2025 Design, Automation & Test in Europe Conference (DATE 2025)

A 101 TOPS/W and 1.73 TOPS/mm^2 6T SRAM-Based Digital Compute-in-Memory Macro Featuring a Novel 2T Multiplier

Priyanshu Tyagi
Indian Institute of Technology (IIT) Roorkee
priyanshu_t@ece.iitr.ac.in

Sparsh Mittal
Indian Institute of Technology (IIT) Roorkee
sparsh.mittal@ece.iitr.ac.in

***Abstract*—In this paper, we propose a 6T SRAM-based all-digital Compute-in-memory (CIM) macro for multi-bit multiply-and-accumulate (MAC) operations. We propose a novel 2T bitwise multiplier, which is a direct improvement over the previously proposed 4T NOR gate-based multiplier. The 2T multiplier also eliminates the need to invert the input bits, which is required when using NOR gates for multipliers. We propose an efficient digital MAC computation flow based on a barrel shifter, which significantly reduces the latency of shift operation. This brings down the overall latency incurred while performing MAC operations to 13ns/25ns (in 65nm CMOS)for 4b/8b operands (in 65nm CMOS @ 0.6V), compared to 10ns/18ns (in 22nm CMOS @ 0.72V) of the previous work. The proposed CIM macro is fully re-configurable in weight bits (4/8/12/16) and input (4/8) bits. It can perform concurrent MAC and weight update operations. Moreover, its fully complete digital implementation circumvents the challenges associated with analog CIM macros. For MAC operation with 4b weight and input, the macro achieves 24 TOPS/W at 1.2 V and 81 TOPS/W at 0.7 V. When using low-threshold-voltage transistors in the 2T multiplier, the macro works reliably even at 0.6V while achieving 101 TOPS/W.**

***Index Terms*—Compute-in-Memory (CIM), Static Random-Access Memory (SRAM), Neural Network (NN), Machine Learning (ML), Multiply-and-Accumulate (MAC).**

I. Introduction

The rapid growth of Artificial Intelligence (AI) and Deep Learning (DL) has transformed tasks like image classification and speech recognition. This progress is driven by the high computational demands of DL algorithms, which depend on complex MAC operations. Traditionally, these computations have been managed by von Neumann architectures [1]–[3], which struggle with the energy costs of data transfers between memory and processing units. However, the ever-growing need for real-time processing has directed attention toward Compute-in-memory (CIM) architectures [4]–[6]. They improve performance and energy efficiency by reducing data-movement overheads. CIM systems re-purpose a memory cell, such as SRAM, to perform computational tasks alongside its primary data storage function.

To realize a MAC operation in the SRAM cell, a binary input is applied to a word line (WL), while a binary weight is stored within the internal storage nodes of the SRAM cell. Such SRAM-based CIM architectures can be broadly categorized into analog CIM and digital CIM. Analog CIM performs MAC operations by accumulating voltage on the bit-line (BL), which is then converted to a digital output using analog-to-digital converters (ADCs). The ability of CIM macros to execute high-precision MAC operations determines inference accuracy in ML applications. However, Analog CIM architectures [7]–[11] encounter problems such as non-linearity, signal degradation, and low signal margin. This makes it difficult to achieve even 8-bit precision, which impacts overall computation accuracy.

Digital CIM architectures integrate computing circuits within the memory array to perform MAC operations and can perform loss-free accumulations [12]–[15]. Such CIM macros provide full accuracy since the bit-width of accumulated output can be set adequately long to support the largest possible sum. However, digital CIM architectures suffer large area overhead due to the utilization of bulky adder trees and non-6T/8T SRAM bit-cells. For example, Chi et al. [14] use 4T NOR gates below each 6T bit-cell for bitwise multiplications. Fujiwara et al. [15] couple NOR gates with 12T bit-cells to carry out 1b weight storage and bitwise multiplications. Kim et al. [12] utilize a compact bit cell comprising of a 6T SRAM cell, a bitwise multiplier (XNOR), two 2-to-1 multiplexers, and a full adder as a building block to perform MAC operations. While previous SRAM-based digital CIM works use non-6T/8T bulky SRAM bit-cells to achieve full precision MAC operations, we achieve the same functionality using a novel 6T+2T SRAM bitcell.

Contributions: In this paper, we propose a novel 6T SRAM-based all-digital CIM architecture for machine-learning applications. We introduce a novel 2T-based multiplier block which, when coupled with the existing 6T SRAM bit-cell, carries out 1b weight storage and bitwise product operations. We summarize our contributions as follows:

1. We introduce an area-efficient 2T-based multiplier. It reduces area, latency, and energy compared to the commonly used 4T NOR-gate-based multiplier.

2. The proposed CIM macro is fully re-configurable in both weight bits (4/8/12/16) and input bits (4/8). Moreover, the input bit precision can be further extended up to 16 bits.

3. The all-digital implementation of the proposed work ensures that the computational accuracy remains unaffected

even for larger output bit precision (> 12b).

4. Our design uses a barrel shifter to perform bit-shifting operations. It reduces the overall cycle time of the MAC operation while increasing system throughput.

5. The proposed CIM methodology enables massive parallelism as all 64 rows of the macro contribute simultaneously.

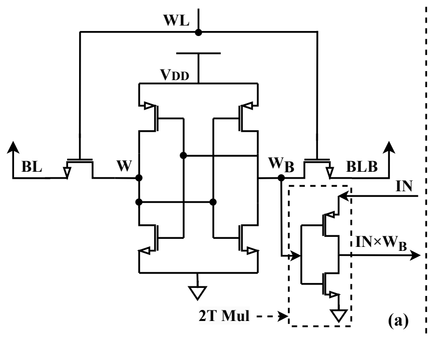
6. The proposed digital CIM macro can realize concurrent MAC + weight update operation and wide-range voltage operation (0.6 V to 1.2 V).

The rest of the article is organized as follows. Section II discusses the proposed novel 6T+2T bitcell, the overall CIM architecture and the methodology. We then outline the salient features of the proposed methodology in Section III. Next, Section IV thoroughly discusses the performance and stability of the proposed architecture via simulation results. It also discusses the simulated inference results before comparing the proposed work with other state-of-the-art works. Finally, Section V concludes this article.

II. Proposed Architecture & Methodology

A. Proposed 6T+2T Bitcell

We propose coupling the 6T SRAM bit-cell with a novel two-transistor multiplier block to facilitate the bit-wise binary product of the weight stored in the 6T bit-cell and the incoming input activation. The proposed 2T multiplier can be thought of as a modified CMOS inverter in which the input activation is provided at the source of the PMOS, and the complement of the weight stored in the SRAM bit-cell is applied as input to the inverter. The two additional transistors clamped to the storage node of the 6T SRAM cell effectively make the proposed bit-cell an 8T SRAM bit-cell, as shown in Figure 1.



(b)

Input (IN)	Weight W (W_B)	IN × W_B
0	0 (1)	0
0	1 (0)	0
1	0 (1)	0
1	1 (0)	1

Fig. 1. (a) Proposed 6T+2T bit-cell, and (b) truth table of 2T-based multiplier

Previous works [14], [15] couple every SRAM bit cell with a 4T NOR gate to achieve bit-serial multiplication. A NOR gate requires complements of input activation and weight bits. The complement of the stored weight bit is already present inside the bit cell. By contrast, our design couples the bit-cell with only two transistors and does not require inverting the input activation bits. As a result, the proposed design reduces area overhead, design complexity, and overall latency.

B. Overall Architecture and Methodology

Figure 2 shows the detailed architecture of the proposed 64×64 6T SRAM-based all-digital CIM macro for MAC operations. The CIM array is divided into 16 (64×4) sub-arrays for MAC operations. Each sub-array has 64 4b weights stored in 6T SRAM cells and 64×4 novel 2T-based bit-wise multipliers. CIM peripherals include a parallel adder tree, a clocked barrel bit-shifter, and a partial-sum accumulator. The macro supports 64 input activations with re-configurable bit-precisions (4/8) and four different bit-precisions (4/8/12/16) for the weights. Our design maintains full accuracy during MAC operations. This is achieved by setting the precision of the accumulated output bits high enough to support the largest possible sum.

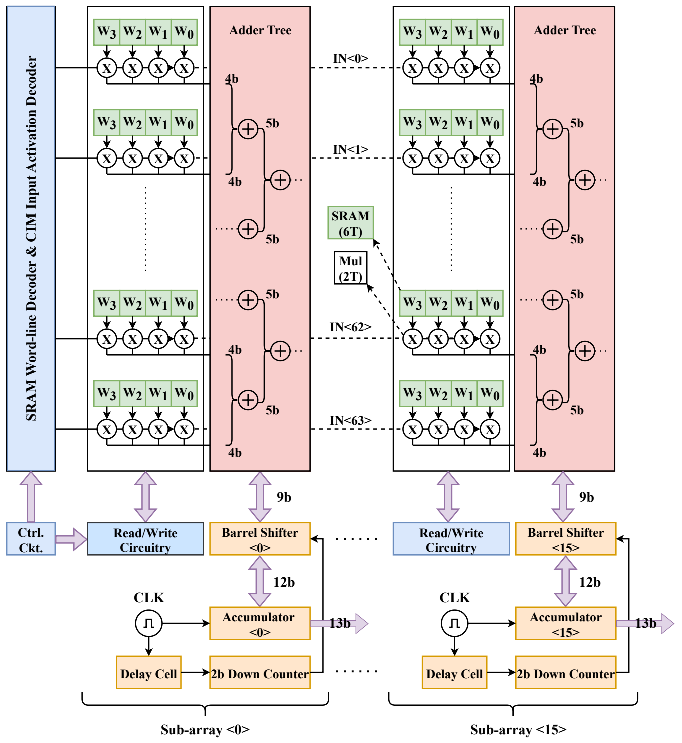


Fig. 2. Block diagram of proposed 64×64 CIM macro along with the peripheral circuitry

The proposed macro operates in two modes - SRAM and CIM mode. The basic read/write operations of memory are carried out in the SRAM mode. MAC operations are carried out in CIM mode without employing WLs and BLs. Since multiple rows can easily contribute without causing stability issues, our architecture achieves high parallelism. In the CIM mode of operation (activation $\neq 0$), in each of the 64 rows, an input activation is applied to the source of the PMOS of the 2T multiplier. For a multi-bit input activation, the bits are applied such that the MSB is fed in the first clock cycle and then the LSBs in the subsequent cycles.

The number of clock cycles required in the CIM operation depends upon the bit precision of the input activation. Figure 3 represents the overall timing waveform of the proposed computation flow. For the activations with 4b precision, multiplication is carried out over four cycles. In each cycle, there are 64 multiplications of 1b input activation and 4b weights, which are fetched directly from the internal storage node of SRAM. In each cycle, 64 4b products are fed to an adder tree, which generates the partial sum of all the multiplier outputs.

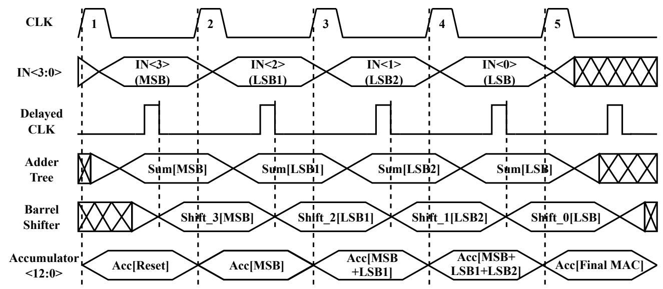


Fig. 3. The timing waveforms of the proposed computation flow

We use a barrel shifter to carry out the shifting operations on the partial sums generated by the adder tree. It utilizes 2-to-1 multiplexers, which select data depending on the input from a 2-bit down counter. The counter receives the clock pulses, corresponding to which it dictates the desired shift in bits of the generated partial sums. Then, in each cycle, the shifted partial sums are accumulated using an accumulator. Therefore, the MAC operation between 4b input and 4b weight takes a total of five cycles. The overall MAC operation can be summarized as: $\text{MAC}_j = \sum(\text{IN}_{3i} \times W_{ji}) \times 2^3 + \sum(\text{IN}_{2i} \times W_{ji}) \times 2^2 + \sum(\text{IN}_{1i} \times W_{ji}) \times 2^1 + \sum(\text{IN}_{0i} \times W_{ji}) \times 2^0$, where $i = 0$ to 63 and $j = 0$ to 3 for 4b weight precision.

III. Salient features of proposed methodology

SRAM-based CIM architectures face several challenges, such as limited precision, high power consumption, and complex integration of digital and analog components [6], [16], [17]. These challenges often hinder the effective implementation of high-precision MAC operations. In this section, we present the salient features of our design, demonstrating how we successfully address these challenges to achieve an efficient and robust CIM methodology.

1. Area efficiency and foundry compatibility: Recent CIM works have explored bit-cells with more than 8 transistors by clamping additional transistors to either/both of the storage nodes [12], [14], [15]. While these designs enhance bit-cell functionality, signal margin and inference accuracy, they lead to increased design complexity and area/energy overheads. While the foundry offers compact design rules for 6T and 8T SRAM, the addition of transistors reduces density by 2X to 4X and necessitates handcrafted design rules [18]. In fact, in advanced technology nodes (e.g. 28 nm or beyond), such exceptions are not permitted [18]. By contrast, our proposed 6T+2T bit-cell is comparable to the 8T bit-cell in terms of layout density and cell stability (as confirmed in Section IV-A). Hence, our design incurs minimal area overhead. 8T SRAM has already been integrated into commercial processors such as AMD's Zen [19], confirming the practical utility of our design. Further, our design does not require inverting input activation bits, which is the case when NOR gates are used as multipliers.

2. High signal margin: Most CIM macros provide limited flexibility in the input and weight bit-precision. Even an 8-bit precision is tough to achieve since it requires complex high-precision readout circuitry. For example, an in-array multiplication of 4-bit input and 4-bit weight results in 2^8 MAC voltages. For a 6T SRAM bit-cell, where the BL swing range is 0.5 V, such a low signal margin (0.5V/256 = 1.95 mV) is highly inadequate to perform readout operations that are robust against process variations. Our proposed approach overcomes this challenge by implementing MAC operation in the digital flow. This approach provides full accuracy since bit-precision of the accumulated result can be easily extended to represent the largest possible sum.

3. Overcoming non-linearity and signal degradation: CIM architectures are usually susceptible to non-linearity and signal degradation due to variations in device characteristics. Non-linearity in memory bit-cells and analog circuitry can introduce errors and inaccuracies in computation, impacting the reliability and overall precision of results. The all-digital implementation of our work seeks to overcome this challenge by eliminating the process variations and overhead associated with data conversion in analog circuitry.

4. High precision: The inference accuracy in DL applications directly depends upon the capability of CIM macro to carry out high-precision MAC operations [20]–[23]. In analog designs, the accuracy is restricted by the column-based ADC's precision ($\leq 8b$). Further, it incurs ADC-related truncation and gains errors. By contrast, the all-digital implementation of the proposed circuit helps us achieve a precision of 13 bits. In a neural network, this precision is sufficient for the vector calculations that follow MAC operations, including batch normalization, pooling, and activation [14].

5. Low latency: In CIM architectures, latency is a critical factor, especially when performing complex operations like MAC. It can arise from bitline charging, sense amplifier activation, and ADC operations, particularly in high-precision tasks. We propose an ADC-less CIM methodology to overcome latency issues. Furthermore, the barrel shifter carries out the entire shift operation in a single cycle. It reduces the overall cycle time of MAC operation and enhances system throughput.

6. Massive parallelism: Our architecture can perform massively parallel MAC operations combined with concurrent weight updates without encountering read/write disturb issues. All 64 rows contribute to computations, ensuring parallelism. Additionally, MAC operations do not interfere with the normal read/write operations. Consequently, weights can be updated simultaneously.

IV. Results

The proposed design has been implemented in Cadence Virtuoso using 65nm CMOS process technology.

A. Layout and bitcell parameters

We compare our design with the standard 6T, 8T SRAM cells and a recently proposed '6T+4T' bit-cell [14]. Figure 4 shows the layouts of these bit cells, and Table I presents their comparative analysis. Evidently, the proposed 6T+2T bitcell is comparable to the 8T SRAM bitcell in terms of layout density and other parameters. In a 65nm process, the 8T

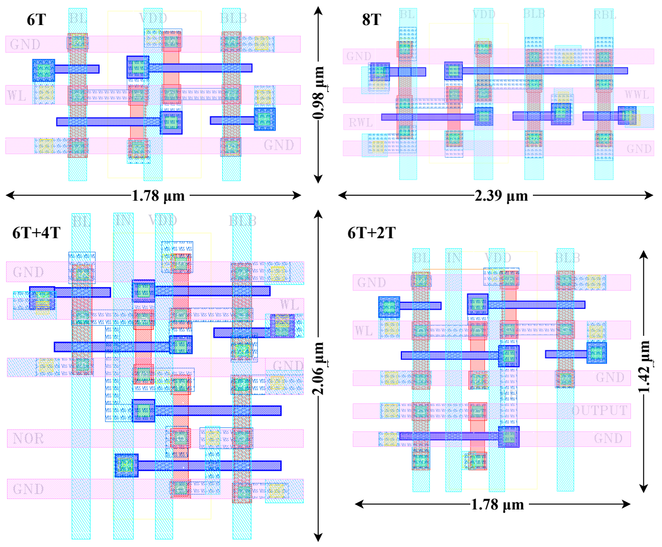


Fig. 4. Layouts of 6T, 8T, 6T+4T and our bit-cell (6T+2T) in 65nm

SRAM cell occupies an area of 2.35 μm², while the 6T+2T bit-cell occupies 2.52 μm², making their sizes almost equivalent. Both the 8T and 6T+2T configurations clamp two additional transistors to the internal storage node, resulting in nearly identical read/write parameters.

Additionally, the proposed 6T+2T bitcell introduces the ability to perform bitwise multiplication, which is not possible with conventional 6T and 8T SRAM bitcells. In previous designs, bitwise multiplication has been achieved by coupling a 4T NOR gate with a 6T SRAM bitcell. However, this method required inverting the applied input bits to achieve the desired functionality. In contrast, our design eliminates the need for input inversion. It directly processes the input bits and uses the complement of the weight bits, which is already stored at one of the SRAM nodes. This approach simplifies the operation and reduces area overhead, design complexity, and overall latency.

TABLE I
COMPARSION OF BIT CELLS AT 65NM CMOS PROCESS AND 1V SUPPLY

Parameters	6T	8T	6T+4T	6T+2T (Ours)
Bitcell Area (um^2)	1.75	2.35	3.67	**2.52**
4kb Array Size (mm^2)	0.0089	0.0123	0.0185	0.0131
Bitwise Multiplication	No	No	Yes	**Yes**
Layout Density	High	High	Low	High
Read Delay (ps)	106.2	102.7	127.5	108.2
Read Energy (fJ/bit)	62.4	58.8	69.2	63.5
Write Delay (ps)	169.6	164.3	216.7	173.2
Write Energy (fJ/bit)	140.4	133.1	155.8	139.7

B. Energy and Area Efficiency

Figure 5 visualizes the area and energy breakdown of our 64×64 CIM architecture. The adder tree occupies the most area as it requires multiple adders to handle high-precision MAC operations. The bit-shifter has the second highest area, as it utilizes 2-to-1 multiplexers. Despite this, the bit-shifter consumes relatively low energy, as our barrel shifter completes shifts in a single cycle and is only active for short periods. The accumulator, however, consumes the most energy because it is continuously active, handling ongoing accumulation of results and demanding constant power.

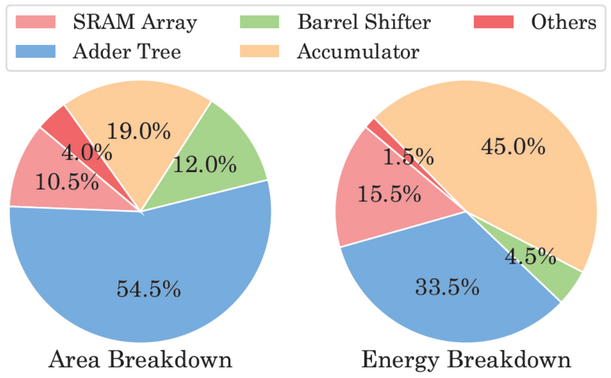


Fig. 5. Area and energy breakdown of the proposed 64×64 CIM macro

The 64×64 macro performs $64 \times 64 \times 2 = 8192$ operations in a single cycle (since 1 MAC = 2 operations) and the duration of each cycle is 4 ns. Hence, the macro performs 0.63 TOPS. Simulation results show that our architecture achieves varying energy efficiency levels depending on the input toggle rates and weights. We carry out measurements for a sparse pattern with a 16% input toggle rate and weights consisting of 50% 1s at 25°C and a 0.7 V supply. For 4b-input and 4b-weight precision, the macro consumes an average power of 8.04 mW. Hence, energy efficiency comes out to be 81 TOPS/W. Figure 6(a) shows the TOPS/W metric for a range of supply voltages for different input/weight precisions.

Sparsity analysis: We further measure TOPS/W for a range of input toggle rates: 100%, 75%, 50%, and 25%, with weights consistently having 50% 1s. Figure 6(b) shows the results across a range of supply voltages. Our results present a clear trend where higher input toggle rates lead to reduced energy efficiency. For example, at 0.7 V, TOPS/W drops to 78 and 66 for 50% and 100% input toggle rates, respectively. This is due to the increased switching activity, leading to higher dynamic power consumption and reduced energy efficiency.

Each 64×4 sub-array, along with its dedicated peripherals such as adder tree, bit-shifter, and accumulator, occupies an area of 0.022 mm^2. The entire 64×64 digital CIM macro utilizes an estimated area of 0.365 mm^2, achieving an area efficiency of 1.73 TOPS/mm^2 in 65nm technology node.

C. Results with standard-V_{TH} and low-V_{TH} transistors

Our 2T multiplier logic shows an intriguing behavior when we use standard threshold voltage (V_{TH}) transistor models. When both the inputs to it are zero, the output voltage (V_{00}) remains around 220 mV instead of the expected 0 V. This phenomenon is attributed to the characteristics of the PMOS transistor. In this scenario, the PMOS transistor is on, but because both the gate and source are at 0 V, the drain of the PMOS cannot be pulled up to V_{DD}. Instead, it settles at the PMOS threshold voltage (V_{THP}), which is 220 mV. However, since V_{THP} is well within the $V_{DD}/2$ value, this small voltage value has no impact on the digital computation accuracy, especially for large V_{DD}.

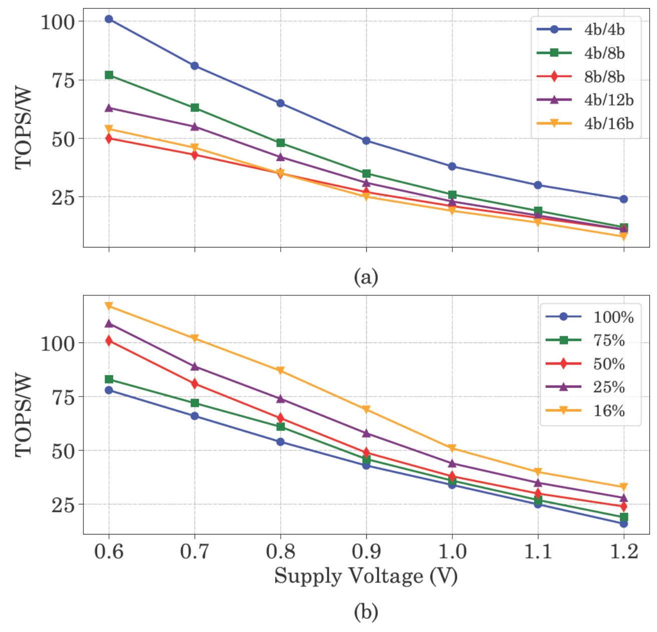


Fig. 6. TOPS/W for V_{DD} levels: (a) for different input/weight precision, and (b) for varying input toggle rates with 50% of weights set to 1s.

D. Stability analysis of 2T multiplier block

In our design, we apply the input to the power supply terminal (source of PMOS) of the 2T multiplier unit. This means that during computation, the output swing is directly influenced by the input voltage. Consequently, the output of the 2T multiplier does not achieve a full swing (as discussed above for V_{00}). However, since our CIM methodology is fully digital, any irregularities in the output from the 2T multiplier are rectified as the signal passes through the adder tree. This ensures that an accurate full voltage swing is maintained at the output bits of the adders. We verified the stability of our design by running Monte Carlo simulations over 1000 points. These simulation results, illustrated in Figure 7 confirm that the output attains full voltage swing across various simulation conditions. This confirms the stability of the proposed 2T multiplier logic.

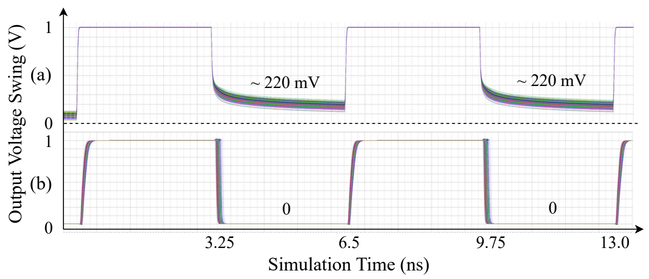


Fig. 7. Monte Carlo analysis of voltage levels observed at the output of (a) 2T multiplier and (b) adder tree. The waveform suggests that even though the 2T multiplier fails to attain full voltage swing at its output, the digital adder tree makes sure that the full voltage swing is obtained.

E. Stability Analysis of overall CIM operation

Corner analysis: We carry out corner analysis for a wide range of V_{DD}, viz., 0.6 V to 1.2 V. The corner analysis shows that on using standard V_{TH} transistor, for V_{DD} values below 0.7 V, the V_{00} becomes close to $V_{DD}/2$, e.g., for V_{DD} = 0.6 V, V_{00} becomes 310 mV at the worst corner. Ideally, V_{00} needs to be 0 (LOW). When it becomes 310 mV, it is interpreted as a HIGH level. This value falls within the meta-stability range of our 2T multiplier, which is a modified inverter. It leads to potential bit flips and inaccurate computations.

Monte Carlo analysis: Based on the above behavior, we ran 5000 Monte Carlo simulations for each V_{DD} between 0.6 V to 1.2 V. For 4b-input/weight MAC operation, we tapped the voltages of the corresponding output bits. For each V_{DD}, we took the average of the observed standard deviation values over all the 13 bits. As shown in Figure 8(a), the standard deviation increases on reducing the voltage and shows a sharp increase for V_{DD} below 0.7 V.

To mitigate these variations at low voltage, we propose using low V_{TH} transistors for 2T multiplier. With them, at V_{DD} of 0.6 V, V_{00} remains 190 mV, which is well within the meta-stability range. Thus, a low V_{TH} transistor helps realize full-precision MAC operations for an even wider supply voltage range of 0.6 V to 1.2 V. However, this comes with the drawback of increased leakage current. This can lead to increased power consumption, especially in idle or standby modes.

F. Application of CIM for a neural network

The proposed macro's bit-width flexibility allows it to support neural networks with a variety of topologies. We implement a Binary Neural Network (BNN) to classify the MNIST dataset using the PyTorch framework. The network comprises two fully connected layers with dropout regularization and employs inference by calculating the KL divergence. We vary the voltage of 2T multiplier logic and record the standard deviation in the final output. We add this 'noise' to BNN layers while training, in order to obtain the best accuracy during inference. After training for 100 epochs, we achieved a training accuracy of 99% and a test accuracy of 98.5%.

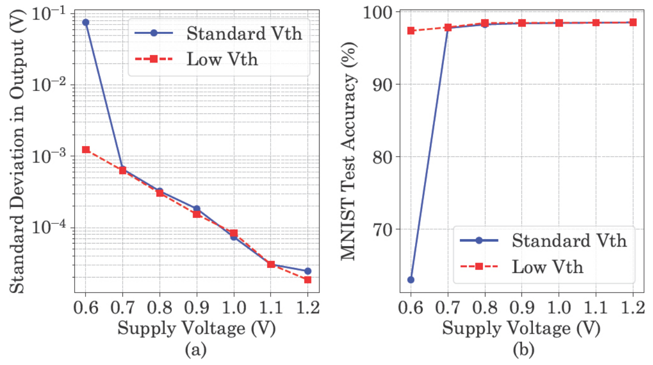
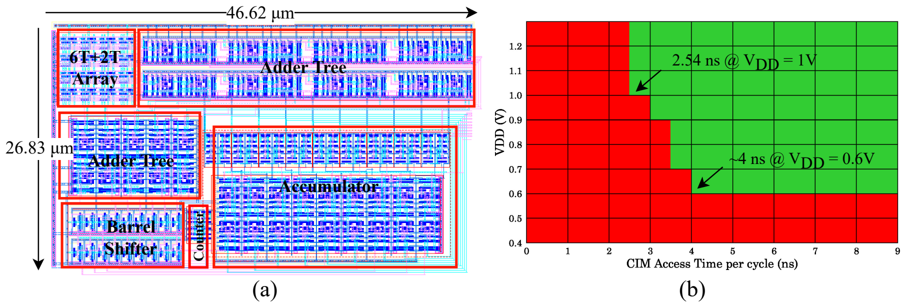


Fig. 8. Impact of V_{DD} on the (a) standard deviation in the output value. (b) BNN accuracy on the MNIST dataset.

Figure 8(b) depicts the test accuracy for various V_{DD} values with both standard V_{TH} and low V_{TH} transistors. Clearly, while a low V_{TH}-transistor maintains accuracy for the entire 0.6-1.2 V range, the standard V_{TH}-transistor maintains it only for the 0.7-1.2 V range. Thus, the standard V_{TH}-transistor shows a drop in accuracy for voltage below 0.7 V. For instance,



Technology Node	65nm
Supply Voltage	0.6 V - 1.2 V
Bitcell	6T+2T (2.52 μm^2)
Macro Area	0.365 mm^2
Reconfigurability	4/8b input & 4/8/12/16b weight
Overall CIM Access Time	8.5 ns (4b) - 25.4 ns (8b)
Frequency	40 MHz @ 0.6 V - 117 MHz @ 1.2 V
Power	6.25 mW @ 0.6 V - 30 mW @ 1.2 V
Simulated Accuracy (MNIST)	98.50%
GOPS	328 (8b) - 630 (4b)
TOPS/W	49 (8b) - 101 (4b) @ 0.6 V
TOPS/mm^2	0.89 (8b) - 1.73 (4b)

(c)

Fig. 9. (a) Layout of one of the sub-arrays (4 × 4) of the proposed CIM macro showing the placement and routing of major digital blocks (b) Shmoo plot of the measurement results for 4b-input and 4b-weight MAC operations (c) Summary table offering a comprehensive overview of the performance metrics of proposed CIM architecture and methodology

TABLE II
COMPARISON WITH STATE-OF-THE-ART CIM-BASED MAC IMPLEMENTATIONS (NA - NOT APPLICABLE, NR - NOT REPORTED)

Parameters	ISSCC'18 [24]	ISSCC'19 [25]	ISSCC'19 [26]	ISSCC'22 [15]	ISSCC'21 [14]	JSSC'17 [13]	ESSCIRC'19 [12]	This work
Domain	Analog	Analog	Analog	Digital	Digital	Digital	Digital	Digital
Technology	65nm	65nm	55nm	5nm	22nm	65nm	65nm	65nm
CIM Array Size	4 kb	16 kb	3.8 kb	64 kb	64 kb	181.5 kb	16 kb	4 kb
Bitcell Used	S6T	10T	Twin-8T	1R1W 12T	6T + 4T NOR	6T	6T	6T+2T
Bitcell Area (μm^2) ↓	0.525	NR	0.865	0.075	0.379	NR	10.528	2.52
CIM Macro Area (mm^2) ↓	NR	0.067	0.037	0.0133	0.202	12.25	0.2272	0.365
Sub-array Size and Count	NR	16 × 16 and 64	NR	64 × 4 and 64	256 × 4 and 16	512 × 64 and 25	128 × 8 and 16	64 × 4 and 16
Supply voltage range (V)	0.8 - 1	1.2	1	0.5 - 0.9	0.72	0.82 - 1.17	0.6 - 0.8	0.6 - 1.2
Input Precision (bit) ↑	1	7	4	4	1 to 8	16	1 to 16	4/8
Weight Precision (bit) ↑	1	1	5	4	4/8/12/16	16	4/8/12/16	4/8/12/16
Output Precision (bit) ↑	1	7	7	14	16 (4b), 24 (8b)	NR	8 to 23	13 (4b), 18 (8b)
Cycle Time (ns) ↓	2.3	150	10.2	NR	10 (4b), 18 (8b)	NR	NR	4 (4b), 7.25 (8b)
Throughput (GOPS) ↑	1780	10.67	17.6	NR	3300 (4b), 917 (8b)	0.042	567 (1b)	630 (4b), 328 (8b)
Energy Efficiency (TOPS/W) ↑	55.6	28.1	18.4	254 (4b), 63 (8b)	89 (4b), 24.7 (8b)	0.12	117.3 (1b)	101 (4b), 49 (8b) @0.6V
Area Efficiency (TOPS/mm^2) ↑	NR	0.16	NR	221 (4b), 55 (8b)	16.3	0.0034	NR	1.73
Simultaneous Weight Updates	NA	NA	NA	Yes	No	No	No	Yes

at 0.6 V, the test accuracy drops to 63.5% from a value of 98.5% achieved at higher voltages. In contrast, using low V_{TH} transistors, accuracy remained close to 98.5% even at 0.6 V.

G. *Overall summary and Comparison with recent works*

Figure 9 summarizes the proposed CIM design and offers a comprehensive overview of its performance metrics and efficiency. In Figure 9(a), the layout of one of the sub-arrays is depicted, showcasing the core architecture of our design. Figure 9(b) features the shmoo plot, which demonstrates the stability of MAC operations across different supply voltages, highlighting the robustness of the design under varying conditions. Lastly, Figure 9(c) provides a detailed table summarizing our design's key features and achievements.

Table II compares our work with state-of-the-art works. The analog approaches [24]–[26] tradeoff accuracy for area and energy efficiency. Due to the limited signal margin of the analog domain, it is challenging to achieve low-voltage operation. Compared to the digital implementations that use NOR gates [14], [15], our design reduces area overhead and overall access time by utilizing 2T multipliers. This also leads to better energy efficiency. Further, our design achieves higher area efficiency compared to the [25] design, which is also implemented in 65nm. Additionally, compared to the designs in [12] and [13], both of which are digital implementations in 65nm, our design demonstrates better throughput, energy efficiency, and area efficiency. Moreover, our design performs MAC operations across the largest supply voltage range of 0.6 V to 1.2 V. The ability of our macro to support concurrent MAC and weight update operations can further improve the overall energy efficiency and system performance.

V. CONCLUSION

In this work, we present a novel 6T SRAM-based all-digital CIM macro designed to enhance performance and energy efficiency. By processing NNs with re-configurable input and weight bit precision and utilizing a custom-designed 2T multiplier, our architecture significantly improves accuracy and energy efficiency over existing analog and digital implementations. Key features of our design include complete digital circuit implementation, bit precision reconfiguration, support for concurrent MAC and weight update operations, and dynamic voltage scaling. Implemented on a 65nm CMOS process, our macro achieves an impressive energy efficiency of up to 81 TOPS/W (101 TOPS/W with low V_{TH} 2T multiplier), demonstrating its potential for high-performance ML applications.

References

[1] S. Park, I. Hong, J. Park, and H.-J. Yoo, "An energy-efficient embedded deep neural network processor for high speed visual attention in mobile vision recognition SoC," *IEEE Journal of Solid-State Circuits*, vol. 51, no. 10, pp. 2380–2388, 2016.

[2] Y.-H. Chen, Krishna *et al.*, "Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks," *IEEE journal of solid-state circuits*, vol. 52, no. 1, pp. 127–138, 2016.

[3] N. P. Jouppi *et al.*, "In-datacenter performance analysis of a tensor processing unit," in *Proceedings of the 44th annual international symposium on computer architecture*, 2017, pp. 1–12.

[4] Z. Lin *et al.*, "A review on SRAM-based computing in-memory: Circuits, functions, and applications," *Journal of Semiconductors*, vol. 43, no. 3, p. 031401, 2022.

[5] D. Kim *et al.*, "An overview of processing-in-memory circuits for artificial intelligence and machine learning," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 12, no. 2, pp. 338–353, 2022.

[6] S. Mittal, G. Verma, B. K. Kaushik, and F. Khanday, "A Survey of SRAM-based In-Memory Computing Techniques and Applications," *Journal of Systems Architecture*, 2021.

[7] J.-W. Su, Y.-C. Chou *et al.*, "16.3 A 28nm 384kb 6T-SRAM Computation-in-Memory Macro with 8b Precision for AI Edge Chips," in *2021 IEEE International Solid-State Circuits Conference (ISSCC)*, vol. 64, 2021, pp. 250–252.

[8] K. Lee, J. Kim *et al.*, "Low-Cost 7T-SRAM Compute-in-Memory Design Based on Bit-Line Charge-Sharing Based Analog-to-Digital Conversion," in *Proceedings of the 41st IEEE/ACM International Conference on Computer-Aided Design*, ser. ICCAD '22, 2022.

[9] C. Yu *et al.*, "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, 2022.

[10] K. Lee *et al.*, "A charge-sharing based 8T SRAM in-memory computing for edge DNN acceleration," in *DAC*, 2021, pp. 739–744.

[11] K. Lee, J. Kim *et al.*, "Low-cost 7t-sram compute-in-memory design based on bit-line charge-sharing based analog-to-digital conversion," in *Proceedings of the 41st IEEE/ACM International Conference on Computer-Aided Design*, 2022, pp. 1–8.

[12] H. Kim *et al.*, "A 1-16b precision reconfigurable digital in-memory computing macro featuring column-MAC architecture and bit-serial computation," in *ESSCIRC*, 2019, pp. 345–348.

[13] Y.-H. Chen *et al.*, "Eyeriss: An Energy-Efficient Reconfigurable Accelerator for Deep Convolutional Neural Networks," *IEEE JSSC*, vol. 52, no. 1, pp. 127–138, 2017.

[14] Y. Chi *et al.*, "16.4 An 89TOPS/W and 16.3 TOPS/mm2 all-digital SRAM-based full-precision compute-in memory macro in 22nm for machine-learning edge applications," in *ISSCC*, vol. 64, 2021, pp. 252–254.

[15] H. Fujiwara *et al.*, "A 5-nm 254-TOPS/W 221-TOPS/mm 2 fully-digital computing-in-memory macro supporting wide-range dynamic-voltage-frequency scaling and simultaneous MAC and write operations," in *ISSCC*, vol. 65, 2022, pp. 1–3.

[16] S. Yu, H. Jiang *et al.*, "Compute-in-Memory Chips for Deep Learning: Recent Trends and Prospects," *IEEE Circuits and Systems Magazine*, vol. 21, no. 3, pp. 31–56, 2021.

[17] C.-J. Jhang, C.-X. Xue *et al.*, "Challenges and trends of SRAM-based computing-in-memory for AI edge devices," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 68, no. 5, pp. 1773–1786, 2021.

[18] S. Yu *et al.*, "Compute-in-memory chips for deep learning: Recent trends and prospects," *IEEE circuits and systems magazine*, vol. 21, no. 3, pp. 31–56, 2021.

[19] T. Burd, Venkataraman *et al.*, "2.2 "Zen 4c": The AMD 5nm Area-Optimized$\times$ 86-64 Microprocessor Core," in *ISSCC*, vol. 67, 2024, pp. 38–40.

[20] J. Zhang *et al.*, "A machine-learning classifier implemented in a standard 6T SRAM array," in *2016 ieee symposium on vlsi circuits (vlsi-circuits)*, 2016, pp. 1–2.

[21] M. Ali *et al.*, "A 35.5-127.2 TOPS/W Dynamic Sparsity-Aware Reconfigurable-Precision Compute-in-Memory SRAM Macro for Machine Learning," *IEEE Solid-State Circuits Letters*, vol. 4, pp. 129–132, 2021.

[22] V. Sharma *et al.*, "A 64 Kb Reconfigurable Full-Precision Digital ReRAM-Based Compute-In-Memory for Artificial Intelligence Applications," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 69, no. 8, pp. 3284–3296, 2022.

[23] T. Xiong *et al.*, "Design Methodology towards High-Precision SRAM based Computation-in-Memory for AI Edge Devices," in *2021 18th International SoC Design Conference (ISOCC)*, 2021, pp. 195–196.

[24] W.-S. Khwa *et al.*, "A 65nm 4Kb algorithm-dependent computing-in-memory SRAM unit-macro with 2.3 ns and 55.8 TOPS/W fully parallel product-sum operation for binary DNN edge processors," in *ISSCC*, 2018, pp. 496–498.

[25] A. Biswas and A. P. Chandrakasan, "Conv-RAM: An energy-efficient SRAM with embedded convolution computation for low-power CNN-based machine learning applications," in *ISSCC*, 2018, pp. 488–490.

[26] X. Si *et al.*, "A twin-8T SRAM computation-in-memory unit-macro for multibit CNN-based AI edge processors," *IEEE Journal of Solid-State Circuits*, vol. 55, no. 1, pp. 189–202, 2019.