A Maximally Row-Parallel MRAM In-Memory-Computing Macro Addressing Readout Circuit Sensitivity and Area

Peter Deaville, Bonan Zhang, Lung-Yen Chen and Naveen Verma
Princeton University, Princeton, NJ, USA (deaville@princeton.edu)

ESSCIRC 2021 - IEEE 47th European Solid State Circuits Conference (ESSCIRC) | 978-1-6654-3751-6/21/$31.00 ©2021 IEEE | DOI: 10.1109/ESSCIRC53450.2021.9567807

***Abstract*—This paper presents the first MRAM-based In-Memory-Computing (IMC) macro, implemented as a 128-kb array in an advanced-node 22nm FD-SOI technology. The design maximizes IMC row parallelism for energy efficiency and throughput, while addressing the critical challenges this raises, namely: high column currents; high output dynamic-range requirements; and large area of peripheral readout circuits. These are addressed through current-insensitive column-multiplexing and high-sensitivity readout circuits, occupying 26% of the macro area. Residual IMC non-idealities, arising from statistical circuit variations, are modeled and incorporated in a chip-generalized one-time neural-network training algorithm, with CIFAR-10 image-classification accuracy demonstrated at 90.1%, equal to ideal digital computation. The design addresses the particularly high sensitivity required for MRAM-based IMC compared to other non-volatile memory technologies, while achieving area-normalized throughput of 758 GOPS/mm^2 and energy efficiency of 5.1 TOPS/W for the macro.**

I. Introduction

Embedded non-volatile memory (eNVM) offers key advantages in ultra-low-power data-intensive applications, such as on-device AI, due to its high density and potential for low-leakage in event-based duty-cycled operation. In-memory computing (IMC) based on eNVM enables significant further energy and throughput benefits in AI applications. Such applications are dominated by high-dimensional matrix-vector multiplies (MVMs), for which IMC activates memory rows in parallel to access a compute result over many stored data, rather than accessing individual bits of data one at a time. This gives energy/throughput gains, but increases output dynamic range, causing a tradeoff with signal-to-noise ratio (SNR).

While recent IMCs works have targeted ReRAM-based eNVM for row-parallel operation [1-4], IMC IP is required in the range of technologies suited for different application needs. In particular, MRAM has emerged as an important foundry offering to address harsh-environment robustness (e.g., to temperature, radiation). However, it poses key challenges to the IMC energy/throughput-vs.-SNR tradeoff, due to lower absolute resistance and lower resistance-state contrast of the bit cells. Both of these directly oppose increasing row parallelism and, thus, IMC throughput and energy-efficiency gains. Thus far, MRAM IMCs have been restricted to bit-level logical operations without addressing these challenges for significant row parallelism [5]. This work overcomes these challenges to demonstrate MRAM-based IMC in a 256-row macro. The major contributions of this work are as follows:

- We demonstrate the first MRAM-based IMC macro, implemented in an advanced-node 22nm FD-SOI technology from GF and achieving high row parallelism, for maximizing IMC gains. We present basic operation and neural-network (NN) CIFAR-10 classification, leveraging a generalized training algorithm.
- We develop a column-multiplexing approach, necessary for enabling readout-circuit pitch matching in high cell-density IMC, where the area overhead of the multiplexing circuitry is minimized through a nonlinearity-insensitive switching scheme.
- We develop high-sensitivity pitch-matched readout circuits suitable for eNVM-based IMC, where low absolute resistance and resistance-state contrast leads to small sensing voltages.

II. Architecture Overview and Challenges

Fig. 1 shows the MRAM IMC macro demonstrated, comprising: a 256(row)×512(col.) 1T-1R array of MTJ-based MRAM bit cells; write/read periphery; and 4:1 multiplexed readout circuitry, providing 4-b IMC outputs. The macro computes a vector inner product in each column, between binary elements stored in the MRAM cells and binary elements provided on the WLs. The result is represented by the conductance between bit-line (BL) and source-line (SL). Fig. 2 shows how multiplication of +/-1 binary-element data is performed, requiring two complementary cells in each column, storing complementary antiparallel(AP)/parallel(P) MTJ states, and driven by complementary WL data (two complementary are needed since multiplication is commutative, yet the stored

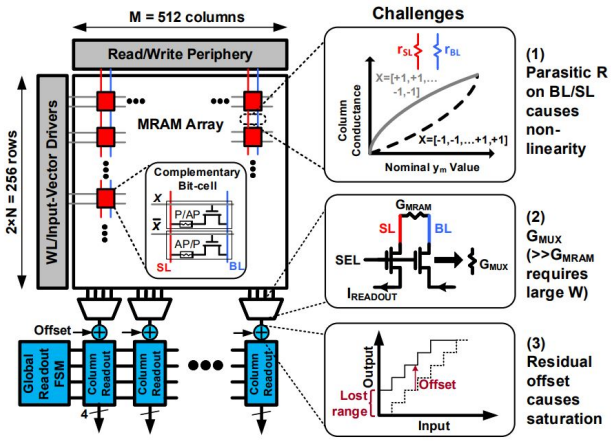

Fig. 1 MRAM IMC macro block diagram and overview of challenges.

MTJ conductance and WL-controlled MOSFET conductance are highly asymmetric).

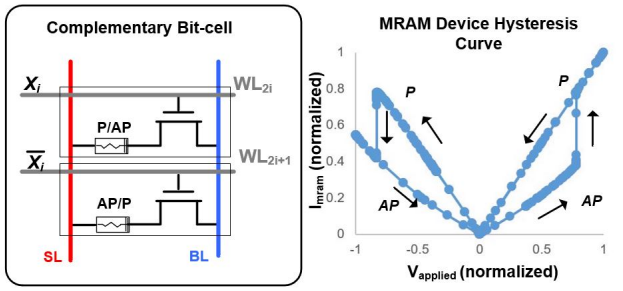

Fig. 2 Multiplication via bit cells storing complementary AP/P state data (left), corresponding to resistance states shown in hysteresis curve (right).

This yields a one-shot 128×128 binary-element MVM. Having 4-b readout circuitry in the architecture is essential for extension to both larger MVMs and multi-bit elements. For larger MVMs, partial inner-products from parallel macros can thus be added. For multi-bit elements bit-parallel/bit-serial processing can be employed [1], where multiple matrix-element bits are stored in parallel columns, and multiple input-vector element bits are provided serially. Each column computation thus remains a binary inner product, and the multi-bit computation is achieved by properly bit-shifting (binary weighting) and adding the digitized column outputs.

Fig. 1 also identifies the major challenges addressed in this design, resulting from large BL-SL conductance when increasing IMC row-parallelism. These include: (1) BL/SL wire resistances introduce spatial data dependency in the column computation; (2) the high sensing currents required for conductance-to-voltage conversion cause nonlinear parasitic series-resistance effects from column-select switches, particularly when keeping switch widths small to reduce area; (3) the small resulting input voltage ranges make readout circuits highly sensitive to offsets, causing significant loss of readout range. These are addressed through circuit and algorithmic techniques, as described in the following sections.

III. Nonlinearity-Insensitive Column Mux'ing

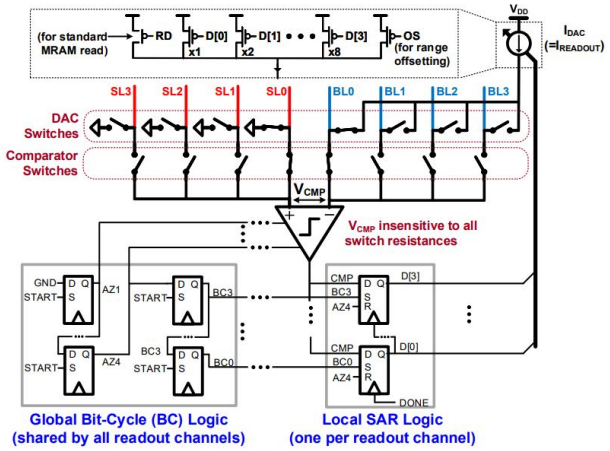

Fig. 3. Readout circuit with nonlinearity-insensitive column-mux'ing.

Fig. 3 shows the column readout circuit. The macro consists of 512 columns 4:1-multiplexed into 128 4-b current-feedback SAR ADCs, whose the bit-cycling (BC) phases are controlled by one global BC state-machine block. Each ADC consists of a high-sensitivity comparator, bit-decision FFs, and a PMOS binary-weighted feedback current(I)-DAC. The I-DAC outputs currents from 600-696μA, to yield adequate voltage across the BL-SL conductance for comparator sensing.

While column multiplexing is necessary for pitch-matching the bit-cell array and readout circuits, a key challenge has been the large area occupied by the mux's themselves. This arises due to the need for wide MOSFET switches, in order to suppress V_{DS} nonlinearity from the large currents used to generate adequate BL-SL sensing voltage across the small-valued and parallelized bit-cell resistances. The proposed circuit is made insensitive to switch nonlinearity by using separate DAC switches (which carry current) and comparator switches (which carry no current), so the comparator input V_{CMP} corresponds to only the voltage across the BL-SL conductance. This enables significant reduction of switch sizes, and thus total readout area.

IV. High-Sensitivitiy Readout Circuits

The readout circuitry is used for both standard memory read operations and IMC column-computation readout. IMC readout sets the sensitivity requirements, due to the high column-computation dynamic range and low parallelized BL-SL resistance (leading to small voltage-sensing range).

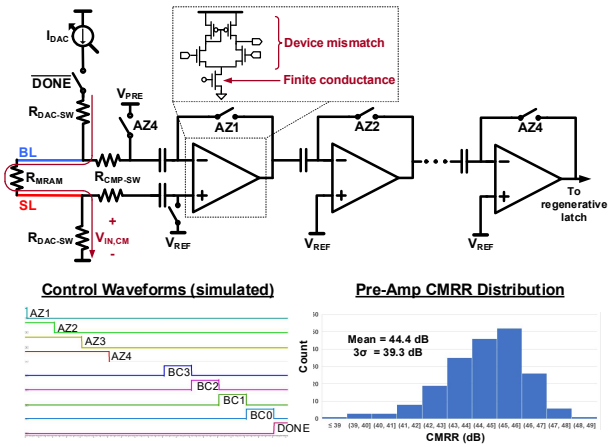

Fig. 4: Details of the high-sensitivity comparator's preamp stages, showing autozeroing and analyzing CMRR to enhance insensitivity to DAC column-select switch resistances (R_{DAC-SW}).

Fig. 4 shows details of the readout circuit, focusing on the high-sensitivity comparator. To minimize the required I-DAC current, which dominates readout-circuit energy, the small resulting V_{CMP} (from Fig. 3) is amplified by four pre-amp stages, with the first stage designed for low thermal noise. The fourth stage feeds a clocked regenerative latch, whose output settling is detected and used to generate a DONE signal, to quickly shut off the I-DAC current path. The pre-amp stages employ input-offset autozeroing via sampling on capacitors through feedback. This ensures pre-amp operation remains in the high-gain region despite offsets, enabling large gain per stage for enhancing sensitivity. Autozeroing is sequenced

from the first preamp to the last, using the AZ1-4 signals (all activated before each SAR-ADC conversion begins), to successively cancel offset error arising due to charge injection from the autozeroing switches.

The primary remaining source of comparator error is input common-mode voltage $V_{IN,CM}$ on V_{CMP}, arising due to current through the DAC-switch resistances ($R_{DAC\text{-}SW}$). This is mitigated by using the differential preamp topology shown, whose 3-σ CMRR, set by device mismatch and finite tail-MOSFET conductance, is nearly 40dB from Monte Carlo simulations.

Standard read operations are performed using the same readout circuitry, but by activating the WL of a single bit-cell and using a separate, fixed bias current I_{READ} instead of the I-DAC feedback current (note, I_{READ} can be significantly lower to generate adequate sensing-voltage with a single bit-cell resistance). The remaining pre-amps and comparator are reused, but with the global BC state machine configured for a single BC phase, to give a 1-b output.

V. Hardware-Generalized Neural Network Training

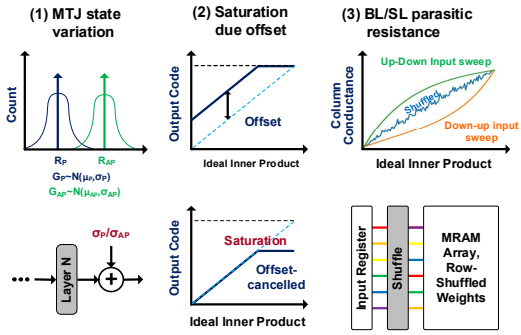

Fig. 5. Stochastic modeling of residual nonidealities.

In addition to circuit approaches, algorithmic approaches are leveraged to address the SNR tradeoff incurred with high row-parallelism MRAM IMC. A stochastic neural-network (NN) training algorithm is employed, which represents the statistical distributions of key noise sources within the macro, and samples from those distributions during loss-function evaluation in the backpropagation algorithm. Representing statistical distributions in this way enables model parameters to be learned that generalize across hardware instantiations of the macro, without further parameter tuning [6].

As illustrated in Fig. 5, three noise sources are considered. First, bit-cell AP/P-state conductance variations are modeled as additive Gaussian noise following NN IMC operations. Second, residual offset in the comparator is modeled as Gaussian-distributed offset in the column ADC, yielding premature ADC-code saturation after digital-domain offset cancellation. Third, the residual spatial data dependence induced by BL/SL resistance (even after aggressive BL/SL strapping within the MRAM array) is further mitigated by shuffling the mapping of input-vector and matrix elements to rows of the macro. This randomizes any spatial patterns in application MVM data.

VI. Prototype Measurements and Demonstrations

The 128-kb MRAM IMC macro is implemented in a GF 22nm FD-SOI technology. Fig. 6 shows the die photo and a layout annotations of the macro. As shown, the high density of the MRAM array itself means that the area efficiency of the readout circuitry is crucial for achieving overall area efficiency. The readout circuity occupies 26% of the total macro area, despite the high-sensitivity circuits employed to support high column parallelism. As shown, the area of the column-multiplexing switches has been significantly reduced compared to previous designs, thanks the nonlinearity-insensitive switching (Fig. 4), which substantially eases the switch-resistance requirements. The remainder of the readout circuitry is dominated by the area of the four pre-amp stages (each with auto-zeroing capacitors implemented as metal-layer fringing structures) required for high sensitivity.

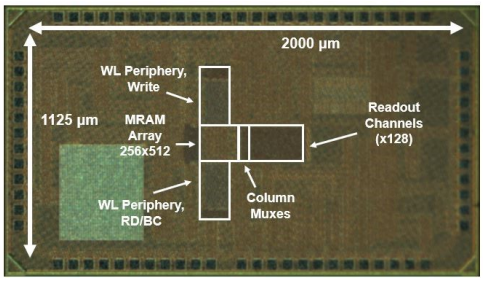

Fig. 6. Prototype die photo.

Fig. 7 shows measurements of the basic operation and NN computations. Column transfer functions (top) are obtained by loading all 1's in the MRAM array and sweeping the number of input vector +1's vs. -1's. The left and middle column-transfer plots correspond to sweeps from top-to-bottom and bottom-to-top of the rows, respectively. The significant bowing is due the residual resistance after BL/SL strapping, which causes a spatial data-dependent nonlinearity. The upper-right plot shows that this nonlinearity is corrected by the algorithmic approach of shuffling the rows, with transfer-function data shown for different macro columns.

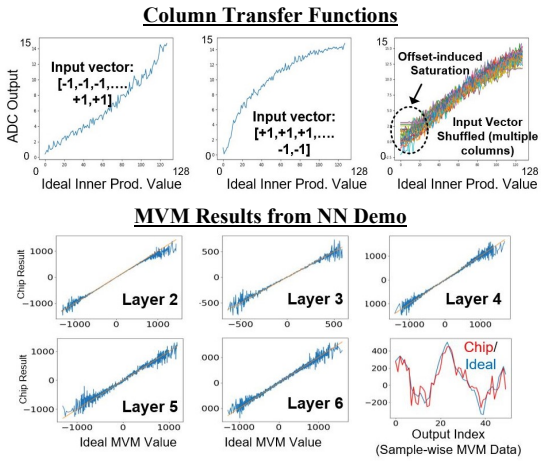

Fig. 7. Transfer functions for single IMC operations (top), showing the need for row-shuffling to overcome nonlinearity, and transfer functions for large MVMs (bottom) mapped to multiple IMC operations, exhibiting good linearity with row-shuffling algorithm.

Also shown (bottom) are averaged transfer functions from large MVM operations mapped to the chip, for a six-convolutional-layer CNN demonstrated for CIFAR-10 image classification (variations at ends are due to less averaging where there is less data from the CNN layer computations). Finally, the lower-right plot shows raw data samples from one of the layer computations, derived from both chip measurement and ideal software simulation for comparison. As seen the row-shuffling algorithm leads to high linearity and good output agreement (noise at the ends is due to variation where there is less CNN data).

The prototype was tested using an FPGA to implement a controller for read, write and IMC operations, as well as Ethernet-based interface to a host PC. Further, Python-based software interfaces were designed to implement MVM and NN-layer kernels, enabling integration with PyTorch for NN performance evaluation and benchmarking against software executions. For IMC execution, software MVM API calls were replaced with the custom functions, which initiated communication and control to the prototype chip through the FPGA, allowing computations to be directly and flexibly delegated to the prototype.

	Liu et al., ISSCC '20	Mochida et al., VLSI '18	Yin et al., T-EDM '20	This Work
Technology	ReRAM, 130nm CMOS	ReRAM, 180nm/40nm CMOS	ReRAM, 90nm CMOS	MRAM, 22nm CMOS
Comp. Density (1b-GOPS/mm^2)	300	243	3424	758
Row Parallelism	784	196	128	128
Column Parallelism	100	64	8	128
ADC Bits	1b	1b	3b	4b
Efficiency (1-b TOPS/W)	78.4	20.7 / 66.5	116	5.1
NN Demo	MNIST, 94.4%	MNIST, 90.4%	CIFAR-10, 83.5%	CIFAR-10 90.1% (one layer in MRAM)

Fig. 8: Comparison table. This work is the only MRAM IMC demonstration comparable to previous ReRAM IMC designs.

Fig. 8 shows the comparison summary, against previous eNVM-based IMC designs performing parallelized multiply accumulates. As shown, while all previous works target ReRAM, this is the only MRAM-based IMC. This work achieves high compute density (normalized to 1-b element operations) of 758 1b-GOPS/mm^2, thanks to high row/column parallelism. While the energy-efficiency of 5.1 1b-TOPS/W is lower than previous designs, this is due to significantly lower resistance and resistance-state contrast of the MRAM technology, which directly affects IMC energy in through the SNR tradeoff. Finally, a 6-layer CNN was mapped to the design for CIFAR-10 image classification (testing 1 layer at a time), using the training algorithm of Sec. V, without any further tuning of the model parameters to the specific test-chip employed for the demo. A classification accuracy of 90.1% is achieved, matching that expected from ideal software-based computation, and one of the few to scale up to a CIFAR-10 classification task.

VII. Conclusions

This work presents the first MRAM-based IMC design enabling high row-parallelism. The foundry MRAM technology, integrated in an advanced-node 22nm FD-SOI process, introduces the challenges of low bit-cell resistance and resistance-state contrast, both of which directly impact critical IMC considerations concerning parallelism and compute SNR. Circuit and algorithmic approaches are developed to address the challenges, including: (a) nonlinearity-insensitive column-multiplexing to reduce the area of column-select switches; (b) high-sensitivity readout and analog-to-digital conversion; and (c) row-wise data shuffling to mitigate data-dependent spatial nonlinearity from BL/SL parasitic resistance. Accompanying the macro, a hardware-generalized stochastic NN training algorithm is developed and used to demonstrate a 6-layer CNN for CIFAR-10 classification, achieving accuracy of ideal software implementation without the need for any chip-specific parameter tuning.

Acknowledgments

Research sponsored by Air Force Research Laboratory (AFRL) and Defense Advanced Research Projects Agency (DARPA), agreement no. FA8650-18-2-7866.

References

[1] Q. Liu et al., "33.2 A Fully Integrated Analog ReRAM Based 78.4TOPS/W Compute-In-Memory Chip with Fully Parallel MAC Computing," 2020 IEEE International Solid- State Circuits Conference - (ISSCC), 2020, pp. 500-502

[2] C. Xue et al., "15.4 A 22nm 2Mb ReRAM Compute-in-Memory Macro with 121-28TOPS/W for Multibit MAC Computing for Tiny AI Edge Devices," 2020 IEEE International Solid- State Circuits Conference - (ISSCC), 2020, pp. 244-246K

[3] R. Mochida et al., "A 4M Synapses integrated Analog ReRAM based 66.5 TOPS/W Neural-Network Processor with Cell Current Controlled Writing and Flexible Network Architecture," 2018 IEEE Symposium on VLSI Technology, 2018, pp. 175-176R. Mochida et al., VLSI, pp. 175–176, 2018.

[4] S. Yin, X. Sun, S. Yu and J. -S. Seo, "High-Throughput In-Memory Computing for Binary Deep Neural Networks With Monolithically Integrated RRAM and 90-nm CMOS," in IEEE Transactions on Electron Devices, vol. 67, no. 10, pp. 4185-4192, Oct. 2020

[5] S. Angizi, Z. He, A. Awad and D. Fan, "MRIMA: An MRAM-Based In-Memory Accelerator," in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 39, no. 5, pp. 1123-1136, May 2020

[6] B. Zhang, L. Chen and N. Verma, "Stochastic Data-driven Hardware Resilience to Efficiently Train Inference Models for Stochastic Hardware Implementations," ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2019, pp. 1388-139.