Design of a Low-Power 8+T SRAM for In-Memory Computing Operations

Chang-Ki Hong and Jeong-Beom Kim

***Abstract*—** SRAM-based in-memory computing is one of the technologies addressing the von Neumann architecture's bottleneck. Designing efficient SRAM bit cells is crucial for the effective implementation of SRAM-based in-memory computing. In this paper, we propose a novel 8+T SRAM design for low-power differential sensing, aimed at enhancing circuit performance and reducing power consumption. The proposed 8+T SRAM bit cells enable parallel execution of logic operations, including ripple-carry adders and content-addressable memory (CAM) functions. Our design reduces power consumption by 21.25% compared to previous 8+T SRAM-based ripple-carry adders, though it introduces a 10.98% increase in propagation delay. Additionally, the power-delay product (PDP) for the adder is lowered by 12.63%. For CAM, our approach decreases energy consumption by 7.06% compared to previous designs. The feasibility of the proposed circuit is validated through SPECTRE simulations with TSMC 65nm CMOS process.

***Keywords*—**In-Memory Computing, SRAM-Based In-Memory Computing, Low-Power Circuit, SRAM, High-Performance Circuit

I. Introduction

The rapid growth of data-intensive applications and the rising demand for efficient processing have posed major challenges to traditional computer architectures. The von Neumann architecture, which separates memory and processing units, faces a performance-limiting issue known as the von Neumann bottleneck. In recent years, in-memory computing has emerged as a promising solution to this problem by enabling computations directly within the memory subsystem. Among various in-memory computing technologies, SRAM-based in-memory computing has attracted significant attention for its potential to deliver low-latency, high-bandwidth data processing [1]-[3]. In standard 6T SRAM-based in-memory computing, the shared read and write bit lines can lead to read failures and data corruption when performing computations on different stored values. To resolve this issue, the 8+T SRAM architecture, with separate read and write terminals, was introduced. This design ensures stable computing operations by utilizing a dedicated read terminal [6], [7]. However, a key drawback of the 8+T SRAM architecture is the increased power consumption due to the additional transistors. To address this issue, this paper proposes a low-power differential sensing 8+T SRAM cell aimed at reducing power consumption. Using the proposed 8+T SRAM, we present designs for a ripple-carry adder and a content-addressable memory (CAM) structure.

This paper is organized as follows. Section II presents the comparison between the previous 8+T SRAM cell structure for in-memory computing and the proposed low-power 8+T SRAM cell structure for in-memory computing. Section III discusses the design of the ripple carry adder and CAM using the proposed 8+T SRAM. Section IV provides a comparison and analysis of simulation results, and Section V concludes the paper.

II. 8+T SRAM Bit Cell

Fig. 1 shows the structure of a previous 8+T SRAM cell. In this architecture, the read access transistor is implemented using NMOS, with the read bit line pre-charged to VDD during computing operations and discharged to GND for logic operations. However, the previous design suffers from increased power consumption due to the higher transistor count compared to a standard 6T SRAM cell.

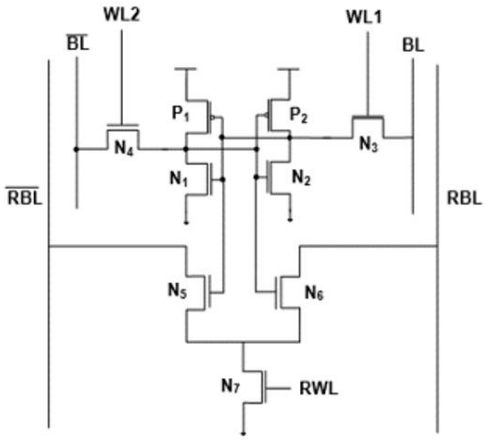


Fig. 1. Previous 8+T SRAM cell.

To solve this power consumption problem, Fig. 2 presents the structure of a low-power 8+T SRAM cell. In this design, the read access transistor is implemented using PMOS instead of NMOS. During computing operations, the read bit line, pre-discharged to GND, is charged to VDD to execute logic functions. The proposed 8+T SRAM structure generates less leakage current in PMOS compared to the NMOS-based read access transistor, which is composed of a forced stack. As a result, the new design reduces power consumption by lowering static power compared to the previous architecture.

Chang-Ki Hong and Jeong-Beom Kim, Dept. of Electronics Eng., Kangwon National University, R. of Korea

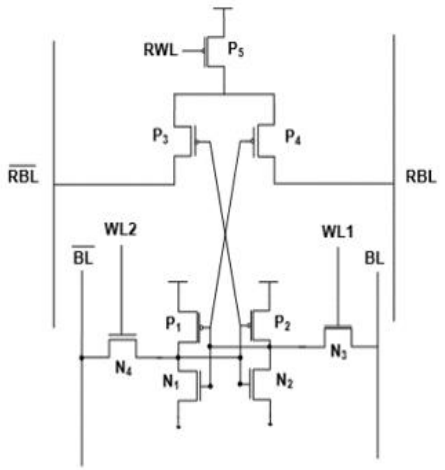


Fig. 2. Proposed 8+T SRAM cell.

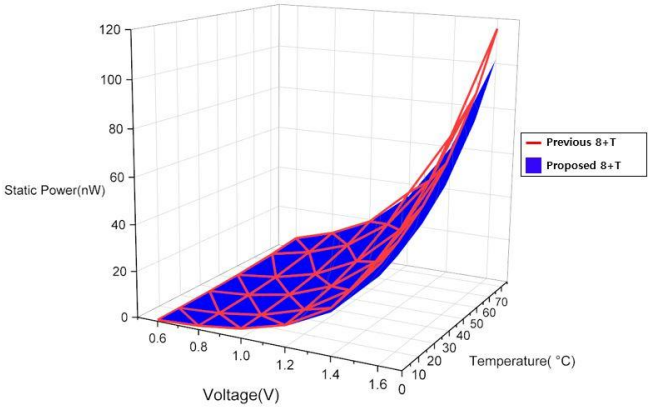


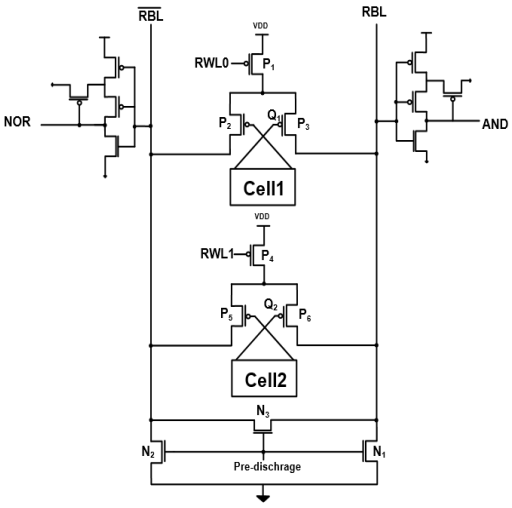
Fig. 3. Comparison of static power between proposed 8+T and previous 8+T SRAM.

Fig. 3 compares the static power of the previous 8+T structure and the proposed 8+T structure across different temperatures and supply voltages. The proposed structure showed a reduction in static power under all conditions, with the most significant decrease being 11.64% at 1.6V and 75ºC.

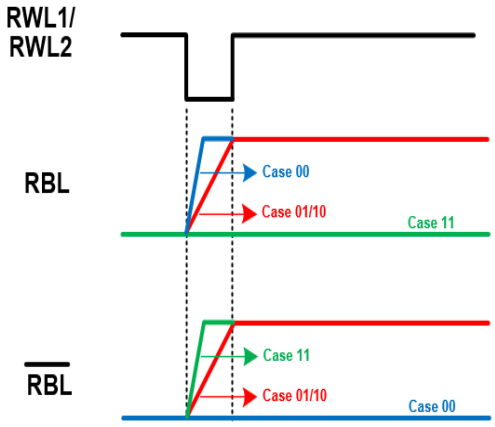
III. PROPOSED IN-MEMORY COMPUTING CIRCUITS

A. *Proposed in-memory computing ripple carry adder*

The In-Memory Computing architecture proposed in Fig. 4(a) consists of a pre-discharge circuit using NMOS, two Half Schmitt trigger inverters, and the low-power 8+T SRAM cell. Unlike the forced stack structure and the previous design, the proposed 8+T SRAM cell employs PMOS read access transistors, which can lead to latency issues due to reduced current driving power during computation [8]. To mitigate the delay caused by this reduced driving capability, a half Schmitt trigger inverter with three PMOS and one NMOS transistor was used. This design helps lower the minimum input switching threshold voltage required for the output to transition from high to low, effectively compensating for the delay. The computing operation of the proposed In-Memory Computing (IMC) circuit begins with the pre-discharge of the Read Bit Line (RBL) and Read Bit Line Bar (RBLB) to GND via the pre-discharge circuit. Subsequently, when a value of '0' is applied to Read Word Line 0 (RWL0) and Read Word Line 1 (RWL1), the RBL and RBLB are charged to VDD through the read access transistor or maintained at GND, depending on the value stored in the SRAM cell.



(a)



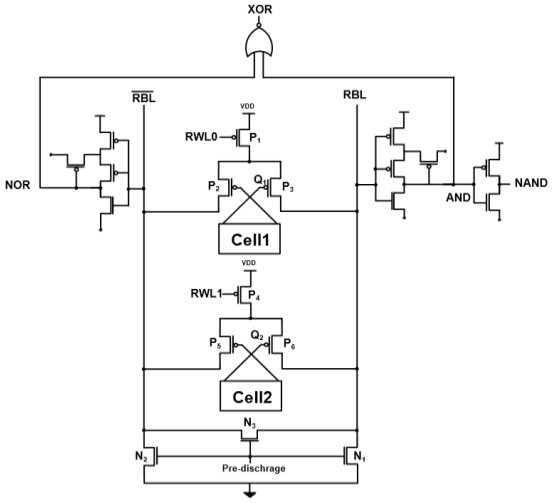
(b)

Fig. 4. (a) Proposed in-memory computing circuit; (b) Timing diagram for reading RBL and RBLB output of Cell 1 and Cell 2.

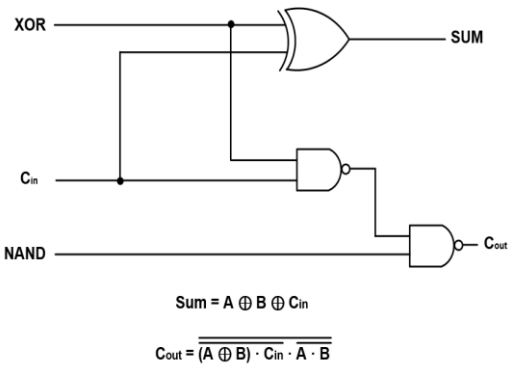
Fig. 4(b) presents a timing diagram illustrating the output changes of RBL and RBLB based on the stored values in Cell 1 and Cell 2. The diagram shows how the RBL and RBLB respond after pre-discharging to the GND state using a pre-discharge circuit. The computing operation begins when a value of 0 is simultaneously applied to RWL1 and RWL2. If both cells store a value of 0, a channel forms in the read access transistors P3 and P6, allowing the RBL to charge to VDD. Conversely, if both cells contain a value of 1, no channel is formed in P3 and P6, keeping the RBL at the GND state. Finally, if Cell 1 and Cell 2 store different values, the RBL charges to VDD through either the P3 or P6 transistor, resulting in a longer charging delay compared to charging through both transistors. The RBLB output inversely follows the changes in the RBL. The truth table in Table 1 demonstrates that the half Schmitt trigger inverter can perform AND operations on RBL and NOR operations on RBLB.

TABLE I: TRUTH TABLE OF PROPOSED IMC CIRCUIT

Q1	Q2	RBL(AND)	RBLB (NOR)
0	0	0	1
0	1	0	0
1	0	0	0
1	1	1	0



(a)



(b)

Fig. 5. Proposed 8+T SRAM cell.

TABLE II: TRUTH TABLE OF PROPOSED IMC CIRCUIT

Q1	Q2	RBL (NAND)	RBLB (NOR)	XNOR
0	0	0	1	1
0	1	1	0	1
1	0	1	0	1
1	1	1	0	0

Fig. 5(a) shows the in-memory computing circuit architecture designed for implementing a full adder, while Fig. 5(b) showcases a specific implementation of the full adder using this architecture. By analyzing the equations for the SUM and Cout operations, we can identify the circuit configuration depicted in Fig. 5(a). In the SUM formula, the XOR operations of operands 'A' and 'B' are realized using the output AND operation of RBL and the output NOR operation of RBLB as inputs to a NOR gate. For the Cout formula, the XOR operations of operands 'A' and 'B' are implemented similarly, while the NAND operations for operands 'A' and 'B' are achieved by adding inverters to the output of the AND operation from RBL. Table 2 outlines the functioning of the in-memory computing circuit for implementing a full adder based on the values stored in Cell 1 and Cell 2. The in-memory computing full adder offers the advantages of reduced power consumption and simplified logic, as it can decrease the number of transistors to previous in-memory computing full adder designs.

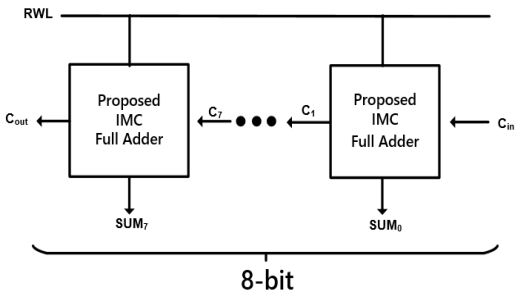


Fig. 6. 8-bit ripple carry adder using the proposed in-memory computing full adder.

Fig. 6 shows the 8-bit ripple carry adder that uses the proposed in-memory computing full adder to efficiently perform addition operations on two 8-bit binary numbers. Each full adder in the ripple carry adder independently computes the SUM and Cout values, which are then propagated to the next stage for the final results. This modular design is well-suited for a range of arithmetic calculations, enabling efficient and scalable addition operations.

B. Proposed In-Memory Computing CAM

The CAM is a specialized type of memory that is essential for applications requiring rapid data retrieval and efficient pattern matching. It is particularly well-suited for high-speed, large-capacity data searches in fields such as big data and artificial intelligence, which have gained significant attention recently.

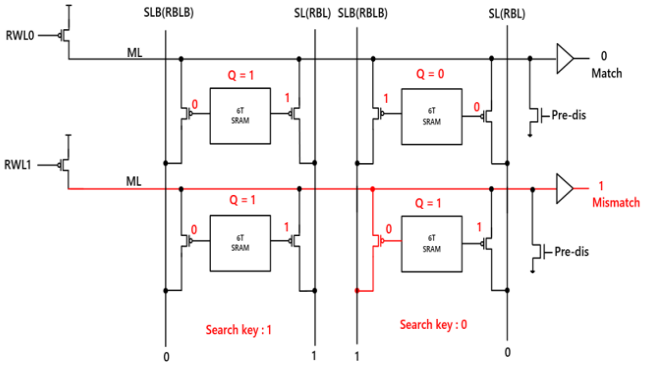


Fig. 7. Binary CAM (BCAM) structure using the proposed 8+T SRAM.

Fig. 7 shows the binary CAM (BCAM) structure utilizing the proposed 8+T SRAM. In this design, the source portion of the read access transistor connected to RBL and RBLB serves as the match line, while RBL and RBLB function as the search lines. The architecture also includes an NMOS pre-discharge transistor for pre-discharging the match line. The BCAM

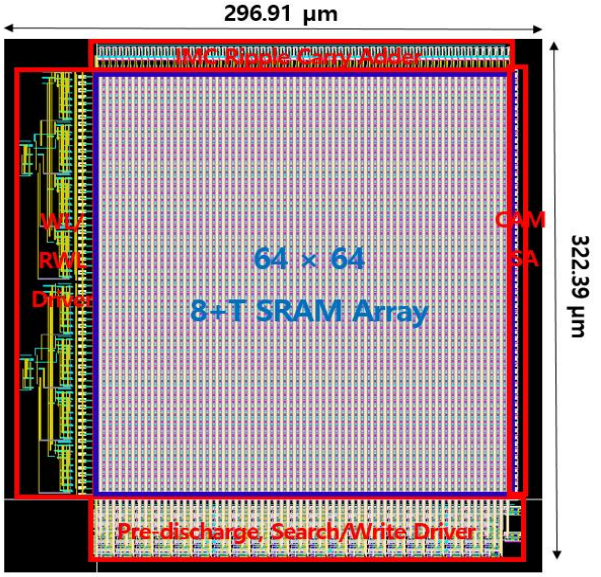
operation begins after the match line is pre-discharged to GND, ensuring that it starts in the GND state. Next, the search driver is enabled to initiate the search operation. If the search value matches the stored data, the read access transistor linked to the match line does not form a channel, preventing the match line from charging to VDD; thus, it remains at GND. If all search values match their corresponding stored values, the match line remains at GND, and the sense amplifier outputs a value of '0' to indicate a successful match. Conversely, if the search value does not match the stored data, the read access transistor connected to the match line forms a channel, causing the match line to charge to VDD. This indicates a mismatch between the search value and the stored data. In this scenario, the match line is charged to VDD, and the sense amplifier outputs a value of '1' to indicate a mismatch. The BCAM search operations described above are summarized in Table 3.

TABLE III: BCAM SEARCH OPERATIONS

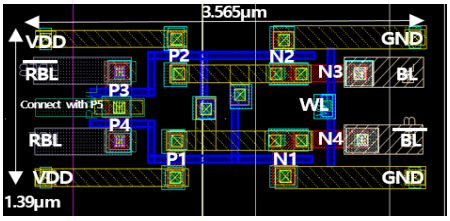
Search key	SL	SLB	Stored data	Charge path
0	0	1	0	-
0	0	1	1	SLB -> ML
1	1	0	0	SL -> ML
1	1	0	1	-

IV. EVALUATION AND SIMULATION RESULTS

Fig. 8(a) presents the design of a 64×64 in-memory computing SRAM array utilizing the proposed 8+T SRAM. The total area of the implemented SRAM array is 0.095 mm² (322.39 µm × 296.91 µm). Fig. 8(b) presents the layout of the proposed 8+T SRAM cell, which has a total area of 4.955 µm² (1.39 µm × 3.565 µm). The source regions of the read access transistors P3 and P4 are connected to the shared read access transistor P5. This design was created using the TSMC 65nm CMOS process, and simulations are performed with SPECTRE.



(a)



(b)

Fig. 8. Proposed 8+T in-memory computing SRAM 64 × 64 array layout view.

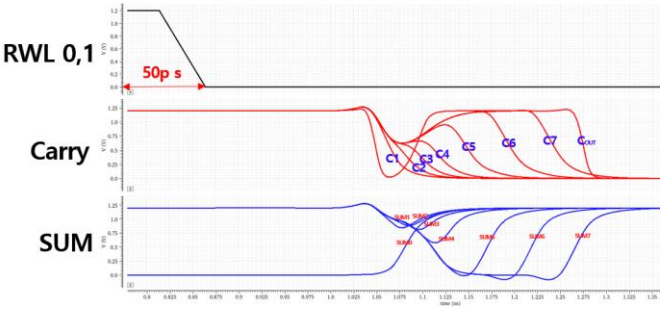


Fig. 9. Simulation results of ripple carry adder.

Fig. 9 shows the simulation results of the ripple carry adder, illustrating the addition operation of the 8-bit ripple carry adder. The addition process begins when a value of '0' is simultaneously applied to RWL0 and RWL1. The operation. 0000 0000 + 1111 1111 = 0 1111 1111 demonstrates the addition process, representing the worst-case scenario in which the charging delay increases the most when different values are stored in the SRAM cell. For the SUM computation, each full adder independently performs the operation, 0 + 1, resulting in SUM0 to SUM7 outputting a value of '1' based on the input from the carry value of the preceding full adder. In the case of the Cout computation, each full adder receives the carry value from the previous stage, leading to sequential outputs of '0' for the carry values of each full adder, ultimately resulting in a Cout value of '0'.

TABLE IV: PERFORMANCE COMPARISON OF 8 BIT IN-MEMORY COMPUTING RIPPLE CARRY ADDER

	Previous work [9]	This work
Propagation delay time[ns]	0.338	0.375
Power consumption[µW]	447.5	352.4
PDP[fJ]	151.2	132.1

Table 4 presents a performance comparison between the ripple carry adder implemented with the previous 8+T SRAM and the ripple carry adder using the 8+T SRAM proposed in this paper. The circuit described in this study demonstrated a 10.98% increase in propagation latency, while achieving a 21.25% reduction in power consumption and a 12.63% reduction in power-delay product (PDP). These results were obtained under typical-typical corner (TT) conditions at an operating frequency of 700 MHz, a supply voltage of 1.2 V, and an operating temperature of 25℃.

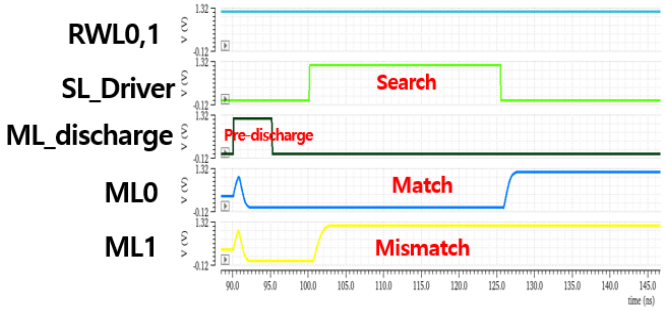


Fig. 10. Simulation results of proposed BCAM.

Fig. 10 illustrates the simulation results of the proposed BCAM. Initially, during BCAM operation, RWL0 and RWL1 are set to '1', preventing the formation of a channel. This ensures that the read access transistors connected to RWL0 and RWL1 do not create an additional charging path in the match line. After this, the match line is pre-discharged to GND, and the search operation begins. For ML0, during the search operation, the GND state is maintained because the search data matches the data stored in the cell. In contrast, for ML1, if the search data does not match the stored data, the match line is charged to VDD, indicating a mismatch between the search data and the stored data.

TABLE V: COMPARISON TABLE

	This work	TVLSI’22 [10]	TVLSI’20 [11]	JSSC’18 [12]	JSSC’21 [13]
Supported IMC	Boolean Logic	Boolean Logic	Boolean Logic	Boolean Logic	Boolean Logic
Functions	Arithmetic, CAM	CAM	CAM	CAM	CAM
Technology	65-nm TSMC	28-nm	65-nm	55-nm DDC	28-nm
Cell type	Proposed 8+T	Standard 8T	Standard 8T	4+2T	10T
Array size	64 × 64 (4 Kb)	128 × 128 (16 Kb)	128 × 128 (16 Kb)	128 × 128 (16 Kb)	64 × 64 (4 Kb)
Supply voltage	0.6 – 1.2 V	0.5 – 0.9 V	0.6–1.2V	-	0.5 – 0.9 V
BCAM energy(fJ/bit)	0.32(0.6V) 0.79(1.2V)	0.79 (1.2V)	0.85 (1.2 V)	0.45 (0.8V)	1.02 (0.9V

Table 5 compares he energy performance of the BCAM implemented with the proposed 8+T SRAM structure against the previous structure. The results indicate that the BCAM using the proposed 8+T SRAM achieves a 7.06% reduction in energy consumption at a supply voltage of 1.2V compared to the BCAM structure utilizing 8T SRAM. Furthermore, the energy efficiency is also enhanced when compared to BCAM structures employing 10T SRAM. The proposed 8+T SRAM allows the BCAM to operate at low power, significantly improving energy efficiency and making it suitable for computing systems that prioritize energy conservation.

V. CONCLUSIONS

The proposed 8+T SRAM architecture is designed to operate at low power, resulting in reduced power consumption for previous in-memory computing circuits based on 8+T SRAM. Using this structure, we successfully implemented both the Ripple Carry Adder and the BCAM. While the Ripple Carry Adder experienced a 10.98% increase in latency, it achieved a significant reduction in power consumption of 21.25% and a decrease in power-delay product (PDP) of 12.63%. Additionally, the BCAM demonstrated a notable 7.06% reduction in energy consumption compared to earlier structures; however, it did not completely address the issue of propagation delay. To effectively manage power consumption and minimize propagation delays, future research should focus on advancing in-memory computing technologies.

REFERENCES

[1] J. Chen, W. Zhao, Y. Wang and Y. Ha, "Analysis and Optimization Strategies Toward Reliable and High-Speed 6T Compute SRAM," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 68, no. 4, pp. 1520-1531, April 2021. https://doi.org/10.1109/TCSI.2021.3054972

[2] M. Nabavi and M. Sachdev, "A 290-mV, 3.34-MHz, 6T SRAM With pMOS Access Transistors and Boosted Wordline in 65-nm CMOS Technology," *IEEE Journal of Solid-State Circuits*, vol. 53, no. 2, pp. 656-667, Feb. 2018. https://doi.org/10.1109/JSSC.2017.2747151

[3] K. Lee, J. Jeong, S. Cheon, W. Choi and J. Park, "Bit Parallel 6T SRAM In-memory Computing with Reconfigurable Bit-Precision," in *Proc. 2020 57th ACM/IEEE Design Automation Conference (DAC)*, 2020, pp. 1-6. https://doi.org/10.1109/DAC18072.2020.9218567

[4] C. Yu, T. Yoo, K. T. C. Chai, T. T. -H. Kim and B. Kim, "A 65-nm 8T SRAM Compute-in-Memory Macro With Column ADCs for Processing Neural Networks," *IEEE Journal of Solid-State Circuits*, vol. 57, no. 11, pp. 3466-3476 Nov. 2022. https://doi.org/10.1109/JSSC.2022.3162602

[5] H. Kim, T. Yoo, T. T. -H. Kim and B. Kim, "Colonnade: A Reconfigurable SRAM-Based Digital Bit-Serial Compute-In-Memory Macro for Processing Neural Networks," *IEEE Journal of Solid-State Circuits*, vol. 56, no. 7, pp. 2221-2233, July 2021. https://doi.org/10.1109/JSSC.2021.3061508

[6] J. P. Kulkarni, A. Goel, P. Ndai and K. Roy, "A Read-Disturb-Free, Differential Sensing 1R/1W Port, 8T Bitcell Array," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 19, no. 9, pp. 1727-1730, Sept. 2011. https://doi.org/10.1109/TVLSI.2010.2055169

[7] A. Agrawal, A. Jaiswal, C. Lee and K. Roy, "X-SRAM: Enabling In-Memory Boolean Computations in CMOS Static Random Access Memories," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 65, no. 12, pp. 4219-4232, Dec. 2018. https://doi.org/10.1109/TCSI.2018.2848999

[8] CAI, Jiangzheng, et al., “A PMOS read-port 8T SRAM cell with optimized leakage power and enhanced performance,” *IEICE Electronics Express*, vol. 14, no. 3 pp. 20161188-20161188, 2017. https://doi.org/10.1587/elex.14.20161188

[9] S. B. Song and Y. M. Kim. “Novel in-memory computing adder using 8+ T SRAM,” *Electronics*, vol. 11, no. 6, March 2022. https://doi.org/10.3390/electronics11060929

[10] J. Chen, W. Zhao, Y. Wang, Y. Shu, W. Jiang and Y. Ha, "A Reliable 8T SRAM for High-Speed Searching and Log-ic-in-Memory Operations," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 30, no. 6, pp. 769-780, June 2022. https://doi.org/10.1109/TVLSI.2022.3164756

[11] Z. Lin et al., "In-Memory Computing with Double Word Lines and Three Read Ports for Four Operands," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 28, no. 5, pp. 1316-1320, May 2020. https://doi.org/10.1109/TVLSI.2020.2976099

[12] Q. Dong et al., "A 4 + 2T SRAM for Searching and In-Memory Computing With 0.3-V VDDmin," *IEEE Journal of Solid-State Circuits*, vol. 53, no. 4, pp. 1006-1015, April 2018. https://doi.org/10.1109/JSSC.2017.2776309

[13] Z. Lin et al., "Two-Direction In-Memory Computing Based on 10T SRAM With Horizontal and Vertical Decoupled Read Ports," *IEEE Journal of Solid-State Circuits*, vol. 56, no. 9, pp. 2832-2844, Sept. 2021. https://doi.org/10.1109/JSSC.2021.3061260

Chang-Gi Hong received B.S. and M.S. degrees in Electronics Engineering from Kangwon National University, Chuncheon, Korea in 2022 and 2024, respectively. In 2024, he joined NETSOL Co. Ltd., Hwasung, Korea, where he is currently a research engineer. His research interests include processing-in-memory circuit and computing-in memory circuit design.

Jeong-Beom Kim received B.S. and M.S. degrees in Electronics Engineering from Inha University, Incheon, Korea in 1985 and 1987, respectively, and Ph.D. degree in Electrical and Electronics Engineering from Pohang University of Science and Technology (POSTECH), Pohang, Korea in 1997. From 1987 to 1992, he worked at R&D center, Goldstar Semiconductor, Seoul, Korea. From 1994 to 1998, he worked at System IC R&D center, Hyundai Electronics, Icheon, Korea. From 1998 to 1999, he was with the School of Electrical and Electronics Engineering, Chungbuk National University, Cheongju, Korea. In 1999, he joined the Faculty of Electrical and Electronics Engineering, Kangwon National University, Chuncheon, Korea, where he is currently a professor. His research interests include processing-in-memory circuit and high-speed CMOS interface circuit design. He received the Best Paper Award from IKEEE in 2005. Prof. Kim is a member of IEEE and KIECS.