

Received 8 July 2025, accepted 21 July 2025, date of publication 24 July 2025, date of current version 1 August 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3592352

RESEARCH ARTICLE

Dual-Mode 2T1C DRAM Process-In-Memory Architecture for Boolean and MAC Operations

YERIM AN^{ID}, HONGGU KIM^{ID}, DONGJUN SON^{ID}, HAKRAE YU, AND YONG SHIM^{ID}

Department of Intelligent Semiconductor Engineering, Chung-Ang University, Dongjak-gu, Seoul 06974, Republic of Korea

Corresponding author: Yong Shim (yongshim@cau.ac.kr)

This work was supported in part by the Chung-Ang University Research Grants in 2024, and in part by the National Research Council of Science and Technology (NST) Grant funded by Korea Government [Ministry of Science and Information and Communication Technology (MSIT)] under Grant CAP22031-000.

ABSTRACT With the increasing demand for intelligent memory, the conventional memory system is more and more equipped with computational logic to support simple arithmetic and Boolean operations required by many real-world applications. Such a trend, called ‘Process-In-Memory’ architecture, tries to utilize most of the memory candidates ranging from the conventional charge-based memory such as SRAM, DRAM, and Flash to the emerging memory devices such as RRAM, PRAM and MRAM. From the application perspective, many researchers are putting efforts to develop efficient memory peripherals with CMOS circuits to support one of two operations: 1) Boolean function, and 2) simple arithmetic operations such as multiplication and accumulation (addition), which is typically referred to as MAC operation. However, there are not many previous works that support both operations in a single system. In this article, we propose a 2T1C DRAM-based PIM architecture that supports dual-mode operation with minimal additional hardware. The proposed architecture has been fabricated using a commercial 65nm CMOS technology and successfully proves that the target operations are performed with a reasonably good accuracy.

INDEX TERMS Processing-in-memory (PIM), DRAM, DRAM compute-in-memory, DNN accelerator, dual mode operation, Boolean function, multiplication-and-accumulation.

I. INTRODUCTION

Recently, there has been an increased interest in emerging applications such as artificial intelligence and machine learning. These areas of research naturally require massive amounts of data to train the network for better performance, and this trend becomes more pronounced as the application becomes sophisticated and performs more complex tasks. This increased data demand results in a higher volume of data being shuttled between the CPU and memory. However, the traditional von Neumann computer architecture maintains a separation between memory and CPU. When this architecture is used to process large amounts of data, a ‘memory wall’ bottleneck occurs due to the performance gap between the CPU and memory. To resolve this problem, the Process-In-Memory (PIM) Architecture is being continuously researched to handle the increased data transfer between CPU and memory. Unlike the von Neumann architecture, the PIM architecture integrates additional computation units inside the

memory chip to execute simple arithmetic operations and/or Boolean functions, which are frequently utilized in AI-driven data processing. This enables efficient data processing and thus performance improvement by minimizing data movement between CPU and memory.

PIM architecture can be divided into several groups depending on 1) the type of memory used and 2) the targeted applications and the required operations. In particular, the detailed operation of a specific PIM architecture is largely affected by the type of the memory device incorporated. Among the candidate of memory devices, SRAM-based PIM is the most representative architecture. Typically, the SRAM cell is used as cache memory in today’s processors due to its process compatibility with logic devices and ability to transfer data at high speed. The SRAM-based PIM architecture can also take advantage of these characteristics. In terms of operation, the SRAM-based PIM systems can be categorized into analog and digital architecture. In the case of SRAM-based digital PIM, much research has been done to perform Boolean logic functions such as AND, NAND, NOR and XOR. Especially, XNOR gate is used for more

The associate editor coordinating the review of this manuscript and approving it for publication was Mario Donato Marino^{ID}.

complex tasks, such as compute bit-wise products [1], [2], [3], [4]. Meanwhile, various analog circuit techniques are used to perform arithmetic operations for the analog SRAM PIM system [5], [6], [7]. Nevertheless, an SRAM cell has a larger cell size (typically more than 6 transistors for a single cell) than other memory types, which hinders efficient area utilization and power consumption.

Among the candidate memory devices for PIM architectures, DRAM cells present an attractive alternative due to their more compact layout footprint compared to SRAM cells. Although DRAM is generally slower than SRAM, recent advancements in device engineering have focused on significantly reducing cell leakage current, thereby minimizing the need for a large storage capacitor. These technological developments make DRAM cells increasingly suitable for compute-centric memory architectures. However, the conventional embedded DRAM cell structure, such as the 1T1C cell, does not feature a separate read and write path. As a result, charge stored in the capacitor can be lost during read operations, which poses challenges in terms of data retention and computational accuracy—particularly in PIM architectures where frequent in-memory computation is required. To address this issue, this work adopts a 2T1C DRAM cell structure with separated read and write paths. This approach enhances both data retention characteristics and memory utilization efficiency, making it more suitable for reliable PIM operations.

Similar to SRAM-based PIM, DRAM-based PIM architectures can also be categorized into digital-based and analog-based systems. Several studies have explored digital PIM implementations using DRAM [8], [9], [10], [11], while others have proposed analog DRAM PIM architectures [12], [13]. Digital PIM architectures offer advantages such as high precision and robustness against noise; however, they suffer from low energy efficiency due to frequent memory access and limited parallelism. On the other hand, analog PIM architectures provide high energy efficiency and enable large-scale parallel computation, but they face challenges such as low precision and vulnerability to noise caused by device mismatch, process variation, and charge injection. Moreover, most of these previous studies have primarily focused on a single type of operation—either Boolean logic functions (digital) or multiplication-and-accumulation (MAC) operations (analog)—resulting in inherent limitations in terms of system flexibility and overall efficiency.

Recently, a number of hybrid PIM architectures have been proposed that combine analog and digital computations, primarily targeting energy-efficient AI edge devices [14], [15], [16], [17], [18], [19], [20]. These architectures typically aim to enhance energy efficiency or compensate for the limited precision of analog computation by integrating digital computation logic. A representative example of such a hybrid approach is the DYAMOND architecture [20], which employs a 1T1C DRAM-based design. It performs analog current accumulation along the bit line and utilizes digital circuits, such as bit-column addition, in the backend to

generate normalized outputs. This strategy enables improved accuracy and energy efficiency for MAC operations through a hybrid analog–digital scheme. In contrast, the architecture proposed in this work adopts a fundamentally different approach. Rather than using digital logic merely as an auxiliary component to enhance analog accuracy, our design integrates two inherently distinct operations—Boolean logic and MAC computation—as independent functional units within a single chip-level PIM system.

As previously discussed, this work proposes a prototype architecture that supports dual-mode operation, integrating both Boolean and MAC operations within a single system. This dual-mode structure is particularly well-suited for hierarchical processing schemes, where the Boolean logic block can act as a low-power event filter and activate the MAC block only when necessary. For example, in smart surveillance systems, the Boolean logic can continuously monitor simple events such as frame differences or sensor triggers. When such an event is detected, the CNN-based MAC unit is selectively activated to perform detailed image recognition. Similarly, in industrial IoT or wearable health monitoring devices, real-time Boolean evaluations can serve as conditions to trigger complex neural inference only when needed. This selective activation strategy significantly reduces energy consumption while maintaining real-time responsiveness, making the proposed architecture highly compatible with the stringent requirements of next-generation edge AI systems.

Based on this concept, we implemented a prototype dual-mode PIM chip using a 2T1C DRAM cell array. The chip successfully performs both Boolean and MAC operations within a unified system, demonstrating minimal hardware overhead and validating the feasibility of future AI edge devices.

This paper presents the following key contributions:

- **Dual-Mode Support in DRAM-based PIM:** We propose a novel 2T1C DRAM-based PIM architecture that supports both analog MAC and digital Boolean operations within a unified system, enabling flexible computation modes without requiring additional memory structures.
- **Prototyped and Measured Dual-Mode Operation:** We fabricated a prototype chip implementing the proposed architecture and validated its dual-mode functionality on real silicon, demonstrating accurate Boolean logic and high-linearity MAC performance with minimal hardware overhead.
- **Edge-AI-Oriented Processing Efficiency:** The dual-mode operation enables energy-efficient, event-driven hierarchical processing schemes suitable for edge AI systems, allowing low-power Boolean pre-processing and selective MAC activation, leading to significant reductions in unnecessary computation and power consumption.

The remaining sections are summarized as follows. In Section II, we describe the basic operation of the 2T1C DRAM cell and the basic concept of dual-mode operation

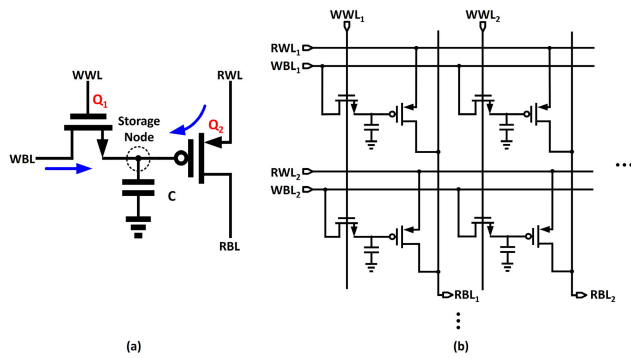


FIGURE 1. Schematic of (a) 2T1C eDRAM cell, (b) 2T1C eDRAM array.

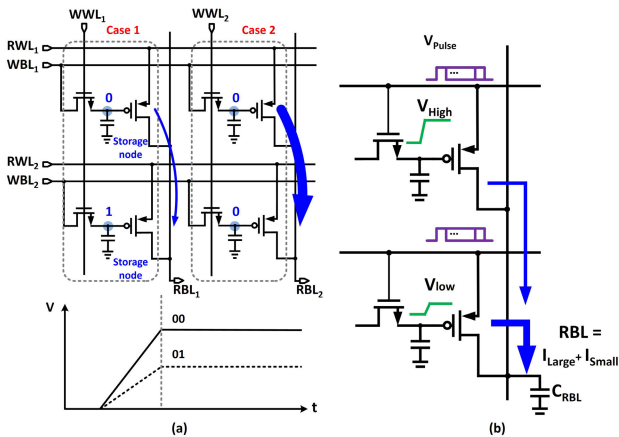


FIGURE 2. Conceptual illustration of in-memory computing operations using the proposed 2T1C DRAM cell: (a) Boolean function, (b) Multiplication-and-Accumulation.

based on the given cell structure. The details of the circuit implementation to enable dual-mode operation are described also in Section II. Section III summarizes the overall architecture of the proposed system, and Section IV presents the measurement results. Section V presents Limitation and Fixes of Architectural Scaling, which discusses additional studies conducted to evaluate the scalability of the proposed architecture. Section VI provides a comparison with prior PIM systems using a performance benchmarking table. Finally, Section VII concludes the paper.

II. PROPOSED 2T1C EMBEDDED DRAM-PIM MACRO

A. 2T1C EMBEDDED DRAM CELL AND BASIC RD/WR OPERATIONS

The dual-mode PIM architecture proposed in this article uses a conventional 2-transistor and 1-capacitor (2T1C) eDRAM cell configuration as shown in Fig. 1. In order to reconcile the performance requirements of 1) the basic read and write (RD/WR) operations as a conventional DRAM cell and 2) dual-mode operation including Multiplication-and-Accumulation (MAC) and Boolean operation as a Process-In-Memory (PIM) cell, nMOS write access transistor (Q_1) and pMOS read access transistor (Q_2) have been selected in the proposed cell design. In the case of write access transistor,

n-type provides fast write speed and efficient data transfer in voltage level that ranging from 0 to V_{DD} . On the other hand, p-type read access transistor helps the read-out mechanism of the proposed PIM architecture that charges the Read Bit-Line (RBL) by providing the current from the Read Word-Line (RWL) with a higher potential. Note that thick oxide transistors have been used for both RD/WR access transistors to achieve the required data retention time of the DRAM cell in a typical CMOS process, by leveraging their inherently low leakage current characteristics.

During the write operation, the voltage level to be stored is presented to the Write Bit-Line (WBL) and is stored to the storage capacitor (C) via the write access transistor when the Write Word-Line (WWL) is driven by the V_{PP} level. The voltage stored level depends on the operating mode. For instance, the minimum (0V) or maximum (V_{DD}) level of the voltage is used for normal memory and Boolean operations, whereas the predefined multi-level analog voltage is used for MAC operation.

Once the data is stored in the storage node Q, the amount of current flowing through the read access transistor during the read operation would be adjusted as the transconductance of the Q_2 transistor is controlled by the stored voltage level. It is worth noting here that the RWL is driven to be high for the predefined time step, with the time duration changing adaptively based on the mode of operation.

B. DUAL-MODE OPERATION USING 2T1C DRAM ARRAY

Apart from the basic RD/WR operations as the conventional DRAM, the proposed PIM array provides two distinct operations for the multipurpose use: 1) Boolean operation and 2) MAC operation. The basic operation principle of both operations is illustrated in Fig. 2.

For Boolean operation, two bit cells connected to the same RBL are activated simultaneously during the read operation which eventually mimics the 2-input logic gates (NAND and NOR functions). The data stored in the bit cells have either data '0' (0V) or data '1' (V_{DD}) value during this operation. Data '0' allows the read current to flow to the RBL (by enabling the maximum transconductance of the pMOS access transistor), while the data '1' prevents current conduction by turning off the pMOS read access transistor.

By exploiting these simple observations, the different combinations of data stored in two bit cells cause different currents to flow during the read operation, which ultimately causes different voltage levels to develop across the RBL capacitor. For example, the data combination '00' makes the maximum possible current to flow along the corresponding RBL and the voltage level at the end of the read operation becomes higher than the other combinations ('01', '10', and '11').

Compared to the relatively simple operating principle of the Boolean function, the MAC operation requires a more complex multiplication function. To perform an approximate arithmetic multiplication using the proposed PIM array, we first map the multiplier to the time duration of the

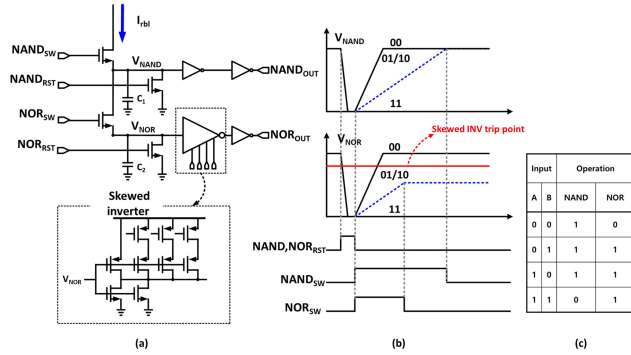


FIGURE 3. (a) Boolean output logic with Skewed inverter, (b) Timing diagram and expected waveform, (c) Expected output table.

write enable time and the multiplicand is assigned to the voltage level of the stored data. Note that the stored voltage level linearly controls the transconductance of the read access transistor. Then, the adjustable write enable time determines the duration of the current flow through the RBL. By combining these two linear factors, the output of the multiplication is represented by the voltage level across the RBL capacitor.

Note that, the above multiplication operation can be performed by several bit cells connected to the same RBL, where the accumulation operation between the multiplication outputs is naturally handled by Kirchhoff's Current Law (KCL) at the same node.

C. COLUMN LOGIC FOR BOOLEAN FUNCTION

The detailed column logic and the corresponding operation timing diagram for Boolean operations are shown in Fig. 3. Note that the Boolean operations supported in the proposed PIM architecture are NAND and NOR functions with two inputs. This means that two rows connected to the same column are activated simultaneously during the Boolean operation, and the amount of the read current (I_{rbl}) varies depending on the different combinations of the stored input data as explained in the previous section. When the I_{rbl} is presented to the Boolean column logic in Fig. 3(a), the current charges two capacitors (C_1 and C_2) and raises the voltage level of the V_{NAND} and V_{NOR} nodes, whose level is eventually developed to the logic high or logic low through two series inverters. It is worth noting that the voltage levels of V_{NAND} and V_{NOR} nodes are first discharged to 0V via $NAND_{RST}$, NOR_{RST} signals, and then are charged via I_{rbl} for predetermined time window dictated by $NAND_{SW}$, NOR_{SW} signals. The input and output of the 2-input NAND/NOR function here follows the truth table in Fig. 3(b). The input combination of '00' ('11') corresponds to the maximum (minimum, almost zero) current flow during the read operation, so the V_{NAND}/V_{NOR} nodes touch the V_{DD} (GND) during the Boolean operation. However, due to the different Boolean NAND/NOR outputs expected for the '01' and '10' input binary combinations, the pulse width of the $NAND_{SW}$ and NOR_{SW} signals is controlled to be different,

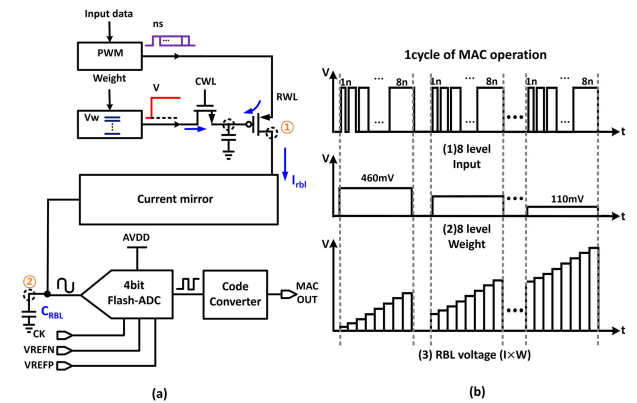


FIGURE 4. (a) MAC output logic with 4bit-Flash ADC, (b) Conceptual expected waveform of 1 cycle MAC operation.

with the enable time of the $NAND_{SW}$ signal being longer than that of the NOR_{SW} signal.

In this way, the output of the '01' and '10' input combinations becomes the logic high level from the $NAND_{OUT}$ node (blue dotted line in Fig. 3(b)), whereas NOR_{OUT} becomes the logic low level due to the insufficient voltage level developed during a shorter charge time (Fig. 3(b)). Note that, to avoid false triggering from the NOR inverter chain due to the intermediate voltage level with '10' and '01' input combinations, a skewed inverter with adjustable logic threshold voltage (V_t) has been adopted (bottom of Fig. 3(a)). From the simulation study, the V_t level of the skewed inverter can be $V_{DD}/2 \pm 15\%$, allowing an adaptive threshold function for robust operation. This level of variation in V_{th} turns out to be sufficient to ensure stable operation of the proposed Boolean logic even under combinations of five extreme process corners (TT, FF, SS, SF, and FS), $\pm 10\%$ of V_{DD} variations, and three different temperatures.

Furthermore, even though this system only supports NAND and NOR gate operations, these two gates are universal gates and can be utilized to create other Boolean logic combinations. In addition, the XOR function can also be implemented by utilizing additional simple logic gates and an inverter with adjustable threshold voltage (Not shown in the figure). In particular, the XNOR gate is used as a multiplier in many PIM papers ([21], [22], [23]). These facts show the scalability of this system and its potential to be used to implement more complex systems.

D. COLUMN LOGIC FOR MULTIPLICATION AND ACCUMULATION FUNCTION

The MAC operation of the proposed dual-mode PIM architecture can support a 3-bit multiplication between the input and weight values at each 2T1C memory cell, the output of which is converted into the RBL current (I_{RBL}) along the column. The basic idea of performing the MAC operation and the required circuit diagrams, along with the expected waveforms, are shown in Fig. 4(a) and (b), respectively.

For the multiplication, the input signal (multiplicand) is mapped to the read enable time (RWL pulse width) by

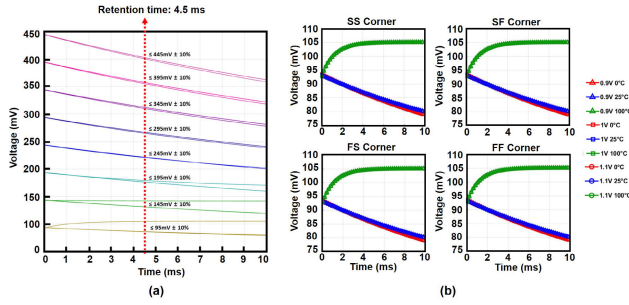


FIGURE 5. Corner simulation result of 8 level weight: (a) All 8 weight levels under PVT variation, (b) Detailed result of the worst case (lowest weight level).

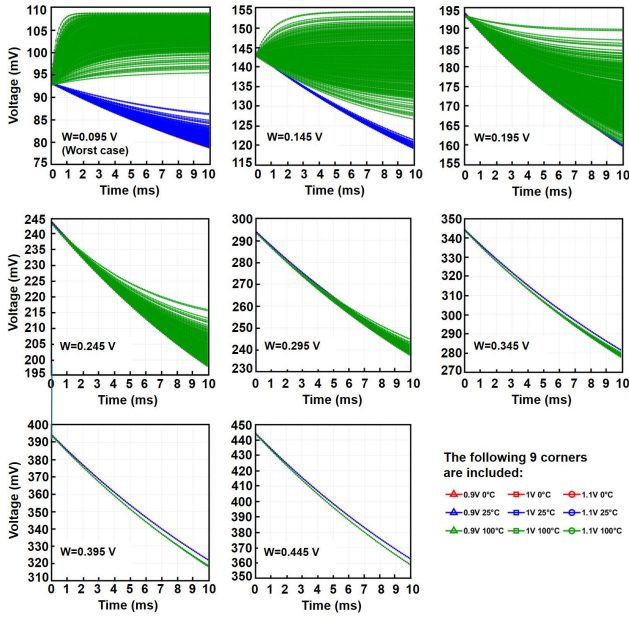


FIGURE 6. Retention time: Monte-Carlo simulation result of 8 level weight (0.095 ~ 0.445V, 50 mV step).

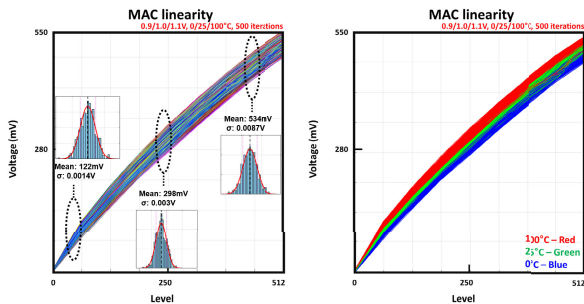


FIGURE 7. MAC linearity graph result of Monte Carlo and Corner simulation (500 iteration).

the Pulse-Width-Modulation (PWM) unit, where the pulse duration ranges from 1ns to 8ns with 1ns increments for fine granularity. The weight (multiplier) stored in the cell capacitor through the multi-level voltage step, which controls the conductivity of the read PMOS transistor. The discrete voltage level ranges from 460mV ~ 110mV at linear step, where the higher (lower) voltage level stored, more (less)

current flows during the read operation. This is because the conductivity of the read access transistor is inversely proportional to the stored voltage level, which modulates the V_{sg} value of the PMOS transistor during MAC operation. The resulting read current flows through the current mirror where the current level is reduced by a factor of 4 and then charges the RBL capacitor (C_{RBL}). Here the benefits of using a current mirror to reduce the current level are twofold. Firstly, the size of the C_{RBL} can be minimized, resulting in an area efficient PIM architecture. Second, by decoupling the original RBL source node (①) from the storage node (②), the non-linearity induced by the MAC current flow can be mitigated. This nonlinearity becomes severe as the voltage level on the RBL storage node increases, which reduces the drain-source voltage of the read access transistor. This situation is exacerbated by large currents flowing during read events, which would occur in the absence of a current mirror. Note that the read current of each cell connected to a single bit line is accumulated in the C_{RBL} according to Kirchhoff's Current Law (KCL), which completes the accumulation part of the MAC operation.

Fig. 4(b) illustrates all the possible MAC operation scenarios based on the proposed PIM architecture. The first column of the upper waveform shows the variation of the read enable time from 1ns to 8ns, which represent the changes in the input data, and the middle waveform shows the fixed weight value, which leads to the minimum conductance of the read access transistor. The multiplication between these two factors (varying pulse width and fixed voltage level) results in the staircase waveform at the bottom which shows the increase in the final read-out voltage across the C_{RBL} . As the weighting value in the middle of graph changes along the x-axis from maximum to minimum voltage level, the multiplication results change accordingly.

The voltage generated at C_{RBL} as a result of the MAC operation is converted to digital code by the ADC. The 4-bit Flash ADC used in this design was custom-designed to ensure reliable digital conversion of the MAC output. The ADC achieves a sampling rate of approximately 66.64MSPS and consumes only 8.35 μ W per MAC operation, as measured from post-layout simulations. The physical layout area of the ADC is approximately 8,192 μ m². To minimize offset-related errors, the internal comparator was custom-designed based on the method described in [24], without applying separate offset calibration. Post-layout simulation results confirmed that the static comparator offset is negligible relative to the full 4-bit LSB range, and thus does not introduce meaningful distortion in the digitized MAC output. These characteristics ensure that the quantization process faithfully preserves the linearity and accuracy of the proposed MAC architecture.

E. RETENTION TIME ANALYSIS AND ITS IMPACT ON MAIN FUNCTIONS

Since the proposed system uses DRAM cells for data storage, data loss due to leakage current over time in the data hold

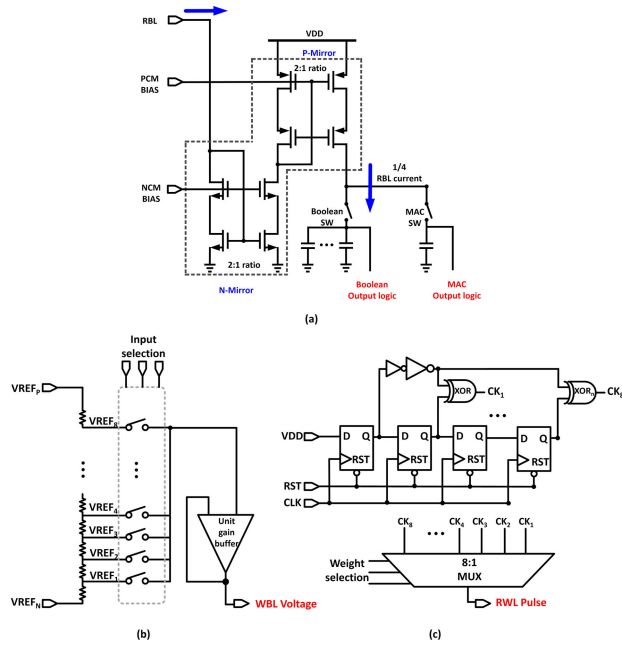


FIGURE 8. Details of column sub-units: (a) N/P Current mirror, (b) WBL driver, (c) RWL driver.

state cannot be avoided. In commercial DRAM processes, periodic refresh operation is essential to solve this problem, but PIM architectures that utilize various cell structures may have a different perspective. In this case, the critical factor is whether the retention time is sufficiently long compared to the time required to perform the MAC operation. From this perspective, to further validate the robustness of the MAC operation, we conducted a comprehensive analysis of retention behavior and output linearity under various PVT conditions. As shown in Fig. 5 and 6, the multi-level analog voltages stored in the cell capacitors demonstrate sufficient retention over 4.5ms, maintaining distinguishable voltage levels across all 8 quantization steps even under worst-case PVT corners. Moreover, Fig. 7 presents the MAC output linearity results based on 500 Monte-Carlo simulations across five process corners (TT,SS,SF,FF,FS) and three temperatures (0°C, 25°C, and 100°C). The proposed architecture consistently exhibits well-separated output voltage levels with low standard deviation ($\sigma < 8.7\text{mV}$), confirming stable MAC behavior under real-world variations.

F. REMAINING COLUMN SUB-UNITS

This section describes the detailed transistor-level implementation of the sub-units of column logic. As mentioned in the previous section, the current mirror is basically adopted to reduce the charging current of C_{RBL} for the 1) expand-ability (considering more cells connected to a single column) and 2) MAC linearity of the proposed system.

Figure 8(a) shows the current mirror with high impedance output resistance used in the design. In order to ensure the operation region of the internal transistors in saturation mode under the different test conditions, the bias-level (PCM BIAS,

NCM BIAS) of the N/P mirror is externally controlled. Once the resultant current from the MAC/Boolean computation is presented to the current mirror, a series of N-mirror and P-mirror reduces the amount of current by a factor of 4. The reduced current then charges the C_{RBL} array, where the capacitors are divided into two groups: one for MAC operation and the other for Boolean operation. The capacitance for Boolean operation is adjustable via a test signal ($300\text{ fF} \pm 100\text{ fF}$) to ensure the correct operation in the presence of various unwanted noise sources and/or process variations.

To perform the MAC operation, two multiplication factors are mapped to 1) 3-bit analog voltage level stored in the bit cell and 2) RWL pulse width with 8 different durations to control the read enable time. These two control signals are generated by the WBL control unit and RWL control unit in Fig. 8 (b) and (c), respectively. First, the bit cell voltage during the MAC operation exploits a resistor ladder with 8 intermediate levels and a unit gain buffer that buffers the selected reference level via an 8-to-1-line selector. Next, the RWL pulse width is controlled by the digital logic and 8:1 MUX as shown in Fig. 8(c). Here the 8 FFs inside the shift register are operating based on the same clock signal to shift the logic '1' level when a new clock edge comes in. The final output pulses ($CK1 \sim CK8$) with different widths are then generated using an XOR gate (XOR_n), where the upper input of each XOR gate is fixed to the first FF output, but the lower input uses the FF outputs of the shift register in sequential order.

III. OVERALL ARCHITECTURE

The proposed PIM architecture for dual-mode operation based on a 2T1C DRAM cell array is shown in Fig. 9. The memory array has a size of 8×8 and the WWL enables the column-wise write operation while the RWL enables the row-wise read operation. When the selected WWL becomes high from address decoding, depending on the type of operation supported by the system, data '0' or '1' will be stored in the bit cell for Boolean operation, or analog voltage level will be stored for MAC operation, both through the WBL.

The output of the operation is then converted into the current flow during the read operation, where the resulting current is presented to the column logic, which consists of current mirror, mode switches, and capacitors, as explained in the previous subsection.

Note that the result of the calculation is represented by the amount of current, which is finally converted to the voltage level by charging the RBL capacitor. The voltage level is then used to determine 1) the Boolean logic (2-input NAND/NOR) output for the Boolean operation or 2) 4-bit digital code from the ADC for the MAC operation, in each output logic.

Note that, in the proposed system, the switching between MAC and Boolean operations is done by selecting one of the two switches that exist after the Current-mirror, as shown in Fig. 8(a). This means that the proposed system does not focus

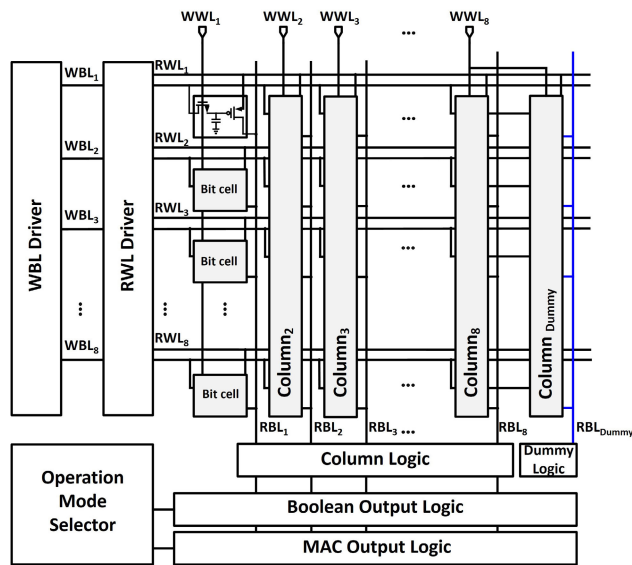


FIGURE 9. Overall PIM architecture with 8×8 eDRAM cell array for Dualmode operation.

on performing two operations simultaneously on a single array, but rather, as mentioned in the introduction, it focuses on applications that perform different operations sequentially. Importantly, this mode-switching approach allows for very efficient hardware with the minimal additional area or power consumption required for dual-mode operation.

It is worth note here that there is an additional column on the far right in Fig. 9. Here the ‘Dummy Column’ has exactly the same bit cell structure as the previous 8 columns.

This dummy column shares identical array-level layout, bit-cell circuitry, and analog readout path with the main computation columns, ensuring that the measurement reflects actual system behavior under the same parasitic and loading conditions. However, instead of the full column logic at the bottom for the normal column, the dummy column only has a current mirror followed by an I-V converter inside the ‘Dummy Logic’. This converts the amount of read current from the MAC calculation results into an analog voltage level that can be monitored by off-chip measurement.

IV. MEASUREMENT RESULTS

The proposed dual-mode PIM architecture is fabricated using a commercial 65nm CMOS process. A micrograph of the chip is shown in Fig. 10(a), which includes an 8×8 memory array, row/column logic, timing control circuit, and ADC. The chip measurement environment is shown in Fig. 10(b), where the FPGA and PC are used to provide multiple input signals such as address, command, and data with the exact timing. This macro achieves 1.74TOPs/W when performing the 3b X 3b MAC operation, as shown in Fig. 10(c).

Figure 11 shows the breakdowns of the power and area of the chip, where it is observed that the power consumption of the array and WBL driver occupies much of the total power. The WBL driver is adopted to each row, and a unit gain buffer

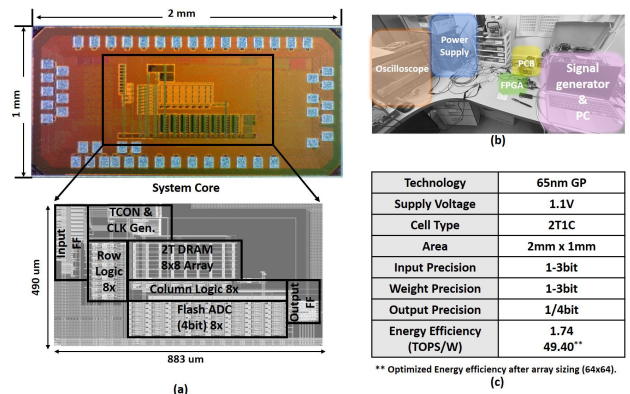


FIGURE 10. (a) Chip die photo, (b) Test setup, (c) Summary table.

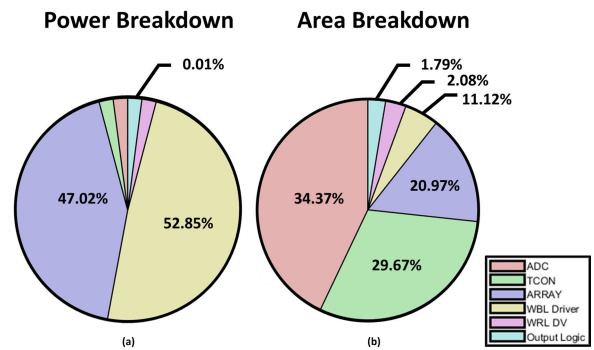


FIGURE 11. (a) Power breakdown, (b) Area breakdown.

is located at the output to drive all the cells connected to a single row to a specific voltage level. This results in the power consumption of the buffer being large enough to account for more than half of the total power consumption. In terms of area, the input/output interface circuits, namely the ADC and the timing controller, occupy a significant portion of the total area, primarily because the array size is relatively small in this prototype design.

A. BOOLEAN OPERATION WITH CONTROL LOGIC

We first verify the Boolean operation of the proposed PIM architecture. To check the outputs of NAND/NOR functions with all possible 2-input cases (‘00’, ‘01’, ‘10’, and ‘11’) at the same time, each input combination is stored in the 2 bit cells along the same column, and then 2 rows are activated simultaneously. Then, the results of 4 cases could be check at the same time at the column logic (Fig. 12(a)). The main control signals for the Boolean operation are depicted in Fig. 12(b), where the upper waveform shows the reset signal to discharge the RBL capacitor. After the reset period, the charge time for the NAND/NOR operation is controlled by the $NAND_{SW}$ and NOR_{SW} signal, where the NOR_{SW} period should have a shorter pulse for proper functionality. Based on the simulation setup described in Fig. 12, the output waveforms from the post-layout simulation are shown in Fig. 13. Here, the top two figures show the intermediate voltage level changes during the Boolean operations,

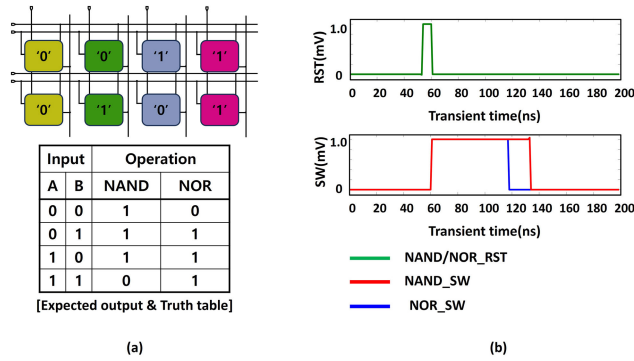


FIGURE 12. (a) Truth table of NAND and NOR, (b) Main control signal waveform for Boolean operation.

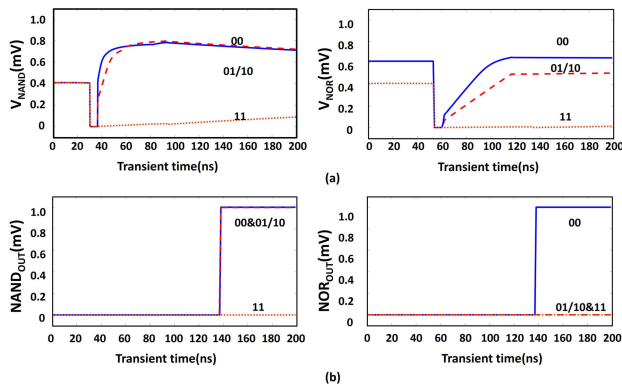


FIGURE 13. Post-layout simulation data of Boolean operation: (a) Node voltages V_{NAND} and V_{NOR} before being processed by the final Boolean output logic, (b) Final output result of NAND and NOR function.

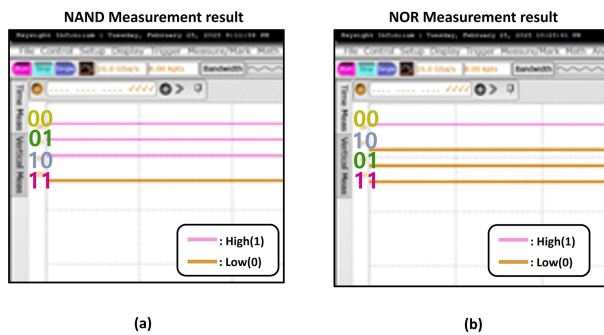


FIGURE 14. Measurement data of Boolean function: (a) NAND logic function, (b) NOR logic function.

while the bottom figure shows the final outputs of the NAND/NOR functions. Compared to the voltage level of V_{NAND} (top left) (~ 780 mV) with 01/10 input combinations, the V_{NOR} level (top right) is lower (~ 490 mV), which makes the final output of the NOR operation low after passing through the skewed inverter chain. The chip measurement results in Fig. 14 also show the outputs of each logic operation, where the NAND output (left) only shows '0', when both inputs are '1', while the NOR output (right) only shows '0' when both inputs are '0'.

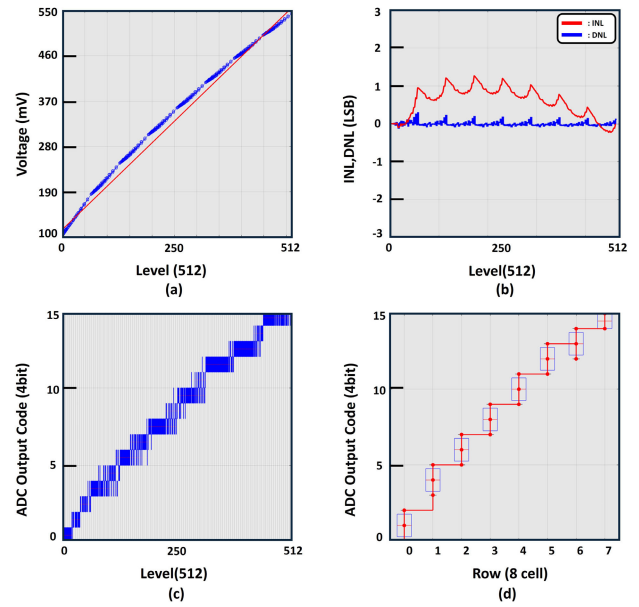


FIGURE 15. Post-layout simulation and measurement result of MAC operation: (a) Linearity of MAC output compared to the ideal line, (b) INL and DNL of MAC output obtained from post-layout simulation, (c) Measured 4-bit ADC output codes for all 512 MAC cases, (d) Averaged 4-bit ADC output codes measured from each of the 8 cells in a single column.

B. MULTIPLICATION AND ACCUMULATION WITH OUTPUT LOGIC

The linearity of the MAC operation using the proposed PIM architecture was verified by both the post-layout simulations and the actual measurement results. First, Fig. 15(a) shows the post-layout simulation results of the MAC operation using all eight cells connected to a single column. Since each cell supports $3\text{-bit} \times 3\text{-bit}$ multiplication (resulting in 64 levels), the proposed system achieves a total of 512 output voltage levels per column. The simulation results show that the overall linearity of all cells is reasonably good when compared with the ideal line. Fig 15(b) compares the INL and DNL values between the ideal MAC operation data and the MAC operation data obtained from the post-layout simulation, based on the graph in Fig 15(a). The simulation results show that as the output voltage increases, some nonlinear curvature appears due to the nonlinear I-V characteristics of the MOS transistor. Nevertheless, the linearity of the MAC operation using the proposed architecture still demonstrates reasonably good linearity ($\leq 1.2LSB$).

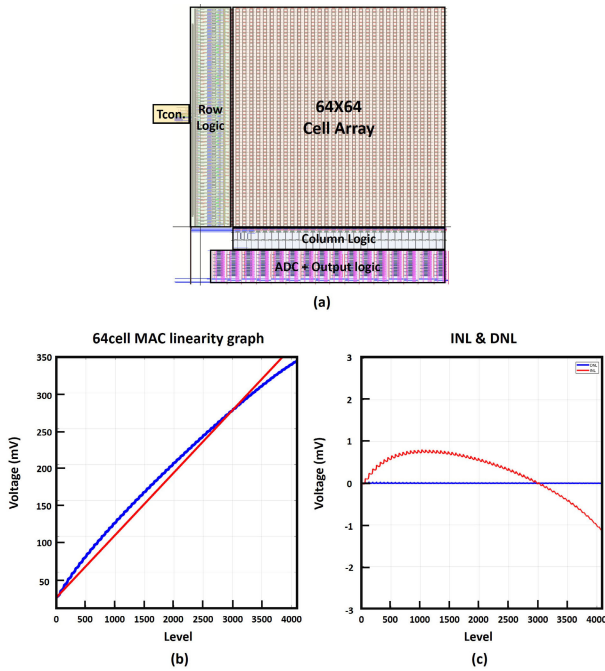
The linearity measurement results of the MAC operation in a single column are illustrated in Figures 15(c) and (d) using 4-bit ADC output codes. The first graph represents the ADC codes for all 512-level cases that can be measured in the single column, while the second graph shows the average ADC codes for each of the 8 cells. The actual measurement results reveal that the ADC output codes increase linearly with the expected MAC output voltage, covering a total of 16 output code levels in the graph. Note that the measurement

TABLE 1. Accuracy comparison of QAT-based MLP inference on MNIST and CIFAR-10 datasets with and without hardware-aware MAC modeling in 8×8 array system.

Data set	MNIST			CIFAR-10		
Model	784-300-10 MLP			RESNET-18		
Type	S/W	QAT + Linear MAC curve (8x8 Array)	QAT + Nonlinear MAC Curve (8x8 Array)	S/W	Linear MAC curve (8x8 Array)	Nonlinear MAC Curve (8x8 Array)
Accuracy(%)	98.01	97.83 (-0.18)	97.76 (-0.25)	94.48	94.39 (-0.09)	94.21 (-0.27)

TABLE 2. Accuracy comparison of QAT-based MLP inference on MNIST and CIFAR-10 datasets with and without hardware-aware MAC modeling in 64×64 array system.

Data set	MNIST			CIFAR-10		
Model	784-300-10 MLP			RESNET-18		
Type	S/W	QAT + Linear MAC curve (64x64 Array)	QAT + Nonlinear MAC Curve (64x64 Array)	S/W	Linear MAC curve (64x64 Array)	Nonlinear MAC Curve (64x64 Array)
Accuracy(%)	98.01	97.83 (-0.18)	97.57 (-0.44)	94.48	94.06 (-0.42)	94.02 (-0.46)

**FIGURE 16.** (a) 64×64 Array PIM system layout, (b) MAC linearity graph of 64×64 Array PIM system, (c) INL, DNL of graph.

results also reflect the nonlinear I-V characteristics of the MOS transistor, however, the overall characteristic of the designed hardware remains linear.

C. EVALUATION OF NEURAL NETWORK ACCELERATION PERFORMANCE

In this section, we evaluate the performance of the proposed hardware by performing inference tasks on the MNIST dataset at the software level. For performance evaluation, we utilized the Brevitas QAT framework and the 784-300-10 MLP model, incorporating the hardware characteristics listed below into the framework.

The input, weight, and output bit precisions were quantized to 3 b, 3 b, and 4 b, respectively. Subsequently, a transfer curve characterizing the nonlinearity observed in the actual MAC measurement results was formulated and integrated

into the MLP model. Based on this, the classification accuracy on the MNIST dataset was evaluated. As shown in Table 1, the accuracy decreased by 0.18% before incorporating the MAC nonlinearity, and by 0.25% after incorporating it. These results demonstrate that, despite the limitations in bit precision and the nonlinearity observed in MAC operations, the hardware architecture designed in this study successfully maintains high inference accuracy.

In addition to MNIST, inference was also conducted on a more complex dataset, CIFAR-10, using a ResNet-18 model. In this experiment, the same quantization scheme (3b input, 3b weight, and 4b output) was applied at the software level to reflect the hardware constraints. Without modeling MAC nonlinearity, the accuracy degradation was only 0.09%, and with the nonlinear transfer characteristics included, the degradation slightly increased to 0.27%. These results further confirm that the proposed MAC design preserves inference reliability even under deeper network architectures and more complex datasets.

Achieving high inference accuracy despite hardware non-idealities was made possible by a combination of circuit-level and algorithmic optimization strategies, some of the highlights of which are summarized below:

- **High Linearity (1.2LSB INL/DNL):** The MAC unit was carefully designed to achieve high linearity, with 1.2LSB-level Integral Nonlinearity (INL) and Differential Nonlinearity (DNL), as demonstrated in the 8×8 prototype results shown in Fig. 15(b).
- **Operation in the Triode Region:** The MAC transistors were configured to operate entirely within the triode region, ensuring a linear I-V response. The DAC voltage range was carefully selected to utilize only the most linear portion of the DRAM cell's I-V characteristics, and current mirrors were employed to stabilize current accumulation.
- **Quantization-Aware Training (QAT):** At the algorithmic level, the Brevitas-based QAT framework was adopted. This allowed the model to adapt during training to both quantization effects and hardware-induced nonlinearities, thereby preserving inference accuracy under constrained precision.

TABLE 3. Comparison table of the proposed dual-mode PIM system with prior works.

	TCASI'19[25]	ACCESS'21[26]	TVLSI'24[27]	TCASI'21[28]	TCASII'22[29]	ISCAS'24[30]	TVLSI'25[31]	TCASII'23[32]	This work
Process	65 nm	180 nm	28nm	65 nm	28nm	65 nm	65 nm	180nm	65 nm
Supply voltage	1V	1.8V	0.9V	1V	0.9V	1.2V	1.2V	1.8V	1.0V
Memory type	Dual-split 6T SRAM	10T SRAM	8T SRAM	4T2C eDRAM	2T1C eDRAM	1T1C eDRAM	2T1C eDRAM	3T2R ReRAM	2T1C eDRAM
Array Size	64x64	128x16	16KB	128x128	2.25KB	16 x 16	128x128	400KB	8x8
IN/W/OUT bit width	1/1/1	4/4/8	5/1/4	1-1.32/1.6-2.81/1-5	1/2/4	4/4/-	4-8/4-8/6	1/1/1	3/3/4
Operation mode	Single	Single	Single	Single	Single	Single	Single	Single	Dual
	Boolean	MAC	MAC	MAC	MAC	MAC	MAC	MACV	Boolean / MAC
Throughput (GOPs) ^(a)	278.2	1.2	410	-	-	6.4	-	200	5.3 341.0*
Energy Efficiency (TOPs/W)	30.49-55.8	24.2 4.90***	59.1 10.96***	17.8-552.25	81.5 15.12***	120.96	55.8	51.3 393.4***	1.74 49.40**
Area Efficiency (TOPs/mm ²)	33.13	169.90	-	309	2.3	-	-	-	1.42****
Target Dataset	MNIST	MNIST	MNIST	MNIST	CIFAR-10	MNIST	MNIST	MNIST	MNIST
			CIFAR-10					CIFAR-10	CIFAR-10
Accuracy	96.1%	97.5%	96.5%	96.8%	91.52%	97.0 %	97.4 %	95.6%	97.70%
			91.4%					86.2%	94.21%

V. LIMITATION AND FIXES OF ARCHITECTURAL SCALING

While the results based on the 8×8 array prototype validate the core functionality, they are not sufficient to fully capture practical issues that may arise during architectural scaling, such as RC delay, parasitic capacitance from interconnects, and signal distortion. To address these potential limitations, we conducted an additional study involving a scaled-up layout implementation of the entire system with a 64×64 memory array. Based on this extended design, we performed post-layout simulations including full parasitic extraction (PEX) to evaluate the actual operating characteristics.

The expanded design includes not only the memory array but also peripheral circuits such as wordline drivers, sense amplifiers, and ADCs. The full layout of the system is presented in Fig. 16(a). Furthermore, the MAC operation results from the post-layout simulation are shown in Fig. 16(b), which visually compares the accumulation output with the ideal linear response to assess nonlinearity across all 64 accumulation cases.

According to the results, the MAC output voltage closely follows an ideal linear response with negligible deviation compared to the 8×8 implementation. This confirms that the proposed architecture maintains high linearity even after system scaling, despite the presence of parasitic components that could introduce delay and signal degradation. The accumulation results continue to exhibit high accuracy, thereby experimentally validating the reliability and scalability of the proposed circuit structure.

In addition, Fig. 16(c) presents the quantitative nonlinearity analysis of the 64×64 system in terms of Integral Nonlinearity (INL) and Differential Nonlinearity (DNL). The

INL and DNL values were measured to be within $\pm 1.16\text{LSB}$, respectively, indicating that the MAC operation retains high linearity across the entire accumulation range. This further supports the feasibility of maintaining accurate computation even in large-scale implementations.

Based on the characteristics of the system with a 64×64 array as described above, we performed inference simulations using the MNIST and CIFAR-10 datasets. The results show only a minimal accuracy drop of 0.44% for MNIST and 0.46% for CIFAR-10 compared to the software baseline, confirming that the extended architecture retains high inference reliability. The accuracy results are summarized in Table 2, and these outcomes are also shown to be competitive in comparison with recent reference works, as presented in Table 3.

The improvement in throughput and energy efficiency stems from both array scaling and a parallel computing strategy that reduces power overhead. This enables the proposed system to achieve accurate inference with low power, demonstrating its scalability and practicality for large-scale PIM applications.

VI. COMPARISON WITH PRIOR PIM SYSTEMS

Table 3 shows a performance comparison between the proposed PIM system and other reference works targeting similar applications. Although the referenced papers differ in aspects such as memory type, array size, and input/output bit resolutions, they all report recognition accuracy and energy efficiency on the same MNIST dataset.

A key observation is that, unlike the other works that support only a single operation (either MAC or Boolean), the proposed system is capable of performing both types of

operations within a single architecture. Furthermore, despite incorporating hardware-level non-linearity into the MAC operation, our system demonstrates superior accuracy on the MNIST dataset compared to the other works.

However, the proposed system shows relatively lower throughput and energy efficiency compared to its counterparts. This is because the prototype chip was designed with a small array size, focusing on validating the feasibility of dual-mode functionality. If the array size is scaled up to 64×64 while maintaining the same architectural structure, the throughput and energy efficiency are expected to improve significantly—up to 341 GOPS and 49.40 TOPS/W—reaching or surpassing the levels reported in the other studies.

The area efficiency values reported in Table 3, like the energy efficiency values, are based on the scaled-up 64×64 array configuration rather than the fabricated 8×8 prototype. To support this scaling without excessive area overhead, several architectural optimizations were applied, including (1) redesigning the 2T1C cell using a fringe-capacitance-based layout, (2) Optimizing the overdesigned unit-gain buffer in the DAC in terms of power consumption, and (3) adopting a column-sharing scheme for the ADC. These improvements significantly reduced both per-cell and peripheral circuit area, leading to a $7 \times$ enhancement in area efficiency, reaching 1.4217 TOPS/mm². Such architectural scaling and design strategies not only improve throughput significantly but also limit power overhead, thereby contributing to the overall enhancement in energy efficiency. This comprehensive metric with respect to area enables competitive spatial efficiency in a variety of PIM systems. In particular, our proposed approach opens up the possibility of highly dense multi-level storage based on conventional charge-based memories (i.e., DRAM and SRAM), even though it requires somewhat burdensome input hardware - which can be improved upon.

VII. CONCLUSION

In this work, 2T1C DRAM cell-based Process-In-Memory architecture for dual mode operation - Boolean and MAC operations is proposed. To validate the target functionality, a prototype chip was designed with a small-scale 8×8 memory array, along with the required row/column circuit and computation logics. The prototype chip was fabricated using TSMC 65nm CMOS technology, operating at a 1V supply voltage and a 1GHz external clock input. The active area of the core occupies 0.482×0.382 mm², and the measurement results verify the high linearity of MAC operations and the accuracy of Boolean operations.

Furthermore, by evaluating the MLP model incorporating the hardware characteristics of the proposed system on the MNIST dataset, we observed only a 0.25% drop in MNIST classification accuracy and a 0.27% drop in CIFAR-10 classification accuracy compared to the software-level baseline, demonstrating the system's robustness against non-idealities and its suitability for real-world AI workloads.

Although the small array size results in relatively low throughput and energy efficiency, our analysis confirms that both metrics can be significantly improved by scaling up the array size while maintaining the same peripheral circuitry. The proposed dual-mode system architecture is well-suited for energy-efficient edge AI applications that require hierarchical processing, where Boolean logic blocks act as always-on event filters and MAC units are selectively activated for complex inference.

REFERENCES

- [1] V. Sharma, J. E. Kim, Y.-J. Jo, Y. Chen, and T. T. Kim, "AND8T SRAM macro with improved linearity for multi-bit in-memory computing," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2021, pp. 1–5.
- [2] A. Agrawal, A. Jaiswal, C. Lee, and K. Roy, "X-SRAM: Enabling in-memory Boolean computations in CMOS static random access memories," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 65, no. 12, pp. 4219–4232, Dec. 2018.
- [3] J. Kim, J. Koo, T. Kim, Y. Kim, H. Kim, S. Yoo, and J.-J. Kim, "Area-efficient and variation-tolerant in-memory BNN computing using 6T SRAM array," in *Proc. Symp. VLSI Circuits*, Jun. 2019, pp. C118–C119.
- [4] S. Jeloka, N. B. Akesh, D. Sylvester, and D. Blaauw, "A 28 nm configurable memory (TCAM/BCAM/SRAM) using push-rule 6T bit cell enabling logic-in-memory," *IEEE J. Solid-State Circuits*, vol. 51, no. 4, pp. 1009–1021, Apr. 2016.
- [5] A. Jaiswal, I. Chakraborty, A. Agrawal, and K. Roy, "8T SRAM cell as a multibit dot-product engine for beyond von Neumann computing," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 27, no. 11, pp. 2556–2567, Nov. 2019.
- [6] M. E. Sinangil, B. Erbagci, R. Naous, K. Akarvardar, D. Sun, W.-S. Khwa, H.-J. Liao, Y. Wang, and J. Chang, "A 7-nm compute-in-memory SRAM macro supporting multi-bit input, weight and output and achieving 351 TOPS/W and 372.4 GOPS," *IEEE J. Solid-State Circuits*, vol. 56, no. 1, pp. 188–198, Jan. 2021.
- [7] A. Biswas and A. P. Chandrakasan, "CONV-SRAM: An energy-efficient SRAM with in-memory dot-product computation for low-power convolutional neural networks," *IEEE J. Solid-State Circuits*, vol. 54, no. 1, pp. 217–230, Jan. 2019.
- [8] V. Seshadri, K. Hsieh, A. Boroum, D. Lee, M. A. Kozuch, O. Mutlu, P. B. Gibbons, and T. C. Mowry, "Fast bulk bitwise AND and OR in DRAM," *IEEE Comput. Archit. Lett.*, vol. 14, no. 2, pp. 127–131, Jul. 2015.
- [9] V. Seshadri, D. Lee, T. Mullins, H. Hassan, A. Boroumand, J. Kim, M. A. Kozuch, O. Mutlu, P. B. Gibbons, and T. C. Mowry, "Ambit: In-memory accelerator for bulk bitwise operations using commodity DRAM technology," in *Proc. 50th Annu. IEEE/ACM Int. Symp. Microarchitecture (MICRO)*, Oct. 2017, pp. 273–287.
- [10] S. Li, D. Niu, K. T. Malladi, H. Zheng, B. Brennan, and Y. Xie, "DRISA: A DRAM-based reconfigurable in-situ accelerator," in *Proc. 50th Annu. IEEE/ACM Int. Symp. Microarchitecture (MICRO)*, Oct. 2017, pp. 288–301.
- [11] Q. Deng, L. Jiang, Y. Zhang, M. Zhang, and J. Yang, "DrAcc: A DRAM based accelerator for accurate CNN inference," in *Proc. 55th ACM/ESDA/IEEE Design Autom. Conf. (DAC)*, Jun. 2018, pp. 1–6.
- [12] T. Yoo, H. Kim, Q. Chen, T. T. Kim, and B. Kim, "A logic compatible 4T dual embedded DRAM array for in-memory computation of deep neural networks," in *Proc. IEEE/ACM Int. Symp. Low Power Electron. Design (ISLPED)*, Jul. 2019, pp. 1–6.
- [13] Y. Wang, H. Tang, Y. Xie, X. Chen, S. Ma, Z. Sun, Q. Sun, L. Chen, H. Zhu, J. Wan, Z. Xu, D. W. Zhang, P. Zhou, and W. Bao, "An in-memory computing architecture based on two-dimensional semiconductors for multiply-accumulate operations," *Nature Commun.*, vol. 12, no. 1, p. 3347, Jun. 2021.
- [14] H. Diao, H. Luo, J. Song, B. Xu, R. Wang, Y. Wang, and X. Tang, "A 28nm 128TFLOPS/W computing-in-memory engine supporting one-shot floating-point NN inference and on-device fine-tuning for edge AI," in *Proc. IEEE Custom Integr. Circuits Conf. (CICC)*, Apr. 2024, pp. 1–2.

- [15] S. Yan, J. Yue, C. He, Z. Wang, Z. Cong, Y. He, M. Zhou, W. Sun, X. Li, C. Dou, F. Zhang, H. Yang, Y. Liu, and M. Liu, "A 28-nm floating-point computing-in-memory processor using intensive-CIM sparse-digital architecture," *IEEE J. Solid-State Circuits*, vol. 59, no. 8, pp. 2630–2643, Aug. 2024.
- [16] H. Jeong, S. Kim, J. Shin, K. Park, and K. J. Lee, "A 273.48 TOPS/W and 1.58 Mb/mm² analog-digital hybrid CIM processor with transpose ternary-eDRAM bitcell," in *Proc. IEEE Asian Solid-State Circuits Conf. (A-SSCC)*, Nov. 2024, pp. 1–3.
- [17] I. Choi, E. J. Choi, D. Yi, Y. Jung, H. Seong, H. Jeon, S.-J. Kweon, I.-J. Chang, S. Ha, and M. Je, "An SRAM-based hybrid computation-in-memory macro using current-reused differential CCO," *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 12, no. 2, pp. 536–546, Jun. 2022.
- [18] P.-C. Wu, J.-W. Su, L.-Y. Hong, J.-S. Ren, C.-H. Chien, H.-Y. Chen, C.-E. Ke, H.-M. Hsiao, S.-H. Li, S.-S. Sheu, W.-C. Lo, S.-C. Chang, C.-C. Lo, R.-S. Liu, C.-C. Hsieh, K.-T. Tang, and M.-F. Chang, "A floating-point 6T SRAM in-memory-compute macro using hybrid-domain structure for advanced AI edge chips," *IEEE J. Solid-State Circuits*, vol. 59, no. 1, pp. 196–207, Jan. 2024.
- [19] S. Jeong, J. Oh, and D. Jeon, "A 28nm 157TOPS/W 446.9Kb/mm² compute-in-memory SRAM macro with analog-digital hybrid computing for deep neural network inference," in *Proc. IEEE Custom Integr. Circuits Conf. (CICC)*, Apr. 2024, pp. 1–2.
- [20] S. Hong, W. Jo, S. Kim, S. Kim, S. Um, K. Sohn, and H.-J. Yoo, "Dyamond: Compact and efficient 1T1C DRAM IMC accelerator with bit column addition for memory-intensive AI," *IEEE J. Solid-State Circuits*, vol. 60, no. 4, pp. 1299–1310, Apr. 2025.
- [21] S. Yin, Z. Jiang, J.-S. Seo, and M. Seok, "XNOR-SRAM: In-memory computing SRAM macro for Binary/Ternary deep neural networks," *IEEE J. Solid-State Circuits*, vol. 55, no. 6, pp. 1733–1743, Jun. 2020.
- [22] H. Jiang, R. Liu, and S. Yu, "8T XNOR-SRAM based parallel compute-in-memory for deep neural network accelerator," in *Proc. IEEE 63rd Int. Midwest Symp. Circuits Syst. (MWSCAS)*, Aug. 2020, pp. 257–260.
- [23] N. Alnatsheh, Y. Kim, J. Cho, and K. K. Choi, "A novel 8T XNOR-SRAM: Computing-in-memory design for Binary/Ternary deep neural networks," *Electronics*, vol. 12, no. 4, p. 877, Feb. 2023.
- [24] B. Razavi, "The design of a comparator [The analog Mind]," *IEEE Solid State Circuits Mag.*, vol. 12, no. 4, pp. 8–14, Apr. 2020.
- [25] X. Si, W.-S. Khwa, J.-J. Chen, J.-F. Li, X. Sun, R. Liu, S. Yu, H. Yamauchi, Q. Li, and M.-F. Chang, "A dual-split 6T SRAM-based Computing-in-Memory unit-macro with fully parallel product-sum operation for binarized DNN edge processors," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 66, no. 11, pp. 4172–4185, Nov. 2019, doi: 10.1109/TCSSI.2019.2928043.
- [26] V. T. Nguyen, J.-S. Kim, and J.-W. Lee, "10T SRAM computing-in-memory macros for binary and multibit MAC operation of DNN edge processors," *IEEE Access*, vol. 9, pp. 71262–71276, 2021.
- [27] C. Peng, J. Guo, S. Yan, Y. Wei, X. Chen, W. Lu, C. Dai, Z. Lin, and X. Wu, "A 28-nm cascode current mirror-based inconsistency-free charging-and-discharging SRAM-CIM macro for high-efficient convolutional neural networks," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 33, no. 7, pp. 2044–2048, Jul. 2025.
- [28] C. Yu, T. Yoo, H. Kim, T. T. Kim, K. C. T. Chuan, and B. Kim, "A logic-compatible eDRAM compute-in-memory with embedded ADCs for processing neural networks," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 68, no. 2, pp. 667–679, Feb. 2021.
- [29] S. Ha, S. Kim, D. Han, S. Um, and H.-J. Yoo, "A 36.2 dB high SNR and PVT/Leakage-robust eDRAM computing-in-memory macro with segmented BL and reference cell array," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 69, no. 5, pp. 2433–2437, May 2022.
- [30] M. Jeong and W. Jung, "MAC-DO: DRAM-based multi-bit analog accelerator using output stationary," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2024, pp. 1–5.
- [31] X. Zhang, Y.-J. Jo, and T. Tae-Hyoung Kim, "A 65-nm 55.8-TOPS/W compact 2T eDRAM-based compute-in-memory macro with linear calibration," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 33, no. 5, pp. 1477–1481, May 2025.
- [32] Y. Li, J. Chen, L. Wang, W. Zhang, Z. Guo, J. Wang, Y. Han, Z. Li, F. Wang, C. Dou, X. Xu, J. Yang, Z. Wang, and D. Shang, "An ADC-less RRAM-based Computing-in-Memory macro with binary CNN for efficient edge AI," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 70, no. 6, pp. 1871–1875, Jun. 2023.



YERIM AN received the B.S. degree from the School of Electrical and Electronics Engineering, Tech University of Korea (TUKOREA), South Korea, in 2022, and the M.S. degree in intelligent semiconductor engineering from Chung-Ang University (CAU), Seoul, South Korea, in 2024, where she is currently pursuing the Ph.D. degree in intelligent semiconductor engineering. Her research interest includes process-in-memory (PIM) architecture design.



HONGGU KIM received the B.S. degree from the School of Electrical and Electronics Engineering, Chung-Ang University (CAU), Seoul, South Korea, in 2022, where he is currently pursuing the integrated M.S. and Ph.D. degree in intelligent semiconductor engineering. His research interests include neuromorphic hardware and software co-optimization and analog compute-in-memory architecture.



DONGJUN SON received the B.S. degree from the School of Electrical and Electronics Engineering, Chung-Ang University (CAU), Seoul, South Korea, in 2024, where he is currently pursuing the integrated M.S. degree in intelligent semiconductor engineering. His research interest includes SRAM based neuromorphic computing architecture.



HAKRAE YU received the B.S. degree from the School of Electrical and Electronics Engineering, Chung-Ang University (CAU), Seoul, South Korea, in 2025, where he is currently pursuing the integrated M.S. and Ph.D. degree in intelligent semiconductor engineering. His research interests include eDRAM and nonvolatile memory based computing-in-memory macro design.



YONG SHIM received the B.S. and M.S. degrees in electronics engineering from Korea University, in 2004 and 2006, respectively, and the Ph.D. degree from the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, USA, in 2018. He was a Memory Interface Designer with Samsung Electronics, Hwaseong, from 2006 to 2013. At Samsung, he has worked on the design and development of a memory interface for synchronous DRAMs (DDR1 and DDR4).

He is currently an Assistant Professor with Chung-Ang University. Prior to joining Chung-Ang University, in 2020, he was an SRAM Designer with Intel Corporation, Hillsboro, OR, USA, from 2018 to 2020, where he was involved in designing circuits for super-scaled next-generation SRAM cache design. His research interests include neuromorphic hardware and algorithm, in-memory computing, robust memory interface design, and emerging devices (RRAM, MRAM, and STO) based unconventional computing models.

...