Fundamental Limits on the Computational Accuracy of Resistive Crossbar-based In-memory Architectures

Saion K. Roy*, Ameya Patil†, and Naresh R. Shanbhag*
*Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, Urbana, IL 61801
†Amazon Lab126, Sunnyvale, CA 94089

***Abstract*—In-memory computing (IMC) architectures exhibit an intrinsic trade-off between computational accuracy and energy efficiency. This paper determines the fundamental limits on the compute SNR of MRAM-, ReRAM-, and FeFET-based crossbars by employing statistical signal and noise models. For a specific dot-product dimension N, the maximum compute SNR (SNR$_{\text{max}}$) is shown to occur at an optimum value of sensing resistance R_s^* where clipping and quantization noise contributions from the analog-to-digital converter (ADC) are balanced out. SNR$_{\text{max}}$ can be further improved by choosing devices with higher resistive contrast $R_{\text{off}}/R_{\text{on}}$, e.g., FeFET, but only until it attains a value in the range 12-15. Beyond this point, mismatch in the input digital-to-analog converters (DACs) and bitcell variations begin to dominate the compute SNR. Finally, by mapping a ResNet-20 (CIFAR-10) network onto resistive crossbars, it is shown that the array-level compute SNR maximizing circuit parameters also maximizes the network-level accuracy.**

***Index Terms*—eNVM, MRAM, ReRAM, FeFET, SNR, in-memory computing, crossbar**

I. Introduction

In-memory computing architectures (IMCs) have emerged as an attractive computational platform for machine learning (ML) workloads due to their ability to overcome the high energy and latency costs associated with data movement inherent in such workloads. Emerging embedded non-volatile memory (eNVM) technologies such as ReRAM, MRAM, and FeFET, are deemed attractive for IMCs [1]–[16] because of their non-volatility and high storage density compared to SRAM.

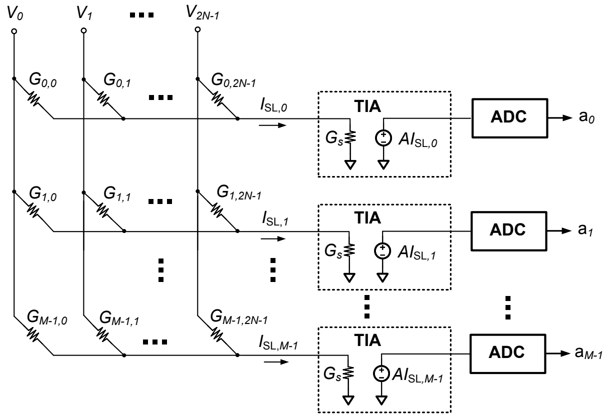
However, these IMCs usually trade-off computational accuracy in order to attain energy efficiency. This trade-off is intrinsic to all IMCs due to their heavy reliance on analog computations – a trade-off that is not well-understood today. Though some work has been done for SRAM-based IMCs [17], [18] it is not clear what the limits on computational accuracy of resistive IMCs are. Unlike digital architectures where accuracy can be conveniently enhanced by assigning more precision to the computation, there are fundamental limits to the computational accuracy of resistive IMCs imposed by various analog non-idealities [6], [11], [15], [16] such as the *resistive contrast* ($R_{\text{off}}/R_{\text{on}}$) of the device type, the input resistance of the read-out circuit (*sensing resistance* R_s), resistive parasitics, readout noise, and the interplay between these noise sources.

Developing a comprehensive understanding of the limits on computational accuracy of resistive IMCs is critical in order to be able to scale-up today's single-bank macro-level designs to multi-bank system architectures. Not surprisingly, multiple approaches to improve the computational accuracy of resistive IMCs have been proposed [19]–[28]. However, these tend to be either empirical design approaches or are simulation-based and therefore are unable to pinpoint the precise limits on accuracy or the key contributors to such limits.

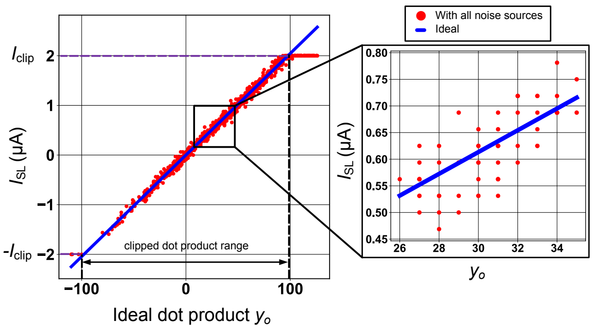
In this paper, we develop an analytical framework for obtaining the fundamental limits on the computational accuracy of ReRAM, MRAM, and FeFET crossbars. We derive expressions for the signal-to-noise ratio (SNR) of array-level computation (*compute* SNR) based on the circuit architecture and noise parameters such as the input digital-to-analog converter (DAC) mismatch, bitcell conductance variations, output clipping noise, and analog-to-digital converter (ADC) quantization noise. We validate these expressions in a commercial $22\,\text{nm}$ node, and employ them to obtain limits on the maximum achievable compute SNR as a function of the sensing resistance R_s, dot-product dimension N, ADC precision, and the resistive contrast $R_{\text{off}}/R_{\text{on}}$ of the device. Finally, we map a ResNet-20 (CIFAR-10) network on to resistive crossbars and demonstrate that the circuit parameter values that maximize the array-level compute SNR also maximize the network-level accuracy. This result enables one to design multi-bank IMC system architectures that can realize deep nets with the maximum achievable network accuracy without relying on tedious simulation-based ad-hoc methodologies.

II. Background

Resistive crossbar architectures employ bit-lines (BLs) perpendicular to source-lines (SLs) with one bitcell (BC) between each BL-SL pair. The conductance of the bitcell is proportional to the weight value stored in the cell. The weight value can be binary (MRAM) or multi-bit (ReRAM, FeFET). The BC has a 1T1R structure, i.e., a transistor followed by a resistive memory device. Input vector $\mathbf{x}$ is provided via voltage DACs on the BLs while the weight vector $\mathbf{w}$ is stored in a row. Figure 1(a) shows the circuit model of a voltage-driven crossbar comprising a transimpedance amplifier (TIA) at the SL, which performs current-to-voltage conversion, followed by an ADC. Word-lines (WLs) connecting the gate terminals of the MOSFETs in the BCs spread across the columns. Multiple (M) WLs are activated simultaneously in order to compute



(a)



(b)

Fig. 1: The voltage-drive resistive crossbar model: (a) circuit schematic, and (b) the impact of various noise sources on the SL current I_{SL} of a ReRAM crossbar with $N = 512$, $R_s = 316\,\Omega$, and a 6 b ADC. The noise in I_{SL} leads to incorrect ADC outputs (red dots in the inset).

M, N-dimensional dot-products or a $M \times N$ matrix-vector multiplication. IMCs strive to have their ADC output approach the ideal N-dimensional dot-product $y_o = \mathbf{w}^{\mathrm{T}}\mathbf{x}$.

The resistive contrast of a memory device is defined as $\frac{R_{\text{off}}}{R_{\text{on}}}$ where R_{on} and R_{off} are the low (on) and the high (off) resistance values of the device respectively. The resistive contrast ranges from 2 for MRAM with $R_{\text{on}} = 3\,\text{k}\Omega$ [29], 12 for ReRAM with $R_{\text{on}} = 25\,\text{k}\Omega$ [14] to 10^3 for FeFET with $R_{\text{on}} = 1\,\text{M}\Omega$ [13]. We make the following assumptions for our SNR analysis:

- the memory device takes binary states R_{on} and R_{off}.
- the sensing resistance R_s is equal to the input impedance of the TIA.

In the case of voltage DAC driven crossbars, the j^{th} BL is driven by $V_{\text{DC}}+V_j$, where V_{DC} denotes the bias voltage for the BL. The input dependent voltage V_j is given by $x_j V_{\text{lsb}}$ with x_j representing the j^{th} element of the input vector $\mathbf{x}$, and V_{lsb} corresponding to the DAC LSB voltage. The TIA holds the SL at V_{DC} thereby ensuring V_j voltage across the j^{th} BC in each row.

The current on the SL will not be zero for zero dot-product as the resistance values are finite. In order to remove the non-zero DC current, the following practices are commonly used [6], [11]:

- Apply complementary DAC voltage on adjacent columns. Input to adjacent columns are the same but with opposite sign i.e. $V_{2k-1} = -V_{2k}\ \forall\ k \in \{1, 2, \ldots, N\}$.
- Employ two bitcells (G_{2k-1}, G_{2k}) to store a single weight bit, where b_{2k} denotes the value stored in bitcell pair (G_{2k-1}, G_{2k}) given by:

$$b_{2k} = \begin{cases} 1, & \text{if } G_{2k-1} > G_{2k} \\ 0, & \text{if } G_{2k-1} = G_{2k} \\ -1, & \text{if } G_{2k-1} < G_{2k} \end{cases}. \tag{1}$$

III. Compute SNR Analysis

A. Signal

Applying KCL and KVL at the BLs and SLs in Figure 1(a), we obtain an expression for the SL current accounting for all current paths as follows:

$$I_{\text{sig}} = \left[\frac{R_{\text{arr}}}{R_{\text{arr}} + R_s}\right]\left(\sum_{k=1}^{N} V_{2k}\Delta G_{2k}\right) = S_I I_{\text{ideal}}, \tag{2}$$

where ΔG_{2k} is expressed as $G_{2k-1} - G_{2k}$ and I_{ideal} denotes the ideal SL current given by $\left(\sum_{k=1}^{N} V_{2k}\Delta G_{2k}\right)$ when $R_s = 0$. The Thevenin resistance looking into the array at the SL (R_{arr}) and the *current scaling factor* (S_I) are as shown below:

$$R_{\text{arr}} = \frac{1}{\sum_{j=1}^{2N} G_j}; S_I = \left[\frac{R_{\text{arr}}}{R_{\text{arr}} + R_s}\right]. \tag{3}$$

The expression in (2) can be interpreted as a fraction S_I of I_{ideal} flowing through the sensing circuit with R_s and R_{arr} appearing in parallel.

On increasing the number of columns (dot product dimension) $2N$ (N), S_I decreases as $R_{\text{arr}} \propto \frac{1}{N}$. From (3), it is observed that reducing R_s increases S_I and hence the current. However, lowering R_s below $500\,\Omega$ is challenging due to significant area overhead.

B. Analog Non-idealities

The SL current in presence of noise can be written as:

$$I_{\text{SL}} = I_{\text{sig}} + I_{\text{nb}} + I_{\text{nd}} + I_{\text{nc}} + I_{\text{nq}}, \tag{4}$$

where I_{sig} is the signal current given by (2). The noise sources (see Fig. 1(b)) are defined as:

- I_{nb} (*bitcell conductance variation*): is the noise current appearing on the SL due to variation in G_{on} and G_{off} values.
- I_{nd} (*input DAC mismatch*): is the input dependent noise current on SL due to the W/L mismatch in the DAC fingers with respect to the reference.
- I_{nc} (*clipping noise*): refers to the noise that arises due to clipping of the I_{SL} beyond the range $(-I_{\text{clip}}, I_{\text{clip}})$.
- I_{nq} (*ADC quantization noise*) $\sim U(-\frac{I_{\text{clip}}}{2^{B_{\text{ADC}}}}, \frac{I_{\text{clip}}}{2^{B_{\text{ADC}}}})$, where $U(L, H)$ denotes the uniform noise between L and H with B_{ADC} being the precision of the ADC.

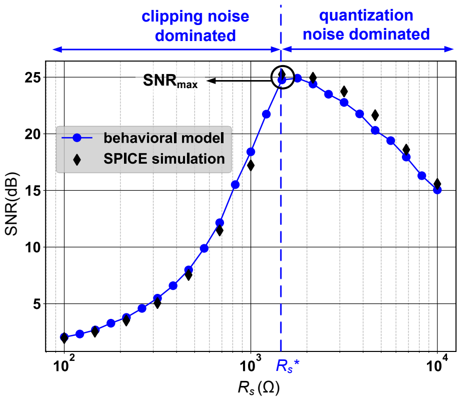


Fig. 2: SNR vs. R_s for ReRAM with $N = 512$ and a 6 b ADC. The SNR achieves a maximum value SNR_{max} at $R_s = R_s^*$ when the clipping and quantization noise are balanced.

We define compute SNR at the SL as:

$$\text{SNR} = \frac{\mathbb{E}[I_{\text{sig}}^2]}{\mathbb{E}[I_{\text{nb}}^2] + \mathbb{E}[I_{\text{nd}}^2] + \mathbb{E}[I_{\text{nc}}^2] + \mathbb{E}[I_{\text{nq}}^2]}, \tag{5}$$

where $\mathbb{E}$ is the expectation taken over the joint distribution of data and noise.

C. Noise Models

The W/L mismatch in the DAC fingers lead to input dependent noise in the DAC output voltage. We combine the mismatch of the complementary $2k^{\text{th}}$ DAC pair into one random variable as:

$$\delta V_{2k} \sim \mathcal{N}\left(0, \sqrt{2|x_{2k}|}\left(\frac{\sigma}{\mu}\right)_{V_{\text{lsb}}} V_{\text{lsb}}\right), \tag{6}$$

where $\left(\frac{\sigma}{\mu}\right)_{V_{\text{lsb}}}$ corresponds to the standard deviation to mean ratio of mismatch in a single DAC finger, and $x_{2k} \in \{-2^{B_x-1}, \ldots, 2^{B_x-1}-1\}$. We, then obtain I_{nd} by substituting $V_{2k} \leftarrow V_{2k} + \delta V_{2k}$ in (2) as:

$$I_{\text{nd}} = \left[\frac{R_{\text{arr}}}{R_{\text{arr}} + R_s}\right]\left(\sum_{k=1}^{N} \delta V_{2k} \Delta G_{2k}\right). \tag{7}$$

Conductance variations arise during the fabrication process causing G_{on} and G_{off} to exhibit spatial variations. Denoting δG_{2k} as the variation in the difference of conductances of the $2k^{\text{th}}$ bitcell pair, we get:

$$\delta G_{2k} \sim \mathcal{N}\left(0, \left(\frac{\sigma}{\mu}\right)_{\text{bc}} \sqrt{G_{\text{on}}^2 + G_{\text{off}}^2}\right), \tag{8}$$

where $\left(\frac{\sigma}{\mu}\right)_{\text{bc}}$ is the standard deviation to mean ratio of bitcell conductance variation. I_{nb} is determined by substituting $\Delta G_{2k} \leftarrow \Delta G_{2k} + \delta G_{2k}$ in (2) as:

$$I_{\text{nb}} = \left[\frac{R_{\text{arr}}}{R_{\text{arr}} + R_s}\right]\left(\sum_{k=1}^{N} V_{2k} \delta G_{2k}\right). \tag{9}$$

Note: from (2) conductance variations will also cause variations in R_{arr} and hence in I_{sig} independent of the input vector $\mathbf{x}$.

IV. Simulation Results

A. Behavioral Models and Their Validation

The signal and noise models from Sections III-A and III-C were verified with SPICE simulations of a 1T1R crossbar array with the MOSFETs from a commercial 22 nm process and resistances describing the bitcell state. The DAC input uses a signed 5 b number with $V_{\text{lsb}} = 3\,\text{mV}$ and $\left(\frac{\sigma}{\mu}\right)_{V_{\text{lsb}}} = 4\%$. The $\left(\frac{\sigma}{\mu}\right)_{\text{bc}}$ for conductance variation was chosen as 4%. The clipping range of $[-2\,\mu\text{A}, 2\,\mu\text{A}]$ is considered for the ADC. We, then model the DAC mismatch (6) as noise proportional to the input which is added to the BL voltage. Conductance variation (8) was modeled as noise in the bitcell resistance. SPICE simulations return the SL current, which is sampled across different inputs. Next, we add clipping and quantization noise to the sampled currents in software. The empirical estimate of compute SNR was obtained by averaging over 10000 (1000) samples in behavioral (SPICE) simulations. Fig. 2 shows that the two estimates match well thereby validating our behavioral models which are employed for obtaining the subsequent results in this paper.

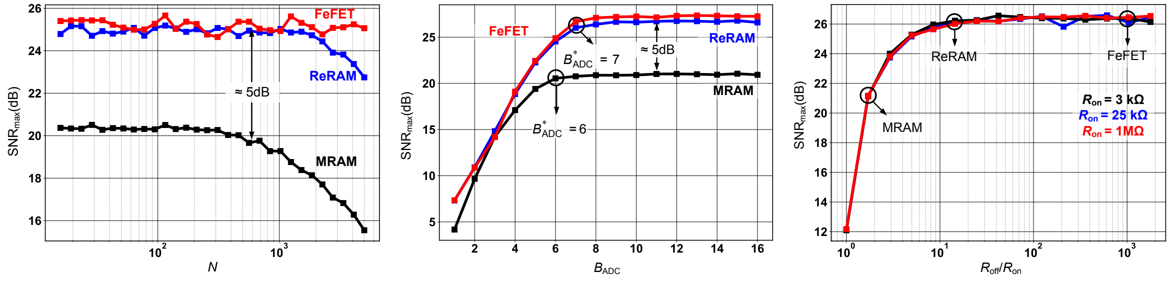
B. SNR Dependence on Circuit, and Device Parameters

Figure 2 shows that the maximum SNR (SNR_{max}) is achieved when the ADC clipping (I_{nc}) and quantization (I_{nq}) noise variances are equal at the SNR-optimum value for the sensing resistance ($R_s = R_s^*$). This trade-off between clipping and quantization noise occurs because the current scaling factor S_I in (3) increases (decreases) as R_s decreases (increases) leading to clipping (quantization) when $R_s < R_s^*$ ($R_s > R_s^*$).

Figure 3(a) shows that the SNR_{max} rolls-off as the dot-product dimension N increases beyond 500 for MRAM and 2000 for ReRAM. This roll-off occurs because the value of R_s^* reduces as N increases and at some point reaches the minimum allowable value of $R_{s,\text{min}}$ (assumed to be $1\,\text{k}\Omega$). For higher values of N, $R_s = R_{s,\text{min}} \neq R_s^*$. In case of FeFET, the R_s^* value is large ($\approx 10\,\text{k}\Omega$) even for large N ($\approx 10^3$) and therefore, no roll-off is observed.

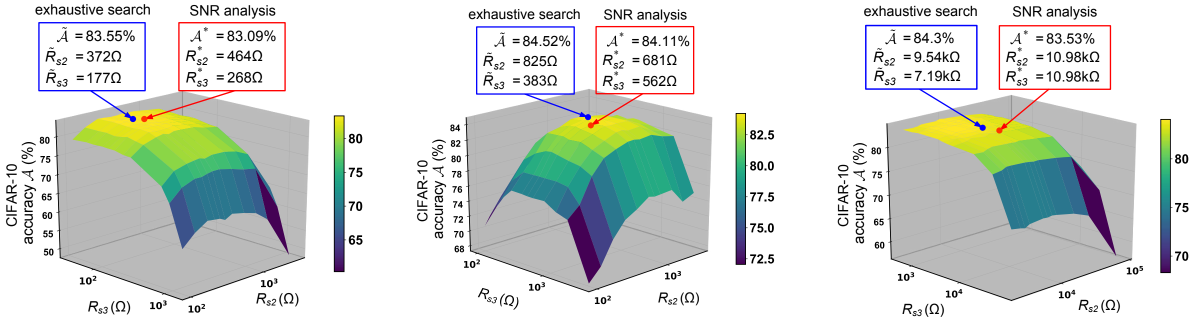
Figure 3(b) shows that the minimum ADC precision (B_{ADC}^*) required for achieving SNR_{max} increases from 6 b for MRAM to 7 b for ReRAM and FeFET. This increase in B_{ADC}^* occurs due to an increase ($\approx 5\,\text{dB}$) in the SNR_{max} for ReRAM and FeFET compared to MRAM as shown in Fig. 3(a). Also, increasing B_{ADC} (lowering quantization noise) does not increase SNR_{max} since DAC mismatch and conductance variations begin to dominate.

Figure 3(c) demonstrates that the SNR_{max} increases with resistive contrast $R_{\text{off}}/R_{\text{on}}$ until a point (12-to-15) and then saturates. The reason being that both R_{arr} and S_I in (3) increase with $R_{\text{off}}/R_{\text{on}}$. This leads to the the signal I_{sig} (2), DAC mismatch I_{nd} (7), and conductance variations I_{nb} (9) all increasing proportionally in (5). When $R_{\text{off}}/R_{\text{on}} <$ 12-15 ($R_{\text{off}}/R_{\text{on}} >$ 12-15) $\mathbb{E}(I_{\text{nc}}^2)$ and $\mathbb{E}(I_{\text{nq}}^2)$ terms in (5) are larger (smaller) compared to $\mathbb{E}(I_{\text{nb}}^2)$ and $\mathbb{E}(I_{\text{nd}}^2)$ resulting in the



(a) (b) (c)

Fig. 3: Maximum SNR trends for ReRAM, MRAM and FeFET: (a) SNR_{max} vs. N for $B_{\text{ADC}} = 6$ and $R_{s,\text{min}} = 1\,\text{k}\Omega$, (b) SNR_{max} vs. B_{ADC} for $N = 512$, and (c) SNR_{max} vs. $R_{\text{off}}/R_{\text{on}}$ for $N = 512$ and $B_{\text{ADC}} = 7$.



(a) (b) (c)

Fig. 4: Surface plot of ResNet-20 (CIFAR-10) accuracy $\mathcal{A}$ vs. (R_{s2}, R_{s3}) with $R_{s1} = R^*_{s1}$: (a) MRAM ($R^*_{s1} = 464\,\Omega$), (b) ReRAM ($R^*_{s1} = 835\,\Omega$), and (c) FeFET ($R^*_{s1} = 8.28\,\text{k}\Omega$). The maximum network accuracy $\tilde{\mathcal{A}}$ *and* the corresponding circuit parameters $(\tilde{R}_{s2}, \tilde{R}_{s3})$ obtained via exhaustive sweep (blue dot) are both close to those obtained by employing compute SNR analysis (red dot), i.e., $\mathcal{A}^*$ and (R^*_{s2}, R^*_{s3}).

increasing (constant) SNR_{max}. Figure 3(c) also indicates that SNR_{max} only depends on the resistive contrast and not on the absolute value of R_{on} as the effect of increasing R_{on} affects both signal and noise equally.

C. System Level Accuracy

This subsection shows that the compute SNR maximizing values R^*_s predicted by our analytical framework at the array-level also maximizes network accuracy. We map a ResNet-20 comprising of three residual blocks each with 6 convolutional layers (3×3 kernels), and having 16, 32, and 64 input channels respectively. With 5 b signed inputs and ternary weights, the network achieves a 84.94% accuracy on CIFAR-10. Following [25], we map the convolutional layers implementing dot products of dimension $N = 144$, 288, 576 to three crossbar arrays consisting of 288, 576, 1152 columns with sensing resistances R_{s1}, R_{s2}, and R_{s3} respectively.

Figure 4 plots the empirical ResNet-20 accuracy over a sweep of (R_{s2}, R_{s3}) while setting $R_{s1} = R^*_{s1}$. The maximum accuracy $\tilde{\mathcal{A}}$ is achieved at $(\tilde{R}_{s2}, \tilde{R}_{s3})$ (indicated by blue dot), which can only be found via exhaustive search. Interestingly, the SNR optimal points (R^*_{s2}, R^*_{s3}) (indicated by red dot) lie very close to $(\tilde{R}_{s2}, \tilde{R}_{s3})$ for all cases. Furthermore, ResNet-20 accuracy with SNR optimal parameters $\mathcal{A}^*$ is within 1% of the empirical maximum $\tilde{\mathcal{A}}$. This shows that maximizing the array-level compute SNR also maximizes the system-level network accuracy.

Considering the case of ReRAM in Fig. 4(b), the maximum CIFAR-10 accuracy of 84.52% is achieved via the exhaustive search at $(\tilde{R}_{s2} = 825\,\Omega,\ \tilde{R}_{s3} = 383\,\Omega)$ and SNR analysis returns an accuracy of 84.11% at $(R^*_{s2} = 681\,\Omega,\ R^*_{s3} = 562\,\Omega)$. Similar observations are made for MRAM (see Fig. 4(a)) and FeFET (see Fig. 4(c)), indicating the generality of our analysis across devices. Notably, the maximum accuracy for our crossbar implementations without retraining is within 2% of the fixed-point baseline value of 84.94%.

V. Conclusion

The analytical framework in this paper enables designers to obtain SNR optimal resistive IMC crossbar parameters without relying on expensive trial and error. Our framework also provides insights such as: 1) increasing device level resistive contrast provides diminishing returns since the input DAC and conductance mismatch begin to dominate; 2) inference accuracy is maximized when quantization and clipping noise in the ADC are balanced. The proposed framework can be extended to other resistive IMC architectures and devices.

Acknowledgment

This work was supported by the JUMP C-BRIC sponsored by SRC and the DARPA FRANC Program.

REFERENCES

[1] C.-X. Xue, W.-H. Chen, J.-S. Liu, J.-F. Li, W.-Y. Lin, W.-E. Lin, J.-H. Wang, W.-C. Wei, T.-W. Chang, T.-C. Chang *et al.*, "A 1Mb multibit ReRAM computing-in-memory macro with 14.6 ns parallel MAC computing time for CNN based AI edge processors," in *2019 IEEE International Solid-State Circuits Conference-(ISSCC)*. IEEE, 2019, pp. 388–390.

[2] C.-X. Xue, T.-Y. Huang, J.-S. Liu, T.-W. Chang, H.-Y. Kao, J.-H. Wang, T.-W. Liu, S.-Y. Wei, S.-P. Huang, W.-C. Wei *et al.*, "A 22nm 2Mb ReRAM compute-in-memory macro with 121-28TOPS/W for multibit MAC computing for tiny AI edge devices," in *2020 IEEE International Solid-State Circuits Conference-(ISSCC)*. IEEE, 2020, pp. 244–246.

[3] C.-X. Xue, J.-M. Hung, H.-Y. Kao, Y.-H. Huang, S.-P. Huang, F.-C. Chang, P. Chen, T.-W. Liu, C.-J. Jhang, C.-I. Su *et al.*, "A 22nm 4Mb 8b-Precision ReRAM Computing-in-Memory Macro with 11.91 to 195.7 TOPS/W for Tiny AI Edge Devices," in *2021 IEEE International Solid-State Circuits Conference (ISSCC)*, vol. 64. IEEE, 2021, pp. 245–247.

[4] 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 *et al.*, "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)*. IEEE, 2018, pp. 494–496.

[5] A. D. Patil, H. Hua, S. Gonugondla, M. Kang, and N. R. Shanbhag, "An MRAM-based deep in-memory architecture for deep neural networks," in *2019 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2019, pp. 1–5.

[6] Q. Liu, B. Gao, P. Yao, D. Wu, J. Chen, Y. Pang, W. Zhang, Y. Liao, C.-X. Xue, W.-H. Chen *et al.*, "A fully integrated analog ReRAM based 78.4 TOPS/W compute-in-memory chip with fully parallel MAC computing," in *2020 IEEE International Solid-State Circuits Conference-(ISSCC)*. IEEE, 2020, pp. 500–502.

[7] C. Li, D. Belkin, Y. Li, P. Yan, M. Hu, N. Ge, H. Jiang, E. Montgomery, P. Lin, Z. Wang *et al.*, "Efficient and self-adaptive in-situ learning in multilayer memristor neural networks," *Nature communications*, vol. 9, no. 1, pp. 1–8, 2018.

[8] F. Cai, J. M. Correll, S. H. Lee, Y. Lim, V. Bothra, Z. Zhang, M. P. Flynn, and W. D. Lu, "A fully integrated reprogrammable memristor–CMOS system for efficient multiply–accumulate operations," *Nature Electronics*, vol. 2, no. 7, pp. 290–299, 2019.

[9] F. M. Bayat, M. Prezioso, B. Chakrabarti, H. Nili, I. Kataeva, and D. Strukov, "Implementation of multilayer perceptron network with highly uniform passive memristive crossbar circuits," *Nature communications*, vol. 9, no. 1, pp. 1–7, 2018.

[10] M. Le Gallo, A. Sebastian, R. Mathis, M. Manica, H. Giefers, T. Tuma, C. Bekas, A. Curioni, and E. Eleftheriou, "Mixed-precision in-memory computing," *Nature Electronics*, vol. 1, no. 4, pp. 246–253, 2018.

[11] W. Wan, R. Kubendran, B. Gao, S. Josbi, P. Raina, H. Wu, G. Cauwenberghs, and H.-S. P. Wong, "A voltage-mode sensing scheme with differential-row weight mapping for energy-efficient RRAM-based in-memory computing," in *2020 IEEE Symposium on VLSI Technology*. IEEE, 2020, pp. 1–2.

[12] I. Boybat, M. Le Gallo, S. Nandakumar, T. Moraitis, T. Parnell, T. Tuma, B. Rajendran, Y. Leblebici, A. Sebastian, and E. Eleftheriou, "Neuromorphic computing with multi-memristive synapses," *Nature communications*, vol. 9, no. 1, pp. 1–12, 2018.

[13] T. Soliman, F. Müller, T. Kirchner, T. Hoffmann, H. Ganem, E. Karimov, T. Ali, M. Lederer, C. Sudarshan, T. Kämpfe *et al.*, "Ultra-low power flexible precision FeFET based analog in-memory computing," in *2020 IEEE International Electron Devices Meeting (IEDM)*. IEEE, 2020, pp. 29–2.

[14] C.-X. Xue, Y.-C. Chiu, T.-W. Liu, T.-Y. Huang, J.-S. Liu, T.-W. Chang, H.-Y. Kao, J.-H. Wang, S.-Y. Wei, C.-Y. Lee *et al.*, "A CMOS-integrated compute-in-memory macro based on resistive random-access memory for AI edge devices," *Nature Electronics*, vol. 4, no. 1, pp. 81–90, 2021.

[15] P. Deaville, B. Zhang, L.-Y. Chen, and N. Verma, "A Maximally Row-Parallel MRAM In-Memory-Computing Macro Addressing Readout Circuit Sensitivity and Area," in *ESSCIRC 2021-IEEE 47th European Solid State Circuits Conference (ESSCIRC)*. IEEE, 2021, pp. 75–78.

[16] R. Mochida, K. Kouno, Y. Hayata, M. Nakayama, T. Ono, H. Suwa, R. Yasuhara, K. Katayama, T. Mikawa, and Y. Gohou, "A 4M synapses integrated analog ReRAM based 66.5 TOPS/W neural-network processor with cell current controlled writing and flexible network architecture," in *2018 IEEE Symposium on VLSI Technology*. IEEE, 2018, pp. 175–176.

[17] M. Kang, Y. Kim, A. D. Patil, and N. R. Shanbhag, "Deep in-memory architectures for machine learning–accuracy versus efficiency trade-offs," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 67, no. 5, pp. 1627–1639, 2020.

[18] S. K. Gonugondla, C. Sakr, H. Dbouk, and N. R. Shanbhag, "Fundamental limits on the precision of in-memory architectures," in *Proceedings of the 39th International Conference on Computer-Aided Design*, 2020, pp. 1–9.

[19] P. Chi, S. Li, C. Xu, T. Zhang, J. Zhao, Y. Liu, Y. Wang, and Y. Xie, "PRIME: A novel processing-in-memory architecture for neural network computation in ReRAM-based main memory," *ACM SIGARCH Computer Architecture News*, vol. 44, no. 3, pp. 27–39, 2016.

[20] W. Wu, H. Wu, B. Gao, P. Yao, X. Zhang, X. Peng, S. Yu, and H. Qian, "A methodology to improve linearity of analog RRAM for neuromorphic computing," in *2018 IEEE Symposium on VLSI Technology*. IEEE, 2018, pp. 103–104.

[21] M. Cheng, L. Xia, Z. Zhu, Y. Cai, Y. Xie, Y. Wang, and H. Yang, "TIME: A training-in-memory architecture for RRAM-based deep neural networks," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 38, no. 5, pp. 834–847, 2018.

[22] S. Jain and A. Raghunathan, "Cxdnn: Hardware-software compensation methods for deep neural networks on resistive crossbar systems," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 18, no. 6, pp. 1–23, 2019.

[23] Z. He, J. Lin, R. Ewetz, J.-S. Yuan, and D. Fan, "Noise injection adaption: End-to-end ReRAM crossbar non-ideal effect adaption for neural network mapping," in *Proceedings of the 56th Annual Design Automation Conference 2019*, 2019, pp. 1–6.

[24] Y. Long, X. She, and S. Mukhopadhyay, "Design of reliable DNN accelerator with un-reliable ReRAM," in *2019 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2019, pp. 1769–1774.

[25] V. Joshi, M. Le Gallo, S. Haefeli, I. Boybat, S. R. Nandakumar, C. Piveteau, M. Dazzi, B. Rajendran, A. Sebastian, and E. Eleftheriou, "Accurate deep neural network inference using computational phase-change memory," *Nature communications*, vol. 11, no. 1, pp. 1–13, 2020.

[26] 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, 2020.

[27] I. Chakraborty, S. Roy, S. Sridharan, M. Ali, A. Ankit, S. Jain, and A. Raghunathan, "Design Tools for Resistive Crossbar based Machine Learning Accelerators," in *2021 IEEE 3rd International Conference on Artificial Intelligence Circuits and Systems (AICAS)*. IEEE, 2021, pp. 1–4.

[28] S. Roy, S. Sridharan, S. Jain, and A. Raghunathan, "TxSim: Modeling training of deep neural networks on resistive crossbar systems," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 29, no. 4, pp. 730–738, 2021.

[29] T. Na, S. H. Kang, and S.-O. Jung, "STT-MRAM sensing: a review," *IEEE Transactions on Circuits and Systems II: Express Briefs*, 2020.