CIM-based Robust Logic Accelerator using 28 *n*m STT-MRAM Characterization Chip Tape-out

Abhairaj Singh[1], Mahdi Zahedi[1], Taha Shahroodi[1], Mohit Gupta[2], Anteneh Gebregiorgis[1], Manu Komalan[2], Rajiv V. Joshi[3], Francky Catthoor[2], Rajendra Bishnoi[1], Said Hamdioui[1]
[1]Computer Engineering Laboratory, TU Delft, The Netherlands: S.Hamdioui@tudelft.nl
[2]IMEC Leuven, Belgium [3]IBM Thomas J. Watson Research Centre, Yorktown, USA

2022 IEEE 4th International Conference on Artificial Intelligence Circuits and Systems (AICAS) | 978-1-6654-0996-4/22/$31.00 ©2022 IEEE | DOI: 10.1109/AICAS54282.2022.9869993

***Abstract*—Spin-transfer torque magnetic random access memory (STT-MRAM) based *computation-in-memory* (CIM) architectures have shown great prospects for an energy-efficient computing. However, device variations and non-idealities narrow down the sensing margin that severely impacts the computing accuracy. In this work, we propose an adaptive referencing mechanism to improve the sensing margin of a CIM architecture for boolean binary logic (BBL) operations. We generate reference signals using multiple STT-MRAM devices and place them strategically into the array such that these signals can address the variations and trace the wire parasitics effectively. We have demonstrated this behavior using an STT-MRAM model, which is calibrated using 1Mbit characterized array. Results show that our proposed architecture for binary neural networks (BNN) achieves up to 17.8 TOPS/W on the MNIST dataset and 130× performance improvement for the text encryption compared to the software implementation on Intel Haswell processor.**

***Index Terms*—STT-MRAM, computation-in-memory, binary logic, binary neural networks**

I. Introduction

Computation-in-memory (CIM) architectures are well-known for their massive parallelism and high energy efficiency, and have been vastly explored to perform boolean binary logic (BBL) operations for applications such as database query, binary neural networks (BNN) and encryption [1]. Emerging STT-MRAM technology stores values in terms of resistance states, and the feature of their state dynamics makes it naturally suitable for the CIM architecture. These devices are non-volatile, compact, scalable and compatible with CMOS technologies [2]. In addition, STT devices exhibit high endurance and allow fast and low-power read/write operations. As illustrated in Fig. 1a, in a CIM architecture, the storing devices are arranged in a crossbar structure, where BBL operations are accelerated by leveraging circuit laws, such as Ohm's law and Kirchhoff's current law. However, current STT-MRAM based CIM suffers from challenges related to small sense margins, which are due to device variations, low tunnel magneto resistance (TMR) value [3] and wire parasitics that can severely impact the accuracy of the system.

Several prior reference schemes have focused on building STT-MRAM based reference circuits and dedicated sense amplifiers, as opposed to using a constant reference signal [4], to adapt to global device variations [5–8]. However, most of these solutions neither consider STT-MRAM and CMOS process, voltage and temperature (PVT) variations, nor the RC delay mismatch to validate the computing accuracy; hence, they could be highly optimistic about design closures. Existing efforts that do consider these effects mainly focus only on read operations and do not implement CIM-based logic operations [3, 9, 10]. Moreover, prior work on STT-MRAM based CIM lacks silicon-driven investigations to accurately determine the impact of these non-idealities on the computing accuracy. In summary, to demonstrate the potential and robustness of a real STT-MRAM based CIM architecture, there is a need for comprehensive investigation using silicon parameters.

In this paper, we develop and validate a design methodology that improves the sensing margins for robust CIM-based BBL operations. The contributions of this paper are:

- A novel referencing scheme to improve read margins where STT-MRAM devices are used in the reference cells that exhibit high tolerance against PVT variations.
- An approach to address RC delay mismatch where these reference cells are split into two sub-cells and placed strategically within the bitcell array that additionally, improves the performance of the logic operations.
- Integration and validation of our scheme on BNN and text-encryption, and analysis on the computing efficiency.

Our simulation results (based on parameters calibrated from an experimentally verified STT-MRAM 1Mbit characterized chip) show that we can achieve up to 17.8 TOPS/W on the MNIST dataset and 130× performance improvement compared to Intel Haswell processor based CPU implementation.

II. Design Methodology

Next, we present our proposed design methodology and implementation of STT-MRAM based CIM for BBL operations.

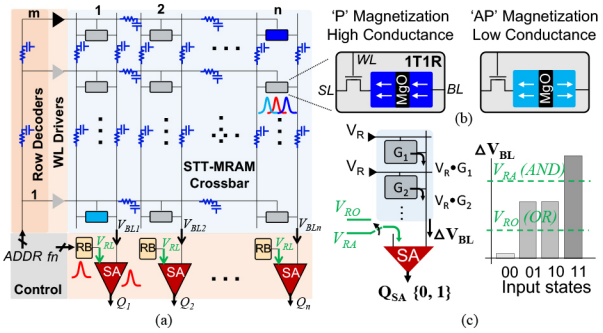


Fig. 1: (a) CIM core and its non-idealities (b) STT-MRAM based 1T1R bitcells (c) Two-row read for CIM-based logic operations.

This work was supported by EU H2020 grant "DAIS" that received funding from ECSEL Joint Undertaking (JU) under grant agreement No 101007273.

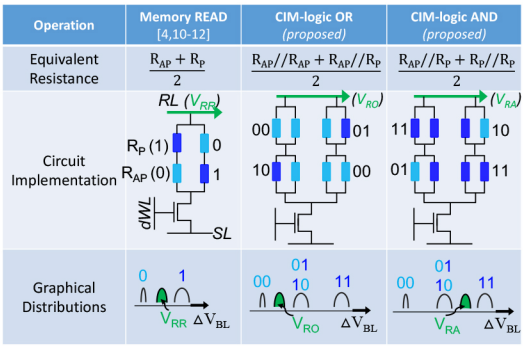


Fig. 2: Reference for read, CIM-logic OR and CIM-logic AND operations.

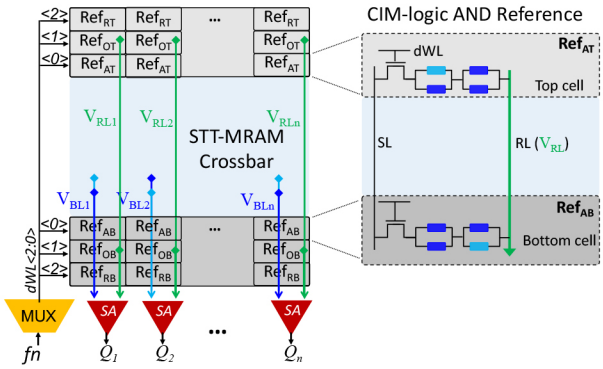


Fig. 3: Proposed reference cells arrangement for read and CIM-based logic operations, and symmetrical reference sub-cell configurations.

A. Motivation and Approach

Fig. 1a summarizes the non-idealities that adversely affect the sensing margins associated with the read and logic operations; i) Wire parasitics: RC delay mismatch along the critical path. ii) Systematic and random variations: in sense amplifiers (SA) and reference blocks (RB) generating reference signals. iii) PVT variations related to CMOS and STT-MRAM. The purpose is to develop a methodology that primarily focuses on generating reliable reference signals to provide and maintain high sensing margins in the presence of the aforementioned non-idealities. In our approach, we constrain the reference signal (reference line, with voltage signal V_{RL}) to experience similar non-idealities from which the input signal (bitline, with voltage V_{BL}) suffers, during a multi-row select read-assisted logic operation. To this end, we (a) build the reference circuits using a combination of STT-MRAM and CMOS devices to address PVT variations, and (b) integrate strategically the reference units within the bitcell array to tackle the increasing wire parasitics along the rows and columns.

B. Reference Generators

The required reference signal must differentiate the two critical resistance states (resulting in minimum margins) out of all the possible input states that determine the logic output. Moreover, the reference signal that falls in the *middle of these two critical states* indisputably offers maximum signal sensing margins. As shown in Fig. 2, this methodology has inspired many prior works to explore the generation of such reference signals for read operations using STT-MRAM devices [11].

To this end, we propose a scheme to arrange STT-MRAM devices in such a way that it produces an average equivalent resistance of the two critical resistance states for the desired operation. For instance, the OR reference falls in the middle of the two critical states 00 (i.e., both in 'AP' state with an equivalent resistance $\langle R_{AP}//R_{AP}\rangle$) and 01/10 (i.e., one in 'AP' and one in 'P' state with an equivalent resistance $\langle R_{AP}//R_P\rangle$). Hence the middle resistance is $\langle\frac{R_{AP}//R_{AP}+R_{AP}//R_P}{2}\rangle$. In a similar fashion, AND reference signal has an effective resistance of $\langle\frac{R_{AP}//R_P+R_P//R_P}{2}\rangle$. Fig. 2 shows the circuit implementations of the proposed reference generators and graphical distribution of input and reference signals V_{RO} and V_{RA} generated for the OR and AND operations, respectively.

C. Reference Arrangement

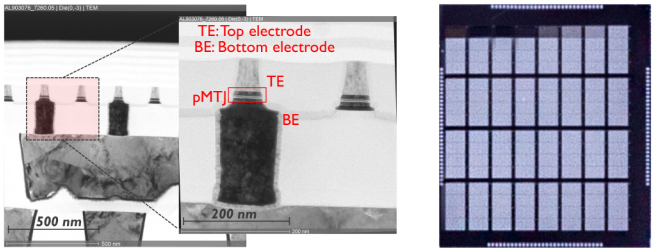
In addition to PVT consideration, it is also important to account the wire delay mismatch related to the position of the accessed bitcell in the crossbar array. For instance, as shown in Fig. 1a, the input signal V_{BL} while accessing the bitcell at position [1,1] (light blue) incurs by far less delay as compared with accessing the bitcell at [m,n] (dark blue). Note that the worst case takes place during a read operation when bitcell [1,1] is in 'AP' state and bitcell [m,n] is in 'P' state. Since the timing of the input and reference signals reaching the SA is critical, the reference signals must ensure enough signal margins including these worst case scenarios. The multi-row read approach further complicates the delay mismatch. Additionally, in terms of different column as shown in Fig. 1a, an accessed bitcell close to the wordline (WL) driver strongly enables pass transistor (NMOS, with high V_{GS}) compared to a cell far from the driver.

Therefore, we propose a scheme of placing the reference generators in such as way that the reference signals capture similar parasitic delays as those experienced by the aforementioned worst-case bitcell accesses. Hence, the reference cells (for e.g., AND operation) as described earlier, are split into two sub-cells, each with a resistance equivalent to the sum of the critical resistance states. These sub-cells are placed as the top Ref_{AT} and the bottom Ref_{AB} cells in each column, as shown in Fig. 3. This solves three purposes: 1) The parallel combination of the two reference sub-cells effectively ensures an average resistance of the two critical resistance states of AND logic operation. 2) The placement at the top and bottom ensures row-wise tracking as the reference signal encounters an average delay of the worst-case bitcell access delays. (3) Placed in each column, these cells ensure column-wise tracking since they suffer with similar WL voltage degradation as experienced by the accessed bitcells. In a similar way, read reference circuit is split to Ref_{RT} and Ref_{RB}, and CIM-logic OR to Ref_{OT} and Ref_{OB}.

III. Results

A. Chip Prototype and Experimental Setup

Fig. 4a shows the microscopic view of CoFeB based perpendicular magnetic tunnel junction (pMTJ) device [12].



(a) (b)

Fig. 4: (a) Microscopic view of CoFeB based pMTJ. (b) Optical image of the fabricated chip depicting 4×8 (32) memory banks.

Parameters	Specifications
Memory, Banks, Array	1 *Mb*, 32, 128 × 512
SA pitch, min. sensing margin	16 bitcells, 40mV
STT Device	$CoFeB$-based pMTJ [12]
Voltage Read/Write* (variations)	0.75 V/1.1 V, 0.9 V (±10 %)
CMOS (variations)	RVT, 28 nm TSMC (3 σ)
Temperature	-40°C to 125°C

TABLE I: Design parameters. *Separate core WL, periphery voltages.

Fig. 4b shows optical images of the fabricated characterization chip prototype which is experimentally verified for memory operations [12]. The STT-MRAM device model is calibrated using characterization array of 4Gbit pMTJ devices, out of which 1Mbit pMTJ are electrically active. Detailed design specifications are summarized in Table I.

B. Circuit-level Simulation Results

1) Evaluation of Wire Parasitics: Fig. 5 highlights the reduced read latency (associated with the required BL and RL discharge times) for four different CIM configurations; The reference block being placed at the 'top' (above the farthest row to the SA), 'bottom' (the nearest), at the 'centre' of the two equally split sub-arrays and the proposed 'split' configuration (two sub-cells at the top and bottom) as shown in Fig. 5a. Fig. 5b shows the spread of V_{BL} associated with Read1 and Read0 at the worst-case bitcell accesses (top-most, Row128) and (bottom-most, Row1) in the 'split' configuration. The worst-case signal margin arises between the cases Read1 at Row128 and Read0 at Row1. Fig. 5c shows that while each referencing scheme ensures that reference signals have average resistance state of the critical states, the average is calculated using effective resistance, for instance, in the 'top' configuration, when top bitcell is accessed. This adversely affects the time required to meet the minimum sensing margins for the bottom (the other worst case) bitcell access. A similar argument can be made for the 'bottom' configuration. In the 'split' configuration, the cells inherently achieves the desired average resistance while taking the top and the bottom worst cases into account. The 'centre' configuration exhibit similar considerations, however, since it affects the symmetry of the bitcell array, it is not considered in our methodology.

Fig. 6a shows the effect of degradation of the WL reaching the bitcell of the same row but with increasing column positions. A constant reference signal (V_{RO} or V_{RA}) which is generated independent and unaware of the column position fails to distinguish between the critical states. Our proposed

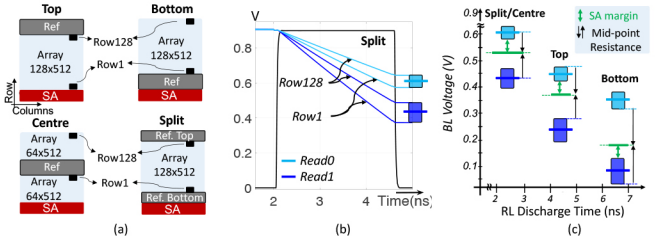


Fig. 5: (a) Different reference arrangements. (b) Variation in V_{BL} for topmost and bottom-most bitcell accesses. (c) Latency comparison.

scheme of placing the reference circuits along the row is adaptive to these undesired voltage drops and therefore, the reference signals (*V_{RO} or *V_{RO}) effectively stay near the middle of the critical states (refer to Fig. 3). In summary, the proposed 'split' configuration improves the performance and ensures high margins for high robustness and scalability.

2) Evaluation of PVT Variations: The validation and performance of the read operation for the global PVT (corner) cases, each with 3σ local variations, are presented in Fig. 6b. The corner cases are represented by [process, voltage, temperature] (including CMOS and STT-MRAM variations) along with the corresponding worst-case read latency. The spread of V_{BL} associated with the Read1, Read0 and spread of V_{RL} associated with the reference signal for each of these corner cases are shown in the figure. Slow corner understandably has a larger spread (more variations) and hence, the required minimum SA read sensing margin is delayed. The reference signals incur a smaller spread compared to the read signals since the reference signal is generated using four devices (two 'P' and two 'AP' states) where the individual spread of the STT-MRAM devices are averaged out. In short, in each of these corner cases, the reference signal is able to distinguish the input states, however, with different timing requirements.

In a similar way, Fig. 6c presents the global PVT (corner) cases, along with their 3σ local variations, for OR and AND operations. The sensing margins associated with the AND operation are smaller compared to that in the OR operation when identical time is invested in the two operations. Although the spread (variations) of STT-MRAM devices in the 'AP' state are larger compared to devices in the 'P' state, the inherent smaller ratio of the critical state resistances in the AND operation concedes to smaller sensing margins.

In summary, our methodology provides high robustness for read and logic operations in the presence of PVT variations.

3) Design Efficiency per Operation: The worst-case latency and average energy per operation are presented in Fig. 6d. For logic operations, XOR is the most time and energy consuming one, because it involves logical combinations of simpler AND and OR operations. Write operation is ∼10 ns, however, it consumes nearly 10× more energy (∼900 fJ) compared to logic operations (∼70-110 fJ) due to high compliance currents.

C. System-level Results

We evaluate the benefit of our STT-MRAM-based CIM design on two applications, namely binary neural network (BNN) and text encryption. Following is the brief summary:

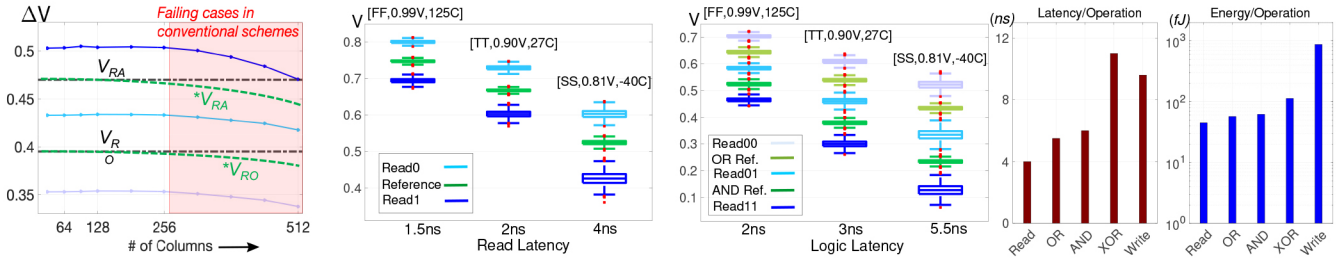


Fig. 6: (a) ΔV $(VDD - V_{BL})$ and $(VDD - V_{RL})$ degradation due to WL degradation with column index. Our proposed scheme maintains high margins in the error-prone red area. Sensing margins and latency for (b) read and (c) logic operations with worst-case PVT analysis. (d) Latency, energy per operation.

BNN: BNN is an efficient way of implementing NN on low-power embedded platforms, as it converts float/integer values into binarized weights and neuron activation. BNN used for our evaluation has two hidden layers with 1024 neurons each and takes binarized 28×28 input image from the MNIST dataset. We program the input vector of each layer on the top-most row and the weights to the rest of the rows in the crossbar. We perform XNOR operations inside the memory by pairing each weight with the input vector and the required post-processing at the periphery. The overhead of communication between the crossbar arrays is ignored. Results show that our design classifies one input image in $\sim$47 ns and consumes $\sim$211 nJ. It utilizes 36 CIM crossbars, each of size 128×512 and delivers 98% hardware accuracy. Compared to CIM baseline with similar accuracy (without the proposed scheme), our implementation achieves 1.8× better performance and energy-efficiency.

Text Encryption: An input text vector is encoded by performing bitwise XOR operation with a predefined key vector. In our implementation, first both the texts and the key are programmed to the crossbar. Second, the crossbar rows containing input text are activated sequentially while the row programmed with the key is active in all the steps. We assume each character has 8 bits implying that it has to be distributed over 8 cells. Fig. 7 shows that compared to the traditional software implementation on Intel Haswell processor using gem5 syscall emulation and CIM baseline (without the proposed scheme), we improve the performance by 130× and 1.4×, respectively. Table II lists the configuration of our CPU baseline architecture. In this simulation, we consider one crossbar which is reprogrammed with the incoming input text. It is clear that employing more crossbars would result in higher throughput improvements due to the parallelization.

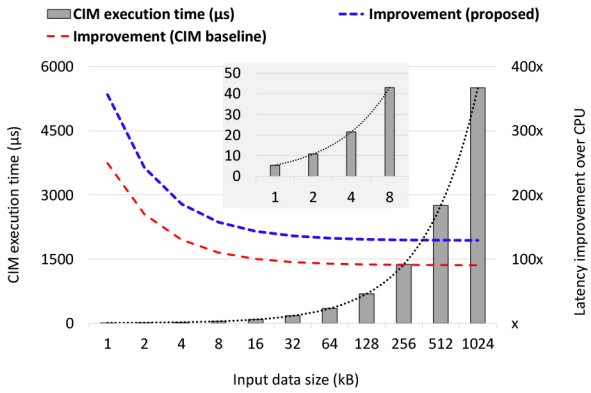


Fig. 7: Execution time to encrypt different input text sizes and the improvement achieved compared to the software implementation

Processor	X86, out-of-order, 3.6 GHz
L1 cache	64 kB I-cache with 64 B D-cache
L2 cache	256 kB with 64 B cache line size
L3 cache	8 MB with 64 B cache line size
Main memory	DDR3 with 8 GB

TABLE II: CPU simulation parameters

IV. Conclusion

This work presents a novel referencing scheme using STT-MRAM-based CIM to perform robust logic operations in the presence of design non-idealities. The paper validates our proposed design using calibrated design parameters extracted from a silicon-verified characterization chip. Performance metrics are determined for logic operations which are employed for a system-level framework and evaluated for BNN and text encryption applications. Results show that our implementation achieves up to 17.8 TOPS/W on the MNIST dataset and 130× performance *i.e.* execution time improvement is expected compared to software implementation on Intel Haswell processor.

References

[1] A. Singh *et al.*, "Low-power Memristor-based Computing for Edge-AI Applications," in *ISCAS*, 2021.

[2] M. Komalan *et al.*, "Cross-layer design and analysis of a low power, high density STT-MRAM for embedded systems," in *ISCAS*, 2017.

[3] S. Sakhare *et al.*, "Enablement of STT-MRAM as last level cache for high performance computing domain at 5nm node," in *IEDM*, 2018.

[4] S. Jung *et al.*, "A crossbar array of magnetoresistive memory devices for in-memory computing," *Nature*, 2022.

[5] S. Jain *et al.*, "Computing in memory with spin-transfer torque magnetic RAM," *TVLSI Systems*, 2017.

[6] Y. Pan *et al.*, "A multilevel cell STT-MRAM-based computing in-memory accelerator for binary convolutional neural network," *Transactions on Magnetics*, 2018.

[7] L. Zhang *et al.*, "A high-reliability and low-power computing-in-memory implementation within STT-MRAM," *Microelect.*, 2018.

[8] S. Resch *et al.*, "Pimball: Binary neural networks in spintronic memory," *ACM TACO*, 2019.

[9] H. Koike *et al.*, "1T1MTJ STT-MRAM cell array design with an adaptive reference voltage generator for improving device variation tolerance," in *IMW*, 2015.

[10] N. Xu *et al.*, "STT-MRAM design technology co-optimization for hardware neural networks," in *IEDM*, 2018.

[11] R. Bishnoi *et al.*, "Read disturb fault detection in STT-MRAM," in *ITC*, 2014.

[12] S. Rao *et al.*, "STT-MRAM array performance improvement through optimization of Ion Beam Etch and MTJ for Last-Level Cache application," in *IMW*, 2021.