ADVANCED INTELLIGENT SYSTEMS
Open Access
www.advintellsyst.com

Analysis of a Memcapacitor-Based Online Learning Neural Network Accelerator Framework

*Ankur Singh, Dowon Kim, and Byung-Geun Lee**

Data-intensive computing tasks, such as training neural networks, are fundamental to artificial intelligence applications but often demand substantial energy resources. This study presents a novel complementary metal-oxide-semiconductor (CMOS)-based memcapacitor framework designed to address these challenges by enabling efficient and robust neuromorphic computing. Utilizing memcapacitor devices, a crossbar array that performs parallel vector-matrix multiplication operations, validated through cadence simulations and implemented in python for scalable accelerator design, is developed. The framework demonstrates outstanding performance across classification tasks, achieving 98.4% accuracy in digit recognition and 85.9% in object recognition. A key aspect of this research is its focus on real-world fabrication nonidealities, including up to 30% device parameter variations, ensuring robustness and reliability under practical deployment conditions. The results emphasize the effectiveness of capacitance-based systems in handling classification tasks while demonstrating resilience to fabrication-induced variations. This work establishes a foundation for scalable, energy-efficient, and robust memcapacitor-based neural networks, advancing the potential for intelligent systems in artificial intelligence-driven applications and paving the way for future innovations in neuromorphic computing.

1. Introduction

Memelements have emerged as a promising class of devices, demonstrating remarkable performance, particularly when deployed in crossbar architectures.[1–3] Their integration into these structures significantly enhances the efficiency of vector-matrix multiplication (VMM) by enabling the parallel execution of product and summation operations through the devices. This capability is particularly beneficial in the domain of convolutional neural networks (CNNs), where extensive matrix operations are fundamental to both training and inference processes. The combination of in-memory computing (IMC) architectures with the adjustable analog memductance of memelements further contributes to power-efficient VMM and training, enabling the development of highly integrated memory architectures. Consequently, a wide array of CNN hardware designs utilizing memelements-based VMM accelerators[3–6] has been proposed, with their effectiveness consistently demonstrated in various studies.

Neuromorphic computing, modeled after brain-like processes and grounded in artificial neural networks, presents effective solutions for a wide range of computationally demanding tasks. Originally conceptualized in the 1980s,[7,8] this field has seen substantial progress with the advent of memristive devices[9] and the introduction of convolutional layers in deep neural networks.[10,11] These innovations have facilitated the development of various resistive neuromorphic systems that employ devices such as memristor,[12–16] phase-change memory,[17] spintronic devices,[18,19] and ferroelectric components, including ferroelectric tunnel junctions[20,21] and ferroelectric field-effect transistors.[22,23] Among these, devices like ferroelectric tunnel junctions and silicon–oxide–nitride–oxide–silicon transistors have achieved remarkable energy efficiencies, reaching up to 100 tera-operations per second per watt.[24] These neuromorphic systems operate by storing synaptic weights in analog form for multiplication tasks, utilizing Kirchhoff's current law to sum currents through crossbar arrays.

Memcapacitive devices, akin to memristive devices but operating on a capacitive principle, hold the promise of lower static power consumption.[25] While theoretical models for memcapacitive devices have been extensively explored,[25–29] practical implementations remain relatively scarce.[30–33] These devices can be realized through various mechanisms, such as implementing a variable-plate distance concept in microelectromechanical systems,[34] utilizing a metal-to-insulator transition material in series with a dielectric layer,[29] modulating the oxygen vacancy front in a traditional memristor,[27] and employing a simple metal–oxide–semiconductor capacitor with memory effects.[31,32] Despite the potential advantages, achieving a high dynamic range in memcapacitive devices poses significant challenges. For instance, devices with a variable-plate distance often encounter large parasitic resistive components when the plate distance is minimal, hindering their performance.[27]

A. Singh, D. Kim, B.-G. Lee
School of Electrical Engineering and Computer Science
Gwangju Institute of Science and Technology
Gwangju 61005, Republic of Korea
E-mail: bglee@gist.ac.kr

The ORCID identification number(s) for the author(s) of this article can be found under https://doi.org/10.1002/aisy.202400795.

DOI: 10.1002/aisy.202400795

Conversely, at large plate distances, these devices face limited lateral scalability. Similar issues arise in memcapacitors designed with varying surface areas[34] or dielectric constants,[33] as these approaches also struggle with scalability and efficiency.

In our study, we introduce a cutting-edge memcapacitor-based neural network framework tailored for data-intensive computing tasks, such as neural network training, which are essential for artificial intelligence (AI) applications but often involve significant energy consumption. We present a novel CMOS-based memcapacitor circuit, validated using the cadence tool, and developed a Python-based model to facilitate the design of a memcapacitive-based accelerator. This framework employs a $5 \times 5 \times 5$ crossbar array of memcapacitor devices to train a neural network for digit classification and canadian institute for advanced research (CIFAR) dataset recognition. Our system effectively addresses the nonideal characteristics of memcapacitive devices, achieving remarkable training accuracies of 98.4% in digit recognition and 94.4% in CIFAR recognition. This research demonstrates the efficacy of memcapacitor-based neural networks in handling classification tasks, showcasing their potential in reducing the energy demands of data-intensive computing. The integration of memcapacitor technology within neural networks not only enhances computational efficiency but also underscores the viability of these systems in neuromorphic computing. Our work sets the stage for further advancements by illustrating the practical applications of memcapacitors in neural network training and classification tasks. The significant training accuracies achieved highlight the promise of memcapacitor-based systems in AI, suggesting a substantial impact on future developments in energy-efficient and high-performance computing.

2. Proposed Memcapacitor Device

2.1. CMOS-Based Model

The memcapacitor represents the relationship between the time integral of charge and flux, defined by the inverse memcapacitance function $M_C(q)$, with constants α and β. The mathematical model of a charge-controlled memcapacitor is expressed as follows:[35]

$$M_C = \frac{V_{in}(t)}{q(t)} = \beta \pm \alpha\sigma(t) \tag{1}$$

The emulator proposed in this work comprises three functional blocks: two operational transconductance amplifiers (OTA) and a buffer. These blocks are interconnected to achieve the desired memcapacitor characteristic, which relates the input voltage V_{in} to the charge $q(t)$. The overall design of the memcapacitor emulator is illustrated in **Figure 1**. In this design, the body terminals of all P-channel metal-oxide semiconductor (PMOS) and N-channel metal-oxide semiconductor (NMOS) transistors are connected to V_{DD} and V_{SS}, respectively. The port characteristics of the OTA[36] can be mathematically defined by the following equations:

$$I_{O_i^+} = g_{mi}(V_{pi} - V_{ni}) \tag{2}$$

where g_{mi}, V_{pi}, and V_{ni} represent the transconductance gain and input voltage of each OTA. The routine analysis yields the following expression for g_m:

$$g_{mi} = K(V_{bi} - V_{SS} - V_{th}) \tag{3}$$

"i" is the OTA number, V_{ss} is the supply voltage of the OTA, V_{th} is the threshold voltage of metal-oxide-semiconductor field-effect transistor (MOSFET) device, and "K" is a parameter of the MOSFET device given by

$$K = \mu_n C_{OX} \frac{W}{L} \tag{4}$$

In this formulation, W represents the channel width, L denotes the channel length, μ_n is the mobility of the carrier, and C_{OX} refers to the oxide capacitance per unit area in the MOSFET.

The detailed working principle of the proposed emulator is analyzed through these functional blocks. OTA-1 and OTA-2, both OTAs, are crucial in defining the dynamic relationship between the charge and the flux. OTA-3, a buffer, ensures proper signal conditioning and stability of the emulator output. This configuration effectively replicates the theoretical behavior of a memcapacitor, providing a practical and implementable circuit design. The analysis shows how the integration of these blocks results in the desired emulation between the input voltage and the resulting charge over time.

In OTA-1, the positive terminal of the input signal V_{inP} is connected. It is important to note that V_{in}'s positive and negative terminals are labeled V_{inP} and V_{inN}, respectively. The transistors M12 and M13 form an input differential pair and are biased by the current sink transistor M14, as illustrated in Figure 1 for OTA-1. This functional block converts the differential voltage into a current, as described below:

$$\begin{aligned} I_{O_1^+} &= g_{m1}(V_{inP} - V_{n1}) \\ I_{O_1^+} &= g_{m1} V_{inP} \end{aligned} \tag{5}$$

In this configuration, g_{m1} represents the transconductance of OTA-1, with V_{n1} grounded as depicted in Figure 1. Functional Block-3 functions as a buffer, where the input signal is applied to the gate of transistor M23. Meanwhile, the gate of transistor M22 is connected to its drain terminal, establishing a negative feedback loop as illustrated in Figure 1. The corresponding analytical model for Functional Block-3 is provided below.

$$I_{22} = g_{22}(V_{P`} - V_X) = g_{23}(V_{N`} - V_X) \tag{6}$$

Given that $g_{22} = g_{23}$, it follows that $V_{P'} = V_{N'}$. Consequently, $V_{N'}$ at the output terminal mirrors the input $V_{P'}$, which is essential for ensuring the reliable operation of the proposed emulator.

The behavior of the memcapacitor is demonstrated through the analysis of the functional blocks shown in Figure 1. A sinusoidal input signal is applied to the proposed emulator between V_{inP} and V_{inN}, as illustrated below.

$$V_{in} = V_{inP} - V_{inN} \tag{7}$$

In the proposed design, a sinusoidal signal is applied to V_{in} across C_1, causing the input current (I_{in}) to flow through it. The voltage across C_1 is mathematically expressed as follows:

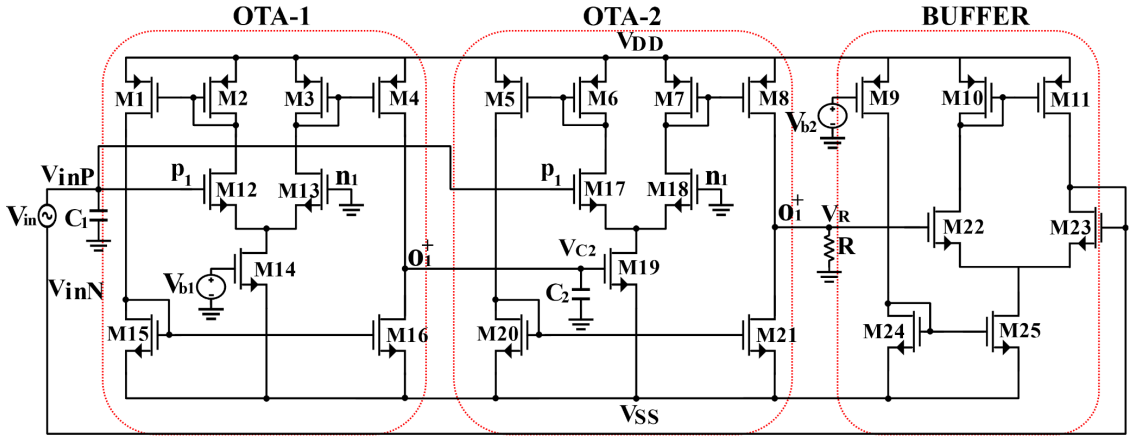

Figure 1. Proposed MOSFET-based memcapacitor circuit.

$$V_{C_1} = V_{\text{in}P} = \frac{1}{C_1}\int I_{\text{in}}dt = \frac{q(t)}{C_1} \tag{8}$$

The output current of OTA-1 is $(g_{m1} * q(t))/C_1$. The output terminal of OTA is connected to V_{b2} of OTA-2, which is the voltage across C_2 and can be expressed as

$$V_{C_2} = V_{b2} = \frac{1}{C_2}\int g_{m1}V_{inP}dt = \frac{g_{m1}}{C_1C_2}\sigma(t) \tag{9}$$

Changing C_2 alters V_{b2} due to the variation in charge, where V_{b2} can be referred to as the charge control voltage. Upon analyzing OTA-2, the output current can be expressed by the following equation:

$$\begin{aligned} I_{O_2^+} &= g_{m2}(V_{p2} - V_{n2}) \\ I_{O_2^+} &= g_{m2}V_{inP} \end{aligned} \tag{10}$$

The above current flows through R generating a voltage V_R as shown below.

$$\begin{aligned} V_R &= I_{O_2^+}R \\ V_R &= g_{m2}V_{\text{in}P}R \end{aligned} \tag{11}$$

Substituting V_{b2} depicted in Equation (9) in Equation (3), g_{m2} can be expressed as

$$g_{m2} = K\left[\frac{g_{m1}\sigma(t)}{C_1C_2} - V_{SS} - V_{th}\right] \tag{12}$$

Now putting Equation (12) into Equation (11) and $V_R = V_{P'} = V_N$ from the buffer, we get

$$V_R = V_{P'} = V_{\text{in}N} = \frac{q(t)RK}{C_1}\left[\frac{g_{m1}\sigma(t)}{C_1C_2} - V_{SS} - V_{th}\right] \tag{13}$$

Consequently, by Substituting Equation (13) into Equation (7), the corresponding memcapacitance can be calculated as follows:

$$\begin{aligned} V_{\text{in}}(t) &= \frac{q(t)}{C_1} - \frac{q(t)RK}{C_1}\left[\frac{g_{m1}\sigma(t)}{C_1C_2} - V_{SS} - V_{th}\right] \\ M_C &= \frac{V_{\text{in}}(t)}{q(t)} = \frac{1}{C_1}[1 + V_{SS} + V_{th}] - \frac{RK}{C_1}\left[\frac{g_{m1}\sigma(t)}{C_1C_2}\right] \end{aligned} \tag{14}$$

The proposed design features a memcapacitor model, initiating with a memcapacitance value calculated as $[(1/C_1)(1 + V_{SS} + V_{th})]$ according to the formula provided above. The rate at which the capacitance changes is determined by $[((RK/C_1)(g_{m1}*)/C_1 * C_2)]$. Here g_{m1} represents the transconductance values of the OTA-1.[37]

The simulation setup for the proposed work comprises cadence virtuoso software on the (TSMC) taiwan semiconductor manufacturing company 180-nmtechnology node. All the simulations were carried out in analog design environment window having 27 °C as the nominal temperature setting and the plot shown in **Figure 2**.[37] Also, the proposed work employs only a positive power supply, that is, V_{DD} which is taken to be 1.8 V_T.

2.2. Memcapacitor Spice Model

The proposed memcapacitor circuit is meticulously designed using TSMC 180 nm CMOS technology, tailored for

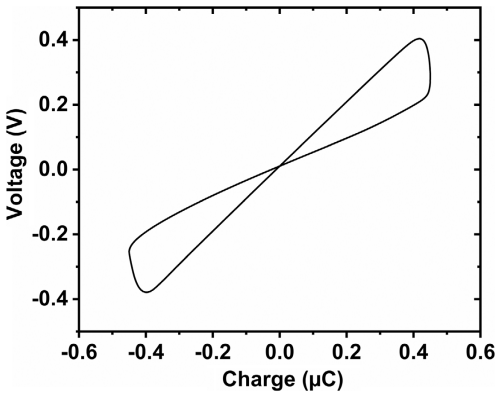

Figure 2. Hysteresis curve of the proposed memcapacitor circuit.

implementing VMM—a fundamental operation in CNN. This circuit leverages the nonlinear characteristics of memcapacitors to perform computational tasks typically associated with neuromorphic computing. The model is built using a combination of NMOS and PMOS transistors with varying channel lengths and widths, carefully selected to optimize the circuit's performance in terms of speed, power consumption, and accuracy. The circuit architecture involves the strategic placement of MOSFETs that act as switches, enabling the memcapacitor to store and manipulate charge in a way that emulates synaptic behavior in biological neurons. The SPICE model incorporates three different CMOS configurations to accommodate the various transistor dimensions required for different stages of the circuit.[38] Additionally, passive components like capacitors and resistors are integrated into the design to stabilize the circuit and control the flow of current.

A sinusoidal voltage source is applied as the input to the circuit, with parameters such as amplitude and frequency finely tuned to mimic the dynamic signals encountered in neuromorphic systems.[39] The transient simulation of this circuit is performed using a step-by-step approach, where the time-dependent voltage across different nodes is analyzed to verify the memcapacitor's performance. This model is pivotal for simulating the behavior of the memcapacitor in a controlled environment, providing insights into its potential application in energy-efficient, high-performance neuromorphic computing.[40] To systematically develop and validate the memcapacitor model, an algorithm has been structured, as illustrated in **Algorithm 1**. The algorithm outlines the following critical steps for implementing the memcapacitor model in python.

Algorithm 1. Construct Memcapacitor Model.

Input:
C_{name}: *Define circuit name*
L_{CMOS}: *CMOS libraries*
M_{NMOS}, M_{PMOS}: *Transistor models*
C_1, C_2: *Capacitors*
R_1: *Resistor*
V_{DD}, V_{SS}, V_{bias}: *Voltage sources*
V_{in}: *Input signal*
P_{sim}: *Simulation parameters*

Output: *Simulated transient response, voltage plots, circuit performance metrics, capacitance*

Initialize C_{name} *and include* L_{CMOS}

Define *transistor parameters* M_{NMOS}, M_{PMOS} *(e.g.* ***W, L****)*

Construct the Circuit:

Add transistor M_{NMOS}, M_{PMOS}

Insert Capacitors C_1, C_2

Place resistor R_1

Apply voltage sources V_{DD}, V_{SS}, V_{bias}

Configure input signal V_{in}

Run transient simulation with P_{sim}

Analyze and plot simulation results

3. Memcapacitive-Based Vector Matrix Multiplication

The memcapacitor-based VMM is a novel and promising approach in the field of neuromorphic computing, where the goal is to mimic the way the human brain processes information.[39] In traditional digital systems, VMM is a core operation in machine learning algorithms, particularly in CNN. However, as these systems scale, the energy consumption and speed of conventional hardware become significant bottlenecks.[41] This is where memcapacitors, with their unique ability to retain information and perform nonlinear operations, offer a distinct advantage. Memcapacitors are passive electronic components that combine the properties of memory and capacitance. Unlike traditional capacitors, which store charge linearly, memcapacitors can store different amounts of charge based on their history of applied voltage, making them ideal for use in systems that require complex, nonlinear operations such as VMM. In a memcapacitor VMM circuit, the capacitance of each memcapacitor represents the synaptic weight in a neural network, while the input voltage corresponds to the input signals applied to the neurons.[42] This setup allows the circuit to perform multiplication and accumulation in a single step, significantly reducing the computational overhead. The VMM can be represented mathematically as

$$(y_1 \ldots . y_n) = (x_1 \ldots . x_m) . \begin{pmatrix} w_{11} & \cdots & w_{1n} \\ \vdots & \ddots & \vdots \\ w_{m1} & \cdots & w_{mn} \end{pmatrix} \tag{15}$$

The output vector y is calculated by the inner product of the input vector x and weight matrix w and each element of the output vector y_i can be expressed as follows:

$$y_i = \sum_{k=1}^{m} x_k \cdot w_{ki} \tag{16}$$

In memcapacitor-based VMM, as shown in **Figure 3**, the weight matrix is implemented by a crossbar memcapacitor array with m-rows and n-columns.[43] The rows and columns of the memcapacitor array are connected to m input pulses and n output, respectively.

4. Results and Discussion

In this section, we present the implementation and evaluation of a memcapacitor-based crossbar array for image classification tasks using two benchmark datasets: Modified national institute of standards and technology (MNIST) and CIFAR. The proposed crossbar array, designed using TSMC 180 nm CMOS technology as outlined in Algorithm 1, serves as the computational backbone for efficient VMM—a fundamental operation in CNN. Furthermore, we investigate the impact of device variations and capacitance changes that may arise due to fabrication inconsistencies, specifically analyzing their effects on the MNIST dataset.

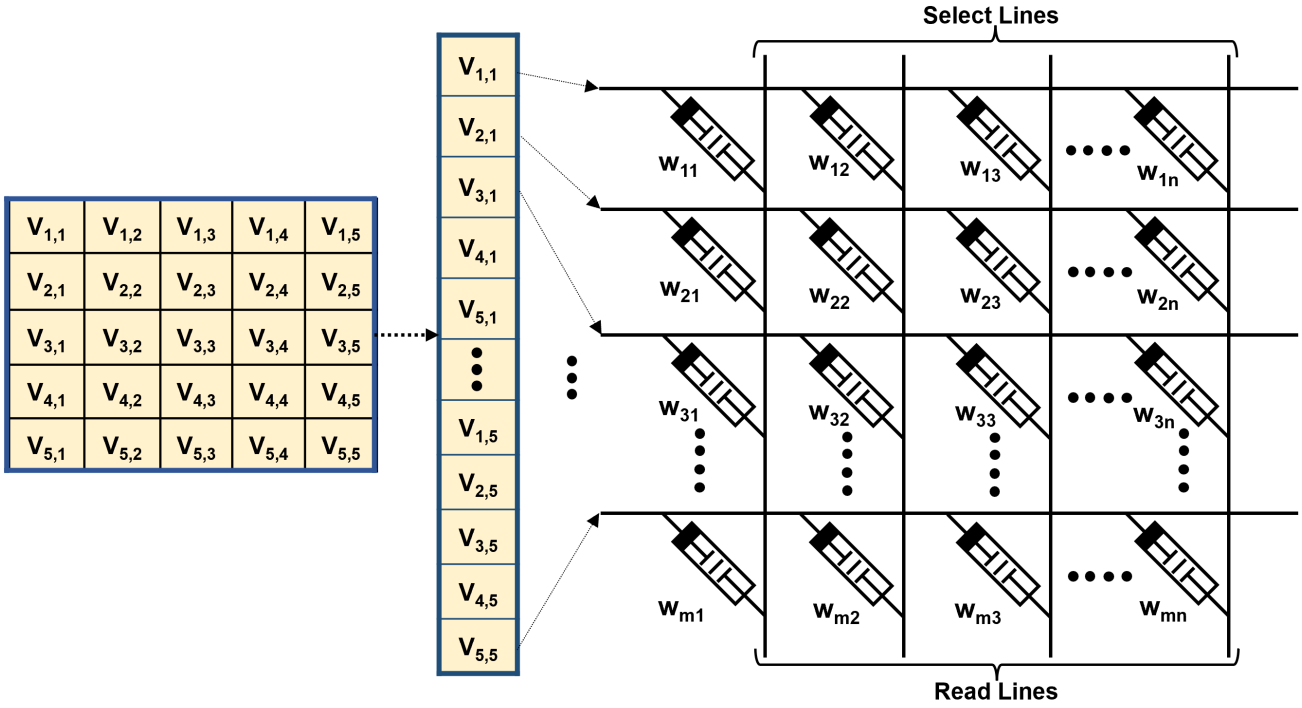

Figure 3. Crossbar structure defined in the simulator using memcapacitor device.

4.1. Object and Digit Classification

The MNIST dataset is a widely recognized benchmark for image classification. It consists of 70 000 grayscale images of handwritten digits ranging from 0 to 9, each with dimensions 28×28 pixels. For our experiments, we partitioned the dataset into 40 000 images for training and 40 000 for testing. Given its relatively simple structure, MNIST serves as an excellent platform for demonstrating the efficiency of our proposed architecture. In contrast, the CIFAR dataset poses a greater classification challenge due to its complexity. It comprises 60 000 color images across 10 categories, with 50 000 images allocated for training and 10 000 for testing. The CIFAR dataset requires deeper networks and larger computational capacity to extract complex features such as shapes, edges, and textures, making it an ideal test case for evaluating the scalability and versatility of our memcapacitor-based system.

The core of our proposed framework is a memcapacitor-based crossbar array. Memcapacitors, as charge-based devices, represent the weights of a neural network through their capacitance values. These capacitance states enable highly efficient IMC, where computation and storage are integrated within the same device. This architecture eliminates the need for frequent data transfers between memory and processing units, a significant drawback of conventional von Neumann systems. Consequently, the memcapacitor-based approach reduces latency and power consumption while achieving high computational throughput. In our system, input image pixel intensities are applied as voltages across the rows of the crossbar array. The trained weights, represented by the capacitance levels in the memcapacitors, are multiplied with these input voltages in analog form. The resulting charge is summed along the columns of the array to complete the VMM operation. This architecture ensures high-speed computation and seamless scalability for larger datasets.

For the MNIST dataset, we implemented a $5 \times 5 \times 5$ VMM configuration as the first convolutional layer of the neural network. To optimize computational efficiency, the input images were resized from 28×28 to 20×20 pixels before processing. The initial capacitance values of the memcapacitors were determined using Equation (14), and the VMM weights were iteratively updated during each training epoch to enhance the model's performance. As illustrated in **Figure 4**, the memcapacitor-based VMM efficiently processes the input data, extracting essential features such as edges, curves, and shapes. The processed outputs are subsequently passed through a neural network comprising (ReLU) rectified linear unit activation functions, a dense layer with 64 units, and a final softmax output layer for classification. **Algorithm 2** was employed toimplement the system for MNIST classification. The system achieved a training accuracy of 98.4% and a testing accuracy of 94.6%, as depicted in **Figure 5**a. A detailed performance analysis is further provided through the confusion matrix in Figure 5b, which highlights the relationship between the predicted labels and the ground truth. The diagonal entries of the matrix confirm the high accuracy of the system across all digit classes. Additionally, the optimized weights of the memcapacitor-based VMM, which correspond to the learnt patterns during training, are presented in **Figure 6**. These results validate the effectiveness of our architecture for the MNIST dataset, showcasing its ability to efficiently process simple image data with high accuracy.

To evaluate the scalability of the proposed architecture, we extended the system to classify the more complex CIFAR dataset.

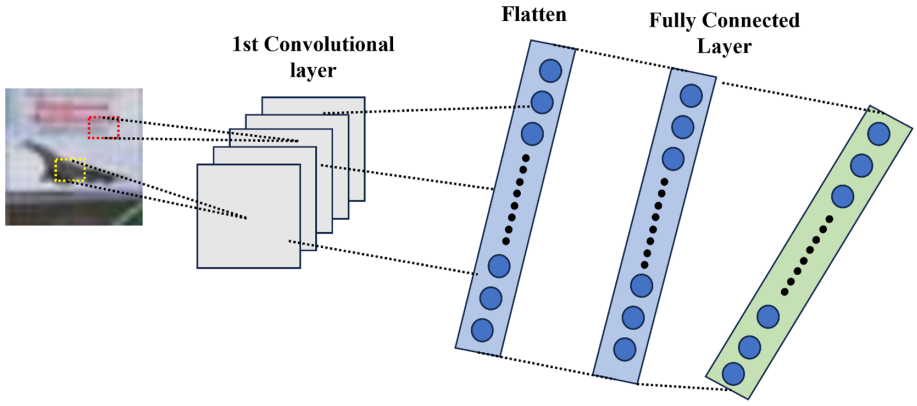

Figure 4. The design of the convolutional layer and fully connected layer, integrated with memcapacitor crossbars, tailored for processing the MNIST dataset.

Algorithm 2. Memcapacitor Based MNIST Classification.

Input:
DATASET: *MNIST dataset*
CNN model: *Define custom VMM model based on memcapacitor model*
Model Parameters: *Define activations functions, fully connected layer*
Parameters: *Training Parameters e.g. epochs, batch size*
Output:*, Trained network, accuracy metrices, plot*
Initialize *Normalize dataset and resize to 20 × 20*
Train model:
5 × 5 × 5 VMM model based on memcapacitor
Flatten layer (16 × 16 to 1D)
Fully Connected Layer of 64 neuron
ReLU activation
Fully Connected Layer of 10 neuron
Softmax activation
Evaluate model such as validation accuracy
Analyze and plot simulation results

For this task, we employed two convolutional layers using memcapacitor-based VMMs. The first layer utilized a 5 × 5 × 5 VMM, while the second layer adopted a 5 × 5 × 15 VMM configuration, as depicted in **Figure 7**. The input images were processed through these convolutional layers, where the memcapacitors efficiently extracted low-level and high-level features, such as edges, textures, and patterns. The outputs were subsequently fed into a neural network consisting of layers with ReLU activation functions, followed by a dense layer with 256 units and a final softmax output layer for classification. As shown in **Figure 8**, the proposed system achieved a training accuracy of 85.9% and a testing accuracy of 71.7%. The accuracy versus epoch curve illustrates the training progression, where the model steadily converged over successive epochs. These results highlight the ability of the memcapacitor-based architecture to handle more complex datasets, demonstrating its robustness and versatility.

4.2. Effect of the Device Variation on Classification

The capacitance of memcapacitor devices, as described in Equation (14), plays a central role in their functionality within

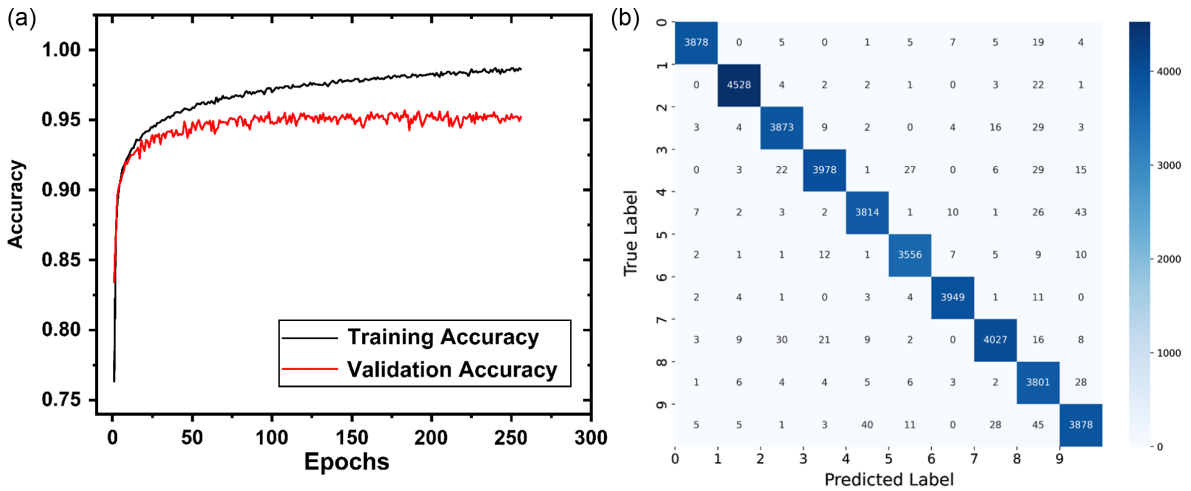

Figure 5. a) MNIST training and validation accuracy per epoch. b) Confusion matrix of model predictions against true labels of memcapacitor-based VMM.

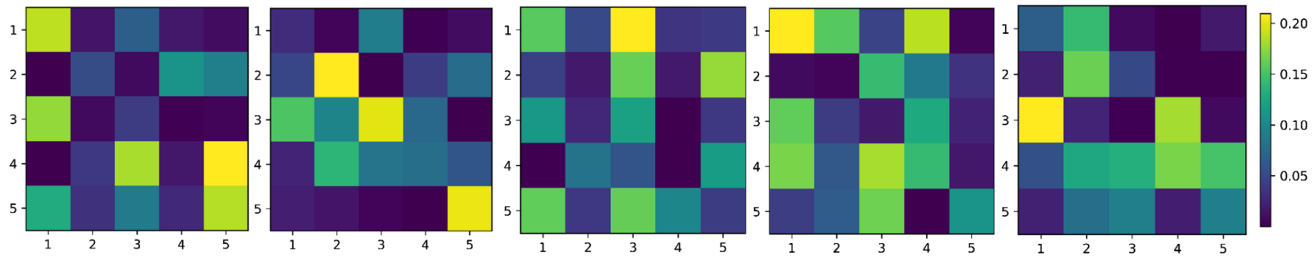

Figure 6. Capacitance maps of all five filters for MNIST classification.

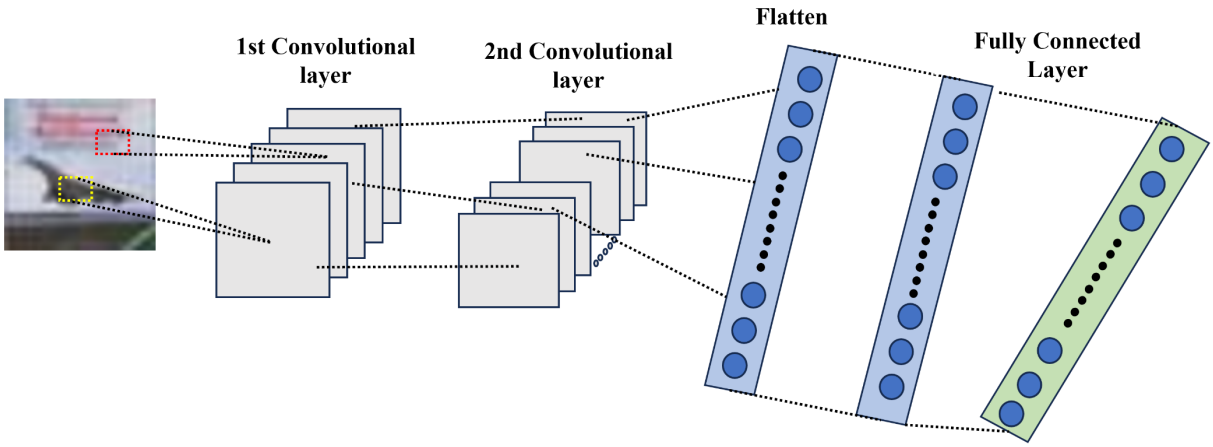

Figure 7. The design of the convolutional layer and fully connected layer, integrated with memcapacitor crossbars, tailored for processing the CIFAR dataset.

VMM operations. This capacitance is influenced by critical parameters, such as the g_{m1}, gain factor (K), and other passive components. However, during the fabrication and testing processes, these parameters are often affected by fabrication and inconsistencies, leading to deviations in the actual capacitance values. Such variations manifest in the hysteresis curve, where the maximum capacitance corresponds to the maximum number of pulses required to transition the device between its extremal conductance states (minimum and maximum). These capacitance states are essential for accurately modeling the physical behavior of memcapacitor devices, which directly impacts the simulation and operation of memcapacitor-based neuromorphic systems.

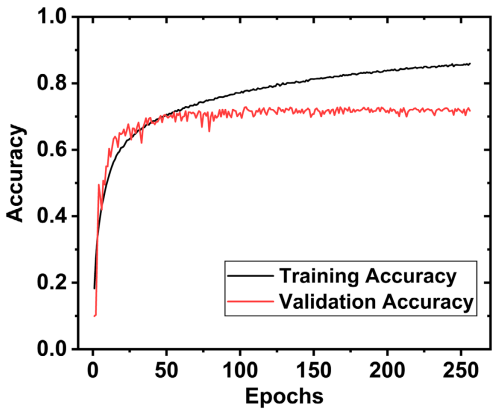

Figure 8. CIFAR training and validation accuracy per epoch of memcapacitor-based VMM.

In the field of nonvolatile memory (NVM) devices, fabrication processes inherently introduce several nonideal characteristics, including variations in capacitance, device-to-device (D2D) inconsistencies, nonlinearity, and asymmetry. These imperfections are particularly prominent in crossbar array configurations, where a large number of devices collectively execute VMM operations. D2D variations—arising from minor discrepancies during fabrication—are a significant challenge as they cause nonuniform capacitance levels across devices. These inconsistencies disrupt weight representation and updates during neural network training, leading to unpredictable network behavior. The cumulative impact of such variations across a large crossbar array amplifies these effects, undermining the precision and performance of the neural network. To evaluate the effect of these nonidealities, we examined training and validation accuracies under varying degrees of D2D capacitance variations, ranging from 5 to 30%, as illustrated in **Figure 9**a,b. The results demonstrate a clear degradation in accuracy as the variation percentage increases, highlighting the profound influence of fabrication-induced inconsistencies on neural network performance.

Addressing the challenges posed by D2D variations and nonideal characteristics is crucial for advancing memcapacitor-based neuromorphic computing technologies. Accurate mathematical models that incorporate the effects of fabrication variations

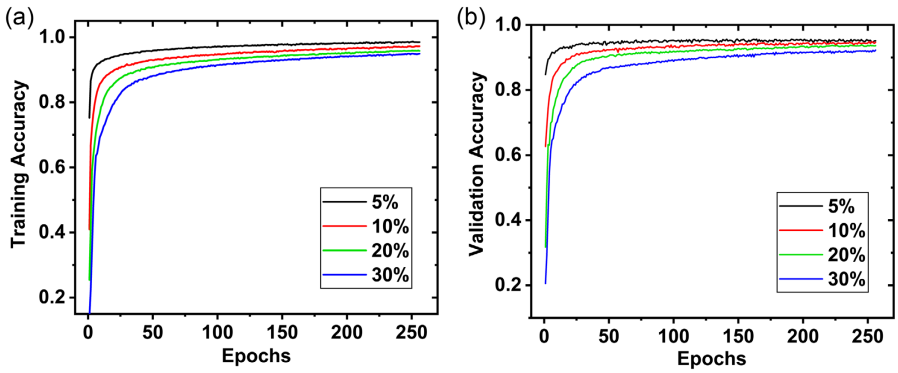

Figure 9. a) Training accuracy and b) validation accuracy per epoch with different D2D levels from 5% to 30%.

are essential for predicting real-world device behavior. Additionally, robust compensation strategies must be implemented during training to mitigate the impact of these variations on weight updates. By incorporating such strategies, the system can maintain high accuracy and reliability despite the presence of fabrication-induced inconsistencies.

4.3. Effect of the Change in Capacitance on Classification

The training and validation accuracies for MNIST classification over 256 epochs are shown in **Figure 10**a,b, illustrating the relationship between the number of epochs and the network's accuracy across three distinct capacitance regions in memcapacitor devices within crossbar arrays. These regions reflect variability in capacitance caused by factors such as D2D variations, fabrication imperfections, or programming inconsistencies. Such variability influences the capacitance ranges within the memcapacitor, which directly impacts the effectiveness of weight updates during the learning process.

The identified capacitance regions correspond to approximate conductance ranges: region 1 spans $[0.01 \times 10^{-6}, 0.7 \times 10^{-6}$ F], region 2 covers $[0.01 \times 10^{-6}, 0.2 \times 10^{-6}$ F], and region 3 is limited to $[0.01 \times 10^{-6}, 0.12 \times 10^{-6}$ F]. Among these, region 1 demonstrates the highest training and validation accuracies, which can be attributed to its broader capacitance range. This broader range allows for greater synaptic weight flexibility during training, facilitating more dynamic weight updates and improved feature representation. As a result, the network achieves better generalization and classification accuracy. In comparison, regions 2 and 3 exhibit moderate accuracies, which suggests that narrower capacitance ranges constrain the network's ability to adapt its weights effectively. The reduced flexibility in weight scaling within these regions limits the learning capacity of the neural network, resulting in lower accuracy outcomes.

This comparative analysis highlights the critical role of capacitance range and distribution in memcapacitor devices, particularly within crossbar arrays. Wider capacitance ranges enable more precise and adaptable weight updates, improving the overall performance of neural networks. Conversely, restricted capacitance variability can lead to suboptimal learning dynamics. Optimizing and maintaining consistent capacitance values across devices is, therefore, essential for enhancing the reliability and accuracy of memcapacitor-based VMM systems in neuromorphic computing applications.

5. Comparison and Discussion

This study presents a comprehensive evaluation of accuracy, device utilization, and robustness against fabrication-induced variations in memcapacitor-based neural networks. A detailed comparison with prior implementations is provided in **Table 1**, showcasing the advantages of the proposed framework in terms of performance and efficiency. The proposed design employs a compact configuration of 125 ($5 \times 5 \times 5$) NVM devices, achieving

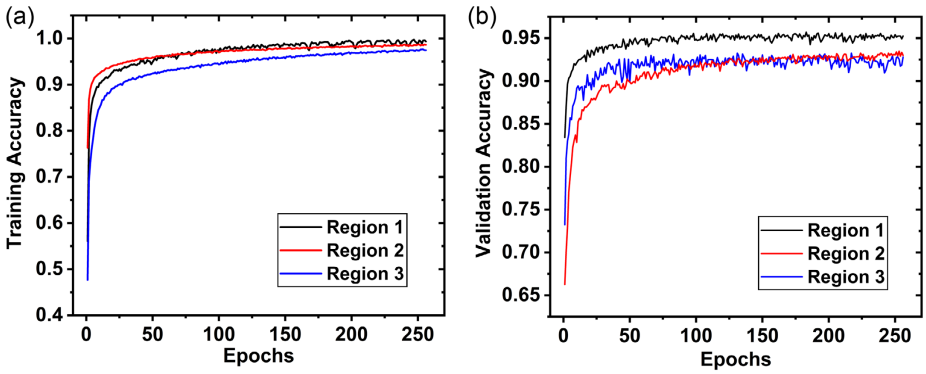

Figure 10. a) Training accuracy and b) validation accuracy per epoch across different regions with different capacitance ranges.

Table 1. Comparison of our proposed framework with earlier design We emphasized our contributions by highlighting them in bold to distinguish them from earlier designs.

References	Device material	No. of device	Device variation [%]	Power Consumption	Tasks	Recognition ratio [%]
[41]	Memristor: FTJ	20×20	3	×	MNIST	94
[42]	Memcapacitor: NAND Flash	8×16	–	×	CIFAR10	92.11
[43]	Memcapacitor: Si	128×128	30	0.71 W	CIFAR10	91.03
[44]	SRAM: CMOS	×	×	0.6 W	MNIST	90
[45]	Intel: LoiHi	×	×	14.7 mW	MNIST	96.3
[46]	Memristor: TiOx	25×25	20	140.8 mW	MNIST	96.2
[47]	Memristor: V_3O_5	28×28	–	3.6 W	MNIST	90
This work	**Memcapacitor: CMOS**	**$5 \times 5 \times 5$**	**30**	**54.6 mW**	**MNIST**	**98.4**
					CIFAR10	**85.9**

an impressive 98.4% accuracy for MNIST digit classification and 85.9% for CIFAR-10 object detection. In contrast, earlier studies, such as those utilizing larger 28×28 NVM arrays, reported lower accuracy, with only 90% recognition for handwritten digits.[41,44–47] The compactness of our design, combined with superior accuracy, highlights its efficiency and suitability for resource-constrained neuromorphic systems. Crucially, the proposed framework demonstrates robustness under real-world fabrication conditions, accounting for device parameter variations up to 30%, a significant improvement over the 3% variation considered in earlier studies.[41] This ensures the framework's reliability and stability for practical deployments, even in large-scale systems. Additionally, the use of capacitance-based devices rather than traditional resistance-based counterparts contributes to significant power savings. With a power consumption of 54.6 mW for MNIST classification, the proposed design outperforms earlier systems that consumed up to 3.6 W.[47] This optimization is especially critical for energy-efficient neuromorphic computing applications.

6. Conclusion

In this research, we have successfully demonstrated the potential of a memcapacitor-based neural network framework for data-intensive tasks such as digit classification and image recognition. By harnessing the unique charge-storage characteristics of memcapacitors, we designed and implemented a CMOS-based circuit capable of efficiently performing VMM while emulating synaptic behavior. The system was validated using two benchmark datasets—MNIST and CIFAR—achieving training accuracies of 98.4% and 85.9%, respectively, which underscores its effectiveness in neuromorphic computing applications. A significant contribution of this study is the investigation of D2D variability and capacitance changes arising from fabrication imperfections. The results demonstrate that wider capacitance ranges enable more dynamic synaptic weight adaptation, leading to improved training and validation accuracies, as observed in region 1 of the capacitance analysis. In contrast, restricted capacitance ranges in regions 2 and 3 constrain the learning capability, resulting in moderate accuracy levels. These findings highlight the critical impact of capacitance variability on the performance and robustness of memcapacitor-based crossbar arrays.

Overall, this study establishes the memcapacitor-based VMM system as a promising approach for energy-efficient, high-performance computing in AI hardware. Future work should explore the scalability of this technology in hardware, integrated with emerging devices, and its application in more complex neural network architectures. By addressing device variations, memcapacitor-based systems have the potential to revolutionize low-power, high-accuracy neuromorphic computing, paving the way for sustainable and efficient AI-driven technologies.

Acknowledgements

This research was supported by the Nanomaterial Technology Development Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science and ICT under grant no. (NRF-2022M3H4A1A01009658).

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

Ankur Singh: conceptualization (lead); data curation (lead); formal analysis (lead); investigation (lead); methodology (lead); resources (lead); software (lead); supervision (lead); validation (lead); visualization (lead); writing—original draft (lead), **Dowon Kim:** conceptualization (equal); data curation (equal); formal analysis (equal); methodology (equal); validation (equal); visualization (equal); writing—original draft (equal), and **Byung-Geun Lee**: data curation (equal); formal analysis (equal); funding acquisition (lead); investigation (lead); methodology (equal); project administration (lead); resources (equal); software (equal); supervision (lead); validation (equal); visualization (equal); writing—original draft (equal); writing—review & editing (equal).

Data Availability Statement

Research data are not shared.

Keywords

In-memory computing, memcapacitor, memristor, neural network, temporal data

Received: September 14, 2024
Revised: December 20, 2024
Published online: March 31, 2025

[1] M. Di Ventra, Y. V. Pershin, L. O. Chua, *Proc. IEEE* **2009**, *97*, 1717.
[2] K.-U. Demasius, A. Kirschen, S. Parkin, *Nat. Electron.* **2021**, *4*, 748.
[3] J. Jang, S. Gi, I. Yeo, S. Choi, S. Jang, S. Ham, B. Lee, G. Wang, *Adv. Sci.* **2022**, *9*, 2201117.
[4] C. Li, D. Belkin, Y. Li, P. Yan, M. Hu, N. Ge, H. Jiang, E. Montgomery, P. Lin, Z. Wang, W. Song, J. P. Strachan, M. Barnell, Q. Wu, R. S. Williams, J. J. Yang, Q. Xia, *Nat. Commun.* **2018**, *9*, 2385.
[5] K. Roy, A. Jaiswal, P. Panda, *Nature* **2019**, *575*, 607.
[6] P. Yao, H. Wu, B. Gao, J. Tang, Q. Zhang, W. Zhang, J. J. Yang, H. Qian, *Nature* **2020**, *577*, 641.
[7] C. Mead, *Proc. IEEE Inst. Electr. Electron. Eng.* **1990**, *78*, 1629.
[8] C. Mead, *Nat. Electron.* **2020**, *3*, 434.
[9] D. B. Strukov, G. S. Snider, D. R. Stewart, R. S. Williams, *Nature* **2008**, *453*, 80.
[10] A. Krizhevsky, I. Sutskever, G. E. Hinton, *Commun. ACM* **2017**, *60*, 84.
[11] Y. Lecun, L. Bottou, Y. Bengio, P. Haffner, *Proc. IEEE Inst. Electr. Electron. Eng.* **1998**, *86*, 2278.
[12] F. M. Bayat, M. Prezioso, B. Chakrabarti, H. Nili, I. Kataeva, D. Strukov, *Nat. Commun.* **2018**, *9*, 1.
[13] F. Cai, J. M. Correll, S. H. Lee, Y. Lim, V. Bothra, Z. Zhang, M. P. Flynn, W. D. Lu,*Nat. Electron.* **2019**, *2*, 290.
[14] M. Prezioso, F. Merrikh-Bayat, B. D. Hoskins, G. C. Adam, K. K. Likharev, D. B. Strukov, *Nature* **2015**, *521*, 61.
[15] L. Hu, J. Yang, J. Wang, P. Cheng, L. O. Chua, F. Zhuge, *Adv. Funct. Mater.* **2021**, *31*, 2005582.
[16] X. Shan, Z. Wang, J. Xie, J. Han, Y. Tao, Y. Lin, X. Zhao, D. Lelmini, Y. Liu, H. Xu,, *Adv. Sci.* **2024**, *11*, 2405160.
[17] W. Geoffrey, B, R. M. Shelby, S. Sidler, C. di Nolfo, J. Jang, I. Boybat, R. S. Shenoy, P. Narayanan, K. Virwani, E. U. Giacometti, B. N. Kurdi, H. Hwang, *IEEE Trans. Electron Devices* **2015**, *62*, 3498.
[18] W. A. Borders, H. Akima, S. Fukami, S. Moriya, S. Kurihara, Y. Horio, S. Sato, H. Ohno, *Appl. Phys. Express* **2017**, *10*, 013007.
[19] J. Grollier, D. Querlioz, K. Y. Camsari, K. Everschor-Sitte, S. Fukami, M. D. Stiles, *Nat. Electron.* **2020**, *3*, 360.
[20] V. Garcia, M. Bibes, *Nat. Commun.* **2014**, *5*, 1.
[21] W. A. Borders, H. Akima, S. Fukami, S. Moriya, S. Kurihara, Y. Horio, S. Sato, H. Ohno, in *2017 IEEE Int. Electron Devices Meeting (IEDM)*, San Francisco, CA, USA, 6 December **2017**, pp. 6.2.1–6.2.4.
[22] H. Mulaosmanovic, J. Ocker, S. Müller, M. Noack, J. Müller, P. Polakowski, T. Mikolajick, S. Slesazeck, in *2017 Symp. on VLSI Technology*, Kyoto, Japan, 8 June **2017**, pp. T176–T177.
[23] V. Agrawal, V. Prabhakar, K. Ramkumar, L. Hinh, S. Saha, S. Samanta, R. Kapre, in *2020 IEEE Int. Memory Workshop (IMW)*, Dresden, Germany, 20 May **2020**.
[24] H. Tsai, S. Ambrogio, P. Narayanan, R. M. Shelby, G. W. Burr, *J. Phys. D Appl. Phys.* **2018**, *51*, 283001.
[25] M. Di Ventra, Y. V. Pershin, L. O. Chua, *Proc. IEEE Inst. Electr. Electron. Eng.* **2009**, *97*, 1717.
[26] J. Martinez-Rincon, M. Di Ventra, Y. V. Pershin, *Phys. Rev. B Condens. Matter Mater. Phys.* **2010**, *81*, 195430.
[27] M. G. A. Mohamed, H. Kim, T.-W. Cho, *Sci. World J.* **2015**, *910126*, 16.
[28] Y. V. Pershin, M. Di Ventra, *Electron. Lett.* **2014**, *50*, 141.
[29] A. K. Khan, B. H. Lee, *AIP Adv.* **2016**, *6*, 095022.
[30] Z. Wang, M. Rao, J.-W. Han, J. Zhang, P. Lin, Y. Li, C. Li, W. Song, S. Asapu, R. Midya, Y. Zhuo, H. Jiang, J. H. Yoon, N. K. Upadhyay, S. Joshi, M. Hu, J. P. Strachan, M. Barnell, Q. Wu, H. Wu, Q. Qiu, R. S. Williams, Q. Xia, J. J. Yang, *Nat. Commun.* **2018**, *9*, 1.
[31] D. Kwon, I.-Y. Chung, *IEEE Electron Device Lett.* **2020**, *41*, 493.
[32] T. You, L.P. Selvaraj, H. Zeng, W. Luo, N. Du, D. Bürger, I. Skorupa, S. Prucnal, A. Lawerenz, T. Mikolajick, O. G. Schmidt, H. Schmidt, *Adv. Electron. Mater.* **2016**, *2*, 1500352.
[33] Q. Zheng, Z. Wang, N. Gong, Z. Yu, C. Chen, Y. Cai, Q. Huang, H. Jiang, Q. Xia, R. Huang, *IEEE Electron Device Lett.* **2019**, *40*, 1309.
[34] A. A. M. Emara, M. M. Aboudina, H. A. H. Fahmy, *Microelectronics* **2017**, *64*, 39.
[35] A. Singh(Preprint) arXiv:2403.03002 Submitted: March **2024**.
[36] A. Singh, S. S. Borah, M. Ghosh, in *2021 IEEE Int. Midwest Symp. on Circuits and Systems (MWSCAS)*, Lansing, MI, USA, 11 August **2021**, pp. 1108–1111.
[37] M. Ghosh, A. Singh, S. S. Borah, J. Vista, A. Ranjan, S. Kumar, *IEEE Trans. Electron Devices* **2022**, *69*, 2248.
[38] F. Bizzarri, A. Brambilla, G. Storti Gajani, *Simul. Model. Pract. Theory* **2018**, *81*, 51.
[39] İ. Bayraktaroğlu, A. Selçuk Öğrenci, G. Dündar, S. Balkır, E. Alpaydın, *Neural Netw.* **1999**, *12*, 325.
[40] A. Singh, S. Choi, G. Wang, M. Daimari, B.-G. Lee, *Neural Networks* **2025**, *182*, 106925.
[41] R. Athle, M. Borg, *Adv. Intell. Syst.* **2024**, *6*, 2300554.
[42] S. Hwang, J. Yu, M. S. Song, H. Hwang, H. Kim, *Adv. Sci.* **2023**, *10*, 2303817.
[43] A. Singh, B.-G. Lee, *IEEE Access* **2023**, *11*, 112590.
[44] K. Ando, K. Ueyoshi, K. Orimo, H. Yonekawa, S. Sato, H. Nakahara, S. Takamaeda-Yamazaki, M. Ikebe, T. Asai, T. Kuroda, M. Motomura, *IEEE J. Solid-State Circuits* **2018**, *53*, 983.
[45] A. Renner, F. Sheldon, A. Zlotnik, L. Tao, A. Sornborger, *Nat. Commun.* **2024**, *15*, 1.
[46] S.-G. Gi, H. Lee, J. Jang, B.-G. Lee, *IEEE Trans. Ind. Electron.* **2023**, *70*, 6442.
[47] S. K. Das, S. K. Nandi, C.V. Marquez, A. Rúa, M. Uenuma, S. K. Nath, S. Zhang, C.-H. Lin, D. Chu, T. Ratcliff, R. G. Elliman, *Adv. Intell. Syst.* **2024**, *6*, 2400191.