VOL. 24, NO. 1, 2025, 47-52
elektrika.utm.my
ISSN 0128-4428

ELEKTRIKA
Journal of Electrical Engineering

Implementation of 10 Transistor SRAM Computing-in-Memory for Binarized Multiply Accumulate Unit

Rue Yie Lim, Afiq Hamzah*, N. Ezaila Alias, Michael Loong Peng Tan and **Izam Kamisian**

Faculty of Electrical Engineering, Universiti Teknologi Malaysia, 81310 UTM Skudai, Johor, Malaysia.

*Corresponding author: mafiq@utm.my

Abstract: The von Neumann bottleneck is a major challenge in the development of energy-efficient processors capable of handling high-workload computations. Computing-in-memory (CiM) technique offers a promising solution to overcome the memory wall restrictions that limit performance. By embedding processing units directly into memory, CiM can mitigate issues of latency and energy consumption during memory access. In this study, we implemented a dual-port design method for a 10-Transistor (10T) SRAM bit-cell to perform Binarized Multiply-Accumulate operation using 45nm CMOS process technology. We use several functional block designs, including isolated read and write paths, to design the 1Kb CiM architecture using the Cadence Virtuoso EDA tool. The proposed 10T SRAM-CiM design supports fully parallel computing, allowing it to perform 32 Binarized MAC operations simultaneously. The design achieves a maximum operating frequency of 100MHz, a throughput of 204.8 GOPS and energy efficiency of 443.15 TOPS/W.

Keywords: Computing-in-Memory (CiM), Binarized Neural Network (BNN), Multiply Accumulate (MAC)

Article History: received 2 November 2024; accepted 19 February 2025; published 30 April 2025

1. INTRODUCTION

Deep Neural Networks (DNNs) have revolutionized artificial intelligence (AI) and machine learning (ML) applications. DNNs consist of multiple hidden layers of Convolutional Neural Networks (CNNs), where Multiply-Accumulate (MAC) operations compute the dot product between N-bit weights and activations. However, the computational workload of the MAC operation in hidden layers is computationally expensive and storage-intensive [1, 2]. To address these issues, Binary Neural Networks (BNNs) have been introduced, which employ parameter quantization and perform Binarized MAC operations using bipolar values (+1 and -1) represented as 0. This is done by the binarization of both the weight and activation inputs, where real-value variables are converted to binary-value variables. This process further simplifies the multiplication in the Multiply-Accumulate (MAC) operation to addition and subtraction through the deterministic sign function as

$$w_b = \begin{cases} +1, & if\ w \geq 0 \\ -1, & otherwise \end{cases} \quad (1)$$

where wb is the weight in binary value while w is the weight in real value.

Figure 1 shows the illustration of how Binarized MAC computation is performed. Initially, the multiplication between the input values and their respective binarized weight will occur in the first hidden layer. All the generated products are added together with a bias. The sign function is then applied to the summation output as an activation to binarize the non-negative and negative inputs into +1 and -1, respectively. During the forward propagation, the weights and inputs in the following layer are also binarized.

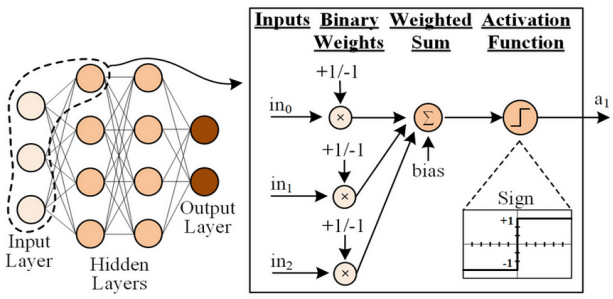

Figure 1. The 1-bit input, weight and output activation of BNN [3].

The concept of binary weight was first introduced in Binary Connect (BC) [4], enabling significant computational savings in DNNs. Binarization has since become a popular approach to achieve high computational efficiency in DNNs, with various methods proposed to optimize the binarization process. By simplifying the MAC operation, binarization reduces the computational complexity of DNNs, enabling efficient processing in resource-constrained environments. While BNNs offer significant computational and storage advantages, recent

studies have shown that the energy consumption and latency issues during digital implementation of CNNs are dominated by frequent data transfer between memory and the processor due to the von Neumann computing architecture's design. This issue conflicts with the low-power requirements of IoT devices that use BNN processors for energy and area-efficient edge computing. To overcome these challenges, the computing-in-memory (CiM) technique has been introduced, which embeds processing units directly into memory to minimize data transfer. While BNNs offer advantages in computational and storage efficiency, the von Neumann architecture's design leads to energy consumption and latency issues. The CiM technique shows promise in minimizing data transfer and addressing these challenges for BNN-based IoT devices.

One of the mainstream approaches for computing-in-memory involves the use of the 6-Transistor (6T) static random-access memory (SRAM) bit-cell, which offers a compact layout. However, the implementation of 6T SRAM-CiM presents significant challenges due to the possibility of write disturbance caused by the cross-coupled mechanism inherent in the conventional 6T SRAM configuration [5], [6]. To address this limitation, modifications to the SRAM bit-cell configuration are necessary to separate its read and write paths, enabling independent access to each bit-cell and preventing interference during the discharging process on one bitline. In this work, we propose a 1Kb Computing-in-Memory (CiM) array based on the 10T SRAM for Binarized Multiply-Accumulate using 45nm CMOS technology. The proposed design addresses the limitation of 6T SRAM-CiM by adopting a dual-port design method with isolated read and write paths. The resulting 10T SRAM-CiM design supports fully parallel computing, allowing it to perform 32 Binarized MAC operations simultaneously. Our findings demonstrate the potential of the proposed 10T SRAM-CiM design to overcome the challenges posed by the von Neumann bottleneck and enable energy-efficient high-performance computing.

2. COMPUTING-IN-MEMORY ARRAY ARCHITECTURE

The overall architecture of the 1Kb 10T SRAM-CiM array is presented in Figure 2. The unit-macro comprises a read-wordline (RWL) driver, 32 rows by 32 columns of 10T bit-cells, a reference array block, a CIM-control block, and a CIM I/O block. The RWL driver features 32 decoders and driver cells, while the CIM I/O block comprises 32 sense amplifiers (SA), 32 evaluation (EVAL) cells, a reference voltage generator cell, and a sense enable generator cell. The reference array block uses three columns of 10T bit-cells to generate the reference voltage and sense enable signal. The EVAL cell sums the RC currents, IRC generated on RBL and RBLX within the same column and converts them into an analog voltage signal that then becomes the input of the SA. The SA compares the analog voltage with the reference voltage to produce the activation output.

The unit-macro operates in two modes: SRAM and CIM. In SRAM mode, the array stores the trained weights by enabling write operations using the read/write control (RW_CTL) and read-write IO blocks (RW_IO). Triggering WWL accesses one row through the RW_IO block. In CIM mode, the unit-macro implements 32 binarized MAC operations for 32 activation inputs (IN) and 32 weights (W) in parallel computation. The weights are stored in m number (m = 0 to 31) of 10T bit-cells within the same column. In MAC computing, each IN[n] is inserted into the RWL driver to activate the RWLn and RWLXn pair (n = 0 to 31) upon activation of multiple rows. The current sum of the n bit-cells is then sensed at RBL and RBLX to determine IWP = IN × W.

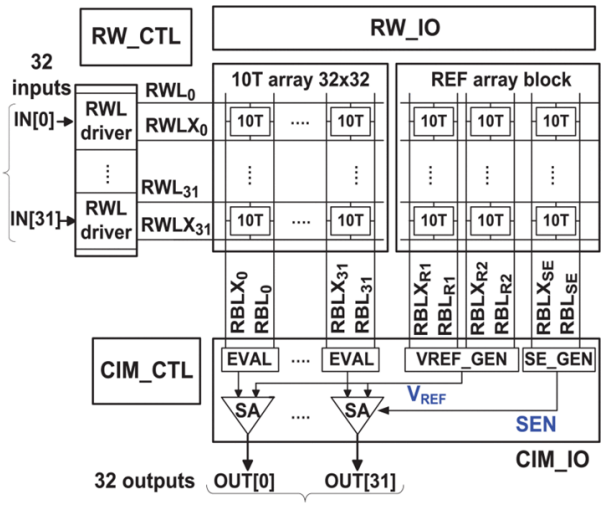

Figure 2. Macro architecture of 1 Kb 10T SRAM-CiM [7].

3. 10T SRAM-CIM BIT CELL FOR BINARIZED MAC

The 10T SRAM-CiM bit-cell is presented in Figure 3 and comprises a conventional 6T SRAM and four additional transistors (M0-M3) for a decoupled read-port [7]. The bit-cell operates in SRAM mode for read/write operations and in CiM mode to perform Multiply-Accumulate (MAC) operations.

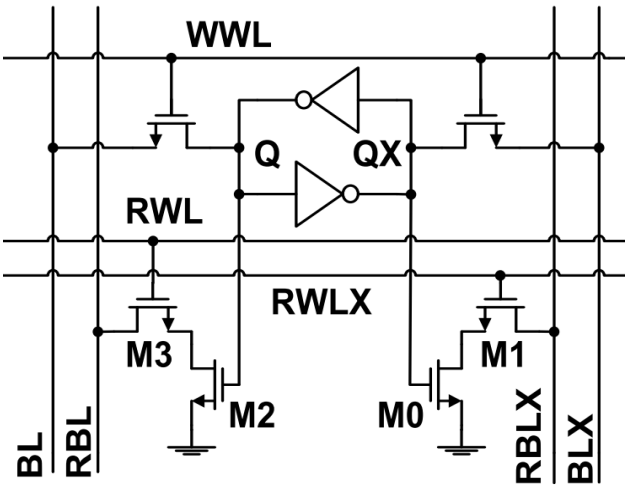

Figure 3. 10T SRAM-CiM bit-cell [7].

In SRAM mode, the read and write schemes are similar

to those of conventional 6T SRAM. The write wordline (WWL) is enabled to write data into the bit-cell, and the stored logic can be read from the bitline (BL). In CiM mode, the activation input, IN[i], is represented by the read wordline (RWL) and complemented read wordline (RWLX). If IN[i] = +1, RWL is asserted as '1', and RWLX is asserted as '0'; and vice versa for IN[i] = -1. The bit-cell stores the binary weight, W, and the output Q is determined by the value of W. If W = +1, output Q = 1, QX = 0, and vice versa. The input weight product (IWP) value of the 10T bit-cell, resulting from the mathematical equation of IN[i] × W[i], is summarized in Table 1. If IWP = +1, the RC current, IRC, is generated in either RBL or RBLX; if IWP = -1, no RC current is generated. The 10T SRAM-CiM design allows the bit-cell to operate in both SRAM and CiM modes, enabling parallel computing and efficient processing of Binarized MAC operations.

Table 1. Truth table of the 10T SRAM-CiM bit-cell [7].

Input			**Weight**			**IWP**		
Value	RWL	RWLX	Value	Q	QX	Value	RBL	RBLX
-1	0	1	-1	0	1	1	0	I_{RC}
-1	0	1	1	1	0	-1	0	0
1	1	0	-1	0	1	-1	0	0
1	1	0	1	1	0	1	I_{RC}	0

The simulation waveform presented in Figure 4 demonstrates the functionality of the bit-cell in both SRAM and CiM mode. The bit-cell operates in SRAM mode when WWL = 0 and in CiM mode when WWL = 1 with the output of the sense amplifier, VOUT shows the results of the truth table shown in Table 1. It can be observed that the bit-cell successfully performed binarized multiplication. In the first WWL cycle, the bit-cell stored '0' (BL = Q = 0) followed by CiM mode for multiplying Input = 0 (RWL = 0) and Weight = 0 (Q = 0) resulting VOUT = 1. In the second WWL cycle, the bit-cell stored '1' (BL = Q = 1) followed by CiM mode for multiplying Input = 0 (RWL = 0) and Weight = 1 (Q = 1) resulting VOUT = 0. In the third WWL cycle, the bit-cell stored '0' (BL = Q = 0) followed by CiM mode for multiplying Input = 1 (RWL = 1) and Weight = 0 (Q = 0) resulting VOUT = 0. Finally, in the fourth WWL cycle, the bit-cell stored '1' (BL = Q = 1) followed by CiM mode for multiplying Input = 1 (RWL = 1) and Weight = 1 (Q = 1) resulting VOUT = 1.

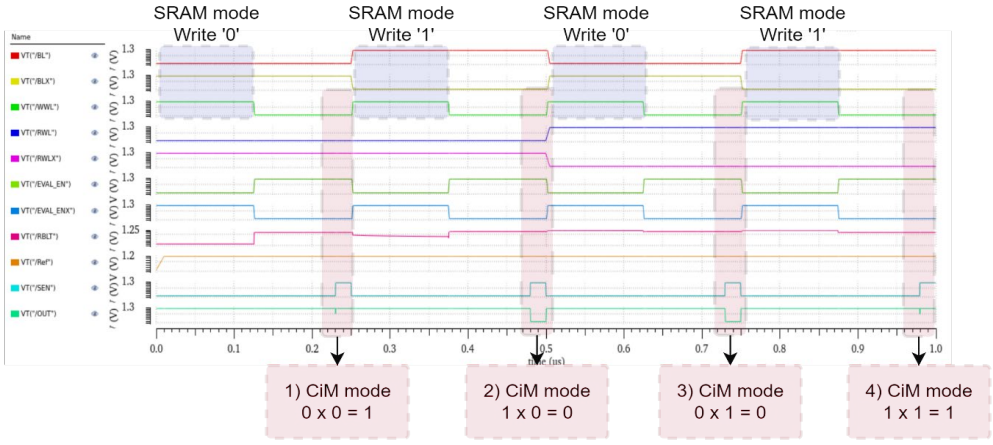

Figure 4. 10T SRAM bit-cell operates in SRAM and CiM mode.

4. BINARIZED MAC FUNCTIONAL VERIFICATION

The memory architecture design includes 32 BL inputs for weight data entrance and 32 RBL outputs for CiM result exit, connected column-wise. Similarly, the 32 WWL inputs and 32 RWL inputs for write enabling signal (Wr_EN) and activation input data entrance, respectively, are connected row-wise. The macro unit undergoes both SRAM and CiM modes, with the same purpose as discussed previously for the single 10T SRAM-CiM bit cell. In SRAM mode, the insertion of the Wr_EN control signal is followed by row-by-row activation of WWL, allowing bit-by-bit storage of weight data. Storing weight data with different logic levels within the same column, rather than the same logic level, is achieved by activating the 32 WWL inputs separately. Activation input data ready on the RWL inputs is transmitted to the bit cells within the same row. After setting up both essential elements, the macro unit enters the CiM mode, consisting of three main stages. In the first stage, each bit cell computes the input-

weight product. The 32 input-weight products on the RBL and RBLX are accumulated in the EVAL cell during the evaluation cycle in the second stage. In the last stage, the finalized binarized MAC result is sensed when the read cycle is activated by triggering the SEN control signal.

To verify the functionality of the memory architecture, an example is shown as in Table 2. The activation input data is set up, with 17 out of 32 inputs having logic '1'. This is followed by setting up the weight data, with eight different sets of weight data showcased. The weight data is stored in the bit cell following row-by-row activation of WWL when running the design in SRAM mode. The manual analysis involves several steps. In step 1, the product between the input and weight is computed bit-wise, referring to Table 1. The number of input-weight products having logic '1' (NIWP1) is recorded by applying the popcount to the computation result. The same working step is applied to record the number of input-weight products having logic '0' (NIWP0). Finally, by comparing NIWP1 and NIWP0, the binarized MAC output (OUT) is obtained. If NIWP1 > NIWP0, OUT = 1, and vice versa. The activation, weight, NIWP1, NIWP0, and OUT for this example are summarized in Table 2.

Table 2. Example of Binarized MAC operation of 32-bit activation input data with 32 sets of 8-bit weight, and its activation output, OUT.

	Input, IN		Weight, W							
			[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]
Data	[0]	0	0	1	1	0	1	0	1	0
	[1]	0	0	1	1	0	1	0	1	0
	[2]	0	0	1	0	1	1	0	1	0
	[3]	0	0	1	0	1	1	0	1	0
	[4]	0	0	1	1	0	1	0	0	1
	[5]	0	0	1	1	0	1	0	0	1
	[6]	0	0	1	0	1	1	0	0	1
	[7]	0	0	1	0	1	1	0	0	1
	[8]	0	0	1	1	0	1	0	1	0
	[9]	0	0	1	1	0	1	0	1	0
	[10]	0	0	1	0	1	1	0	1	0
	[11]	0	0	1	0	1	1	0	1	0
	[12]	0	0	1	1	0	1	0	0	1
	[13]	0	0	1	1	0	1	0	0	1
	[14]	0	0	1	0	1	1	0	0	1
	[15]	1	0	1	0	1	1	0	0	1
	[16]	1	0	1	1	0	1	0	1	0
	[17]	1	0	1	1	0	1	0	1	0
	[18]	1	0	1	0	1	0	1	1	0
	[19]	1	0	1	0	1	0	1	1	0
	[20]	1	0	1	1	0	0	1	0	1
	[21]	1	0	1	1	0	0	1	0	1
	[22]	1	0	1	0	1	0	1	0	1
	[23]	1	0	1	0	1	0	1	0	1
	[24]	1	0	1	1	0	0	1	1	0
	[25]	1	0	1	1	0	0	1	1	0
	[26]	1	0	1	0	1	0	1	1	0
	[27]	1	0	1	0	1	0	1	1	0
	[28]	1	0	1	1	0	0	1	0	1
	[29]	1	0	1	1	0	0	1	0	1
	[30]	1	0	1	0	1	0	1	0	1
	[31]	1	0	1	0	1	0	1	0	1
Pop-count	N_{IWP1}		15	17	15	17	3	29	15	17
	N_{IWP0}		17	15	17	15	29	3	17	15
Binarized MAC OUT			0	1	0	1	0	1	0	1

5. MEMORY ARRAY EVALUATION

The computing-in-memory architecture of 1Kb (1024-bit) size based on 10T SRAM is implemented using 45nm standard CMOS technology. The 1Kb SRAM-CiM macro unit is operated with a nominal supply voltage and body-biasing voltage of 1.2V. By using process corner analysis, the operating frequency of this macro unit design has been determined at various temperatures and process corners. Both typical NMOS typical PMOS (tt) and fast NMOS fast PMOS (ff) process corners are chosen to determine the maximum operating frequency (fmax) of the SRAM-CiM. Table 3 shows the comparison result of the maximum

operating frequency at different process corners and temperatures. Some other important parameters such as throughput, throughput density, and energy efficiency (Eeff) are then calculated.

Table 3. Performance of SRAM-CiM at different process corners and temperatures.

Process Corner	f_{max} (MHz)	
	25°C	85°C
tt	1	1
ff	100	100

A deeper analysis is conducted to investigate whether the benefits/drawbacks of this SRAM-CiM design are derived from the process technology or the circuit design technique. A technology scaling factor Stech of 1.44 is used for 65nm CMOS process technology [8]. This approach is aimed at ensuring the fairness of the whole comparison process by assuming other works are also implemented using 45nm CMOS technology. The performance metrics are calculated based on the formulas: $throughput \propto S_{tech}$ and $E_{eff} \propto (S_{tech})^2$. By referring to Table 3, the operating frequency for this SRAM-CiM macro unit design is recorded at a maximum value of 100MHz. It can achieve a relatively high throughput of 204.8 GOPS and a throughput density of 204.8 GOPS/Kb. The total energy efficiency of this macro unit is measured at a value of 443.15 TOPS/W.

These data are further compared with the previous works [9], [10], [6], [11], and [12] as shown in Table 4. The analysis results after applying the scaling factor, Stech show that this work has approximately 20X, 18X, 284X, and 5X improvements in terms of operating frequency, throughput, throughput density, and energy efficiency, as compared to [9]. Other works with similar 65nm CMOS process technology as in [9] are [6], [11], and [12]. As compared to [6], this work is slightly weaker in throughput performance by a factor of 1.95X. However, this can be compromised by its 2.05X higher throughput density and 3.83X higher energy efficiency performances than [6]. The comparison results obtained between this work and [11] show that this work has 3.7X greater throughput density, but 4.32X smaller throughput and 1.89X lower performance in terms of energy efficiency. Apart from operating frequency, this work is weaker than [12] in other performance metrics. This work can achieve 2X higher operating frequency than [12]. Nevertheless, it is weaker than [12] in terms of throughput, throughput density, and energy efficiency, by a factor of 11.52X, 5.76X, and 3.14X, respectively. In comparison to the work [10] that utilizes the 45nm CMOS process technology, it is observed that this work has comparatively higher achievement in all performance metrics. To summarise the comparison results between this work and [10], this work has approximately 4.5X, 24X, 186X, and 22X enhancement in terms of operating frequency, throughput, throughput density, and energy efficiency, respectively.

Table 4. Comparison of proposed SRAM-CiM.

	[9]	[10]	[4]	[11]	[12]	This work
Technology	65nm	45nm	65nm	65nm	65nm	45nm
Macro size	16Kb	8Kb	4Kb	16Kb	2Kb	1Kb
Cell structure	10T	10T	Split-WL 6T	12T	8T1C	10T
Input (bit)	6	1	1	1	1	1
Weight (bit)	1	1	1	1	1	1
Output (bit)	6	5	1	3.5	5	1
Operating frequency (MHz)	5	22.2	N/A	N/A	50	100
Throughput (GOPS)	8 (11.5)*	8.5	278 (400)*	614 (884)*	1638 (2358.7)*	204.8
Throughput density (TOPS/Kb)	0.5 (0.72)*	1.1	69.5 (100.08)*	38.4 (55.3)*	819 (1179.4)*	204.8
Energy efficiency (TOPS/W)	40.3 (83.6)*	19.9	55.8 (115.7)*	403 (835.7)*	671.5 (1392)*	443.15

* Scaling factor (S_{tech}) is applied.

6. CONCLUSION

A 1Kb computing-in-memory based on the proposed 10T SRAM is designed and evaluated. The employment of the dual-port approach has prevented this SRAM-CiM design from the write disturbance issue that is faced in the conventional 6T SRAM topology. The simulation results show that this SRAM-CiM design possesses full functionality at a nominal supply voltage of 1.2V in 45nm standard CMOS process technology. This CiM architecture can operate in a way to complete 32 MAC operations between the activation input data and weight data at one-time goes. The bit-by-bit storage mechanism makes the logic level of the data going to be stored within the same column varied. This allows two datasets to undergo MAC computation in all possible combinations. The simulation results returned from the functional verification flow evidence that this design has hit the goal of performing the binarized MAC operation between two datasets in different combinations. The MAC execution

between two datasets in different combinations will result in a variety of total read bitline voltage. Therefore, finding a reference voltage value that is suitable for all possible kinds of situations is considered the challenging part of this project.

The comparison results corresponding to the situation after applying scaling factor, Stech, show that this macro unit has a great achievement in getting a maximum operating frequency of 100MHz, which is the highest among all the SRAM-CiM studies. The advantage of getting a high operating frequency will be further extended to a high throughput of 204.8 GOPS, a great improvement from [9] and [10]. This value is still slightly lower than the throughput gained in literature [4], [11], and [12]. However, by considering the macro size, this design can achieve a comparatively high throughput density of 204.8GOPS/Kb, surpassing all the previous works including [4] and [11], except [12]. The high energy efficiency of 443.15 TOPS/W is credited for this design, which is higher than all the previous works, but slightly lower than [12].

ACKNOWLEDGMENT

Authors would like to acknowledge the financial support of the Ministry of Higher Education (MoHE), Malaysia and to the Research Management Center (RMC) of Universiti Teknologi Malaysia (UTM) for providing an excellent research environment and financial support under the UTM Fundamental Research (UTMFR) grant, reference code R.J130000.7851.5F128 (PY/2022/04292) in which to complete this work.

REFERENCES

[1] G B. Moons and M. Verhelst, ‘‘An energy-efficient precision-scalable Con-vNet processor in 40-nm CMOS,’’ IEEE J. Solid-State Circuits, vol. 52, no. 4, pp. 903–914, Apr. 2017.

[2] Y. Li and Y. Du, ‘‘A novel software-defined convolutional neural networks accelerator,’’ IEEE Access, vol. 7, pp. 177922–177931, Nov. 2019.

[3] Dubey, Anuj & Cammarota, Rosario & Aysu, Aydin. (2020). BoMaNet: Boolean Masking of an Entire Neural Network. 1-9.

[4] M. Courbariaux, Y. Bengio, and J.-P. David, “BinaryConnect: Training deep neural networks with binary weights during propagations,” in Proc. NIPS, 2015, pp. 3123–3131.

[5] M. Kang, S. K. Gonugondla, A. Patil, and N. R. Shanbhag, ‘‘A multifunctional in-memory inference processor using a standard 6T SRAM array,’’ IEEE J. Solid-State Circuits, vol. 53, no. 2, pp. 642–655, Feb. 2018.

[6] X. Si, M.-F. Chang, W.-S. Khwa, J.-J. Chen, J.-F. Li, X. Sun, R. Liu, S. Yu, H. Yamauchi, and Q. Li, ‘‘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.

[7] Nguyen, V. T., Kim, J. S., & Lee, J. W. (2021). 10T SRAM Computing-in-Memory Macros for Binary and Multibit MAC Operation of DNN Edge Processors. IEEE Access, 9, 71262–71276.

[8] J. P. Uyemura, Introduction to VLSI Circuits and Systems. Hoboken, NJ, USA: Wiley, 2002.

[9] Biswas, A., & Chandrakasan, A. P. (2019). CONV-SRAM: An Energy-Efficient SRAM With In-Memory Dot-Product Computation for Low-Power Convolutional Neural Networks. IEEE Journal of Solid-State Circuits, 54(1), 217–230.

[10] Geethumol, T & K. S, Sreekala & Dhanusha, P. (2017). Power and Area Efficient 10T SRAM with Improved Read Stability. ICTACT Journal on Microelectronics. 3. 337-344.

[11] 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.

[12] Z. Jiang, S. Yin, J.-S. Seo, and M. Seok, ‘‘C3SRAM: An in-memory computing SRAM macro based on robust capacitive coupling computing mechanism,’’ IEEE J. Solid-State Circuits, vol. 55, no. 7, pp. 1888–1897, Jul. 2020.