A High-Performance in-Memory Half Adder and Full Adder Architecture based on 8T and 8+T SRAM Cells

K. Ezhilarasan
Associate Professor Department of Electronics and Communication Engineering SJC Institute of Technology, Chikkaballapura.

Ramya K.V.
Department of Electronics and Communication Engineering, SJC Institute of Technology, Chikkaballapur

Sabha R.
Department of Electronics and Communication Engineering, SJC Institute of Technology, Chikkaballapura

Sri Lakshmi S.R.
Department of Electronics and Communication Engineering, SJC Institute of Technology, Chikkaballapura

ABSTRACT
The increasing volume of data processing has created a growing gap between memory access and computational speed in conventional computing systems. In a conventional von Neumann architecture, operands are repeatedly transferred between memory and processing units, resulting in additional latency and energy consumption. Inmemory computing (IMC) addresses this limitation by performing selected computations within the memory array itself. SRAM is a promising technology for IMC because of its high operating speed, CMOS compatibility, and suitability for dense memory arrays. In particular, 8T SRAM provides an isolated read path that can be exploited for logic computation without directly disturbing the stored data, while 8+T SRAM provides enhanced read capability for more complex operations. Previous research has demonstrated SRAM-based Boolean functions, Half Adders, and Full Adders using computational read paths. However, achieving an appropriate balance between propagation delay, power consumption, transistor count, sensing complexity, and reliable arithmetic operation remains challenging. This work investigates an in-memory Half Adder and Full Adder architecture based on 8T and 8+T SRAM cells. The 8T SRAM structure is employed for Half Adder computation, while the 8+T SRAM structure is used for Full Adder computation. The proposed circuits are intended for transistor-level implementation and simulation in Cadence Virtuoso, followed by evaluation using propagation delay, power consumption, power-delay product (PDP), transistor count, and functional reliability.

Keywords
In-memory computing, SRAM, 8T SRAM, 8+T SRAM, Half Adder, Full Adder, Cadence Virtuoso.

1. INTRODUCTION
The separation of memory and processing in conventional von Neumann systems requires data to move repeatedly between storage and computational units. As modern applications process increasingly large amounts of data, this movement becomes a significant source of latency and energy consumption. In-memory computing attempts to reduce this overhead by utilizing the electrical behavior of memory cells and bitlines to perform computation close to where operands are stored. SRAM has attracted considerable interest for this purpose because of its fast access characteristics and compatibility with standard CMOS implementation. SRAM-based in-memory computing research has demonstrated that logical and arithmetic operations can be integrated into memory arrays rather than being performed exclusively in external processing circuitry [1],[4]. The 8T SRAM cell is particularly attractive for logic-in-memory applications because the additional transistors create a read path that is separated from the storage nodes. This allows the read bitline to be manipulated for computation while maintaining the stored state. Previous studies have used this characteristic to implement Boolean operations including AND, OR, NAND, NOR, XOR, and XNOR [4], [5], [8]. These developments subsequently led to arithmetic structures in which 8T SRAM cells were used for Half Adder computation and 8+T differential SRAM structures were investigated for Full Adder computation [12], [17], [18], [24]. Binary addition forms one of the fundamental operations in digital systems. A Half Adder performs the addition of two binary inputs, A and B, and produces SUM and CARRY outputs. The SUM output is represented by the XOR operation, while the CARRY output is generated by the AND operation. A Full Adder extends the Half Adder by incorporating a third input, the carry-in signal Cin, and generates SUM and COUT outputs. These arithmetic functions can be decomposed into Boolean operations that are suitable for implementation through computational SRAM read paths. Recent studies have demonstrated the direct implementation of arithmetic operations within SRAM structures. Chen et al. demonstrated a 1-bit Full Adder operation using an 8T SRAM-based processing-in-memory architecture [11]. Kumar and Nayak investigated a fast Half Adder using 8T SRAM [12], while Song and Kim proposed an inmemory Full Adder using 8+T SRAM [17]. Jung and Kim further investigated a Full Adder based on 8+T differential SRAM [18]. Desai et al. combined 8T and 8+T SRAM cells for Half Adder and Full Adder computation [3]. These studies establish the feasibility of performing arithmetic operations within SRAM while highlighting the importance of bitline behavior, sensing, power consumption, delay, and transistor-level implementation. Based on these developments, this work focuses specifically on the design and evaluation of an inmemory Half Adder using 8T SRAM and an inmemory Full Adder using 8+T SRAM. The proposed circuits are implemented at transistor level using

Cadence Virtuoso and evaluated in terms of propagation delay, power consumption, power-delay product, transistor count, and functional correctness. The present work is limited single-bit Half Adder and Full Adder architectures, while extension to larger arithmetic structures is considered as future work.

2. LITERATURE REVIEW

[1] R. S. Siddanath et al. (2026) studied custom 6T, 8T, and 10T SRAM layouts in planar CMOS and FinFET technologies, focusing on read structures, sensing, device matching, and parasitic effects. Their analysis of bitline behavior and sensing is relevant to computational SRAM design. These considerations support the implementation of 8T and 8+T SRAM cells in the proposed architecture.

[2] K. Becker et al. (2026) investigated a 6TSRAM-based in-memory computing system for edge intelligence. The work demonstrates how word-line activation, bitline behavior, and sensing can be utilized for computation within SRAM arrays. These concepts provide broader support for the in-memory computing approach adopted in this work.

[3] K. Desai et al. (2025) proposed in-memory Half Adder and Full Adder architectures using 8T and 8+T SRAM cells. The 8T cell generates Half Adder outputs through RBL discharge, while the 8+T cell uses complementary RBL and RBLB signals for Full Adder computation. This work directly supports the SRAM-based arithmetic structures considered in the proposed architecture.

[4] S. S. Kumar and J. Nayak (2025) presented a reconfigurable 8T SRAM architecture supporting Boolean operations and XOR computation. Since XOR is essential for SUM generation, the design provides useful logic-level considerations for SRAM-based arithmetic. Its emphasis on sensing reliability is also relevant to the proposed 8T Half Adder.

[5] H.-J. Shin and S.-H. Jo (2024) developed a low-power 8T SRAM compute-in-memory macro for edge-AI applications. The design exploits isolated read operation and voltage-based computation to improve energy efficiency. These characteristics provide useful considerations for optimizing the computational read path of SRAMbased arithmetic circuits.

[6] K. Lee et al. (2024) presented a P-8T SRAM compute-in-memory macro based on bitline charge sharing. Their work demonstrates the importance of controlled bitline charging and discharging for computational speed and energy efficiency. These factors are relevant to optimizing the bitline operation of the proposed SRAM arithmetic architecture.

[7] L. Wu et al. (2024) investigated an optimized 8T SRAM bit-cell with emphasis on transistor sizing and cell-level characteristics. The study highlights the effects of transistor dimensions on read current, bitline discharge, stability, and capacitance. These parameters are important for improving the delay and power performance of the proposed 8T SRAM Half Adder.

[8] P. Tiwari and P. K. Misra (2024) investigated a fast and energy-efficient logic-in-memory module using SRAM cells. Their work demonstrates that Boolean operations can be performed within the memory structure without external data transfer. This concept supports the implementation of arithmetic operations such as XOR, AND, and OR for SRAM-based adders.

[9] H. Xiao et al. (2024) proposed bidirectional and operand-controllable SRAM-based in-memory computing for Boolean logic and search operations. The architecture provides flexible operand control and computational direction within an SRAM array. Such controllability is useful for applying multiple input signals in SRAM-based arithmetic operations.

[10] T.-D. Nguyen et al. (2024) presented a softerror-resilient and power-efficient SRAM architecture for ADC-less in-memory computing. The work emphasizes computational reliability while reducing energy and peripheral overhead. These characteristics are relevant to maintaining reliable logic levels and correct SUM and CARRY outputs in SRAM-based arithmetic.

[11] L. Chen et al. (2023) developed a full-custom 8T SRAM processing-in-memory architecture for a pixel-parallel vision processor. Their design implements a 1-bit Full Adder using XOR, AND, and OR operations within the SRAM-oriented structure. This demonstrates the feasibility of integrating arithmetic functionality directly into an 8T SRAM platform.

[12] S. S. Kumar and J. Nayak (2023) proposed a fast Half Adder using 8T SRAM for in-memory computing. The design generates SUM and CARRY through the computational read path of the SRAM cell. This work provides a direct foundation for the 8T SRAM Half Adder developed in the present research.

[13] K. Soundrapandiyan et al. (2023) investigated an assist-based reconfigurable sense amplifier for energy-efficient SRAM-based inmemory computing. Their work highlights the importance of sensing circuits in interpreting computational bitline variations. These considerations are relevant to reducing sensing overhead while maintaining reliable arithmetic computation.

[14] Z. Xuan et al. (2023) proposed an ADC-free SRAM-based in-memory computing macro for high-precision MAC operations. The architecture reduces the need for conventional high-overhead data-conversion circuitry. This approach provides useful considerations for achieving low-power and low-latency computation in the proposed SRAMbased arithmetic architecture.

[15] A. K. Rajput et al. (2023) investigated an energy-efficient hybrid SRAM-based in-memory computing macro for edge-AI applications. Their work focuses on reducing energy associated with data movement by performing computation within memory. This principle supports integrating Half Adder and Full Adder operations directly into SRAM structures.

[16] H. Kim (2023) developed a reconfigurable SRAM-based digital near-memory computing macro supporting different computational widths. The study demonstrates the flexibility of memoryintegrated computation for varying computational requirements. This provides useful background for the scalability of the proposed SRAM-based arithmetic architecture.

[17] S. Song and Y. Kim (2022) proposed an inmemory computing adder using 8+T SRAM. Their architecture combines SRAM access and Full Adder computation without transferring operands to separate conventional logic. This work provides a direct foundation for the use of 8+T SRAM in the proposed Full Adder.

[18] J. Jung and Y. Kim (2022) presented a highperformance, low-power Full Adder using 8+T differential SRAM for computation-in-memory. The differential read capability of the cell is utilized to perform Full Adder computation. This study directly supports the suitability of 8+T SRAM for the Full Adder proposed in this research.

[19] J. Chen et al. (2022) proposed a reliable 8T SRAM for high-speed searching and logic-inmemory operations. The architecture employs a dedicated computational access path and sensing circuitry for improved speed and reliability. These features are relevant to designing the computational read path of the proposed 8T Half Adder.

[20] F. Dong et al. (2022) investigated design methodologies and trends in SRAM-based compute-in-memory circuits. Their study provides an overview of circuit techniques used to integrate computation into SRAM. This work helps position the proposed 8T and 8+T SRAM arithmetic architecture within the broader development of inmemory computing.

[21] S. Kim et al. (2022) proposed an SRAMbased in-memory computing macro for memoryaugmented neural networks. The work demonstrates the use of SRAM arrays as both storage and computational resources. This supports the general principle of reducing data movement through memory-integrated computation.

[22] P. K. Bharti et al. (2022) explored computein-memory using conventional 6T SRAM for different workloads. Their work highlights limitations of conventional SRAM structures during computational operation. These limitations motivate the use of additional read transistors in 8T and 8+T SRAM cells for arithmetic computation.

[23] Z. Lin et al. (2022) presented a review of SRAM-based computing-in-memory techniques, computational functions, and applications. The study provides a broad overview of the evolution of SRAM from conventional storage toward computational memory. It establishes the architectural background for the proposed 8T and 8+T SRAM arithmetic structures.

[24] J. Han and Y. Kim (2021) proposed a fast Half Adder using 8T SRAM for computation-inmemory. Their architecture demonstrates the generation of SUM and CARRY within an SRAM computational environment. This work forms an important precedent for the 8T Half Adder investigated in the present research.

[25] C. Yeswanth and A. Acharya (2021) investigated Boolean and logic circuit implementation using 8T SRAM for in-memory computing. The isolated read path enables logic operations such as XOR, which is essential for binary addition. These findings support the Boolean implementation of the proposed SRAMbased Half Adder and Full Adder.

[26] A. K. Gupta and A. Acharya (2021) explored a 9T SRAM cell for in-memory computing using additional transistors for computational functionality. The study demonstrates how dedicated memory paths can enhance computational capability. This provides background for understanding the development of 8T and 8+T computational SRAM structures.

[27] J. Zhang et al. (2021) proposed an 8T SRAM array with configurable word lines for in-memory logic operations. The architecture supports operations including OR, NOR, XNOR, and XOR through controlled SRAM operation. This demonstrates the importance of word-line control for reliable SRAM-based arithmetic computation.

[28] K. Lee et al. (2021) proposed a chargesharing-based 8T SRAM architecture for inmemory computing. Their work demonstrates efficient computation through charge movement within the SRAM array. The effects of bitline capacitance and charge sharing provide useful considerations for optimizing the computational read path.

[29] M. E. Sinangil et al. (2021) presented a 7-nm compute-in-memory SRAM macro supporting multi-bit inputs, weights, and outputs. The work demonstrates the potential for high-throughput and energy-efficient computation using SRAM. Although targeted at neural processing, it provides broader context for efficient computational SRAM architectures.

[30] S. Jain et al. (2021) proposed an SRAMbased signed in-memory computing architecture for broad-purpose computation. Their work demonstrates the applicability of memoryintegrated processing beyond specialized workloads. This broader perspective supports the development of SRAM structures capable of performing arithmetic operations within the memory array.

3. METHODOLOGY

The proposed architecture combines 8T and 8+T SRAM cells to perform arithmetic computation directly within the memory structure. The basic objective is to utilize the 8T SRAM cell for Half Adder computation and the 8+T SRAM cell for Full Adder computation.

This approach exploits the dedicated read paths of the SRAM structures to perform arithmetic operations while maintaining the stored data. The proposed architecture is therefore focused on single-bit arithmetic computation within the memory array.

For the Half Adder, the two input operands are represented by A and B. The outputs are SUM and CARRY, given by

$SUM=A \oplus B \quad CARRY=AB$

Table 1: Truth Table of the Proposed 8T SRAM Half Adder

A	B	SUM	CARRY
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

These Boolean functions are mapped onto the computational read path of the 8T SRAM cell. The isolated read path enables the read bitline to represent the result of the selected logic operation while reducing direct disturbance of the storage nodes. The controlled RBL discharge and sensing approach used in related work provides the basis for the Half Adder implementation [3].

For the Full Adder, the three input operands are A, B, and Cin. The outputs are SUM and COUT, given by

$SUM=A \oplus B \oplus Cin \quad COUT=AB+Cin(A \oplus B)$

Table 2: Truth Table of the Proposed 8+T SRAM Full Adder

A	B	Cin	SUM	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0

1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

The 8+T SRAM structure is used for the Full Adder because its enhanced read circuitry provides complementary read signals through RBL and RBLB. This differential read behavior can be used to detect the computational state and generate the required SUM and COUT outputs. Previous 8+T SRAM research has demonstrated the suitability of this structure for Full Adder computation [3], [17], [18].

3.1 SRAM Cell Design

The first stage of the methodology is the transistorlevel implementation of the 8T and 8+T SRAM cells. The storage and read paths are separately examined to ensure correct write, read, and computational behavior. The 8T structure is analyzed for stable single-ended read operation, while the 8+T structure is analyzed for differential RBL/RBLB operation. These characteristics are important for reliable computational sensing [1], [3], [17].

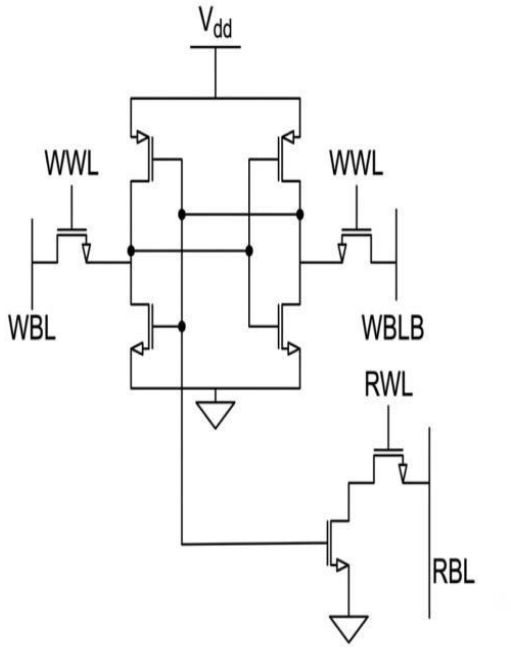


Fig 1: Transistor-Level Schematic of the Proposed 8T SRAM Cell

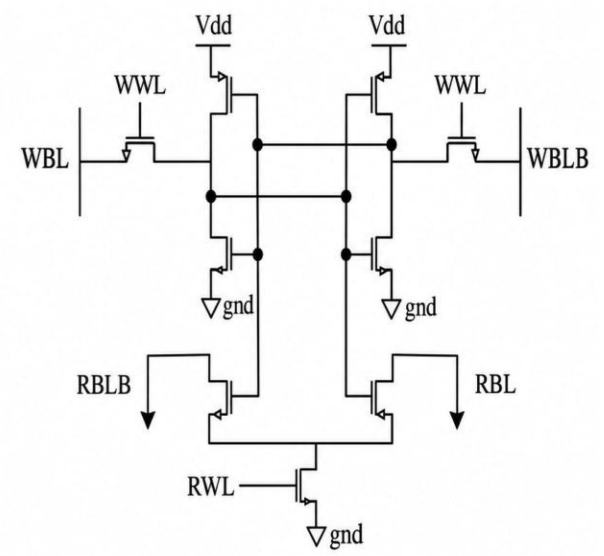


Fig 2: Transistor-Level Schematic of the Proposed 8+T SRAM Cell

3.1.1 Functional Verification

The SRAM cells are first tested independently using transient simulations. Logic '0' and logic '1' are written into the cells and subsequently read through the dedicated read paths. This step verifies that the cells retain the stored information while supporting computational read operations. The read response of the 8T cell and the complementary RBL/RBLB response of the 8+T cell are examined before integrating them into the arithmetic circuits.

3.1.2 Boolean Operation Implementation

After SRAM cell verification, the computational paths are configured to perform the Boolean functions required for binary addition. XOR is required for SUM generation, while AND and OR operations contribute to CARRY generation. Previous studies have demonstrated Boolean computation using 8T SRAM structures, providing the basis for mapping these functions onto the computational read paths [8], [25], [27].

3.2 Half Adder Implementation

The 8T SRAM structure is used to implement the Half Adder. The two input operands are applied to the selected computational rows, and the resulting RBL response is evaluated to generate SUM and CARRY. The circuit is verified for all four possible combinations of A and B. The resulting transient waveforms are compared with the theoretical Half Adder truth table to confirm functional correctness [12], [24].

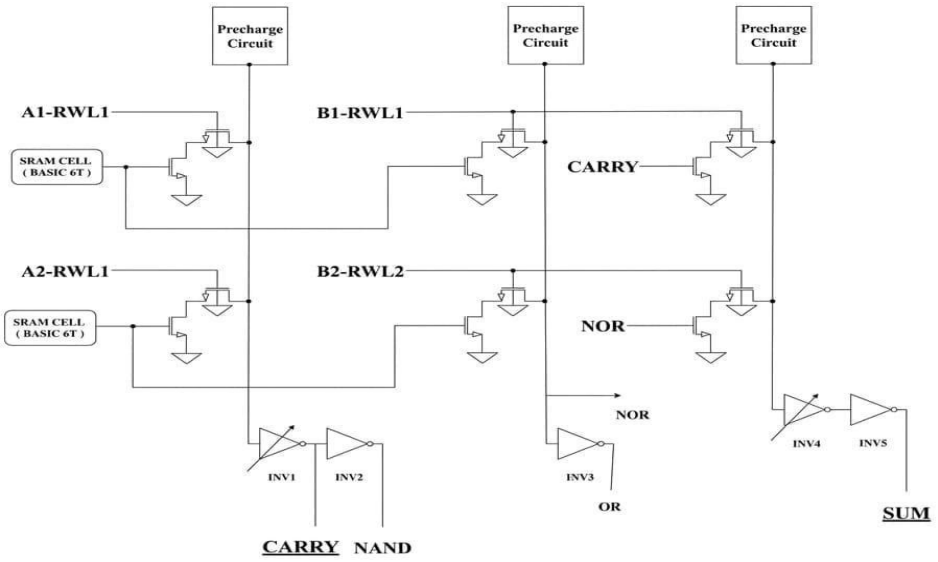


Fig 3: Proposed half -adder circuit using 8T SRAM

3.3 Full Adder Implementation

COUT outputs are evaluated using the computational read path. All eight possible combinations of A, B, and Cin are

The Full Adder is constructed using the 8+T SRAM structure.

The two operand inputs A and B are combined with the carry-in signal Cin, and the resulting SUM and considered during functional verification. The differential read capability of the 8+T SRAM cell is used to improve detection of the computational outputs [17], [18].

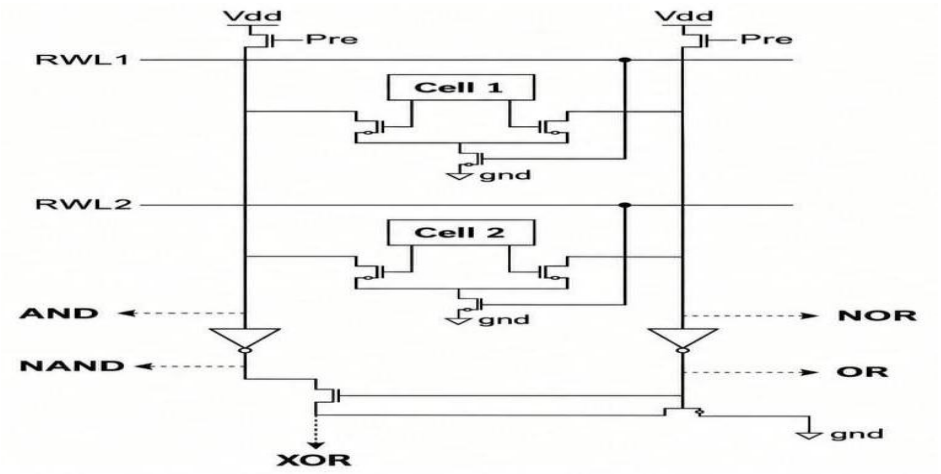


Figure 4: Proposed full- adder circuit using 8+T SRAM

3.4 Cadence Virtuoso Implementation

The proposed SRAM cells and arithmetic circuits are implemented and simulated at transistor level using Cadence Virtuoso. The simulation sequence consists of SRAM cell verification, Boolean operation verification, Half Adder simulation, and Full Adder simulation. Transient analysis is used to observe the internal read-path behavior and the resulting arithmetic outputs. The 2025 8T/8+T study provides a closely related reference because its architecture was also evaluated through transistor-level Cadence Virtuoso simulations [3].

3.4.1 Performance Evaluation

The proposed Half Adder and Full Adder are evaluated using propagation delay, power consumption, power-delay product, and transistor count. These parameters provide a combined assessment of computational speed, energy efficiency, and circuit complexity. Propagation delay is obtained from the transition of the relevant input or control signal to the valid arithmetic output, while power is evaluated from the transient switching behavior of the SRAM cells, bitlines, word lines, and sensing circuitry. PDP is subsequently calculated from power and propagation delay.

3.4.2 Robustness Analysis

Following functional verification, the proposed circuits can be evaluated under different operating conditions. Process, voltage, and temperature variations can be investigated to determine their effect on the generated SUM, CARRY, and COUT outputs. Monte Carlo analysis can additionally be used to examine the effect of transistor-level variations on circuit performance. Such analysis is relevant to SRAM-based circuits because variations in transistor characteristics can influence bitline discharge, sensing margins, and output stability [1], [4].

4. RESULTS AND DISCUSSION

4.1 8T SRAM Cell Simulation Results

The designed 8T SRAM cell is simulated using Cadence Virtuoso to evaluate its write, read, and data-retention characteristics. During the write operation, the write word line is activated and complementary data is applied through WBL and WBLB. The stored information is retained by the cross-coupled storage nodes. During the read operation, the read word line is

activated, allowing the stored data to be sensed through the separate read path. The decoupled read structure reduces direct interaction between the internal storage nodes and the read circuitry.

The transient waveform demonstrates the expected read behavior for the stored logic states. The complementary response of the read signals indicates that the cell can distinguish the stored information correctly. Consistent operation across successive write and read cycles demonstrates stable cell behavior and establishes the required basis for using the 8T SRAM cell in the proposed Half Adder.



Fig 5: Transient Simulation Waveform Showing Write and Read Operation of the Proposed 8T SRAM Cell

4.2 8+T SRAM Cell Simulation Results

The proposed 8+T SRAM cell is simulated using Cadence Virtuoso to verify its write and differential read behavior. During the write phase, the required data is applied through WBL and WBLB while the write word line is activated. After the write operation, the read phase is initiated through the read word line. The additional transistors in the 8+T SRAM cell provide an enhanced read path with complementary RBL and RBLB outputs. The transient response demonstrates complementary behavior at RBL and RBLB. For one stored logic state, RBL moves toward the higher level while RBLB remains relatively low, whereas the opposite response occurs for the other stored state. This differential read behavior confirms the ability of the 8+T SRAM cell to represent the stored information reliably and establishes its suitability for the proposed Full Adder implementation.

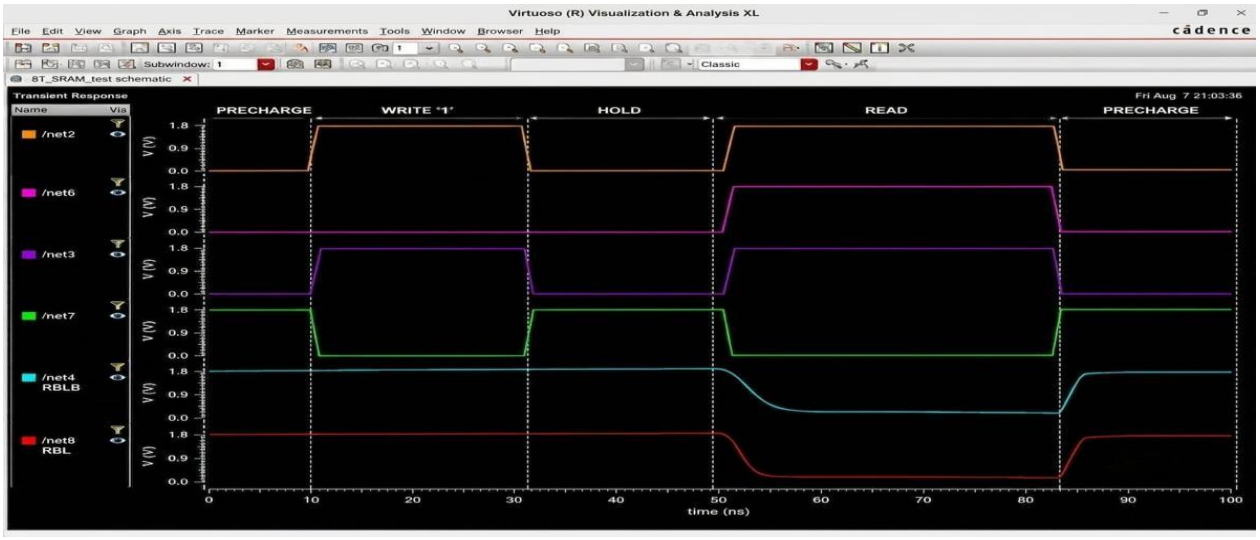


Fig 6: Transient Simulation Waveform Showing Write and Differential Read Operation of the Proposed 8+T SRAM Cell

4.3 Half Adder Results

The proposed 8T SRAM-based Half Adder is evaluated for all four possible combinations of the input operands A and B. The SUM and CARRY outputs obtained from the computational read path are observed using transient simulation. For input combinations 00, 01, 10, and 11, the simulated outputs are compared with the theoretical Half Adder truth table.

The simulation results are used to verify the functional correctness of the proposed Half Adder and to determine its propagation delay, power consumption, and PDP. The obtained values are compared with previously reported 8T SRAM Half Adder architectures, particularly the works reported in [12] and [24].

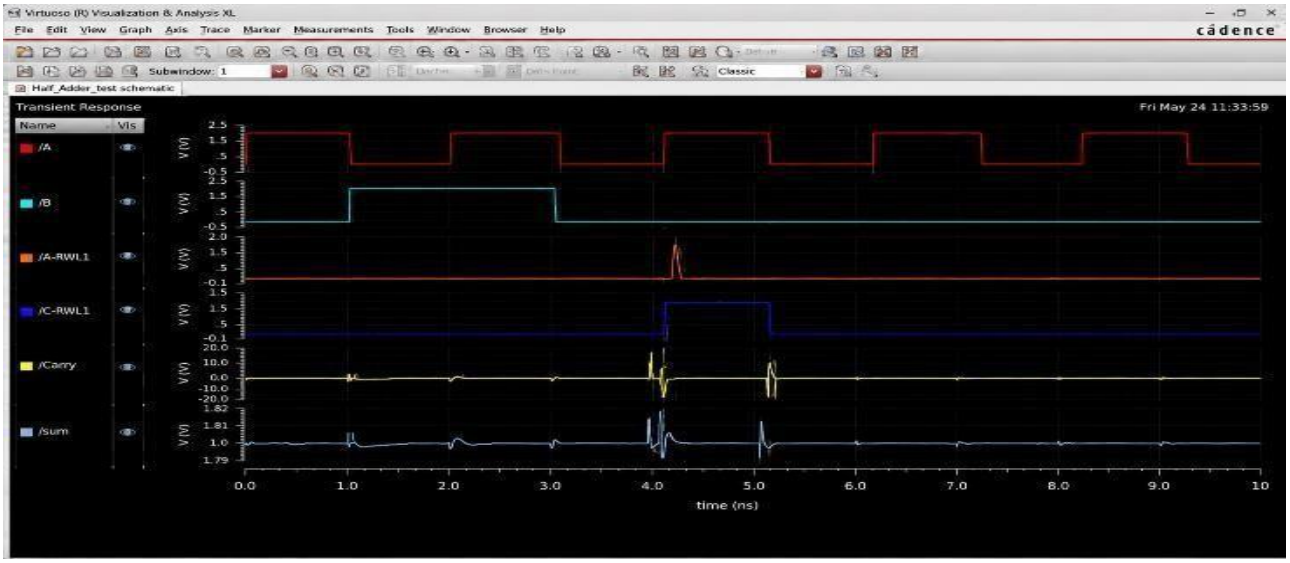


Fig 7: Transient Simulation Waveform of the Proposed 8T SRAM-Based Half Adder

The transient response verifies the expected SUM and CARRY behavior for the four input combinations. The SUM output follows the XOR operation of the two input operands, while the CARRY output follows the AND operation. Thus, the proposed 8T SRAM-based Half Adder performs the required arithmetic operation within the memory structure.

4.4 Full Adder Results

The proposed 8+T SRAM-based Full Adder is evaluated for all eight possible combinations of A, B, and Cin. The SUM and COUT outputs are observed using transient simulation and compared with the theoretical Full Adder truth table. The differential read capability of the 8+T SRAM structure is used to detect the computational outputs.

The simulation results are used to verify functional correctness and to determine the propagation delay, power consumption, and PDP of the Full Adder. The obtained results are compared with the relevant 8+T SRAM Full Adder architectures reported in [17] and [18].

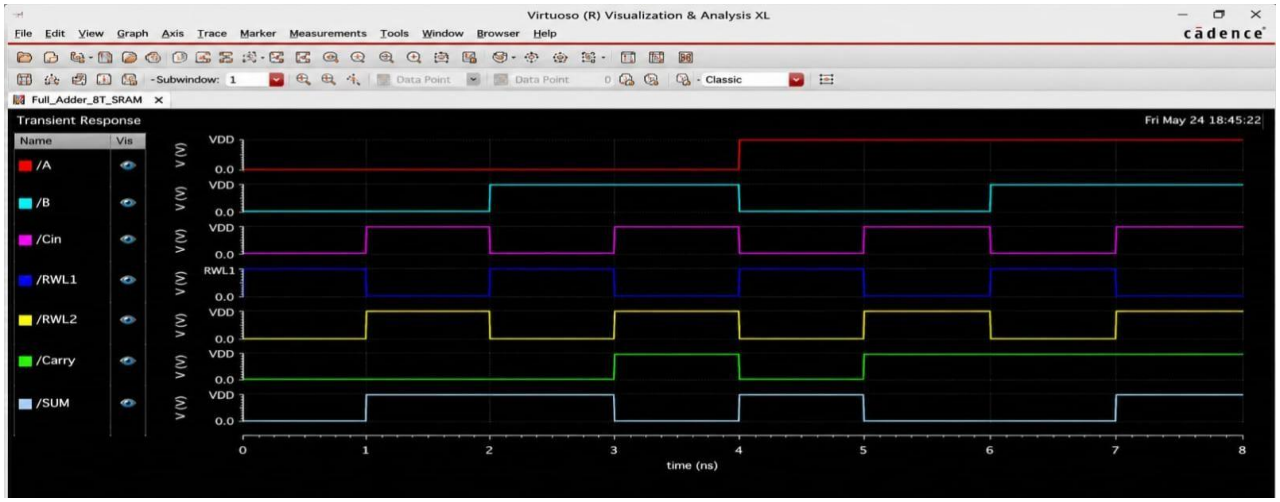


Fig 8: Transient Simulation Waveform of the Proposed 8+T SRAM-Based Full Adder

The simulated SUM and COUT outputs follow the expected Full Adder truth table for all eight input combinations. This confirms that the proposed 8+T SRAM structure can perform the complete addition operation while using the memory cell itself as part of the computational path.

4.5 Performance Comparison

The performance of the proposed in-memory adders was evaluated using the gpdk90 technology library in Cadence Virtuoso. A supply voltage of 1.2 V was used for the proposed designs. The Half Adder was implemented using an 8T SRAM cell, whereas the Full Adder was implemented using an 8+T SRAM cell. The transistor dimensions used in the design were W = 120 nm and L = 100 nm for the NMOS transistors and W = 240 nm and L = 100 nm for the PMOS transistors.

Transient analysis was carried out to verify the SUM and CARRY outputs for the different input combinations. The performance of the proposed designs was evaluated in terms of carry delay, sum delay, power consumption, and power-delay product (PDP). The delay measurement was performed using the 50% voltage transition point of the corresponding signals.

Since the supply voltage is 1.2 V, the 50% voltage level is given by

$$V(50\%) = VDD / 2$$

$$V(50\%) = 1.2 / 2 = 0.6\ V$$

The carry propagation delay is calculated as

$$TCARRY = tCARRY,50\% - tRWL,50\%$$

where tCARRY,50% is the time at which the CARRY output reaches 50% of VDD, and tRWL,50% is the corresponding 50% transition time of the RWL signal.

Similarly, the sum propagation delay is obtained from TSUM = tSUM,50% − tRWL,50% where tSUM,50% represents the 50% transition time of the SUM output.

The average power consumption of the circuit is calculated over the simulation interval as

$$PAVG = (1/T) \int_0^T VDD(t)\, IDD(t)\, dt$$

where T represents the total simulation period, VDD(t) is the supply voltage, and IDD(t) is the current drawn from the supply.

The Power-Delay Product is used to evaluate the trade-off between power consumption and propagation delay. For the Full Adder, the PDP is calculated using the carry delay as

$$PDP = PAVG \times TCARRY$$

For the proposed 8+T Full Adder,

$$PDP = 42.50 \times 0.096$$

$$PDP = 4.08 \text{ fJ}$$

The percentage improvement in the performance parameters is calculated using

$$\text{Improvement (\%)} = ((XREF - XPROP) / XREF) \times 100$$

where XREF represents the value of the reference design and XPROP represents the value of the proposed design.

4.5.1 Half Adder Performance

The proposed Half Adder uses an 8T SRAM cell to perform the arithmetic operation within the memory structure. The performance comparison is made in terms of carry and sum propagation delays.

Table 3: Performance Comparison of the Proposed 8t Half Adder

DESIGN	CARRY DELAY (NS)	SUM DELAY (NS)
Reference Half Adder	0.889	1.540
Proposed 8T Half Adder	0.110	0.620
Improvement (%)	87.63	59.74

For the proposed 8T Half Adder, the carry-delay improvement is calculated as

$$\text{ImprovementCARRY} = ((0.889 - 0.110) / 0.889) \times 100$$

$$= 87.63\%$$

Similarly, the improvement in sum delay is

$$\text{ImprovementSUM} = ((1.540 - 0.620) / 1.540) \times 100$$

$$= 59.74\%$$

The results indicate that the proposed 8T Half Adder provides a substantial reduction in both carry and sum propagation delays compared with the reference Half Adder. The reduction in delay indicates that the in-memory computation approach can perform the required arithmetic operation with reduced signal propagation time.

4.5.2 Full Adder Performance

The proposed Full Adder is implemented using an 8+T SRAM cell. Its performance is evaluated in terms of carry delay, sum delay, power consumption, and PDP.

Table 4: Performance Comparison of the Proposed 8+T Full Adder

Design	Carry Delay (ns)	Sum Delay (ns)	Power (μW)	PDP (fJ)
Reference Full Adder	0.128	0.296	42.73	12.65
Proposed 8+T Full Adder	0.096	0.130	42.50	4.08
Improvement (%)	25.00	56.08	0.54	67.75

The proposed 8+T Full Adder achieves a reduction in both carry and sum propagation delay compared with the reference design. The carry delay is reduced from 0.128 ns to 0.096 ns, corresponding to an improvement of 25.00%.

The improvement in sum delay is calculated as $\text{ImprovementSUM} = ((0.296 - 0.130) / 0.296) \times 100$

$$= 56.08\%$$

The reduction in power consumption is given by $\text{ImprovementPOWER} = ((42.73 - 42.50) / 42.73) \times 100$

$$= 0.54\%$$

The PDP improvement is calculated as

$$\text{ImprovementPDP} = ((12.65 - 4.08) / 12.65) \times 100$$

$$= 67.75\%$$

The proposed design therefore shows a considerable reduction in sum delay and PDP while maintaining nearly the same power consumption as the reference design. The PDP reduction demonstrates the improved energy-delay characteristic of the proposed 8+T Full Adder.

4.6 Overall Discussion

The simulation results demonstrate the functional operation of both proposed in-memory arithmetic circuits. The 8T SRAM cell is used as the basis for the Half Adder, while the 8+T SRAM cell provides the differential read capability required for the Full Adder. The transient simulations verify the expected SUM and CARRY or COUT outputs for the respective input combinations.

For the Half Adder, the proposed 8T design shows 87.63% improvement in carry delay and 59.74% improvement in sum delay compared with the reference design. For the Full Adder, the proposed 8+T design shows 25.00% improvement in carry delay, 56.08% improvement in sum delay, 0.54% reduction in power consumption, and 67.75% improvement in PDP.

Overall, the results indicate that performing arithmetic operations within the SRAM structure can reduce the

dependence on separate conventional logic circuits. The proposed 8T and 8+T SRAM-based architectures therefore demonstrate the potential of in-memory computation for implementing compact and high-performance arithmetic operations.

5. CONCLUSION

This work investigates an in-memory arithmetic architecture based on 8T and 8+T SRAM cells for Half Adder and Full Adder computation. The literature demonstrates that the isolated read path of 8T SRAM can be utilized for Boolean and arithmetic operations while maintaining separation between storage and read operations. Previous research has established 8T SRAM-based Half Adder architectures and 8+T SRAM-based Full Adder architectures, providing the foundation for the present work [12], [17], [18], [24]. The proposed architecture utilizes the 8T SRAM cell for Half Adder computation and the 8+T SRAM cell for Full Adder computation. The computational operations are performed through the SRAM read paths, thereby reducing the need to transfer operands from memory to separate external arithmetic circuitry. The circuits are implemented at transistor level using Cadence Virtuoso and simulations. verified through transient The proposed Half Adder and Full Adder are evaluated in terms of propagation delay, power consumption, power-delay product, transistor count, and functional correctness. These parameters provide a comprehensive basis for assessing the speed, energy efficiency, and circuit complexity of the proposed architecture. Further analysis under process, voltage, and temperature variations can be used to assess circuit robustness. The present work is specifically limited to Half Adder and Full Adder architectures based on 8T and 8+T SRAM cells. Extension of the proposed Full Adder architecture toward larger multi-bit arithmetic structures can be considered as future work.

6. REFERENCES

[1] R. S. Siddanath, S. Mondal, M. Goswami, and K. Kandpal, “A Comprehensive Survey of Custom Layout Techniques for 6T, 8T, and 10T SRAM Bitcells Across Planar CMOS and FinFET Technology Nodes,” IEEE Access, 2026, doi: 10.1109/ACCESS.2026.3664102.

[2] K. Becker, M. Zimmerling, W.-J. Fang, M. L. Fuad, D. K., and H.-G. Herrmann et al., “A 6TSRAM-Based In-Memory Computing System for Sensor-Driven Edge Intelligence,” AI Sensors, 2026.

[3] K. Desai, Y. Agrawal, S. Rajendran, and V. Palaparthy, “Design and optimization of inmemory half-adder and full-adder circuits using 8T and 8+T SRAM cells for enhanced logic operations,” Microsystem Technologies, vol. 31, pp. 3835–3846, 2025.

[4] S. S. Kumar and J. Nayak, “Effective 8T Reconfigurable SRAM for Data Integrity and Versatile In-Memory Computing-Based AI Acceleration,” Electronics, vol. 14, no. 13, p. 2719, 2025.

[5] H.-J. Shin and S.-H. Jo, “Low-Power 8T SRAM Compute-in-Memory Macro for Edge AI Processors,” Applied Sciences, vol. 14, p. 10924, 2024.

[6] K. Lee, J. Kim, and J. Park, “A 28-nm 50.1TOPS/W P-8T SRAM Compute-In-Memory Macro Design with BL Charge-Sharing-Based InSRAM DAC/ADC Operations,” IEEE Journal of Solid-State Circuits, vol. 59, pp. 1926–1937, 2024.

[7] L. Wu, C. Wang, and H. Mo, “Optimization of 8T SRAM BIT-cell design,” in Proc. CSTIC, Shanghai, China, 2024.

[8] P. Tiwari and P. K. Misra, “Fast and energy efficient logic in memory module using SRAM cells,” in Proc. IEEE CONECCT, 2024.

[9] H. Xiao, R. Zhao, Y. Liu, Y. Liu, and J. Chen, “Bi-Directional and Operand-Controllable InMemory Computing for Boolean Logic and Search Operations with Row and Column Directional SRAM,” Micromachines, vol. 15, 2024.

[10] T.-D. Nguyen, M.-S. Le, T.-N. Pham, and I.-J. Chang, “TA-Quatro: Soft Error-Resilient and Power-Efficient SRAM Cell for ADC-Less Binary Weight and Ternary Activation In-Memory Computing,” Electronics, vol. 13, 2024.

[11] L. Chen, C. Shi, J. He, J. Yu, H. Wang, J. Lu, L. Liu, N. Wu, and M. Tian, “An 8-T Processingin-Memory SRAM Cell-Based Pixel-Parallel Array Processor for Vision Chips,” IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 70, no. 11, pp. 4249–4260, 2023.

[12] S. S. Kumar and J. Nayak, “Efficient implementation of fast half adder using 8T SRAM for in-memory computing applications,” in Proc. MoSICom, 2023.

[13] K. Soundrapandiyan, S. K. Vishvakarma, and B. S. Reniwal, “Enabling Energy-Efficient InMemory Computing With Robust Assist-Based Reconfigurable Sense Amplifier in SRAM Array,” IEEE Journal of Emerging and Selected Topics in Circuits and Systems, vol. 13, pp. 445–455, 2023.

[14] Z. Xuan, C. Liu, Y. Zhang, Y. Li, and Y. Kang, “A Brain-Inspired ADC-Free SRAM-Based InMemory Computing Macro with High-Precision MAC for AI Application,” IEEE Transactions on Circuits and Systems II, vol. 70, pp. 1276–1280, 2023.

[15] A. K. Rajput, A. K. Tiwari, and M. Pattanaik, “An energy-efficient hybrid SRAM-based inmemory computing macro for artificial intelligence edge devices,” Circuits, Systems, and Signal Processing, vol. 42, 2023.

[16] H. Kim, “A 1–16b Reconfigurable 80Kb 7T SRAM-Based Digital Near-Memory Computing Macro for Processing Neural Networks,” IEEE Transactions on Circuits and Systems I, vol. 70, 2023.

[17] S. Song and Y. Kim, “Novel In-Memory Computing Adder Using 8+T SRAM,” Electronics, vol. 11, no. 6, p. 929, 2022.

[18] J. Jung and Y. Kim, “A High-Performance, Low-Power 8-Bit Full-Adder Using 8+T Differential SRAM for Computation-in-Memory,” in Proc. IEEE ISOCC, pp. 131–132, 2022.

[19] J. Chen, W. Zhao, Y. Wang, Y. Shu, W. Jiang, and Y. Ha, "A Reliable 8T SRAM for High-Speed Searching and Logic-in-Memory Operations," IEEE Transactions on Very Large Scale Integration Systems, vol. 30, no. 6, pp. 769–780, 2022.

[20] F. Dong, X. Si, and M.-F. Chang, "Design Methodology and Trends of SRAM-Based Compute-in-Memory Circuits," in Proc. IEEE ICSICT, 2022.

[21] S. Kim, W. Lee, S. Kim, S. Park, and D. Jeon, "An In-Memory Computing SRAM Macro for Memory-Augmented Neural Network," IEEE Transactions on Circuits and Systems II, vol. 69, 2022.

[22] P. K. Bharti, S. Jain, K. R. Pillai, S. V. Sayyaparaju, G. S. Kalsi, J. Mekie, and S. Subramoney, "Compute-In-Memory Using 6T SRAM for a Wide Variety of Workloads," in Proc. IEEE ISCAS, 2022.

[23] Z. Lin, Z. Tong, J. Zhang, F. Wang, T. Xu, Y. Zhao, X. Wu, C. Peng, W. Lu, Q. Zhao, and J. Chen, "A Review on SRAM-Based Computing InMemory: Circuits, Functions, and Applications," Journal of Semiconductors, vol. 43, no. 3, 2022.

[24] J. Han and Y. Kim, "A Fast Half Adder Using 8T SRAM for Computation-in-Memory," in Proc. IEEE ICCE-Asia, 2021.

[25] C. Yeswanth and A. Acharya, "In-memory Computing Based Boolean and Logical Circuit Design using 8T SRAM," in Proc. DevIC, pp. 430434, 2021.

[26] A. K. Gupta and A. Acharya, "Exploration of 9T SRAM Cell for In Memory Computing Application," in Proc. DevIC, pp. 461–465, 2021.

[27] J. Zhang, Z. Lin, X. Wu, C. Peng, W. Lu, Q. Zhao, and J. Chen, "An 8T SRAM Array with Configurable Word Lines for In-Memory Computing Operation," Electronics, vol. 10, no. 3, p. 300, 2021.

[28] K. Lee, S. Cheon, J. Jo, W. Choi, and J. Park, "A Charge-Sharing based 8T SRAM In-Memory Computing for Edge DNN Acceleration," in Proc. ACM/IEEE DAC, pp. 739–744, 2021.

[29] M. E. Sinangil et al., "A 7-nm Compute-inMemory SRAM Macro Supporting Multi-Bit Input, Weight and Output and Achieving 351 TOPS/W and 372.4 GOPS," IEEE Journal of Solid-State Circuits, vol. 56, no. 1, pp. 188–198, 2021.

[30] S. Jain, L. Lin, and M. Alioto, "±CIM SRAM for Signed In-Memory Broad-Purpose Computing from DSP to Neural Processing," IEEE Journal of Solid-State Circuits, vol. 56, no. 10, pp. 2981– 2992, 2021.