C3CIM: Constant Column Current Memristor-based Computation-in-Memory Micro-architecture

Yashvardhan Biyani, Rajendra Bishnoi, Theofilos Spyrou and Said Hamdioui
Computer Engineering Lab, Delft University of Technology, Delft, The Netherlands
Email: {Y.Biyani, R.K.Bishnoi, T.Spyrou, S.Hamdioui}@tudelft.nl

***Abstract*—Advancements in Artificial Intelligence (AI) and Internet-of-Things (IoT) have increased demand for edge AI, but deployment on traditional AI accelerators, like GPUs and TPUs, using von Neumann architecture, suffer from inefficiencies due to separate memory and compute units. Computation-in-Memory (CIM), utilizing non-volatile memristor devices to leverage analog computing principles and perform in-place computations, holds great potential in improving computational efficiency by eliminating frequent data movement. However, standard implementation of CIM faces several challenges, primarily high power consumption and subsequently induced non-linearity, debating its viability for edge devices. In this paper, we propose C3CIM, a novel memristor-based CIM micro-architecture, featuring a new bit-cell and array design, targeting efficient implementation of Neural Networks (NN). Our architecture uses a constant current source to perform Multiply-and-Accumulate (MAC) operations with a very low computation current (10 to 100 nA), thereby significantly enhancing power efficiency. We adapted C3CIM for Spiking Neural Networks (SNN) and developed a prototype using TSMC 40nm CMOS node for on-silicon validation. Furthermore, our micro-architecture was benchmarked using two SNN models based on N-MNIST and IBM-Gesture datasets, for comparison against current state-of-the-art (SOTA). Results show up to 35x reduction in power along with 6.7x saving in energy compared to SOTA, demonstrating promising potential of this work for edge AI applications.**

I. Introduction

Recent advancements in AI together with the rise of IoT have led to a significant demand for edge AI, however, traditional AI accelerators like GPUs and TPUs employ the von-Neumann architecture. The inefficiencies resulting from separate memory and computation units demands for 100-1000x energy efficient devices in order to practically deploy AI at the edge [3]. Additionally, CMOS memory technology struggles with leakage and scalability issues [4]. Memristor-based CIM architecture [5]–[7] aims to solve this challenge by performing computations directly within the memory minimizing data movement and thereby enhancing efficiency [8]. Here, emerging non-volatile memristor devices are used not only to store weights (data), but also perform dot product operations simultaneously with the corresponding values of input vectors. The partial results are then typically accumulated using analog techniques. Nevertheless, large-scale deployment of such architectures diminishes energy-efficiency benefits as the generated crossbar currents might be arbitrarily high. It further introduce challenges such as non-linearity [9] and read-disturb, potentially impacting the neural network accuracy.

Several strategies have been proposed to reduce power consumption, such as using lower absolute memristor conductances [10], [11] or lowering read voltages [12]. Low conductance levels are undesirable due to their high susceptibility to device-to-device as well as cycle-to-cycle variations [13], [14], inducing noise into the system and impacting the overall performance. On the other hand, the technique of reducing read voltages relies heavily on peripheral circuits and faces limitations, particularly in maintaining voltage stability. Moreover, the crossbar current still remains dependent on the inputs and weights. Consequently, higher currents that may arise will not only lead to increased power consumption but also introduce non-linearity in the output due to significant IR drops in the Source-lines (SL) and Bit-lines (BL). Furthermore, another work [15], [16] attempts to limit the current by utilizing a constant current to drive the crossbar column. However, the limited margins do not allow high scalability at the array-level, which still results in increased energy consumption. Therefore, there is a decisive need for an energy-efficient as well as cost-effective approach to CIM, capable of delivering linear MAC outputs.

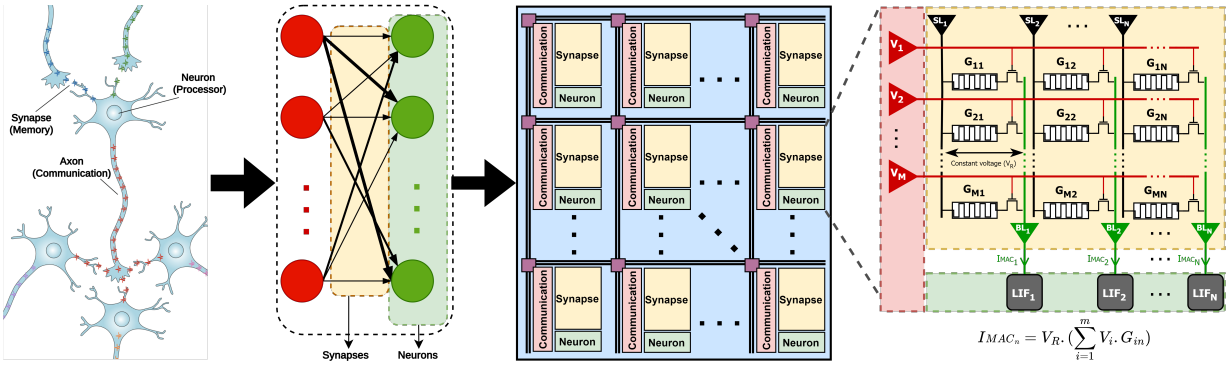
In this paper, we present a novel approach to CIM that utilizes a constant current source to perform MAC operations via a serial arrangement of memory cells in a column. Within our proposed crossbar, we employ a very low current (in the order of sub-μA) for the computation, which remains independent of input values, thereby enhancing overall energy-efficiency. The contributions are as follows.

- C3CIM: A constant column current based CIM crossbar, utilising a new 2T1R bit-cell design, to enable ultra-low power MAC operations
- Development of a SNN micro-architecture based on C3CIM to validate its functionality
- Prototype at TSMC 40nm CMOS node for on-silicon demonstration and validation of C3CIM
- Comprehensive system-level simulations using custom-developed SNN models to benchmark the proposed work against SOTA

Benchmark results from deploying custom-developed Le-Net [17] style SNN models, trained on the N-MNIST [18] and IBM-Gesture [19] datasets, indicate an overall power reduction of up to 35x and energy savings of around 6.7x compared to conventional CIM architectures. The measurement results

This work is partially funded by the European Union, CONVOLVE (Grant No. 101070374), NEUROKIT2E (Grant No. 101112268) and Ferro4EdgeAI (Grant No. 101135656).

978-3-9826741-0-0/DATE25/© 2025 EDAA



(a) (b) (c) (d)

Fig. 1. SNN and its implementation with CIM (a) Biological neuron [1]; (b) Spiking Neural Network; (c) Typical Neuromorphic hardware consisting of an array of CIM crossbar tiles [2]; (d) RRAM-based 1T1R CIM crossbar

from the prototype not only validate the micro-architecture's functionality on silicon but also demonstrate its potential for further current reduction and lower-power operations.

The rest of this paper is organized as follows. Section II summarises the fundamentals of SNN and CIM. Section III explains the concept of C3CIM, followed by detailed description of the proposed SNN micro-architecture in Section IV. Section V provides the simulation results along with discussion on the proposed work. Section VI and VII presents the prototype characterisation and system-level results respectively. Finally, Section VIII concludes the paper.

II. Background

A. CIM-based SNN

SNNs [20] are a brain-inspired class of neural networks that aim to mimic the behavior of biological neurons, as illustrated in Figure 1a. With the goal of achieving low-power implementation of AI akin to the human brain, SNNs, similar to their biological counterparts, transmit information through discrete spike events over time. This approach enables them to capture and utilize the temporal dynamics of the input rather than its absolute magnitude. The spikes then travel through a network of synapses or weighted paths, where they are processed as they propagate towards several other connected neurons at the end. The event-driven and asynchronous nature of SNNs not only boosts computational efficiency by exploiting sparse activity, but also renders them suitable for implementation on CIM architectures owing to reduced peripheral overhead. The inputs (spikes), being binary in nature, can be directly transmitted to a CIM crossbar without requiring specialized peripheral circuits. The weighted paths or synapses can be emulated using the crossbar structure, as shown in Figure 1d, where the MAC operations occur between the input spikes and weights. The resulting current can directly drive a standard Leaky Integrate-and-Fire (LIF) circuit [21], which is a simple circuit implementation of the LIF activation commonly employed in SNNs. This is in contrast to Artificial Neural Networks (ANNs) implemented on CIM architectures, which typically require Digital-to-Analog Converters (DACs) and Analog-to-Digital Converters (ADCs) to process the input and output respectively. These peripheral components often incur significant power and area overhead, diminishing benefits of CIM [22]. Subsequently, SNNs as an application complement CIM architectures, fully harnessing their advantages, albeit at the cost of increased computational latency.

B. Memristor-based CIM architecture

A memristor-based CIM architecture aims to perform computations within the memory using emerging memristor technologies [23] such as Resistive Random Access Memory (RRAM) [24]. A standard CIM architecture consists of a memristor crossbar, accompanied by suitable peripheral circuits to handle data input and output. By leveraging analog computing, MAC operations, fundamental to AI, can be directly performed within the memory crossbar as a single operation in the analog domain. Moreover, it allows for fully parallel execution of multiple MAC operations, limited only by the crossbar size or the complexity of the supporting peripheries. This makes CIM instrumental in mitigating memory bottlenecks associated with traditional von Neumann architectures, thereby enabling energy-efficient edge AI. Figure 1d shows a typical RRAM-based 1T1R CIM crossbar, configured for binary inputs. A RRAM operates on the principle of reversible conductive filament formation, allowing it to store data in the form of discrete conductance states. In practice, it has a low conductance state (LCS) and a high conductance state (HCS) representing bit '0' and bit '1' respectively, with the potential of having intermediate conductance states for storing multi-bit data. By appropriately programming the conductance of bit cells, neural network weights are stored within the crossbar. When inputs are applied as analog voltages, the resulting bit-cell current follows Ohm's law, being the product of the applied voltage and the programmed conductance. This constitutes a multiplication operation between the inputs and the weights. Furthermore, the parallel arrangement of bit-cells in a column causes the currents to accumulate at their respective BL and produce a final output current, thus performing addition and completing the MAC operation. Conventionally, DACs at the input and ADCs at the output support this simplified CIM demonstration, as inputs and outputs are usually sourced from or stored in digital memory. However, this requirement may vary depending on the data flow or the application.

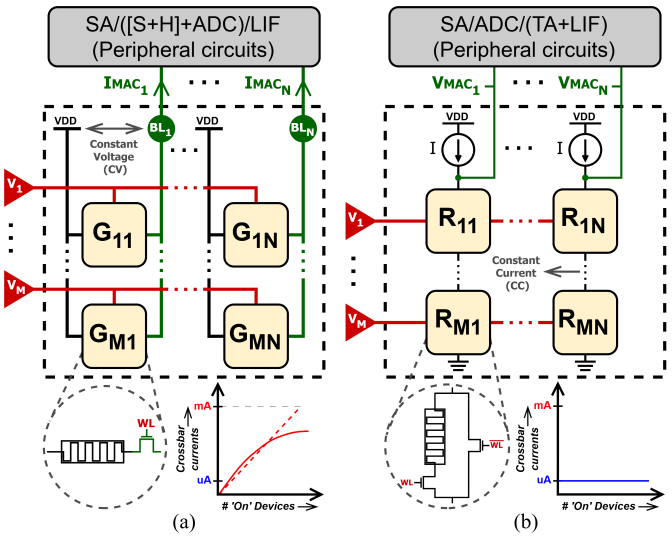


Fig. 2. Overview of CIM crossbar design (a) Conventional crossbar design; (b) Proposed crossbar design

III. C3CIM OVERVIEW

A. Conventional CIM

Figure 2a depicts a conventional CIM crossbar performing MAC operations between binary inputs and weights. It typically employs 1T1R bit-cells connected in parallel within a column, across their respective SL and BL maintained at a fixed voltage difference. Here, the binary voltages applied at the Word line (WL) serve as the input vector while the programmed conductance's of the memristor array represent the weight matrix. The resulting vector of currents generated at the BLs is then analogous to the Vector-Matrix Multiplication (VMM) operation between the input vector and the weight matrix. Since the magnitude of the crossbar currents represent the outcome of the VMM operation, the power consumption is non-deterministic and the crossbar can potentially have arbitrarily high power consumption for large outputs. Furthermore, these large currents cause other non-ideal effects, such as IR drops, power saturation, etc. inducing non-linearity into the system, as shown.

B. Proposed C3CIM

In contrast, the proposed crossbar design, shown in Figure 2b, features 2T1R bit-cells arranged in series within a column. The input vector, in this case, is represented by binary voltages applied simultaneously in a complimentary manner to both WL and WLbar of the corresponding rows of bit-cells, while the programmed resistances of the memristor array serve as the weight matrix. Unlike the former, each column is subjected to a constant current to perform the same VMM operation, but this time as a proportional voltage output at the top of each column. This approach not only overcomes the aforementioned challenged of the conventional CIM crossbar but also shows promise in improving power efficiency for MAC operations by up to two orders of magnitude.

IV. C3CIM DESIGN FOR SNN

To illustrate this improvement, we introduce a micro-architecture based on C3CIM crossbar, designed to support SNNs. As shown in Figure 3, the micro-architecture incorporates three stages to replicate the functionality of an SNN neuron, which is discussed further in detail.

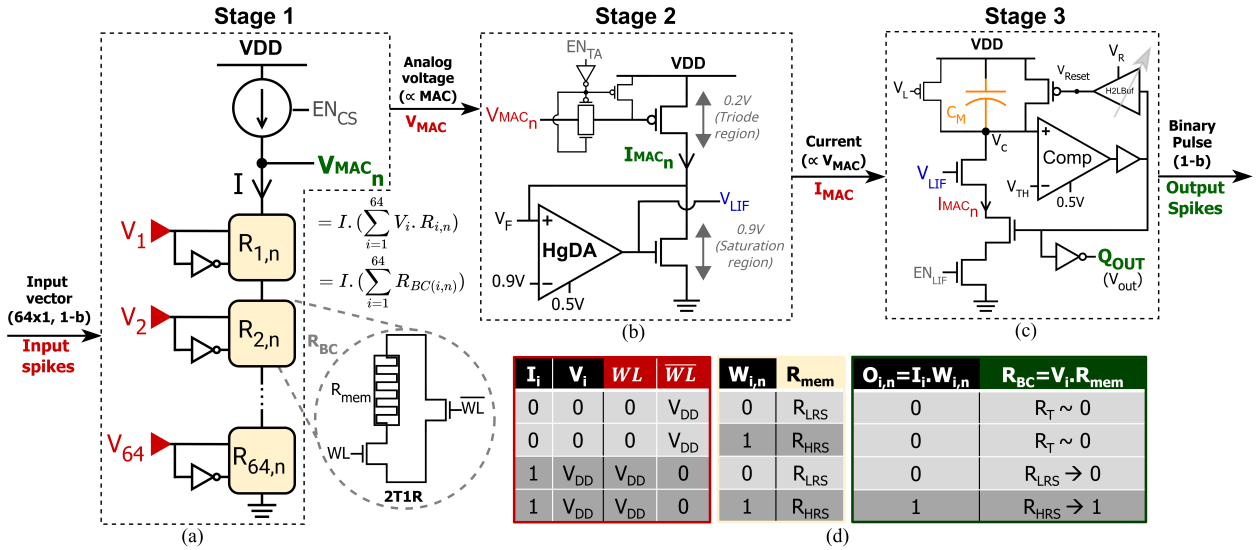
A. Stage I - C3CIM crossbar

Figure 3a illustrates a column of the C3CIM crossbar, using a 2T1R bit-cell configuration. The bit-cell, as shown, consists of two parallel paths: the memristive path, which includes a memristor (R_{mem}) that stores binary data, and the auxiliary path. Both paths further consist of pass transistors, whose gates are controlled via the Word-line (WL) for the memristive path and the Word-line bar (WLb) for the auxiliary path, enabling switching between the paths by appropriately controlling the transistor gates.

When complementary binary signals are applied as input to the bit-cell via WL and WLb, this ensures only one path is active at a given time. With this arrangement, the effective resistance of the bit-cell (R_{BC}) can be determined by the truth table shown in Figure 3d. If the input is 0, the inverted signal at WLb activates the auxiliary path. Assuming the "on" resistance of the pass transistors (R_T) is negligible, the activation of the auxiliary path causes R_{BC} to be same as (R_T), thereby rendering it negligible as well. Conversely, when the input is 1, the memristive path is activated. In this case, R_{BC} depends primarily on the state of the memristor. If the memristor stores a 0, corresponding to a low resistance state (R_{LRS}), R_{BC} remains negligible. However, if the memristor stores a 1, analogous to a high resistance state (R_{HRS}), R_{BC} becomes significant.

Subsequently, this process constitutes a multiplication operation between the input and weight in the form of altered bit-cell resistance. The resistance is significant only when both input and weight are 1. Multiple such operations can be performed in parallel by providing inputs to each row of bit-cells via their respective WL and WLb. Connecting bit-cells in series within each column leads to the accumulation of their effective resistances, resulting in an overall column resistance that is linearly proportional to the output of the MAC operation between the input and weight vectors. Finally, by inducing a constant current into the column through a current source at the top, enabled via EN_{CS}, the column resistance can be read out as a proportional voltage output (V_{MAC}), in accordance with Ohm's law.

Since the column current is fixed by design, the power consumed by the crossbar is now constant and independent of the state of the crossbar, making it deterministic. By maintaining extremely low column currents, such as 100 nA in this case, this crossbar can achieve MAC operations with very high power efficiency while also mitigating the problem of non-linearity. Moreover, the 2T1R bit-cell configuration inherently addresses the non-zero 'on' resistance of the pass transistors, as one of the two paths is always active regardless of the input state. With identical pass transistors in both the paths, the total number of transistors in the current path remains fixed, at 64 in this case. Consequently, the cumulative voltage drop due to these transistors is nearly constant and input-independent, allowing it to be treated as a fixed offset for easier post-processing. Furthermore, the input loading is negligible since the inputs are applied at the gate of the pass transistors. This makes



I_i	V_i	WL	$\overline{WL}$
0	0	0	V_{DD}
0	0	0	V_{DD}
1	V_{DD}	V_{DD}	0
1	V_{DD}	V_{DD}	0

$W_{i,n}$	R_{mem}
0	R_{LRS}
1	R_{HRS}
0	R_{LRS}
1	R_{HRS}

$O_{i,n}=I_i.W_{i,n}$	$R_{BC}=V_i.R_{mem}$
0	$R_T \sim 0$
0	$R_T \sim 0$
0	$R_{LRS} \rightarrow 0$
1	$R_{HRS} \rightarrow 1$

(d)

Fig. 3. Hardware implementation of a SNN neuron; (a) C3CIM crossbar column; (b) Trans-conductance amplifier (TA); (c) LIF circuit; (d) 2T1R bit-cell truth table

the crossbar design flexible as any number of columns can be accommodated in parallel, by having buffers at appropriate intervals to regenerate the binary signals.

B. Stage II - Trans-conductance amplifier (TA)

As illustrated in Figure 3b, the TA serves as an intermediate stage to facilitate the C3CIM crossbar, which generates output in the form of voltages, for implementing SNN. When enabled via EN_{TA}. the amplifier is designed to generate a current proportional to the applied input voltage, which is the V_{MAC} from the previous stage. It is not required in the case of conventional crossbars, which generate outputs directly as currents, but at the expense of high power consumption

The desired functionality is achieved by utilising a MOS transistor operating in the triode/linear region, where the channel current varies linearly with the gate-to-source voltage (V_{GS}), provided the drain-to-source voltage (V_{DS}) is held constant. By maintaining a suitably low V_{DS}, such as 200 mV across the PMOS transistor in this case, the device is constrained to operate within the triode/linear region. To accomplish this, an NMOS transistor operating in the saturation region is connected in series with the PMOS transistor, with their drains tied together at a common node (V_F). A differential amplifier actively monitors this node in a negative feedback configuration and adjusts the gate voltage of the NMOS transistor (V_{LIF}) such that V_F remains close to the externally applied reference voltage of 900 mV.

The output voltage from the previous stage (V_{MAC}) is then applied to the gate of the PMOS transistor, while its source is held at a fixed supply voltage of 1.1V (V_{DD}). This causes the PMOS transistor to generate a current that is linearly proportional to its gate-to-source voltage (V_{GSP}), and by extension, to its gate voltage or V_{MAC}. Since the NMOS transistor operates in the saturation region with its source grounded, its current is mainly governed by its V_{GS} or V_{LIF} and remains largely unaffected by its V_{DS} or V_F. Moreover, due to the series connection of the transistors, V_{LIF} must be generated such that the induced current in the NMOS transistor matches that of the PMOS transistor. Finally, this V_{LIF} allows the matched current, which is linearly proportional to V_{MAC}, to be mirrored into the LIF circuit in the next stage via current mirroring techniques for further processing.

C. Stage III - LIF circuit

Stage 3 consists of a LIF circuit, as illustrated in Figure 3c, which aims to mimic the behavior of a biological neuron. On enabling the circuit via EN_{LIF}, a membrane capacitor (C_{mem}) is charged at a rate proportional to the magnitude of the induced current, effectively integrating the current over time as a voltage across the capacitor (V_C). Since the induced current is mirrored from the previous stage (I_{MAC_n}), it is proportional to the output of the MAC operation. As a result, the capacitor integrates charges that are proportional to the MAC operation in a given computation cycle, when integrated for a fixed time.

When V_C reaches a predetermined threshold (V_{TH}), the circuit generates a binary pulse at the output or fires an output spike, which then serves as input to subsequent neurons. This is followed by resetting the membrane capacitor to its initial value after a specified refractory period. The refractory period, during which the circuit ignores any new input, is adjustable via V_R and governs the high-to-low transition time of a buffer. The leaky behavior is emulated by an additional transistor operating in saturation, which continuously induces a small, constant current into the capacitor determined by its gate voltage (V_L). This circuit effectively implements the LIF model in hardware enabling the implementation of SNN using CIM paradigm.

D. Design optimisation: Charge pump

The serial arrangement of bit-cells in the column results in significant RC loading, causing the output voltage to take approximately 500 ns (Figure 4b) to settle to its final value. This latency is nearly two orders of magnitude greater than the SOTA solutions, which, despite the high power efficiency of our micro-architecture, could negate any energy consumption benefits and potentially worsen overall performance. To address this issue, we incorporated a charge pump mechanism designed

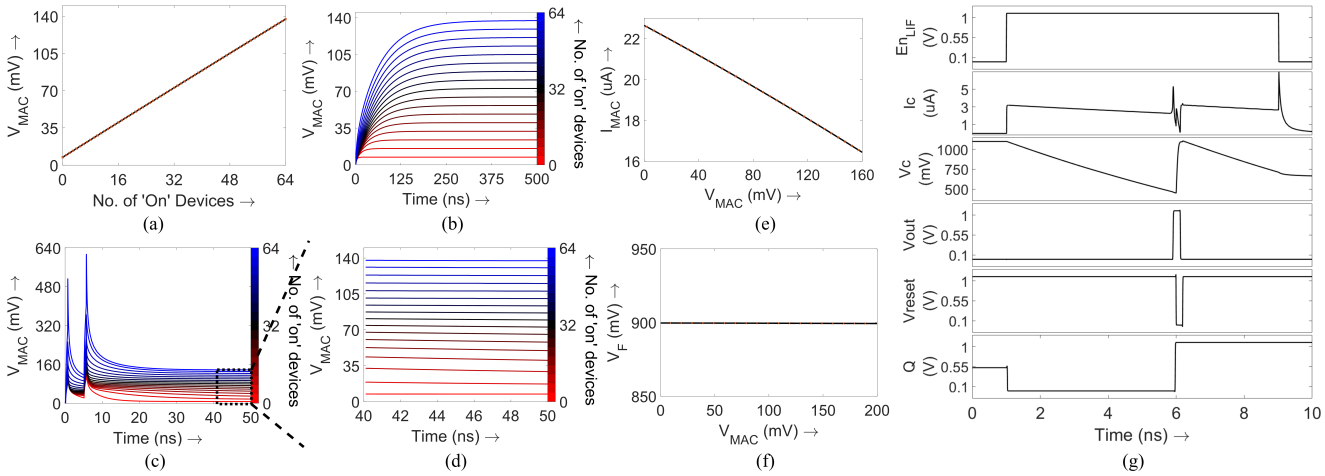


Fig. 4. Simulation plots of the proposed, C3CIM-based, SNN micro-architecture (a) V_{MAC} v/s No. of 'on' devices; (b) V_{MAC} v/s time (with charge-pump) (c) V_{MAC} v/s time (w/o charge-pump); (d) Zoomed-in view of the plot in (b); (e) I_{MAC} v/s V_{MAC}; (f) V_F v/s V_{MAC}; (g) Timing diagram of the LIF circuit

to deliver high-current pulses for very short durations. These pulses are delivered from the top of the crossbar column, from where the constant current is also induced. This rapidly elevates the output voltage before allowing it to stabilize at a lower value, significantly accelerating the process. As a result, latency is reduced by an order of magnitude to 50 ns, as shown in Figure 4c. Consequently, our design now surpasses SOTA solutions not only in power efficiency but also in overall energy consumption.

V. Circuit-level Simulations

This section presents the per-stage SPICE-level behavior and performance of the proposed C3CIM micro-architecture for SNN.

A. Simulation setup

TABLE I
SIMULATION SETUP SPECIFICATIONS

Parameters	Specifications
RRAM device	HfO_x [25]
HRS/LRS	20 KΩ/2 KΩ
Devices per column	64
Read current	100 nA
Voltage supply	1.1 V
CMOS technology	TSMC 40 nm
Process corner	Typical
Temperature	27°C

Table I summarizes the design specifications for the circuit-level analysis. For simulation purposes, the crossbar column consists of 64 bit cells in series, i.e. 64 rows, in the previously proposed 2T1R configuration. The column current source is designed to provide a constant current of 100 nA, with the provision to deliver short current pulses of up to 5 µA for the charge pump. The bit cell design incorporates the HfO_x-based RRAM model that is modified to store a single bit of data. As a result, it exists in either a low resistive state (LRS) of 2 KΩ (representing binary '0') or a high resistive state (HRS) of 20 KΩ (representing binary '1'), yielding an R_{on}/R_{off} ratio of 10.

B. Simulation results

Figure 4 presents the simulation results, with Figure 4a illustrating the input and output characteristics of the C3CIM crossbar column. As seen, V_{MAC} is plotted against the number of 'on' devices, linearly swept from 0 to 64, where 'on' devices correspond to bit-cells with both input '1' and weight '1'. The plot shows a linear relationship between V_{MAC} and the number of 'on' devices, with the entire curve shifted along the y-axis by a constant value, indicating the fixed offset due to the non-zero resistance of the pass transistors. Figure 4b shows the transient behavior of the crossbar column, where V_{MAC} typically requires approximately 500 ns to settle to its final value. However, as shown in Figures 4c and 4d, the implementation of a charge pump significantly reduces this latency to 50 ns, demonstrating its efficacy in enhancing the column's performance. Furthermore, Figure 4e presents a plot of I_{MAC} against the incoming V_{MAC}, linearly swept from 0 to 160 mV (the range of the crossbar column). The linear relationship between I_{MAC} and V_{MAC}, along with V_F being maintained at a nearly constant 900 mV throughout the entire range of V_{MAC} (as shown in Figure 4f), validates the functionality of the proposed TA. Finally, the timing diagram in Figure 4g illustrates the behavior of the LIF circuit, which fires and resets upon reaching the set threshold value of 500 mV.

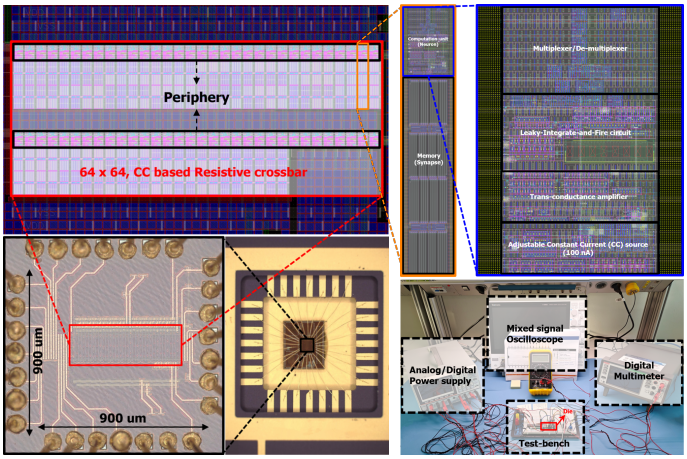
C. Discussion

While this approach offers notable power benefits, it also comes with certain shortcomings that must be weighed against these advantages. Primarily, the massive RC chain formed by numerous bit-cells arranged in series within a column, significantly increases operational latency. In ordinary case, as already seen, the latency can be as high as 500 ns. Secondly, the crossbar's functionality requires the inclusion of an extra transistor in the bit-cell to enable the necessary computations, thereby increasing the crossbar area. Lastly, the design supports only binary inputs as the signals are applied to the gate of the pass transistor, functioning as a switch.

VI. Prototype Characterisation

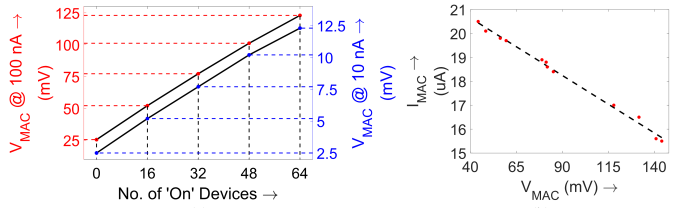
A. Prototype details

Figure 5a depicts the prototype developed to validate our micro-architecture on silicon. Fabricated using TSMC's 40 nm



(a) Microscopic view of the chip (b) Measurement Test-bench

Fig. 5. Prototype for the proposed micro-architecture



(a) (b)

Fig. 6. Measurement results (a) V_{MAC} v/s No. of 'on' devices; (b) I_{MAC} v/s V_{MAC}

CMOS process, the chip has an area of 1mm² and features a 64x64 C3CIM crossbar array at its core. Each of the 64 columns in this crossbar array is equipped with dedicated periphery (Stages 2 and 3) to enable fully parallel operation. To ensure reliability, the RRAMs are emulated using standard resistive devices, hard-coded to predefined resistances/weights derived from off-chip learning. The column current is designed to be externally tunable to compensate for design or fabrication variations and to facilitate precise measurements. An onboard multiplexer allows for simultaneous readout of the outputs from each stage, one column at a time. To maintain design simplicity, no clock is used, thanks to the asynchronous nature of the micro-architecture. Additionally, all the analog reference voltages are supplied externally.

B. Characterisation results

The simulation results are further validated by the characterisation results where, as seen in Figure 6a, the measured output voltage (V_{MAC}) across multiple columns is linearly proportional to the number of 'on' devices in each column. Since the current was adjustable, measurements were conducted not only at 100 nA but also at reduced column currents as low as 10 nA, and the output, as seen, still maintains a linear proportion to the number of 'on' devices. This highlights the effectiveness of the approach and its potential for enabling even lower-power MAC operations. Figure 6b shows the TA in operation, where the measured output current (I_{MAC}) is in a near linear relationship with the V_{MAC}. This behavior also aligns with the simulation results, validating its functionality.

TABLE II
SYSTEM-LEVEL SPECIFICATIONS

Parameters	Specifications	
Datasets	N-MNIST	IBM-Gesture
Test-accuracy	91.8%	84.8 %
Time-steps	300	1450
Quantized	Yes	Yes
Weight precision	6-bit	6-bit

TABLE III
HARDWARE COMPARISON TABLE ON DEPLOYING BOTH THE SNN MODELS TRAINED ON N-MNIST AS WELL AS IBM-GESTURE DATASETS

Parameters	[16] ISSCC-'22	[12] AICAS-'23	**This work**
Technology Supply	40nm 0.9V	40nm 1.1V	40nm 1.1V
Storage device Storage	PCM 1b	Resistive 1b	Resistive 1b
Bitcell R_{High}/R_{Low}	1T1R -	1T1R 200KΩ/2KΩ	**2T1R** 20KΩ/2KΩ
Sensing mode	Voltage	Current	Voltage
Accumulation	8	64	64
Average Power (mW)	396⋆/1970.4◇	531.6⋆/1040.2◇	15.3⋆/72◇
Latency/Inference (μS)	22.9⋆/88.5◇	9.8⋆/37.7◇	93⋆/359.6◇
Energy/Inference (μJ)	9.1⋆/174.3◇	5.2⋆/39.2◇	1.4⋆/25.9◇
TOPS/W	25.4⋆/25.6◇	44.4⋆/113.8◇	**161.7⋆/172.3**◇

⋆: N-MNIST
◇: IBM-Gesture

VII. SYSTEM-LEVEL RESULTS

To benchmark our micro-architecture and demonstrate its efficacy compared to the SOTA solutions, we developed two SNN models trained on the N-MNIST and IBM Gesture datasets. These models, adapted from the LeNet architecture, incorporate spiking convolutional as well as fully connected layers with LIF activation. We utilized the SLAYER SNN framework [26] for model development, with the details summarized in Table II. The spiking convolutional and fully connected layers, where MAC operations occur, are readily mappable to both our micro-architecture and SOTA solutions. For a fair comparison, mainly the crossbar design is adapted from each work while the LIF circuit is assumed to be same. Table III presents a comprehensive performance comparison of our micro-architecture against SOTA solutions, as evaluated on the developed SNN models.

VIII. CONCLUSION

In this paper, we propose a novel approach to CIM crossbar for MAC operations, utilizing a constant current source and a new 2T1R bit-cell design. We also propose a micro-architecture, based on this crossbar array, to facilitate SNN implementation. This is followed by the development of a prototype for on-silicon demonstration and validation of our approach. Benchmarking against SOTA using custom-developed SNN models reveals our work achieving up to 35X higher power efficiency along with upto 6.7x energy savings, demonstrating the potential of our approach in enabling energy-efficient MAC operations with memristor-based CIM crossbars. Additionally, our measurement results suggest further possibilities for reducing current and enabling even lower-power MAC operations.

REFERENCES

[1] S. Bains, "The business of building brains," *Nature Electronics*, 2020.

[2] B. Rajendran, A. Sebastian, M. Schmuker, N. Srinivasa, and E. Eleftheriou, "Low-Power Neuromorphic Hardware for Signal Processing Applications: A Review of Architectural and System-Level Design Approaches," *IEEE Signal Processing Magazine*, 2019.

[3] A. Mehonic and A. J. Kenyon, "Brain-inspired computing needs a master plan," *Nature*, 2022.

[4] *As nodes advance, so must power analysis*, 2014. [Online]. Available: https : / / semiengineering . com / as - nodes - advance - so - must - power - analysis/.

[5] S. Hamdioui, H. A. Du Nguyen, M. Taouil, *et al.*, "Applications of Computation-In-Memory Architectures based on Memristive Devices," in *2019 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2019.

[6] A. Gebregiorgis, H. A. Du Nguyen, J. Yu, *et al.*, "A Survey on Memory-centric Computer Architectures," *J. Emerg. Technol. Comput. Syst.*, 2022.

[7] A. Gebregiorgis, A. Singh, A. Yousefzadeh, *et al.*, "Tutorial on memristor-based computing for smart edge applications," *Memories - Materials, Devices, Circuits and Systems*, 2023.

[8] M. Hu, C. E. Graves, C. Li, *et al.*, "Memristor-Based Analog Computation and Neural Network Classification with a Dot Product Engine," *Advanced Materials*, 2018.

[9] A. Singh, M. A. Lebdeh, A. Gebregiorgis, R. Bishnoi, R. V. Joshi, and S. Hamdioui, "SRIF: Scalable and Reliable Integrate and Fire Circuit ADC for Memristor-Based CIM Architectures," *IEEE Transactions on Circuits and Systems I: Regular Papers*, 2021.

[10] A. Ankit, I. E. Hajj, S. R. Chalamalasetti, *et al.*, "PUMA: A Programmable Ultra-efficient Memristor-based Accelerator for Machine Learning Inference," *ASPLOS*, 2019.

[11] A. Shafiee, A. Nag, N. Muralimanohar, *et al.*, "ISAAC: A Convolutional Neural Network Accelerator with In-Situ Analog Arithmetic in Crossbars," *ISCA*, 2016.

[12] A. Singh, R. Bishnoi, A. Kaichouhi, S. Diware, R. V. Joshi, and S. Hamdioui, "A 115.1 TOPS/W, 12.1 TOPS/mm ^{2} Computation-in-Memory using Ring-Oscillator based ADC for Edge AI," in *2023 IEEE 5th International Conference on Artificial Intelligence Circuits and Systems (AICAS)*, 2023.

[13] A. Chen and M.-R. Lin, "Variability of resistive switching memories and its impact on crossbar array performance," in *2011 International Reliability Physics Symposium*, 2011.

[14] J.-H. Lee, D.-H. Lim, H. Jeong, H. Ma, and L. Shi, "Exploring Cycle-to-Cycle and Device-to-Device Variation Tolerance in MLC Storage-Based Neural Network Training," *IEEE Transactions on Electron Devices*, 2019.

[15] J.-H. Yoon, M. Chang, W.-S. Khwa, Y.-D. Chih, M.-F. Chang, and A. Raychowdhury, "A 40nm 100Kb 118.44TOPS/W Ternary-weight Computein-Memory RRAM Macro with Voltage-sensing Read and Write Verification for reliable multi-bit RRAM operation," in *CICC*, 2021.

[16] W.-S. Khwa, Y.-C. Chiu, C.-J. Jhang, *et al.*, "A 40-nm, 2M-Cell, 8b-Precision, Hybrid SLC-MLC PCM Computing-in-Memory Macro with 20.5 - 65.0TOPS/W for Tiny-Al Edge Devices," in *2022 IEEE International Solid- State Circuits Conference (ISSCC)*, 2022.

[17] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, 1998.

[18] G. Orchard, A. Jayawant, G. K. Cohen, and N. Thakor, "Converting Static Image Datasets to Spiking Neuromorphic Datasets Using Saccades," *Frontiers in Neuroscience*, 2015.

[19] A. Amir, B. Taba, D. Berg, *et al.*, "A Low Power, Fully Event-Based Gesture Recognition System," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[20] M. Pfeiffer and T. Pfeil, "Deep Learning With Spiking Neurons: Opportunities and Challenges," *Frontiers in Neuroscience*, 2018.

[21] Z. Yang, Y. Huang, J. Zhu, and T. T. Ye, "Analog Circuit Implementation of LIF and STDP Models for Spiking Neural Networks," in *Proceedings of the 2020 on Great Lakes Symposium on VLSI*, 2020.

[22] A. Singh, S. Diware, A. Gebregiorgis, *et al.*, "Low-Power Memristor-Based Computing for Edge-AI Applications," in *2021 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2021.

[23] D. Ielmini and H. S. Wong, "In-memory computing with resistive switching devices," *Nature Electronics*, 2018.

[24] B. Hajri, H. Aziza, M. M. Mansour, and A. Chehab, "RRAM Device Models: A Comparative Analysis With Experimental Validation," *IEEE Access*, 2019.

[25] *EMRL*. [Online]. Available: https://emrl.de/JART.html.

[26] S. B. Shrestha and G. Orchard, "SLAYER: Spike Layer Error Reassignment in Time," in *Advances in Neural Information Processing Systems*, 2018.