TU Delft

Delft University of Technology

Document Version
Final published version

Licence
Dutch Copyright Act (Article 25fa)

Citation (APA)
Sehgal, A., Kumar Shukla, A., Diware, S., Soni, S., Dhull, S., Shreya, S., Roy, S., & Bishnoi, R. K. (2025). Enhancing Parallelism and Energy-Efficiency in SOT-MRAM based CIM Architecture for On-Chip Learning. In *Proceedings of the 2025 62nd ACM/IEEE Design Automation Conference (DAC)* IEEE. https://doi.org/10.1109/DAC63849.2025.11424425

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright
In case the licence states “Dutch Copyright Act (Article 25fa)”, this publication was made available Green Open Access via the TU Delft Institutional Repository pursuant to Dutch Copyright Act (Article 25fa, the Taverne amendment). This provision does not affect copyright ownership.
Unless copyright is transferred by contract or statute, it remains with the copyright holder.

Sharing and reuse
Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy
Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

This work is downloaded from Delft University of Technology.

Enhancing Parallelism and Energy-Efficiency in SOT-MRAM based CIM Architecture for On-Chip Learning

Anubha Sehgal*, Alok Kumar Shukla†, Sumit Diware‡, Sandeep Soni*, Seema Dhull§, Sonal Shreya¶, Sourajeet Roy*, Rajendra Bishnoi‡
* Indian Institute of Technology Roorkee, † Madan Mohan Malaviya University of Technology
‡ Delft University of Technology, § Global Foundaries, ¶ Aarhus University
{anubha_s, sandeep_s, sourajeet.roy}@ece.iitr.ac.in, aks.ece@mmmut.ac.in, seema@globalfoundaries.com,
{s.s.diware, r.k.bishnoi}@tudelft.nl, sshreya@ece.au.dk

2025 62nd ACM/IEEE Design Automation Conference (DAC) | 979-8-3315-0304-8/25/$31.00 ©2025 IEEE | DOI: 10.1109/DAC63849.2025.11424425

***Abstract*—Computational-In-Memory (CIM) architectures have emerged as energy-efficient solutions for Artificial Intelligence (AI) applications, enabling data processing within memory arrays and reducing the bottleneck associated with data transfer. The rapid advancement of AI demands real-time on-chip learning but implementing this with CIM architectures poses significant challenges, such as limited parallelism and energy-efficiency during training and inference. In this paper, we propose a novel CIM architecture specifically designed for on-chip learning applications, which capitalizes on the unique properties of Spin-Orbit Torque (SOT) technology to enhance both parallelism and energy-efficiency in computation. The proposed architecture incorporates a bulk-write mechanism for SOT-cell based arrays, enabling efficient weight updates during on-chip training. Additionally, we develop a scheme to process vector elements concurrently for vector-matrix multiplications during inference. To achieve this, we design multi-port bit-cell access capabilities along with their associated control mechanisms. Simulation results show a 5.82× reduction in latency and a 3.20× improvement in energy-efficiency compared to standard SOT-MRAM based CIM, with negligible overhead.**

***Index Terms*—Spin-Orbit Torque, multi-port memory, on-chip learning, neural networks, computation-in-memory**

I. Introduction

The rising demand for large AI datasets, along with advancements in edge computing and Internet-of-Things devices, has amplified the need for energy-efficient hardware architectures. Traditional computing systems rely heavily on data movement between processing and memory units, facing memory-wall challenges like high energy consumption and latency [1], [2]. To address these challenges, *Computing-In-Memory* (CIM) presents a promising solution by integrating memory and computation within the same hardware unit that minimizes the data movement [3]–[7]. Spintronic devices, especially *Spin-Orbit Torque Magnetic Random Access Memory* (SOT-MRAM), are ideal for such architectures due to their high-density data storage and energy-efficient operation [8]–[11]. SOT-MRAM based CIM architectures use the resistance states of SOT devices for analog matrix-vector multiplications within memory, enhancing efficiency and reducing access latency during their write operations [12]. Although such CIM architectures offer good parallelism, the demand for high computational throughput is much greater, especially for implementing on-chip learning, which requires continuous real-time updates directly within the hardware.

Limited work has been done to enhance the parallelism of CIM architectures for on-chip learning. For instance, the work proposed in [13] facilitates N parallel MAC operations across columns in the memory array by clamping the source lines with low-power adjustable resolution *Analog to Digital Converters* (ADCs) for enhancing parallelism in *Resistive Random Access Memory* (ReRAM) based CIM architecture. A higher ADC resolution leads to better system recognition accuracy while consuming more energy and latency. Moreover, a voltage-based differential referencing-in-array scheme has been proposed to enable multi-operand logic operations with high parallelism. However, this technique is limited to logic operations and does not apply to *Multiply-and-Accumulate* (MAC) operations [14]. Several approaches aim to boost the energy-efficiency of CIM architectures for on-chip learning. For example, an energy-efficient heterogeneous multicore processor proposed in [15] is designed for a deep belief network, equipped with transposable weight memory for on-chip local learning. Nevertheless, it encounters bottlenecks, including the area-inefficient transposable memory based on registers and limited parallelism caused by core utilization degradation. Moreover, an *Static Random Access Memory* (SRAM) based CIM architecture proposed in [16] uses pipeline optimization for CIM training to improve the energy-efficiency. However, the periphery circuit of transposable systems can also introduce complexity and latency during online training, limiting their effectiveness in real-time applications. Since most existing methods either demand high energy or suffer from slow read/write operations, there is a need for a low-latency, energy-efficient CIM solution that ensures high parallelism and supports efficient on-chip learning.

In this paper, we present a novel SOT-MRAM based CIM architecture designed for on-chip learning that leverages the

This work was supported by the Scheme for Promotion of Academic and Research Collaboration (SPARC), Grant No: SPA-2136-ECD-CNA/23-24.

Authorized licensed use limited to: TU Delft Library. Downloaded on March 12,2026 at 12:20:13 UTC from IEEE Xplore. Restrictions apply.

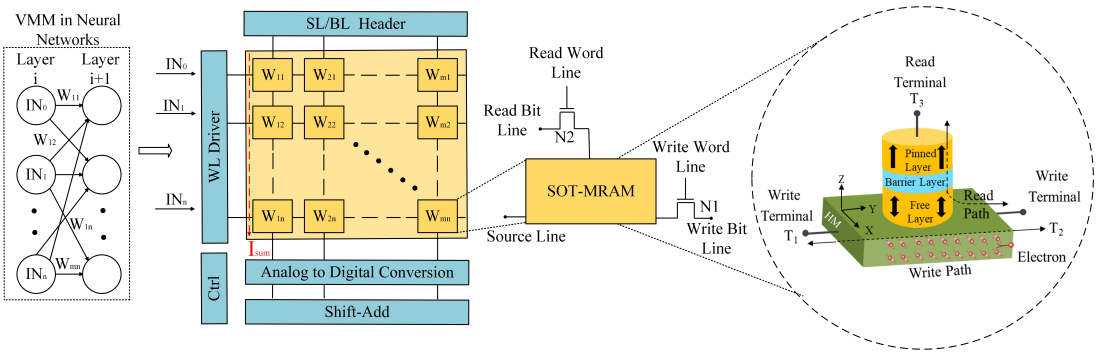


Fig. 1. Vector-matrix multiplication in Computation-In-Memory (CIM) architecture using SOT-MRAM bit-cell for neural networks.

unique features of the bit-cell to enhance parallelism and energy-efficient computation. Our contributions for this paper are as follows:

- We propose a bulk-write mechanism that facilitates effective weight updates during on-chip training for SOT-MRAM bit-cell based arrays.
- We introduce a concurrent processing scheme for vector-matrix multiplication, utilizing a multi-port SOT-MRAM bit-cell and their associated control mechanisms during inference.
- We perform a comprehensive evaluation of our system architecture for CIM-based image classification.

Simulation results demonstrate that our proposed approach can reduce latency by 5.82× and improve energy-efficiency by 3.20× compared to standard single-port SOT-MRAM based CIM architecture.

The rest of the paper is organized as follows: Section II presents the background of CIM architecture, SOT-MRAM technology, and on-chip learning. Section III presents the proposed CIM architecture, and Section IV illustrates the results. Finally, Section V concludes the paper.

II. Background

A. Computation-In-Memory Architecture

CIM architecture addresses data transfer bottlenecks by performing the operations and storage in the same physical location, thereby overcoming the overhead associated with costly data movements inherent in conventional von Neumann architectures [17]–[19]. Fig. 1 demonstrates the CIM architecture for neural networks employing *Vector-Matrix Multiplication* (VMM). The network weights are mapped as resistance states of bit-cells in the crossbar array. Multiple inputs are applied simultaneously to the bit-cells in each row, which are then multiplied by weights and generate current. The resulting current values are summed up in a column to execute VMM operations and then converted to digital outputs using ADCs. This approach provides high computational parallelism and improves energy-efficiency [20].

B. Spin-Orbit Torque Technology

A SOT-MRAM cell stores data in the form of magnetic resistance having a low switching current and faster access speed [21], [22]. It consists of a barrier oxide layer sandwiched between two ferromagnetic layers, a free and pinned/fixed layer, as shown in Fig. 1. The magnetic orientation of the pinned layer is always fixed, while that of the free layer can be freely rotated depending on the direction of the current. It is a three-terminal device having isolated read and write terminals that eliminate the oxide barrier breakdown issue. The write current is applied between two write terminals (T_1 and T_2) through heavy metal that generates spin-orbit torque on the magnetization of the free layer. This torque causes the magnetic orientation of the free layer to change, resulting in a change in resistance values due to the *Spin Hall Effect* (SHE) and Rashba effect [23]–[25]. When the magnetic orientations of the free and pinned layers are aligned in the same direction, the MRAM cell exhibits a low resistance state while the antiparallel exhibits a high resistance state. For read operations, a low read current is applied between the read terminal (T3) and the write terminal (T2). This read current is sensed by a sense amplifier to determine the stored value. A typical bit-cell structure is shown in Fig. 1.

C. On-Chip Learning

On-chip learning enables a computing system to execute machine learning tasks directly within the hardware where the data is processed. Integrating learning algorithms with memory elements such as SOT-MRAM devices on the same chip facilitates real-time updates and adaptations. The training process involves four steps, namely 1) feed-forward, 2) backpropagation for error calculation, 3) weight gradient calculation, and 4) weight update [26], [27]. During forward propagation, data is processed and stored in memory cells, which perform computations like matrix multiplications. The intermediate activations in each layer are stored in the buffers and the error between the predicted output and the label is calculated. During backpropagation, the weight gradients of each layer are calculated based on learning algorithms by convolution between the error and the activation obtained in the feed-forward process. Finally, the weights of the current layer are updated by the learning rate. For inference, the system uses the trained model to make predictions directly, thus reducing latency and energy consumption by minimizing

Authorized licensed use limited to: TU Delft Library. Downloaded on March 12,2026 at 12:20:13 UTC from IEEE Xplore. Restrictions apply.

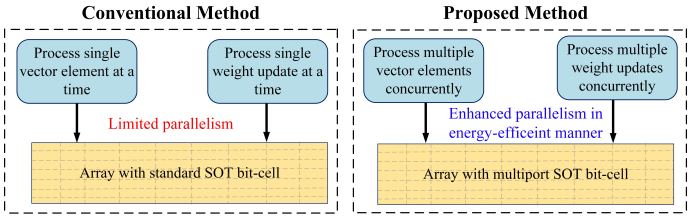


Fig. 2. Overview of conventional and proposed SOT-MRAM architectures.

data transfer. This integration enhances efficiency and enables adaptive, real-time learning and decision-making [28].

III. Proposed SOT-MRAM based CIM Architecture

In this section, we first present an overview of our proposed CIM architecture for on-chip learning with a detailed explanation of bit-cell design, control circuit, and weight calculation block. Subsequently, we provide the design details for weight update and inference.

A. Overview

As discussed, the on-chip training process requires real-time weight updates. For a SOT-MRAM based CIM architecture, this process is primarily divided into two components, namely, weight updates and the inference engine. As shown in Fig. 2, in conventional methods, weight updates are performed by programming cells one at a time, means the number of cycles required is equal to the number of write operations. Additionally, during inference, conventional approaches process one vector element at a time for vector-matrix multiplication, as shown in Fig. 2. This limits parallelism in conventional SOT-MRAM based CIM architectures, making them not suitable for fast real-time applications such as real-time ECG monitoring, autonomous vehicle control, and drone navigation, all of which demand rapid processing. In the proposed method, we utilize a bulk-write technique to write multiple cells simultaneously. Due to the symmetric resistance characteristics of write-1 and write-0 operations in SOT-MRAM cell, we can drive higher currents (depending on the technology and architecture), allowing for more efficient programming. To enhance the inference engine, we process multiple vector elements concurrently. This is achieved by modifying the existing SOT-MRAM bit-cell design with a multi-port bit-cell configuration. As a result, the proposed architecture offers greater parallelism and improved performance for on-chip learning applications. The next subsections detail the architecture and its components.

B. Architecture Design

Fig. 3 presents the macro-level architecture, illustrating the composition of a single CIM tile. Each tile executes the MAC operation that involves the input and the weights stored in the SOT-MRAM bit-cell. Multiple tiles can be cascaded together to form a fully connected neural network. Each tile consists of mainly the SOT-MRAM bit-cell, the control circuit, data converters, and weight calculation/update blocks. The core of the architecture utilizes SOT-MRAM cells having isolated read and write paths that allow read and write currents to flow simultaneously through the same bit-cell without affecting each other's functionality. The control circuit dynamically adjusts the write current value based on the number of cells being written. It is designed to handle the maximum number of inputs per clock cycle, enhancing parallelism by activating the corresponding word lines efficiently by using an input encoding design unit. It also manages multiple weight gradient calculations along with multiple weight updates. The read outputs are then conveyed on ADCs, which are then added together in parallel with the help of a local summation unit. Then, the sum of all local summation units is consolidated by the global summation unit.

C. SOT-MRAM based Multi-Port Bit-Cell Design

To enhance the energy-efficiency and parallelism in CIM architecture, a multi-port SOT-MRAM bit-cell design is proposed. The schematic of the proposed multi-port SOT-MRAM based bit-cell is shown in Fig. 4 (a). Please note that the write current in SOT-cell does not pass through the SOT device layer stacks, it flows through a heavy metal layer instead, as shown in Fig. 1. This ensures that the write current for both write-0 and write-1 operations remains the same at the cell level. In contrast, other two-terminal memristor devices such as ReRAM, *Spin Transfer Torque MRAM* (STT-MRAM), and *Phase-Change Memory* (PCM) exhibit asymmetry, as the current passes through the device itself, and its resistance value creates this imbalance [29], [30]. To address this asymmetry beyond the bit-cell level, we have incorporated an additional transistor (N2) into the write path of the conventional bit-cell design, as illustrated in Fig. 4 (a) [31].

During a write operation, both *Write Word Lines* (WWL1 and WWL2) are set to 1, turning ON transistors N1 and N2, while *Read Word Lines* (RWLs) are set to 0, keeping N3-N8 OFF. The sizes of the N1 and N2 are kept large to drive a high current for writing. The *Source Line* (SL) and *Word Bit Line* (WBL) are kept either 1 or 0 depending upon the bit to be written. The write path is thus established between N1 and

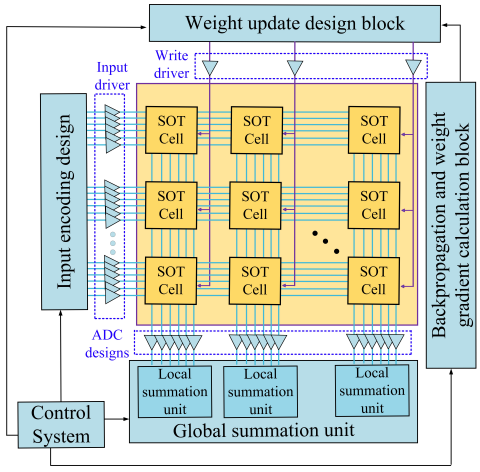
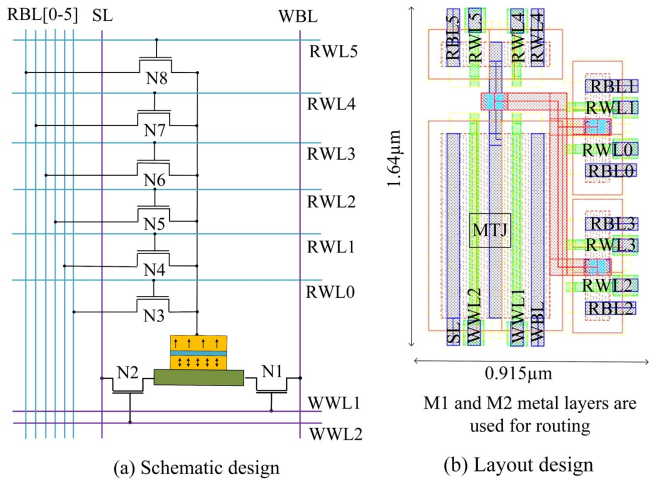


Fig. 3. Macro-level architecture of proposed single tile.

Authorized licensed use limited to: TU Delft Library. Downloaded on March 12,2026 at 12:20:13 UTC from IEEE Xplore. Restrictions apply.



(a) Schematic design (b) Layout design

Fig. 4. Illustration of one-write six-read multi-port SOT-MRAM bit-cell.

N2. For a read operation, both SL and WBL are set to 0. To establish the read path, N2 is kept ON by setting WWL1 to 1 and WWL2 to 0, while RWLs are set to 1 whenever the read transistors need to be accessed. In contrast to the 2T1R bit-cell in Fig. 1, the two WWLs are used so that only one transistor out of N2 or N1 is set ON during the read operation. To sense the output, the sense amplifier is connected to the *Read Bit Line* (RBL) at one end and reference at another. The layout of a one-write and six-read bit-cell design, shown in Fig. 4 (b), having one write port and six read ports, implemented using *Generic Process Design Kit* (GPDK) 45 nm technology and utilizing two metal layers.

D. Design Implementation Strategy

1) Bulk write mechanism for weight updates: As we know, weight updates are a crucial step where the computing engine adapts itself for real-time applications using an on-chip training mechanism. In our proposed architecture, we perform these updates for multiple cells simultaneously using a bulk-write mechanism, as shown in Fig. 5. After backpropagation and weight gradient calculation, the new weights are stored in memory, and the write 0's and 1's to be updated are identified. We first select the 1's, store their locations in latches, then pass a high current using write drivers, selecting those to be written based on the latch values. Since the write-1 operation shares the current in the same direction and it has symmetric current paths, so it is easily managed using write drivers.

We have conducted a cell-level analysis to determine the effective current (I_e) at which energy consumption is minimum that is required for a successful write thus, optimizing energy-efficiency (see Fig. 7 (a)). The maximum current provided by the write driver depends on both the technology and design. Therefore, the number of cells (N) that can be written in parallel is the ratio of the total current (maximum) to the effective current (I_e) for writing a single cell. If the number of cells exceeds N, additional iterations are needed to complete the write-1 operation. Here, the control circuit dynamically adjusts the current based on the number of cells, ensuring energy-efficient writing. In this way, weights are mapped to the conductance values of SOT-MRAM bit-cells across columns in the crossbar array in parallel while maintaining energy-efficiency. Once the write-1 process is complete, we perform the write-0 operation, following a similar procedure but reversing the current direction.

2) Design for inference: In CIM architectures, inference operations typically involve applying input vectors to crossbars, which represent matrices, in order to perform vector-matrix multiplications. In our proposed architecture, multiple vectors are applied to the crossbars through a multi-port cell design, as illustrated in Fig. 6. The vector elements are connected to the gates of the access transistors. When a read voltage is applied through the RBL, a dot product operation is performed, and the result is quantized using an ADC. We employ a five-bit precision SAR ADC design in this work, as described by [32]. Based on our cell-level analysis, we can perform up to six sets of MAC operations (for example, executing six separate vector-matrix multiplications in parallel) with our multi-port bit-cell design while maintaining a margin of 40 mV [14]. Furthermore, once the values for all vectors are obtained, they are combined to generate the final output for the neural network.

IV. Results

In this section, we provide a detailed description of the setup used for calculating results, followed by a comprehensive explanation of both circuit-level and system-level evaluation.

A. Setup

For the circuit-level implementation, we employed the SOT-MRAM based model proposed in [33] with parameters depicted in Table I. We employed a GPDK 45 nm SPICE model to simulate the CMOS components and design the layout. The simulations are performed using the Cadence

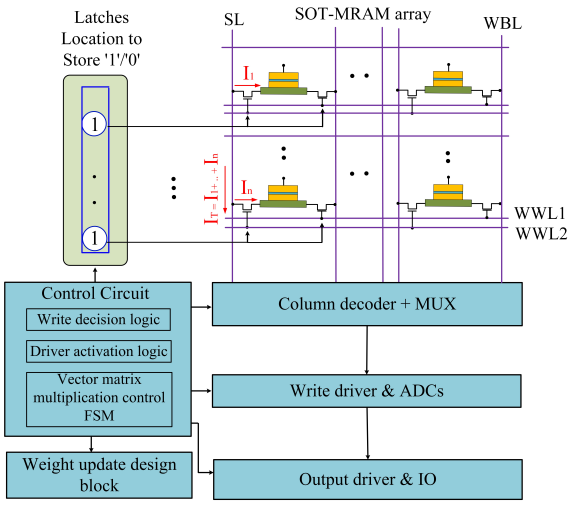


Fig. 5. Weight update implementation using bulk write mechanism.

Authorized licensed use limited to: TU Delft Library. Downloaded on March 12,2026 at 12:20:13 UTC from IEEE Xplore. Restrictions apply.

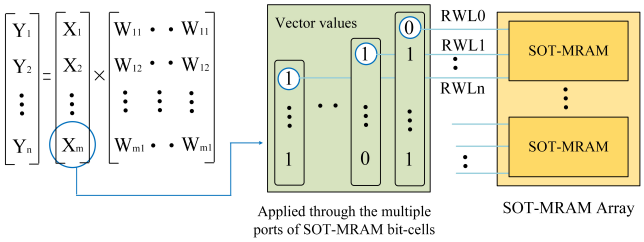


Fig. 6. Multiple vector multiplication for inference using multi-port design.

Virtuoso with a supply voltage of 1.2 V and temperature of 27°C as well as process variation of $\pm 3\sigma$ is considered. We used Cadence Genus for synthesis and Calibre PEX for parasitic extraction, while also considering the worst-case scenarios for the target cell, row, and column. For system-level implementation, we used the VGG-8 architecture to classify 32×32 color images from the CIFAR-10 dataset [34], [35]. This architecture consists of six convolutional layers followed by two fully connected layers to perform the classification. Each convolution layer is followed by a max pooling layer. The DNN + Neurosim V2.0 framework is utilized to evaluate CIM inference and training performance, which facilitates software/hardware co-design [26].

B. Circuit-Level Evaluation

In SOT-MRAM, the switching delay depends upon the amount of current flowing through the bit-cell, as the current increases a shorter switching delay is obtained. The cell is symmetric, so for writing both 1 and 0, the same delay is obtained. Fig. 7 (a) shows the energy consumption for various current overdrive values (I_e/I_c, where I_e is the effective current value at which energy consumption is minimum (159 μA) and I_c is the critical current (128 μA). As illustrated, the energy consumption is minimized for a current overdrive factor of 1.24 value. Using the above I_e value, the number

TABLE I
SOT-MRAM PARAMETERS

Parameters	Value
Saturation Magnetization	1×10^6 A/m
Damping Factor	0.08
Spin Hall Angle	0.21
Critical Current	128 μA
Free Layer Dimensions	150 nm × 150 nm × 0.6 nm
Heavy Metal Dimensions	200 nm × 150 nm × 3 nm

TABLE II
PERFORMANCE OF VARIOUS COMPONENTS AND OPERATIONS FOR STANDARD AND PROPOSED (SIX WRITE AND SIX READ) CIM ARCHITECTURE USING 128 × 128 ARRAY

	Standard SOT-CIM		Proposed SOT-CIM	
Components	Latency (s)	Energy (J)	Latency (s)	Energy (J)
Accumulation	5.32	0.22	4.31	0.15
WG Computation	17.63	1.89	10.08	0.62
Feed-Forward	17.79	0.53	13.61	0.30
Error in BP	14.89	0.77	10.63	0.48
Weight Gradient	24.50	21.39	15.01	15.25
Weight Update	0.72	0.015	0.02	0.004

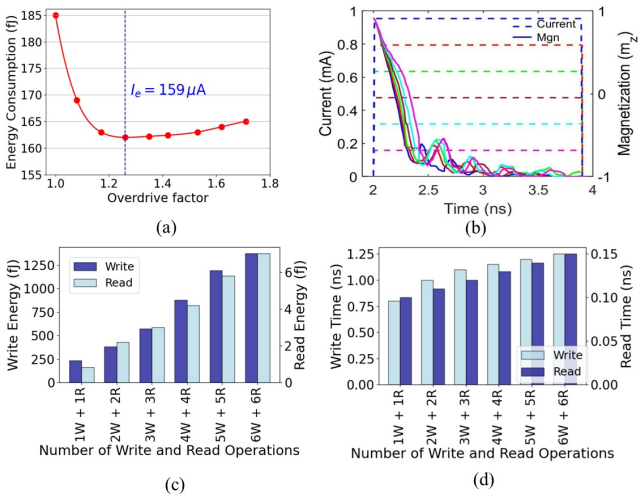


Fig. 7. (a) Energy consumption versus overdrive factor for a single bit-cell, (b) Illustration of bulk-write: current flow and switching behavior, (c) Write and read energy (d) Write and Read timings for multi-port bit-cell

of MRAMs that can be written in an energy-efficient way is calculated to be six, considering the maximum current the technology can handle is 1 mA. Fig. 7 (b) illustrates the current and switching behavior of six SOT-MRAM cells, with a maximum current of 953 µA required to ensure successful switching of all six MTJs. For the read operation, six ports were chosen because adding another port results in incorrect read values due to process variations and reduced read margins, making it unreliable for data retrieval. Additionally, Fig. 7 (c) and (d) show the performance of the proposed approach in terms of energy and time measurements for writing to the cell and reading from the cell using standard one write + one read and proposed six write + six read operations, respectively.

C. Crossbar Sub-System Evaluation

The synaptic crossbar array of size 128 × 128 has been implemented on the Cadence Virtuoso for each layer separately. Each column's output is fed to an output neuron. The results in terms of latency and energy consumption are measured over 256 epochs having 13M parameters and 0.7M operations. Table II shows the detailed hardware estimation results for latency, energy, and area consumption by main components and operations (feed-forward, error in backpropagation, weight gradient, and weight update) in the CIM accelerator. Fig. 8 (a) and (b) presents the energy and latency consumption for various components respectively, such as ADC, accumulation, weight update, and gradient calculation. Latency and energy consumption are mainly contributed by feed-forward, error computation, gradient computation, and weight update. Among these, weight gradient computation dominates total energy and latency due to the multiple read and write operations involving error and activation functions to compute gradients. It is observed that weight updates utilize less computation energy and latency compared to other computations because they are performed only once per batch.

Authorized licensed use limited to: TU Delft Library. Downloaded on March 12,2026 at 12:20:13 UTC from IEEE Xplore. Restrictions apply.

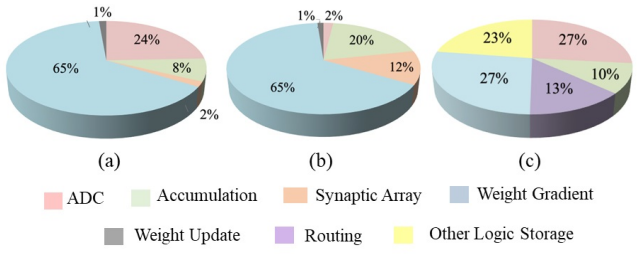


Fig. 8. Breakdown results of the proposed circuit for (a) Energy, (b) Latency, and (c) Area by main components for 128 × 128 array.

Fig. 8 (c) shows the area utilized by the crossbar array, routing, ADCs, accumulation module, weight gradient storage array, and other peripheral circuitry. The ADC and weight gradient computation occupy a large area because these units are made up of conventional CMOS technology and are separate units from CIM arrays. Other logic and storage units occupy comparable areas for data transfer. The accumulation logic includes processing elements, adder trees, and gradient units.

Fig. 9 provides the system-level comparison of latency, area, energy, and throughput for reading and writing parallelly. The latency is decreased as multiple cells are accessed in parallel simultaneously. The throughput is increased due to high parallelism, which allows simultaneous processing of multiple vector elements. This parallelism reduces the number of cycles needed for computations and thus decreases the overall energy per inference. With each additional port, more operations can be performed concurrently, resulting in lower energy consumption per operation. The benefits of a multi-port design come at the cost of increased area due to additional circuitry to manage the parallel access. It is observed that SOT-MRAM based multi-port neural networks are 5.82× and 3.20× more efficient in terms of latency and energy-efficiency, respectively compared to single port architecture. The drawback of a multi-port architecture is higher area consumption. The tile area of a multi-port configuration is increased by 2.67× compared to a single-port configuration.

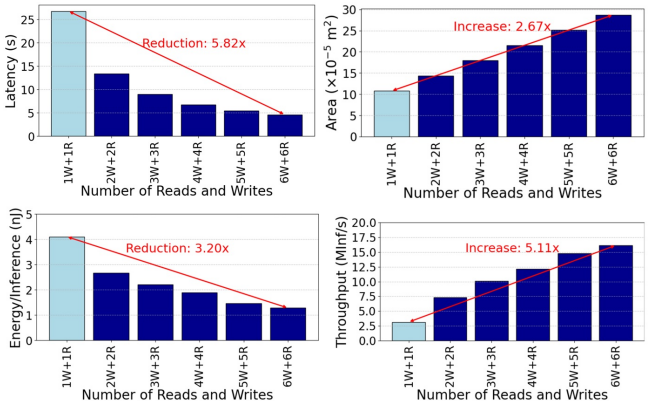


Fig. 9. System-level comparison between standard SOT cell (1R1W) and various multi-ported configuration cell designs. [R = Read, W = Write]

TABLE III
SYSTEM-LEVEL PERFORMANCE OF DIFFERENT CIM ARCHITECTURES FOR ON-CHIP LEARNING

	[26]	[36]	[12]*	Proposed
Technology (nm)	32	7	45	45
Device	SRAM	SOT	SOT	SOT
Area (mm^2)	138.95	10.80	31.67	50.79
Energy-Efficiency (TOPS/W)	-	46.10	20.28	47.15
Throughput (TOPS)	-	-	0.61	3.17
Training Energy-Efficiency (TOPS/W)	1.94	-	2.00	3.49
Training Throughput (TOPS)	0.78	-	0.85	2.04

* Values extracted from DNN + Neurosim V2.0 using bit-cell design in [12]

D. Comparison with State-of-the-Art

Table III compares the system-level performance of different CIM architectures for *Deep Neural Network* (DNN). The area occupied by the proposed architecture increases to 50.79 mm² due to the addition of multiple ports. Despite the larger area, the proposed work achieves energy-efficiency of 47.15 TOPS/W, closely matching the 7 nm technology of 46.10 TOPS/W due to the increased parallelism enabled by multi-port design, which enhances throughput and reduces energy per operation. Notably, the throughput is significantly increased in the proposed work, with a value of 3.17 TOPS, much larger than the 0.61 TOPS evaluated in standard SOT-MRAM. The training energy-efficiency of the proposed SOT-MRAM based CIM architecture is improved to 3.49 TOPS/W compared to 2.00 TOPS/W for standard SOT-MRAM bit-cell. The proposed architecture significantly increases training throughput to 2.04 TOPS, compared to 0.78 TOPS in SRAM and 0.85 TOPS in standard SOT-MRAM.

V. CONCLUSION

This paper presents a novel SOT-MRAM based CIM architecture aimed at enhancing the on-chip learning and inference process. The architecture utilizes multi-port memory to significantly boost parallelism, along with a control circuit that optimizes input requests to maximize port utilization and handles the maximum number of inputs. The results demonstrate a substantial reduction in latency by 5.82× and an increase in energy-efficiency by 3.20× compared to standard single-port designs. These findings highlight the potential of the proposed architecture to significantly improve latency, and increase energy-efficiency for on-chip learning in AI and machine learning applications.

REFERENCES

[1] A. Gholami, Z. Yao, S. Kim, C. Hooper, M. W. Mahoney, and K. Keutzer, "AI and memory wall," IEEE Micro, vol. 44, no. 3, pp. 33-39, 2024.

[2] A. Sehgal, S. Dhull, S. Roy, and B. K. Kaushik, "Advancements in memory technologies for artificial synapses," Journal of Materials Chemistry C, vol. 12, pp. 5274-5298, 2024.

[3] K. Roy, I. Chakraborty, M. Ali, A. Ankit, and A. Agrawal, "In-memory computing in emerging memory technologies for machine learning: An overview," ACM/IEEE Design Automation Conference (DAC), pp. 1-6, 2020.

[4] F. H. Meng and W. D. Lu, "Compute-in-memory technologies for deep learning acceleration," IEEE Nanotechnology Magazine, vol. 18, no. 1, pp. 44-52, 2024.

Authorized licensed use limited to: TU Delft Library. Downloaded on March 12,2026 at 12:20:13 UTC from IEEE Xplore. Restrictions apply.

[5] F. Oboril, R. Bishnoi, M. Ebrahimi, and M. B. Tahoori, "Evaluation of hybrid memory technologies using SOT-MRAM for on-chip cache hierarchy," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 34, no. 3, pp. 367-380, 2015.

[6] S. Diware, S. Dash, A. Gebregiorgis, R. V. Joshi, C. Strydis, S. Hamdioui, and R. Bishnoi, "Severity-based hierarchical ECG classification using neural networks", IEEE Transactions on Biomedical Circuits and Systems (TBioCAS) , vol. 17, pp. 77-91, 2023.

[7] S. Diware, A. Singh, A. Gebregiorgis, R. V. Joshi, S. Hamdioui, and R. Bishnoi, "Accurate and energy-efficient bit-slicing for RRAM-based neural networks", IEEE Transactions on Emerging Topics in Computational Intelligence (TETCI), vol. 7, pp. 164-177, 2022.

[8] V. D. Nguyen, S. Rao, K. Wostyn, and S. Couet, "Recent progress in spin-orbit torque magnetic random-access memory," NPJ Spintronics, vol. 2, no. 1, pp. 48, 2024.

[9] D. C. Worledge and G. Hu, "Spin-transfer torque magnetoresistive random access memory technology status and future directions," Nature Reviews Electrical Engineering, vol. 1, pp. 730-747, 2024.

[10] G. Prenat, K. Jabeur, P. Vanhauwaert, G. D. Pendina, F. Oboril, and R. Bishnoi, "Ultra-fast and high-reliability SOT-MRAM: From cache replacement to normally-off computing," IEEE Transactions on Multi-Scale Computing Systems, vol. 2, no. 1, pp. 49-60, 2016.

[11] R. Bishnoi, F. Oboril, and M. B. Tahoori, "Non-volatile non-shadow flip-flop using spin orbit torque for efficient normally-off computing," 21st Asia and South Pacific Design Automation Conference (ASP-DAC), Macao, China, pp. 769-774, 2016.

[12] Y. Luo, P. Kumar, Y. C. Liao, W. Hwang, F. Xue, W. Tsai, S. X. Wang, A. Naeemi, and S. Yu, "Performance benchmarking of spin-orbit torque magnetic RAM (SOT-MRAM) for deep neural network accelerators," Design and Applications of Emerging Computer Systems, pp. 67-89, 2023.

[13] Q. Liu, B. Gao, P. Yao, D. Wu, J. Chen, Y. Pang, W. Zhang, Y. Liao, C. Xue, W. Chen, J. Tang, Y. Wang, M. Chang, H. Qian, and H. Wu, "33.2 A fully integrated analog ReRAM based 78.4 TOPS/W compute-in-memory chip with fully parallel MAC computing," IEEE International Solid-State Circuits Conference (ISSCC), pp. 500-502, 2020.

[14] A. Singh, R. Bishnoi, R. V. Joshi, and S. Hamdioui, "Referencing-in-array scheme for RRAM-based CIM architecture," Design, Automation and Test in Europe Conference and Exhibition (DATE), pp. 1413-1418, 2022.

[15] J. Wu, X. Huang, L. Yang, J. Wang, B. Liu, Z. Wen, J. Li, G. Yu, K. S. Chong, and C. Wang, "An energy-efficient deep belief network processor based on heterogeneous multi-core architecture with transposable memory and on-chip learning," IEEE Journal on Emerging and Selected Topics in Circuits and Systems, vol. 11, no. 4, 2021.

[16] L. Huijbregts, H. H. Liu, P. Detterer, S. Hamdioui, A. Yousefzadeh, and R. Bishnoi, "Energy-efficient SNN architecture using 3nm FinFET multiport SRAM-based CIM with online learning," Proceedings of the 61st ACM/IEEE Design Automation Conference, pp. 1-6, 2024.

[17] S. Yu, H. Jiang, S. Huang, X. Peng, and A. Lu, "Compute-in-memory chips for deep learning: Recent trends and prospects," IEEE Circuits and Systems Magazine, vol. 21, no. 3, pp. 31-56, 2021.

[18] S. Diware, K. Chilakala, R. V. Joshi, S. Hamdioui, and R. Bishnoi, "Reliable and energy-efficient diabetic retinopathy screening using memristor-based neural networks", IEEE Access, vol. 12, pp. 47469-47482, 2024.

[19] S. Diware, A. Gebregiorgis, R. V. Joshi, S. Hamdioui, and R. Bishnoi, "Mapping-aware biased training for accurate memristor-based neural networks", IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS), pp. 1-5, Hangzhou China, 2023.

[20] M. Ali, S. Roy, U. Saxena, T. Sharma, A. Raghunathan, and K. Roy, "Compute-in-memory technologies and architectures for deep learning workloads," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 30, no. 11, 2022.

[21] Y. Kim, X. Fong, K.-W. Kwon, M.-C. Chen, and K. Roy, "Multilevel spin-orbit torque MRAMs," IEEE Transactions on Electron Devices, vol. 62, no. 2, pp. 561–568, 2015.

[22] Z. He, S. Angizi, F. Parveen, and D. Fan, "High performance and energy-efficient in-memory computing architecture based on SOT-MRAM," IEEE/ACM International Symposium on Nanoscale Architectures (NANOARCH), pp. 97-102, 2017.

[23] L. Liu, C. F. Pai, Y. Li, H. W. Tseng, D. C. Ralph, and R. A. Buhrman, "Spin-torque switching with the giant spin hall effect of tantalum," Science, vol. 336, no. 6081, pp. 555-558, 2012.

[24] T. Endoh, H. Honjo, K. Nishioka, and S. Ikeda, "Recent progresses in STT-MRAM and SOT-MRAM for next generation MRAM," IEEE Symposium on VLSI Technology, pp. 1-2, 2020.

[25] I. M. Miron, K. Garello, G. Gaudin, P. J. Zermatten, M. V. Costache, S. Auffret, S. Bandiera, B. Rodmacq, A. Schuhl, and P. Gambardella,"Perpendicular switching of a single ferromagnetic layer induced by in-plane current injection," Nature, vol. 476, no. 7359, pp. 189–193, 2011.

[26] X. Peng, S. Huang, H. Jiang, A. Lu, and S. Yu, "DNN+Neurosim V2.0: An end-to-end benchmarking framework for compute-in-memory accelerators for on-chip training," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 40, no. 11, pp. 2306-2319, 2021.

[27] J. Bilski, B. Kowalczyk, A. Marjański, M. Gandor, and J. Zurada, "A novel fast feedforward neural networks training algorithm," Journal of Artificial Intelligence and Soft Computing Research, vol. 11, no. 4, pp. 287-306, 2021.

[28] Z. Zhang, X. Liu, H. Zhou, S. Xu, and C. Lee, "Advances in machine-learning enhanced nanosensors: From cloud artificial intelligence toward future edge computing at chip level," Small Structures, vol. 5, pp. 2300325, 2023.

[29] N. K. Upadhyay, H. Jiang, Z. Wang, S. Asapu, Q. Xia, and J. J. Yang, "Emerging memory devices for neuromorphic computing," Advanced Materials Technologies, vol. 4, pp. 1800589, 2019.

[30] Y. Kim, S. K. Gupta, S. P. Park, G. Panagopoulos, and K. Roy, "Write-optimized reliable design of STT MRAM," ACM/ IEEE International Symposium on Low Power Electronics and Design, pp. 3-8, 2012.

[31] R. Bishnoi, F. Oboril, and M. B. Tahoori, "Low-power multi-port memory architecture based on spin-orbit torque magnetic devices," Great Lakes Symposium on VLSI, pp. 409-414, 2016.

[32] H. Jiang, W. Li, S. Huang, S. Cosemans, F. Catthoor, and S. Yu, "Analog-to-digital converter design exploration for compute-in-memory accelerators," IEEE Design and Test, vol. 39, no. 2, pp. 48-55, 2022.

[33] K. Kazemi, G. E. Rowlands, E. Ipek, R. A. Buhrman, and E. G. Friedman, "Compact model for spin–orbit magnetic tunnel junctions," IEEE Transactions on Electron Devices, vol. 63, no. 2, pp. 848-855, 2016.

[34] K. Simonyan, and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," ArXiv Preprint, arXiv:1409.1556, 2014.

[35] R. Doon, T. Kumar Rawat, and S. Gautam, "Cifar-10 classification using deep convolutional neural network," IEEE Punecon, pp. 1-5, 2018.

[36] J. Doevenspeck, K. Garello, B. Verhoef, R. Degraeve, S. Van Beek, D. Crotti, F. Yasin, S. Couet, G. Jayakumar, I. A. Papistas, P. Debacker, R. Lauwereins, W. Dehaene, G.S. Kar, S. Cosemans, A. Mallik, and D. Verkest, "SOT-MRAM based analog in-memory computing for DNN inference," IEEE Symposium on VLSI Technology, pp. 1-2, 2020.

Authorized licensed use limited to: TU Delft Library. Downloaded on March 12,2026 at 12:20:13 UTC from IEEE Xplore. Restrictions apply.