Computer Fraud and Security
ISSN (online): 1873-7056

An Energy-Efficient SRAM Compute-In-Memory Macro for Accelerating Large-Scale Image Binary Template Matching in Holographic Counterpart Integration via Matrix-Matrix Dot Product and Summation

Shangwei Xie[1,2*], Yi Zhan[1], Shushan Qiao[1,2]
[1]Institute of Microelectronics of the Chinese Academy of Sciences Beijing, 100029, China,
[2]University of Chinese Academy of Sciences, Beijing, 100049, China
Corresponding E-mail: xieshangwei@ime.ac.cn

Abstract: In consumer electronics, Holographic Counterpart Integration (HCI) is used to capture holographic images. The features from these images can be extracted using binary template matching techniques. Compute-In-Memory (CIM) technology as a crucial hardware acceleration technique, can significantly enhance the energy efficiency of the above operations. A hybrid 8T SRAM CIM architecture based on Matrix-Matrix Dot Product and Summation (MM-DPS) operations is proposed in this paper. The architecture decouples ADC usage from array capacity, effectively avoiding the increase in power consumption as the array width grows. Additionally, a CIM array design with multi-level bitlines and two-stage accumulation is introduced, along with a low-power RC-BSC ADC. The RC-BSC ADC ensures robustness against PVT variations and significantly reduces ADC usage. Implemented using 55nm CMOS technology, the proposed CIM macro achieves a throughput of 7489 GOPS and an energy efficiency of 17769.46 TOPS/W.

Keywords: Holographic counterpart integration, consumer electronics, computing in memory, binary template matching

1 Introduction

The consumer electronics industry is witnessing rapid evolution, fueled by an escalating demand for immersive and intelligent devices. Holographic Counterpart Integration has emerged as a revolutionary technology in this space. It enables the generation and manipulation of holographic images, providing smartphones, VR/AR devices, and other gadgets with stunning 3D visuals. This not only creates an interactive experience that was previously unattainable but also ushers in new possibilities for user engagement. Functionally, Holographic Counterpart Integration captures the complete light - field data, encompassing details such as object depth and texture. To fully exploit this rich data source, binary image template matching plays a crucial role. This technique simplifies the complex holographic data, extracting essential features like edges and shapes, which are fundamental for object recognition tasks within the holographic context.

Nevertheless, processing holographic images through template matching is computationally demanding. This is where Compute - In - Memory (CIM) technology proves invaluable. As a powerful hardware accelerator, CIM mitigates data transfer bottlenecks by bringing computational operations closer to the memory. This not only significantly enhances the speed of template matching but also improves energy efficiency, making it an indispensable solution for power - conscious consumer electronics. In the subsequent sections, we will delve deeper into the integration of these technologies and explore their far - reaching impact.Various types of memory can support CIM, with Static Random-Access Memory (SRAM) emerging as one of the ideal choices due to its maturity, wide commercialization, and fast data access speeds compared to non-volatile memories like RRAM. Therefore, designing a high-efficiency SRAM-based CIM chip for specific applications is essential.

The schematic diagram of the traditional CIM macro architecture, primarily designed for vector-matrix multiply-accumulate (VM-MAC) operations in neural networks, is shown in Fig. 1(a). VM-MAC generates highly parallel outputs, with the number of output results equal to the array width M. Each column output requires data conversion and additional components such as reference voltage generators, which increase power consumption and area overhead. Moreover, the input-output scale limitations of traditional neural network layers restrict the use of large CIM arrays, hindering performance and energy efficiency improvements.

To address these issues, Matrix-Matrix Dot Product and Summation (MM-DPS), where two matrices are multiplied element-wise and then summed, can serve as a potential computational paradigm for CIM, as illustrated in Fig. 1(f). Under MM-DPS, the CIM array has only one output requiring ADC conversion, significantly reducing the number of required ADCs. MM-DPS is widely applicable in binary template matching (BTM), binary filtering, error correction codes, and especially beneficial for large-scale image binary template matching tasks[1-3], where larger CIM arrays can be utilized without stringent size constraints, enhancing energy efficiency.

However, applying CIM to accelerate MM-DPS poses challenges. First, MM-DPS may require more input wordline drivers than VM-MAC, which is undesirable. Second, traditional CIM architectures are geared towards vector-matrix computations in neural networks and need modification to accommodate MM-DPS.

Furthermore, previous studies [4-6] have indicated that ADC power consumption significantly contributes to overall system power. To reduce ADC usage, Kim et al. [7] proposed a capacitive CIM circuit that enhances energy efficiency through compact, low-power ADCs and programmable on-chip reference voltage generators. However, this primarily reduces the ADC's own power consumption without decreasing the frequency of ADC usage per column. Yu et al. [8] designed a binary search-based ADC to efficiently reduce ADC usage while using bitcells for reference voltage and calibration. Yet, this design cascades reference and calibration bitcells with the original array cells, resulting in longer bitlines and additional delays, area, and power overhead. Jeong et al. [9] proposed an ADC-free analog CIM processor but sacrificed noise resilience and susceptibility to PVT variations. In contrast, Kim et al. [10] adopted a digital CIM approach that eliminates the ADC and uses digital circuits to enhance system noise immunity, though this inherently falls under near-memory computing methods, requiring multi-level adders and increasing memory access overhead and computational delays. Xue et al. [11] replaced ADCs with Time-to-Digital Converters (TDCs), leading to reduced anti-interference capabilities.

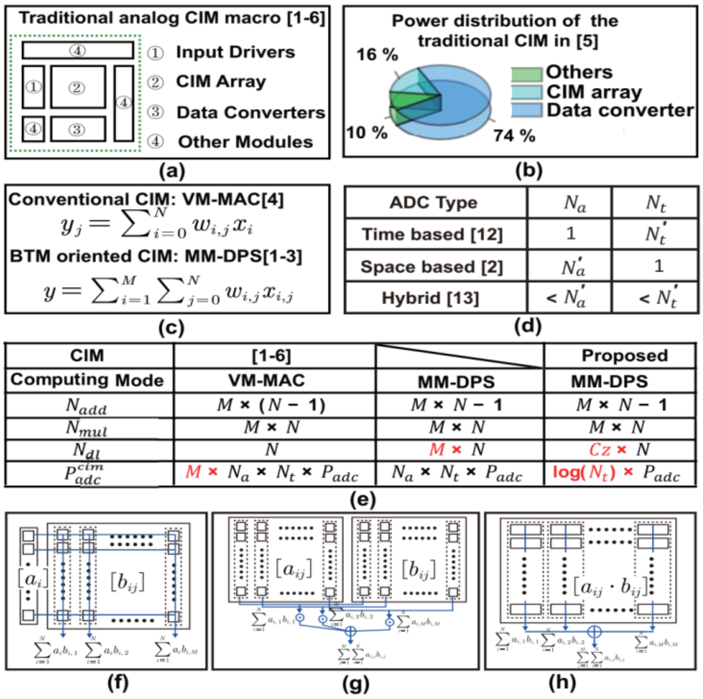
To address these issues, this paper proposes a high-energy-efficiency MM-DPS based SRAM CIM macro designed for large-scale image binary template matching with fewer ADCs. The specific contributions are:

1) A Skewer-Style CIM architecture supporting MM-DPS computation mode is designed. Unlike traditional VM-MAC-based solutions, this architecture decouples ADC usage from array capacity, ensuring ADC frequency does not increase with array width.

2) A CIM array utilizing multi-level bit lines and a two-level accumulation method is proposed, along with a hybrid 8T bitcell that supports 1-bit weights and signed inputs for efficient MM-DPS hardware implementation.

3) A low-power ADC based on replicated columns and binary search comparisons is developed, effectively reducing ADC usage per column while maintaining robustness against PVT variations during the comparison process.

2 Proposed CIM


Traditional analog CIM macro [1-6]
① Input Drivers
② CIM Array
③ Data Converters
④ Other Modules
①
②
③
④
(a)
Power distribution of the traditional CIM in [5]
16 %
10 %
74 %
Others
CIM array
Data converter
(b)
(c)
(d)
(e)
$[a_i]$
$[b_{ij}]$
$\sum_{i=1}^{N} a_i b_{i,1}$
$\sum_{i=1}^{N} a_i b_{i,2}$
$\sum_{i=1}^{N} a_i b_{i,M}$
(f)
$[a_{ij}]$
$[b_{ij}]$
$\sum_{i=1}^{N} a_{i,1} b_{i,1}$
$\sum_{i=1}^{N} a_{i,2} b_{i,2}$
$\sum_{i=1}^{N} a_{i,M} b_{i,M}$
$\sum_{j=1}^{M}\sum_{i=1}^{N} a_{i,j} b_{i,j}$
(g)
$[a_{ij} \cdot b_{ij}]$
$\sum_{i=1}^{N} a_{i,1} b_{i,1}$
$\sum_{i=1}^{N} a_{i,2} b_{i,2}$
$\sum_{i=1}^{N} a_{i,M} b_{i,M}$
$\sum_{j=1}^{M}\sum_{i=1}^{N} a_{i,j} b_{i,j}$
(h)

Conventional CIM: VM-MAC[4]

$$y_j = \sum_{i=0}^{N} w_{i,j} x_i$$

BTM oriented CIM: MM-DPS[1-3]

$$y = \sum_{i=1}^{M} \sum_{j=0}^{N} w_{i,j} x_{i,j}$$

ADC Type	N_a	N_t
Time based [12]	1	N_t'
Space based [2]	N_a'	1
Hybrid [13]	$< N_a'$	$< N_t'$

CIM	[1-6]		Proposed
Computing Mode	VM-MAC	MM-DPS	MM-DPS
N_{add}	$M \times (N-1)$	$M \times N - 1$	$M \times N - 1$
N_{mul}	$M \times N$	$M \times N$	$M \times N$
N_{dl}	N	$M \times N$	$Cz \times N$
P_{adc}^{cim}	$M \times N_a \times N_t \times P_{adc}$	$N_a \times N_t \times P_{adc}$	$\log(N_t) \times P_{adc}$

Figure 1: (a) The traditional CIM macro. (b) Power distribution of the traditional CIM in [7]. (c) Two computing

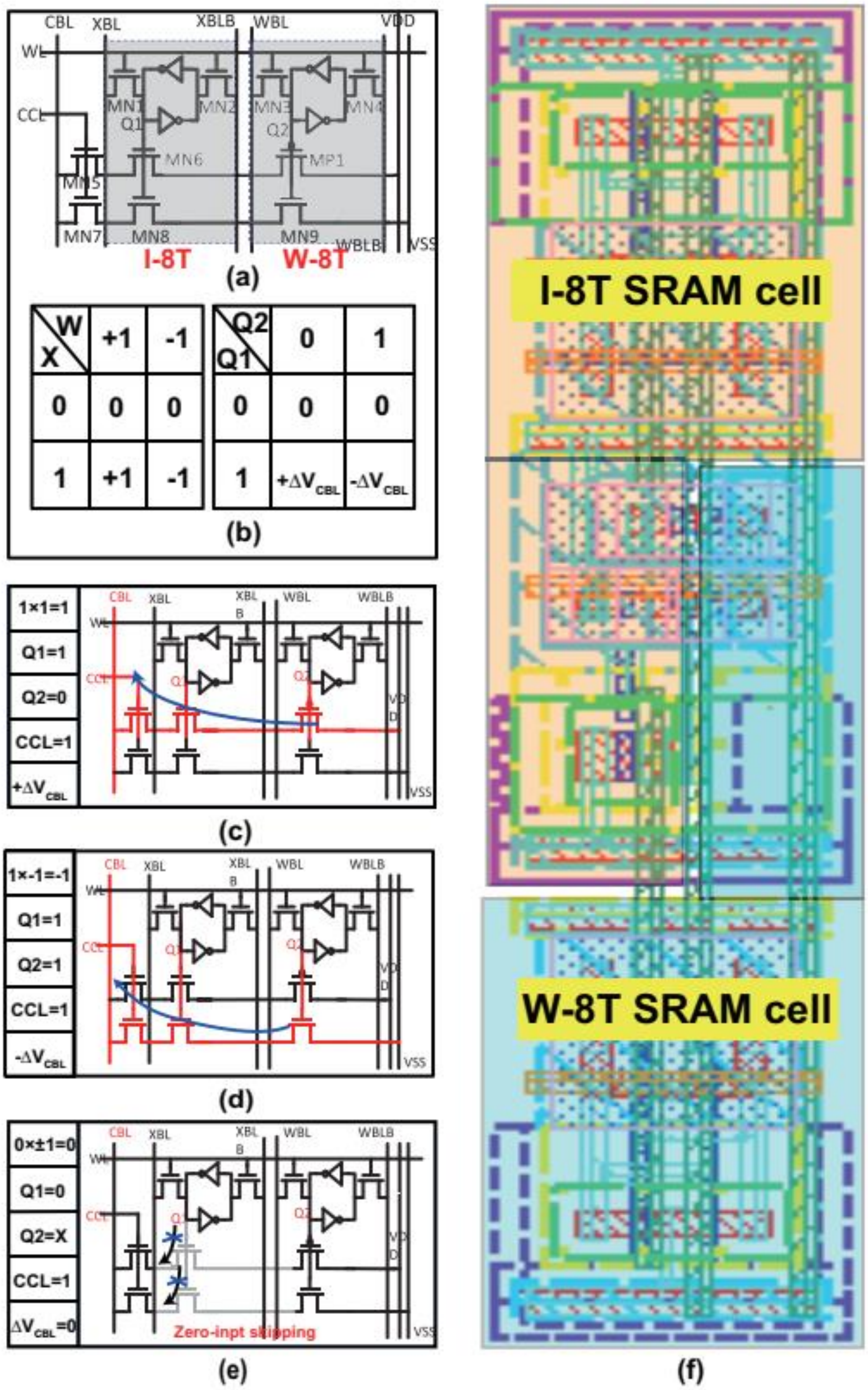
modes. (d) Parameter selection of different ADC types. (e) The comparison table of CIM schemes based on different Computing modes. (f) The architecture for Vector-Matrix Multiply-Accumulate (VM-MAC) operations.(g) The architecture for Matrix-Matrix Dot Product and Summation (MM-DPS) operations. (h) The Skewer-Style architecture proposed in this paper specifically targets MM-DPS operations

2.1 Analysis of Computing modes

Fig. 1(c) illustrates the differences between MM-DPS and VM-MAC in terms of their formulas. Fig. 1(f) and (g) highlight the architectural differences between the two modes. Fig. 1(e) shows the differences in the number of additions (N_{add}), multiplications (N_{mul}), the scale of SRAM input wordline drivers (N_{wl}), and ADC power consumption (P_{adc}^{cim}) required per CIM operation, given the height and width of the CIM array. It can be seen that while the computational load within the array is similar for both modes, there are significant differences in the power consumption of the input wordline drivers and ADCs. Assuming each column output of the CIM array contains ADCs, and each column performs executions, Fig. 1(d) shows the values of and corresponding to different types of ADCs. In traditional VM-MAC mode, the total ADC executions required are $M \times N_a \times N_t$, whereas in DPS mode, only ADC executions are needed, independent of the array width M. This means that even as the array width expands, the number of ADC executions remains stable, effectively controlling the growth in ADC power consumption. However, MM-DPS requires a larger input wordline driver scale of $M \times N$, leading to increased input power. The need for independent wordline drivers for each cell poses significant design challenges. To balance reduced ADC executions and manageable input power, we propose a CIM architecture for DPS with input drivers ($C_z \ll M$) and binary search for ADC usage.

2.2 Proposed Hybrid 8T Bitcells

To achieve 1-bit multiplication in MM-DPS, an hybrid 8T bitcell structure based on charge sharing is proposed, as shown in Fig. 2(a). Compared to the traditional bitcell that only stores weights, an additional SRAM cell is included to temporarily store the input bit. For the 1-bit signed weights and 1-bit unsigned inputs, different structures of 8T SRAM cells are employed respectively. The weight bitcell uses a W-8T structure, which consists of a 6T SRAM cell and two NMOS transistors. The input bitcell adopts an I-8T structure, containing a 6T SRAM cell, one NMOS transistor, and one PMOS transistor. Unlike the sandwich structure integrating inputs and weights at the RAM level [12], this paper integrates inputs and weights at the bitcell level. It helps to reduces the number of input driving wires from to during layout design. In this design, and $C_z = 2$, reducing the input driving wires by 128 times. primarily comes from the write word line (WWL) and compute control line (CCL). Fig. 2(b) presents the truth table for 1b input and 1b weight multiplication along with the corresponding truth table for SRAM inner points. Figs. 2(c), (d), and (e) illustrate the conduction paths for pull-down (discharge), pull-up (charge), and zero-crossing ($0 \times 1 = 0$) scenarios, respectively, demonstrating the accurate execution of 1-bit multiplication. When performing $1 \times 1 = 1$, the internal nodes of the bitcell are Q1 = 1 and Q2 = 0, with the computation control line (CCL) set to 1. In this case, the pull-up (charge) path conducts, charging the parasitic capacitance on the bitline, thereby increasing the bitline voltage by ΔV_{CBL}. For $1 \times -1 = -1$, the internal nodes are Q1 = 1 and Q2 = 1, with CCL also set to 1. Here, the pull-down (discharge) path conducts, discharging the parasitic capacitance on the bitline, causing the bitline voltage to decrease by ΔV_{CBL}. If an input of 0 is encountered, the corresponding bitcell's internal



X \ W	+1	-1
0	0	0
1	+1	-1

Q1 \ Q2	0	1
0	0	0
1	$+\Delta V_{CBL}$	$-\Delta V_{CBL}$

Figure 2: (a) The inner structure of the proposed hybrid 8T CIM bitcell. (b) The detailed operand table of the proposed bitcell. (c) The conduction of the pull-down path. (d) The conduction of the pull-up path. (e) The input zero-skipping operation. (f) The layout of the proposed bitcell.

node Q1 is set to 0, regardless of the state of Q2. Both charge and discharge paths are turned off, keeping the bitline voltage in a hold state, thus implementing a zero-skipping operation. Fig. 2(f) shows the layout of the hybrid 8T bitcells.

2.3 Overall CIM Structure

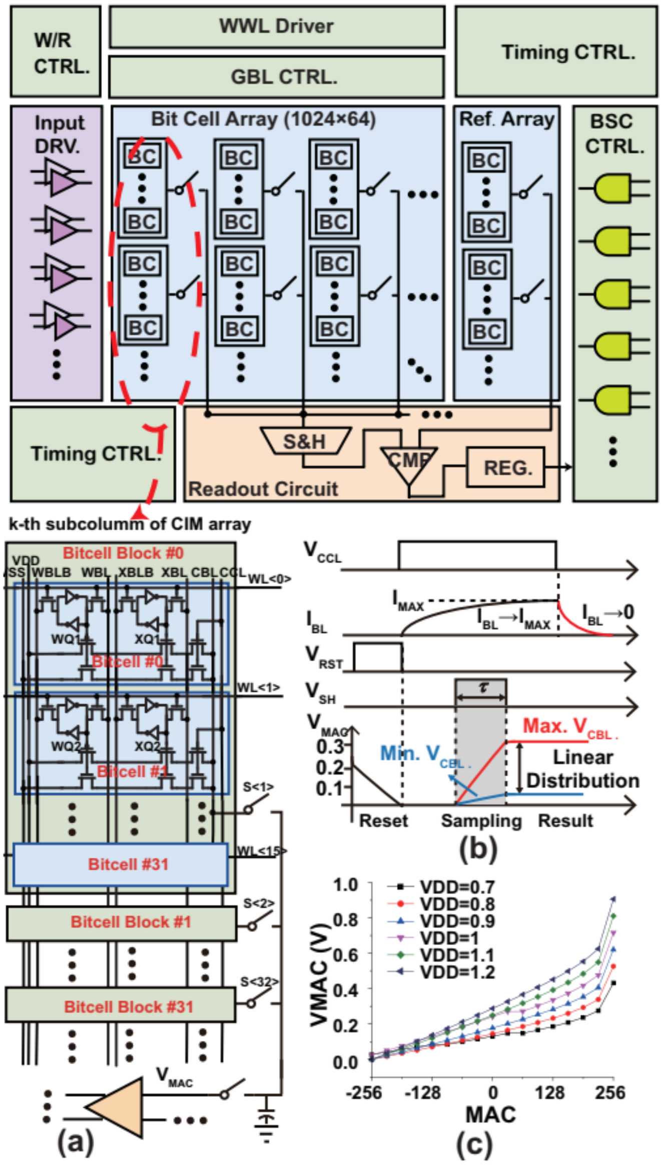


Figure 3: (a) The overall archiecture of the proposed CIM macro and the k-th subcollum of the CIM array. (b) the complete time sequece of the cim opration (c) the relationship between MAC and VMAC with varied VDD.

Fig. 3 illustrates the proposed CIM architecture designed for MM-DPS operations. This architecture primarily includes the input driver, a 1024x256 array, a timing controller, a global bitline controller, weight write word line drivers, readout circuits, reference bitcell replica columns, binary search logic, and more. The readout circuits consist of output sample-and-hold circuits, ADCs, shift registers, among other components. As shown in Fig. 3(a), for the k-th column of the CIM array, it is divided into 32 segments, each connected to the global bitline via switches. Within each segment, 32 cells share a single bitline. The two-level bitline design effectively reduces bitline parasitic capacitance, facilitating the expansion of the array height. Similar to CIM schemes based on charge sharing or current [9, 13], this design leverages parasitic capacitance charging and discharging. Consequently, the bitline voltage is rapidly clamped to a fixed potential, ensuring stable operation. To cope with subsequent multiple comparisons, a sample-and-hold circuit is added at the array output to substitute the array. The specific timing process is shown in Fig. 3(c). After output sampling and holding, the global bitline controller promptly deactivates the bitline switches, which helps the current on the local and global bitlines drop rapidly, effectively reducing the power consumption of the array. Fig.3(d) shows the relationship between the theoretical MAC value and the array output voltage under different supply voltages, indicating that increasing the supply voltage helps improve linearity.

2.4 Low Power ADC with the Replica Collum and Binary Search Comparison (RC-BSC ADC)

Traditional ADCs can be categorized into time-based[14], spatial-based[4], or spatio-temporal pipeline[15] types. ADCs utilizing Binary Search Comparison (BSC) have become popular due to their low power and fast comparison times. Yu et al. [8] first applied BSC ADCs in a CIM architecture, shown in Fig. 5. Here, reference bitcells (RBCs) are attached to bitlines as ADC reference voltage sources, and offset bitcells (OBCs) are included for correction. and denote the number of CIM bitcells (CBCs) and the original bitline length, respectively, while and represent the total bitcells and bitline length with RBCs and OBCs added. and account for the extra bitline lengths from OBCs and RBCs, and is the length from the array end to the ADC. The row word lines use binary search to control RBCs for reference voltages. However, this design has drawbacks. It requires two RBC groups for positive "+1" and negative "-1" weights, each with fixed outputs. RBCs and OBCs cascade with CBCs, increasing bitline length by over 37.5%, thus raising parasitic capacitance and affecting accumulation operations. Different numbers of RBCs and CBCs impact parasitic capacitance contributions, and RBC switching logic affects the entire bitline’s capacitance. Consequently, the RBC reference voltage and CBC voltage differ in precision and are variably influenced by PVT (Process, Voltage, Temperature) variations, causing unreliable ADC outputs.

To address these issues, we propose an ADC architecture based on Replica Columns (RC) and Binary Search Comparison (BSC), termed RC-BSC ADC, shown in Fig. 5(a). The ADC comprises a comparator, an output register, a BSC controller, replica columns, and wordline drivers. The RC-BSC ADC divides the comparison process into five stages, with the replica columns segmented accordingly. Each stage sets a segment of the replica columns to generate the reference voltage. A shift-bit register stores previous comparison results, and a timer sets the next segment to all "-1" or "+1" weights.

Compared to [8], the proposed ADC architecture has three main advantages. Firstly, the replica columns operate in parallel with the CIM array, maintaining the original CBL (column bitline) length. Secondly, the number of RBCs in the replica columns matches the CBCs in the CIM, and proper layout ensures similar bitline lengths, making the reference and MAC voltages similarly affected by PVT variations. Lastly, each weight bit cell in the replica columns can switch between "-1" and "+1," eliminating the need for two separate weight sets. The binary-search ADC archiecture can be seen in Figure 4.

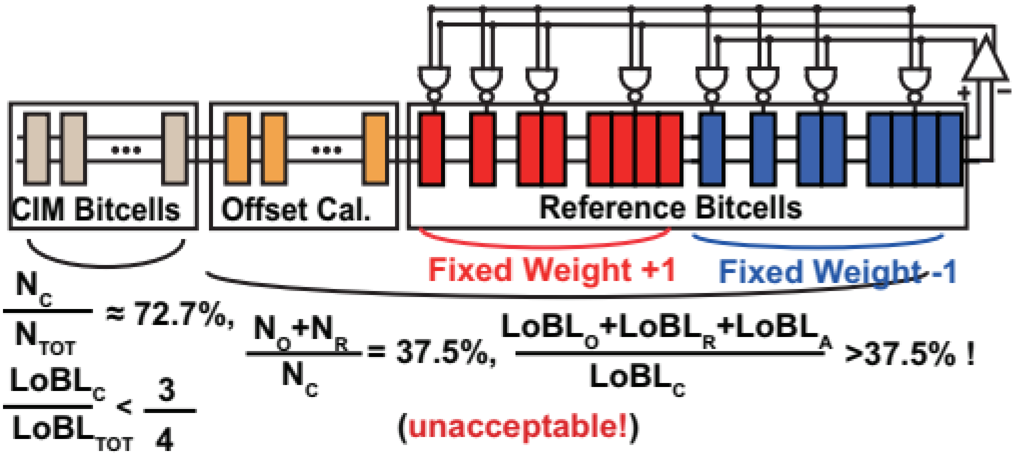

CIM Bitcells
Offset Cal.
Reference Bitcells
+
−
Fixed Weight +1
Fixed Weight -1
$\frac{N_C}{N_{TOT}} \approx 72.7\%,$
$\frac{N_O+N_R}{N_C} = 37.5\%,$
$\frac{LoBL_O+LoBL_R+LoBL_A}{LoBL_C} > 37.5\%$!
$\frac{LoBL_C}{LoBL_{TOT}} < \frac{3}{4}$
(unacceptable!)

Figure 4: The binary-search ADC archiecture in [8]

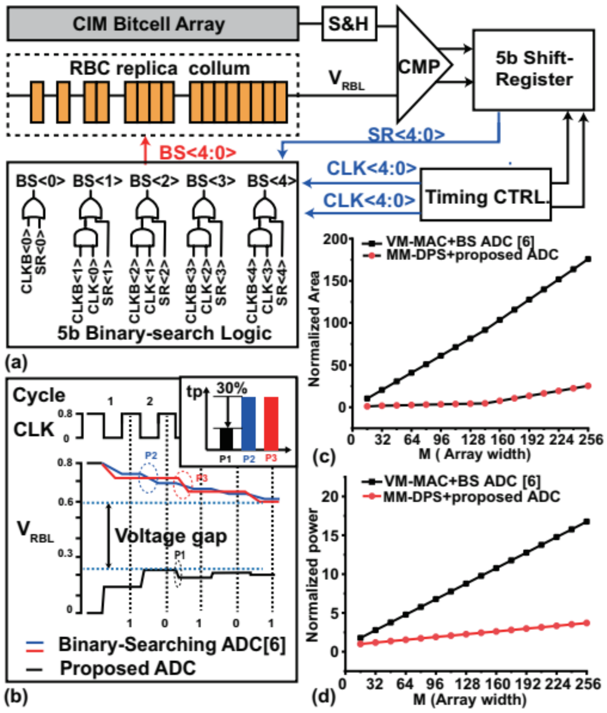


Figure 5: (a) The proposed RC-BSC ADC archietecture. (b) the time sequece comprision of the ADC. (c) Normalized area comparision with varied array width. (d) Normalized power comparision with varied array width.

Fig. 5(b) shows the differences between the RC-BSC ADC and the BS ADC in [8]. Our approach allows comparisons at lower voltages, reducing power consumption. The design in [8] requires higher pre-charge voltages, leading to larger voltage differences. Shorter bitlines and reduced parasitic capacitance in our design reduce transition time for each reference voltage by 30%. Figs. 5(c) and (d) show that, when scaling the array to boost computing capacity, the proposed MM-DPS mode with RC-BSC ADC exhibits slower growth in ADC power consumption and area compared to the traditional VM-MAC and BS ADC combination [8].

3 Result

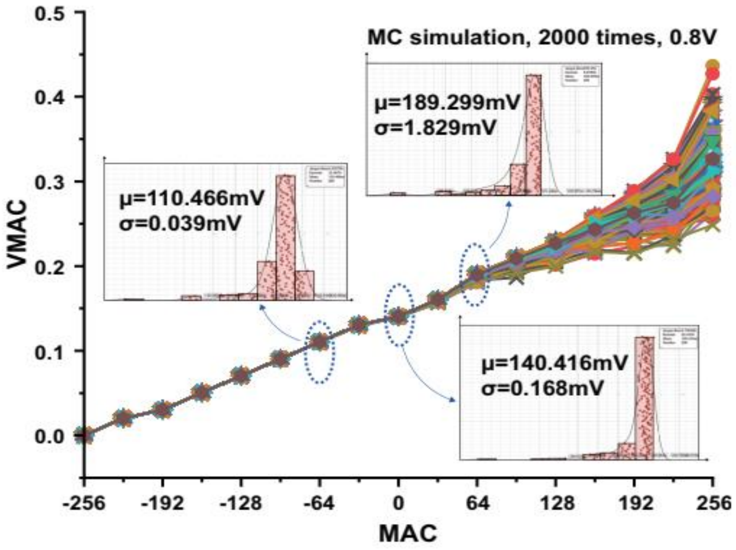


Figure 6: MC simulation of the proposed CIM macro.

Fig.6 presents the results of 2000 Monte Carlo simulations performed on the MM-DPS operation in the proposed CIM scheme. The linearity is good when the MAC value is less than 64, and the curve becomes more erratic after surpassing 64. On the other hand, based on the research results in [4] and our practical simulation tests, the probability distribution of the MAC operation outcomes in an MLP model under the MNIST dataset is primarily concentrated within the range [-64, +64]. This conveniently avoids the more volatile interval (64, 256]. When the MAC value is -64, the mean is 110.466 mV, and the standard deviation is 0.039 mV. When the MAC value is 0, the mean is 140.416 mV, and the standard deviation is 0.168 mV. When the MAC value is +64, the mean is 189.299 mV, and the standard deviation is 1.829 mV. The standard deviations at all three points are much smaller than the ADC comparison interval, which is mV in our design. Therefore, the proposed CIM design exhibits good linearity in the range [-64, +64] and is less affected by random mismatches and process variations.

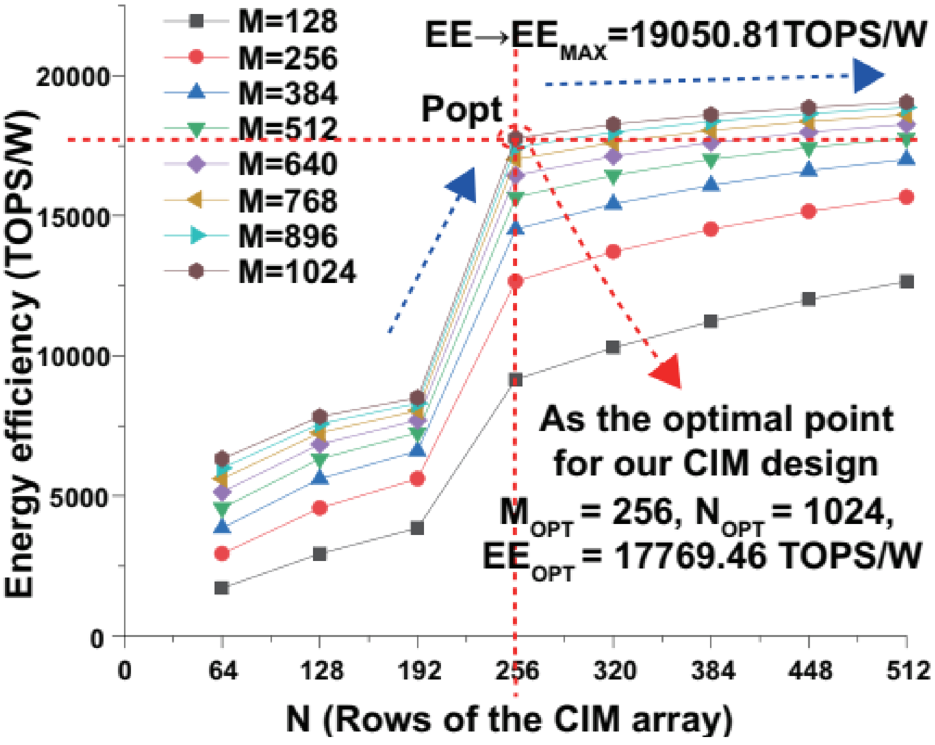
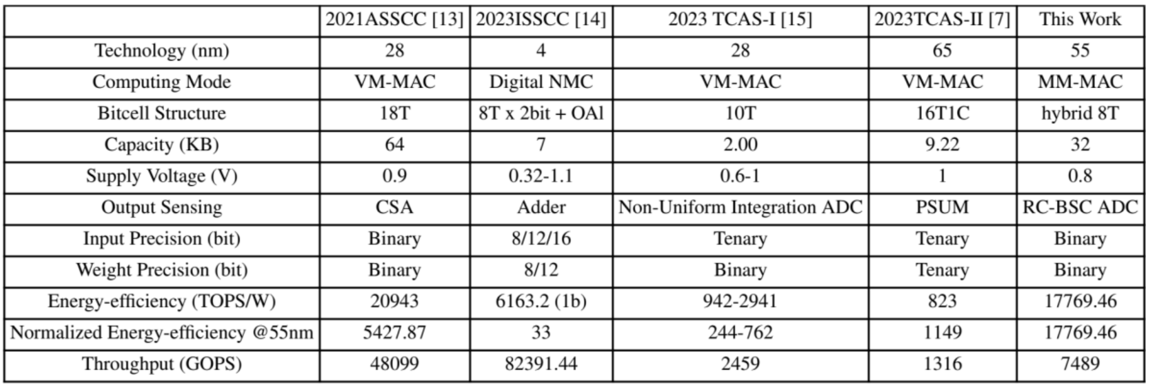


Figure 7: The relationship between energy efficiency of the CIM macro and the array size.

Fig.7 demonstrates the energy efficiency of the CIM macro based on the MM-CIM design for different array sizes. In the simulations, considering the actual size of conventional SRAM arrays, the array width is taken up to 512 and the length up to 1024. As M and N increase, the energy efficiency also significantly increases. If M is kept constant, as N changes from 192 to 256, the energy efficiency grows rapidly. After N exceeds 256, the growth gradually flattens; even if N changes from 256 to 512, doubling the area, the energy efficiency only increases by 7.21%, reaching 19050.81 TOPS/W. Additionally, considering the layout and routing requirements, the optimal array size is determined by the inflection point P_{opt}in the figure, which corresponds to N = 256 and M = 1024, achieving an energy efficiency of 17769.46 TOPS/W.

Table 1 is a comparison table with previous CIM and near-memory computing (NMC) designs. The highest energy efficiency design for CIM currently comes from [16] achieved 20943 TOPS/W using VM-MAC at 28nm. [17] realized 6162.2 TOPS/W using digital NMC at 4nm. This study achieved 17769.46 TOPS/W using MM-DPS at 55nm. For ease of comparison, the energy efficiencies of[16], [17] and [18] are scaled down to 55nm according to the square of the technology node ratio. It is found that this study’s energy efficiency is more than three times that of [16].

Table 1: Performance comparison between the proposed CIM and existing solutions



	2021ASSCC [13]	2023ISSCC [14]	2023 TCAS-I [15]	2023TCAS-II [7]	This Work
Technology (nm)	28	4	28	65	55
Computing Mode	VM-MAC	Digital NMC	VM-MAC	VM-MAC	MM-MAC
Bitcell Structure	18T	8T x 2bit + OAI	10T	16T1C	hybrid 8T
Capacity (KB)	64	7	2.00	9.22	32
Supply Voltage (V)	0.9	0.32-1.1	0.6-1	1	0.8
Output Sensing	CSA	Adder	Non-Uniform Integration ADC	PSUM	RC-BSC ADC
Input Precision (bit)	Binary	8/12/16	Tenary	Tenary	Binary
Weight Precision (bit)	Binary	8/12	Binary	Tenary	Binary
Energy-efficiency (TOPS/W)	20943	6163.2 (1b)	942-2941	823	17769.46
Normalized Energy-efficiency @55nm	5427.87	33	244-762	1149	17769.46
Throughput (GOPS)	48099	82391.44	2459	1316	7489

4 Conclusion

In this brief, we introduce a 32KB SRAM-CIM macro designed to accelerate large-scale image binary template matching for holographic counterpart integration. Implemented using 55-nm technology and supporting Matrix-Matrix Dot Product and Summation (MM-DPS) computing mode, this macro is tailored for efficient processing of complex image data. We have developed a novel ADC architecture, the RC-BSC ADC, which utilizes binary search and replicated columns. When integrated with the MM-DPS mode, the RC-BSC ADC architecture drastically reduces the total number of ADC usages from under the VM-MAC mode to merely $log(N_t)$. Additionally, this architecture improves the resilience of computational outcomes against PVT variations. Operating at a voltage of 0.8V, this macro achieves a throughput of 7489 GOPS, and an energy efficiency of 17769.46 TOPS/W with 1-bit weights and 1-bit inputs. This work presents a significant leap forward in integrating advanced CIM architectures with holographic technologies, paving the way for more efficient and powerful image processing solutions in consumer electronics.

Data sharing agreement

The datasets used and analyzed during the current study are available from the corresponding author on reasonable request.

Declaration of Conflicting Interests

The authors declared no potential conflicts of interest with respect to the research, author-ship, and publication of this article.

Funding

The authors received no financial support for the research.

References

[1] M. Atallah, “Faster image template matching in the sum of the absolute value of differences measure,” vol. 10, no. 4, pp. 659–663.

[2] F. Tang and H. Tao, “Fast Multi-scale Template Matching Using Binary Features,” in *2007 IEEE Workshop on Applications of Computer Vision (WACV ’07)*. IEEE, pp. 36–36.

[3] H. Yang, C. Huang, F. Wang *et al.*, “Robust Semantic Template Matching Using a Superpixel Region Binary Descriptor,” vol. 28, no. 6, pp. 3061–3074.

[4] S. Yin, Z. Jiang, J.-S. Seo *et al.*, “XNOR-SRAM: In-Memory Computing SRAM Macro for Binary/Ternary Deep Neural Networks,” *IEEE Journal of Solid-State Circuits*, pp. 1–11, 2020.

[5] L. Lu and D. A. Tuan, “A 47 TOPS/W 10T SRAM-based Multi-Bit Signed CIM with Self-Adaptive Bias Voltage Generator for Edge Computing Applications,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, pp. 1–1, 2023.

[6] Y. Shu, H. Zhang, Q. Deng *et al.*, “CIMC: A 603TOPS/W In-Memory-Computing C3T Macro with Boolean/Convolutional Operation for Cryogenic Computing,” in *2023 IEEE Custom Integrated Circuits Conference (CICC)*. IEEE, pp. 1–2.

[7] E. Kim, H. Oh, N. Kang *et al.*, "A Capacitive Computing-In-Memory Circuit with Low Input Loading SRAM Bitcell and Adjustable ADC Input Range," *IEEE Transactions on Circuits and Systems II: Express Briefs*, pp. 1–1, 2023.

[8] C. Yu, K. T. C. Chai, T. T.-H. Kim *et al.*, "A Zero-Skipping Reconfigurable SRAM In-Memory Computing Macro with Binary-Searching ADC," in *ESSDERC 2021 - IEEE 51st European Solid-State Device Research Conference (ESSDERC)*. Grenoble, France: IEEE, Sep. 2021, pp. 131–134.

[9] H. Jeong, S. Kim, K. Park *et al.*, "A Ternary Neural Network Computing-in-Memory Processor With 16T1C Bitcell Architecture," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 70, no. 5, pp. 1739–1743, May 2023.

[10] H. Kim, J. Mu, C. Yu *et al.*, "A 1-16b Reconfigurable 80Kb 7T SRAM-Based Digital Near-Memory Computing Macro for Processing Neural Networks," vol. 70, no. 4, pp. 1580–1590. [Online]. Available: https://ieeexplore.ieee.org/document/10012044/

[11] C. Xue, X. Qiao, X. Ren *et al.*, "A 768.7-2124.2 TOPS/W Time-Domain Computing-in-Memory Macro With Low Static Leakage and PrecisionConfigurable TDC," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 71, no. 5, pp. 2789–2793, May 2024.

[12] J. Yang, Y. Kong, Z. Wang *et al.*, "24.4 Sandwich-RAM: An Energy-Efficient In-Memory BWN Architecture with Pulse-Width Modulation," in *2019 IEEE International Solid- State Circuits Conference - (ISSCC)*. San Francisco, CA, USA: IEEE, Feb. 2019, pp. 394–396.

[13] Z. Jiang *et al.*, "C3SRAM: An In-Memory-Computing SRAM Macro Based on Robust Capacitive Coupling Computing Mechanism," *IEEE Journal of Solid-State Circuits*, vol. 55, no. 7, pp. 1888–1897, Jul. 2020.

[14] Y.-J. Jo, B. P. Yap, D.-H. Yoon *et al.*, "DenseCIM: Binary Weighted-Capacitor SRAM Computation-In-Memory with Column-by-Column Dynamic Range Calibration SAR ADC," in *2023 IEEE Custom Integrated Circuits Conference (CICC)*.

[15] M. Brandolini, Y. J. Shin, K. Raviprakash *et al.*, "A 5 GS/s 150 mW 10 b SHA-Less Pipelined/SAR Hybrid ADC for Direct-Sampling Systems in 28 nm CMOS," *IEEE Journal of Solid-State Circuits*, vol. 50, no. 12, pp. 2922–2934, Dec. 2015.

[16] C.-S. Lin, F.-C. Tsai, J.-W. Su *et al.*, "A 48 TOPS and 20943 TOPS/W 512kb Computation-in-SRAM Macro for Highly Reconfigurable Ternary CNN Acceleration," in *2021 IEEE Asian Solid-State Circuits Conference (A-SSCC)*. Busan, Korea, Republic of: IEEE, Nov. 2021, pp. 1–3.

[17] H. Mori, W.-C. Zhao, C.-E. Lee *et al.*, "A 4nm 6163-TOPS/W/b, 4790-TOPS/mmˆ*{2}*/b SRAM Based Digital-Computing-in-Memory Macro Supporting Bit-Width Flexibility and Simultaneous MAC and Weight Update," in *2023 IEEE International Solid- State Circuits Conference (ISSCC)*. San Francisco, CA, USA: IEEE, Feb. 2023, pp. 132–134.

[18] S. Cheon, K. Lee, and J. Park, "A 2941-TOPS/W Charge-Domain 10T SRAM Compute-in-Memory for Ternary Neural Network," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 70, no. 5, pp. 2085–2097, May 2023.