Journal of Electronics and Electrical Engineering
https://ojs.wiserpub.com/index.php/JEEE

Research Article

A Schematic-Level Digital Near-Memory Computing Architecture Utilizing 1T3R Memristor Arrays and Pipelined MAC for Pattern Recognition

Zeyuan Hou, Xiaomeng Wang, Yang Yi*

Bradley Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA, 24061, USA
E-mail: yangyi8@vt.edu

Received: 4 May 2026; **Revised:** 3 July 2026; **Accepted:** 8 July 2026; **Published:** 19 August 2026

Abstract: The traditional von Neumann architecture faces a severe memory wall bottleneck in modern data-intensive artificial intelligence applications. Memristor-based Computing-in-Memory (CIM) is a promising approach to reduce data movement, but conventional analog CIM remains vulnerable to read noise, sneak path currents, device variation, and the area and power overhead of high-resolution Analog-to-Digital Converters (ADCs). This work presents a schematic-level digital near-memory computing architecture based on a 1-Transistor-3-Resistor (1T3R) bit-slicing scheme for 3-bit signed weight storage. Each weight is mapped to three binary memristor states using two's complement representation, and the sensed digital outputs are processed by a low-bit pipelined Multiply-Accumulate (MAC) and Rectified Linear Unit (ReLU) backend. This design avoids high-resolution analog current readout and shifts the main computation to compact digital logic. The memristor array, write-inhibit and read control scheme, and digital processing blocks are implemented and functionally verified at schematic-level in Cadence Virtuoso using the open-source SkyWater 130 nm Process Design Kit (PDK) at a nominal 1.8 V supply. For system evaluation, a Python simulation framework is developed to incorporate the same low-bit quantization, signed weight mapping, finite bit digital arithmetic, and modeled device nonidealities. Hardware-aware simulations on Modified National Institute of Standards and Technology (MNIST) indicate that the proposed architecture can maintain functional classification capability under 3-bit weight and 2-bit input constraints. The reported macro power, performance, and area values are first-order projections from schematic simulations and scaling assumptions. Layout implementation, parasitic extraction, and post-layout validation remain future work.

Keywords: computing-in-memory, memristor, 1-Transistor-3-Resistor (1T3R), quantization-aware training, edge Artificial Intelligence (AI)

1. Introduction

Artificial Intelligence (AI) and Deep Neural Networks (DNNs) are now widely used in computer vision and pattern recognition. Their inference workloads rely heavily on repeated Multiply-Accumulate (MAC) operations and frequent access to stored weights. In conventional von Neumann systems, the physical separation between memory and compute units creates substantial data movement, which increases latency and energy consumption and contributes to the well-known memory wall problem [1], [2].

DOI: https://doi.org/10.37256/jeee.52202610117

Computing-in-Memory (CIM) has therefore been explored as a way to perform computation closer to the stored data and reduce memory access overhead. Memristor-based crossbars are a promising platform for this direction because they can combine dense nonvolatile storage with local computation. At the same time, recent reviews have emphasized that large scale deployment of memristor-based neural hardware is still limited by practical device and circuit nonidealities [3].

Among emerging nonvolatile memories, memristors, also known as Resistive Random Access Memory (RRAM) devices, are attractive candidates for CIM because of their high integration density, nonvolatility, low power operation, and compatibility with Complementary Metal-Oxide-Semiconductor (CMOS) processes. Prior studies have demonstrated monolithic integration of memristive synapses and volatile neurons with spatiotemporal dynamics and gain modulation [4]. Other works have reported integrated memristor system-on-chip for edge learning tasks through hardware and algorithm co-design [5], [6].

Despite this progress, analog memristor CIM remains challenging for deterministic DNN inference. Analog vector-matrix multiplication based on Kirchhoff's Current Law can provide high parallelism, but the computed result can be affected by parasitic voltage drops, sneak path currents, resistance variation, and sensing noise [1]. In addition, converting analog currents back to the digital domain often requires Analog-to-Digital Converters (ADCs), which can introduce considerable area and power overhead.

These limitations have motivated interest in digital CIM and digital near-memory processing. Recent CMOS digital CIM macros, such as the 4 nm design reported by Taiwan Semiconductor Manufacturing Company (TSMC) [7], show that digital processing can provide accuracy and scalability advantages in advanced technologies. This work follows a related design direction but applies it to binary memristor arrays, where memristors are used as low-bit digital storage elements and the main arithmetic is performed by CMOS logic after sensing. This approach is intended to reduce dependence on analog accumulation and high-precision data conversion.

To address these issues, this paper presents a schematic-level digital near-memory computing architecture that combines binary 1-Transistor-3-Resistor (1T3R) memristor storage with low-bit CMOS digital processing. The current validation includes schematic-level circuit simulations in Cadence Virtuoso and Python-based hardware-aware system evaluation. Layout implementation, parasitic extraction, and post-layout verification are not included in this work and remain future tasks. The main contributions of this work are as follows:

1. A 1T3R bit-slicing scheme is used to represent 3-bit signed weights in two's complement form with three binary memristor states. This mapping supports signed weight storage while avoiding high-resolution analog-to-digital conversion.

2. A write-inhibit array control scheme is implemented and evaluated at schematic-level. The scheme reduces sneak path effects and protects unselected cells from write disturbance during programming.

3. A pipelined digital MAC and Rectified Linear Unit (ReLU) backend is constructed using sky130 CMOS logic to process the digitized outputs from the memristor array with finite-bit digital arithmetic.

4. A Python simulation framework is developed to evaluate the architecture under hardware constraints, including quantization, signed weight mapping, finite-bit arithmetic, and modeled device nonidealities.

2. Background and motivation

2.1 *Memristor fundamentals*

A memristor is a two-terminal nanoscale device exhibiting hysteretic Voltage-Current (V-I) characteristics, typically composed of a metal-oxide dielectric switching layer (e.g., HfO_x or TiO_x) sandwiched between two electrodes (Figure 1).

The resistance switching relies on the stochastic growth and rupture of internal Conductive Filaments (CF). Applying a positive set Voltage (V_{SET}) forms the CF, transitioning the device to a Low Resistance State (LRS, logic '1'), while a reverse reset Voltage (V_{RESET}) ruptures the CF, restoring it to a High Resistance State (HRS, logic '0'). The resulting hysteretic V-I loop is shown in Figure 2, with the proposed memristor array and its peripheral circuitry are shown in Figure 3.

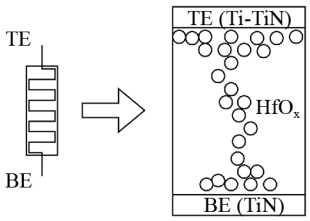

Figure 1. Physical structure of a metal-oxide memristor. The active switching layer (e.g., HfO_x) is sandwiched between two metal electrodes; resistance switching is governed by the formation and rupture of a Conductive Filament (CF)

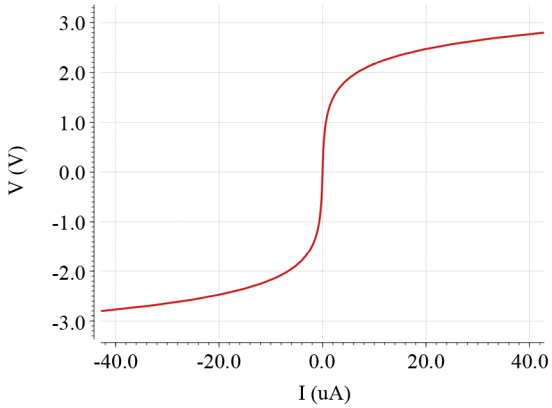

Figure 2. Hysteretic V-I characteristic of a memristor. Under SET/RESET voltages the device toggles between Low Resistance State (LRS) and High Resistance State (HRS), exhibiting the non-volatile bistable behavior exploited by the proposed digital-CIM scheme

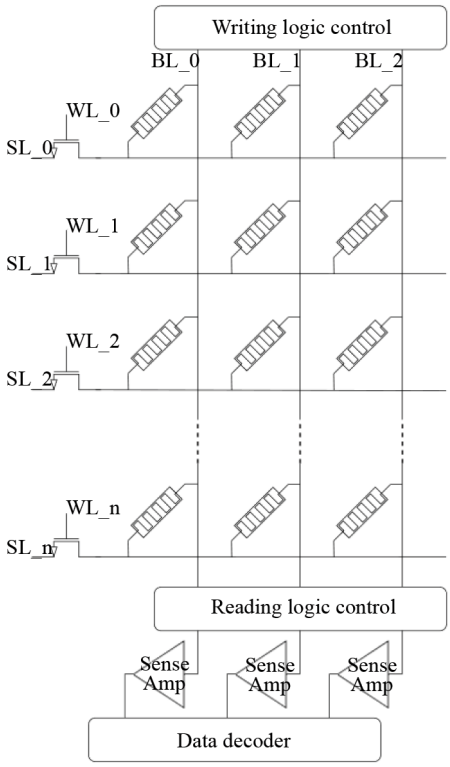

Figure 3. Schematic of the 1T3R memristor array and its write/read circuit

2.2 *Challenges in analog computing-in-memory*

Traditional memristive CIM architectures rely on Kirchhoff's Current Law (KCL) to perform analog Vector-Matrix Multiplication (VMM). While theoretically energy-efficient, analog CIM faces several practical physical bottlenecks:

• Precision Degradation: Sneak path currents [8], IR drops along parasitic wire resistances, and device-to-device conductance variations can corrupt the analog summation.

• ADC Overhead: Converting high-precision analog currents back to the digital domain requires area- and power-intensive ADCs, which can account for a large portion of the total macro area.

• Reliability: Multi-Level Cell (MLC) programming is highly susceptible to resistance drift, making it difficult to maintain deterministic weights over time.

2.3 *The digital-CIM paradigm*

To mitigate these analog nonidealities, the digital-CIM paradigm has emerged as a promising alternative. By treating memristor cells as binary storage elements and performing bit-wise operations in the digital domain, digital-CIM removes the need for high-resolution ADCs and reduces sensitivity to small conductance fluctuations. This work applies the digital design philosophy to a 1T3R bit-slicing architecture, connecting algorithm-level signed weights with device-level binary operation.

3. Proposed 1T3R digital-CIM architecture

3.1 *System overview*

The proposed architecture adopts a modular dataflow to improve throughput, as summarized in Figure 4. It consists of a digital input decoder, a 1T3R memristive array, and a pipelined digital backend. Unlike analog CIM that accumulates currents on a bitline, this design digitizes the outputs after the sense amplifiers, so the following MAC operations are performed in the digital domain and are less sensitive to analog read noise.

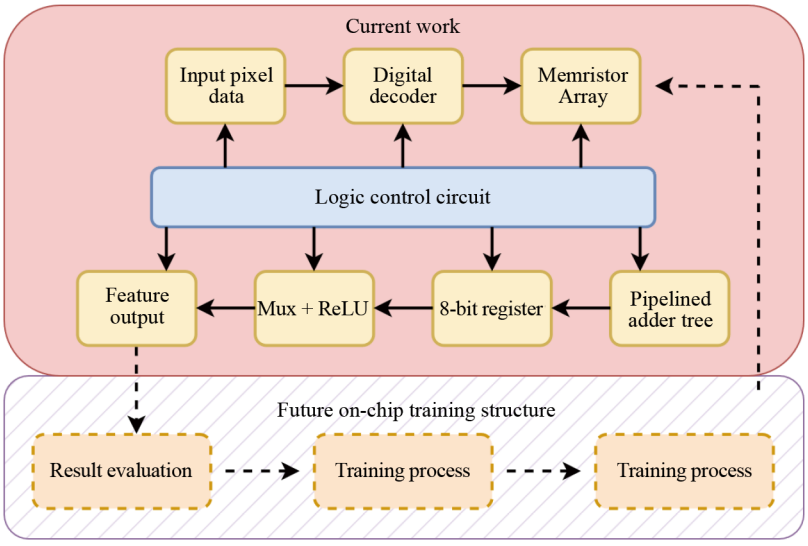

Figure 4. Top-level dataflow of the proposed digital-near-memory architecture: digital input decoder, 1T3R memristive array with sense amplifiers, and a pipelined digital MAC and ReLU backend

3.2 *3-bit spatial bit-slicing and signed representation*

To implement the signed weights required by neural network algorithms, we propose a 1-Transistor-3-Resistor (1T3R) spatial bit-slicing scheme. A 3-bit signed logical Weight (W_{logic}) is mapped onto three independent binary memristors (G_{b2}, G_{b1}, G_{b0}) according to two's complement logic:

$$W_{phy} = -4G_2 + 2G_1 + G_0 \tag{1}$$

where G_2 serves as the sign bit. By assigning a negative weight to G_2, the architecture can execute signed MAC operations within a unified digital adder tree. This choice avoids separate subtraction logic or sign-magnitude handling, providing a consistent numerical foundation for Quantization-Aware Training (QAT) (Table 1). The complete bit-to-cell mapping is illustrated in Figure 5.

Table 1. Mapping logic between 3-bit logical weights and 1T3R physical states

Weight (W_{logic})	2's C	Memristor state	Physical mapping W_{phy}
+ 3	011	HRS—LRS—LRS	$0(-4)+1(2)+1(1)=3$
+ 2	010	HRS—LRS—HRS	$0(-4)+1(2)+0(1)=2$
+ 1	001	HRS—HRS—LRS	$0(-4)+0(2)+1(1)=1$
0	000	HRS—HRS—HRS	$0(-4)+0(2)+0(1)=0$
− 1	111	LRS—LRS—LRS	$1(-4)+1(2)+1(1)=-1$
− 2	110	LRS—LRS—HRS	$1(-4)+1(2)+0(1)=-2$
− 3	101	LRS—HRS—LRS	$1(-4)+0(2)+1(1)=-3$
− 4	100	LRS—HRS—HRS	$1(-4)+0(2)+0(1)=-4$

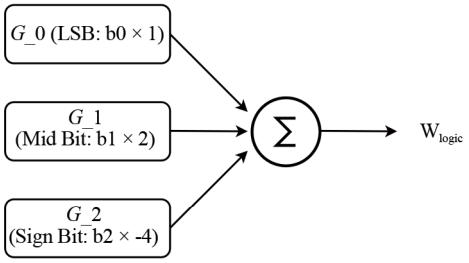

Figure 5. 3-bit spatial bit-slicing scheme. A signed weight W_{logic} is decomposed into three binary memristors G_{b2}, G_{b1}, G_{b0} following two's-complement encoding, with G_{b2} acting as the sign bit

3.3 *Array control and disturbance mitigation*

To verify the 1T3R logic, a 6×3 test array was implemented using the sky130 PDK. To isolate sneak paths and reduce write disturbance during programming, a write-inhibit biasing scheme is employed. During the SET phase, selected wordlines are driven to 1.7 V while unselected sourcelines are pulled to an inhibit voltage of 1.2 V. This keeps the voltage differential across unselected cells below the switching threshold and reduces the risk of accidental state transitions. The resulting bimodal resistance population across programmed cells is plotted in Figure 6.

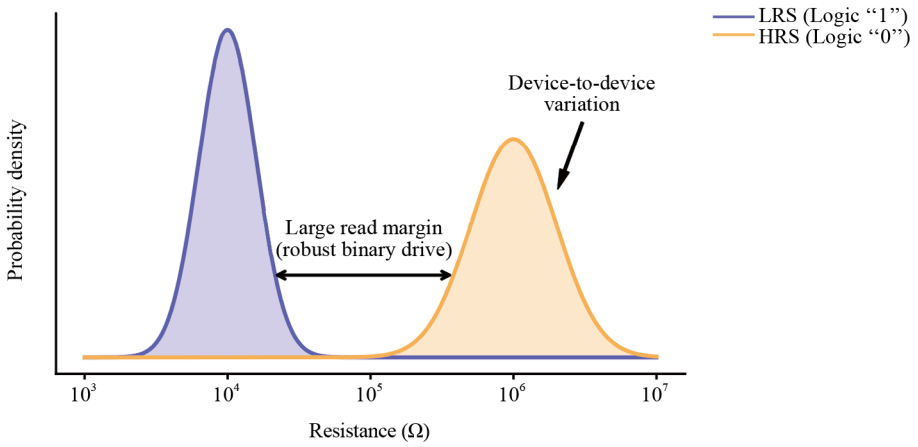

Figure 6. Resistance state distribution of the 1T3R memristor array. The bimodal populations correspond to LRS, logic 1, and HRS, logic 0, supporting reliable two level programming

3.4 *Pipelined MAC and digital ReLU engine*

The digitized outputs from the array are processed by a pipelined MAC engine. At the schematic level, an 8-bit accumulator is implemented and verified as a representative accumulation block for partial-product processing. In the Python hardware-aware system evaluation, the 64-product MAC operation is partitioned across multiple 8-bit accumulator blocks rather than being accumulated in a single 8-bit register. This block-level implementation is used to verify the digital accumulation and ReLU datapath under the low-bit operating conditions of the proposed architecture.

Furthermore, a digital ReLU activation is implemented without additional comparators. By using the two's complement representation, the most significant bit, A_{acc} [7], indicates the sign of the result. A simple multiplexer, controlled by the Most Significant Bit (MSB), truncates negative values to zero:

$$A_{out} = \begin{cases} 0, & A_{acc}[7] = 1 \\ A_{acc}, & A_{acc}[7] = 0 \end{cases} \tag{2}$$

This architecture shows a schematic-level combinational delay of 3.78 ns in the evaluated digital datapath, supporting operation at the target frequency of 100 MHz under the current simulation assumptions.

4. Hardware-aware simulation and results

4.1 *Quantization-aware training model*

To evaluate the system-level behavior of the hardware architecture, we constructed a QAT model that reflects the main physical constraints rather than relying only on a floating-point model. In PyTorch, a custom Straight-Through Estimator (STE) is used to approximate gradients through discrete quantization during backpropagation. To further connect algorithmic QAT and physical deployment, recent studies have emphasized the need to include realistic SPICE-level conductance errors and cycle-to-cycle variations in the simulation loop [9]. Following this direction, our model includes STE-based quantization and modeled analog instability. In addition, recent comparative studies on memristor crossbars show that the fraction of cells in the LRS strongly affects inference power at the edge [10]. Motivated by this constraint, the QAT framework introduces L2 regularization to penalize LRS formation, which can reduce static array leakage and dynamic power consumption.

In the Python hardware-aware model, a finite-bit activation approximation is applied to reflect the limited precision of the digital backend. Specifically, the accumulated hidden-layer output is shifted and clamped as A_{sim} = clip(floor(A_{acc}/4), −4, 3), where clip(., −4, 3) limits the activation to the 3-bit signed range used in the low-bit simulation.

This shift-and-clamp operation is used only in the Python hardware-aware simulation to model finite-bit digital arithmetic effects and is distinct from the schematic-level ReLU implementation. The schematic ReLU datapath uses the mux-based sign truncation described in Eq. (2), which sets negative accumulated values to zero.

4.2 *Performance evaluation*

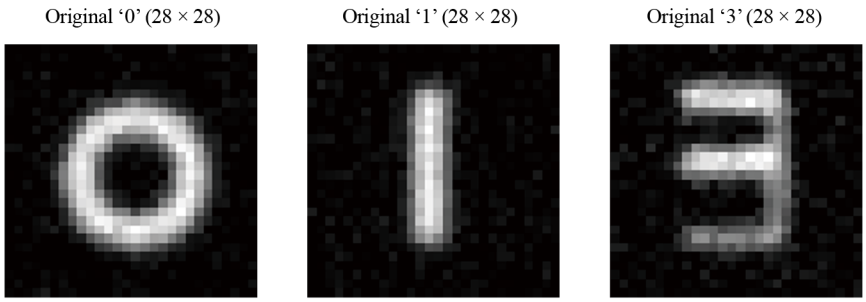

Figure 7. Original 28 × 28 grayscale MNIST input

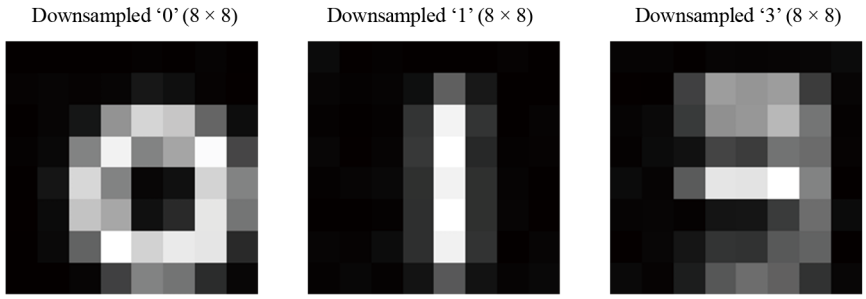

Figure 8. Down-sampled 8 × 8 representation used to match the macro's input dimension

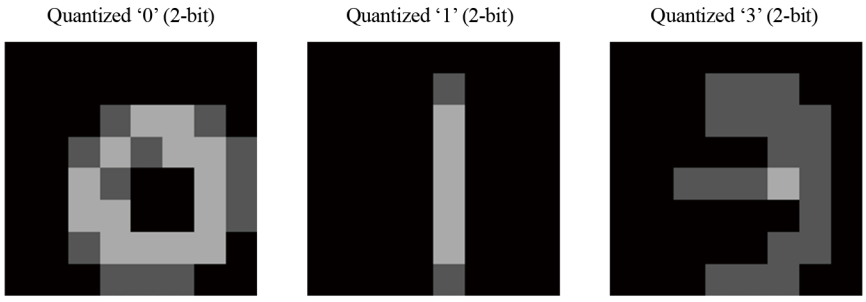

Figure 9. Inputs after 2-bit linear quantization (values ∈ {0, 1, 2, 3}), corresponding to four discrete read voltage magnitudes applied to the array

We evaluated the architecture on the Modified National Institute of Standards and Technology (MNIST) dataset, which consists of grayscale images of handwritten digits categorized into 10 classes (0-9). Taking a minimalist two-layer

Multi-Layer Perceptron (MLP) of 64 × 32 to 32 × 10 as a case study, the original 28 × 28 images (Figure 7) are down-sampled to an 8 × 8 (64 pixels) size (Figure 8) to fit the macro dimensions. Furthermore, the original 8-bit grayscale pixel intensities (0-255) are linearly quantized into 2-bit unsigned values (0, 1, 2, 3), as illustrated in Figure 9. This non-negative unsigned input naturally corresponds to the four non-negative discrete read voltage magnitudes without requiring a sign bit. The corresponding numerical 8 × 8 input matrices after 2-bit quantization are shown in Figure 10.

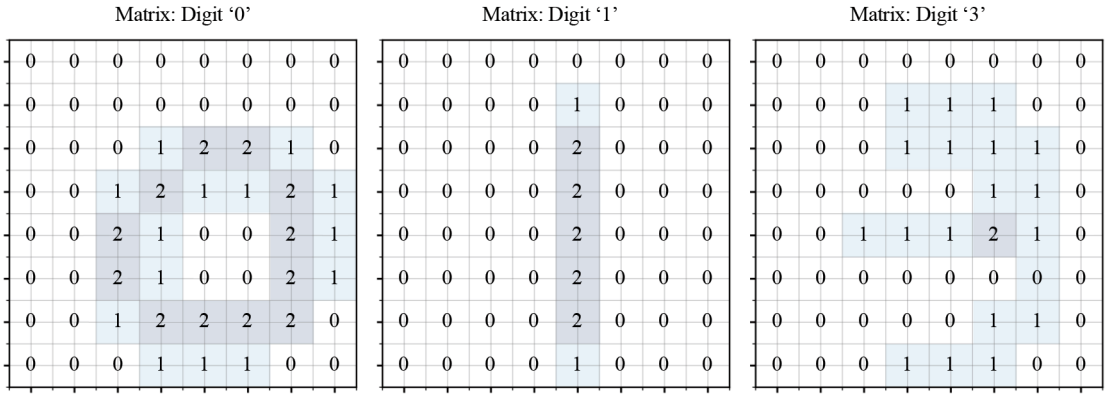

Figure 10. Numerical 8 × 8 input matrices after 2-bit quantization for representative MNIST digits. Each entry belongs to {0, 1, 2, 3} and corresponds to the discrete input level applied to the digital near memory evaluation flow

Table 2. Ablation study of hardware constraints on the 8 × 8 MNIST task

Model	Input/Weight	Hardware constraints	Sparsity reg.	Test accuracy	Zero physical weights
FP32 software baseline	FP32/FP32	No quantization or hardware activation	No	95.37%	0.00%
Input quantized baseline	2-bit unsigned/FP32	Input quantization only	No	92.16%	0.00%
Weight quantized baseline	2-bit unsigned/3-bit signed	Input and weight quantization	No	81.28%	44.93%
Hardware activation baseline	2-bit unsigned/3-bit signed	Quantization plus shift and clamp activation	No	75.96%	50.00%
Proposed hardware-aware model	2-bit unsigned/3-bit signed	Quantization plus shift and clamp activation	Yes	79.70%	78.08%

Table 2 summarizes the ablation study using the same 8 × 8 MNIST input representation and the same 64 to 32 to 10 MLP topology. The FP32 software baseline reaches 95.37% test accuracy without any hardware constraints. Applying only 2-bit unsigned input quantization reduces the accuracy to 92.16%, indicating that low-precision input encoding introduces a moderate accuracy loss. Adding the 3-bit signed weight quantization associated with the 1T3R two's complement mapping further reduces the accuracy to 81.28%, showing that aggressive weight quantization is the dominant source of degradation. When the hardware style shift and clamp activation is included, the accuracy becomes 75.96%, reflecting the additional cost of finite-bit digital activation. With sparsity regularization enabled, the proposed hardware-aware model reaches 79.70% test accuracy while increasing the zero physical weight ratio to 78.08%. This result suggests that sparsity regularization improves the hardware friendly weight distribution and also recovers part of the classification accuracy.

Overall, the ablation study, as summarized in Figure 11, shows that the reported classification performance should be interpreted as a hardware-constrained tradeoff rather than as an isolated software accuracy result.

As an additional capacity sensitivity check, the number of hidden neurons was swept from 16 to 128 under the same 2-bit input quantization, 3-bit signed weight quantization, hardware style activation, no bias setting, and sparsity regularization. In this fixed seed sweep, the test accuracy remains around 76% for 16, 32, and 64 hidden neurons, while the 128 hidden neuron model reaches 80.12%. The overall zero physical weight ratio also increases from 62.92% to 91.75%. These results suggest that increasing model capacity can partially recover accuracy under strict low-bit hardware constraints, while sparsity regularization continues to push a large fraction of physical weights to the zero state.

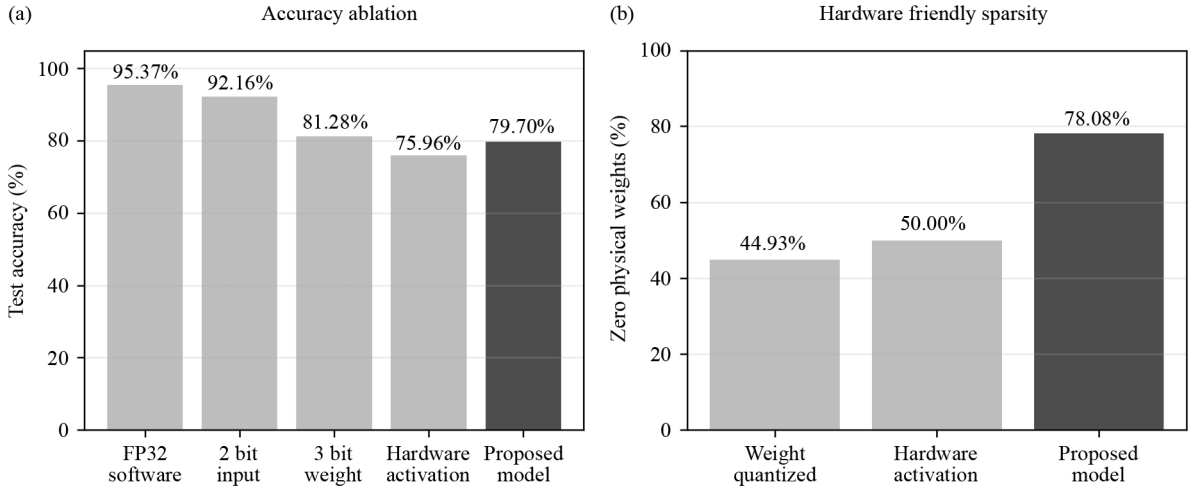

Figure 11. Hardware-aware ablation results on the 8 × 8 MNIST task. (a) Test accuracy decreases as low-bit input quantization, 3-bit signed weight quantization, and hardware style activation are introduced, while sparsity regularization partly recovers accuracy; (b) The zero physical weight ratio increases to 78.08% in the proposed hardware-aware model

Table 3. Bit-level robustness sensitivity under 1T3R physical weight mapping

Fault/Error rate	Read bit flip accuracy drop (pp)	Mixed stuck at fault accuracy drop (pp)
0.0%	0.00 pp	0.00 pp
0.1%	1.28 ± 1.70 pp	0.69 ± 0.99 pp
0.5%	13.76 ± 9.31 pp	4.49 ± 2.54 pp
1.0%	38.55 ± 8.58 pp	15.60 ± 4.40 pp
2.0%	39.61 ± 6.23 pp	19.05 ± 11.06 pp
5.0%	61.62 ± 5.93 pp	50.27 ± 5.40 pp

To further assess reliability under physical nonidealities, a bit level robustness sensitivity analysis was performed using the same proposed hardware-aware setting. The primary clean accuracy reported for the proposed model remains 79.70% in Table 2. Because the robustness sweep was conducted as an independent stochastic training run, Table 3 reports accuracy drops measured relative to the clean reference within that robustness run. These drop values are used to show degradation trends and do not replace the primary 79.70% accuracy reported in Table 2. In this analysis, the trained 3-bit weights were decomposed into the 1T3R binary memristor planes G_2, G_1, and G_0. Read bit flips and mixed stuck-at faults were then injected into the physical bit planes before reconstructing the signed weight as $W_{\text{phy}} = -4G_2 + 2G_1 + G_0$. At a 0.1% error or fault rate, the mean degradation is small: 1.28 percentage points for read bit flips and 0.69 percentage points for mixed stuck-at faults. At a 0.5% rate, read bit flips become more damaging, producing a 13.76 point drop, while mixed stuck-at faults produce a 4.49 point drop. The difference occurs because read bit flips invert the sensed bit during

inference, whereas mixed stuck-at faults affect only cells whose stuck value differs from the intended stored state. At 1% and above, both models degrade noticeably, indicating that the current 1T3R mapping can tolerate very low physical bit error rates but remains sensitive when the physical bit error or fault rate increases from the sub-percent range toward the percent range. These results motivate future work on read margin improvement, fault-aware remapping, redundancy, and error correction.

To provide a preliminary hardware feasibility analysis, a first-order Power, Performance, and Area (PPA) estimate was conducted based on the open-source SkyWater 130 nm PDK and the associated sky130_fd_pr_reram physical models. For an assumed 64 × 32 macro operating at 100 MHz, the projected throughput is 6.4 Giga Operations Per Second (GOPS). The target 100 MHz operating frequency is supported by schematic-level timing simulations: the measured write time is 8.91 ns in Figure 12, the read time is 4.38 ns in Figure 13, and the digital datapath delay is 3.78 ns in Figure 14. The resulting timing margin suggests that the target frequency is feasible at schematic-level. Because no layout or parasitic extraction has been completed, the reported macro area of approximately 0.02 mm^2 should be interpreted as a first-order estimate rather than a post-layout result.

Using a low read voltage of 0.2 V and leveraging the 78.08% zero physical weight ratio observed in the proposed hardware-aware model, the projected array read power is reduced to the microwatt range under the adopted assumptions. When combined with the digital MAC tree, the projected total dynamic power is approximately 0.7 mW, corresponding to an estimated energy efficiency of about 9.1 TOPS/W. These values are based on schematic-level simulations and first-order scaling, and they require further validation through layout implementation and post-layout simulation.

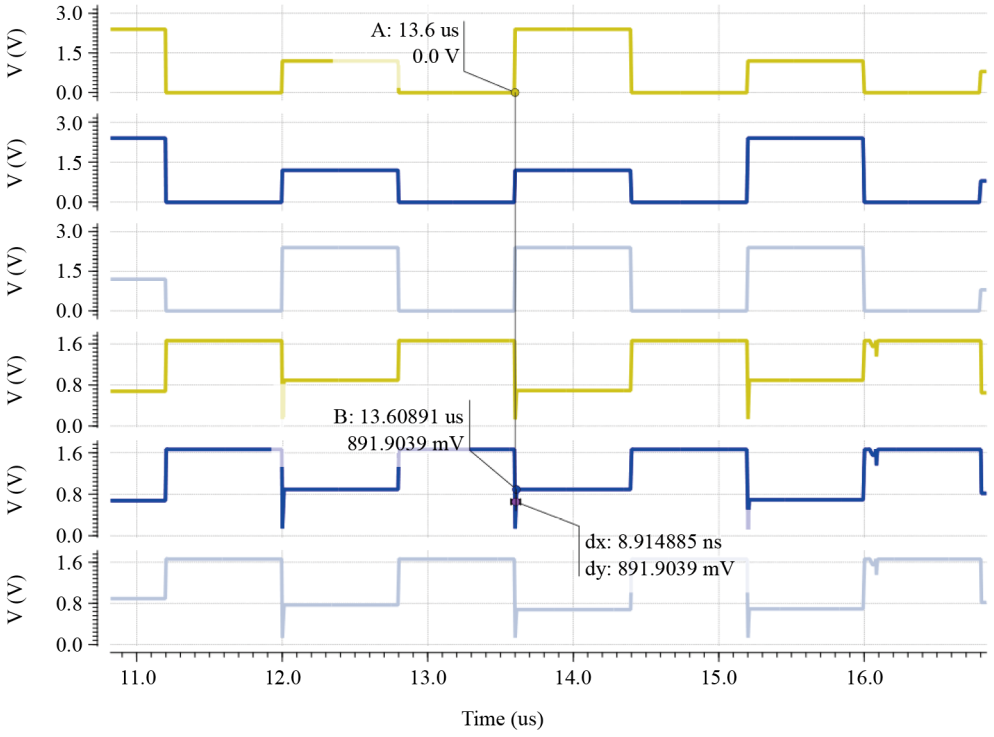

Figure 12. Write inhibit biasing scheme during the SET phase. Selected wordlines are driven to 1.7 V while unselected sourcelines are held at the inhibit voltage of 1.2 V, keeping the differential across unselected cells below the switching threshold

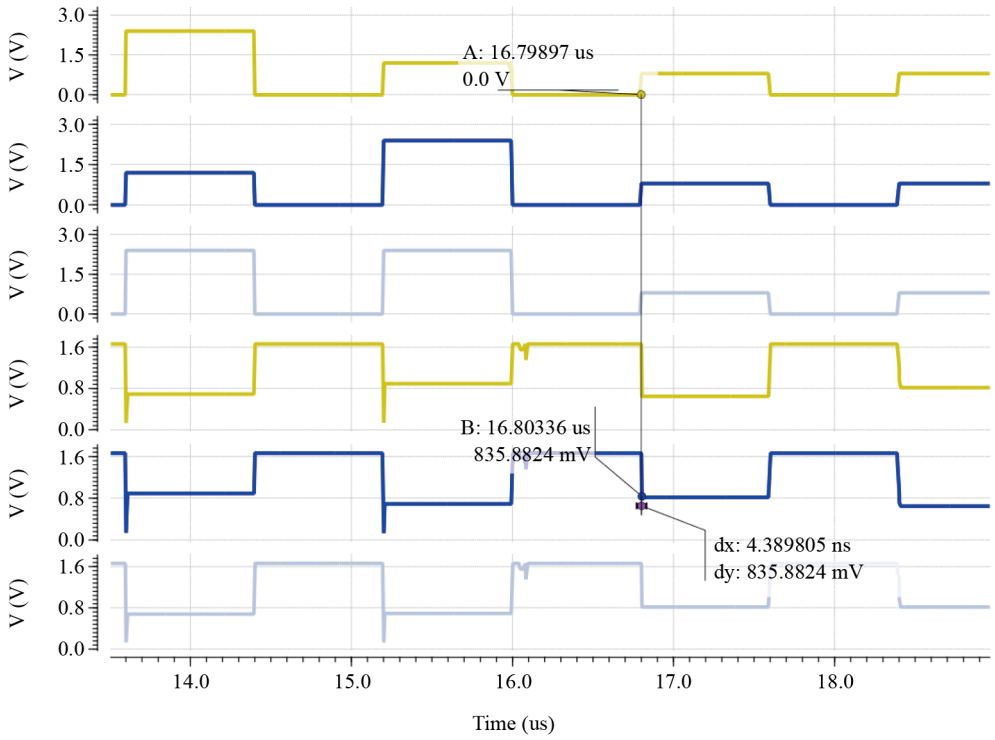

Figure 13. Read operation of a memristor cell. A low read Voltage (Vread = 0.2 V) is applied across the selected row; the resulting bitline current is sensed and digitized before any analog accumulation occurs

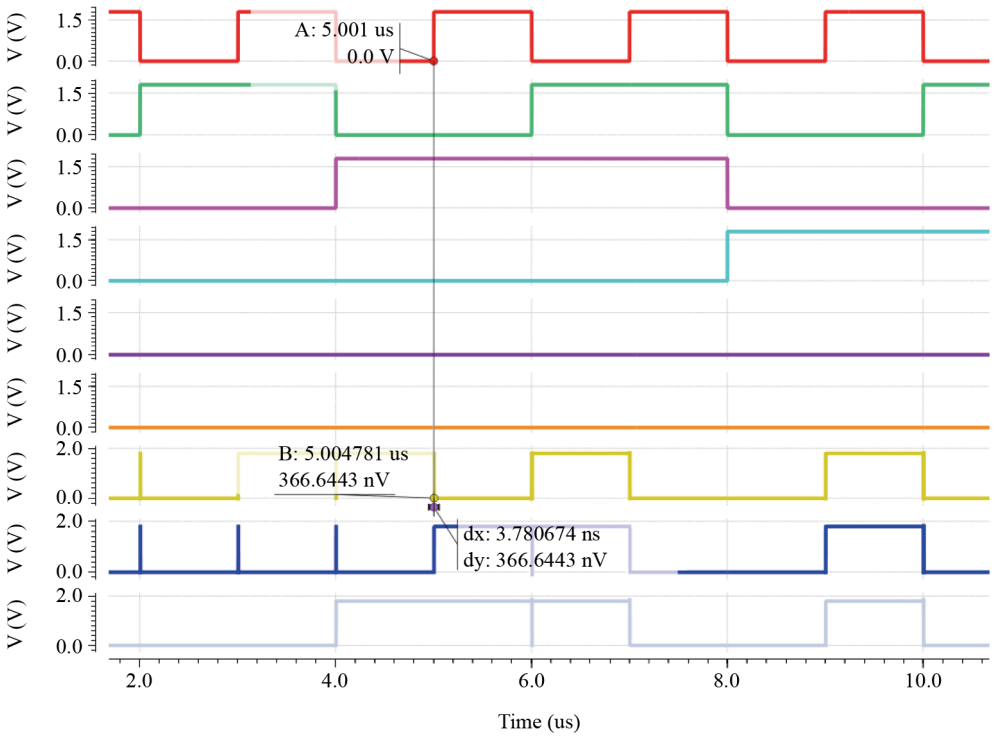

Figure 14. Representative 8-bit accumulator and ReLU block of the pipelined digital MAC datapath. The accumulator processes partial products, with the MSB driving a multiplexer-based ReLU that sets negative results to zero

Compared with fabricated CMOS digital CIM macros and fully integrated memristor-based Systems on Chips (SoCs), this work targets a different validation scope. Recent Static Random-Access Memory (SRAM) digital CIM macros demonstrate high silicon-level efficiency in advanced CMOS technologies, while fully integrated memristor chips demonstrate edge learning with fabricated devices [6], [7]. In contrast, the present work focuses on a schematic-level 1T3R signed weight mapping and low-bit digital near-memory backend using binary memristor states. Therefore, the reported accuracy and PPA values should be interpreted as an early feasibility study rather than a direct post-layout or silicon-level comparison. The main value of this work is the physical bit-slicing scheme, the write-inhibit array control, and the hardware-aware evaluation flow for a 3-bit signed 1T3R digital near-memory architecture.

Table 4 summarizes the source and interpretation of the reported PPA metrics. It separates schematic-level simulation results from calculated projections and first-order estimates, and clarifies that full layout implementation, parasitic extraction, and post-layout power analysis have not been completed in this work.

Table 4. PPA methodology and assumptions

Metric	Value	Source/Method	Interpretation
Write delay	8.91 ns	Cadence Virtuoso schematic transient simulation, Figure 12	Schematic level result
Read delay	4.38 ns	Cadence Virtuoso schematic transient simulation, Figure 13	Schematic level result
Digital datapath delay	3.78 ns	Cadence Virtuoso schematic transient simulation, Figure 14	Schematic level result
Target frequency	100 MHz	Selected based on schematic timing margin	Projected operating point
Throughput	6.4 GOPS	Calculated from the assumed 64-operation macro evaluation at 100 MHz	Calculated projection
Macro area	~ 0.02 mm^2	First order cell and block level area estimate	Not post-layout area
Total dynamic power	~ 0.7 mW	Schematic level block estimates combined with scaling assumptions	First order estimate
Energy efficiency	~ 9.1 TOPS/W	Projected throughput divided by projected total dynamic power	Estimated metric

5. Discussion and future work

While the proposed architecture demonstrates potential tolerance for offline inference, future work is needed to extend it toward *in situ* on-chip training [11]. The main future directions are summarized as follows.

5.1 *Pulse-based in-situ weight updates*

The current 1T3R digital slicing scheme provides a stable foundation for weight representation. Future work will explore fine-grained voltage pulses for SET and RESET operations to modulate memristance states directly on-chip. Prior work on monolithically integrated memristive networks [4] suggests that volatile memristive neurons may support richer spatiotemporal dynamics. However, mapping learning rules such as Spike-Timing-Dependent Plasticity (STDP) onto physical crossbars is affected by device variation. Therefore, future dynamic update schemes should be combined with variation-tolerant online learning frameworks [12]. In addition, experimental demonstrations of multilayer memristor networks have shown that closed-loop *in situ* learning can adapt to static device-to-device variation and IR drops [13]. Recent fully integrated memristor SoC work in Science [6] also indicates that simplified sign- and threshold-based learning rules can train edge AI networks without high-precision conductance tuning. These results provide a possible direction for extending the present 1T3R architecture toward low-precision on-chip learning.

5.2 *Hardware-mapped backpropagation*

To further reduce power consumption during training, future work may investigate hardware mapping of backpropagation. This would use the transposable nature of the 1T3R crossbar array to pass error signals backward through the same physical grid. Specialized peripheral circuits could then support partial derivative calculations using physical laws such as Ohm's Law and Kirchhoff's Current Law. The feasibility of *in situ* learning has been explored in accelerators such as Ziksa [14], which integrates regression and backpropagation modules with fabricated TiN/TaOx/TaTiN memristor crossbars. Such systems show that hardware-oriented training can help mitigate the nonidealities of multilevel networks. As the architecture moves toward spiking dynamics, future learning engines may also use surrogate gradient approximations to support backpropagation through time [15].

5.3 *Endurance, fault tolerance, and reliability optimization*

On-chip training imposes strict requirements on the write endurance of memristive devices. Future iterations of this architecture will incorporate sparsity-aware update algorithms, which update only the most significant weights to reduce physical wear on the memristors. In addition, memristor crossbars can suffer from hard defects such as stuck-at faults during fabrication and prolonged operation. To improve system reliability, future online learning frameworks may integrate hardware-efficient knowledge distillation techniques [16]. By using a multi-loss distillation mechanism during online training, the architecture could identify and bypass stuck-at fault defects and maintain inference accuracy under hardware degradation. Furthermore, edge deployment exposes the system to ambient input noise and cycle-to-cycle programming variation. Prior evaluations of memristive edge AI indicate that aggressive weight quantization can increase this vulnerability, while moderate noise injection during training can improve variation tolerance and support sublinear energy scaling [17]. Therefore, adding a noise-aware regularization scheme is a useful future direction for improving reliability in practical edge environments.

5.4 *Reconfigurable bit-precision architecture*

Driven by the diverse accuracy requirements of modern DNNs, recent digital CIM macros have explored reconfigurable precision architectures [18]. While the current 1T3R bit-slicing scheme is optimized for 3-bit weights, future designs can explore dynamic precision configurability. By adding programmable shift-and-add logic in the digital peripheral circuits, multiple 1T3R slices could be grouped to support higher precisions, such as 4-bit Integer (INT4) to 8-bit Integer (INT8), for accuracy-sensitive layers, or used separately for more energy constrained layers. This adaptability may support a more flexible multi-precision neuromorphic accelerator.

5.5 *Cross-domain signal processing*

While the current evaluation focuses on image classification, the high-throughput nature of the 1T3R MAC pipeline may also be useful for time series data. Future iterations could adapt this architecture for real-time biomedical signal processing, inspired by neuromorphic diagnostic systems [19], expanding the application scope from static images to dynamic sensing tasks.

5.6 *Hybrid memristive-memcapacitive systems*

While memristors provide a nonvolatile foundation for inference, frequent switching during on-chip training can increase energy consumption. Recent co-design frameworks suggest that integrating memcapacitors, which rely on reactive rather than dissipative properties, can reduce the power consumption of weight updates [20]. Future research will evaluate a hybrid architecture where 1T3R memristive slices are complemented by capacitive elements to reduce static leakage during gradient accumulation and improve edge AI efficiency.

6. Conclusion

This paper presents a schematic-level digital near-memory computing architecture based on 1T3R memristor arrays and a pipelined digital backend. By storing 3-bit two's complement weights through spatial bit-slicing and using a write-inhibit scheme for unselected rows, the design reduces dependence on analog current accumulation and high-resolution ADCs. The MAC and ReLU backend processes digitized array outputs with low-bit digital arithmetic, while the Python-based hardware-aware evaluation demonstrates 79.70% MNIST recognition accuracy with a 78.08% zero physical weight ratio under strict quantization constraints. Additional bit-level sensitivity analysis indicates small degradation at a 0.1% physical bit error or fault rate, but larger degradation as read errors or permanent defects approach the sub-percent to percent range. The reported PPA values are first-order projections based on schematic-level simulations and scaling assumptions. Full layout implementation, parasitic extraction, post-layout simulation, and larger benchmark evaluation remain important future work.

Author contributions

Conceptualization, Z.H. and Y.Y.; methodology, Z.H.; software, Z.H.; validation, Z.H.; formal analysis, Z.H. and X.W.; investigation, Z.H.; resources, Z.H. and X.W.; data curation, Z.H., and X.W; writing—original draft preparation, Z.H.; writing—review and editing, Z.H., X.W. and Y.Y.; visualization, Z.H.; supervision, Z.H. and Y.Y.; project administration, Z.H. and Y.Y. All authors have read and agreed to the published version of the manuscript.

Funding

This research received no external funding.

Data availability statement

The publicly available MNIST dataset was used in this study. No new experimental datasets were collected. Additional simulation results supporting the findings of this study are available from the corresponding author upon reasonable request.

Acknowledgements

The authors express their thanks to the people helping with this work. In the preparation of this work, the authors used Gemini to assist with text editing and formatting refinement. After using this tool/service, the authors reviewed and edited the content as needed, and take full responsibility for the content of the publication.

Conflict of interest

The authors declare no conflicts of interest.

References

[1] F. Nowshin and Y. Yi, "Memristor-based deep spiking neural network with a computing-in-memory architecture," in 2022 23rd International Symposium on Quality Electronic Design (ISQED), 2022, pp. 1-6. Available: https://doi.org/10.1109/ISQED54688.2022.9806206.

[2] X. Wang, Z. Zhou, and Y. Yi, "Transforming AI landscape with neuromorphic computing and chiplets," in *Energy-Efficient Devices and Circuits for Neuromorphic Computing*, F. A. Khanday, Ed. Elsevier, 2026, pp. 405-428. Available: https://doi.org/10.1016/B978-0-443-29981-0.00004-5.

[3] F. Aguirre, A. Sebastian, M. Le Gallo, W. Song, T. Wang, J. J. Yang, W. Lu, M.-F. Chang, D. Ielmini, Y. Yang, et al., "Hardware implementation of memristor-based artificial neural networks," *Nature Communications*, vol. 15, no. 1, p. 1974, 2024. Available: https://doi.org/10.1038/s41467-024-45670-9.

[4] Q. Duan, Z. Jing, X. Zou, Y. Wang, K. Yang, T. Zhang, S. Wu, R. Huang, and Y. Yang, "Spiking neurons with spatiotemporal dynamics and gain modulation for monolithically integrated memristive neural networks," *Nature Communications*, vol. 11, no. 1, p. 3399, 2020. Available: https://doi.org/10.1038/s41467-020-17215-3.

[5] Y. Zhang, L. Chu, and W. Li, "A fully-integrated memristor chip for edge learning," *Nano-Micro Letters*, vol. 16, no. 1, p. 166, 2024. Available: https://doi.org/10.1007/s40820-024-01368-7.

[6] W. Zhang, P. Yao, B. Gao, Q. Liu, D. Wu, Q. Zhang, Y. Li, Q. Qin, J. Li, Z. Zhu, et al., "Edge learning using a fully integrated neuro-inspired memristor chip," *Science*, vol. 381, no. 6663, pp. 1205-1211, 2023. Available: https://doi.org/10.1126/science.ade3483.

[7] H. Mori, W.-C. Zhao, C.-E. Lee, C.-F. Lee, Y.-H. Hsu, and C.-K. Chuang, "A 4nm 6163-TOPS/W/b 4790-TOPS/mm^2/b SRAM based digital-computing-in-memory macro supporting bit-width flexibility and simultaneous MAC and weight update," in 2023 IEEE International Solid-State Circuits Conference (ISSCC), 2023, pp. 132-134. Available: https://doi.org/10.1109/ISSCC42615.2023.10067555.

[8] M. A. Zidan, H. A. H. Fahmy, M. M. Hussain, and K. N. Salama, "Memristor-based memory: The sneak paths problem and solutions," *Microelectronics Journal*, vol. 44, no. 2, pp. 176-183, 2013. Available: https://doi.org/10.1016/j.mejo.2012.10.001.

[9] M. R. Eslami, D. Biswas, S. Takhtardeshir, S. S. Sharif, and Y. M. Banad, "On-chip learning with memristor-based neural networks: Assessing accuracy and efficiency under device variations, conductance errors, and input noise," *arXiv*, 2024. [Online]. Available: https://doi.org/10.48550/arXiv.2408.14680. [Accessed May 3, 2026].

[10] T. V. Nguyen, J. An, and K.-S. Min, "Comparative study on quantization-aware training of memristor crossbars for reducing inference power of neural networks at the edge," in 2021 International Joint Conference on Neural Networks (IJCNN), 2021, pp. 1-6. Available: https://doi.org/10.1109/IJCNN52387.2021.9533834.

[11] R. Hasan, T. M. Taha, and C. Yakopcic, "On-chip training of memristor crossbar based multi-layer neural networks," *Microelectronics Journal*, vol. 66, pp. 31-40, 2017. Available: https://doi.org/10.1016/j.mejo.2017.05.005.

[12] Y. Zheng, Y. He, R. Shen, J. Li, R. Wang, and D. Wang, "Variation-tolerant circuit design and online learning framework for memristor-based Trace-STDP SNN," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 16, no. 2, pp. 348-362, 2026. Available: https://doi.org/10.1109/JETCAS.2026.3663895.

[13] C. Li, D. Belkin, Y. Li, P. Yan, M. Hu, N. Ge, H. Jiang, E. Montgomery, P. Lin, Z. Wang, et al., "Efficient and self-adaptive in-situ learning in multilayer memristor neural networks," *Nature Communications*, vol. 9, no. 1, p. 2385, 2018. Available: https://doi.org/10.1038/s41467-018-04484-2.

[14] A. M. Zyarah, N. Soures, L. Hays, R. B. Jacobs-Gedrim, S. Agarwal, M. Marinella, and D. Kudithipudi, "Ziksa: On-chip learning accelerator with memristor crossbars for multilevel neural networks," in 2017 IEEE International Symposium on Circuits and Systems (ISCAS), 2017, pp. 1-4. Available: https://doi.org/10.1109/ISCAS.2017.8050531.

[15] J. K. Eshraghian, M. Ward, E. O. Neftci, X. Wang, G. Lenz, G. Dwivedi, M. Bennamoun, D. S. Jeong, and W. D. Lu, "Training spiking neural networks using lessons from deep learning," *Proceedings of the IEEE*, vol. 111, no. 9, pp. 1016-1054, 2023. Available: https://doi.org/10.1109/JPROC.2023.3308088.

[16] M. Guo, X. Zhang, G. Dou, and H. H.-C. Iu, "A knowledge distillation online training circuit for fault tolerance in memristor crossbar array-based neural networks," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 72, no. 11, pp. 7120-7131, 2025. Available: https://doi.org/10.1109/TCSI.2025.3561803.

[17] M. R. Eslami, D. Biswas, S. Takhtardeshir, S. S. Sharif, and Y. M. Banad, "Efficient memristor neural networks for edge AI: On-chip learning, noise robustness, device-variation tolerance, and sublinear energy and time scaling," *IEEE Transactions on Computers*, vol. 75, no. 5, pp. 1723-1736, 2026. Available: https://doi.org/10.1109/TC.2026.3661114.

[18] J. Bazzi, M. E. Fouda, and A. Eltawil, "Reconfigurable precision INT4-8/FP8 digital compute-in-memory macro for AI acceleration," in 2025 IEEE International Symposium on Circuits and Systems (ISCAS), 2025, pp. 1-5. Available: https://doi.org/10.1109/ISCAS56072.2025.11044064.

[19] M. Sharifshazileh, K. Burelo, J. Sarnthein, and G. Indiveri, "An electronic neuromorphic system for real-time detection of high frequency oscillations (HFO) in intracranial EEG," *Nature Communications*, vol. 12, no. 1, p. 3095, 2021. Available: https://doi.org/10.1038/s41467-021-23342-2.

[20] A. Singh and B.-G. Lee, "Framework for in-memory computing based on memristor and memcapacitor for on-chip training," *IEEE Access*, vol. 11, pp. 112590-112599, 2023. Available: https://doi.org/10.1109/ACCESS.2023.3324375.