

MIRAGE: MRAM-Based Near ADC-Less Compute-In-Memory Macro for Deep Learning Acceleration

Mainakh Mukherjee, Ayan B. Pranta, Utkarsh Saxena, Anushka Mukherjee,
Deepika Sharma, Gaurav Kumar K. and Kaushik Roy

School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, USA.

Abstract—Non-volatile memory (NVM) based Compute-in-Memory (CiM) architectures have emerged as a promising compute primitive for accelerating deep neural networks (DNNs) by performing in-situ matrix–vector multiplications (MVMs). Among various NVMs, STT-MRAM (Spin Transfer Torque based Magnetoresistive Random Access Memory) shows potential due to its high endurance, low energy consumption and high density. However, existing STT-MRAM CiM designs typically rely on multi-bit analog-to-digital converters (ADCs) at the peripherals to digitize accumulated bit-line currents. While enabling high-precision computation, ADCs add substantial energy, latency, and area overheads. To alleviate such problems, we propose a system-technology co-design approach to a Near ADC-Less CiM design with ternary partial-sums called MIRAGE. The accuracy is maintained by considering hardware level partial sum quantization in the training loop. Specifically, we develop an STT-MRAM based CiM macro which features differential bitcells and an adaptive threshold sensing that is amenable to the requirements posed by ternary partial-sum quantization. We do a thorough energy, area, latency, and sense margin analysis along with robust benchmarking against conventional 1T-1MTJ (1 transistor-1 Magnetoresistive Tunnel Junction) based MRAM CiM. The proposed CiM macro occupies $\sim 20\%$ less area, consumes $1.8\times$ less MVM energy and shows $5\times$ better latency with improved distinguishability compared to 1T-1MTJ CiM macro while achieving better accuracy.

Index Terms—Compute-In-Memory, STT-MRAM, Analog-To-Digital Converter, Near ADC-less

I. INTRODUCTION

The relentless push for better AI models has driven the development of more complex neural network architectures for vision, speech, language, and recommendation workloads [1]. The continuous increase in the size of models calls for energy-efficient but powerful hardware for edge applications. However, meeting this demand at the edge is difficult due to strict constraints on power consumption and chip area. Additionally, as model capacity grows, the cost of data movement between memory and compute dominates the cost of processing them. The “memory wall” throttles both throughput and energy efficiency, drastically affecting performance [2]–[4]. Compute-in-Memory (CiM) has emerged as a promising solution, executing matrix–vector multiplications (MVMs), the core computing requirements of AI workloads, within the memory itself, thereby alleviating the memory wall [5]. CiM based memory parallel architectures enable fast and efficient MVMs, essential for DNN (Deep Neural Network) processing [6]. The memory

²In this paper we use “ADC-Less” to refer to the use of sense amplifiers. Note, however, a sense amplifier is effectively a 1-bit ADC

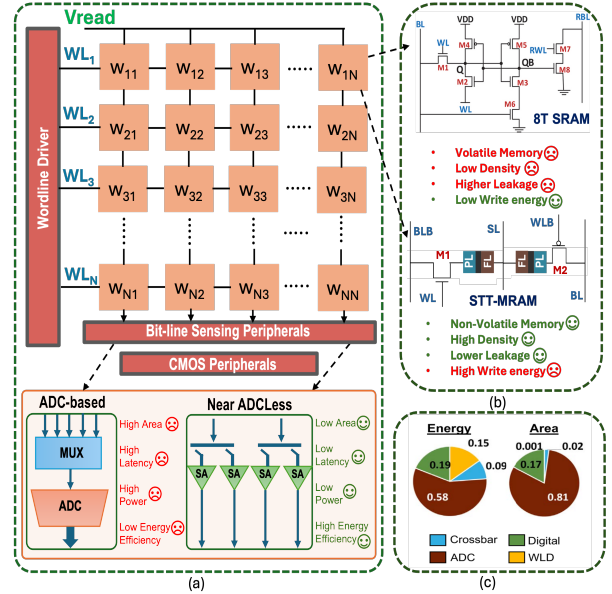


Fig. 1. (a) Overview of Near ADC-Less² CiM design with ternary partial sums requiring two sense amplifiers for the conversions. This alleviates the ADC bottleneck, leading to improvements in hardware performance compared to ADC based design. (b) 8T-SRAM vs STT-MRAM, where STT-MRAM shows promising features of high density, lower leakage and non-volatility (c) Area and energy breakdown in ADC-based CiM accelerators

technology used in CiM impacts the system and algorithm level performance. While Static Random Access Memory (SRAM) based CiM hardware [7], [8] is appealing due to its process maturity, SRAM based analog CiM incurs substantial leakage and density penalties at scale [9]–[12]. To that effect, STT-MRAM has emerged as a promising alternative for analog CiM [13] offering non-volatility (improved leakage), CMOS compatibility, high-endurance, and a device stack that enables compact arrays (Fig. 1(b)).

Despite the energy efficiency of analog CiM, its deployment at the system level is limited by the complexity of ADCs (Fig. 1(c)). Maximizing MVM parallelism ideally requires one ADC per column (bit-line) of the crossbar array. However, ADCs are large and hence, to save area, fewer multiplexed ADCs are used per crossbar array, leading to lower throughput [14]–[16]. The required ADC precision is set by the column partial-sum precision, which depends on (i) the number of simultaneously active rows and (ii) cell-level read precision (sense margin, process variability). Note, reduced ADC resolution reduces power/area/latency but is usually associated

with fewer rows (word-lines) being simultaneously activated, sacrificing parallelism. Conversely, activating multiple rows enlarges the dynamic range, but requires higher ADC resolution, making readout the bottleneck [6], [17]. Practical arrays also suffer from IR drop on bit-lines and process/device variations [18], further elevating ADC cost. Prior efforts reduced ADC precision via input sparsity [19], [20] or partial-sum quantization (PSQ) [21]–[23]. However, hardware-only methods hit accuracy limits. Going further, recent partial-sum quantization frameworks aim to reduce the precision of ADCs by using a quantization-aware training (QAT) [24]–[27]. Among them, [27] shows that partial sums can be quantized to binary (1,-1) or ternary values (-1,0,+1) with minor impact on workload accuracy. However, such an approach requires proper co-design of hardware to reap the benefits.

To address such issues, we adopt a system–technology co-design (STCO) approach. The compute primitive requires new design choices to enable ternary quantization. This motivates novel bitcell and peripheral-circuit designs so that circuit-level improvements in MRAM cells translate to system-level efficiency on inference workloads.

Accordingly, we propose a near-ADC-less, STT-MRAM-based analog CiM macro. We develop an algorithm-aware hardware design, integrating ternary partial sum quantization, PSQ [27], in the training loop to mitigate accuracy degradation. This approach poses two critical demands: (i) handling signed weights within the same crossbar and (ii) providing fine-grained, programmable thresholds. We meet both requirements with a differential bitcell and an MTJ-based sense amplifier (SA) that supports adaptive thresholds.

In this context, we make the following contributions:

- 1) We introduce a near ADC-less STT-MRAM CiM macro called MIRAGE, developed through STCO where partial sums of crossbar arrays are ternary quantized. Such quantization is leveraged to design a novel differential bitcell and MTJ-based sense-amplifiers as peripherals. The design eliminates the need for full precision ADCs, leading to high column level parallelism for MVM operations with improved latency and energy efficiency.
- 2) The proposed STT-MRAM based differential pull-up pull-down (PUPD) bitcell encodes (-1, 0, 1) suitable for ternary quantization of partial sums. We implemented the CiM array along with peripherals in GF 22nm CMOS FDX as the front-end process and MTJ (Tunneling Magnetoresistance, TMR of $\approx 120\%$), at the back-end.
- 3) Comprehensive energy, area, latency, sense-margin analysis and benchmarking against a conventional 1T-1MTJ MRAM CiM macro show $\sim 20\%$ lower macro area, $1.8\times$ lower MVM energy consumption, and $5\times$ better latency with improved distinguishability. System-level evaluations on RESNET-18 (IMAGENET) and RESNET-20 (CIFAR-10) yield $1.7\text{--}1.8\times$ and $4.4\text{--}4.8\times$ improved energy & latency, and up to 2% higher accuracy under noisy conditions.

The remainder of the paper is organized as follows. Section II provides necessary background information. Section III presents a review of related works. Section IV describes our base-line 1T-1MTJ framework. Section V details our proposed differential STT-MRAM based Near ADC-less hardware. Section VI discusses the results & analysis. We conclude in Section VII.

II. BACKGROUND

A. Analog Crossbar Array Based Matrix Multiplication

MVM in DNNs can be accelerated with analog MRAM crossbars [28]. Weights are stored as MTJ (parallel or anti-parallel) resistive states, activations are applied as row voltages via Digital-to-Analog Converters (DACs), and by Ohm’s law ($I=GV$) each cell produces a current that sums along bit-lines (BL) to form partial sums, which are then digitized by ADCs. Usually 1-bit DACs are used with input streaming. Along with weight slicing (multiple crossbars are used to implement the weights), processing over multiple cycles leads to higher precision MVMs [6], [29]. However, reducing ADC precision below the theoretical bound introduces errors.

B. STT-MRAM

MRAM is a high-density eNVM [30], offering high endurance, integrability with CMOS VLSI process, and fast read and write speed compared to other eNVM technologies. A standard MRAM bitcell comprises an access transistor and an MTJ [31]. An MTJ consists of three layers: a pinned magnetic layer (PL), a free magnetic layer (FL), and an oxide barrier between them. The MTJ stores bits based on the magnetic orientation of the FL with respect to PL. If the magnetization aligns, the device is considered to offer low resistance (R_P) otherwise high resistance (R_{AP}).

C. Non-Ideality and Low MTJ TMR

Non-idealities in MRAM-based CiM hardware degrade compute reliability and reduce inference accuracy. In addition to parasitic effects such as wire, driver, and sink resistances causing IR-drop along bit-lines (BL) and source-lines (SL), the low TMR of MTJs further compresses sense margins [33]–[36]. This poor distinguishability between resistance states leads to deviations in output currents, which propagate through the sense path and corrupt partial sums. These errors accumulate across columns and, depending on crossbar size, device scaling, and bit-slice/stream precision, can significantly degrade overall DNN inference accuracy [18], [37]–[39].

III. RELATED WORKS

Although existing MRAM-based CiM accelerators for AI workloads achieve high throughput and energy efficiency, yet they rely on per-column multi-bit ADCs/IDCs (Current to Digital Converter). The authors in [40] present a 22 nm, 128-Kb MRAM CiM with row/column-parallel differential readout, where multi-bit converters dominate energy and area. A 28 nm, 2 Mb STT-MRAM macro $22.4\text{--}41.5$ TOPS/W macro presented in [28] uses a refined differential bitcell, but accurate multi-bit ADCs remain essential in the readout chain. To address

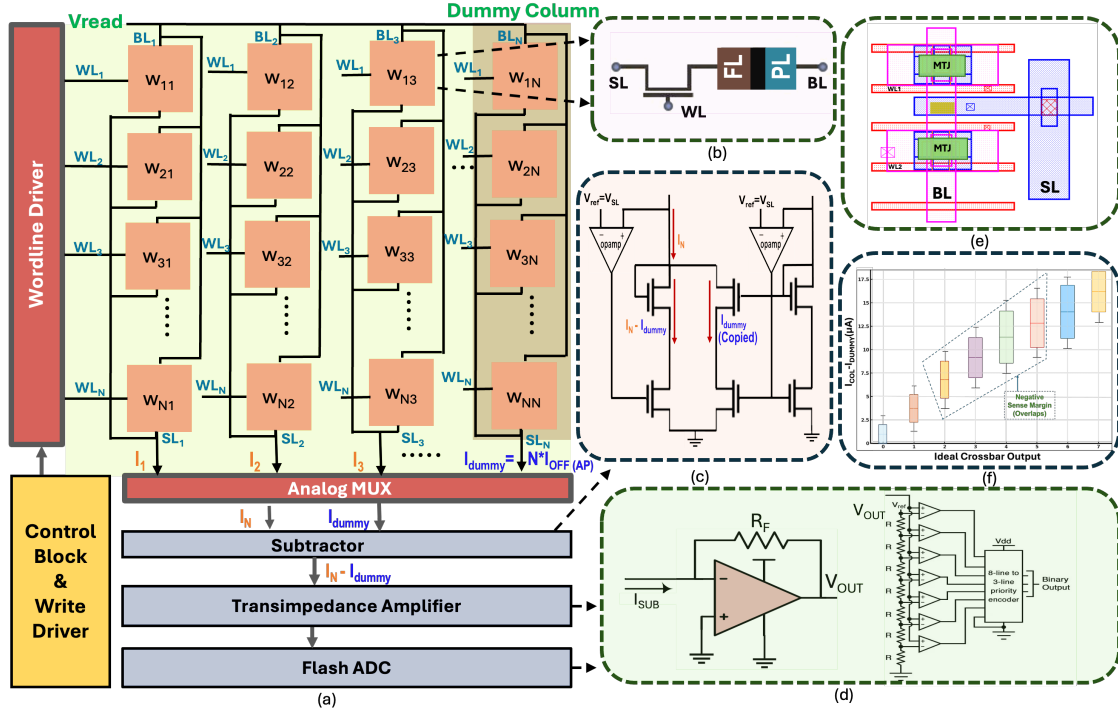


Fig. 2. (a) Circuit diagram of the baseline 1T-1MTJ MRAM CiM Macro (b) 1T-1MTJ bitcell (c) An analog subtractor circuit to reduce the contribution of leakage [32] from the inactive rows. (d) Transimpedance Amplifier (TIA) to convert the subtracted current to voltage and 3-bit Flash ADC to digitize it (8 partial wordlines are activated, PWA-8) (e) Layout of 2 connected 1T-1MTJ cells with the transistors at the front end of the line (FEOL, metal 1 to metal 3), and the MTJs at the back end (BEOL, metal 3 onwards). (f) Sense Margin plot of 64x64 crossbar with multiple weight combinations

this issue, authors in [27] replace multi-bit ADCs with 1-bit sense amplifiers with quantization aware training, reducing conversion energy by up to 9.6x with less than 1% accuracy loss. The authors in [7] extends the above work where compact digital CiM units apply QAT-learned scale factors, further reducing conversion cost while preserving accuracy.

IV. 1T-1MTJ STT-MRAM FRAMEWORK

A. 1T-1MTJ Bitcell & 64x64 Crossbar with Peripherals

Fig. 2(b) shows a standard 1T-1MTJ bitcell with the MTJ placed on the NMOS drain. The MTJ has $R_P=2.8\text{ k}\Omega$ and $R_{AP}=6.17\text{ k}\Omega$ ($\text{TMR}\approx 120\%$), storing a binary weight (0, 1). A 64×64 crossbar is designed using this 1T-1MTJ bitcell. A dummy column (all cells in the AP state) is included, and its current is subtracted column-wise to cancel off-current/leakage contributions [32]. We use partial-wordline activation of 8 (PWA-8) and evaluate one column at a time. An analog multiplexer (MUX) selects the column, and the dummy-column current (Fig. 2(c)) is subtracted from the active-column current (current mirrors are used to copy the dummy column current). The Transimpedance Amplifier (TIA) converts the subtracted current to voltage, which is then digitized by a 3-bit flash ADC (Fig. 2(d)). With 1 column reserved for dummy, the remaining 63 columns are evaluated in 63×8 cycles.

B. Circuit Simulations & Sense Margin Analysis

We designed a 64×64 crossbar (PWA-8: eight MVM rows per cycle with per-column operation) in GF22nm CMOS FDX including all read peripherals (Fig. 2(a)). During read operation,

$V_{\text{read}} = 0.4\text{ V}$ is applied to the BL while the SL is held at 0.3 V when the WLs are activated. We perform row-wise bidirectional writes using write drivers at both ends of each column to pass current from FL to PL and vice-versa. Fig. 2(e) shows the layout of two connected 1T-1MTJ cells. We perform post-layout parasitic extraction to observe crossbar non-idealities and assess their impact on sense margin. We conduct sense-margin analysis across multiple input-weight combinations, and the resulting read-current distributions exhibit substantial overlap between states, including negative sense margins (Fig. 2(f)), indicating vulnerability to read errors. Circuit-level simulations are performed in Cadence® and Spectre®.

C. Issues with 1T-1MTJ MRAM Architecture

- Distinguishability & variation: Low TMR and device variation reduces sense margin. Subtraction removes leakage but not mismatch/IR-drop induced spread, increasing errors.
- Throughput bound: One ADC multiplexed between all columns along with PWA-8 serializes computation (63×8 conversions), reducing array-level parallelism.
- ADC/TIA dominance: Multi-bit conversion and the TIA dominate energy, area, and latency.
- Area/routing overhead: Dummy column, subtractor network, TIA, muxing, and ADCs consume peripheral area and complicate layout/power-grid design.

Our Near ADC-less MRAM design with differential bitcell, multi-threshold sense amplifiers, and QAT directly mitigates these bottlenecks; eliminating per-column ADC/TIA overhead,

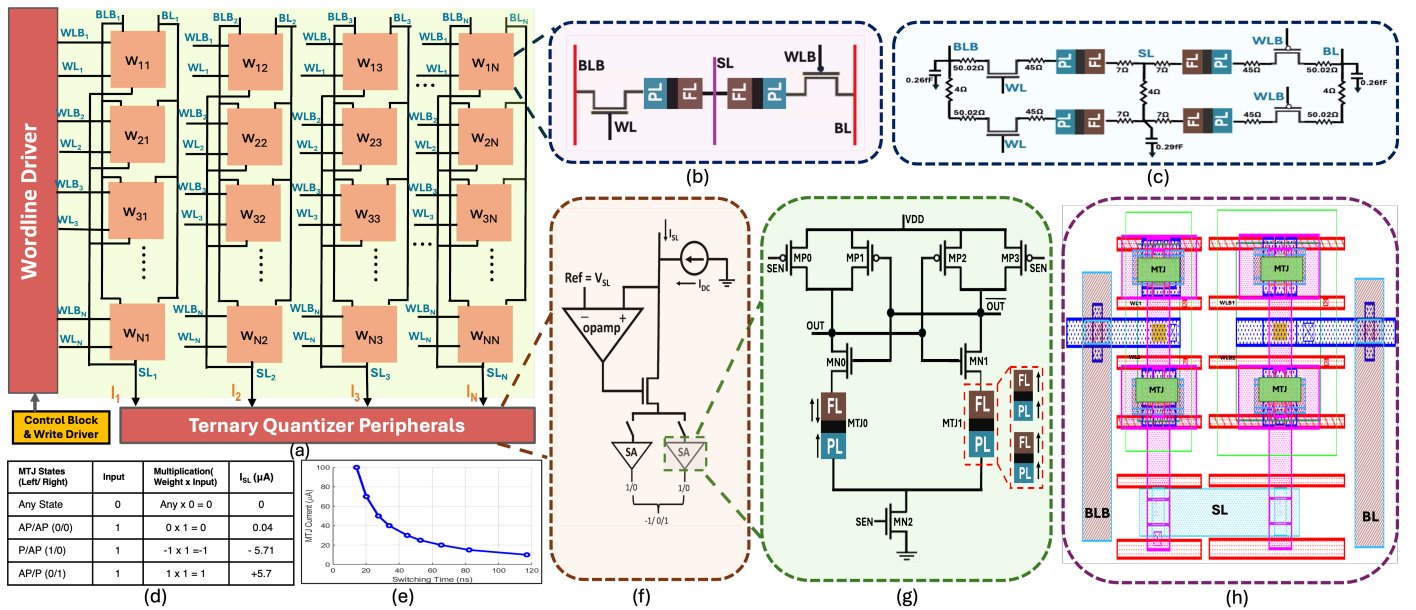


Fig. 3. (a) Near ADC-less macro (b) Proposed pull-up pull-down (PUPD) bitcell (c) Extracted parasitics from the layout (2 connected PUPD Bitcells) (d) Multiplication output from bitcell according to programmed MTJ states and input (e) Magnitude of required current to flip MTJ vs pulsewidth of current (f) Ternary quantizer peripherals (fixing SL) (g) Pre-Charge Sense Amplifiers (PCSA) [MTJ1, the reference MTJ, consists of two MTJs connected in series, each MTJ being in parallel configuration to make the effective resistance between R_P and R_{AP}] (h) Layout of 2 connected PUPD (2T-2MTJ) Bitcells with the transistors at the front end of the line (FEOL, metal 1 to metal 3), and the MTJs at the back end (BEOL, metal 3 onwards).

improving sense margin and parallelism, and reducing energy, area & latency while maintaining accuracy.

V. PROPOSED PULL-UP PULL-DOWN STT-MRAM HARDWARE

A. Partial-Sum Quantization (PSQ)

In ternary PSQ proposed in [27], the column partial sum ps is quantized to $\{-1, 0, +1\}$ using two thresholds, eliminating the need for multi-bit ADCs. A learned scale α and two symmetric thresholds (scale factors) implement the quantizer:

$$ps_t = \alpha \cdot \begin{cases} +1, & ps \geq \alpha/2, \\ 0, & -\alpha/2 < ps < \alpha/2, \\ -1, & ps \leq -\alpha/2. \end{cases}$$

The thresholds are trained along with neural network weights so that the ternary outputs align with the floating-point dynamic range. The authors in [27] adopt CiM oriented quantization-aware training, where both weights and activations are quantized. This co-design enables one to achieve comparable accuracy to the floating-point baseline with only ternary partial sums. While they propose a PSQ training methodology, no real hardware implementation is shown. However, in this work, we design our hardware to implement the algorithm. **First**, the PSQ algorithm requires a very fine-grained quantization scheme. The authors observe that reducing the granularity of thresholds impacts final accuracy by more than 5%. To that effect, we design our sense amplifier to support such fine-grained thresholds. **Second**, they show that storing the signed weights within the same array, i.e., performing subtraction of signed weights within the array, performs better than storing signed weights in different crossbars. We therefore adopt in-array signed-weight

subtraction in our design. Accordingly, we implement Near ADC-less PSQ in hardware using our PUPD MRAM bitcell, a 64×64 crossbar, and a column-end precharge sense amplifier (PCSA) based ternary quantizer. Column currents sum across the active rows to form ps , which is then compared to the two thresholds $\{-\alpha/2, \alpha/2\}$ using a pair of sensing decisions (e.g., PCSAs or comparators). The resulting code $\{-1, 0, +1\}$ is then scaled digitally by α . Inference, therefore reduces to applying row voltages, accumulating resistive currents ($I=GV$) along the column, performing two comparisons, and multiplying by a scalar value. This preserves column parallelism while removing the multi-bit ADCs and achieving low conversion energy and latency, while QAT maintains accuracy.

B. PUPD Bitcell

Fig. 3(b) shows a PUPD MRAM bitcell with the left MTJ on the NMOS drain and right MTJ on the PMOS drain, driven by word-line (WL) and its complement word-line-bar (WLB). We use STT-MTJs characterized by $R_P=2.8 \text{ k}\Omega$ and $R_{AP}=6.17 \text{ k}\Omega$ ($\text{TMR} \approx 120\%$). The cell realizes signed weights $w \in \{-1, 0, +1\}$ and performs inherent in-cell subtraction at SL by combining the left and right branch currents. The bitcell operation is summarized in Fig. 3(d). When the input bit is 0, $V_{WL} = 0$ and $V_{WLB} = V_{DD} = 0.8 \text{ V}$, so both transistors are off and no current flows through the SL (i.e., $w \times 0 = 0$). When the input bit is 1, $V_{WL} = V_{DD}$ and $V_{WLB} = 0$, turning both devices on; the resulting output current I_{SL} is set by the MTJ states. When both MTJs are AP, the left and right branch currents are equal and cancel at the SL node by Kirchhoff's current law, giving zero output current (i.e., $0 \times 1 = 0$). When the left MTJ is P and the right MTJ is AP, the left branch carries

more current than the right, producing an inward current at the SL node (i.e., $(-1) \times 1 = -1$). Conversely, when the left MTJ is AP and the right MTJ is P, there is an outward current at the SL node (i.e., $1 \times 1 = 1$). With subtraction performed inside the bitcell, no column-end analog subtractor or dummy column (as in the 1T-1MTJ architecture) is required, reducing area and energy. Fig. 3(c) shows the extracted RC network modeled from the layout of a 2×1 PUPD cell pair (Fig. 3(h)).

C. 64×64 Crossbar and Precharge Sense Amplifier

We design a 64×64 crossbar using the PUPD bitcells (Fig. 3(a)). A full MVM evaluation completes in 8 cycles, with 8 wordlines activated per cycle (incorporating PWA-8) and all columns sensed in parallel. Row-wise bidirectional writes use drivers at both ends of each column (BL, BLB, and SL) to pass current from FL to PL and vice versa. During read operation, $V_{\text{read}} = 0.35$ V is applied to the BL & $V_{\text{read}} = 0.45$ V is applied to the BLB with an operational amplifier holding the SL bias at $V_{\text{SL}} = V_{\text{DD}}/2$ (Fig. 3(f)). Each column includes two PCSAs (Fig. 3(g)) that implement the comparisons to $\{-\alpha/2, \alpha/2\}$. Thresholds are set by gating the switching MTJ's conduction window in each PCSA. Longer windows require lower current (lower threshold) while shorter windows require higher current (higher threshold), yielding symmetric decisions around zero (Fig. 3(e)). To keep the analog interface unipolar, a DC offset current I_{DC} ($I_{\text{DC}} = 50 \mu\text{A}$ for PWA-8) is added to the column current (Fig. 3(f)) so the PCSA input is $I_{\text{SL}} + I_{\text{DC}} \geq 0$ (In simulations we use a PMOS as a constant current source). Each cycle performs pre-charge/reset, asserts eight wordlines, time-multiplexes evaluation against the two thresholds within the same settling period, latches the ternary output and then applies the trained scale factor α digitally to the resulting code.

D. Circuit Simulation Framework

Circuit level simulations are done on Cadence[®] & Spectre[®] including bitcell-level parasitics extracted from the layouts, to observe array level non-idealities. Our design is based on GF22nm CMOS FDX and a compact model of MTJ. We analyse functionality of CiM under PWA-8, stability of the SL bias, and mapping from PCSA conduction windows to effective column current thresholds.

VI. RESULTS & DISCUSSION

A. Sense Margin Analysis

Fig. 4 shows the sense margin analysis of the PUPD MRAM crossbar including post-layout parasitics & Monte Carlo (MC) simulations. We simulated 1000 MC trials per output state for each column, modeling variations considering—(i) 25 mV standard deviation (σ) of transistor threshold voltage V_{th} , (ii) 1.5% standard deviation (σ) of MTJ oxide thickness, and (iii) 5% standard deviation (σ) of MTJ diameter. For each GV (input-weight) combination within a PWA-8 slice, the ideal partial sum maps to a distribution of column currents. Process variations and IR-drop broaden these distributions and create overlaps near the decision boundaries. To avoid multi-bit ADCs, we quantize the column partial sum with two symmetric thresholds

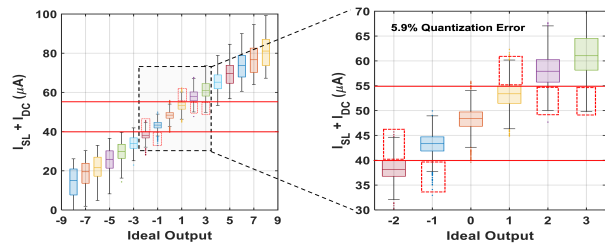


Fig. 4. Sense Margin plots of PUPD MRAM Crossbar

($\pm\alpha/2$). Ternary PSQ yields a 5.95% error concentrated near the thresholds, while large magnitude sums remain robust. QAT learns the per-layer/column scale α (and mild asymmetry), re-centering the current histograms relative to the ternary regions. As a result, accuracy is maintained while eliminating multi-bit conversion and its associated energy, area and latency overheads.

B. Circuit Simulations

TABLE I
ARRAY-LEVEL CHARACTERISTICS FOR MVM

Bitcell (22nm FDX)	Macro Area (mm ²)	MVM	
		Energy (pJ)	Latency (ns)
1T-1MTJ (with multi-bit ADC)	0.0142	1440	3200
Our Approach (MIRAGE)	0.0113	800	640

Table I summarizes MVM area, energy, and latency for the 1T-1MTJ (baseline) and the proposed PUPD MRAM CiM macro design. The 1T-1MTJ bitcell is smaller, but its peripherals (ADC, subtractor, TIA and WL drivers) dominate total area. In the PUPD macro, the ADC is replaced by sense amplifiers, yielding an estimated 20% smaller macro area despite the larger bitcell (as shown in Fig. 5).

With peripherals as the bottleneck, replacing ADCs with sense amplifiers directly reduces energy and latency. The baseline pays the cost of multi-bit conversion and ADC column multiplexing, which raises both MVM energy and conversion time. Our readout performs only two comparisons per column (at $\pm\alpha/2$) and applies a small digital scale, delivering $\sim 1.8 \times$ lower MVM energy and $\sim 5 \times$ lower MVM latency than 1T-1MTJ, with $\sim 9 \times$ improved Energy-Delay Product (EDP) while preserving column-level parallelism. Furthermore, when scaled to larger arrays, higher IR-drops and mismatches degrade the column current, but our approach, MIRAGE, mitigates this by in-cell differential subtraction using a 2T-2MTJ PUPD cell, unlike 1T-1MTJ, which performs subtraction at the column end with a dummy column that is not suitable under scaling because of the increase in variations.

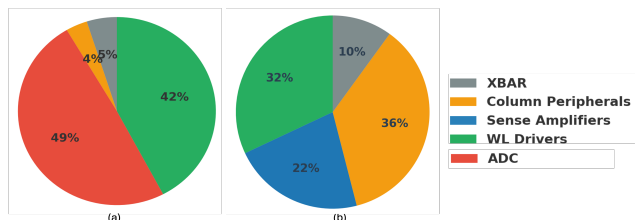


Fig. 5. Area breakdown of various components of (a) 1T-1MTJ CiM macro (Baseline) (b) PUPD MRAM CiM Macro (MIRAGE)

C. System Level Results

To evaluate system-level performance, we use PUMA [6]- a cycle-accurate simulator publicly available on Github. PUMA provides overall latency and energy overhead estimation for a CiM-based spatial architecture. At the topmost level we have a network-on-chip connecting multiple tiles and a global buffer for off-chip DRAM access. Each tile contains its own instruction memory, shared memory and multiple cores. Each core in turn contains multiple MVM units (MVMU), register file, Scalar-and-Vector functional units. We replace the the CiM macros residing in each of these MVMUs with our own 1T-1MTJ and PUPD macros. Table II shows the accelerator architecture parameters.

TABLE II
SPATIAL ACCELERATOR (PUMA) SPECIFICATIONS

Component	Parameter	Value
Global Buffer	Storage	256KB
Tile	# per node	8
Shared Memory	Storage	256KB
Core	# per tile	8
MVMU	# per core	8
ALU	ALU width	64
Register File (RF)	Storage	4KB
CiM Macro	1T-1MTJ	3bit ADC
Peripherals	PUPD	1bit Sense-Amplifier

We evaluate our 1T-1MTJ and PUPD macro performances by running inference on two CNN models: RESNET-18 on IMAGENET and RESNET-20 on CIFAR-10. Fig. 6 shows the normalized energy and latency results of PUPD macro against 1T-1MTJ macro. We observe $1.78-1.81\times$ energy improvement and $4.4-4.8\times$ latency improvement, which follows a similar trend to our macro-level improvement numbers. Fig. 7 shows the energy breakdown of the PUPD macro-included spatial architecture.

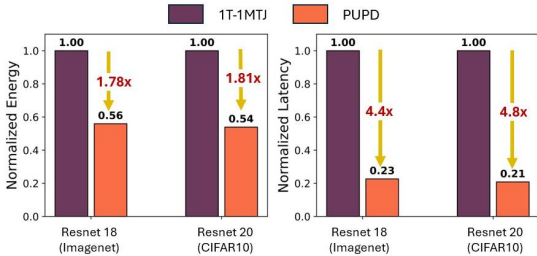


Fig. 6. Energy and latency measurements for RESNET-18 (IMAGENET) and RESNET-20 (CIFAR-10). The numbers are normalized against the 1T-1MTJ macro

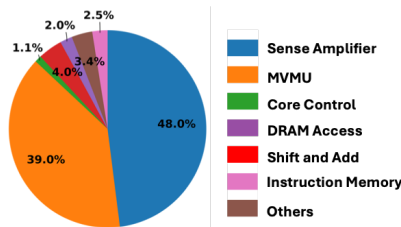


Fig. 7. Energy breakdown of spatial architecture with PUPD macro units

D. Accuracy Analysis

We evaluate workload accuracy under noisy conditions by injecting multiplicative Gaussian noise with unit mean and

standard deviation σ into the pre-quantizer partial sums. Let ps denote the partial sums before quantization. The perturbed partial sums are

$$ps_{\text{noise}} = ps \odot \varepsilon, \quad \varepsilon \sim \mathcal{N}(1, \sigma^2), \quad (1)$$

where \odot denotes elementwise multiplication. Using these noisy partial sums, we measure end-to-end top-1 accuracy on CIFAR-10 with RESNET-20 across multiple weight/activation precisions. Results for several noise levels are reported in Table III. Overall, ternary PSQ exhibits substantially improved noise robustness: as σ increases, accuracy degrades more gracefully than the baseline. In the noise-free regime ($\sigma = 0$), the baseline attains slightly higher accuracy due to the absence of partial-sum quantization. However, at higher noise levels, ternary PSQ maintains higher accuracy, indicating stronger resilience to multiplicative non-idealities.

TABLE III
ACCURACY (%) OF RESNET-20 UNDER NOISE WITH DIFFERENT STANDARD DEVIATIONS.

Weights/Activations	Method	Noise std (σ)			
		0	0.1	0.2	0.3
3/3	Ternary PSQ	90.3	90.2	89.9	85.4
	Baseline	90.6	90.8	89.7	83.4
4/4	Ternary PSQ	90.9	90.7	90.4	89.4
	Baseline	91.1	90.9	90.2	88.8

VII. CONCLUSION

We present MIRAGE, an STT-MRAM based analog CiM macro suitable for edge inference. The energy-efficiency and low-latency of MIRAGE is obtained by an algorithm hardware co-design methodology that quantizes the partial sums to ternary values during training so that power hungry multi-bit ADCs can be replaced by simple sense amplifiers (1-bit ADCs). In order to effectively utilize such simplified peripherals, the core STT-MRAM based CiM macro is designed using a novel differential PUPD bitcell using 2 transistors and 2 MTJs (2T-2MTJ). Compared to a conventional 1T-1MTJ CiM macro with full-precision ADC, MIRAGE shows $\sim 20\%$ lower area, $1.8\times$ lower MVM energy, and $5\times$ lower latency. The optimized design of MIRAGE MVM unit translates into system-level efficiency, achieving approximately $4.4-4.8\times$ better latency and $1.7-1.8\times$ lower energy consumption compared to 1T-1MTJ CiM baseline for RESNET-18 (IMAGENET) and RESNET-20 (CIFAR-10) workloads. We also achieve 2% better accuracy for ternary PSQ compared to the baseline 1T-1MTJ design (under noisy conditions).

ACKNOWLEDGMENT

This work was supported in part by CHEETA: CMOS+MRAM Hardware for Energy-Efficient AI, through the Microelectronics Commons (ME Commons) program of the U.S. Department of Defense (DoD), administered by the Applied Research Institute (ARI); by the Center for the Co-Design of Cognitive Systems (CoCoSYS), a research center under the Joint University Microelectronics Program (JUMP) 2.0, a Semiconductor Research Corporation (SRC) initiative sponsored by DARPA, and by the Department of Energy.

REFERENCES

- [1] N. Maslej, L. Fattorini, E. Brynjolfsson, J. Etchemendy, K. Ligett, T. Lyons, J. Manyika, H. Ngo, J. C. Niebles, V. Parli, Y. Shoham, R. Wald, J. Clark, and R. Perrault, "Artificial intelligence index report 2023." 2023. [Online]. Available: <https://arxiv.org/abs/2310.03715>
- [2] P. Villalobos, J. Sevilla, T. Besiroglu, L. Heim, A. Ho, and M. Hobbhahn, "Machine learning model sizes and the parameter gap," 2022. [Online]. Available: <https://arxiv.org/abs/2207.02852>
- [3] W. A. Wulf and S. A. McKee, "Hitting the memory wall: Implications of the obvious," *ACM SIGARCH Computer Architecture News*, vol. 23, no. 1, pp. 20–24, 1995.
- [4] S. A. McKee, "Reflections on the memory wall," in *Proceedings of the 1st Conference on Computing Frontiers (CF)*, 2004, p. 162.
- [5] A. Ankit *et al.*, "Circuits and architectures for in-memory computing-based machine learning accelerators," *IEEE Micro*, vol. 40, no. 6, pp. 8–22, 2020, nov/Dec.
- [6] A. Ankit, *et al.*, "PUMA: A programmable ultra-efficient memristor-based accelerator for machine learning inference," in *Proceedings of the International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, 2019, pp. 715–731.
- [7] S. Negi *et al.*, "Algorithm hardware co-design for ADC-less compute-in-memory accelerator," *IEEE Transactions on Circuits and Systems for Artificial Intelligence*, vol. 1, no. 2, pp. 191–203, Dec. 2024.
- [8] H. Zhang *et al.*, "A 40 nm 33.6 TOPS/W 8t-SRAM computing-in-memory macro with DAC-less spike-pulse-truncation input and ADC-less charge-reservoir-integrate-counter output," in *Proceedings of the 2021 IEEE International Conference on Integrated Circuits, Technologies and Applications (ICTA)*, Zhuhai, China, 2021, pp. 123–124.
- [9] K. Yoshioka, S. Ando, S. Miyagi, Y.-C. Chen, and W. Zhang, "A review of SRAM-based compute-in-memory circuits," *arXiv preprint*, 2024. [Online]. Available: <https://arxiv.org/abs/2411.06079>
- [10] J. Kim, K. Lee, and J. Park, "A charge domain p-8t SRAM compute-in-memory with low-cost DAC/ADC operation for 4-bit input processing," *arXiv preprint*, 2022. [Online]. Available: <https://arxiv.org/abs/2211.16008>
- [11] M.-S. Le, T.-N. Pham, T.-D. Nguyen, and I.-J. Chang, "Prim: a variation-aware binary-neural-network framework for process-resilient compute-in-memory," *arXiv preprint*, 2021. [Online]. Available: <https://arxiv.org/abs/2110.09962>
- [12] A. Kneip and D. Bol, "Impact of analog non-idealities on the design space of 6t-sram current-domain dot-product operators for in-memory computing," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 68, no. 6, pp. 1931–1944, 2021.
- [13] Y. Xie *et al.*, "Roadmap to neuromorphic computing with emerging technologies," *arXiv preprint*, 2024. [Online]. Available: <https://arxiv.org/abs/2407.02353>
- [14] K. Roy, I. Chakraborty, M. Ali, A. Ankit, and A. Agrawal, "In-memory computing in emerging memory technologies for machine learning: An overview," in *Proceedings of the 57th ACM/IEEE Design Automation Conference (DAC)*, 2020, pp. 1–6.
- [15] 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, 3rd Quarter.
- [16] W. He *et al.*, "2-bit-per-cell RRAM-based in-memory computing for area-/energy-efficient deep learning," *IEEE Solid-State Circuits Letters*, vol. 3, pp. 194–197, 2020.
- [17] M. Ali *et al.*, "A 65 nm 1.4–6.7 TOPS/W adaptive-SNR sparsity-aware CIM core with load balancing support for DL workloads," in *Proceedings of the IEEE Custom Integrated Circuits Conference (CICC)*, 2023, pp. 1–2.
- [18] M. Jung *et al.*, "A crossbar array of magnetoresistive memory devices for in-memory computing," *Nature*, vol. 601, pp. 211–216, 2022.
- [19] M. Ali, I. Chakraborty, U. Saxena, A. Agrawal, A. Ankit, and K. Roy, "A 35.5–127.2 TOPS/W dynamic sparsity-aware reconfigurable-precision compute-in-memory SRAM macro for machine learning," *IEEE Solid-State Circuits Letters*, vol. 4, pp. 129–132, 2021.
- [20] J.-W. Su *et al.*, "16.3 a 28 nm 384 kb 6t-SRAM computation-in-memory macro with 8b precision for AI edge chips," in *Proceedings of the IEEE International Solid-State Circuits Conference (ISSCC)*, 2021, pp. 250–252.
- [21] S. Yin, Z. Jiang, J.-S. Seo, and M. Seok, "XNOR-SRAM: In-memory computing SRAM macro for binary/ternary deep neural networks," *IEEE Journal of Solid-State Circuits*, vol. 55, no. 6, pp. 1733–1743, Jun. 2020.
- [22] H. Jiang, S. Huang, X. Peng, and S. Yu, "MINT: Mixed-precision RRAM-based in-memory training architecture," in *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS)*. Piscataway, NJ, USA: IEEE Press, 2020, pp. 1–5.
- [23] R. Liu *et al.*, "Parallelizing SRAM arrays with customized bit-cell for binary neural networks," in *Proceedings of the 55th ACM/IEEE Design Automation Conference (DAC)*. Piscataway, NJ, USA: IEEE Press, 2018, pp. 1–6.
- [24] Y. Kim *et al.*, "Extreme partial-sum quantization for analog computing-in-memory neural network accelerators," *ACM Journal on Emerging Technologies in Computing Systems*, vol. 18, no. 4, pp. 1–19, 2022.
- [25] H. Kim *et al.*, "Algorithm/hardware co-design for in-memory neural network computing with minimal peripheral circuit overhead," in *Proceedings of the 57th ACM/IEEE Design Automation Conference (DAC)*. Piscataway, NJ, USA: IEEE Press, 2020, pp. 1–6.
- [26] A. Azamat *et al.*, "Quarry: Quantization-based ADC reduction for ReRAM-based deep neural network accelerators," in *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. Piscataway, NJ, USA: IEEE Press, 2021, pp. 1–7.
- [27] U. Saxena and K. Roy, "Partial-sum quantization for near adc-less compute-in-memory accelerators," in *2023 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED)*, 2023, pp. 1–6.
- [28] H. Cai, Z. Bian, Y. Hou, Y. Zhou *et al.*, "A 28 nm 2 mb stt-mram computing-in-memory macro with a refined bit-cell and 22.4–41.5 tops/w for ai inference," in *2023 IEEE International Solid-State Circuits Conference (ISSCC)*, San Francisco, CA, USA, 2023, p. –.
- [29] A. Shafiee *et al.*, "ISAAC: A convolutional neural network accelerator with in-situ analog arithmetic in crossbars," *ACM SIGARCH Computer Architecture News*, vol. 44, no. 3, pp. 14–26, 2016.
- [30] A. Agrawal, C. Wang, T. Sharma, and K. Roy, "Magnetoresistive circuits and systems: Embedded non-volatile memory to crossbar arrays," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 68, no. 6, pp. 2281–2294, 2021.
- [31] J. Li, P. Ndaï, A. Goel, S. Salahuddin, and K. Roy, "Design paradigm for robust spin-torque transfer magnetic ram (stt mram) from circuit/architecture perspective," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 18, no. 12, pp. 1710–1723, 2009.
- [32] E. Yu, G. K. K. U. Saxena, and K. Roy, "Ferroelectric capacitors and field-effect transistors as in-memory computing elements for machine learning workloads," *Scientific Reports*, 2024.
- [33] V.-T. Nguyen, Q.-K. Trinh, R. Zhang, and Y. Nakashima, "Stt-bsnn: An in-memory deep binary spiking neural network based on stt-mram," *IEEE Access*, vol. 9, pp. 151 373–151 385, 2021.
- [34] A. Gebregiorgis *et al.*, "Dealing with non-idealities in memristor-based computation-in-memory designs," in *Proceedings of the IFIP/IEEE International Conference on Very Large Scale Integration (VLSI-SoC)*, 2022, pp. 1–6.
- [35] Y. Zhou *et al.*, "Error-aware training for in-RRAM computing design considering non-ideal effects in crossbar arrays and peripheral circuits," in *Proceedings of the ACM International Symposium on Computer Architecture and Machine Learning Systems (SysML)*, 2025.
- [36] H. Wang *et al.*, "Compute-in-memory with non-volatile elements for neural networks," *Advanced Materials*, vol. 34, no. 52, p. 2204944, 2022.
- [37] S. Roy, K. Patil, and N. R. Shanbhag, "Fundamental limits on the computational accuracy of resistive crossbar-based in-memory architectures," in *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS)*, 2022, pp. 1466–1470.
- [38] Z. Wang, M. Victor, and A. Gupta, "Comparative evaluation of memory technologies for synaptic crossbar arrays – part i: Robustness-driven device-circuit co-design and system implications," *IEEE Transactions on Computers*, 2024.
- [39] Z. Wang *et al.*, "Comparative evaluation of memory technologies for synaptic crossbar arrays – part ii: Design knobs and dnn accuracy trends," *IEEE Transactions on Computers*, 2024.
- [40] H. Cai, Z. Bian, Y. Hou, Y. Zhou, J.-I. Cui, Y. Guo, X. Tian, B. Liu, X. Si, Z. Wang, J. Yang, and W. Shan, "33.4 a 28nm 2mb stt-mram computing-in-memory macro with a refined bit-cell and 22.4 - 41.5tops/w for ai inference," in *2023 IEEE International Solid-State Circuits Conference (ISSCC)*, 2023, pp. 500–502.