SRAM CIM-Based Matrix Multiplication for Edge Intelligence

Dr. S. Senthilmurugan[1], Tamil Selvan R[2], Swamy Nathan G[3], Sibi Raj V V[4], Vikkass Shethra Balan M[5]
[1]*Assistant Professor, Department of ECE, SRM Valliammai Engineering College, Kattankulathur, Chengalpattu, India*
[2,3,4,5]*Department of ECE, SRM Valliammai Engineering College, Kattankulathur, Chengalpattu, India*

***Abstract*— Static Random Access Memory (SRAM) is a key component in modern digital systems due to its high speed, low latency, and reliability. With the rapid advancement of Artificial Intelligence (AI), Internet of Things (IoT), and edge computing systems, there is an increasing demand for efficient memory architectures that can support both storage and computation. Conventional systems suffer from high energy consumption and latency due to frequent data transfer between memory and processor. This paper presents the design and implementation of a 4×4 SRAM memory array using a conventional 6-transistor (6T) SRAM cell in the Cadence Virtuoso environment. The proposed design integrates essential peripheral circuits such as pre-charge circuits, write drivers, sense amplifiers, and address decoders to ensure efficient operation. Detailed simulation results demonstrate successful read, write, and hold operations with stable performance. Furthermore, this work explores the potential of extending SRAM toward Compute-in-Memory (CIM) architectures, enabling vector–matrix multiplication directly within memory. This significantly reduces data movement and improves computational efficiency. The proposed design serves as a foundation for future AI-based and edge computing applications.**

Index Terms— SRAM, CIM, Cadence Virtuoso, 6T Cell, Edge Computing, Matrix Multiplication

I. INTRODUCTION

Memory plays a fundamental role in modern VLSI systems by enabling the storage and retrieval of data required for computation. Among various memory technologies, Static Random Access Memory (SRAM) is widely used due to its fast access speed, low latency, and reliable operation. Unlike dynamic memory, SRAM does not require periodic refreshing, making it suitable for high-performance applications such as cache memory and embedded systems.

With the rapid growth of data-intensive applications, particularly in artificial intelligence, machine learning, and edge computing, the demand for efficient and high-speed memory architectures has increased significantly. In conventional computing systems, data is continuously transferred between the processor and memory, which leads to increased delay and power consumption. This limitation, often referred to as the memory bottleneck, reduces overall system efficiency.

To address this issue, the concept of Compute-in-Memory (CIM) has emerged as a promising approach. CIM enables computational operations to be performed directly within the memory array, thereby minimizing data movement and improving energy efficiency. SRAM-based architectures are especially suitable for implementing CIM due to their compatibility with CMOS technology and ability to support parallel operations.

In this work, a 4×4 SRAM memory array based on the conventional 6T SRAM cell is designed and implemented using the Cadence Virtuoso design environment. The proposed system integrates essential peripheral circuits to support efficient read, write, and hold operations. The performance of the design is verified through simulation, and its potential for extending toward CIM-based matrix multiplication is also explored.

II. LITERATURE SURVEY

Recent developments in SRAM design have focused on improving both performance and energy efficiency, particularly for applications in artificial intelligence and edge computing. The concept of Compute-in-Memory (CIM) has gained significant attention as it enables computations to be carried out directly within memory, thereby reducing data transfer between processor and memory. Studies such as [1] highlight the effectiveness of SRAM-based CIM architectures in achieving faster processing and lower power consumption.

Further research has explored the architectural evolution of SRAM for supporting computational capabilities. Works in [3] and [4] discuss how SRAM arrays can be utilized for parallel operations like vector–matrix multiplication, which are essential in machine learning applications. These studies emphasize the importance of integrating memory and computation to overcome limitations of conventional architectures.

In addition to computational aspects, the stability and reliability of SRAM cells have been widely investigated. The analysis presented in [2] focuses on improving the robustness of 6T SRAM cells under varying operating conditions, while [12] examines low-power design techniques to enhance energy efficiency without compromising performance. These works underline the importance of optimizing transistor-level design for better noise margin and reliable operation.

Fundamental design principles for CMOS and SRAM circuits are well established in standard textbooks such as [5], [6], and [7], which provide detailed insights into circuit behavior, layout design, and simulation techniques. Moreover, industry-standard tools like Cadence Virtuoso [9], [10] are widely used for implementing and verifying SRAM designs through schematic and layout simulations.

Overall, existing literature indicates a growing shift toward memory-centric computing, where SRAM plays a dual role in storage and computation. Building upon these concepts, the present work focuses on designing a 4×4 SRAM array and evaluating its potential for CIM-based applications.

III. SRAM FUNDAMENTALS

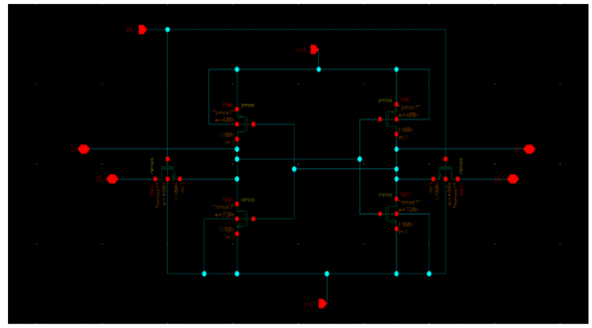

Fig 1: 6T SRAM Cell Structure

Static Random Access Memory (SRAM) is a type of volatile memory that is widely used in digital systems where fast data access is required. Unlike Dynamic Random Access Memory (DRAM), SRAM does not depend on periodic refreshing to maintain stored data. Instead, it uses a feedback-based circuit structure that holds data as long as power is supplied, resulting in faster operation and improved reliability.

The basic storage element of SRAM is the 6-transistor (6T) cell. This cell is composed of two cross-coupled CMOS inverters that form a bistable circuit capable of storing a single bit of information. In addition, two access transistors are connected to the inverters, which control the interaction between the storage nodes and the external bit lines. The internal nodes of the cell store complementary logic values, ensuring stable data representation.

The operation of an SRAM cell can be categorized into three modes: hold, read, and write. In the hold mode, the word line is kept inactive, and the stored data is preserved within the feedback loop of the inverters. During the read operation, the bit lines are pre charged, and activating the word line allows the stored data to influence the bit line voltages. A sense amplifier is then used to detect the small voltage difference and determine the stored value. In the write operation, external data is applied to the bitlines, and the wordline is activated so that the new data overwrites the existing content of the cell.

The stability of an SRAM cell is a critical design factor and is often evaluated using parameters such as Static Noise Margin (SNM). SNM indicates the ability of the cell to withstand noise without losing its stored data. Proper sizing of transistors and careful circuit design are essential to achieve a balance between read stability and write capability.

Overall, SRAM provides a reliable and high-speed solution for memory applications, making it an important component in modern computing systems.

IV. PROPOSED SRAM ARRAY DESIGN

The proposed system consists of a 4×4 SRAM memory array constructed using multiple 6T SRAM cells arranged in a structured matrix form. Each memory cell is capable of storing one bit of data, and the overall array is organized into rows and columns to enable efficient access and control. The rows of the array are driven by wordlines, while the columns are connected through complementary bitlines (BL and $\overline{BL}$), allowing data to be read from and written into the memory.

To ensure proper functionality of the SRAM array, several supporting circuits are incorporated into the design. A precharge circuit is used to initialize the bitlines to a defined voltage level before a read operation, which helps in accurate sensing of stored data. The write driver circuit is responsible for applying input data onto the bitlines during the write process, ensuring that the stored value is updated correctly.

A sense amplifier is included to detect small voltage differences between the bitlines during read operations and convert them into clear digital signals. This improves the speed and accuracy of data retrieval. Additionally, an address decoder is used to select a specific row within the array based on the input address signals, enabling controlled access to individual memory cells. An isolation mechanism is also incorporated to prevent unwanted interactions between different circuit blocks, thereby improving signal integrity.

The working of the proposed system involves three primary operations. During the write operation, input data is applied through the write driver, and the corresponding wordline is activated to store the data in the selected cell. During the read operation, the bitlines are precharged, and activation of the wordline allows the stored data to influence the bitline voltages, which are then sensed by the amplifier. In the hold operation, the wordline remains inactive, and the stored data is retained within the cell without any external interference.

Overall, the proposed SRAM array design ensures reliable operation with efficient integration of peripheral circuits. The structured approach enables scalable memory implementation and provides a foundation for extending the design toward advanced applications such as Compute-in-Memory systems.

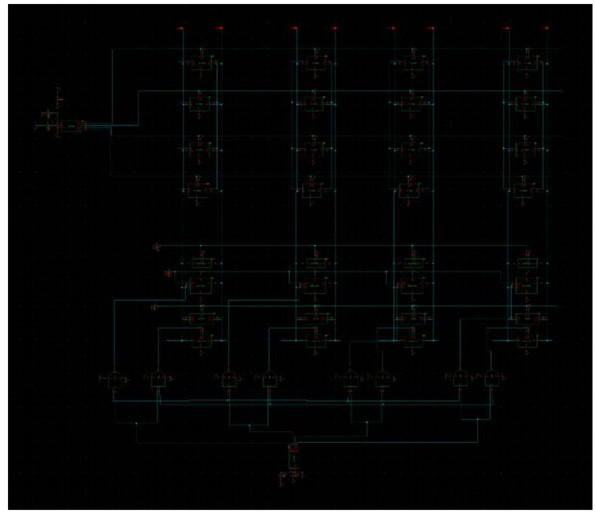

Fig 2: 4×4 SRAM Array Architecture

V. RESULTS AND ANALYSIS

The proposed 4×4 SRAM array was designed and verified using the Cadence Virtuoso simulation environment. The performance of the memory system was evaluated through transient analysis by observing the behavior of key signals such as the wordline, bitlines, and internal storage nodes over time. The results confirm that the designed SRAM cell operates correctly under different modes of operation.

During the write operation, the input data applied through the bitlines successfully alters the state of the internal nodes. The complementary storage nodes switch to the intended logic levels when the wordline is activated, indicating proper write functionality. The transition between logic states occurs smoothly, without any noticeable delay or instability.

In the read operation, the bitlines are initially precharged, and activation of the wordline allows the stored value to influence the bitline voltages. A small voltage difference is generated between the bitlines, which is sufficient for accurate detection by the sense amplifier. The stored data remains undisturbed during this process, confirming non-destructive read behavior.

During the hold condition, the wordline remains inactive, isolating the memory cell from the bitlines.

The cross-coupled inverter structure maintains the stored value without any change, demonstrating reliable data retention. The absence of switching activity in this mode indicates low power consumption and stable operation.

The combined waveform analysis shows multiple cycles of write, read, and hold operations, with consistent and repeatable behavior. The transitions between different modes are smooth, and no glitches or irregularities are observed, indicating robust performance of the SRAM design.

In addition to functional verification, the stability of the SRAM cell was analyzed using Static Noise Margin (SNM). The obtained results indicate that the cell has sufficient noise tolerance and maintains its stored state even under small disturbances. This confirms that the design is stable and suitable for practical applications.

Overall, the simulation results demonstrate that the proposed SRAM array achieves reliable operation, good stability, and efficient performance, making it suitable for high-speed memory applications and potential extension toward advanced computing architectures.

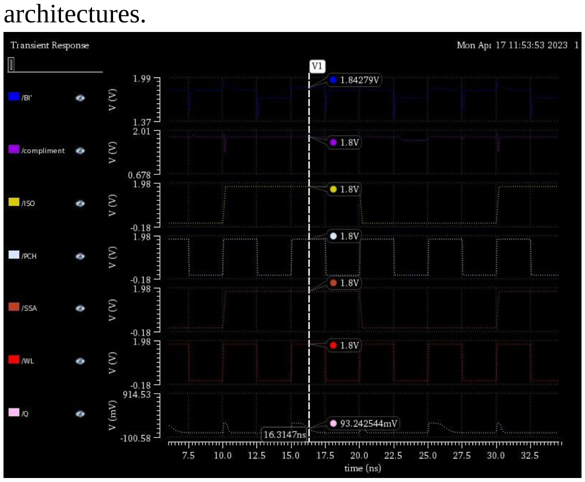

Figure 3: sram_full_wr_

VI. STABILITY ANALYSIS (SNM)

The stability of the SRAM cell is an important factor that determines its ability to retain stored data without being affected by noise or disturbances. One of the widely used methods to evaluate this stability is through the Static Noise Margin (SNM), which represents the maximum noise voltage the cell can tolerate without changing its state.

The SNM is determined by analyzing the voltage transfer characteristics of the two cross-coupled inverters present in the SRAM cell. By plotting these characteristics, a butterfly-shaped curve is obtained. The largest square that can be fitted within this curve indicates the SNM value. This graphical method provides a clear understanding of how well the cell can resist noise during operation.

A higher SNM value implies better stability and improved reliability of the SRAM cell. In the proposed design, the obtained characteristics show a well-defined region, indicating that the cell maintains its logic state without unintended switching. This confirms that the design has good immunity to noise and external disturbances.

The analysis also suggests that the SRAM cell remains stable across different modes of operation, including read, write, and hold conditions. Proper sizing of transistors and balanced design of the inverter pair contribute to achieving this stability.

Overall, the SNM evaluation demonstrates that the designed SRAM cell exhibits strong stability and is capable of reliable operation, making it suitable for use in high-speed and low-power memory applications.

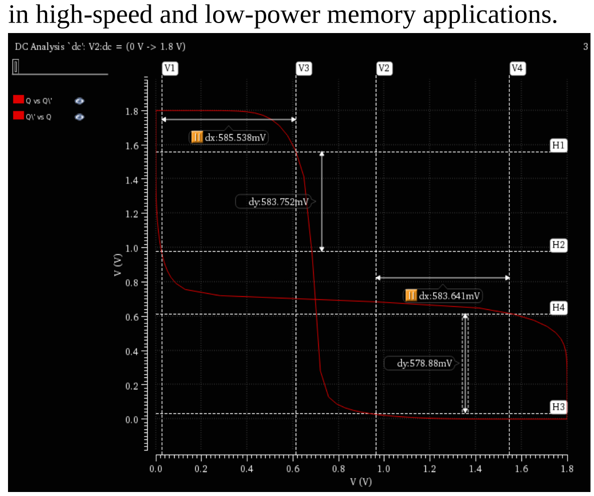

Fig 4: SNM Butterfly Curve

VII. COMPUTE-IN-MEMORY (CIM)

Compute-in-Memory (CIM) is an emerging approach in modern computing that aims to reduce the overhead associated with data transfer between memory and processing units. In conventional architectures, data must be repeatedly moved back and forth between the processor and memory for computation, which increases latency and power consumption. CIM addresses this limitation by enabling certain computational tasks to be performed directly within the memory array itself.

SRAM-based architectures are well suited for implementing CIM due to their fast access speed and compatibility with standard CMOS technology. In such systems, memory cells are not only used for storing data but also participate in basic computational operations. By activating multiple wordlines simultaneously, it becomes possible to process multiple data elements in parallel, improving overall computational efficiency.

One of the key operations that can be implemented using SRAM-based CIM is vector–matrix multiplication. In this method, input data is applied across the wordlines, while the stored data within the memory array represents the matrix values. The resulting bitline signals reflect the combined effect of these inputs, effectively performing multiplication and accumulation operations within the memory itself.

This approach significantly reduces data movement and enhances energy efficiency, making it highly beneficial for applications such as artificial intelligence, machine learning, and edge computing. Additionally, CIM enables parallel processing, which improves speed and throughput in data-intensive tasks. Although the current design focuses primarily on SRAM functionality, it provides a foundation for extending the system toward CIM-based operations. With further modifications, such as enabling multi-wordline activation and incorporating additional sensing mechanisms, the proposed design can be adapted to support advanced in-memory computing applications.

VIII.CONCLUSION

In this work, a 4×4 SRAM memory array based on the conventional 6T SRAM cell was successfully designed and analyzed using the Cadence Virtuoso environment. The proposed system integrates key supporting circuits, including precharge units, write drivers, sense amplifiers, and decoders, which together enable efficient memory operations.

The behavior of the SRAM cell was validated through simulation, where the write operation correctly updated stored data, the read operation retrieved data without disturbing the stored value, and the hold condition ensured stable data retention. The observed waveforms demonstrated smooth transitions between different operating modes, indicating reliable and consistent performance of the design.

The stability of the SRAM cell was further examined using Static Noise Margin (SNM) analysis. The results indicate that the cell maintains its stored state even in the presence of small disturbances, confirming good noise immunity and robust operation. This stability is essential for ensuring dependable performance in practical memory applications.

In addition to its primary function as a storage unit, the proposed design also highlights the potential of SRAM for Compute-in-Memory applications. By enabling operations such as vector–matrix multiplication within the memory array, the design can help reduce data movement and improve overall system efficiency.

Overall, the developed SRAM array demonstrates a balance between performance, stability, and efficiency. The work provides a strong basis for future enhancements and supports the development of memory-centric computing systems for advanced applications.

REFERENCES

[1] W. Gul, M. Shams, and D. Al-Khalili, "FinFET 6T-SRAM All-Digital Compute-in-Memory for Artificial Intelligence Applications: An Overview and Analysis," Micromachines, vol. 14, no. 1535, pp. 1–26, 2023.

[2] S. Agrawal and A. Dubey, "Study and Analysis of Resilient CMOS 6T SRAM Using AI Application," Journal of Emerging Technologies and Innovative Research (JETIR), vol. 11, no. 4, pp. 1–6, Apr. 2024.

[3] Z. Zhang, Z. Guo, Y. Luo, et al., "From Macro to Microarchitecture: Reviews and Trends of SRAM-Based Compute-in-Memory Circuits," Science China Information Sciences, vol. 66, pp. 1–20, 2023.

[4] Z. Lin, Z. Tong, J. Zhang, et al., "A Review on SRAM-Based Computing-in-Memory: Circuits, Functions, and Applications," Journal of Semiconductors, vol. 43, no. 3, pp. 031401-1–031401-18, 2022.

[5] S.-M. Kang and Y. Leblebici, CMOS Digital Integrated Circuits: Analysis and Design, 3rd ed. New York: McGraw-Hill, 2003.

[6] J. M. Rabaey, A. Chandrakasan, and B. Nikolić, Digital Integrated Circuits: A Design Perspective,

2nd ed. Pearson Education, 2003.
[7] J. Baker, CMOS Circuit Design, Layout, and Simulation, 3rd ed. Wiley-IEEE Press, 2010.
[8] R. J. Baker, H. W. Li, and D. E. Boyce, CMOS: Circuit Design, Layout, and Simulation. IEEE Press, 1998.
[9] Cadence Design Systems, Virtuoso Schematic Editor and Analog Design Environment User Guide, 2022.
[10] Cadence Design Systems, Virtuoso Layout Suite User Guide, 2022.
[11] B. Razavi, Design of Analog CMOS Integrated Circuits. New York: McGraw-Hill, 2001.
[12] N. Wakharde and V. Mhaiskar, "Design and Analysis of 6T SRAM Cell for Low Power Applications," International Journal of Engineering Research & Technology, vol. 4, no. 5, pp. 358–362, May 2015.