SCIENCE CHINA
Information Sciences

• Supplementary File •

A 28nm Compute-In-Memory SRAM Macro with Adaptive Quantization and Replica-Column-Assisted Computation

Zhiting LIN[1,2,†] , Miao LONG[1,†] , Yang YANG[1], Fugui JIANG[1], Juntao GE[1], Chenglong DUAN[1], Jianping XIA[1] & Xiulong WU[1,2,*]

[1]*School of Integrated Circuits, Anhui University, Hefei 230601, China*
[2]*Anhui Provincial high-performance Integrated Circuit Engineering Research Center, Hefei 230601, China*

Appendix A Motivation and contribution

The utilization of Computing-In-Memory (CIM) has been widely put forth as a highly potential solution to address the "memory wall" bottleneck that is prevalent in the conventional von Neumann architecture. Modern data-intensive applications, such as convolutional neural networks (CNNs) and transformer models, require massive multiply-and-accumulate (MAC) operations, which lead to unacceptable latency and power consumption due to constant data shuffling between the separated memory and processing units. Numerous studies have been conducted on CIM, encompassing both digital CIM [1] and analog CIM [2] approaches. The former typically achieves high computational accuracy but exhibits significant area overhead and energy consumption, while the latter offers superior area/energy efficiency but often experiences accuracy degradation due to device non-linearity, process variations, and imprecise analog computation. In practice, the design of analog CIM architectures necessitates meticulous deliberation on the trade-off between computational accuracy and area/energy efficiency. Among various analog CIM implementations, static random-access memory (SRAM) is highly favored due to its excellent logic compatibility, high endurance, and advanced scaling capabilities in modern CMOS technology nodes. In Figure A1(a), we present a qualitative comparison illustrating the advantages of integrating computation directly into the SRAM array to break the von Neumann bottleneck. However, achieving high-precision multi-bit MAC operations entirely within the analog domain of SRAM arrays still faces several critical physical and architectural limitations that must be resolved.

Appendix A.1 Challenges in SRAM-based analog CIM

Previous analog CIM macros have extensively explored SRAM devices, offering greater flexibility for different tasks [3-6]. However, when deploying analog MAC operations on shared bitlines, the inherent analog characteristics of SRAM cells severely impact the overall system reliability. As illustrated in Figure A1(b), the relevant challenges collectively manifest in three critical circuit-level aspects:

• **Challenge 1: Non-linear and Mismatched Bitline Discharge.** The simultaneous activation of multiple input wordlines (WLs) during CIM operations results in dynamic variations of the drain-source voltage (V_{DS}) across the access transistors of the SRAM cells. This dynamic change leads to a highly non-linear discharge behavior on the shared bitline, severely compromising the linearity of the MAC results [7, 8]. Furthermore, process, voltage, and temperature (PVT) variations, coupled with significant IR drops along long bitlines in large arrays, introduce massive mismatches in discharge rates across different columns, severely degrading inter-column computational consistency.

• **Challenge 2: High Area and Power Cost of ADCs.** In SRAM-based analog CIM, the accumulated analog voltage on the bitlines must be quantized back into digital outputs for subsequent neural network layers. Traditionally, this is achieved using power-hungry Analog-to-Digital Converters (ADCs), such as Flash or SAR ADCs [9-13]. These ADCs incur substantial area overhead?often occupying 17% to over 22% of the total macro area, as depicted in the area breakdown in Figure A1(b). This bulky peripheral circuitry heavily limits the computational density and overall energy efficiency of the CIM macro, making it difficult to scale up for larger parallel operations.

• **Challenge 3: Precision-Reliability Trade-off (Pseudo-write).** Traditional WL input encoding methods rely on a fixed least significant bit (LSB) voltage for bitline discharge[14-16]. This rigid approach creates a severe dilemma: if the discharge time is kept short to protect the cells, small MAC results will generate a tiny bitline voltage drop (ΔV), making them highly susceptible to thermal noise and leading to large relative quantization errors. Conversely, if the discharge time is extended to increase the LSB margin, large MAC results will cause excessive bitline discharge. This excessive voltage drop on the bitline can disturb the precharged internal storage nodes of the SRAM cell, triggering catastrophic "pseudo-write" events that permanently corrupt the stored weights.

* Corresponding author (email: xiulong@ahu.edu.cn)
† Lin Z T and Long M have the same contribution to this work.

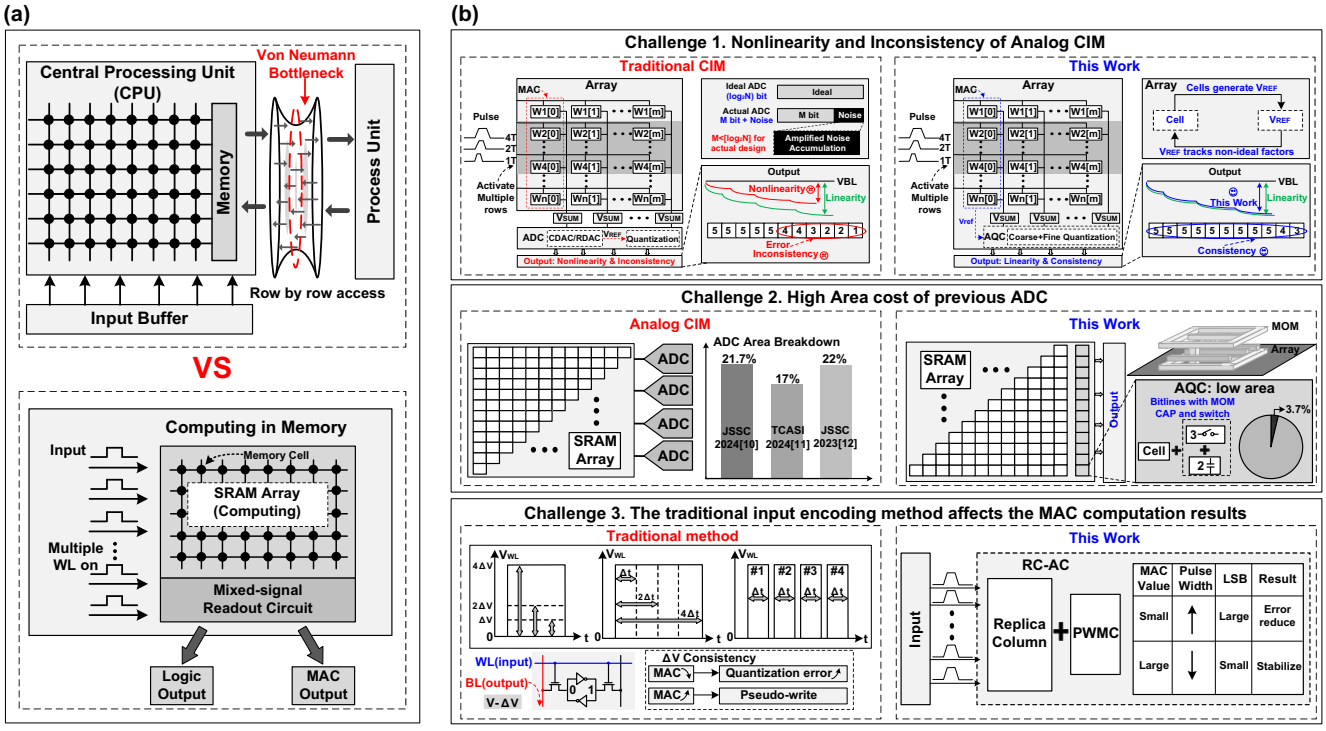

Figure A1 (a) Comparison between traditional Von Neumann architecture and the proposed CIM architecture. (b) Design challenges in SRAM-based analog CIM macros, highlighting non-linearity, ADC area breakdown, and the pseudo-write vs. quantization error trade-off.

Appendix A.2 Our contribution

To fundamentally address the aforementioned challenges encountered in SRAM-based analog CIM, this work proposes a novel 28nm SRAM-based CIM macro, designated as AQC-CIM, which significantly enhances linearity, discharge consistency, and resilience to PVT variations. The three main innovations of our proposed architecture are as follows:

- **A Shared SRAM Cell Architecture for Quantization and Computation:** We propose a mechanism where the exact same SRAM cells and identical bitline discharge paths are utilized for both computing the MAC results and generating the quantization reference voltages. By forcing the reference voltage to track the exact same non-linear discharge curve and PVT-induced jitter as the computation bitline, this shared-cell approach inherently cancels out non-linearities and inter-column inconsistencies.
- **An Adaptive Quantization Circuit (AQC):** To replace area-hungry and power-intensive ADCs, we introduce a highly compact capacitance comparison-based AQC. This circuit converts the analog discharge voltage into a digital value through a two-stage process (coarse quantization followed by fine quantization). This innovation drastically reduces the peripheral circuit area overhead to merely 3.7% of the total macro, while lowering control complexity and maintaining high accuracy.
- **A Replica Column-Assisted Circuit (RC-AC) paired with an Adjustable-Step Circuit (ASC):** To eliminate the precision-reliability trade-off, the RC-AC dynamically controls the bitline discharge time based on the input sparsity. It prevents pseudo-write risks by automatically terminating the wordline pulse for large MACs, and extends the pulse width to maximize the LSB voltage for small MACs. Concurrently, the ASC dynamically regulates the number of coarse quantization levels, effectively reducing the necessary comparison cycles and boosting overall throughput.

Appendix B Proposed SRAM-CIM Architecture and MAC Principle

The proposed AQC-CIM architecture designed for accurate and area/energy-efficient multi-bit MAC operations is presented in Figure B1(a). The architecture comprises a highly integrated 64×64 (4Kb) core array utilizing 4096 standard 6T SRAM cells. Alongside the core memory array, the peripheral circuitry includes the RC-AC for dynamic pulse width modulation, the AQC for ADC-less quantization, 6-to-64 decoders for row/column selection, and specific precharge timing control circuits.

Appendix B.1 Principle of MAC Operation

The MAC operation leverages the inherent discharge behavior of standard 6T SRAM cells, executed entirely within the analog domain on a single column. The detailed schematic diagram and the corresponding truth table are illustrated in Figure B1(b).

During the computation phase, the multiplier (weight) is stored in the internal node Q of the 6T SRAM cell, denoted as K, while the multiplicand (input activation) is represented by the main wordline (MWL), denoted as A. Before computation, all main bitlines (MBL) and bitline bars (MBLB) are precharged to the supply voltage (V_{DD}). When $A = 1$ (MWL is activated) and $K = 1$ (i.e., Q=1, QB=0), the MBL discharges through the access and pull-down transistors. If $A = 0$, the MWL remains inactive and the MBL voltage is unchanged. Because all SRAM cells in a specific column share the identical MBL, the total discharge current is accumulated. For a

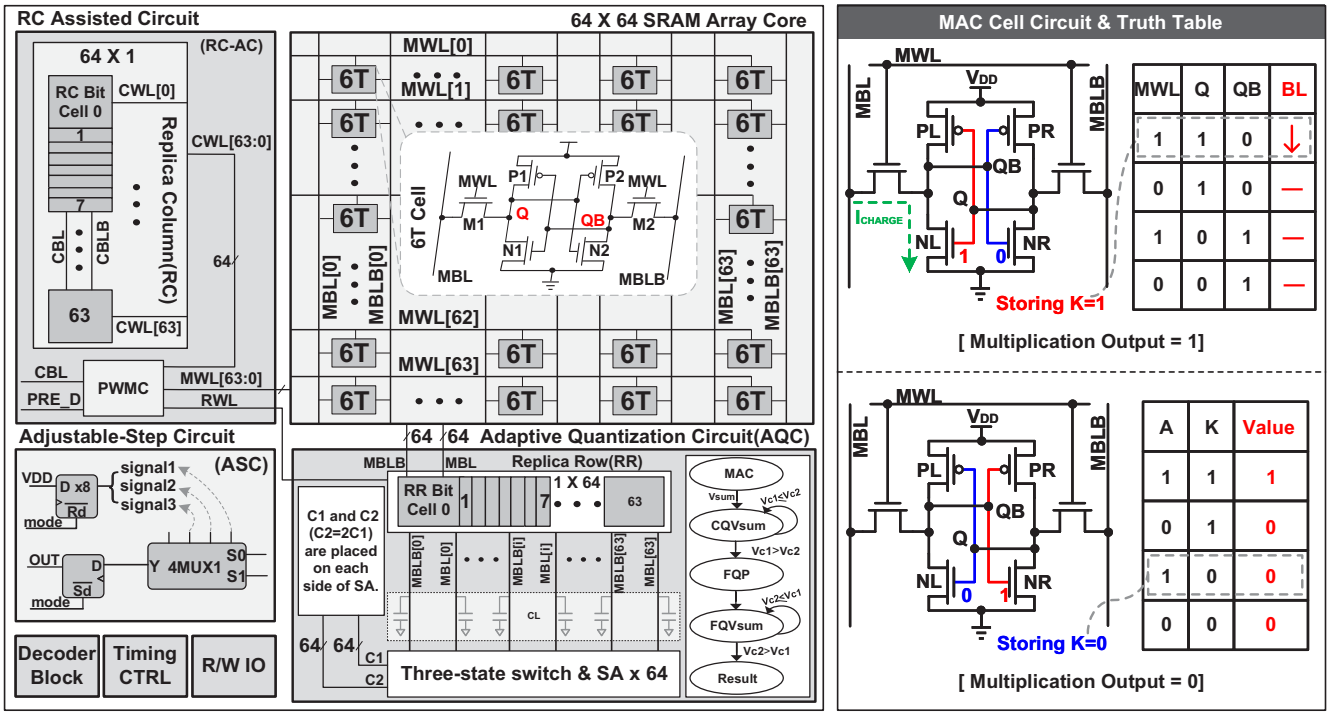

Figure B1 (a) Overall structure of the proposed AQC-CIM macro. (b) Schematic diagram of the 6T SRAM cell during computation and the truth table for the multiplication logic.

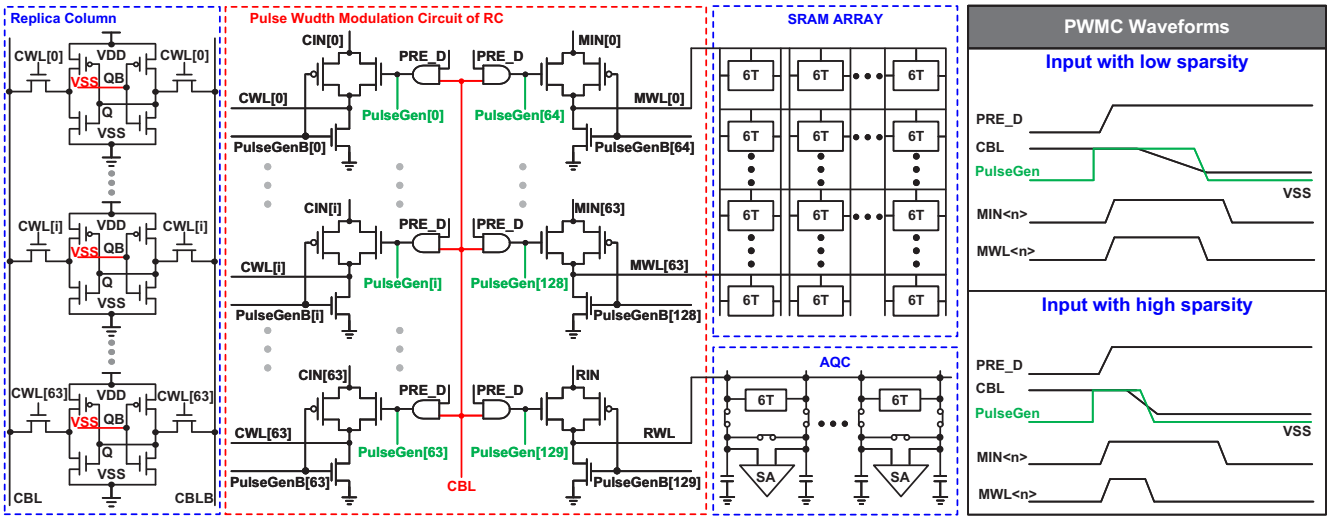

Figure B2 Detailed architectural schematic of the RC-AC PWMC and the corresponding timing waveforms showcasing dynamic pulse width modulation.

column with K cells outputting a logical '1', the cumulative discharge over a fixed pulse width T results in an analog bitline voltage drop of $K \times \Delta V$. Consequently, the final analog voltage representing the MAC sum on the bitline is $V_{\mathrm{SUM}} = V_{\mathrm{DD}} - K\Delta V$.

Appendix B.2 Detailed Operation of the RC-AC and PWMC

To resolve the critical precision-reliability trade-off (Challenge 3), we implement the RC-AC in conjunction with a Pulse Width Modulation Circuit (PWMC), as detailed in Figure B2.

The RC-AC consists of a 64×1 replica column of modified 6T SRAM cells. To guarantee that these replica cells provide a consistent reference discharge without suffering from pseudo-writes, the gate of one inverter in the replica cell is permanently tied to ground (V_{SS}). This asymmetric hardening effectively forces the cell to reliably retain a logical "0" while keeping the grounded transistor strongly conductive for reference discharge.

The dynamic operation proceeds in two key steps:

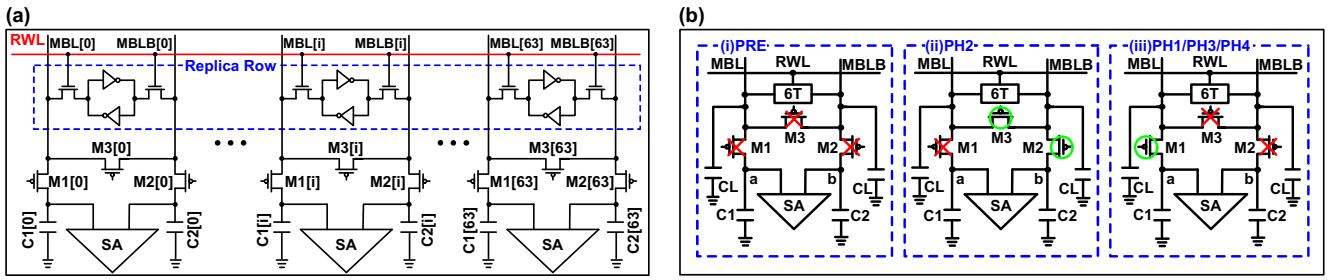

Figure C1 (a) Schematic diagram of the Adaptive Quantization Circuit (AQC). (b) Internal circuit switch status configurations across the four stages (PH1 to PH4).

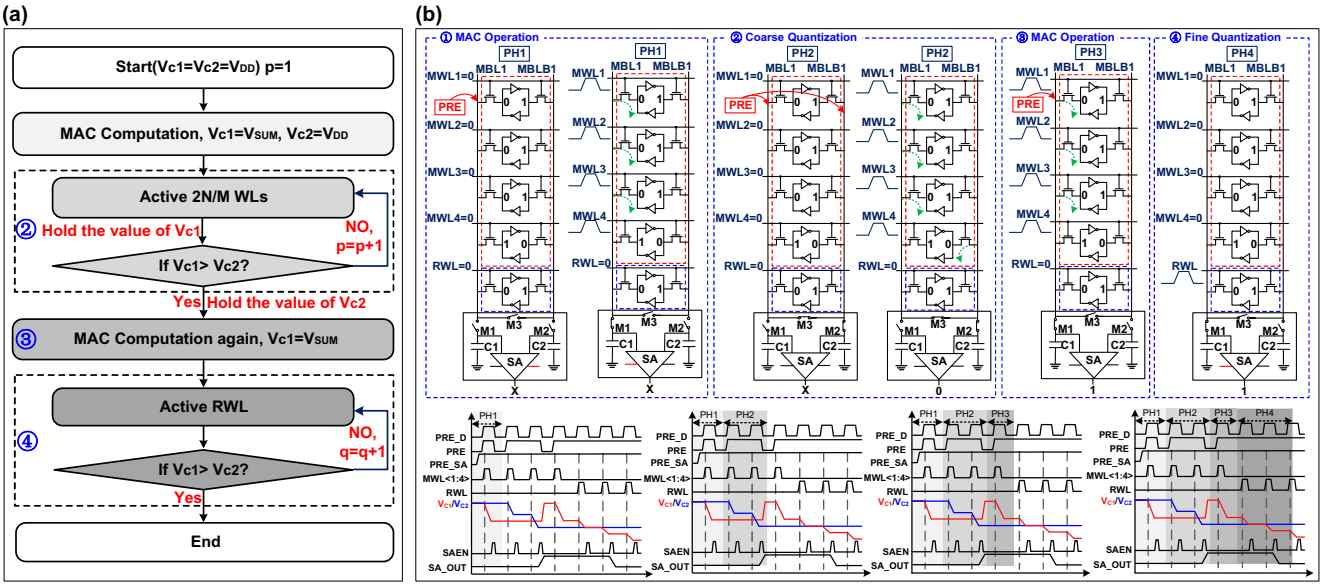

Figure C2 (a) Algorithm flow chart for the AQC operations. (b) Detailed timing and waveform diagram corresponding to the multi-phase quantization.

- **Step1. Precharge and Discharge Initiation:** The replica bitline (CBL) is precharged to V_{DD}. The logic activates a transmission gate, allowing the external array input signal (MIN) to propagate to the MWLs, thus initiating the discharge process simultaneously on the computation bitlines and the replica CBL.

Step2. Dynamic Discharge Termination: As the replica CBL discharges, its voltage drops. Once it falls below the predefined threshold of the inverter, the control logic triggers the PWMC to pull the MWLs down to V_{SS}, instantly halting the discharge across the entire array. For inputs yielding small MAC results (fewer active rows), the CBL discharges slower, automatically extending the MWL pulse width. This larger pulse width increases the ΔV per LSB, vastly improving reading margins and reducing relative quantization errors. Conversely, for large MAC results, the CBL discharges rapidly, shutting off the MWLs quickly to prevent the MBLs from dropping too low, entirely averting the risk of pseudo-writes.

Appendix C Adaptive Quantization Strategy and Adjustable-Step Circuit

To translate the analog V_{SUM} back into the digital domain without relying on massive, power-hungry ADCs, an AQC based on dynamic capacitive comparison is introduced. As illustrated in Figure C1(a), the AQC incorporates a replica row that shares the replica wordline (RWL) with the main array cells. Beneath the bitlines, the AQC incorporates two specifically sized capacitors (C_1 and C_2, where $C_2 = 2C_1$) and a shared sense amplifier (SA).

Appendix C.1 Four-Phase Quantization Workflow

The core principle of the AQC relies on the physical relationship $\Delta V = \frac{T \cdot \Delta I_{\mathrm{MBL}}}{C_{\mathrm{MBL}}}$. Since the total capacitance directly dictates the voltage drop for a given discharge current, manipulating the capacitor connection allows us to perform multi-level quantization iteratively. The process is divided into four strictly controlled phases (PH1 to PH4), with switch statuses shown in Figure C1(b) and the timing diagram in Figure C2:

- **PH1 (First Computation Phase):** The initial MAC operation is executed with the input stimuli. The resulting voltage drop reflects the cumulative MAC sum. This analog voltage is captured and stored on capacitor C_1 as $V_{\mathrm{SUM}} = V_{\mathrm{DD}} - K\Delta V$.

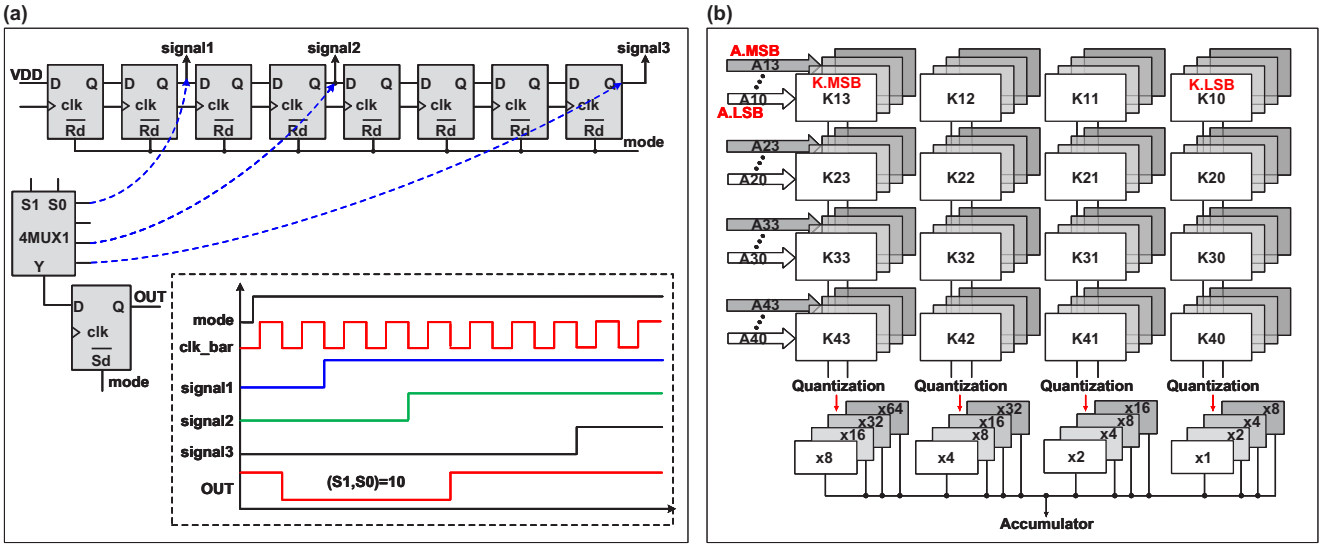

Figure C3 (a) Detailed logic diagram of the Adjustable-Step Circuit (ASC) utilizing D flip-flops. (b) Schematic representing the spatial and temporal accumulation of multiple bits for higher precision operations.

• **PH2 (Coarse Quantization Phase):** This phase determines the coarse interval of the MAC result. The switches reconfigure the total capacitance on the MBL to $2C_L + 2C_1$. The circuit then iteratively activates $\frac{2N}{M}$ MWLs per cycle (where N is the total rows and M is the coarse level count). This action generates a sequential reference voltage $V_{\mathrm{REF}_i} = V_{\mathrm{DD}} - i \cdot \frac{N}{M} \cdot \Delta V$ on C_2. The SA compares this internally generated V_{REF_i} against the stored V_{SUM}. Once the SA output flips, the index p is recorded, successfully establishing the coarse lower bound of the MAC result.

• **PH3 (Second Computation Phase):** Because the coarse quantization phase inherently altered the charge on the MBL, a second identical MAC computation is rapidly performed to restore the exact V_{SUM} onto C_1. Since V_{REF} is now securely buffered on C_2, the SA is disabled during this phase to save power.

• **PH4 (Fine Quantization Phase):** To resolve the exact digital value within the coarse interval, the RWL is activated cycle-by-cycle. Each activation discharges V_{SUM} by exactly one LSB (ΔV). The SA continuously compares the decreasing V_{SUM} against the fixed V_{REF} stored on C_2. By recording the number of discharge steps q required to flip the SA, the exact MAC result is calculated using Equation (C1), and the final digital output k is extracted via Equation (C2):

$$V_{\mathrm{SUM}} = V_{\mathrm{DD}} - p\left(\frac{N}{M}\right) \cdot \Delta V + (q-1)\Delta V \tag{C1}$$

$$k = p\left(\frac{N}{M}\right) - q + 1 \tag{C2}$$

Crucially, because V_{REF} is generated intrinsically using the exact same column cells and bitline capacitance as the computation, it perfectly traces and cancels out any non-linear discharge behaviors and PVT-induced jitter.

Appendix C.2 ASC and Multi-bit Expansion

Executing a fixed number of coarse levels can lead to redundant cycles for highly sparse inputs. To mitigate this, an ASC is integrated, as depicted in Figure C3(a). The ASC generates dynamic control signals using a cascading chain of asynchronously resettable D flip-flops. Depending on the input sparsity detected, the ASC adjusts the coarse quantization granularity M (e.g., switching from 8 levels to 4 or 2). For a 64-input scenario, utilizing an ASC level of 8 reduces the fine quantization from a maximum of 16 cycles down to 8 cycles, drastically reducing the overall latency and boosting energy efficiency.

Furthermore, Figure C3(b) illustrates the extension to multi-bit computations. If an N-bit weight and M-bit input are utilized, the architecture performs bit-serial processing over multiple cycles. The MAC outputs of different bit-significances are scaled (e.g., shifted by $\times 8$, $\times 4$, $\times 2$, $\times 1$) and accumulated digitally at the periphery, granting the macro full flexibility to handle high-precision neural network inference.

Appendix D Performance, Scalability, and Measurement Results

The proposed AQC-CIM macro was fully fabricated using a standard TSMC 28nm CMOS process. Extensive silicon measurements, combined with rigorous post-layout simulations, heavily validate the macro's exceptional robustness against environmental variations, its superior linearity, and its highly compact area footprint.

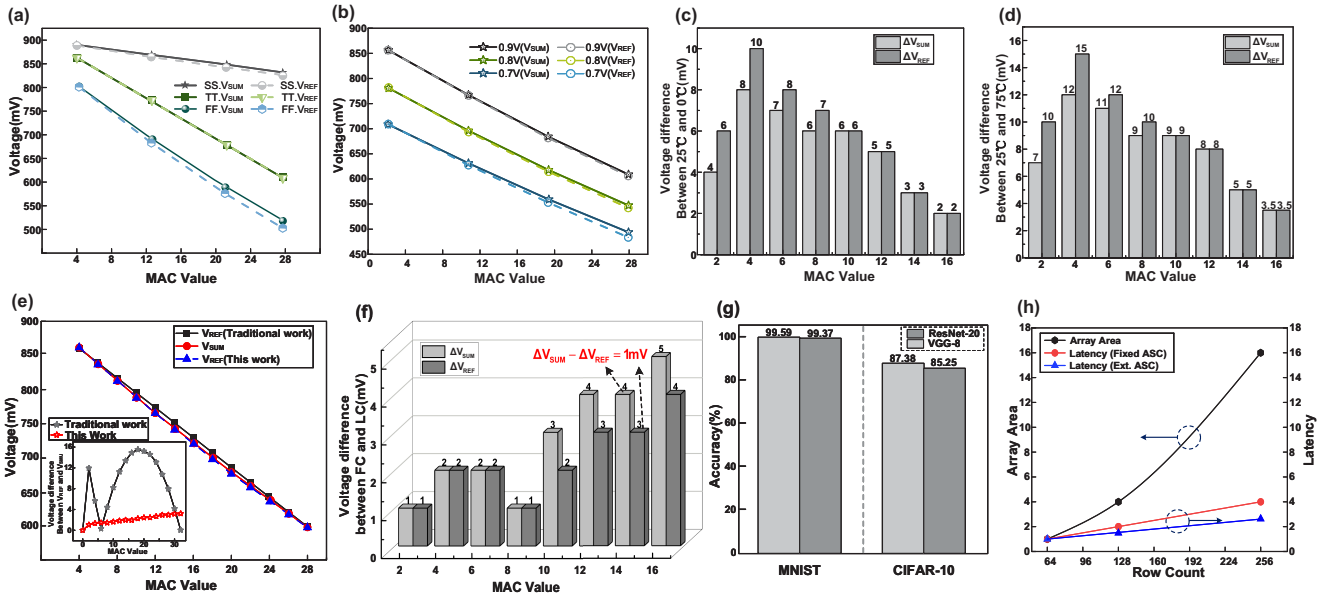

Figure D1 (a) V_{SUM} and V_{REF} corresponding to different MAC values under three different process. (b) V_{SUM} and V_{REF} under three different power supply voltages. (c) Two V_{SUM} differences and two V_{REF} differences detected at room temperature and low temperature. (d)Two V_{SUM} differences and two V_{REF} differences detected at room temperature and high temperature. (e) V_{SUM},V_{REF}(traditional),V_{REF}(this work) of different MAC values in the same column. (f) Two V_{SUM} differences and two V_{REF} differences detected in the first and last columns. (g) Classification accuracy for ResNet20 and VGG8 models across MNIST and CIFAR10 datasets. (h) Scalability trends of array area and quantization comparison latency with/without ASC scaling.

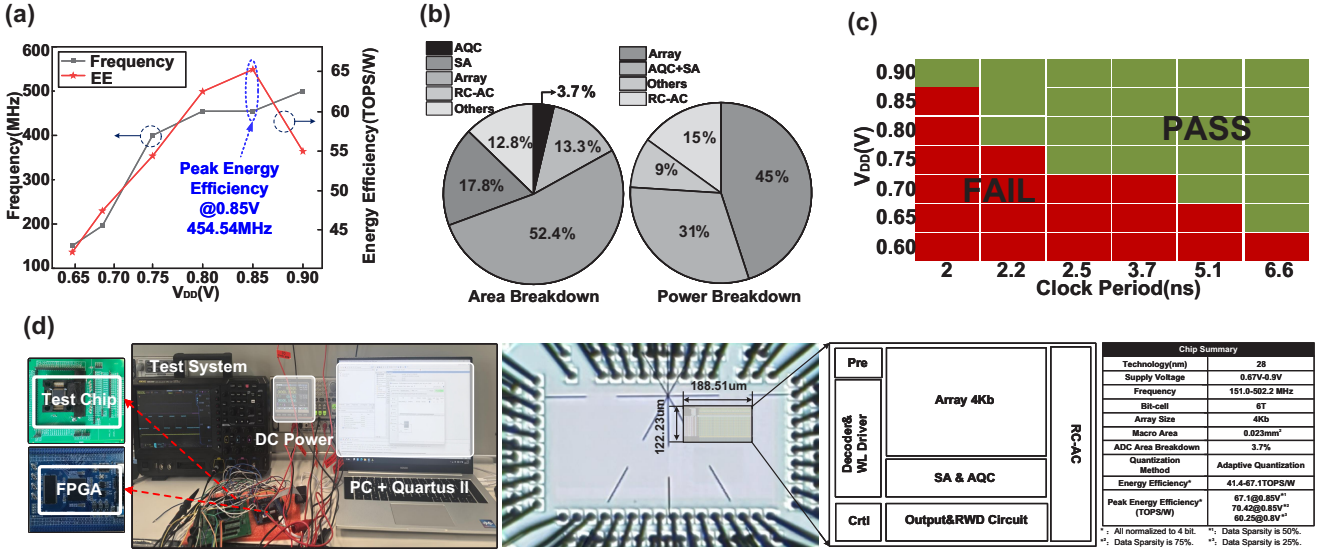

Figure D2 Test chip measurement results. (a) Area and power breakdown pie charts. (b) Operating frequency and energy efficiency curve. (c) Voltage-frequency Shmoo plot. (d) Testing platform setup and detailed Chip Summary table.

Appendix D.1 PVT Resilience and Non-linearity Mitigation Analysis

The core advantage of the shared-cell AQC architecture is its inherent immunity to PVT variations. As illustrated in Figure D1(a), the voltages V_{SUM} and V_{REF} were evaluated across TT, FF, and SS process corners. Despite the absolute discharge speed varying drastically (with FF being the fastest and SS being the slowest), the dynamically generated V_{REF} perfectly traces V_{SUM} in every scenario. Figure D1(b) confirms this tracking capability across supply voltage variations from 0.9V down to 0.7V.

Temperature resilience is validated in Figure D1(c) and (d). Even when contrasting extreme thermal variations (0°C vs. 25°C, and 25°C vs. 75°C), the maximum discrepancy between the computation voltage ΔV_{SUM} and the reference voltage ΔV_{REF} never exceeds 3 mV. This tiny fluctuation is far below the offset threshold required to falsely flip the SA, providing a foolproof guarantee against temperature-induced quantization failures.

Regarding non-linearity, traditional fixed-reference quantization schemes suffer severely. As shown in Figure D1(e), traditional methods exhibit up to a 9 mV error due to the non-linear V_{DS} shift of the SRAM access transistors. However, our proposed method forces

Table D1 Comparison with state-of-the-art SRAM CIM studies

Metric	This Work	JSSC'24 [10]	JSSC'23 [30]	JSSC'24 [31]	JSSC'25 [32]	SSCL'24 [33]
Technology (nm)	**28**	28	22	28	65	40
Macro Size	**4Kb**	2Kb	128Kb	36Kb	40.5Kb	64Kb
Cell Type	**6T**	8T	9T	4T1T	6T	6T
Macro Area (mm^2)	**0.023**	0.032	0.25	0.173	0.0725	0.212
Quantization Method	**AQC**	Flash ADC	SAR ADC	SAR ADC	TD-ADC	BuADC
Mitigate Inconsistency	**Yes**	No	No	No	Yes	No
Mitigate Nonlinearity	**Yes**	No	No	No	Yes	No
Adjust LSB Voltage	**Yes**	No	No	No	Yes	No
ADC Area Breakdown (%)	**3.7**	21.8	13	-	-	-
Energy Eff. (TOPS/W)*	**41.4-67.1**	22.2	62	16.4/25.8	18.6	60.8

*[1]:Input precision normalized to 4 bit. *[2]: Data Sparsity is 50%

V_{REF} to follow the identical non-linear discharge path, crushing the error margin down to 1 mV. Similarly, inter-column inconsistencies caused by spatial IR drop are eliminated (Figure D1(f)).

To prove system-level viability, standard neural network models were mapped onto the macro, as shown in Figure D1(g). The macro achieves 87.38% inference accuracy for ResNet20 and 85.25% for VGG8 on the CIFAR-10 dataset, and an impressive 99.59% on MNIST, demonstrating almost zero accuracy degradation compared to ideal floating-point software baselines.

Appendix D.2 Scalability Trends

The architecture exhibits extremely robust physical and temporal scalability, detailed in Figure D1(h). Because the SRAM array area scales quadratically ($O(N^2)$) while the AQC and RC-AC peripheral overhead scales linearly ($O(N)$), the relative area penalty of the proposed quantization circuitry becomes increasingly negligible in massive high-density memory arrays. From a latency perspective, expanding the ASC configuration (e.g., adding more D flip-flops to support 16 or 32 coarse levels) ensures that the worst-case quantization cycles remain near-constant or grow extremely slowly, easily supporting array sizes of 128×128 or 256×256 without creating timing bottlenecks.

Appendix D.3 Silicon Measurement Results and Benchmark Comparison

Figure D2(a) to (d) summarize the empirical data measured directly from the fabricated 28nm test chip. The AQC-CIM macro operates reliably across a wide frequency spectrum from 151.0 MHz to 502.2 MHz. When operations are normalized to 4-bit precision, the macro achieves a remarkable peak energy efficiency of 67.1 TOPS/W at 0.85V.

Crucially, the pie chart in Figure D2(a) illustrates the area breakdown: the entire AQC and SA module occupies a minuscule 3.7% of the total chip area ($0.023mm^2$). This is a paradigm shift compared to traditional architectures. Table D1 provides a comprehensive comparison with state-of-the-art SRAM CIM macros published in recent top-tier journals [10, 30-33]. While competing works rely on bulky Flash ADCs (21.8% area) or SAR ADCs (13% area), our AQC fundamentally eliminates these components, delivering superior area efficiency, built-in non-linearity mitigation, and dynamic LSB voltage adjustments.

References

1 D. Wang, C.-T. Lin, G. K. Chen, P. Knag, R. K. Krishnamurthy, and M. Seok, "DIMC: 2219TOPS/W 2569F2/b Digital In-mMemory Computing Macro in 28 nm Based on Approximate Arithmetic Hardware," in IEEE International Solid-State Circuits Conference (ISSCC) Dig. Tech. Papers, Feb. 2022, pp. 266-268.

2 H. Fujiwara et al., "A 5-nm 254 TOPS/W 221 TOPS/mm2 fully-digital computing-in-memory macro supporting wide-range dynamic-voltagefrequency scaling and simultaneous MAC and write operations," in IEEE Int. Solid-State Circuits Conf. (ISSCC), San Francisco, CA, USA, 2022, pp. 1-3.

3 J.-W. Su et al., "A 8-b-precision 6T SRAM computing-in-memory macro using segmented-bitline charge-sharing scheme for AI edge chips," IEEE J. Solid-State Circuits, vol. 58, no. 3, Mar. 2023, pp. 877-892.

4 X. Si et al., "A local computing cell and 6T SRAM-based computingin-memory macro with 8-b MAC operation for edge AI chips," IEEE J. Solid-State Circuits, vol. 56, no. 9, Sep. 2021, pp. 2817-2831.

5 P. Dong et al., "HeatViT: Hardware-efficient adaptive token pruning for vision transformers," in Proc. IEEE Int. Symp. HighPerformance Comput. Archit. (HPCA), Feb. 2023, pp. 442-455.

6 T. Xia et al., "An energy-and-area-efficient CNN accelerator for universal powers-of-two quantization," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 70, no. 3, Mar. 2023, pp. 1242-1255.

7 N. Fasfous et al., "HW-FlowQ: A multi-abstraction level HW-CNN co-design quantization methodology," ACM Trans. Embedded Comput. Syst., vol. 20, no. 5s, Oct. 2021, pp. 1-25.

8 Z. Lin et al., "Cascade Current Mirror to Improve Linearity and Consistency in SRAM In-Memory Computing," in IEEE Journal of Solid-State Circuits, vol. 56, no. 8, Aug. 2021, pp. 2550-2562.

9 Z. Chen, Q. Jin, J. Wang, Y. Wang, and K. Yang, "MC2-RAM: An in-8T-SRAM computing macro featuring multi-bit charge-domain computing and ADC-reduction weight encoding," in Proc. IEEE/ACM Int. Symp. Low Power Electron. Design (ISLPED), Jul. 2021, pp. 1-6.

10 K. Lee et al., "A 28-nm 50.1 TOPS/W P-8T SRAM Compute-In Memory Macro Design With BL Charge-Sharing-Based In-SRAM DAC/ADC Operations," in IEEE Journal of Solid-State Circuits, vol. 59, no. 6, June 2024, pp. 1926-1937.

11 X. Qiao et al., "A 16.38TOPS and 4.55POPS/W SRAM Computing-in-Memory Macro for Signed Operands Computation and Batch Normalization Implementation," in IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 71, no. 4, April 2024, pp. 1706-1718.

12 R. Sehgal et al., "A Bit-Serial, Compute-in-SRAM Design Featuring Hybrid-Integrating ADCs and Input Dependent Binary Scaled Precharge Eliminating DACs for Energy-Efficient DNN Inference," in IEEE Journal of Solid-State Circuits, July 2023, vol. 58, no. 7, pp. 2109-2124.

13 E. J. Choi et al., "SRAM-based computing-in-memory macro with fully parallel one-step multibit computation," IEEE Solid-State Circuits Lett., vol. 5, 2022, pp. 234-237.
14 Z. Chen et al., "CAP-RAM: A charge-domain in-memory computing 6T-SRAM for accurate and precision-programmable CNN inference," IEEE J. Solid-State Circuits, vol. 56, no. 6, Jun. 2021, pp. 1924-1935.
15 S. Jain, L. Lin, and M. Alioto, "±CIM SRAM for signed in-memory broad-purpose computing from DSP to neural processing," IEEE J. Solid-State Circuits, vol. 56, no. 10, Oct. 2021, pp. 2981-2992.
16 P.-C. Wu et al., "A 22 nm 832 Kb hybrid-domain floating-point SRAM in-memory-compute macro with 16.2-70.2TFLOPS/W for high-accuracy AI-edge devices," in IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers, Feb. 2023, pp. 126-128.
17 G. Jedhe et al., "A 12 nm 137 TOPS/W digital compute-in-memory using foundry 8T SRAM bitcell supporting 16 kernel weight sets for AI edge applications," in Proc. IEEE Symp. VLSI Technol. Circuits (VLSI Technol. Circuits), Jun. 2023, pp. 1-2.
18 A. Guo et al., "A 28 nm 64-kb 31.6-TFLOPS/W digital-domain floating-point-computing-unit and double-bit 6T-SRAM computing-in-memory macro for floating-point CNNs," in IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers, 2023, pp. 128-130.
19 H. Mori et al., "A 4 nm 6163-TOPS/W/b 4790-TOPS/mm2/b SRAM based digital-computing-in-memory macro supporting bit-width flexibility and simultaneous MAC and weight update," in IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers, 2023, pp. 132-134.
20 Y. He et al., "A 28 nm 38-to-102-TOPS/W 8b multiply-less approximate digital SRAM compute-in-memory macro for neural-network inference," in IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers, Feb. 2023, pp. 130-132.
21 B. Yan et al., "A 1.041-Mb/mm2 27.38-TOPS/W signed-INT8 dynamiclogic-based ADC-less SRAM compute-in-memory macro in 28 nm with reconfigurable bitwise operation for AI and embedded applications," in IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers, Feb. 2022, pp. 188-190.
22 Jiang Z, Yin S, Seo J S, et al. "C3SRAM: An In-Memory-Computing SRAM Macro Based on Robust Capacitive Coupling Computing Mechanism," in IEEE Journal of Solid-State Circuits, vol. 55, no. 7, 2020, pp. 1888-1897.
23 Yin S, Jiang Z, Seo J S, et al. "XNOR-SRAM: In-Memory Computing SRAM Macro for Binary/Ternary Deep Neural Networks," IEEE Journal of Solid-State Circuits, vol. 55, no. 6, 2020, pp. 1733-1743.
24 Jia H, Valavi H, Tang Y, et al. "A Programmable Heterogeneous Microprocessor Based onBit-Scalable In-Memory Computing," IEEE Journal of Solid-State Circuits, vol. 55, no. 9, 2020, pp. 2609-2621.
25 Su J W, Chou Y C, Liu R, et al. "A 8-b-Precision 6T SRAM Computing-in-Memory Macro Using Segmented-Bitline Charge-Sharing Scheme for AI Edge Chips," IEEE Journal of Solid-State Circuits, 2022, pp. 1-16.
26 H. Jeong, S. Kim, J. Shin, K. Park and K. J. Lee, "A 273.48 TOPS/W and 1.58 Mb/mm2 Analog-Digital Hybrid CIM Processor with Transpose Ternary-eDRAM Bitcell," 2024 IEEE Asian Solid-State Circuits Conference (A-SSCC), Hiroshima, Japan, 2024, pp. 1-3,
27 C. Yu et al., "A Logic-Compatible eDRAM Compute-In-Memory With Embedded ADCs for Processing Neural Networks," in IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 68, no. 2, 2021, pp. 667-679
28 D. Kim, H. Jeong, K. Park, S. Kim, S. An and K. Lee, "A 0.74 uJ/decision and 22.59 TOPS/W Keyword Spotting CIM Processor with Short-Current-Free Multi-level ReRAM and Adaptive-Decision-Level Nonlinear ADC," 2025 IEEE International Symposium on Circuits and Systems (ISCAS), London, United Kingdom, 2025, pp. 1-5,
29 E. Lee et al., "A charge-domain scalable-weight in-memory computing macro with dual-SRAM architecture for precision-scalable DNN accelerators," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 68, no. 8, Aug. 2021, pp. 3305-3316.
30 H. Wang et al., "A Charge Domain SRAM Compute-in-Memory Macro With C-2C Ladder-Based 8-Bit MAC Unit in 22-nm FinFET Process for Edge Inference," in IEEE Journal of Solid-State Circuits, vol. 58, no. 4, April 2023, pp. 1037-1050.
31 C. Zhao et al., "A 28-nm 36 Kb SRAM CIM Engine With 0.173 um2 4T1T Cell and Self-Load-0 Weight Update for AI Inference and Training Applications," in IEEE Journal of Solid-State Circuits, vol. 59, no. 10, 2024, pp. 3277-3289.
32 Z. Chen et al., "PICO-RAM: A PVT-Insensitive Analog Compute-In-Memory SRAM Macro With In Situ Multi-Bit Charge Computing and 6T Thin-Cell-Compatible Layout," in IEEE Journal of Solid-State Circuits, vol. 60, no. 1, Jan. 2025, pp. 308-320.
33 A. Guo et al., "A 28 nm 128-kb Exponent- and Mantissa-Computation-In-Memory Dual-Macro for Floating-Point and INT CNNs" in IEEE Journal of Solid-State Circuits, vol. 60, no. 10, Oct. 2025, pp. 3639-3654.