Acore-CIM: build accurate and robust mixed-signal CIM cores with RISC-V controlled self-calibration

Omar Numan, *Student Member*, Gaurav Singh, *Student Member*, Kazybek Adam, *Student Member*, Jelin Leslin, *Student Member*, Aleksi Korsman, *Student Member*, Otto Simola, *Student Member*, Marko Kosunen, *Member, IEEE*, Jussi Ryynänen, *Member, IEEE*, and Martin Andraud, *Member, IEEE*

***Abstract*—The development of accurate and robust compute-in-memory (CIM) architectures is a central research focus for accelerating artificial intelligence (AI) tasks, particularly deep neural networks (DNNs). Significant interest has emerged in analog and mixed-signal CIM architectures to improve the efficiency of data storage and computation, as well as to manage the large data volumes required by DNNs. Recent advances in emerging non-volatile memory (eNVM) solutions have driven progress in resistive mixed-signal CIM cores. However, mixed-signal CIM computing cores continue to face integration and robustness challenges that limit their widespread adoption in end-to-end AI computing systems. From an integration perspective, resistive and eNVM-based CIM cores require integration with a control processor to enable programmable acceleration. Additionally, SRAM-based CIM architectures remain more efficient and easier to program than eNVM-based alternatives. In terms of robustness, analog circuits are more susceptible to variations, leading to computation errors and reduced accuracy. This study addresses both challenges by introducing a self-calibrated mixed-signal CIM accelerator system-on-chip (Acore-CIM), fabricated using 22-nm FD-SOI technology. Integration is achieved through (1) a CIM architecture that combines the density and programmability of SRAM-based weight storage with multi-bit computation using linear resistors, alongside (2) an open-source programming and testing strategy for CIM systems. Accuracy and robustness are enhanced through automated RISC-V-controlled on-chip calibration, which improves compute signal-to-noise ratio (SNR) by 25–45 % across columns (reaching 18–24 dB) and reduces process-induced multiply-and-accumulate (MAC) error spread by up to 82 %. To demonstrate scalability, the proof-of-concept system-on-chip (SoC) is shown to be extendable to recent high-density linear resistor technologies, further enhancing computing performance.**

***Index Terms*—AI accelerator, computing-in-memory (CIM), deep neural network (DNN), self-calibration, RISC-V.**

This work was supported in part by the Academy of Finland through Project EHIR under Grant 13334487 and through Project WHISTLE under Grant 332218. This article was recommended by Associate Editor S. Ozev. *(Omar Numan and Gaurav Singh contributed equally to this work.) (Corresponding author: Omar Numan.)*

Omar Numan, Gaurav Singh, Kazybek Adam, Jelin Leslin, Aleksi Korsman, Otto Simola, Marko Kosunen, Jussi Ryynänen, and Martin Andraud are with the Department of Electronics and Nanoengineering, Aalto University, 02150 Espoo, Finland (e-mail: firstname.lastname@aalto.fi).

Omar Numan is also with the Nanoelectronics Group, VTT Technical Research Centre of Finland Ltd, FI-02044 VTT Espoo, Finland (e-mail: omar.numan@vtt.fi).

Gaurav Singh and Martin Andraud are also with the Institute of Information and Communication Technologies, Electronics and Applied Mathematics (ICTEAM), UCLouvain, 1348 Louvain-la-Neuve, Belgium (e-mail: firstname.lastname@uclouvain.be).

I. Introduction

The recent push for embedded artificial intelligence (AI), particularly deep neural networks (DNNs), in resource-limited systems has intensified the need for energy-efficient hardware *accelerators*. Compute-in-memory (CIM) cores have consequently attracted significant attention [1]. CIM leverages the inherent parallelism of DNN computations, which primarily consist of numerous multiply-and-accumulate (MAC) operations, by performing MAC directly in memory to minimize data transfers. Mixed-signal CIM further enhances efficiency by utilizing analog computing principles [2]. In these systems, MAC operations adhere to Ohm's law, with DNN weights represented as variable resistances (conductances) [3] or capacitances [4], and are multiplied by analog inputs. Accumulation is achieved natively through currents [3], [5] or charge sharing [4], [6]. Progress in resistive-based CIM cores is closely linked to the advancement of emerging non-volatile memories (eNVMs), which aim to enable multi-bit weight encoding, straightforward access, and non-volatility. Notable eNVMs include resistive RAM (RRAM) [7]–[10], spin-transfer torque magnetoresistive RAM (STT-MRAM) [11], ferroelectric RAM (FeRAM) [12], and phase change memory (PCM) [13]. Although these emerging technologies offer significant promise, substantial challenges remain in developing robust mixed-signal CIM solutions. Specifically, the analog nature of computations continues to present obstacles to the widespread deployment of mixed-signal CIM designs.

As a starting point, CIM architectures must provide sufficient computational accuracy for typical DNN tasks. In particular, a compute signal-to-noise ratio (SNR) in the $20-22$ dB range limits the loss in computational accuracy to $5\,\%$ compared to fixed-point computation [14]. This study adopts the compute SNR definition from [15], which explicitly considers both noise and distortion. For resistive-based CIM arrays, recent evaluations using multiple eNVM technologies [16] indicate that inherent device variability prevents achieving sufficient compute SNR without calibration. This challenge has prompted research into calibration strategies for various CIM cores [3], [8], [17]–[19] and resistive-based architectures [20], [21]. However, calibration techniques often introduce area, power, or timing overheads, thereby reducing system efficiency. Therefore, there is a need to develop automated and integrated calibration routines that minimize system-level overheads in area, power, and latency while achieving the required compute SNR.

Based on the initial observation, three key challenges are identified for developing more accurate and robust resistive-based CIM cores: **1** providing efficient and variation-tolerant computation units that leverage SRAM integration and weight storage, while opening the path for future eNVM adoption, particularly high-density linear resistive technologies; **2** simplifying the integration of CIM cores into complete AI acceleration systems with RISC-V processors through open-source programming, testing, and calibration routines; and **3** offering automated RISC-V-controlled self-calibration solutions to meet required accuracy targets. This work adopts a holistic approach to address these challenges and achieve accurate, robust mixed-signal CIM designs.

To address **Ch. 1**, an accuracy- and robustness-oriented MAC cell is designed, employing a multiplication scheme based on a multi-bit current-mode R-2R multiplying digital-to-analog converter (MDAC) topology, where each weight bit is stored in a 6T-SRAM cell [22], [23]. A key advantage of this approach is the use of SRAM weight storage and high-linearity resistors instead of multi-state eNVM devices, such as high-density linear resistor (HDLR) technologies [12], [24]. HDLRs offer higher resistance values and occupy smaller areas than typical poly resistors in CMOS, and can be post-processed on top of CMOS wafers. To tackle **Ch. 2**, a complete *proof-of-concept* SoC is designed, composed of a CIM core and a RISC-V control processor. The programming framework is fully open source, facilitating the testing and characterization of CIM-based designs in full systems. To address **Ch. 3**, an automated built-in self-calibration (BISC) routine is demonstrated on the SoC, operating in real time and fully controlled by the RISC-V core. The BISC mitigates offset and gain errors, improving the compute SNR by 25 to 45 % ($6-8$ dB) across all CIM columns, and raising the compute SNR to 18 to 24 dB. BISC further compensates for mismatch and process variations, reducing MAC error spread by up to 82 %. This approach ensures sufficient computational accuracy while minimizing calibration overheads by leveraging the existing AI computation system. The proof-of-concept SoC was fabricated and measured in 22-nm FD-SOI technology. Additionally, all hardware and software integration and calibration routines have been released as open source, including a model of the complete SoC. This release enables comprehensive testing of the architecture and programming routines in software prior to deployment on the physical device.

The structure of this paper is as follows. Section II reviews challenges in the design, robustness, and accuracy of mixed-signal CIM, with a focus on resistive-based arrays. Section III describes the proof-of-concept accelerator SoC that serves as the baseline for this study. Sections IV, V, and VI address the three aforementioned challenges within the proposed SoC architecture. Section VII presents measurement results from the proof-of-concept chip. Finally, Section VIII presents the conclusions drawn from this study.

II. Design and Robustness Challenges of Mixed-Signal CIM

A. SRAM versus eNVM CIM design

On the one hand, SRAM-based CIM has evolved from binary and ternary content-addressable memory for search applications to architectures that support logic and MAC operations, particularly in AI edge devices. This evolution is driven by its high endurance, rapid write speeds, low write energy, and compatibility with modern CMOS processes [25]. Various cell architectures are employed: compact 6T cells enable energy-efficient MAC operations, while less area-efficient 7T, 8T, and 10T cells enhance signal margin, reduce read disturbances, and support more complex functions [25], [26]. As a result, SRAM-based CIM systems now support multi-bit MAC operations [3]. However, several challenges persist. For instance, increasing MAC precision with compact 6T-SRAM cells often reduces signal margins, which diminishes readout accuracy and raises the probability of read disturbances, particularly when the signal margin is less than the analog-to-digital converter (ADC) input offset due to limited bitline voltage swing. Multi-bit MAC operations require additional cycles, increasing latency and reducing area efficiency. Furthermore, activating multiple wordlines can improve energy efficiency but may induce read instability due to larger bitline voltage swings, potentially leading to data flipping. Process variations also introduce input offsets in analog readout circuits, necessitating larger signal margins to maintain precision.

On the other hand, eNVM-based CIM cores integrate memory and processing within a single cell. Recent advancements in these architectures [7]–[10] have leveraged metal-oxide layers to achieve high-speed switching and computation. For example, RRAMs utilize a metal-insulator-metal structure in which conductive filaments are formed and ruptured through redox reactions. This mechanism makes RRAMs suitable for high-density, low-power, multi-bit memory applications. By supporting both logic operations and data storage within the same cell, MAC operations can be executed directly in memory, reducing data movement and substantially improving energy efficiency and computational speed.

Despite recent advancements, several challenges continue to affect the design of eNVM-based systems [10], [27]–[29]. For example, eNVM cells may struggle with higher multi-bit MAC operations. Implementations that use an 8-to-1 bitline multiplexer and serial access can reduce area and energy efficiency by up to 97 % and 94 %, respectively, compared to 1-bit operations [8]. Furthermore, eNVM cells typically require precise high-voltage pulses for programming, which increases circuit complexity. Additional peripheral circuits may also be necessary to prevent unintentional overwriting, known as write-disturb issues. From a computational perspective, increasing the number of active wordlines reduces the current-domain sensing margin, leading to errors caused by parasitic resistances in bitlines and multiplexers. Moreover, analog non-idealities, particularly in the summing amplifier, degrade current-sensing linearity and necessitate complex corrective measures.

In that regard, SRAM-based CIM cores remain up to 7× more efficient than their eNVM-based counterparts [26]. This efficiency advantage is attributed to three primary factors: **(I)** SRAMs continue to benefit substantially from technology scaling, **(II)** eNVMs have not yet achieved the same level of integration as SRAMs and are generally more challenging to program, and **(III)** the use of analog weights increases the demands on peripheral circuits such as amplifiers and data converters, thereby reducing both area and energy efficiency. Consequently, it is advantageous to develop a MAC computing unit that leverages the efficient programming and weight storage of SRAM, while also enabling the integration of eNVMs and HDLRs for power-efficient computation.

B. RISC-V as a co-processor for end-to-end CIM acceleration

While CIM arrays provide high throughput for MAC operations, a standalone CIM core is typically insufficient to execute complete deep learning workloads. Practical DNN inference requires additional processing beyond MACs, including data reshaping, control flow, and non-linear or element-wise operations, which are not efficiently supported within 2D CIM arrays [30]. Prior work has therefore highlighted the need for a general-purpose processing core to orchestrate end-to-end execution and to handle operations that remain outside the CIM datapath. Recent studies identify RISC-V and CIM co-design as a promising approach to bridge this gap: compute-intensive convolution/MAC kernels are offloaded to CIM, while a RISC-V core executes remaining layers and auxiliary tasks (e.g., depthwise convolutions, element-wise functions, and system-level control), optionally augmented with custom instructions or small accelerator blocks to reduce overhead [31]–[33]. This heterogeneous approach enables full-model deployment within an IoT/edge device context and provides a programmable interface for testing and characterization.

Motivated by these observations, this work adopts a RISC-V core as a co-processor to **(I)** orchestrate CIM operation through a clean memory-mapped interface, **(II)** support end-to-end execution by managing non-MAC workloads, and **(III)** automate calibration and characterization routines in software, enabling repeatable bring-up and evaluation of mixed-signal CIM cores in full systems.

C. Robustness of mixed-signal CIM cores

Resistive-based CIM cores are inherently susceptible to multiple non-idealities that degrade computational accuracy. These non-idealities originate from peripheral components, interconnect resistances, and device-level variations, resulting in errors that propagate throughout the CIM core [34]. Ensuring signal integrity for DNN workloads necessitates a comprehensive understanding of these effects and the application of appropriate corrective measures. Fig. 1 presents a detailed breakdown of the primary non-idealities impacting resistive-based CIM cores, organized as follows:

a) Non-ideal DACs ❶*:* Input DACs exhibit inaccuracies due to finite output impedance (R_{DAC}), load dependency (R_{L}), and process variations, causing deviations in applied voltages (V_{DAC}) across CIM array rows.

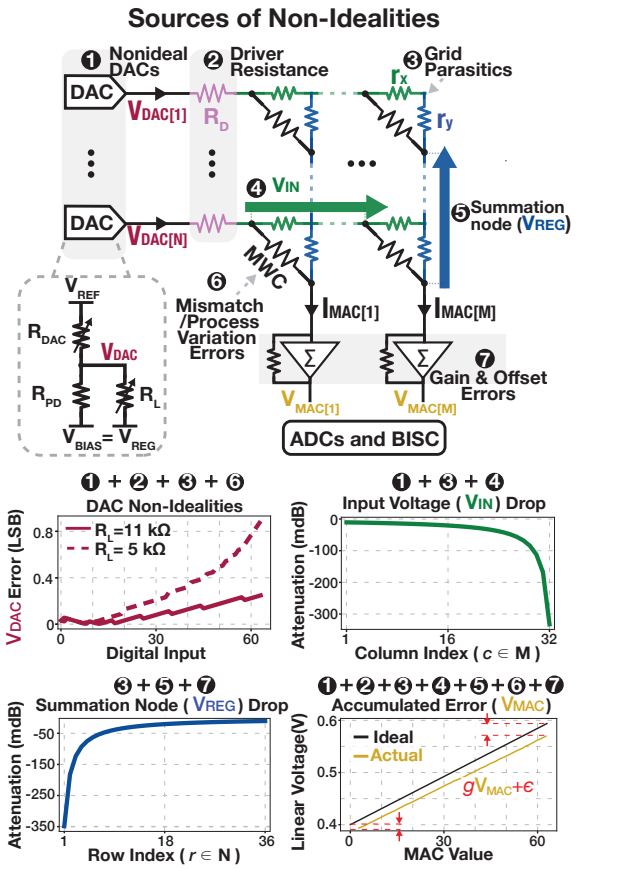


Fig. 1. Sources of non-idealities in resistive CIM cores: (1) DAC non-linearities, (2) driver resistance, (3) grid parasitics, (4) input attenuation, (5) summation-node voltage drop, and (6) process/mismatch variation. Example plots show DAC error, signal attenuation, and V_{REG} drop. These effects accumulate in MAC outputs ($I_{\mathrm{MAC}} \rightarrow V_{\mathrm{MAC}}$), causing gain ($g$) and offset ($\epsilon$) errors, motivating BISC calibration.

b) Driver resistance ❷*:* The finite output resistance of driver circuits (R_{D}) introduces voltage drops along the rows, further altering applied input levels.

c) Parasitic wire resistances ❸*:* Interconnect resistances (r_x, r_y) introduce voltage attenuation across both rows and columns, leading to signal degradation.

d) Input signal attenuation ❹*:* The combined effects of driver resistance and wire parasitic elements progressively reduce the input voltage (V_{IN}) at successive cross-points.

e) Summation node regulation voltage drop ❺*:* Parasitic resistances along the columns affect the virtual ground regulation voltage (V_{REG}), reducing accuracy in the accumulated current summation.

f) MAC cell conductance variability ❻*:* After chip fabrication, device-level variations and mismatch can shift operating points and add additional errors.

g) Summing amplifier offset and gain errors ❼*:* The Summing Amplifier ($\sum$ Amp) at the column outputs introduces offset and gain errors—input-referred offset voltage shifts MAC outputs, causing systematic bias, while a finite open-loop gain reduces I-V conversion accuracy. Additionally, the finite input impedance disrupts the virtual ground (V_{REG}), altering the accumulated currents. These distortions collectively degrade computational precision and linearity.

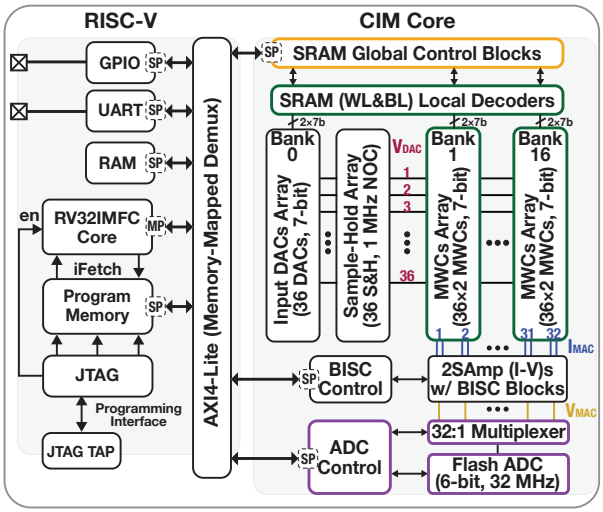


Fig. 2. Architecture of the Acore-CIM SoC: Top-level block diagram with 32-bit RISC-V, mixed-signal CIM accelerator, and AXI4-Lite interconnect.

The plots in Fig. 1 illustrate how these non-idealities affect CIM operation:

- DAC Non-Idealities (❶+❷+❸+❻)→Increased DAC output (V_{DAC}) errors, mainly due to load resistance (R_{L}) variations.
- Input Voltage Drop (❶+❸+❹)→Progressive attenuation of V_{IN} across MAC cell inputs, highlighting the effect of interconnect resistances.
- Summation Node Voltage Drop (❸+❺+❼)→Decrease in V_{REG} across rows due to parasitic resistance buildup.
- Accumulated Error (❶ **to** ❼)→Discrepancies between ideal and actual V_{MAC}, where gain (g) and offset (ϵ) errors emerge from the cumulative impact of these non-idealities. Error accumulation in CIM arrays is not strictly correlated with physical distance; it depends on the array state, programmed conductances, and input patterns. In addition to systematic errors, random variations—such as thermal noise, flicker noise, and inherent device mismatches—also contribute to performance variability.

In summary, resistive-based CIM cores suffer from interdependent non-idealities, where errors vary with array size, conductance states, input patterns, process variations, and random noise sources. To ensure robust compute accuracy under variations and analog non-idealities, a self-calibration mechanism is essential. Section VI details our BISC-based calibration strategy, which primarily targets systematic errors while recognizing that a residual random error floor remains.

III. Proof-of-Concept CIM Accelerator SoC

This section provides a high-level overview of the proposed Acore-CIM proof-of-concept CIM accelerator SoC. As illustrated in Fig. 2, the system integrates a RISC-V processor, a mixed-signal CIM core, and an AXI4-Lite interface for configuration and control. The subsequent discussion details the internal organization of the CIM core, including the array structure, the SRAM-based control hierarchy, and the column-level readout and digitization path.

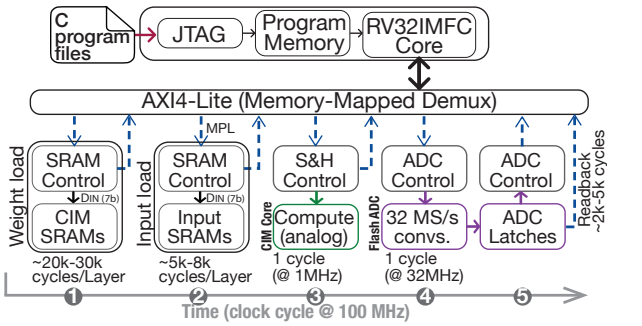


Fig. 3. Timing diagram of Acore-CIM during a DNN layer execution: (1) weight load, (2) input load, (3) MAC in one S&H cycle, (4) ADC conversion and latching of 32 columns, and (5) digital readback. Cycle counts shown at 100 MHz RISC-V clock.

A. RISC-V core and AXI4-Lite interface

The backbone of the proposed Acore-CIM is a custom-designed, five-stage RV32IMFC RISC-V compliant core, illustrated in Fig. 2 and made available as open-source [35]. This RISC-V core integrates 64 kB of program memory and 64 kB of RAM. It supports integer multiplication and division, single-precision floating-point operations, and compressed 16-bit instructions via the I, M, F, and C extensions. The core and AXI4-Lite interface, developed in Chisel HDL [36], are fully parameterizable to facilitate selection of application-specific extensions. In this work, all extensions are enabled, and the core achieves benchmark scores of 0.628 DMIPS/MHz and 2.07 Coremark/MHz. Programming of the core is accomplished via a JTAG interface, while communication with the CIM core and other peripherals, such as UARTs and GPIOs, is managed via an AXI4-Lite interconnect, which provides processor-controllable interfaces for external communication.

The RISC-V processor manages the CIM core through memory-mapped registers exposed on the AXI4-Lite bus. These registers enable writing instructions and configuration fields, initiating operations, and polling completion status. The timing diagram in Fig. 3 illustrates the resulting task parallelism. Execution begins with a C program loaded onto the RISC-V, which encodes the complete layer sequence, including register writes for weights, activations, and control strobes. During runtime, the C program coordinates the following sequence over AXI4-Lite: (1) streaming layer weights into the CIM SRAM banks; (2) loading input activations into the input SRAM; (3) asserting the S&H trigger to broadcast inputs and complete the analog MAC within a single 1 µs window; (4) time-multiplexing the 32-column output voltages into the 6-bit flash ADC for conversion at 32 MHz during the subsequent 1 µs, with codes latched as they are generated; and (5) reading back the latched results into on-chip RAM, thereby completing one inference pass.

B. CIM accelerator core

The CIM accelerator is organized into four main architectural blocks.

1) CIM Array and Bank Structure: The CIM core consists of a 36×32 crossbar array composed of multi-bit weight

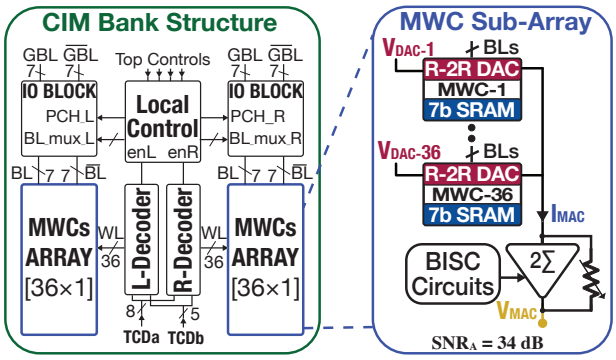


Fig. 4. CIM bank and MWC sub-array architecture. The CIM bank comprises SRAM-controlled MDAC-based weight cells (MWCs). Each MWC sub-array employs SRAM-controlled 7-bit R–2R MDACs and column BISC summing amplifiers for readout and calibration.

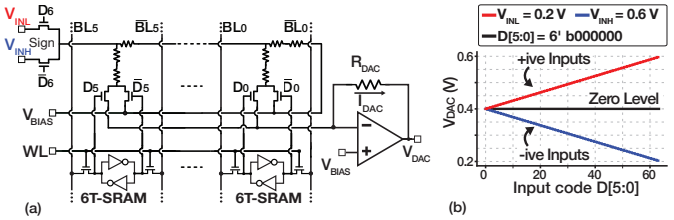


Fig. 5. Input DAC architecture. (a) Circuit of the 6-bit R-2R MDAC cell with an additional sign bit for dual-polarity operation. (b) Transfer characteristics showing V_{DAC} versus digital input code $D_{5:0}$, for both positive and negative inputs selected by the sign bit D_6.

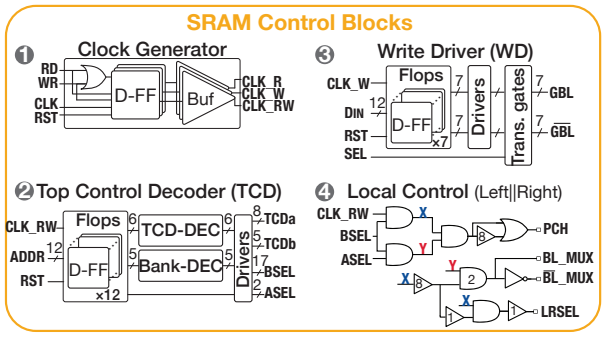


Fig. 6. SRAM control hierarchy with clock generator, decoders, and BL/BLB drivers.

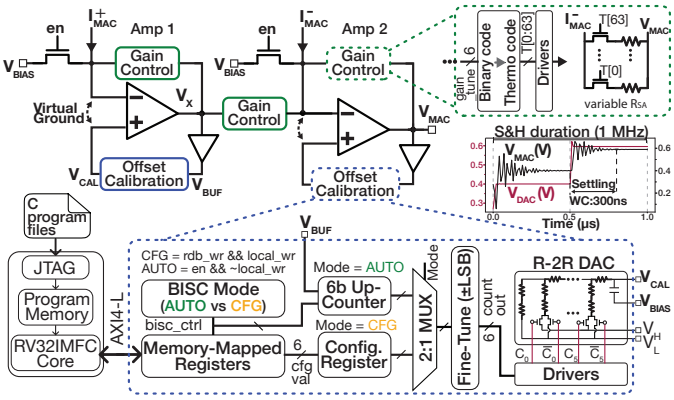


Fig. 7. BISC-controlled dual current summing amplifier (2ΣAmp) architecture. The circuit accumulates positive and negative column currents and converts them into V_{MAC}, which fully settles within the 1 µs S&H window. Integrated BISC blocks provide offset calibration and programmable gain control.

cells. Each cell is implemented as a 7-bit current-mode R–2R MDAC, with switches programmed by local SRAM. The 32 columns are grouped into 16 banks, each containing two columns that share SRAM control and peripheral I/O circuitry. Fig. 4 provides an overview of the CIM bank organization and the detailed MWC sub-array structure, highlighting the SRAM-controlled MDAC network and the column BISC summing amplifiers utilized for readout and calibration.

The input DAC array (Bank-0) employs the same banking scheme, with each DAC assigned to a row and driving 36 horizontal lines. Each DAC output is buffered by a sample-and-hold (S&H) circuit, which ensures stable signal distribution across the 32 columns. The S&H stage stabilizes the DAC outputs, suppresses noise, and aligns timing, thereby preserving input integrity prior to entering the CIM array. This configuration also provides flexibility to accommodate both analog and digital inputs with minimal architectural modification [37]. Fig. 5(a) depicts the input DAC cell, which is an R-2R MDAC with $B_D = 6+1$ bits: a 6-bit digital input ($D_{5:0}$) and a sign bit (D_6) for polarity control. For positive inputs, the DAC selects $V_{\mathrm{INL}} = 0.2$ V, while for negative inputs, it selects $V_{\mathrm{INH}} = 0.6$ V. To support signed MAC results, the analog zero level is set at $\frac{V_{\mathrm{INH}}+V_{\mathrm{INL}}}{2} = 0.4$ V (V_{BIAS}), establishing a zero-centered operating point. The corresponding transfer function is presented in Fig. 5(b).

2) SRAM Control Blocks: Fig. 6 illustrates the SRAM interface blocks. These blocks are configured by the RISC-V core, which writes `RD`, `WR`, `ADDR`, and `DIN` into memory-mapped registers using C code. These signals are transmitted via the AXI4-Lite bus to the global control block. The global control block operates a clock generator for read and write operations and a top-control decoder (TCD) that combines the programmed address with pre-decoded signals. The resulting wordline (WL) signals are distributed to local decoders within each bank. On the bitline (BL) side, write drivers (WDs) generate BL/BLB pairs, and I/O blocks provide precharge and BL multiplexing for left- or right-array selection. The half-butterfly SRAM architecture optimizes area efficiency and routing complexity while maintaining stable access for weight storage.

3) Output Summing Stage:

a) Summing amplifier: Each column terminates in a dual summing amplifier (2ΣAmp) circuit, depicted in Fig. 7. This circuit individually accumulates positive and negative column currents, then performs differential subtraction using a programmable feedback resistor network to produce a voltage output. A thermometer decoder modulates the effective transresistance for gain control, and offset calibration is achieved through a voltage-mode DAC and counter. The 2ΣAmp is instantiated 32 times, corresponding to one per CIM column. Each inference operation is completed within a single S&H period ($T_{S\&H} = 1\ \mu$s), during which all outputs reach full settling. The output voltage, V_{MAC}, is given by:

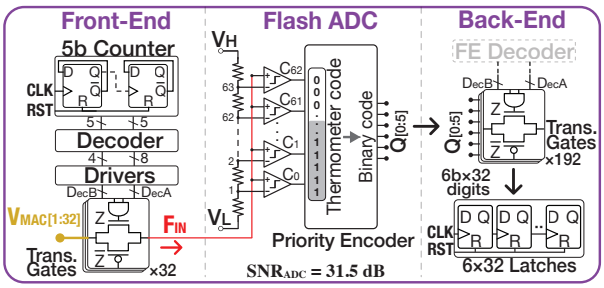


Fig. 8. A compact 6-bit Flash ADC with a 32:1 multiplexer digitizing 32 columns at 32 MHz.

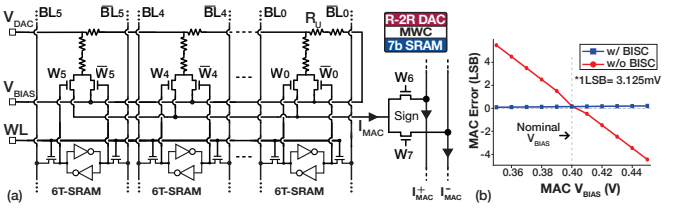


Fig. 9. (a) Schematic of the MDAC Weight Cell (MWC), implemented as a 7-bit R-2R DAC with 6-bit weight value $W_{5:0}$ and dual sign bits W_6, W_7 for polarity control and idle mode. (b) Post-layout result of MAC output error against deliberate V_{BIAS} offset injection and recovery using BISC.

$$V_{MAC} = R_{SA} \cdot I_{MAC} + V_{CAL}, \tag{1}$$

where R_{SA} denotes the transresistance of the amplifier and V_{CAL} represents a programmable offset, initially set to V_{BIAS}. Each operational amplifier in the 2ΣAmp has a DC gain of 60 dB and is configured to maintain the summing node at virtual ground, $V_{\mathrm{GND}} = V_{\mathrm{BIAS}}$. This configuration ensures that the current from the array cells remains proportional to the applied row voltages and is summed linearly on each branch.

b) BISC Control flow: The BISC circuitry is embedded into the output summing stage, where it calibrates and corrects non-linearities in the analog MAC output. As this process occurs in the analog domain prior to digitization, BISC eliminates offset and gain errors before the ADC, thereby enhancing the effective precision of the analog MAC. The calibration routine is managed by the RISC-V core via a 32-bit memory-mapped register, `bisc_ctrl`, which is accessible through the AXI4-Lite interface (Fig. 7). This register includes fields for mode selection (AUTO or CFG), manual configuration values, fine-tuning of the offset (±1 LSB), gain codes, and reset commands. In AUTO mode, a 6-bit up-counter iteratively sweeps calibration codes to minimize the residual offset. In CFG mode, the value written to `bisc_ctrl` is directly applied to the offset DAC. A 2:1 multiplexer selects between counter and register outputs, followed by a fine-tuning stage that drives the offset DAC (V_{CAL}) to adjust the summing amplifier's virtual ground. Gain codes are simultaneously applied to scale the transresistance. From the software perspective, a C program writes to `bisc_ctrl` to configure, initiate, and monitor calibration. The sequence involves: **(1)** programming the mode and parameters, **(2)** triggering AUTO or CFG calibration, **(3)** waiting for completion, and **(4)** reading back status or results. Further details regarding BISC operation are provided in Section VI.

4) Quantization/Flash ADC: Fig. 8 shows the on-chip 6-bit flash ADC shared across 32 CIM columns through a 32:1 analog multiplexer. During each inference cycle ($T_{S\&H}$ = 1 μs), the 32 summing-amplifier outputs are time-multiplexed and digitized at 32 MHz (one column per clock). The ADC consists of a resistor ladder, comparator array, and priority encoder, with default reference levels $(V_{ADC}^{L}, V_{ADC}^{H}) = (0.2, 0.6)$ V. The ADC employs a compact resistor-ladder topology. While area-efficient, this architecture imposes a hardware-level upper bound on the achievable column compute SNR due to quantization noise and inherent comparator non-idealities (e.g., input-dependent offset variations). Consequently, the ADC-column SNR is defined by $\mathrm{SNR}_{col} = \left(\frac{1}{\mathrm{SNR}_A} + \frac{1}{\mathrm{SNR}_{ADC}}\right)^{-1}$ [26], where SNR_A is the pre-ADC analog SNR and SNR_{ADC} accounts for both quantization and these residual circuit non-idealities.

At the network level, DNN inference typically does not require resolving the full bit-growth precision, and prior work shows that maintaining a moderate dot-product compute SNR (e.g., SNR_{col} in the 10-40 dB range, depending on the model and training strategy) is often sufficient to preserve inference accuracy, especially when using quantization-aware training (QAT) [26]. Accordingly, selecting $B_Q = 6$ provides a favorable trade-off between energy/area and accuracy for this proof-of-concept SoC, while leaving clear headroom for future readout-chain improvements if higher effective precision is required. The quantized per-column output $\hat{Q}_c$ is computed as:

$$\hat{Q}_c = \frac{V_{MAC} - V_{ADC}^{L}}{\left(V_{ADC}^{H} - V_{ADC}^{L}\right)/\left(2^{B_Q} - 1\right)}. \tag{2}$$

IV. Multi-bit Mixed-Signal CIM Core with Linear Resistors

This section presents the contribution addressing **Ch. 1**, which introduces a reliability-oriented R-2R MDAC weight cell (MWC) configuration, as illustrated in Fig. 9(a). The proposed MWC stores weight values in fast, high-density SRAM cells and utilizes an R-2R MDAC to enable efficient multi-bit multiplication. This design reduces dependence on wordline activations and mitigates read disturbances. Furthermore, it enhances signal stability and exploits the compact 6T-SRAM cell without sacrificing performance. From an integration standpoint, various HDLR technologies may serve as the computing element, enabling a denser implementation. For instance, the mega-ohm resistor (MOR) [12] integrates a 5 MΩ resistor within a 0.25 µm² area, corresponding to approximately 20 MΩ per 1 µm², which is substantially higher than conventional high-density polysilicon resistors.

A. MDAC weight cell architecture

The MWC employs an R-2R MDAC topology with SRAM-based weight storage. The digital weight (B_W = 6+1 bits) modulates the MDAC conductance, enabling multi-bit multiplication of the input voltage (V_{DAC}) to produce the output

TABLE I
PERFORMANCE ESTIMATION OF THE PROPOSED CIM CORE WITH VARIOUS RESISTIVE TECHNOLOGIES

	Polysilicon (22-nm) (This work)	**MOR [12]**	**WOx [24]**	**RRAM (22-nm) [38]**
Unit Resistance R_U $(M\Omega)$	0.385	7	28	0.03
MWC Area (1 bit - 6 bit) (μm^2)	17 - 120	1 - 8	1 - 8	0.05 - 0.4
Unit Current (μA)*	2.6	0.15	0.036	33
Area Improv. **Power Improv.**	**Baseline**	14× 17×	14× 70×	225× 0.08×

* per MWC, assuming 1V operation, i.e., $I = 1/R_U$

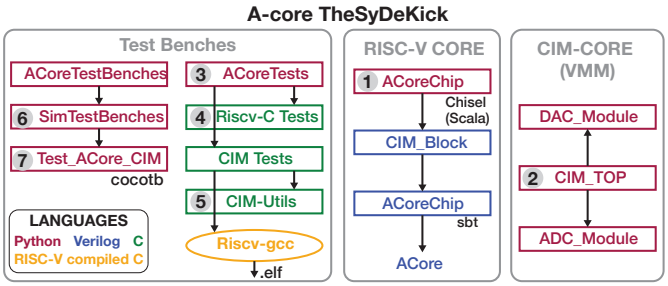


Fig. 10. Illustration of the open-source simulation framework of the proposed Acore-CIM SoC, including the Acore RISC-V core (implemented in RTL), the CIM core (modeled in Python), and various test benches.

current (I_{MAC}). The 6 LSBs ($W_{[5:0]}$) define the effective weight, while two sign bits (W_6, W_7) control polarity and idle mode. These values are written into the local SRAM cells via the memory-mapped register `sram_din`, which configures both weight and sign bits. When both sign bits are 0, the MWC is disabled, reducing leakage and minimizing offset at the 2ΣAmp and ADC. With $W_6 = 1$, the current is directed to the positive summation line (I^{+}_{MAC}); with $W_7 = 1$, to the negative line (I^{-}_{MAC}). This ensures that like-signed currents align across rows. The net column current is therefore:

$$I_{MAC_c} = \sum_{i=0}^{N} \left(\frac{V_{DAC}(i) - V_{BIAS}}{R_U} \cdot \frac{D(i)}{2^{B_W}} \right) \cdot \left(W_6[i] - W_7[i] \right), \quad (3)$$

where V_{BIAS} is the bias voltage, set to 0.4 V to establish the zero level in the analog data path. R_U represents the R-2R MDACs' unit resistance, D is the 6-bit weight value (in integer format) corresponding to the i^{th} cell, and $(W_6\,,\, W_7)$ is the state of the sign bit.

The MWC is highly sensitive to the bias voltage V_{BIAS}, which sets the analog zero level of the MAC operation. Unlike global supply variations, small deviations in V_{BIAS} directly induce systematic offsets at the summation amplifier input, degrading MAC accuracy across columns. As shown in Fig. 9(b), a ± 50 mV shift in V_{BIAS} results in multi-LSB errors in the accumulated MAC output. The integration of BISC compensates for this sensitivity by re-centering the operating point, reducing MAC error by over an order of magnitude.

B. Extension possibility with HDLR technologies

Although the proof-of-concept SoC demonstrates the functionality of the proposed design concepts, the use of polysilicon resistors impacts the area and energy efficiency. However, alternative resistive technologies, such as HDLR materials, offer the potential to significantly improve these metrics. To explore this potential, we evaluated the performance of the proposed MWC using different resistive technologies, taking the unit resistance (R_U) value of the polysilicon-based design as a baseline. For HDLR technologies, we assume that the R-2R resistor network can be post-processed atop an SRAM cell, with a resistor value of $3R_U$ occupying roughly $1\,\mu m^2$. This approach could enable a 128×128 MWC cell array to fit within the same footprint used by the proof-of-concept SoC. Table I summarizes these comparisons and their implications.

a) Polysilicon resistors: The baseline design uses high-density polysilicon resistors fabricated in a 22-nm FD-SOI process. This implementation supports a 36×32 MWC array and provides a reference for evaluating other technologies.

b) Mega-ohm resistor (MOR) technology: MOR technology [12] can achieve a resistance of $5\,M\Omega$ in a $0.25\,\mu m^2$ area, translating to approximately $20\,M\Omega$ per $1\,\mu m^2$. For this study, we assume an R_U of $7\,M\Omega$, reducing current draw to $150\,nA$ per MWC. This results in a 14× improvement in both throughput and area efficiency and a 17× reduction in power consumption, excluding peripherals.

c) WOx resistor technology: WOx-based resistors [24] offer an even higher resistance density. With an estimated R_U of $28\,M\Omega$, each MWC draws only $36\,nA$ of current, yielding up to 70× energy efficiency improvement compared to the baseline, excluding peripheral components.

d) RRAM integration: We also evaluated RRAM, fabricated using the same 22-nm node as the proof-of-concept design [38]. While RRAM provides dense non-volatile memory integration, its current draw per cell ($33\,\mu A$) limits its energy efficiency, resulting in significantly higher power consumption compared to other technologies.

V. OPEN-SOURCE SYSTEM SIMULATION AND MEASUREMENT FRAMEWORK

This section details the contribution that tackles **Ch. 2**. To promote CIM-based systems, extend SoC testing, and enable future integration, we developed a complete open-source simulation framework for the proposed SoC[1]. It relies on the System Development Kit (SyDeKick) environment for system modeling[2]. This simulation framework supports simulation, measurement, and test routines for complex SoCs, where each block can be modeled at different abstraction levels (behavioral, RTL, schematic, or post-layout), tested, and then replaced with the physical design while keeping the same measurement setup. It also supports efficient co-simulation of software and hardware to prepare programming or test routines. All measurements in the following subsections are based on this framework.

[1]The complete source code is available at: https://gitlab.com/a-core/
[2]https://github.com/TheSystemDevelopmentKit

A. Framework description

Fig. 10 depicts the block diagram of the proposed simulation framework, which has three main components: the Acore RISC-V core in RTL, the CIM core in Python, and various test benches. The RISC-V and accelerator communicate over the same AXI4-Lite bus as the fabricated SoC. This interface is implemented in the "CIM_{Block}" file, along with the processor's Verilog code. Because the CIM core is implemented in Python, a co-simulation between the RTL (RISC-V and AXI4-Lite interface) and Python (CIM) is required, which is implemented using CoCoTB[3]. The "$ACoreTestBenches$" file establishes a framework to halt the RTL simulation at each clock cycle and run the "CIM_{top}" file for CIM simulation, enabling bidirectional data transfer via "CIM_{Block}". This approach avoids long CIM simulation times and uses the same test bench (compiled from C code by gcc) as the physical chip, enabling efficient testing of various programs on the system.

B. Detailed simulation operation

Fig. 10 details the main files used in the simulation framework. The explanation below links to their respective markers, denoted with a circle (X). The AcoreChip file includes the complete RTL code for the processor in Chisel and the AXI4 interface to the CIM core (1). This code is compiled into Verilog via sbt, a Scala-based build tool. The CIM core is modeled in Python (2), with separate models for data converters to enable independent testing and rapid prototyping. The model interfaces with the "CIM_{Block}" via the AXI4 interface, so data can be sent to and from the Python model during co-simulation. With SyDeKick functionalities, all CIM core blocks can be replaced by their hardware entities, either at the simulation level (netlist or RTL code) or as the physical SoC itself. The "ACoreTests" file contains all system tests (3), including standard RISC-V tests for the processor (4) and tests for CIM core utility functions (5). Each test's C code can be written and compiled with the RISCV-gcc compiler, producing an *.elf* instructions file, which can also be loaded into the SoC chip's program memory. The "ACoreTestBenches" file compiles all simulation source files and starts the simulation with CoCoTB (6), generating the co-simulation with the RTL test bench and the compiled RISCV-gcc file running the processor code (7). The same test benches are used in simulation and can be directly ported to the physical design, ensuring consistency between pre- and post-silicon validation.

VI. RISC-V-Controlled Built-In Self-Calibration

This section covers **Ch. 3**. As previously discussed, mixed-signal CIM cores experience errors due to process variations, device mismatches, parasitics, and amplifier non-idealities, which degrade MAC accuracy. An on-chip BISC is introduced to systematically identify and calibrate these errors. The BISC method is managed by the RISC-V core and supports automation. This approach enables real-time error correction with minimal system overhead. The complete BISC implementation is available within the open source framework.

[3]https://www.cocotb.org/

A. Error modeling and correction strategy

Without errors, the nominal output of the 2ΣAmp is given by $V_{MAC,nom} = R_{SA}\, I_{MAC} + V_{CAL}$. In reality, the MAC result is affected by analog–domain non-idealities that introduce both gain and offset errors. To model them, let us consider a gain error factor α_A (ideally 1) and an additive offset error β_A, which modify the output as:

$$V_{MAC,act} = \alpha_A\, R_{SA}\, I_{MAC} + V_{CAL} + \beta_A. \tag{4}$$

To determine the appropriate calibration values, we introduce gain and offset correction factors, γ_{SA} and Δ_{CAL}. To compensate for gain error, we define a correction factor γ_{SA} so that $\gamma_{SA}\,\alpha_A = 1$, yielding a calibrated transresistance:

$$R'_{SA} = \gamma_{SA}\, R_{SA} = \frac{R_{SA}}{\alpha_A}. \tag{5}$$

Second, we cancel the offset error by setting $\Delta_{CAL} + \beta_A = 0$, resulting in a corrected calibration voltage:

$$V'_{CAL} = V_{CAL} + \Delta_{CAL} = V_{CAL} - \beta_A. \tag{6}$$

Hence, applying the corrected values R'_{SA} and V'_{CAL} to the actual MAC output (4) via the BISC circuits yields a calibrated output $V_{\text{MAC},BISC}$. This restores the ideal behavior of $V_{\text{MAC},nom}$ over the full range of I_{MAC}. To correct these errors, Fig. 7 shows the 2ΣAmp circuit uses a digital potentiometer in its negative feedback path and a 6-bit voltage-mode R-2R DAC controlled by an up-counter in the positive feedback loop. This design makes R_{SA} and V_{CAL} tunable, as detailed in the next subsection.

B. ADC integration and combined error analysis

When the CIM core is integrated into a system, the MAC outputs are quantized by an ADC. Hence, in a complete ADC-column analysis, the ADC's error must be considered. The nominal ADC transfer function is:

$$Q_{\text{nom}} = C_{ADC}\left[R_{SA}\, I_{MAC} + V_{CAL} - V^L_{ADC}\right], \tag{7}$$

where C_{ADC} is the constant ADC conversion factor defined as $\left(2^{B_Q}-1\right)/\left(V^H_{ADC} - V^L_{ADC}\right)$. In addition to the MAC non-idealities, the ADC itself introduces errors, which are modeled using digital gain factor α_D and offset factor β_D. As a result, the system's response is expressed in a linear form as:

$$\hat{Q}_{\text{act}} = \alpha_D\, C_{ADC}\left[\alpha_A\, R_{SA}\, I_{MAC} + V_{CAL} + \beta_A + \frac{\beta_D}{\alpha_D\, C_{ADC}} - V^L_{ADC}\right]. \tag{8}$$

Since the ADC-column output is measured after digitization, the column's linear error can be expressed by modeling the measured ADC output as:

$$\hat{Q}_{\text{act}} = \hat{g}_{\text{tot}}\, Q_{\text{nom}} + \hat{\epsilon}_{\text{tot}}\,, \tag{9}$$

where $\hat{g}_{\text{tot}}$ and $\hat{\epsilon}_{\text{tot}}$ denote the combined gain and offset errors, arising from the complete CIM column, including the ADC.

By comparing the coefficients of I_{MAC} and the constant terms in (7) and (8), and by setting $V_{\text{CAL}} = V_{\text{ADC}}^{L}$, we obtain:

$$\hat{g}_{\text{tot}} = \alpha_A\,\alpha_D, \quad \text{and} \quad \hat{\epsilon}_{\text{tot}} = \alpha_D\,C_{ADC}\,\beta_A + \beta_D. \tag{10}$$

The equations in (10) show that the overall gain error is the product of the analog and digital gain errors, while the total offset error is the sum of the digital offset and the analog offset, scaled by the factor C_{ADC} and the digital gain.

Assuming that the ADC has been characterized independently (i.e., its gain error α_D and offset error β_D are known), the analog errors (α_A and β_A) of the CIM core outputs can be extracted from the Equations in (10) as:

$$\alpha_A = \frac{\hat{g}_{\text{tot}}}{\alpha_D}\ , \quad \text{and} \quad \beta_A = \frac{\hat{\epsilon}_{\text{tot}} - \beta_D}{\alpha_D\,C_{ADC}}\ . \tag{11}$$

These values serve as correction factors to compute the final calibrated parameters and are used to update the tunable elements in the 2ΣAmp circuit. Specifically, using the corrections from (5) and (6), we obtain:

$$R'_{\text{SA}} = \frac{\alpha_D\,R_{\text{SA}}}{\hat{g}_{\text{tot}}}\ , \quad \text{and} \quad V'_{\text{CAL}} = V_{\text{CAL}} - \frac{\hat{\epsilon}_{\text{tot}} - \beta_D}{\alpha_D\,C_{ADC}}\ . \tag{12}$$

Fig. 11(a) shows the measured BISC-calibrated R_{SA} and V_{CAL} values across the 32 CIM columns. After calibration, R_{SA} is adjusted within $10.2 - 11.4$ kΩ and V_{CAL} within 0.390–0.403 V, compared to the default uncalibrated values of $R_{\text{SA}} = 10.7$ kΩ and $V_{\text{CAL}} = 0.4$ V.

C. BISC routine

Once the theoretical correction factors are determined, the BISC routine must apply the appropriate test vectors to extract the actual errors (the "*online characterization*" phase), then calculate the optimal correction factors based on the previous analysis (the "*online correction*" phase).

1) Online characterization phase: To characterize the complete CIM column (the combined 2ΣAmp + ADC chain), we apply a set of known MAC operations, leading to a known accumulated current $\{I_{\text{MAC}}\}$. We then measure the corresponding digital outputs $\hat{Q}_{\text{act}}$, which are then compared with the nominal outputs $\{Q_{\text{nom}}\}$. Using the linear relationship of (9), we estimate the measured gain and offsets with a least-squares fit over Z test points, yielding:

$$\hat{g}_{\text{tot}} = \frac{Z\sum\left(Q_{\text{nom}}\,\hat{Q}_{\text{act}}\right) - \sum Q_{\text{nom}}\sum\hat{Q}_{\text{act}}}{Z\sum\left(Q_{\text{nom}}\right)^2 - \left(\sum Q_{\text{nom}}\right)^2}\ , \tag{13}$$

$$\hat{\epsilon}_{\text{tot}} = \frac{\sum\hat{Q}_{\text{act}} - \hat{g}_{\text{tot}}\sum Q_{\text{nom}}}{Z}\ . \tag{14}$$

Selecting the test set size (Z) involves a critical trade-off. A full-range sweep, measuring every LSB, yields high accuracy and reveals non-linearity but is too time-consuming for large arrays. A minimal two-point method allows rapid calibration with low overhead but risks missing mid-range distortions and is more prone to noise. To balance these extremes, we use a small set of 4-8 equally spaced test vectors across the dynamic range and perform a least-squares regression. The BISC also allows measuring the output multiple times at each test point to average out random errors, such as thermal and flicker noise.

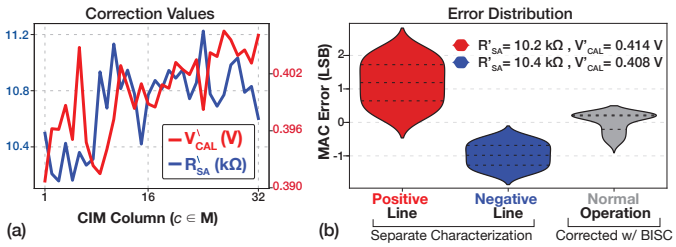


Fig. 11. (a) Measured BISC-calibrated parameters (R_{SA}, V_{CAL}) extracted across all 32-column array. (b) Measured MAC error distribution for a CIM column, comparing normal operation before correction and after BISC calibration. BISC reduces the mean systematic offset and the error variance.

2) Online correction phase: Based on the estimated $\hat{g}_{\text{tot}}$ and $\hat{\epsilon}_{\text{tot}}$, the 2ΣAmp trimming values are updated as per Equation (12). Specifically, the digital potentiometer in the negative feedback path is fine-tuned for gain correction, while the calibration DAC output in the positive feedback loop is adjusted for offset correction. These updates compensate for systematic, linear errors in the analog signal path.

D. Additional considerations for BISC

a) ADC Clipping: Residual errors may cause $V_{MAC,\text{act}}$ to exceed ADC limits, leading to saturation. To ensure valid calibration data and prevent clipping, adjust the ADC reference voltages V_{ADC}^{L} and V_{ADC}^{H} so the full dynamic range, including error margins, stays within the ADC's linear region. For example, if the nominal output $V_{MAC,nom}$ is $0.4-0.6$ V and a ± 5 % margin is expected, set the ADC references to about $(V_{ADC}^{L},\, V_{ADC}^{H}) \approx (0.39\,\text{V},\, 0.61\,\text{V})$ to keep all test points in range.

b) Separate calibration: In the proposed BISC-controlled 2ΣAmp, each amplifier may exhibit distinct nonlinearities due to device mismatch, process variations, and parasitic elements. Therefore, offset and gain errors are measured and corrected independently for each amplifier to achieve precise compensation for the unique non-idealities and asymmetries of each line. Fig. 11(b) illustrates the outcome of this process for a selected CIM column. The "Positive Line" and "Negative Line" distributions display distinct amplifier error profiles prior to calibration. The BISC routine subsequently determines individual trimming values for each amplifier. The "Normal Operation" distribution shows that these adjustments substantially reduce errors, underscoring the need for calibration to achieve precise, robust mixed-signal CIM performance.

The pseudocode of the BISC routine is listed in Algorithm 1. The BISC routine is flexible and can be executed after a system reset, following a classification task, or periodically at predefined intervals, depending on application needs. This demonstrates the seamless integration of BISC with the RISC-V core, ensuring efficient real-time calibration if needed.

E. Mismatch and process variation

Process and device mismatch are major challenges in current-mode CIM architectures because they cause variations

Algorithm 1: BISC calibration routine

Initialization:

Set `RISC-V controls`.

Initialize Calibration Variables:

- $V_{CAL} \leftarrow \frac{V_L+V_H}{2}$, & $R_{SA} \leftarrow \frac{R_U}{N}$.
- $V_{ADC}^{L} \leftarrow 0.95\, V_{ADC}^{L,\text{default}}$.
- $V_{ADC}^{H} \leftarrow 1.05\, V_{ADC}^{H,\text{default}}$.

Store ADC Parameters: $(\alpha_D,\, \beta_D,\, C_{ADC})$.

Define Test Vectors: $\{V_t[Z], W_t[Z]\}$ and Q_{exp}.

Characterization Phase:

foreach *column c in CIM array* **do**

- Write $W_t \leftarrow W_{\max}$. $V_t \leftarrow$ stepped input.
- $\{V_t[Z], W_t[Z]\} \rightarrow \{I_{MAC}[Z]\}$.
- Measure $\hat{Q}_{act}$ per column.
- Compute: $\hat{g}_{tot} \leftarrow$ per Eq.(13), $\hat{\epsilon}_{tot} \leftarrow$ per Eq.(14).

Correction Phase:

foreach *column c in CIM array* **do**

Gain Correction: $R'_{SA} \leftarrow \dfrac{\alpha_D\, R_{SA}}{\hat{g}_{tot}}$.

Offset Correction: $V'_{CAL} \leftarrow V_{CAL} - \dfrac{\hat{\epsilon}_{tot} - \beta_D}{\alpha_D\, C_{ADC}}$.

Update BISC circuits with R'_{SA}&V'_{CAL}.

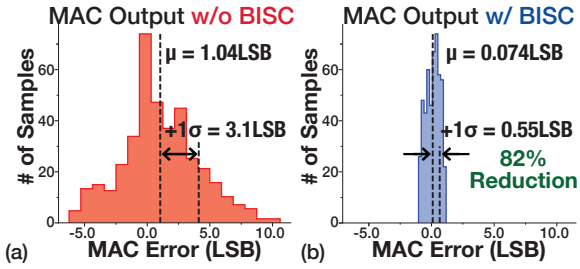


Fig. 12. Post-layout Monte-Carlo simulation of MAC error under mismatch and process variation across 500 iterations: (a) without, and (b) with BISC.

in effective weights and accumulated currents across the array, degrading accuracy. Fig. 12(a) shows post-layout Monte-Carlo results (500 iterations) under mismatch and process variations, where the proposed BISC significantly reduces MAC error spread. BISC compensates for offset shifts from resistor and transistor mismatch and re-centers the summation amplifier's operating point, improving robustness across process corners.

In comparison, prior works on process-variation resilient current-domain CIMs, such as [39], employ additional replica columns and custom wordline drivers to track and compensate for global process drift. Our approach is distinct in that it leverages the existing calibration path to compensate both mismatch and process spread, incurring minimal area/power overhead. As a result, we achieve up to 82% reduction in process-induced spread across the CIM outputs, as quantified in Fig. 12(b).

F. Applicability to other CIM readout architectures

Although demonstrated here on Acore-CIM, the proposed RISC-V-controlled BISC is not tied to a specific memory device. The key requirement is a column readout path that

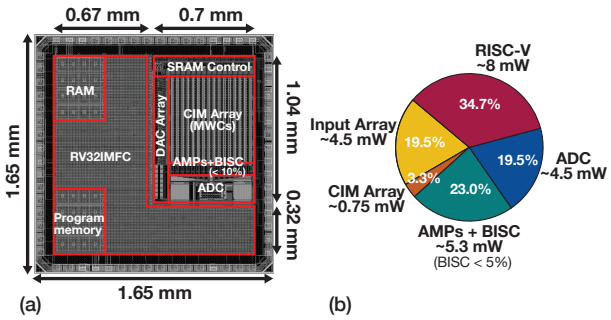


Fig. 13. (a) Acore-CIM chip microphotograph. (b) Power distribution of the Acore-CIM (1152 active MWCs).

can be approximated by a first-order transfer within the operating range, i.e., a systematic offset and gain mapping between the analog accumulated quantity (current/voltage at the sensing node) and the digitized output code. This condition is common in mixed-signal CIM macros that perform column-wise accumulation and digitize the result through an analog front-end (e.g., TIA/summing amplifier) followed by an ADC, where column-to-column mismatch and global process shifts introduce systematic dispersion in the effective transfer characteristics. As a result, a BISC-style characterization and per-column offset/gain compensation can serve as a lightweight software-calibration layer for resistive CIM systems that use current-domain sensing with TIA-based readout and ADC digitization [19], [28], as well as voltage-sensing CIM macros that rely on an analog conditioning/readout chain prior to quantization [10].

VII. Measurement Results of The Acore-CIM Prototype

Fig. 13(a) shows a microphotograph of the prototype SoC chip, fabricated in 22-nm FD-SOI technology. The CIM core occupies $0.73\,\text{mm}^2$ and integrates a 36×32 MWC array, a 2ΣAmp stage, BISC, 36×1 input DACs, SRAM read/write controls, codecs, and a time-multiplexed flash ADC. The RISC-V core and other digital circuits cover an additional $1.14\,\text{mm}^2$. Fig. 13(b) shows the power distribution of the prototype chip. The measurement setup uses a custom PCB connected to a ZC706 FPGA for control, with power supplied and measured by Keysight N6705C DC Power Analyzers.

A. CIM variation and calibration

Table II presents a calibration-centric comparison of state-of-the-art CIM architectures in current- and charge-domain designs. The table contrasts calibration targets, mechanisms, controllers, overheads (Area/Power/Latency/Complexity rated as L/M/H), and peak recovery metrics, reported as accuracy gains, noise reduction (SNR/ENOB), and linearity and offset statistics. Previous strategies typically address a single non-ideality, such as sense amplifier offset, row or column mismatch, or PVT drift, and often rely on replica circuits, dedicated controllers, or offline characterization. In contrast, the Acore-CIM architecture uses a unified BISC scheme that corrects offsets, gain, mismatches, and process variation within

TABLE II
COMPARISON OF CALIBRATION STRATEGIES IN CIM ACCELERATORS: TARGETS, MECHANISMS, OVERHEADS, AND RECOVERY

Work	Weight Memory & MAC Domain	Readout (Quantization)	Calib. Targets	Calib. Mechanism	Controller	Overhead (A/P/L/C)	Max. Recovery†
JSSC'24 [3]	6T-SRAM + LBDAC Current-domain	Analog (no ADC) / 0.83 MHz	Row/col mirror mismatch	Weight pre-distortion* + offline char.	Digital ctrl. + SPI host	– / – / H / M	σ_{offset}: 2.66→0.46 LSB ΔAcc: +1.55 pp
TCAAI'25 [40]	10T1C-SRAM bitcell Charge-domain	DSCI SAR + SA (digital O/P)	SA offset + LF noise	SA offset trim + kickback reduction	Digital FSM + SPI host	L / L / L / L	$\sigma_{\text{INL}} \in [0.3, 3.4]$ LSB 95% offsets $\leqslant$ 1 LSB
JSSC'22 [41]	8T-SRAM bitcell Current-domain	Per-column SA-ADC (1–5 b)	Per-column offset	Binary-search trim + 32 replica BC/col	Column-local digital ctrl.	M / L / M / M	DNL: $[-0.256, +0.314]$ LSB INL: $[-0.116, +0.270]$ LSB
JSSC'24 [18]	SRAM buffers + AMEM Current-domain (ANU)	5b ADC + FIFO (digital O/P)	PVT drift (MAC) & AMEM retention	BG-bias auto-comp (MAC) AMEM write-verify pulse-subtraction VTC	Global digital controller	M / L / H / H	σ_{offset}: 1.9→0.19 LSB ΔAcc: +2.9 pp (±10%V) ΔAcc: +1.3 pp ($-20\sim85^\circ$C)
This SoC	6T-SRAM + R-2R MWC Current-domain	2ΣAmp front-end + TM flash ADC	Per-column offset/gain (process, mismatch, non-idealities)	BISC (per-col calib.: least-squares fitting)	RISC-V (via AXI4-Lite)	M / M / M / M	ΔSNR: +6 dB (+1.1 bit ΔENOB) σ_{offset}: 3.1→0.55 LSB $\sigma_e \in [-0.7, 0.4]$ LSB ΔAcc: +3.63 pp

† Reported values show the peak calibration gain, expressed as accuracy, noise (SNR/ENOB), or linearity/offset statistics (offset, INL/DNL).
* Hybrid calibration: characterization performed off-line.
Legend: A/P/L/C = Area / Power / Latency / Complexity; L/M/H = Low/Medium/High; "–" = not reported.

TABLE III
PERFORMANCE METRICS OF THE ACORE-CIM COMPARED WITH STATE-OF-THE-ART CIM DESIGNS

Work	Tech (nm)	Max Precision (I:W:O)	f_{inf} (MHz)	Norm. 1b-TOPS† $= 2\,B_{\text{in}}^{\parallel} N_{\text{w}}^{\parallel} f_{\text{inf}}$	Norm. Area Eff. (1b-TOPS/mm^2)¶	Norm. Energy Eff. (1b-TOPS/W)¶	MNIST Acc. (%)*
JSSC'24 [3]	180	4:4:Analog	0.83	$2 \times (4) \times (16 \times 16 \times 4) \times 0.83 = 0.0068$	$0.0068/0.1296 = 0.053$	$0.0068/65\,\mu\text{W} = 104.6$	95.6
TCAAI'25 [40]	22	8:4:8	0.85‡	$2 \times (1) \times (1152 \times 256 \times 4) \times 0.85 \approx 0.50$	2.55	125	98.6
JSSC'22 [41]	65	1:1:5	200	$2 \times (1) \times (64 \times 128 \times 1) \times 200 = 3.3$	$3.3/0.065 = 50.8$	$3.3/0.21\,\text{W} = 15.8$	96.4
JSSC'24 [18]	28	5:5:5	2‡	$2 \times (5) \times (9 \times 28 \times 5) \times 2 = 0.025$/Path	$0.025/0.2436 = 0.10$/Path	$0.025/0.502\,m\text{W} = 50$/Path	98.1
This SoC	22	7:7:6	1	$2 \times (7) \times (36 \times 32 \times 7) \times 1 = 0.113$	$0.113/0.728 = 0.155$	$0.113/15\,\text{mW} = 7.5$	97.0

† Normalization: 1b-TOPS $= 2\,B_{\text{in}}^{\parallel} N_{\text{w}}^{\parallel} f_{\text{inf}}$. ‡ Estimated/derived at the stated operating point. ¶ Uses the *active* macro/path/tile (area or power).
* Accuracies after calibration.

the existing 2ΣAmp stage, with minimal analog overhead (area < 10 %, power < 5 %). Combined with RISC-V–based control, this approach enables multi-metric recovery, improving SNR, ENOB, linearity, and inference accuracy, while maintaining moderate overheads. The main advantage of this method is its versatility and system-level efficiency, rather than optimization for a single error source.

Fig. 14(a) compares average CIM macro outputs to ideal MAC values in uncalibrated and BISC-calibrated modes. The uncalibrated outputs show a clear offset from the ideal curve, while the BISC-calibrated results align closely, indicating reduced spatial variation and improved accuracy. Fig. 14(b) presents the per-column transfer error relative to the ideal output code across the applied input sweep points. For each column c, the code error $e_c[n] = \hat{Q}_{c,\text{act}}[n] - Q_{c,\text{nom}}[n]$ is reported in LSB, where 1 LSB corresponds to $3.125\,\text{mV}$. The marker represents the mean of $e_c[n]$ across the sweep points, while the error bar denotes $\pm 1\sigma$. Following BISC, the column-wise mean error is more uniform across the 32 columns (≈ -0.25 LSB), and the spread is reduced ($\approx \pm 0.5$ LSB), which demonstrates improved column-to-column matching of the static transfer characteristics.

While BISC removes the dominant offset and gain errors, the system's final accuracy is limited by the hardware readout chain. Two factors create a residual error floor that calibration cannot fix: the finite resolution and imperfections of the compact flash ADC, and complex higher-order analog distortions that our linear BISC model does not capture. Measurements confirm this limit. Once BISC corrects the major mismatches, further calibration yields diminishing returns as it reaches this hardware noise floor. Future iterations will consider enhanced readout architectures and extend the calibration framework to model and compensate higher-order errors.

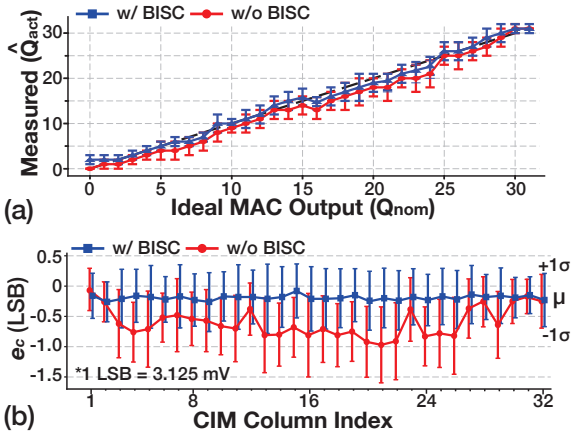


Fig. 14. (a) A measured comparison of spatial variation enhancement across CIM columns without and with BISC. (b) Measured per-column transfer error $e_c[n]$ relative to $Q_{c,\text{nom}}$, before and after BISC.

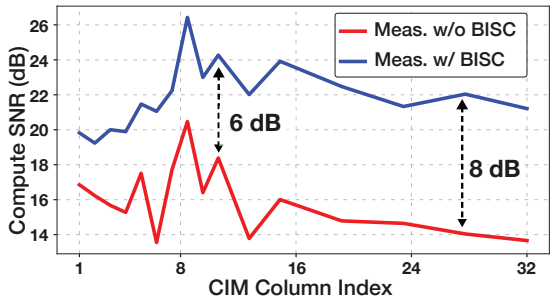


Fig. 15. Compute SNR boost across CIM columns with BISC, achieving an average boost of 6 dB.

B. Compute SNR Evaluation

The compute SNR of the CIM core is evaluated following the method in [15], where the per-column SNR, SNR_c, is defined by:

$$\mathrm{SNR}_c = \frac{\sigma^2_{Q_{\mathrm{nom}}}}{\sigma^2_e}, \quad \text{with} \quad e = \hat{Q}_{\mathrm{act}} - Q_{\mathrm{nom}}. \tag{15}$$

Here, $\sigma^2_{Q_{\mathrm{nom}}}$ represents the variance of the ideal MAC output and σ^2_e the variance of the error signal, where e is the difference between the expected and measured outputs.

Fig. 15 shows the measured SNR across the CIM columns, comparing uncalibrated operation and BISC-calibrated results. As shown, BISC provides up to 8 dB improvement in computational SNR (6 dB on average, with enhancements in every column). Hence, the system's ENOB can attain 4 bits (with average increasing from 2.3 to 3.3 bits).

C. Neural network inference demonstration

To evaluate the system-level impact of BISC on end-to-end inference, we deploy fully connected neural networks (MLPs) for MNIST handwritten digit classification [42]. We first consider a single-hidden-layer MLP (784I–72H–10O). The fixed-point software baseline achieves 94.23 % accuracy. On silicon, running inference without BISC yields 88.7 % accuracy due to column-wise transfer-function mismatch and analog non-idealities. After executing the on-chip BISC routine, which compensates per-column offset and gain by tuning R_{SA} and V_{CAL}, the measured accuracy improves to 92.33 %, corresponding to a 3.63 pp absolute improvement.

We further evaluate a higher-capacity two-hidden-layer MLP (784I–72H–36H–10O). The corresponding fixed-point software baseline is 98.1 % accuracy, while the hardware accuracy with BISC is 97.0 % (Table III). This result is consistent with the measured column compute SNR after BISC (average $\mathrm{SNR_c} \approx 22$ dB, Fig. 15), indicating that the calibration effectively reduces column-to-column variability and enables accurate inference at the system level. At full utilization, the CIM core consumes 16.9 nJ of energy per inference cycle.

D. Chip performance summary

Table III compares the performances of our SoC with state-of-the-art SRAM- and RRAM-based CIM designs. The normalized CIM's throughput is defined as $1\text{b-TOPS} = 2\,B^{\|}_{\mathrm{in}} N^{\|}_{\mathrm{w}} f_{\mathrm{inf}}$, where $B^{\|}_{\mathrm{in}}$ is the number of input bits processed in parallel per inference cycle, $N^{\|}_{\mathrm{w}}$ is the number of weight bit-cells active in parallel across the core per cycle, and f_{inf} is the cycle rate. This explicitly counts weight-bit concurrency within multi-bit MWCs and replicated weight-bit columns. The CIM macro is evaluated at $f_{\mathrm{inf}} = 1$ MHz and achieves a peak throughput of 113 1b-GOPS, with an energy efficiency of 7.5 1b-TOPS/W. When considering the full system—including input generation, weight updates, and output reading via the RISC-V core—the system reaches a peak throughput of 3.05 1b-GOPS and an energy efficiency of 0.122 1b-TOPS/W.

VIII. Conclusion

This work presents Acore-CIM, a 22-nm FD-SOI SoC that addresses the density, integration, and accuracy challenges inherent in mixed-signal CIM architectures. By combining an SRAM-based MDAC compute core with a tightly coupled RISC-V co-processor and an automated calibration flow, we mitigate analog non-idealities and process spread. Measured results show that BISC achieves an average of 22 dB column compute SNR and reduces MAC error spread by 82 %. These results establish a viable path for programmable, robust CIM systems. Future work will focus on enhancing digitization effectiveness, compensating for residual nonlinearities, integrating higher-density linear resistor technologies, and scaling the compute core to support diverse end-to-end DNN workloads.

IX. Acknowledgment

The authors disclose that OpenAI ChatGPT (GPT-5.2 Thinking and GPT-5.3 Thinking) was used to improve the readability and clarity of selected parts of the manuscript, including Algorithm 1, to assist with proofreading, and to help refine the LaTeX formatting of Tables I, II, and III. All scientific content, technical claims, experimental results, source code, figures, and tables were created, checked, and finalized by the authors, who take full responsibility for the content of this article.

References

[1] J.-s. Seo, J. Saikia, J. Meng, W. He, H.-s. Suh, Anupreetham, Y. Liao, A. Hasssan, and I. Yeo, "Digital Versus Analog Artificial Intelligence Accelerators: Advances, trends, and emerging designs," *IEEE Solid-State Circuits Magazine*, vol. 14, no. 3, pp. 65–79, 2022.

[2] B. Murmann, "Mixed-Signal Computing for Deep Neural Network Inference," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 29, no. 1, pp. 3–13, 2021.

[3] S.-Y. Peng, I.-C. Liu, Y.-H. Wu, T.-J. Lin, C.-J. Chen, X.-Z. Li, Y.-Q. Cheng, P.-H. Lin, K.-H. Hung, and Y. Tsao, "An SRAM-Based Reconfigurable Cognitive Computation Matrix for Sensor Edge Applications," *IEEE Journal of Solid-State Circuits*, vol. 59, no. 2, pp. 636–648, 2024.

[4] Z. Jiang, S. Yin, J.-S. Seo, and M. Seok, "C3SRAM: An In-Memory-Computing SRAM Macro Based on Robust Capacitive Coupling Computing Mechanism," *IEEE Journal of Solid-State Circuits*, vol. 55, no. 7, pp. 1888–1897, 2020.

[5] W.-H. Chen, K.-X. Li, W.-Y. Lin, K.-H. Hsu, P.-Y. Li, C.-H. Yang, C.-X. Xue, E.-Y. Yang, Y.-K. Chen, Y.-S. Chang, T.-H. Hsu, Y.-C. King, C.-J. Lin, R.-S. Liu, C.-C. Hsieh, K.-T. Tang, and M.-F. Chang, "A 65nm 1Mb nonvolatile computing-in-memory ReRAM macro with sub-16ns multiply-and-accumulate for binary DNN AI edge processors," in *2018 IEEE International Solid - State Circuits Conference - (ISSCC)*, 2018, pp. 494–496.

[6] H. Valavi, P. J. Ramadge, E. Nestler, and N. Verma, "A 64-Tile 2.4-Mb In-Memory-Computing CNN Accelerator Employing Charge-Domain Compute," *IEEE Journal of Solid-State Circuits*, vol. 54, no. 6, pp. 1789–1799, 2019.

[7] Y. He, J. Yue, X. Feng, Y. Huang, H. Jia, J. Wang, L. Zhang, W. Sun, H. Yang, and Y. Liu, "An RRAM-Based Digital Computing-in-Memory Macro With Dynamic Voltage Sense Amplifier and Sparse-Aware Approximate Adder Tree," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 70, no. 2, pp. 416–420, 2023.

[8] J.-M. Hung, T.-H. Wen, Y.-H. Huang, S.-P. Huang, F.-C. Chang, C.-I. Su, W.-S. Khwa, C.-C. Lo, R.-S. Liu, C.-C. Hsieh, K.-T. Tang, Y.-D. Chih, T.-Y. J. Chang, and M.-F. Chang, "8-b Precision 8-Mb ReRAM Compute-in-Memory Macro Using Direct-Current-Free Time-Domain Readout Scheme for AI Edge Devices," *IEEE Journal of Solid-State Circuits*, vol. 58, no. 1, pp. 303–315, 2023.

[9] M. Giordano, K. Prabhu, K. Koul, R. M. Radway, A. Gural, R. Doshi, Z. F. Khan, J. W. Kustin, T. Liu, G. B. Lopes, V. Turbiner, W.-S. Khwa, Y.-D. Chih, M.-F. Chang, G. Lallement, B. Murmann, S. Mitra, and P. Raina, "CHIMERA: A 0.92 TOPS, 2.2 TOPS/W Edge AI Accelerator with 2 MByte On-Chip Foundry Resistive RAM for Efficient Training and Inference," in *2021 Symposium on VLSI Circuits*, 2021, pp. 1–2.

[10] J.-H. Yoon, M. Chang, W.-S. Khwa, Y.-D. Chih, M.-F. Chang, and A. Raychowdhury, "A 40-nm 118.44-TOPS/W Voltage-Sensing Compute-in-Memory RRAM Macro With Write Verification and Multi-Bit Encoding," *IEEE Journal of Solid-State Circuits*, vol. 57, no. 3, pp. 845–857, 2022.

[11] T. Shimoi, K. Matsubara, T. Saito, T. Ogawa, Y. Taito, Y. Kaneda, M. Izuna, K. Takeda, H. Mitani, T. Ito, and T. Kono, "A 22nm 32Mb Embedded STT-MRAM Macro Achieving 5.9ns Random Read Access and 5.8MB/s Write Throughput at up to Tj of 150 °C," in *2022 IEEE Symposium on VLSI Technology and Circuits (VLSI Technology and Circuits)*, 2022, pp. 134–135.

[12] D. Saito, T. Kobayashi, H. Koga, N. Ronchi, K. Banerjee, Y. Shuto, J. Okuno, K. Konishi, L. Di Piazza, A. Mallik, J. Van Houdt, M. Tsukamoto, K. Ohkuri, T. Umebayashi, and T. Ezaki, "Analog In-memory Computing in FeFET-based 1T1R Array for Edge AI Applications," in *2021 Symposium on VLSI Circuits*, 2021, pp. 1–2.

[13] Le Gallo, et al., "A 64-core mixed-signal in-memory compute chip based on phase-change memory for deep neural network inference," *Nature Electronics*, vol. 6, no. 9, pp. 680–693, 2023.

[14] C. Sakr and N. Shanbhag, "An Analytical Method to Determine Minimum Per-Layer Precision of Deep Neural Networks," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 1090–1094.

[15] N. R. Shanbhag and S. K. Roy, "Benchmarking In-Memory Computing Architectures," *IEEE Open Journal of the Solid-State Circuits Society*, vol. 2, pp. 288–300, 2022.

[16] S. K. Roy *et al.*, "Energy-accuracy trade-offs for resistive in-memory computing architectures," *IEEE Journal on Exploratory Solid-State Computational Devices and Circuits*, vol. 10, pp. 22–30, 2024.

[17] M. E. Sinangil, B. Erbagci, R. Naous, K. Akarvardar, D. Sun, W.-S. Khwa, H.-J. Liao, Y. Wang, and J. Chang, "A 7-nm Compute-in-Memory SRAM Macro Supporting Multi-Bit Input, Weight and Output and Achieving 351 TOPS/W and 372.4 GOPS," *IEEE Journal of Solid-State Circuits*, vol. 56, no. 1, pp. 188–198, 2021.

[18] J.-O. Seo, M. Seok, and S. Cho, "A 44.2-TOPS/W CNN Processor With Variation-Tolerant Analog Datapath and Variation Compensating Circuit," *IEEE Journal of Solid-State Circuits*, vol. 59, no. 5, pp. 1603–1611, 2024.

[19] J. M. Correll, L. Jie, S. Song, S. Lee, J. Zhu, W. Tang, L. Wormald, J. Erhardt, N. Breil, R. Quon, D. Kamalanathan, S. Krishnan, M. Chudzik, Z. Zhang, W. D. Lu, and M. P. Flynn, "An 8-bit 20.7 TOPS/W Multi-Level Cell ReRAM-based Compute Engine," in *2022 IEEE Symposium on VLSI Technology and Circuits (VLSI Technology and Circuits)*, 2022, pp. 264–265.

[20] J.-F. Li, "Testing and Reliability of Computing-In Memories: Solutions and Challenges," in *2022 IEEE International Test Conference in Asia (ITC-Asia)*, 2022, pp. 55–60.

[21] Z. Yan, X. S. Hu, and Y. Shi, "On the reliability of computing-in-memory accelerators for deep neural networks," 2022. [Online]. Available: https://arxiv.org/abs/2205.13018

[22] R. Kier, R. Harrison, and R. Beer, "An MDAC synapse for analog neural networks," in *2004 IEEE International Symposium on Circuits and Systems (IEEE Cat. No.04CH37512)*, vol. 5, 2004.

[23] R. Laajimi, B. Hamdi, and N. Ayari, "A low power bulk-driven MDAC synapse," in *2011 International Conference on Applied Electronics*, 2011, pp. 1–5.

[24] W. Choi, M. Kwak, S. Heo, K. Lee, S. Lee, and H. Hwang, "Hardware Neural Network using Hybrid Synapses via Transfer Learning: WOx Nano-Resistors and TiOx RRAM Synapse for Energy-Efficient Edge-AI Sensor," in *2021 IEEE International Electron Devices Meeting (IEDM)*, 2021, pp. 23.1.1–23.1.4.

[25] C.-J. Jhang, C.-X. Xue, J.-M. Hung, F.-C. Chang, and M.-F. Chang, "Challenges and Trends of SRAM-Based Computing-In-Memory for AI Edge Devices," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 68, no. 5, pp. 1773–1786, 2021.

[26] N. R. Shanbhag and S. K. Roy, "Comprehending In-memory Computing Trends via Proper Benchmarking," in *2022 IEEE Custom Integrated Circuits Conference (CICC)*, 2022, pp. 01–07.

[27] A. Levy, L. R. Upton, M. D. Scott, D. Rich, W.-S. Khwa, Y.-D. Chih, M.-F. Chang, S. Mitra, B. Murmann, and P. Raina, "EMBER: Efficient Multiple-Bits-Per-Cell Embedded RRAM Macro for High-Density Digital Storage," *IEEE Journal of Solid-State Circuits*, vol. 59, no. 7, pp. 2081–2092, 2024.

[28] S. D. Spetalnick *et al.*, "A 40-nm Compute-in-Memory Macro With RRAM Addressing IR Drop and Off-State Current," *IEEE Solid-State Circuits Letters*, vol. 7, pp. 10–13, 2024.

[29] S.-T. Wei, B. Gao, D. Wu, J.-S. Tang, H. Qian, and H.-Q. Wu, "Trends and challenges in the circuit and macro of RRAM-based computing-in-memory systems," *Chip*, vol. 1, no. 1, p. 100004, 2022.

[30] M. Neseem, C. McCullough, R. Hsin, C. Leichner, S. Li, I. S. Chong, A. Howard, L. Lew, S. Reda, V.-M. Rautio, and D. Moro, "PikeLPN: Mitigating Overlooked Inefficiencies of Low-Precision Neural Networks," in *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024, pp. 15 996–16 005.

[31] G. Kang, D. Park, H. Lee, S. Jung, J. Park, J. G. Min, Y. Lee, and J. Kung, "RISC-V Driven Orchestration of Vector Processing Units and eFlash Compute-in-Memory Arrays for Fast and Accurate Keyword Spotting," in *Proceedings of the 30th Asia and South Pacific Design Automation Conference*, ser. ASPDAC '25. New York, NY, USA: Association for Computing Machinery, 2025, p. 1174–1180.

[32] Y.-C. Guo *et al.*, "CIMR-V: An End-to-End SRAM-based CIM Accelerator with RISC-V for AI Edge Device," in *2024 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2024, pp. 1–5.

[33] K. Ueyoshi, I. A. Papistas, P. Houshmand, G. M. Sarda, V. Jain, M. Shi, Q. Zheng, S. Giraldo, P. Vrancx, J. Doevenspeck, D. Bhattacharjee, S. Cosemans, A. Mallik, P. Debacker, D. Verkest, and M. Verhelst, "DIANA: An End-to-End Energy-Efficient Digital and ANAlog Hybrid Neural Network SoC," in *2022 IEEE International Solid- State Circuits Conference (ISSCC)*, vol. 65, 2022, pp. 1–3.

[34] S. Jain, A. Sengupta, K. Roy, and A. Raghunathan, "RxNN: A Framework for Evaluating Deep Neural Networks on Resistive Crossbars," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 40, no. 2, pp. 326–338, 2021.

[35] A-core, an open-source risc-v processor implementation gitlab repository. [Online]. Available: https://gitlab.com/a-core

[36] Chisel, software-defined hardware. [Online]. Available: https://www.chisel-lang.org/

[37] K. Adam, D. Monga, O. Numan, G. Singh, K. Halonen, and M. Andraud, "Evaluating an Analog Main Memory Architecture for All-Analog In-Memory Computing Accelerators," in *2024 IEEE 6th International Conference on AI Circuits and Systems (AICAS)*, 2024, pp. 248–252.

[38] O. Golonzka *et al.*, "Non-Volatile RRAM Embedded into 22FFL FinFET Technology," in *Symposium on VLSI Technology*, 2019, pp. 230–231.

[39] K. Prasad, S. Shubham, A. Biswas, and J. Mekie, "Process Variation Resilient Current-Domain Analog In Memory Computing," in *2023 Design, Automation Test in Europe Conference Exhibition (DATE)*, 2023, pp. 1–2.

[40] A. Kneip, M. Lefebvre, P. Maistriaux, and D. Bol, "Imagine: An 8-to-1b 22nm fd-soi compute-in-memory cnn accelerator with an end-to-end analog charge-based 0.15-8pops/w macro featuring distribution-aware data reshaping," *IEEE Transactions on Circuits and Systems for Artificial Intelligence*, pp. 1–14, 2025.

[41] C. Yu, T. Yoo, K. T. C. Chai, T. T.-H. Kim, and B. Kim, "A 65-nm 8T SRAM Compute-in-Memory Macro With Column ADCs for Processing Neural Networks," *IEEE Journal of Solid-State Circuits*, vol. 57, no. 11, pp. 3466–3476, 2022.

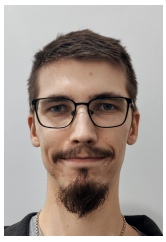
[42] L. Deng, "The mnist database of handwritten digit images for machine learning research," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 141–142, 2012.

Omar Numan (Student Member, IEEE) received his M.Sc. degree in Micro- and Nano-electronic Circuit Design from Aalto University, Finland, in 2020, and is currently pursuing a Ph.D. at the same institution. His research focuses on the design of advanced analog and mixed-signal integrated circuits, with a particular emphasis on compute-in-memory architectures for energy-efficient AI accelerators. He has contributed to the development of low-power, high-performance circuit designs that integrate memory and processing for next-generation AI hardware.

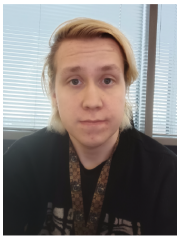
Gaurav Singh (Student Member, IEEE) earned his M.Sc. degree in VLSI from Amity University, Noida, India, in 2015. He is currently advancing his doctoral studies at the Department of Electronics and Nanoengineering, Aalto University, Finland. His research focuses on the development of low-power system-on-chip (SoC) solutions for sensor data processing. His work particularly explores microprocessors, low-power SRAM memories, and serial interfaces to enhance communication and debugging. His research work also includes integrating RISC-V processors with Compute In-Memory cores as hardware accelerators, aimed at improving power efficiency in AI applications.

Kazybek Adam (Student Member, IEEE) received his M.Sc. degree in Electrical and Electronic Engineering from Nazarbayev University, Kazakhstan, in 2019. He is currently pursuing his doctoral studies at the Department of Electronics and Nanoengineering, Aalto University, Finland. His research is in analog circuit design for AI accelerators with focus on Sample and Hold circuits, analog memories, and noise analysis.

Jelin Leslin (Student Member, IEEE) is a doctoral candidate at Aalto University specializing in hardware-aware AI models. He earned a master's degree in Electrical Engineering in 2020 from the Technical University of Eindhoven and KTH Royal Institute of Technology. His research focuses on energy-efficient AI model implementations through model compression, hardware-aware training, and custom hardware design. During his master's, he worked with Volvo Trucks on hardware acceleration for motion control in autonomous vehicles, which was also the focus of his thesis.



Aleksi Korsman (Student Member, IEEE) received his M.Sc. degree in Micro- and Nanoelectronic Circuit Design from Aalto University, Finland, in 2022. He is currently pursuing a Ph.D. degree at the same institution. His research interests include integrating hardware accelerators with microprocessors, especially for signal processing applications.



Otto Simola (Student Member, IEEE) received his M.Sc. degree in Micro- and Nanoelectronics Circuit Design from Aalto University, Finland, in 2023. Currently, he is pursuing his Ph.D. degree at the institution. His research topic is programmatic circuit design especially relating to processor hardware acceleration and digital signal processing applications.

Marko Kosunen (S'97−M'07) received his M.Sc, L.Sc and D.Sc (with honors) degrees from Helsinki University of Technology, Espoo, Finland, in 1998, 2001 and 2006, respectively. He is currently an Associate Professor at Aalto University, Department of Electronics and Nanoengineering. Academic years 2017-2019 he visited Berkeley Wireless Reserarch Center, UC Berkeley, on Marie Sklodowska-Curie grant from European Union. In addition to his academic duties, currently he is one of the three co-chairs of Microelectronics Finland, a academia-industry collaboration organization for microelectronics research and education. He has authored and co-authored more than hundred journal and conference papers and holds several patents. His current research interests include programmatic circuit design methodologies, digital intensive time-based data converters and transceiver circuits, and RISC-V microprocessor implementations with DSP accelerators.

Jussi Ryynänen (S'99−M'04−SM'16) was born in Ilmajoki, Finland, in 1973. He received the M.Sc. and D.Sc. degrees in electrical engineering from the Helsinki University of Technology, Espoo, Finland, in 1998 and 2004, respectively. He is a full professor and the Dean of the School of Electrical Engineering, Aalto University, Espoo. He has authored or co-authored more than 200 refereed journal and conference papers in analog and RF circuit design. He holds seven patents on RF circuits. His research interests are integrated transceiver circuits for wireless applications. Prof. Ryynänen is currently an SG Member for the European Solid-State Circuits Conference (ESSCIRC) and the IEEE Nordic Circuits and Systems Conference (NORCAS). He has served as a TPC member of the IEEE International Solid-State Circuits Conference (ISSCC) and as a Guest Editor for the IEEE Journal of Solid-State Circuits.

Martin Andraud (Member, IEEE) is an assistant professor in microelectronics at UCLouvain, Belgium, and a visiting professor at Aalto University, Finland. He received his PhD from Grenoble University, France, in 2016. He was a postdoctoral researcher successively with TU Eindhoven in 2016 and KU Leuven from 2017 to 2019. Between 2019 and 2024, he was an assistant professor at Aalto University, Finland. His research interests include the interface between edge AI, hardware/software co-design, testing, and reliability of custom ASIC for various AI accelerators, for instance, mixed-signal Compute-In-Memory architectures and alternative to deep learning models (probabilistic reasoning or hybrid AI).