stephen Deiss, Bin Gao, Priyanka Raina, and Siddharth Joshi. 2020. 33.1 A 74 TMACS/W CMOS-RRAM neurosynaptic core with dynamically reconfigurable dataflow and in-situ transposable weights for probabilistic graphical models. In *2020 IEEE International Solid-State Circuits Conference-(ISSCC)*. IEEE, 498–500.

[55] Qi Liu, Bin Gao, Peng Yao, Dong Wu, Junren Chen, Yachuan Pang, Wenqiang Zhang, Yan Liao, Cheng-Xin Xue, and Wei-Hao Chen. 2020. 33.2 A fully integrated analog ReRAM based 78.4 TOPS/W compute-in-memory chip with fully parallel MAC computing. In *2020 IEEE International Solid-State Circuits Conference-(ISSCC)*. IEEE, 500–502.

[56] Qingxi Duan, Zhaokun Jing, Xiaolong Zou, Yanghao Wang, Ke Yang, Teng Zhang, Si Wu, Ru Huang, and Yuchao Yang. 2020. Spiking neurons with spatiotemporal dynamics and gain modulation for monolithically integrated memristive neural networks. *Nature Communications* 11 (2020), 1–13.

[57] Fabian Schilling. 2016. The effect of batch normalization on deep convolutional neural networks.

Received 1 September 2022; revised 23 June 2023; accepted 8 November 2023