Vol. 08, No. 1 (2026) 331-340, doi: 10.24874/PES08.01A.006

Proceedings on Engineering Sciences

www.pesjournal.net

NOVEL IN-MEMORY COMPUTING CIRCUIT USING MULLER-C ELEMENT

Gururaj A Tapashetti
Umadevi S[1]

Received 18.04.2025
Revised 11.06.2025
Accepted 17.07.2025

Keywords:

Von Neumann bottleneck, Memory wall, SRAM, In-Memory Computing (IMC),Parallel execution

A B S T R A C T

The proposed research work addresses the von Neumann bottleneck, a limitation in computing systems where data transfer between memory and processor components slows down performance. In contrast, In-Memory Computing (IMC) reduces energy consumption and enhances computing efficiency. The research introduces an 8+T Static Random Access Memory *(SRAM) IMC circuit, built using 8+T differential SRAM technology, and explores its applications. The 8+T SRAM IMC circuit enables parallel execution of all logic operations, including concurrent bitwise and SRAM read functions. It supports NAND, NOR, and XOR operations simultaneously, outperforming conventional IMC circuits in speed by 38%, 16%, and 49%, respectively, while consuming 18% less power. Additionally, a multiplier implemented with this IMC circuit demonstrates a 40% reduction in power and a 36% increase in performance compared to conventional multipliers. The study provides a detailed investigation of the 8+T SRAM IMC circuit and its design parameters.*

1. INTRODUCTION

The von Neumann architecture, introduced by John von Neumann in the 1940s, has become the cornerstone of modern computing systems. In this design, the central processing unit (CPU) and memory are distinct components, requiring data and instructions to be exchanged between them via a shared communication channel, or bus. This architecture was groundbreaking at the time, as it enabled flexible programming and formed the basis for the development of modern computers. However, despite its long-standing use, this design presents certain limitations, particularly when applied to today's high-performance computing environments.

One of the key issues with the von Neumann architecture is the so-called "von Neumann bottleneck," or "memory wall." This term refers to the performance bottleneck caused by the gap between the speeds of the CPU and memory. While processor performance has advanced tremendously over the past few decades—thanks to Moore's Law and the continued miniaturization of transistors—the performance of memory access has not kept pace. This discrepancy between the speed of the CPU and memory access has become a major limiting factor in overall system performance.

In a conventional von Neumann system, data must constantly be transferred back and forth between memory and the CPU. This data movement consumes significant energy, and as the performance gap between memory and the CPU grows, the energy and time required for data transfers increase, negatively affecting the overall system performance. The energy consumption involved in

[1] Corresponding author: Umadevi S
Email:umadevi.s@vit.ac.in

transferring data between memory and the CPU, along with the delay it introduces, has become a critical challenge in modern computing, particularly in data-intensive applications such as artificial intelligence, big data analytics, and high-performance scientific computing.

The von Neumann bottleneck is largely due to the separation of computation and memory, which forces the system to rely on slow memory access and high data transfer energy. As a solution, researchers have been exploring alternative computing paradigms, and one of the most promising concepts that has emerged is IMC. IMC is an approach that seeks to overcome the limitations of traditional architectures by integrating computational capabilities directly into the memory array, effectively reducing or eliminating the need to transfer data between memory and the CPU.

IMC proposes that, instead of moving data back and forth between the CPU and memory, computations can be performed where the data resides—in the memory itself. This concept significantly reduces the energy and time spent on data transfers, thereby boosting overall system efficiency. Moreover, IMC offers the potential for parallel processing and data access, leading to improvements in performance for certain types of applications, especially those involving large datasets and frequent memory access. As a result, IMC is seen as a key technology for mitigating the memory wall problem and improving energy efficiency in computing systems.

In particular, recent studies have concentrated on developing IMC circuits that leverage Static Random Access Memory (SRAM) cells, as SRAM is commonly used in cache memory and offers high-speed data access. One such approach is based on 8+T Differential SRAM, a type of SRAM cell that incorporates additional transistors to facilitate logical operations within the memory array. This study introduces an IMC circuit built on 8+T Differential SRAM as a solution to the von Neumann bottleneck, allowing for the execution of logic operations directly within the memory.

In the context of this research, the proposed IMC circuit provides significant advantages over conventional memory architectures. Unlike standard 8+T SRAM cells that only store and retrieve data, the proposed IMC circuit is capable of performing logic operations such as NAND, NOR, and XOR directly within the memory array. By integrating these logic gates into the memory, the IMC circuit can execute certain computations in parallel, reducing the need for data transfers to the CPU and improving system performance.

A key feature of the proposed IMC circuit is its ability to perform simultaneous NAND, NOR, and XOR operations, which is a notable improvement over existing IMC designs that typically focus on individual logic operations. Furthermore, the circuit is optimized for faster XOR operation delays, which can be beneficial for a variety of applications that rely on efficient XOR computations, such as error detection and cryptography.

To demonstrate the capabilities of this IMC circuit, the researchers propose implementing a 2x2 array multiplier using the logic operations derived from the circuit. Multiplication is a fundamental operation in many computing applications, and by using the IMC circuit to perform the multiplication within the memory, the system can achieve significant improvements in speed and energy efficiency. The 2x2 array multiplier serves as a proof of concept for the broader potential of IMC circuits to handle more complex operations in future implementations.

By reducing the need for frequent data transfers between memory and the CPU, the IMC circuit based on 8+T Differential SRAM helps to alleviate the von Neumann bottleneck, offering a more efficient and energy-conscious approach to computing. This research represents a step forward in the development of **next-generation computing architectures** that blend logic and memory, enabling faster and more energy-efficient computations in data-intensive tasks.

In conclusion, the proposed IMC circuit addresses a critical limitation of the traditional von Neumann architecture by embedding computational logic directly into the memory. The use of 8+T Differential SRAM allows for simultaneous execution of essential logic operations, which not only reduces memory access times but also improves energy efficiency. The demonstration of a 2x2 array multiplier based on the IMC circuit further highlights its potential for real-world applications, providing a glimpse into the future of memory-centric computing architectures that could redefine the landscape of modern computing systems.

2. LITERATURE REVIEW

Agrawal et al. (2018) stated the application of 8T SRAM cells in IMC scenarios. They highlight the benefits of 8T cells, particularly their independent read and write ports, which facilitate efficient read-compute-store processes within the memory array. The authors demonstrate that 8T SRAM cells enable a variety of Boolean logic operations, including XOR, IMP, NAND, and NOR, thereby enhancing digital vector computations in SRAM arrays.

Song and Kim (2021) found that through a research on IMC as a potential solution for the von Neumann bottleneck. By integrating logic into the memory array, IMC aims to improve performance and reduce data transfers between the memory and the processor. IMC enhances system performance and energy efficiency by performing calculations directly within the memory. Song and Kim introduced a novel logic circuit for IMC

that utilizes 8+T differential SRAM. This circuit, which employs a Muller C-element, demonstrates faster performance compared to conventional IMC methods, enabling simultaneous execution of NAND, NOR, and XOR operations. Their proposed IMC circuit shows significant gains in bitwise operation speed, underlining its potential to enhance computer efficiency.

Song and Kim (2022) stated in a study which presents an 8+T SRAM In-Memory Computing (IMC) circuit built on an 8+T differential SRAM, introducing both an IMC full adder (FA) and an approximate adder. These adders take advantage of the simultaneous data reading and computation abilities provided by the 8+T SRAM IMC circuit, leading to improved performance in terms of speed and energy efficiency. The research compares various adder designs, including the 8+T SRAM Read + FA, IMC approximation adder, SRAM read access, and the newly proposed IMC FA. The findings reveal that the IMC adders show notable enhancements in speed and energy savings when compared to conventional approaches. Furthermore, Ielmini and Wong (2018) have investigated the application of resistive switching devices in IMC, contributing to the broader research landscape focused on improving computational efficiency by merging memory and processing functions.

Ielmini and Wong (2018) stated that the structures and functions of various memory technologies, including MRAM, RRAM, and phase-change memories. He emphasizes the importance of resistive switching memory (RRAM) in IMC circuits, especially for applications like neuron integration, and fire and synaptic potentiation/depression. Research also explores multilayer RRAM programming techniques within hardware neural networks, discussing how analog ReRAM non-ideality can affect neuromorphic computing performance.

Reuben et al. (2020) found about an overview of memristors and various configurations of non-volatile memory arrays. Memristors, known for their ability to store data through resistance and enable efficient computation, hold significant potential for IMC applications. Their paper serves as a foundational resource for understanding the role of memristive logic in the post-CMOS era.

Yeswanth and Acharya (2021) stated the importance of digital logic, SRAM, and IMC in modern systems. The authors focus on the limitations of the von Neumann architecture, particularly regarding its inefficiencies in energy consumption and performance, which have prompted the exploration of IMC as a way to enhance system throughput and overall effectiveness. The paper discusses the implementation of IMC using 8T SRAM, especially for NAND and NOR operations, and stresses the vital role of digital logic and SRAM in computing systems. The authors highlight that the number of transistors in integrated circuits has been increasing at a rapid rate, doubling approximately every two years, in line with Moore's Law. This has led to increasing complexity and the need for more efficient processing architectures. A key issue with the von Neumann architecture is the energy cost and performance degradation associated with transferring data between processing units and memory, particularly for high-demand applications like artificial intelligence (AI). The concept of IMC aims to address this by reducing the need for data movement between memory and processors, thereby improving both performance and energy efficiency. The document provides insights into the circuit designs and threshold voltage calculations required to implement IMC with 8T SRAM, particularly for executing NAND and NOR operations. By minimizing data transfers and embedding logic operations directly into memory, the proposed approach offers significant improvements in efficiency and performance for modern computing systems

Gauchi et al. (2019) introduces a method for assessing the interconnect costs in data-centric architectures designed for IMC. They suggest dividing the memory into several sub-tiles to achieve substantial improvements in both energy efficiency and processing speed, compared to using a single large IMC memory unit. Their research primarily targets data-centric architectures and highlights how IMC can enhance memory throughput, system performance, and energy consumption. By utilizing multiple smaller memory instances connected through wire interconnects, their approach shows a potential 78% decrease in energy usage and a 49% improvement in processing speed relative to a single large IMC unit. The study provides a detailed examination of IMC applications, emphasizing the importance of precise memory sizing and interconnect modeling. This research is particularly relevant for memory designers aiming to develop scalable IMC systems capable of mitigating the challenges associated with the memory wall.

Gupta and Acharya (2021) explored the use of SRAM cells for implementing IMC, with a focus on 8T and 9T SRAM configurations. Their research delves into how these configurations can perform Boolean logic operations such as NAND, NOR, AND, and OR directly within the memory. The study compares the performance of these different SRAM architectures, examining their functionality during read and write operations. The authors emphasize the role of sense amplifiers and the integration of extra transistors to enhance performance. They also highlight how 9T SRAM offers distinct advantages over 6T and 8T SRAM configurations, particularly in reducing leakage current, as well as improving read and write access times. The study presents how NAND, NOR, AND, and OR logic gates are implemented using 9T SRAM for IMC, demonstrating the improved energy efficiency and overall effectiveness of this approach.

Lue et al. (2021) explored the performance of Flash-based computing-in-memory (CIM) systems, utilizing a vertical split-gate Flash device. They emphasize the advantages of this technology in minimizing data movement and reducing power consumption. The study suggests that Flash CIM can serve as an efficient alternative to traditional SRAM and ALU for multiply-accumulate (MAC) computations, facilitating high parallelism and increasing overall computational throughput. The researchers highlight that Flash CIM, with stationary weights, can significantly decrease data movement, by up to 85%, making it especially beneficial for neural networks with heavy workloads. The vertical split-gate architecture supports high-density and cost-effective Flash memory, enabling parallel MAC operations and multi-core processing. Their approach prioritizes non-volatile memory and high-density Flash technology, aiming to improve both computational speed and energy efficiency through enhanced parallelism.

Chen et al. (2019) presented a reconfigurable 8T SRAM design aimed at addressing the memory bottleneck in the von Neumann architecture by incorporating logic functionality within the memory array. This innovative design enhances memory performance by enabling not only data storage but also operations such as ternary content-addressable memory (TCAM), left-shift, and right-shift operations. These functionalities are achieved through adjustments to the peripheral circuitry, which introduces multiplexers between the read bit line and output registers, enabling efficient execution of these tasks. The 8T SRAM's capability to operate at a lower supply voltage compared to conventional 6T SRAM helps reduce design complexity and mitigates variability-related challenges. Importantly, the reconfigurable nature of the peripheral circuitry allows for the introduction of TCAM and shift operations without modifying the fundamental structure of the 8T SRAM cell.

Chen et al. (2020) proposed a 2T2R ReRAM architecture designed to support ternary content-addressable memory (TCAM), in-memory logic operations, and dot product computations, particularly suited for applications like deep neural networks (DNNs). Their design emphasizes improvements in latency and power efficiency, particularly for big data and machine learning tasks. By utilizing reconfigurable sense amplifiers and advanced word-line drivers, the architecture facilitates a variety of operations without the complexity of traditional bit cells. The TCAM functionality is particularly valuable for high-speed search operations in applications like network routing and database management. Furthermore, the architecture supports efficient in-memory logic and dot product calculations, reducing delays and power consumption associated with frequent memory access. With its split word-lines and 2T2R ReRAM design, this architecture offers high-density storage and accelerates processing for data-intensive applications.

Rajput and Pattanaik (2020) examines the use of 8T SRAM bit-cells to implement arithmetic circuits for IMC, with the goal of addressing challenges such as the memory wall, power wall, and instruction parallelism wall in modern computing systems. It details the functioning of half adder and half subtractor circuits and introduces a proposed sensing scheme aimed at enhancing energy efficiency. The authors identify the limitations inherent in von Neumann architectures and propose the CIM architecture as a potential solution. They stress the significance of IMC using 8T SRAM cells to facilitate boolean logic operations and arithmetic functions directly within the memory array. The paper's experimental approach, findings, and discussions center on comparing the energy efficiency and performance of the 8T SRAM bit-cell against traditional SRAM cells, demonstrating notable advancements in energy consumption and read noise margin.

Kulkarni et al. (2010) investigates the design and performance of an 8T SRAM bitcell in comparison to the conventional 6T bitcell commonly used in SRAM caches. The research aims to enable low-voltage operation while addressing the critical challenges of read stability and write-ability. The authors show that the 8T bitcell is capable of overcoming design challenges often encountered in traditional 6T SRAM bitcells during read and write operations. By using separate transistors for read and write access, the 8T bitcell offers several key advantages, such as read-disturb-free operation, differential sensing, and dual-port functionality. Data from a 90-nm CMOS test chip demonstrate that the 8T bitcell has fewer read and hold failures, better write-ability, and reduced leakage compared to a 6T bitcell of similar area, all while maintaining comparable performance.

Wu et al. (2010) focuses on the development of a 65nm Embedded Low Power SRAM Compiler, emphasizing a versatile design approach tailored for SRAM compiler platforms. The paper details the design of a configurable embedded low-power SRAM compiler built on a 65nm CMOS process, capable of producing various SRAM IP module files. The design methodology integrates a semi-automated design flow, which can be adapted to different types of regular circuits and allows for migration across various technology nodes. Additionally, the authors introduce both a single-port and a dual-port SRAM compiler within the 65nm CMOS framework, highlighting the platform's flexibility in generating different types of SRAM IPs.

Ebrahimi-Azandaryani et al. (2019) introduced a new adder architecture aimed at minimizing energy consumption, known as the Block-based Carry Speculative Approximate Adder (BCSA). This design focuses on reducing the length of the carry chain, enhancing accuracy, and lowering the frequency of output errors. They compared the BCSA adder with various other approximate adders by analyzing factors

such as energy use, delay, area, and output quality. Their findings show a significant 50% reduction in the cost function.The BCSA adder operates by dividing the adder into distinct blocks, each non-overlapping, and speculates on the carry output using the input values of the current and subsequent blocks. This approach effectively limits the carry chain to a maximum of two blocks, even in worst-case scenarios, which leads to reduced average delay. Additionally, the authors included an error detection and recovery mechanism to improve accuracy and further lower the output error rate. When compared to leading approximate adders, the BCSA adder demonstrated significant enhancements in energy efficiency, delay reduction, area optimization, and output quality. The architecture's focus on partitioning and carry speculation allows it to achieve higher performance with minimal energy consumption.

Xu et al. (2018) examines the design and evaluation of various accuracy-configurable adders, emphasizing the Simple Accuracy-Reconfigurable Adder (SARA). It provides an overview of SARA's operation, architecture, and advantages relative to other adder types. SARA consists of segments made up of subadders that deliver carry predictions to enhance critical delay optimization. This design enables switching between precise and approximate operational modes, which significantly affects power usage and delay. The architecture of SARA minimizes delay by leveraging carry predictions, creating a balance among accuracy, power consumption, and efficiency in delay.

Akbari et al. (2016) introduces a Reconfigurable Approximate Adder called RAP-CLA, capable of operating in both exact and approximate modes. The paper outlines the adder's design, analyzes its accuracy, and compares its performance with other approximate adders. A key feature of the RAP-CLA is its ability to switch between exact and approximate modes without needing an external correction mechanism. The adder achieves lower delay and power consumption compared to similar adders, albeit with a trade-off in accuracy. The study also highlights the adder's suitability for image processing tasks, showing improved peak signal-to-noise ratio (PSNR) over other designs.

Gurumurthy and Prahalad (2010) explore the design and implementation of a 16x16 multiplier based on principles of Vedic Mathematics, specifically utilizing the Karatsuba-Ofman algorithm. They evaluate the performance of their proposed "Array of Array" multiplier against the traditional Booth Radix-4 multiplier, focusing on aspects such as delay, power dissipation, and resource utilization. The design of the 16x16 multiplier is structurally derived from a 4x4 block and incorporates the Urdhva Triyagbhyam multiplication sutra. The multiplier was simulated using VHDL and implemented on a Xilinx FPGA SPARTAN-3E device. The findings demonstrate that this new multiplier significantly decreases delay, power consumption, and resource usage when compared to the Booth Radix-4 multiplier. This design aims to improve speed, efficiency, and resource utilization in various applications, including signal processing, image processing, and server technologies.

Rathod et al. (2020) explores different types of multipliers, such as Vedic, Array, and CIFM multipliers, along with their respective architectures and implementations. It also addresses floating-point and complex multipliers, focusing on their structures and operational mechanisms. The Vedic multiplier uses ancient multiplication methods, particularly the Urdhva-Tiryakbhyam sutra, to perform multiplication for 24x24 bit numbers. In contrast, the Array multiplier is a basic and sequential design that employs add-and-shift techniques for performing multiplication. The CIFM multiplier optimizes the 24-bit multiplication process by breaking it down into four parallel 12-bit modules, which leads to notable reductions in power consumption and computation time. The floating-point multiplier is designed to handle the sign, exponent, and mantissa bits effectively to ensure precision in floating-point arithmetic. Finally, the complex multiplier facilitates the multiplication of complex numbers by integrating their real and imaginary components, followed by floating-point addition or subtraction to finalize the results.

Kang et al. (2014) introduces an in-memory classifier architecture that combines data storage and computation in a standard 6T SRAM, addressing limitations of traditional digital accelerators. The authors present a system overview, an algorithm for training the classifier, circuit-level design details, and prototype measurement results. Their analysis highlights the benefits of the in-memory architecture in terms of bandwidth, latency, energy, and signal-to-noise ratio (SNR), showcasing significant energy savings and improved efficiency compared to traditional systems.

Zhang et al. (2017) explores the power and performance factors in CVN architecture with regard to Boolean logic functions, introducing In-Memory Computing (IMC) as a method to reduce both power usage and latency. The authors propose a 4+2T SRAM design aimed at enhancing searching and IMC processes, which functions effectively at a low supply voltage of 0.3 V. The CVN architecture in their research covers key operations such as memory read cycles, data transfer, and computation within the ALU, all of which contribute to power consumption and delay. By processing data directly within the on-chip memory, IMC significantly minimizes both power and latency. The proposed 4+2T SRAM design not only facilitates efficient searching and in-memory computing but also achieves greater area efficiency and reduces energy consumption with a low VDDmin of 0.3 V.

Dong et al. (2017) presents a configurable memory circuit capable of executing logical operations within the

memory array. This innovation enhances energy efficiency and performance by minimizing data transfers and allowing the Arithmetic Logic Unit (ALU) to focus on more complex calculations. The authors delve into the structure of traditional Content Addressable Memory (CAM) designs, the various operational modes of Static Random Access Memory (SRAM), and the concept of conducting logic operations within memory. Their research emphasizes that this configurable memory circuit facilitates bit-wise logical operations on data words held in memory, resulting in improved energy efficiency and overall performance. In standard CAM architectures, data is organized in rows, with XNOR gates used for bit-wise comparisons and match-line sense amplifiers indicating whether there is a match or mismatch. When functioning in SRAM mode, the memory operates in its conventional manner, handling reads and writes row-wise, while the incorporation of logic operations in memory permits specific logical manipulations on the row-wise stored SRAM data.

Jeloka et al. (2016) presents a novel approach to developing a high-bandwidth, low-power DRAM technology, first introduced at the ISSCC 2014 conference. The paper describes the overall architecture, testing strategies, and the technical challenges faced in implementing stacked DRAM with a microbump interface. A logic-interface chip is included between the interposer and the stacked DRAM to improve both reliability and testability. This chip consists of a base logic die, core DRAM with dual channels, and a physical layer (PHY) that enables communication between the DRAM and the memory controller. Several testing techniques, such as impedance monitoring, pad-loopback tests, and serial port analysis, are employed to detect and troubleshoot issues within the stacked DRAM configuration. In addition, the design incorporates innovative features like a semi-independent row and column command interface, along with a single-bank refresh mechanism to improve performance and test efficiency. The collaborative efforts of the authors—Lee, D. U., Kim, K. W., Kim, H., Kim, J. Y., Park, Y. J., and Hong, S.—mark a significant step forward in advancing high-bandwidth memory technologies.

Lee et al. (2014) explores the design and benefits of Spin-Transfer Torque Computing-in-Memory (STT-CiM), which is an advanced form of Spin-Transfer Torque Magnetic RAM (STT-MRAM) that supports in-memory computing (IMC). The paper highlights the advantages STT-CiM offers in terms of reduced memory energy consumption and improved system performance. It also provides a review of related research on STT-MRAM technology, alongside a detailed analysis of the STT-CiM design, its architectural improvements, and experimental results. The authors underscore how STT-CiM enhances system efficiency by enabling operations within the memory itself. Additionally, the introduction of Instruction Set Architecture (ISA) extensions for programmable processors demonstrates the potential of STT-CiM to improve both energy efficiency and overall performance in memory systems.

3. PROPOSED METHODOLOGY

The IMC circuit presented in this research is fundamentally based on 8+T Differential SRAM. This innovative design plays a crucial role in boosting the efficiency of computing systems by minimizing the data transfer requirements between memory and processors, effectively tackling the challenges posed by the von Neumann bottleneck. As illustrated in Figure 1, the proposed IMC architecture utilizes an 8+T Differential SRAM configuration, incorporating inverters along with a Muller C-element to facilitate essential logic operations, including NAND, NOR, and XOR.

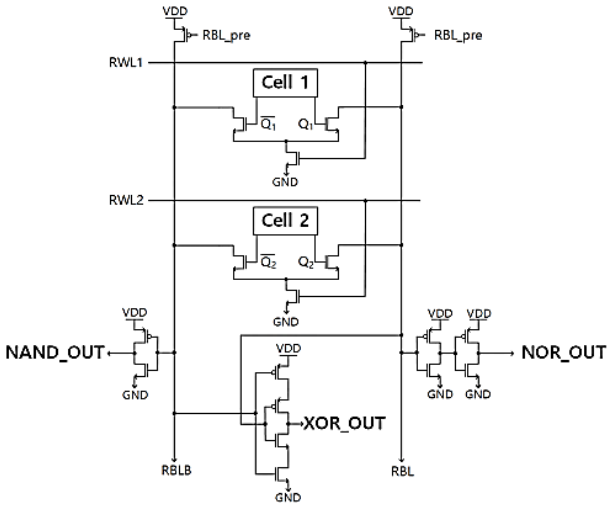

Figure 1. Proposed IMC

Traditionally, SRAM cells primarily serve the purpose of data storage and retrieval. However, the integration of logic elements directly within the memory array enables the proposed IMC circuit to perform fundamental logic operations internally, leading to a notable enhancement in system performance. This integration reduces the reliance on frequent data transfers between memory and the processor, which in turn decreases both energy consumption and computational latency. The circuit relies on inverters and the Muller C-element as its foundational components to execute these logic operations.

The inverter performs the NAND operation when it receives input from a specific node, referred to as RGBB. The buffer, which consists of two inverters, is responsible for computing the NOR operation when it receives input from another node, labeled RBL. The Muller C-element, a widely used component in asynchronous circuits due to its ability to synchronize signals, is used in this case to perform the XOR operation. The inputs for the Muller C-element are derived from nodes RBL and RBLB, and its ability to perform XOR operations with high speed and accuracy is a key advantage of this design.
The operational process initiates with both nodes, RBL and

RBLB, being pre-charged to a value of '1'. The discharge of the RBL node occurs when both word lines, RWL1 and RWL2, are activated at the same time and when any cell in the memory array has a stored value of '1'. As this node discharges, the associated buffer, consisting of two inverters connected to the RBL node, generates an output of '0'. Likewise, the RBLB node also starts with a pre-charged value of '1', but it discharges under the same conditions—when RWL1 and RWL2 are simultaneously selected, provided that at least one memory cell contains a '0'. After discharging, the inverter linked to the RBLB node produces an output of '1'.

The Muller C-element plays a critical role in the XOR operation by receiving inputs from both the RBL and RBLB nodes. Initially, the C-element is pre-charged to '0', and it only generates an output when both input values are identical. When the conditions are met, and Q1 (the value from RBL) is '1' and Q2 (the value from RBLB) is '0', both nodes discharge to '0', resulting in the C-element producing a value of '1'. Conversely, when the values of Q1 and Q2 are both '0' or both '1', the C-element retains its initial state of '0'.

The logic operations generated from this IMC design enable the construction of a half adder, an essential component in digital arithmetic. A half adder is a combinational circuit that sums two single-bit binary inputs, producing both a sum and a carry output. By leveraging the NAND, NOR, and XOR logic operations, the IMC circuit successfully realizes the half adder. This half adder then serves as a fundamental building block for creating a more intricate circuit—a 2x2 array multiplier, as illustrated in Figure 2.

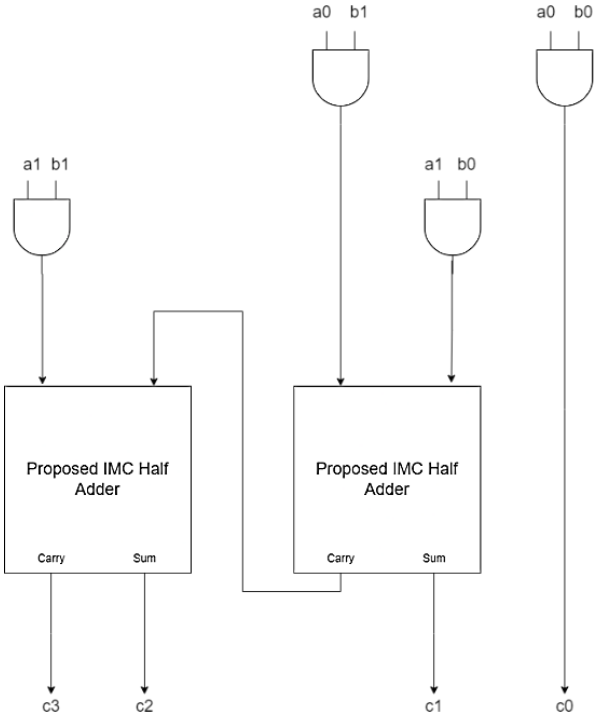

Figure 2. IMC based 2X2 array multiplier

An array multiplier is a type of digital combinational circuit specifically designed for multiplying two binary numbers. It achieves this by organizing multiple full adders and half adders in a structured array. This arrangement enhances the efficiency of binary multiplication, as it enables the concurrent processing of several product terms, which are subsequently combined to yield the final output. The multiplication operation starts with an array of AND gates, which produce the various partial product terms. Each of these terms represents the result of the bitwise multiplication of the corresponding bits from the multiplier and multiplicand. Finally, the partial products are added together using the array of adders to generate the complete product.

Traditionally, binary multiplication required a sequence of add and shift operations to generate the partial products and combine them. In this sequential approach, each bit of the multiplier is examined in turn, and the corresponding partial product is generated by shifting the multiplicand and adding it to the previous partial product. This process can be time-consuming, as it requires multiple clock cycles to complete the operation.

However, an array multiplier offers a significant improvement in performance by allowing all partial products to be generated and summed simultaneously. This eliminates the need for sequential operations and enables the multiplication of two binary numbers to be completed in a single micro-operation. The only time required for this operation is the propagation delay through the combinational logic gates of the multiplier array. As a result, the array multiplier is considerably faster than traditional multiplication methods, especially in applications where large numbers of multiplications must be performed.

Despite its advantages in terms of speed, the array multiplier was initially considered inefficient due to the large number of logic gates required to implement it. This limitation made array multipliers impractical for early computing systems, which were constrained by the high cost and limited availability of hardware components. However, with the advent of integrated circuits and advances in semiconductor technology, the cost and size of logic gates have decreased dramatically, making the array multiplier a viable and highly efficient solution for modern computing systems.

To illustrate the construction of a simple array multiplier using combinational logic, consider the example of multiplying two 2-bit binary numbers. Let the multiplicand be represented by bits b1 and b0, and the multiplier by bits a1 and a0. The resulting product will be a 4-bit value, denoted as c3c2c1c0. The AND gates in the multiplier array are used to generate the partial products, which are then combined using the half adders and full adders to produce the final result.

By employing the IMC circuit's logic operations to implement a half adder and then using the half adder to construct a 2x2 array multiplier, this study demonstrates the efficiency and practicality of the proposed design. The integration of logic operations into the memory array significantly reduces the need for data transfer, improves energy efficiency, and accelerates the overall computational process, particularly in arithmetic operations such as multiplication.

4. SIMULATION AND RESULTS

4.1 Proposed IMC

Figure 3 illustrates the outputs of the proposed IMC circuit, demonstrating how operations such as NOR, NAND, and XOR are realized based on the inputs from nodes wbl1 and wbl2. The resulting outputs are contingent on the values provided to these input nodes.

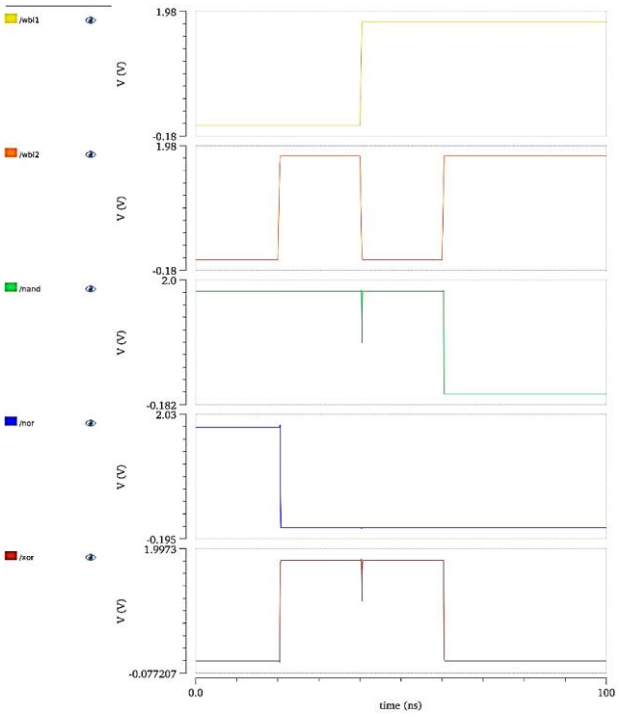

Figure 3. Simulated waveform of IMC

Table 1 outlines a comparison of logic computation delays across three different configurations: the 8+T SRAM IMC circuit, the 8+T SRAM combined with standard logic gates, and the 8T SRAM with skewed inverters. This serves as a benchmark and includes a virtual scenario for the 8+T SRAM with integrated logic computation.

In the configuration where 8+T SRAM is combined with traditional logic computation, standard logic gates are integrated alongside the SRAM cell. In contrast, the 8+T SRAM IMC circuit directly connects the SRAM cell to inverters and a Muller C-element, which sets it apart from the other setups. The 8+T SRAM IMC circuit has the advantage of performing simultaneous logic computations across two selected words, whereas the 8+T SRAM with traditional logic computation can only process one word at a time. Likewise, the 8T SRAM with skewed inverters also supports logic operations when two words are selected concurrently. Notably, this configuration shows lower power consumption in the IMC setup when compared to conventional logic processing approaches.

Table 1. Comparison of delay and power in Conventional logic computation and IMC circuit logic computation.

Circuits	NOR Delay (ps)	NAND Delay (ps)	XOR Delay (ps)	Power (mW)
8+T SRAM Read Operation + Conventional logic computation	392.4	253.78	459.9	3.838
8+T SRAM IMC Circuit	338.3	183.9	308.7	3.253

4.2 Proposed 2X2 Array Multiplier

The output of the IMC based 2X2 array multiplier has been shown in the figure 4., where there are two 2-bit inputs a1a0 and b1b0 which are multiplied to get a 4-bit output c3c2c1c0.

The IMC based array multiplier had a delay of 307.7 ps and for conventional multiplier was 418.5 ps. The IMC based array multiplier consumes lesser energy which is 3.748mW compared to conventional multiplier which consumes 6.626mW.

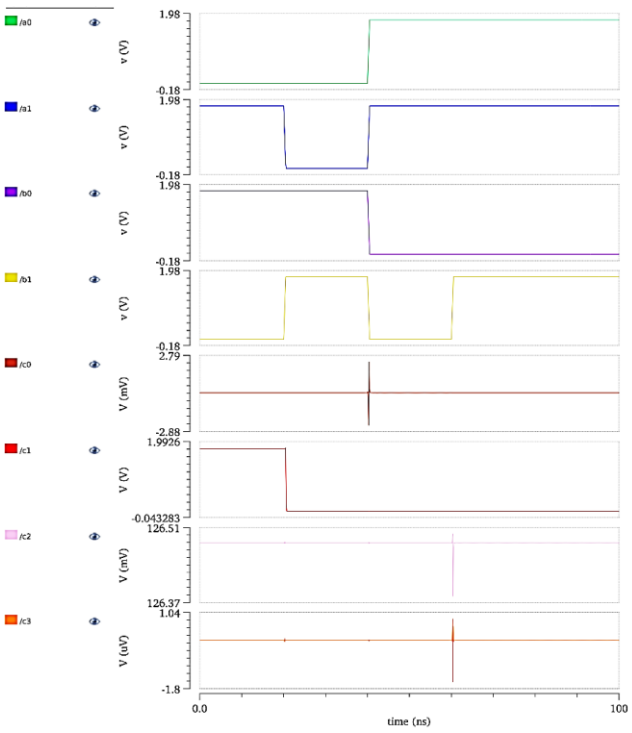

Figure 4. Simulated 2X2 Array Multiplier

5. CONCLUSION

In conclusion, using IMC circuits offers better performance, energy efficiency, scalability, and cost-effectiveness than traditional logic computation, signaling a prospective paradigm shift in computing. There's a notable enhancement in performance, with NOR/NAND/XOR operations improving by 38%, 16%, and 49% respectively. Additionally, energy consumption is lowered by 18% compared to conventional logic computation. The same can be seen in the case of 2X2 array multiplier, where performance was faster by 36% and more efficient by 40% then conventional logic computation.

References:

Agrawal, A., Jaiswal, A., Lee, C., & Roy, K. (2018). X-SRAM: Enabling in-memory Boolean computations in CMOS static random access memories. *IEEE Transactions on Circuits and Systems I: Regular Papers*, *65*(12), 4219-4232. https://doi.org/10.1109/TCSI.2018.2848999

Akbari, O., Kamal, M., Afzali-Kusha, A., & Pedram, M. (2016). RAP-CLA: A reconfigurable approximate carry look-ahead adder. *IEEE Transactions on Circuits and Systems II: Express Briefs*, *65*(8), 1089-1093.https://doi.org/10.1109/TCSII.2016.2633307

Chen, H. C., Li, J. F., Hsu, C. L., & Sun, C. T. (2019, April). Configurable 8T SRAM for enbling in-memory computing. In *2019 2nd International Conference on Communication Engineering and Technology (ICCET)* (pp. 139-142). IEEE. https://doi.org/10.1109/ICCET.2019.8726871

Chen, Y., Lu, L., Kim, B., & Kim, T. T. H. (2020, October). Reconfigurable 2T2R ReRAM with split word-lines for TCAM operation and in-memory computing. In 2020 IEEE International Symposium on Circuits and Systems (ISCAS) (pp. 1-5). IEEE. https://doi.org/10.1109/ISCAS45731.2020.9180665

Dong, Q., Jeloka, S., Saligane, M., Kim, Y., Kawaminami, M., Harada, A., ... & Sylvester, D. (2017). A 4+ 2T SRAM for Searching and In-Memory Computing With 0.3-V $ V_ {\mathrm {DDmin}} $. *IEEE Journal of Solid-State Circuits*, *53*(4), 1006-1015.https://doi.org/10.1109/JSSC.2017.2776309

Ebrahimi-Azandaryani, F., Akbari, O., Kamal, M., Afzali-Kusha, A., & Pedram, M. (2019). Block-based carry speculative approximate adder for energy-efficient applications. *IEEE Transactions on Circuits and Systems II: Express Briefs*, *67*(1), 137-141. https://doi.org/10.1109/TCSII.2019.2901060

Gauchi, R., Kooli, M., Vivet, P., Noel, J. P., Beigné, E., Mitra, S., & Charles, H. P. (2019, October). Memory sizing of a scalable SRAM in-memory computing tile based architecture. In *2019 IFIP/IEEE 27th International Conference on Very Large Scale Integration (VLSI-SoC)* (pp. 166-171). IEEE. https://doi.org/10.1109/VLSI-SoC.2019.8920373

Gupta, A. K., & Acharya, A. (2021, May). Exploration of 9T SRAM cell for in memory computing application. In *2021 Devices for Integrated Circuit (DevIC)* (pp. 461-465). IEEE.https://doi.org/10.1109/DevIC50843.2021.9455838

Gurumurthy, K. S., & Prahalad, M. S. (2010, October). Fast and power efficient 16× 16 Array of Array multiplier using Vedic Multiplication. In *2010 5th International Microsystems Packaging Assembly and Circuits Technology Conference* (pp. 1-4). IEEE. https://doi.org/10.1109/IMPACT.2010.5699463

Ielmini, D., & Wong, H. S. P. (2018). In-memory computing with resistive switching devices. *Nature electronics*, *1*(6), 333-343. https://doi.org/10.1038/s41928-018-0092-2

Jeloka, S., Akesh, N. B., Sylvester, D., & Blaauw, D. (2016). A 28 nm configurable memory (TCAM/BCAM/SRAM) using push-rule 6T bit cell enabling logic-in-memory. *IEEE Journal of Solid-State Circuits*, *51*(4), 1009-1021.https://doi.org/10.1109/JSSC.2016.2515510

Kang, M., Keel, M. S., Shanbhag, N. R., Eilert, S., & Curewitz, K. (2014, May). An energy-efficient VLSI architecture for pattern recognition via deep embedding of computation in SRAM. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 8326-8330). IEEE. https://doi.org/10.1109/ICASSP.2014.6855225

Kulkarni, J. P., Goel, A., Ndai, P., & Roy, K. (2010). A read-disturb-free, differential sensing 1R/1W port, 8T bitcell array. IEEE transactions on very large scale integration (VLSI) systems, 19(9), 1727-1730. https://doi.org/10.1109/TVLSI.2010.2055169

Lee, D. U., Kim, K. W., Kim, K. W., Kim, H., Kim, J. Y., Park, Y. J., ... & Hong, S. (2014, February). 25.2 A 1.2 V 8Gb 8-channel 128GB/s high-bandwidth memory (HBM) stacked DRAM with effective microbump I/O test methods using 29nm process and TSV. In *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)* (pp. 432-433). IEEE. https://doi.org/10.1109/ISSCC.2014.6757501

Lue, H. T., Hu, H. W., Hsu, T. H., Hsu, P. K., Wang, K. C., & Lu, C. Y. (2021, May). Design of computing-in-memory (CIM) with vertical split-gate flash memory for deep neural network (DNN) inference accelerator. In *2021 IEEE International Symposium on Circuits and Systems (ISCAS)* (pp. 1-4). IEEE. https://doi.org/10.1109/ISCAS51556.2021.9401723

Rajput, A. K., & Pattanaik, M. (2020, June). Implementation of Boolean and arithmetic functions with 8T SRAM cell for in-memory computation. In 2020 International Conference for Emerging Technology (INCET) (pp. 1-5). IEEE. https://doi.org/10.1109/INCET49848.2020.9154137

Rathod, R., Ramesh, P., Zele, P. S., & Annapurna, K. Y. (2020, November). Implementation of 32-bit complex floating point multiplier using Vedic multiplier, array multiplier and combined integer and floating point multiplier (CIFM). In *2020 IEEE International Conference for Innovation in Technology (INOCON)* (pp. 1-5). IEEE. https://doi.org/10.1109/INOCON50539.2020.9298363

Reuben, J. (2020). Rediscovering majority logic in the post-CMOS era: A perspective from in-memory computing. *Journal of low power Electronics and Applications*, *10*(3), 28.https://doi.org/10.3390/jlpea10030028

Song, S., & Kim, Y. (2021, October). Novel in-memory computing circuit using muller C-element. In *2021 18th International SoC Design Conference (ISOCC)* (pp. 81-82). IEEE. https://doi.org/10.1109/ISOCC53507.2021.9613964

Song, S., & Kim, Y. (2022). Novel in-memory computing adder using 8+ T SRAM. *Electronics*, *11*(6), 929. https://doi.org/10.3390/electronics11060929

Wu, S., Zheng, X., Gao, Z., & He, X. (2010, April). A 65nm embedded low power SRAM compiler. In *13th IEEE Symposium on Design and Diagnostics of Electronic Circuits and Systems* (pp. 123-124). IEEE. https://doi.org/10.1109/DDECS.2010.5491802

Xu, W., Sapatnekar, S. S., & Hu, J. (2018). A simple yet efficient accuracy-configurable adder design. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, *26*(6), 1112-1125.https://doi.org/10.1109/TVLSI.2018.2803081

Yeswanth, C., & Acharya, A. (2021, May). In-memory Computing based Boolean and logical Circuit Design using 8T SRAM. In *2021 Devices for Integrated Circuit (DevIC)* (pp. 430-434). IEEE. https://doi.org/10.1109/DevIC50843.2021.9455869

Zhang, J., Wang, Z., & Verma, N. (2017). In-memory computation of a machine-learning classifier in a standard 6T SRAM array. *IEEE Journal of Solid-State Circuits*, *52*(4), 915-924.https://doi.org/10.1109/JSSC.2016.2642198

Gururaj A Tapashetti
M.Tech, VLSI Design
School of Electronics Engineering,
Vellore Institute of Technology,
Chennai,
India
gururaj.atapashetti2023@vitstudent.ac.in
ORCID 0009-0008-3964-4811

Umadevi S
Centre for Nanoelectronics and VLSI Design,
School of Electronics Engineering,
Vellore Institute of Technology,
Chennai,
India
Umadevi.s@vit.ac.in
ORCID 0000-0001-7742-9209