2024 IEEE 6th International Conference on AI Circuits and Systems (AICAS) | 979-8-3503-8363-8/24/$31.00 ©2024 IEEE | DOI: 10.1109/AICAS59952.2024.10595553

A 266.7 TOPS/W Computing-in Memory Using Single-Ended 6T 4-kb SRAM in 16-nm FinFET CMOS Process

Cheng-Yao Lo*[1], Lean Karlo Santos Tolentino*†‡[1], Jhih-Ying Ke*, Jeffrey S. Walling§, Yang Yi§, and Chua-Chin Wang*¶

*Department of Electrical Engineering, National Sun Yat-Sen University, Kaohsiung, Taiwan
†Department of Electronics Engineering, Technological University of the Philippines, Manila, Philippines
‡Center for Artificial Intelligence and Nanoelectronics, Integrated Research and Training Center, Technological University of the Philippines, Manila, Philippines
§Bradley Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA, USA
¶Department of Electronics & Communications Engineering, Vel Tech University, Chennai, India
Email: ccwang@ee.nsysu.edu.tw

***Abstract*—To minimize substantial energy and transmission latency cost caused by the von Neumann bottleneck, a novel computing-in-memory (CIM) architecture utilizing a single-ended 6T 4-kb SRAM in 16-nm 1P11M FinFET process is proposed in this paper. The use of the single-ended 6T SRAM is the CIM's defining feature; the path directly connecting from the complement bitline (BLB) to Q was removed. By altering the value of Qb, write operations of 0 or 1 could be performed. Then, employing ultra-low threshold voltage (ULVT) transistors in the 2T Switch eliminates the need of a current compensation circuit and its corresponding coupled capacitor used in avoiding charge-sharing problems. Moreover, employing ULVT devices in the Control Circuits for both SRAM core array and CIM core minimizes the delay in these circuits due to the V_{th} drop in the 16-nm technology node with ultra-low supply voltage. This CIM enables the performance of operations such as addition, signed multiplication, and Boolean logical functions which are necessary for convolutional neural networks (CNN). At a maximum frequency of 1 GHz, it achieves an energy efficiency of 266.7 TOPS/W and area efficiency of 470.588 GOPS/mm^2 as evidenced by post-layout simulation results.**

***Index Terms*—computing-in-memory (CIM), convolution, FinFET, SRAM, von Neumann bottleneck.**

I. Introduction

Traditional AI and neural network applications used the von Neumann architecture, which mainly included memory and an arithmetic logic unit (ALU) for computations. However, the von Neumann bottleneck persists due to data transmission between memory and ALU, limiting timing, throughput, and energy efficiency. Several researchers have investigated computing-in-memory (CIM) architectures [1], [2] to overcome these limitations by performing calculations directly in the memory array. Unlike conventional systems, CIM eliminates data transfer between memory arrays and processors. Instead, data are retrieved from memory, modified, and stored, requiring dedicated circuitry for supplementary operations and seamless switching between calculation sources and destinations.

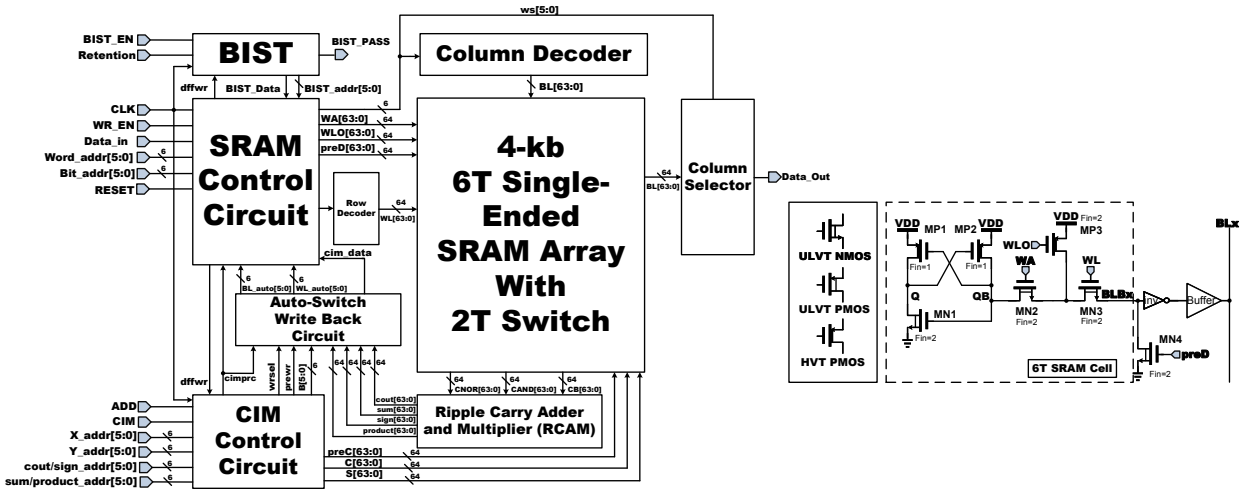
SRAMs are preferred over DRAMs in CIM devices for their faster bit-wise logical operations and data read speed, improving reliability for AI applications [1], [2]. However, SRAMs' higher power consumption and larger size are drawbacks. Resolving these issues led to innovative designs, such as the 4T load-less SRAM [3]. A disturb-free, single-ended (S.E.) 6T SRAM was introduced to implement Boolean logic and addition operations [4]. AI applications and convolution calculations require handling both negative and positive values for 1 and 0, especially during simultaneous addition and multiplication operations. Finally, a previous SRAM-based CIM was implemented in 40-nm CMOS technology [5], [6]. Its ripple carry adder (RCA) and multiplier (RCAM) block was carried out utilizing a disturb-free, S.E. 7T 1-kb SRAM. This implementation employed the full swing-gate diffusion input (FS-GDI) method, which offers reduced chip area cost, low power consumption, and full voltage swing resolution. However, it has a low energy efficiency and area efficiency at 7.66 TOPS/W and 27 GOPS/mm^2, respectively. To improve the prior CIM's energy and area efficiency, several of its blocks' circuits are modified and implemented using 16-nm FinFET CMOS process. To achieve an 8-bit input and 8-bit weight CIM, a 4-kb SRAM array is implemented in this regard.

II. System Architecture of the Proposed CIM

The CIM architecture presented in this study, shown in Fig. 1(a), showcases a single-ended 6T SRAM-based design. Notably, commonly used circuits from previous researches are omitted for discussion in this paper [6]. The unique contributions of this research are elaborated in the subsequent sections.

[1]The two principal authors made an equal contribution to this study.

★This research utilized an EDA tool provided by TSRI (Taiwan Semiconductor Research Institute). Funding for this project was extended by the NSTC (National Science and Technology Council) of Taiwan through specified grant numbers, NSTC 110-2221-E-110-063-MY2 and 112-2221-E-110-063-MY3.



(a) (b)

Fig. 1. (a) Block diagram of the proposed CIM; (b) Proposed single-ended 6T SRAM

A. Design of the Proposed CIM

The previous paper illustrates a 4×4 array prototype designed to showcase their CIM [6]. In our design, capacitors coupled to pre-charge circuits are removed as they are deemed unnecessary. The CIM's process for performing addition and multiplication is stated as follows:

1) A 2T Switch acts as a monitor to identify the rows to be used as operands for the addition operation. Any two rows within the array can be chosen as operands. Previously, a current compensation circuit was added to the 2T Switch to resolve charge-sharing problems in their CIM [6]. However, it is not needed in our proposed design since ULVTs (V_{th} = 0.1 to 0.2 V) are used instead of SVTs (standard V_{th} transistors).
2) CBi, CNORi, and CANDi, where i ranges from 0 to 3, are generated and serve as inputs linked to the respective Ripple Carry Adder and Multiplier (RCAM) unit at the base of each column.
3) The RCAM block performs the multiplication and addition operations using the provided inputs.
4) A multiplexer (MUX) takes supervision of writing back the product or sum to a designated memory address, contingent upon the reception of respective selection signals.

B. 6T SRAM unit

Referring to Fig. 1(b), a specific memory cell is selected through external input of 6-bit address signals and 6-bit data signals, decoded by row and column decoders. During read or write operations, control is managed through signals WA, WL, and WLO, which activate MN2 and MN3 or MP3, respectively, initiating the pre-discharge (preD). In read or write actions, a pre-discharge is applied to the complement bitline (BLBx) to prevent data '0' from being affected by relying on subthreshold current. When a memory cell is not selected, preD is set to high, grounding the BLBx, preventing leakage current from the SRAM cell and any adverse effects on the BLBx, thus avoiding additional power consumption.

TABLE I
READ/WRITE OPERATION MODES OF THE 6T SRAM CELL

Modes	Read 1/0	Write 1	Write 0	Hold
preD	0	1	1	1
WA	1	1	1	0
WLO	1	1	0	1
BL	1/0	1	1	1
BLB	0/1	0	0	0
WL	1	1	0	0

- Read 1 or 0: WLO is set to high, which turns off MP3. Simultaneously, WA is set to high, activating MN2, and WL is set to high, activating MN3. The voltage stored in QB is read onto the bitline (BLx) through MN2 and MN3, along with the inverter driven by a large-width buffer on the BLBx.
- Write 0: WA is set to high, turning on MN2. PreD, WL, and WLO are set to low, turning off MN3 and activating MP3. This configuration sets QB to high, enabling MN1. Q discharges to low due to the activation of MN1.
- Write 1: PreD is kept to high switching MN3 on and pulling BLB to GND. WA is set to high, turning on MN2. Both WL and WLO are set to high, turning on MN3 and turning off MP3. This arrangement sets QB to low, enabling MP1. Q charges to high as a result.

The operations described above are summarized in Table I.

C. CIM Control Circuit

As shown in Fig. 2, CIM Control Circuit comprises the CIM Timing Control Circuit and the Address Selecting Control

Circuit. Its primary function is to generate corresponding operation control signals and the addresses for the operations we intend to perform. When either the CIM or MUL signal is high, it indicates the need to commence an operation. The OP signal is then pulled to high, initiating the output of relevant operation control signals by the operation timing control circuit. If the CIM signal is logic 1, it triggers the address selection control circuit to select the corresponding addend and addend address data, allowing the memory to automatically perform addition at the specified operation addresses starting from BL0 and continuing sequentially until the CIM signal is deactivated.

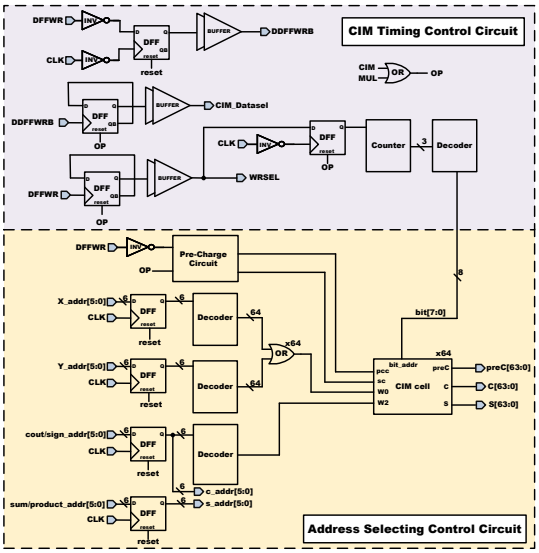


Fig. 2. CIM Control Circuit.

The address selection control circuit selects the multiplicand, multiplier, and sign bits of the multiplicand and multiplier address data when the MUL is high. Starting at BL0 and consecutively until the MUL signal is deactivated, the memory automatically performs multiplication at the given operation addresses.

An example of the addition control waveform is shown in Fig. 3. When CIM is logic 1, the Prec signal undergoes pre-charging, and subsequently, the S signal chosen by the Carry_addr address begins the operation. Dffwr represents the waveform of wr_en sampled by CLK, indicating the read/write interval. When Dffwr is high, it indicates writing, otherwise, it signifies reading.

D. Auto-Switch Write Back Circuit

Write back is crucial for mathematical operations like addition (ADD) and multiplication (MUL). The sum or product goes to the chosen address. According to Fig. 4, there are three separate blocks. The green block indicates the bit line (BL) auto-switching circuit. Data selection is handled by the second block (in blue) when the OP is high. The addition operation will use CIM_Data as the carry and sum. When MUL is initiated, product selection and output of the product

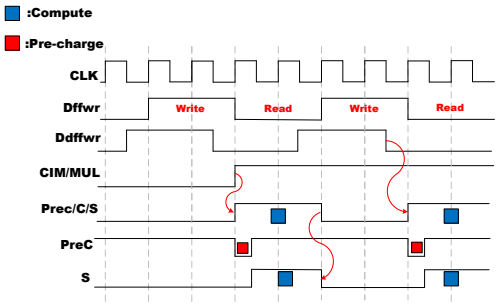


Fig. 3. Timing sequence of additions.

will be performed. Write back begins with BL0 and moves to MSB. As shown in Fig. 4, when OP is 1, the automatic write-back circuit stores computed results from BL0 into designated storage units for addition (Carry and Sum) or multiplication (Product and Signproduct). The ripple-carry addition requires WL (Word Line) switching between carry_addr and sum_addr performed by the third block (in orange) to store data at separate addresses during computation.

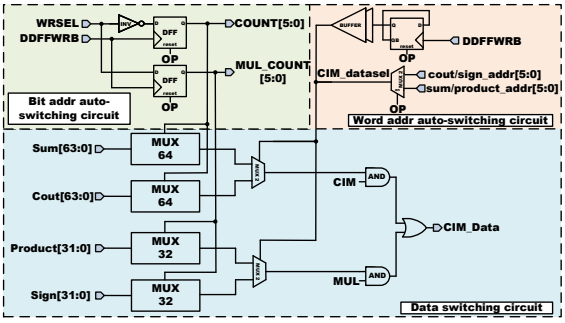


Fig. 4. Auto-switch Write Back circuit.

III. Simulation Results

The proposed CIM was designed using TSMC 16-nm 1P11M FinFET process. Its layout and floorplan are shown in Fig. 5. Its core and chip area are 0.35×0.485 mm^2 and 0.995×1.02 mm^2, respectively.

Fig. 6 and 7 shows the post-layout simulations of the proposed CIM's addition and multiplication operations, respectively. For example, at inputs X = 1110_1010 and Y = 1011_1100, we have Sum = 1010_0110 when the addition is performed. For bit-by-bit signed multiplication, when input X = 1110_1010 with its respective sign per bit, signX = 1010_1010, and weight Y = 1011_1100 with its respective sign per bit, signY = 1010_1010, we have Product = 1010_1000 and Sign = 0000_0000.

Our proposed CIM's throughput is $\frac{16}{0.2\times10^{-9}}$ = 80 GOPS = 0.08 TOPS, where the typical processing element cell's precharge value for multiplication and addition operations is 0.2×10^{-9} s/operation. In the RCAM, each set consists of 8 input bits, and there are two operations. Therefore, a total of

TABLE II
PERFORMANCE COMPARISON OF SRAM-BASED CIMS

	ISSCC [7]	ISSCC/JSSC [8]/ [9]	ISSCC/JSSC [10]/ [11]	ISSCC [12]	TVLSI [6]	JSSC [13]	ISCAS [14]	This work
Year	2018	2019	2020	2020	2021	2021	2022	2023
Process (nm)	65	55	28	7	40	28	28	16 (1P11M)
Verification	Meas.	Meas.	Meas.	Meas.	Meas.	Meas.	Meas.	Sim.
Supply Voltage (V)	1.0	1.0	0.85-1.0	0.65-1	0.9	1.2	0.9	0.8
SRAM Cell	6T	Twin 8T	6T	8T	7T (S.E.)	6T	6T + LMU	6T (S.E.)
Input Bits	8	4	8	4	4	4	8	8
Weight Bits	8	5	8	4	4	2	8	8
Array Size (kb)	128	3.8	64	4	1	65.536	64	4
Model	SVM	CNN	CNN	-	CNN	CNN	CNN	CNN
Energy Efficiency (TOPS/W)	3.125	18.37	7.6	262.3-610.5	7.66	49.4	42.1	266.7
Bitwise Energy Efficiency[1] (TOPS·bits^2/W)	200	367.4	486.4	9768	122.56	319.2	2694.4	**17068.8**
Area Efficiency (GOPS/mm^2)	N.A.	451.83	N.A.	116375	27	3400	61.337	470.588
Bitwise Area Efficiency[2] (TOPS·bits^2/mm^2)	N.A.	9.0366	N.A.	1862	0.432	27.2	3.9256	30.118

[1]Bitwise Energy Efficiency = Energy Efficiency× Input Bits × Weight Bits , [2]Bitwise Area Efficiency = Area efficiency× Input Bits × Weight Bits

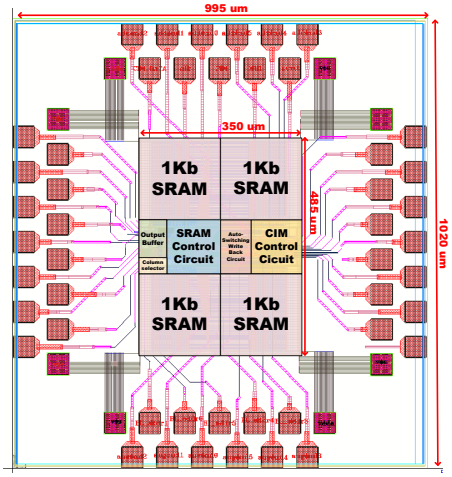


Fig. 5. Proposed 4-kb SRAM-based CIM's layout.

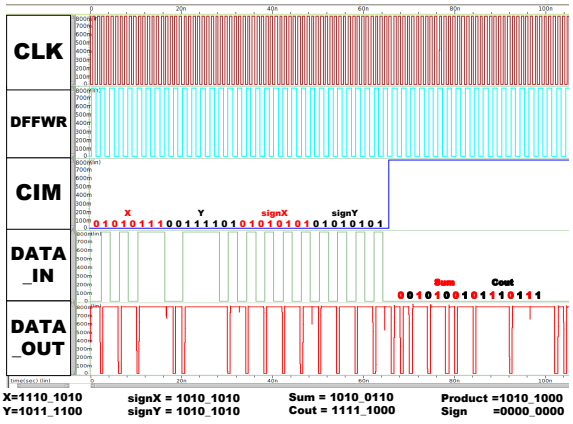


Fig. 6. Demonstrating the proposed CIM's addition operation.

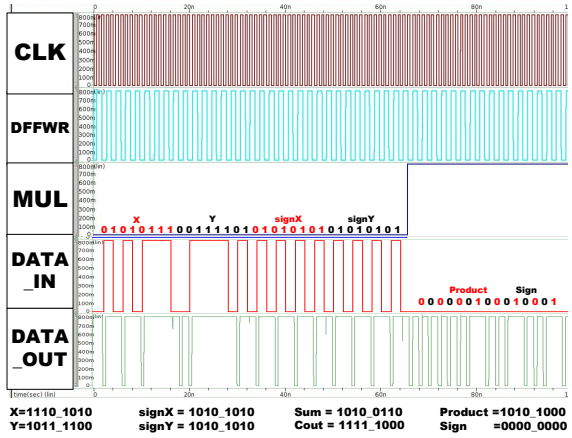


Fig. 7. Demonstrating the proposed CIM's multiplication operation.

16 sets are involved in parallel calculations simultaneously. Our CIM's energy efficiency is 266.67 TOPS/W. The area efficiency, expressed as the ratio of throughput to core area, is 470.588 GOPS/mm^2.

Table II shows the performance comparison of recent SRAM-based CIMs. Notably, our CIM has better energy efficiency among others except Ref. [12] which is fabricated using 7-nm process. However, when bitwise energy efficiency is considered, our CIM is the best, since its input and weight bitwidth is 8 bits unlike Ref. [12]'s which only has 4-bit width.

IV. CONCLUSION

A 266.7 TOPS/W CIM is designed using 16-nm FinFET process. It utilizes a novel single-ended 6T SRAM operating at 1 GHz clock frequency. It has the best bitwise energy efficiency among all CIMs. Fabrication and measurement of the CIM in silicon will be our future work.

REFERENCES

[1] Q. Dong, S. Jeloka, M. Saligane, Y. Kim, M. Kawaminami, A. Harada, S. Miyoshi, M. Yasuda, D. Blaauw, and D. Sylvester, "A 4 + 2T SRAM for searching and in-memory computing with 0.3-V V_{DDmin}," *IEEE Journal of Solid-State Circuits (JSSC)*, vol. 53, no. 4, pp. 1006-1015, Apr. 2018.

[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 (TCAS-I)*, pp. 1-14, Jul 2018.

[3] C.-C. Wang, Y.-L. Tseng, H.-Y. Leo, and R. Hu, "A 4-kB 500-MHz 4-T CMOS SRAM using low-V/sub THN/ bitline drivers and high-V/sub THP/ latches," *IEEE Trans. on Very Large Scale Integration Systems (TVLSI)*, vol. 12, no. 9, pp. 901-909, Sep. 2004.

[4] C.-C. Wang, N. Sulistiyanto, T.-Y. Tsai, and Y.-H. Chen, "Multifunctional in-memory computation architecture using single-ended disturb-free 6T SRAM," *Lecture Notes in Electrical Engineering*, vol. 619, pp. 49-57, 2020.

[5] C.-C. Wang, C.-Y. Huang and C.-H. Yeh, "SRAM-based computation in memory architecture to realize single command of add-multiply operation and multifunction," in *Proc. 2021 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1-4, May 2021.

[6] C.-C. Wang, L. K. S. Tolentino, C.-Y. Huang and C.-H. Yeh, "A 40-nm CMOS multifunctional computing-in-memory (CIM) using single-ended disturb-free 7T 1-Kb SRAM," *IEEE Trans. on Very Large Scale Integration Systems (TVLSI)*, vol. 29, no. 12, pp. 2172-2185, Dec. 2021.

[7] S. K. Gonugondla, M. Kang, and N. Shanbhag, "A 42pJ/decision 3.12TOPS/W robust in-memory machine learning classifier with on-chip training," in *Proc. 2018 IEEE International Solid-State Circuits Conference - (ISSCC)*, pp. 490-492, Feb. 2018.

[8] X. Si, J.-J. Chen, Y.-N. Tu, W.-H. Huang, J.-H. Wang, Y.-C. Chiu, W.-C. Wei, S.-Y. Wu, X. Sun, R. Liu, S. Yu, R.-S. Liu, C.-C. Hsieh, K.-T. Tang, Q. Li, and M.-F. Chang, "24.5 A twin-8T SRAM computation-in-memory macro for multiple-bit CNN-based machine learning," in *Proc. 2019 IEEE International Solid-State Circuits Conference - (ISSCC)*, pp. 396-398, Mar. 2019.

[9] X. Si, J.-J. Chen, Y.-N. Tu, W.-H. Huang, J.-H. Wang, Y.-C. Chiu, W.-C. Wei, S.-Y. Wu, X. Sun, R. Liu, S. Yu, R.-S. Liu, C.-C. Hsieh, K.-T. Tang, Q. Li, and M.-F. Chang, "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, Jan. 2020.

[10] J.-W. Su, X. Si, Y.-C. Chou, T.-W. Chang, W.-H. Huang, Y.-N. Tu, R. Liu, P.-J. Lu, T.-W. Liu, J.-H. Wang, Z. Zhang, H. Jiang, S. Huang, C.-C. Lo, R.-S. Liu, C.-C. Hsieh, K.-T. Tang, S.-S. Sheu, S.-H. Li, H.-Y. Lee, S-C. Chang, S. Yu, and M.-F. Chang, "15.2 A 28nm 64Kb inference-training two-way transpose multibit 6T SRAM compute-in-memory macro for AI edge chips," in *Proc. 2020 IEEE International Solid-State Circuits Conference - (ISSCC)*, pp. 240-242, Feb. 2020.

[11] J.-W. Su, J.-W. Su, X. Si, Y.-C. Chou, T.-W. Chang, W.-H. Huang, Y.-N. Tu, R. Liu, P.-J. Lu, T.-W. Liu, J.-H. Wang, Y.-L. Chung, J.-S. Ren, F.-C. Chang, Y. Wu, H. Jiang, S. Huang, S.-H. Li, S.-S. Sheu, C.-I. Wu, C.-C. Lo, R.-S. Liu, C.-C. Hsieh, K.-T. Tang, S. Yu, and M.-F. Chang, "Two-way transpose multibit 6T SRAM computing-in-memory macro for inference-training AI edge chips," *IEEE Journal of Solid-State Circuits*, vol. 57, no. 2, pp. 609-624, Feb. 2022.

[12] Q. Dong, M. E. Sinangil, B. Erbagci, D. Sun, W.-S. Khwa, H.-J. Liao, Y. Wang, and J. Chang, "15.3 A 351TOPS/W and 372.4GOPS compute-in-memory SRAM macro in 7nm FinFET CMOS for machine-learning applications," in *Proc. 2020 IEEE International Solid-State Circuits Conference - (ISSCC)*, pp. 242-244, Feb. 2020.

[13] Z. Chen, Z. Yu, Q. Jin, Y. He, J. Wang, S. Lin, D. Li, Y. Wang, and K. Yang, "CAP-RAM: A charge-domain in-memory computing 6T-SRAM for accurate and precision-programmable CNN inference," *IEEE Journal of Solid-State Circuits*, vol. 56, no. 6, pp. 1924-1935, June 2021.

[14] N. Pan, X. Cui, X. Qiao, K. Xiao, Q. Guo, and Y. Wang, "A 28nm 64Kb SRAM based inference-training tri-mode computing-in-memory macro," in *Proc. 2022 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 2561-2565, May 2022.