2025 62nd ACM/IEEE Design Automation Conference (DAC) | 979-8-3315-0304-8/25/$31.00 ©2025 IEEE | DOI: 10.1109/DAC63849.2025.11132099

Late Breaking Results:

In-Memory Arithmetic: Enabling Division with Stochastic Logic

Farzad Razi*, Mehran Shoushtari Moghadam§, M. Hassan Najafi§, Sercan Aygun†, and Marc Riedel*
*Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN, USA
§Electrical, Computer, and Systems Engineering Department, Case Western Reserve University, Cleveland, OH, USA
†School of Computing and Informatics, University of Louisiana at Lafayette, Lafayette, LA, USA
{frazi, mriedel}@umn.edu, {moghadam, najafi}@case.edu, sercan.aygun@louisiana.edu

***Abstract*—Designing an efficient arithmetic division circuit has long been a major challenge. Traditional binary computation methods rely on complex algorithms that require multiple cycles, complex control logic, and substantial hardware resources. Implementing division with emerging in-memory computing technologies is even more challenging due to susceptibility to noise, process variation, and the complexity of binary division. In this work, we propose an in-memory division architecture leveraging stochastic computing (SC), an emerging technology known for its high fault tolerance and low-cost design. Our approach utilizes a magnetic tunnel junction (MTJ)-based memory architecture to efficiently execute logic-in-memory operations. Experimental results across various process variation conditions demonstrate the robustness of our method against hardware variations. To assess its practical effectiveness, we apply our approach to the *Retinex Algorithm* for image enhancement, demonstrating its viability in real-world applications.**

I. Introduction and Motivation

As CMOS transistors continue to scale into the *nanometer* regime, computing systems face increasing challenges such as short-channel effects, leakage currents, and manufacturing variability. These contribute to higher energy consumption and reduced reliability, limiting the efficiency of modern architectures [1], [2]. Addressing these issues requires solutions across the computing hierarchy, from device-level innovations to architectural advancements [3], [4]. At the device level, technologies like FinFET and gate-all-around carbon nanotube FETs (GAA-CNTFETs) enhance gate control, mitigating short-channel effects and reducing power leakage [2], [3]. Spintronic devices, such as magnetic tunnel junctions (MTJs), offer enhanced reliability, particularly for in-memory structures [5], due to their resistive nature, stochastic switching behavior, and inherent resilience to radiation. MTJs exhibit excellent integration potential, as they can be fabricated as an independent layer above conventional transistor-based circuits. This vertical integration makes them compatible with various transistor technologies, facilitating seamless integration into modern computing systems. At the architectural level, novel paradigms such as In-Memory Computing (IMC) and Stochastic Computing (SC) have been proposed to overcome the limitations of the traditional von Neumann architecture, where the separation of the processing and memory unit leads to inefficiency. IMC integrates logic operations within memory cells, thereby minimizing data movement, reducing memory access overhead, lowering leakage power, and enhancing overall performance by freeing up memory bandwidth [5]. SC, another emerging computing paradigm, deviates from traditional binary computing by representing values as randomly distributed bit-streams, enabling simple execution of complex arithmetic operations using basic logic gates. For example, *multiplication* can be performed using a single `AND` gate or addition can be implemented using a multiplexer (`MUX`) circuit. This inherent simplicity makes SC highly attractive for low-cost, energy-efficient arithmetic operations, particularly when integrated with IMC technologies to further enhance computational efficiency and scalability.

While extensive research has focused on hardware-efficient implementations of addition and multiplication, division remains significantly more complex due to its iterative and non-linear nature. Traditional hardware division relies on computationally expensive algorithms that require multiple cycles, complex control logic, and substantial hardware resources, making it inherently slower and more power-intensive than other arithmetic operations. Implementing division in IMC presents additional challenges, including precision loss, non-linearity of memory devices, and the need for iterative approximation techniques. Given these challenges and the application-level importance of division, this work proposes a simple and efficient in-memory division architecture leveraging SC. The proposed design utilizes in-memory `AND` and `OR` logic operations, efficiently implemented using FinFET technology and MTJ devices. Integrating MTJs into IMC structures substantially reduces power consumption and enhances energy efficiency [6]. To demonstrate the effectiveness of our approach, we implement a real-world application: the *Retinex Algorithm* for image enhancement [7].

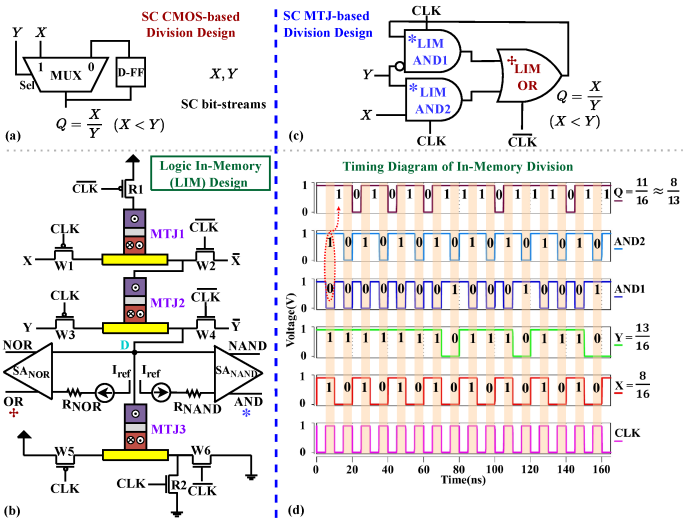

Fig. 1. Divider design: (a) SC circuit [8], (b) Proposed LIM gate-level circuit, (c) Schematic of the magnetic in-memory `AND`/`OR` gates, (d) Timing diagram of the input ($X=8, Y=13$) and output ($Q=\frac{8}{13}$) signals with N=16.

II. Proposed Approach

As illustrated in Fig. 1(a), division in SC can be implemented using a simple `MUX`-based design that includes a delay element (`D-FF`) [8]. The circuit operates on stochastic bit-streams, where the probability of observing a '1' represents the encoded data. A bit-stream of length N containing M '1's corresponds to the value $\frac{M}{N}$. The input bit-streams, X and Y, are processed using the `MUX` circuit to compute the output Q, where $Q=X$ when $Y=1$ and retains its previous value when $Y=0$. For instance, given input bit-streams $X=1010101010101010$ ($P_X=\frac{8}{16}$) and $Y=1111111011101110$ ($P_Y=\frac{13}{16}$), the output $Q=\frac{X}{Y}=1010101110111011$ ($P_Q=\frac{11}{16}$), which closely approximates the expected result of $\frac{8}{13}$.

To implement SC division within memory, we propose a `MUX`-based design leveraging logic-in-memory (LIM) techniques proposed in [9] (Fig. 1(b)). The proposed design is shown in Fig. 1(c). This architecture integrates in-plane anisotropy spin Hall effect magnetic tunnel junctions (SHE-MTJs) with FinFET transistors, enabling logical operations directly within memory. The circuit operates in two distinct phases—*preparation* and *evaluation*—controlled by a clock signal (CLK). In the *preparation* phase, CLK=0, so R1 and R2 transistors are off, and the magnetization of the MTJs' free layers is adjusted to store the input values through W1 to W6 transistors. For example, when $X=1$, the write current flows through W1 to MTJ1 and W2, whereas for $X=0$, the write current reverses direction. When CLK transitions to 1, transistors W1 to W6 turn off, and a read current flows through

TABLE I
IMC DIVISION ACCURACY (MAE: MEAN ABSOLUTE ERROR)

Bit-stream length, N	16	32	64	128	256	512	1024
MAE (%)	12.51	8.46	6.07	4.24	2.92	2.15	1.61

MAE is measured between the LIM divider *versus* accurate 8-bit binary division.

R1, MTJs, and R2. Voltage division occurs at node D between the equal resistances of MTJ1 and MTJ2, and the resistance of MTJ3, which remains constant. The dual sense amplifiers (SAs) compare the voltage at node D with reference voltages, generating AND/NAND and OR/NOR outputs, as shown in Fig. 1(b). In the proposed architecture (Fig. 1(c)), AND1 and AND2 synchronize with the CLK signal, while OR remains active in the opposite phase. Consequently, when CLK=0, OR operates in evaluation mode, providing its output to AND1, which is in the preparation phase. When CLK=1, the outputs of AND1 and AND2 are processed and fed into the OR inputs. The key advantage of the proposed design to the conventional CMOS-based SC division [8] is the elimination of the delay component (D−FF) through this clock difference technique, simplifying the design and improving efficiency.

III. EVALUATION

We conducted circuit-level simulations using HSPICE, as illustrated in Fig. 1(d). The proposed in-memory divider completes division in $(N+\frac{1}{2})$ cycles for two N-bit bit-streams. Simulation results, summarized in Table I, demonstrate that the proposed LIM-based divider achieves accuracy comparable to conventional CMOS-based SC division. For $N=16$ bit inputs, the design operates with a total power consumption of $152.2\mu W$, a preparation delay of $1.8ns$, and an evaluation delay of $2.3ps$. However, emerging technologies are often susceptible to fabrication process variations due to the intricate nature of manufacturing and the extremely small device dimensions. This vulnerability is particularly challenging when utilizing traditional binary computing [10]. To evaluate the design's sensitivity to process variations, we conducted a comprehensive Monte Carlo simulation across 500 iterations, incorporating a Gaussian distribution with $\pm 3\sigma$ variations (where σ is the standard deviation). We applied a 10% variation to critical FinFET parameters, including fin height (H_{fin}), gate length (L_g), fin thickness (T_{fin}), and gate oxide thickness (T_{ox}), as well as key MTJ parameters such as tunnel magnetoresistance (TMR), free layer area (Area), and resistance-area product (RAP). The simulation results, illustrated in Fig. 2, indicate that while fabrication process variations impact preparation and evaluation delays and power consumption, the output remains robust and stable.

IV. CASE STUDY: RETINEX ALGORITHM

The Retinex algorithm is an image enhancement technique designed to process low-illumination images. A given source image, $I(x,y)$, consists of two components: reflectance, $R(x,y)$, which represents the true color and texture of the image, and illumination, $L(x,y)$, which encodes lighting information. This relationship can be expressed as: $I(x,y)=R(x,y)\cdot L(x,y)$. The objective of the Retinex algorithm is to estimate $R(x,y)$ by normalizing the illumination, which is achieved through $R(x,y)=\frac{I(x,y)}{L(x,y)}$. To approximate $L(x,y)$, a Gaussian filter is applied: $L(x,y)=I(x,y)*G(x,y)$, where $G(x,y)$ is a Gaussian kernel [11]. The division operation involved in computing $R(x,y)$ is implemented using two distinct approaches: IMC and conventional binary arithmetic. The Gaussian filter for estimating $L(x,y)$ is computed in binary format and is shared across both approaches. Fig. 3 presents a comparative performance analysis of a traditional binary divider versus the proposed in-memory divider when applied to a low-illumination input image. The results indicate that the proposed SC-based approach (with N=256) achieves superior Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM), and color channel distribution compared to its binary counterpart. These findings highlight the efficiency of the proposed method at the application level, showcasing a novel application of SC beyond its well-known robustness to noise and soft errors. By leveraging the LOL (LOw-Light) dataset [12], the proposed approach achieves an average PSNR

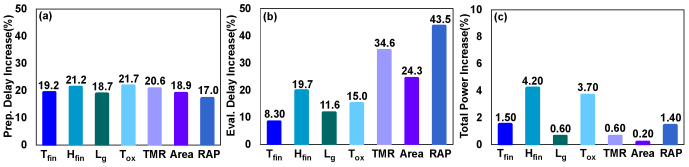

Fig. 2. Increment of performance parameters by fabrication process variations. (a) Preparation delay, (b) Evaluation delay, and (c) Total power.

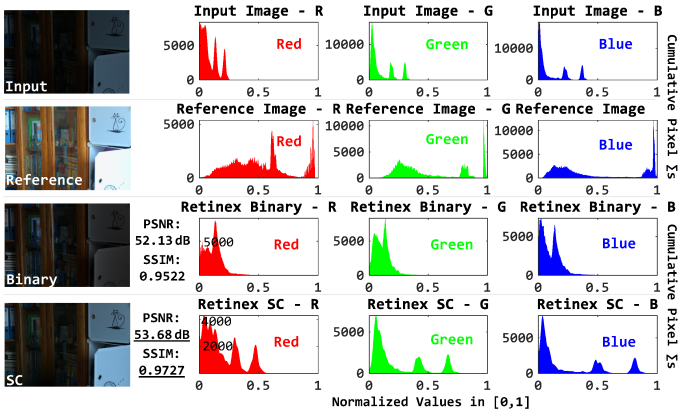

Fig. 3. LOL Dataset sample input performance: *Binary* vs. *Proposed* Division

improvement of $+0.374dB$ across all images, while the overall SSIM improvement is $+0.0034$.

V. CONCLUSION

This paper presents a novel in-memory SC divider leveraging Magnteic Tunnel Junction (MTJ) and FinFET transistors to enable efficient division with reduced power consumption and latency. By eliminating the D−FF through a clock-difference technique, the proposed design achieves a compact, high-performance operation, as validated through simulations. Monte Carlo simulation results confirm that despite variations in preparation and evaluation delays as well as power consumption, the overall output remains robust. This work extends the capabilities of process-in-memory architectures, offering a promising solution for low-power, high-performance computing, particularly for applications requiring division operations.

VI. ACKNOWLEDGMENT

This work is supported in part by the National Science Foundation under grants #2019511, #2339701, and a generous gift from NVIDIA.

REFERENCES

[1] T. Hiramoto, "Five nanometre CMOS technology," *Nat. El.*, vol. 2, no. 12, 2019.
[2] H. H. Radamson, Y. Miao, Z. Zhou, Z. Wu, Z. Kong, J. Gao, H. Yang, Y. Ren, Y. Zhang, J. Shi *et al.*, "Cmos scaling for the 5 nm node and beyond: Device, process and technology," *Nanomaterials*, vol. 14, no. 10, p. 837, 2024.
[3] Q. Zhang, Y. Zhang, Y. Luo, and H. Yin, "New structure transistors for advanced technology node cmos ics," *National Science Review*, vol. 11, no. 3, p. nwae008, 2024.
[4] J. Seekings, P. Chandarana, M. Ardakani, M. Mohammadi, and R. Zand, "Towards efficient deployment of hybrid snns on neuromorphic and edge ai hardware," in *IEEE ICONS*, 2024.
[5] F. Razi, M. Hossein Moaiyeri, and S. Mohammadi, "A magnetic reconfigurable ternary nor/nand logic for logic-in-memory applications," in *Spin*, vol. 11. World Sci., 2021.
[6] M. Morsali, S. Tabrizchi, R. T. Velpula, M. B. S. Muthu, H. P. T. Nguyen, M. Imani, A. Roohi, and S. Angizi, "Energy-efficient near-sensor event detector based on multilevel ga2o3 rram," in *ISVLSI*, 2024.
[7] S. Munaf, A. Bharathi, and A. N. Jayanthi, "Fpga-based low-light image enhancement using retinex algorithm," *NATURE Sci. Reports*, vol. 14, no. 1, 2024.
[8] T.-H. Chen and J. P. Hayes, "Design of division circuits for stochastic computing," in *2016 ISVLSI*, 2016, pp. 116–121.
[9] F. Razi, M. H. Moaiyeri, and S. Mohammadi, "Toward efficient logic-in-memory computing with magnetic reconfigurable logic circuits," *IEEE Magn. Let.*, vol. 13, pp. 1–5, 2022.
[10] F. Khodayari, A. Amirany, K. Jafari, and M. H. Moaiyeri, "Low-cost and variation-aware spintronic ternary random number generator," *Circ., Sys., & Sig. Proc.*, vol. 43, no. 2, 2024.
[11] B. Petro, C. Sbert, and J.-M. Morel, "Multiscale retinex," *IPOL: Image Proc. On Line*, 2014.
[12] W. Xiong, D. Liu, X. Shen, C. Fang, and J. Luo, "Unsupervised low-light image enhancement with decoupled networks," 2022.