In-Memory Transposable Multibit Multiplication Based on Diagonal Symmetry Weight Block

Zhongzhen Tong, Yue Zhao, Jin Zhang, Zhiting Lin, Xiaoyang Lin, and Xiulong Wu

***Abstract*— A possible approach to overcome the von Neumann bottleneck and meet the increasing demand for better computing performance is to computing in-memory (CIM). The results of the in-memory calculations are primarily reflected in the vertical bitline (BL) analog voltage. However, the nonlinearity of the BL discharge deteriorates with the increase in discharge voltage. In this study, we propose a diagonal symmetry weight block (DSWB) based on an eight-transistor (8T) static random access memory (SRAM) that can achieve multibit transposable operations. In addition, to guarantee linearity and complete multibit multiplication operations, we propose a cascode current mirror (CCM)-based multiplier. To achieve low-overhead and more efficient quantification, our proposed CIM macro uses a counter-type quantization circuit to read out the analog calculation results. We simulated the performance of the proposed 8T SRAM in a 28-nm complementary metal–oxide–semiconductor process. The integral nonlinearity (INL) of the proposed CCM-based CIM decreased by approximately 54.4% compared with the traditional CIM. Furthermore, the proposed in-memory multibit multiplication throughput density was 6.74 GOPS/kb; this throughput density improvement is approximately 3.3–10.5 times higher than the existing CIM works.**

***Index Terms*— Computing in-memory (CIM), counter-type quantization circuits, diagonal symmetry weight block (DSWB), static random access memory (SRAM), transposable multibit multiplication.**

I. Introduction

Presently, almost all the advanced computer systems are being developed based on the von Neumann architecture. The typical characteristic of a CPU is the separation of memory and arithmetic logic unit (ALU), which leads to the von Neumann bottleneck. In this separated structure, the throughput between the ALU and memory is limited compared to the amount of memory; therefore, handling large amounts of data requires frequent data transfer between them. Consequently, the limited throughput results in significant energy consumption.

As the von Neumann bottleneck cannot be solved effectively using traditional digital solutions, many researchers have proposed the concept of computing in-memory (CIM) [1], [2], [3]. The CIM technology can alleviate bottlenecks, improve throughput, and reduce energy costs. Presently, most of the pioneering works on CIM use the voltage of a vertical bitline (BL) to represent the multiplication result, including a strategy based on BL shifting for in-memory multibit multiplication [4] and a 1-to-8-bit configurable static random access memory (SRAM) CIM macro based on a basic 6-transistor (6T) cell, in which the calculation results are obtained by the difference in voltage between two BLs [3]. However, the nonlinearity of the BL discharge deteriorates as the discharge voltage increases, and the calculation disturbance becomes increasingly significant as the number of bit cells involved in the operation increase [5].

In addition, relying solely on BL to obtain output results requires the rearrangement of data and complex writing techniques. Once the data are written, it is difficult to read out by word line [6]. Fortunately, this problem can be solved using in-memory transposable operations. Moreover, it can support matrix transpose and two-way propagation (forward and backward) for deep neural networks [7], which further improves the universality of the CIM architecture.

Some auxiliary circuits have been added to SRAM-based CIM; among these, two important auxiliary circuits are the weighing modules and analog-to-digital conversion (ADC) modules. Capacitor array weighting technology is often utilized for higher linearity and precision operation [8], [9]; however, using a large number of capacitors increases the power consumption and takes up more area. In contrast, quantification circuits process the final calculation result, which is crucial for calculating the accuracy of the entire system. Considering the overhead of quantization circuits, most existing CIM works have selected multiplexing quantization circuits [10], [11], [12]; however, these efforts reduce the throughput of the operations.

To overcome these challenges, this study proposes an eight-transistor (8T) SRAM. The advantages of the proposed structure are given as follows.

1) The diagonal symmetry weight block (DSWB) based on the 8T cell supports two-directional multibit operations, where the vertically connected read BL (RBL) or the horizontally connected source line (SL) can be selected to participate in the calculation and improve the uniformity of device density.
2) It realizes multibit multiplication while guaranteeing operational linearity. A cascode current mirror (CCM) clamps the RBL/SL voltage and proportionally mirrors the read current, which is also multiplexed as a multibit multiplier.
3) A counter-type quantization circuit realizes quantization with a low area cost. Therefore, it can be allocated to each row or column, thereby increasing the computing parallelism.

The remainder of this brief is organized as follows. Section II presents an overview of the proposed CIM-macro and its operating principle. Section III presents the performance evaluation of the proposed architecture. Finally, Section IV concludes the study.

II. Overview of the Proposed CIM-Macro and Operating Principle

A. Overview of the Proposed CIM-Macro

Fig. 1(a) shows the overall CIM-macro comprising 256 DSWBs for multibit multiplication operation, two sets of CCM-based multiplier

Manuscript received 14 October 2022; revised 10 January 2023 and 15 March 2023; accepted 6 April 2023. This work was supported in part by the National Natural Science Foundation of China, under Grant 62074001; in part by the Joint Funds of the National Natural Science Foundation of China, under Grant U19A2074; in part by the Key Research and Development Program of Anhui Province under Grant 2022a05020044; and in part by the Science Fund for Distinguished Young Scholars of Anhui Province under Gant 2108085J35. *(Corresponding author: Zhiting Lin.)*

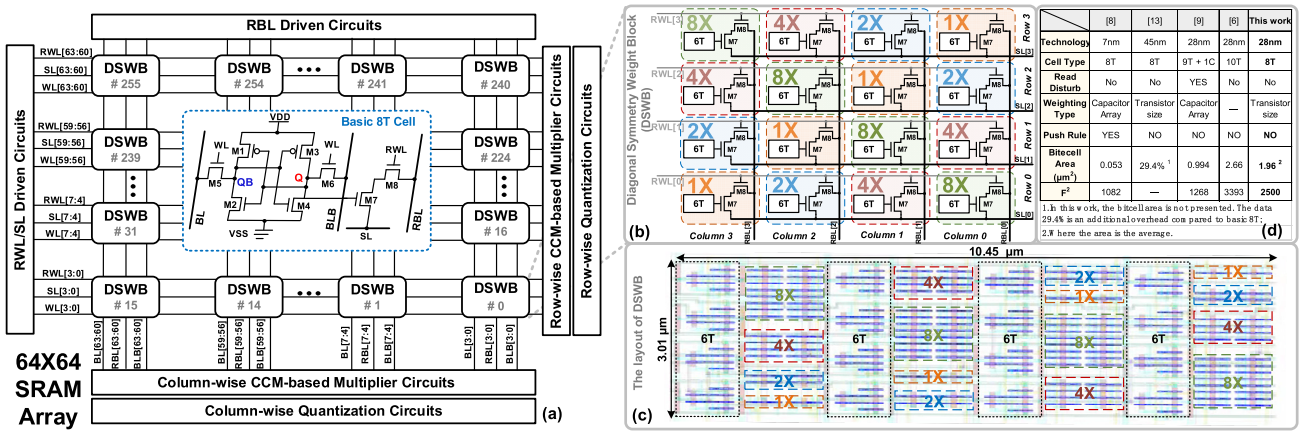
Zhongzhen Tong and Xiaoyang Lin are with the School of Integrated Circuit Science and Engineering, Beihang University, Beijing 100191, China.

Yue Zhao, Zhiting Lin, and Xiulong Wu are with the School of Integrated Circuits, Anhui Provincial High-Performance Integrated Circuit Engineering Research Center, Anhui University, Hefei 230601, China (e-mail: ztlin@ahu.edu.cn).

Jin Zhang is with the School of Management Science and Engineering, Anhui University of Finance and Economics, Bengbu 233030, China.

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TVLSI.2023.3266597.

Digital Object Identifier 10.1109/TVLSI.2023.3266597



	[8]	[13]	[9]	[6]	This work
Technology	7nm	45nm	28nm	28nm	28nm
Cell Type	8T	8T	9T + 1C	10T	8T
Read Disturb	No	No	YES	No	No
Weighting Type	Capacitor Array	Transistor size	Capacitor Array	—	Transistor size
Push Rule	YES	NO	NO	NO	NO
Bitcell Area (μm^2)	0.053	29.4% [1]	0.994	2.66	1.96 [2]
F^2	1082	—	1268	3393	2500

1. In this work, the bitcell area is not presented. The data 29.4% is an additional overhead compared to basic 8T.
2. # here the area is the average.

Fig. 1. (a) Overall SRAM architecture. (b) Schematic of DSWB. (c) Layout of DSWB. (d) Comparison of SRAM cell area with existing ones.

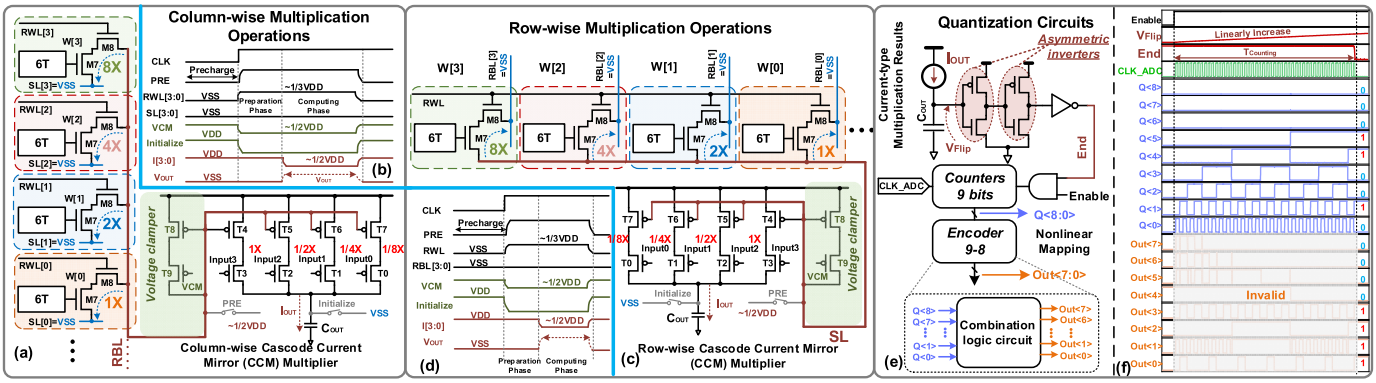


Fig. 2. Column-wise multiplication operation: (a) schematic and (b) timing diagrams. Row-wise multiplication operation: (c) schematic and (d) timing diagrams. Quantization circuits operation: (e) schematic of quantization circuit and (f) timing diagrams.

circuits to guarantee linearity and realize multibit input, two driven circuits for enabling read word lines (RWLs), SLs and RBLs, and two sets of low-overhead counter-type quantization circuits for output results in two directions.

The CIM-macro can be operated in two modes: memory mode and CIM mode. A detailed description of conventional SRAM mode [8] is not provided in this study. In the CIM mode, row- and column-wise multibit multiplications are realized by controlling the RBLs/SLs and the CCM-based multipliers. Thereafter, the two-direction calculation results can be read out using the counter-type quantization circuits.

Fig. 1(b) shows a schematic of the DSWB consisting of an 8T cell. Unlike the existing CIM work [13], where the width-to-length ratios of the four adjacent columns read-decoupled access transistors are 8 and 4:2:1, respectively, in the proposed DSWB, the width-to-length ratios of the read-decoupled access transistors are distributed diagonally and symmetrically. For example, column 1 is consistent with row 1, in which the width-to-length ratios of the read-decoupled access transistors of the four adjacent cells are 2, 1, 8, and 4. This arrangement offers the advantage of realizing bidirectional calculations and achieving better device density uniformity. If SLs connect to VSS, RBLs can discharge to SLs through M8 to M7; similarly, if RBLs connect to VSS, SLs can also discharge to RBLs through M7–M8, thereby realizing bidirectional calculation. Due to the inconsistent width-to-length ratio, the weight trained by software needs to be preprocessed according to the DSWB placement form. Fig. 1(c) shows the layout of the DSWB, occupying an area of 31.45 μm^2. The average area of 1-bit cell is 1.96 μm^2 [Fig. 1(d)]. This incurs an area overhead of ~20% compared to the basic 8T SRAM. However, compared to [13], which had a similar weighting type, our proposed cell area is slightly lower.

B. *Principle of Column-Wise Multibit Multiplication*

When the CIM-macro performs column-wise operations, multirow reads are realized by activating RWLs. Fig. 2(a) shows the principle of column-wise multibit multiplication. The 4-bit weight is stored in an SRAM array. The 4-bit input precision is achieved by proportionally mirroring the RBL current (I_{RBL}). T0–T9 build a current mirror, where T8 and T9 form a voltage clamper; T0–T7 proportionally mirror the current of T8 and T9 to output capacitance (C_{OUT}) according to the input values. The width-to-length ratios of T4/T3, T5/T2, T6/T1, and T7/T0 are 1, 1/2, 1/4, and 1/8, respectively. The RBL varies slightly during discharge because of T8 and T9, that is, the voltage clamper. Therefore, throughout the entire discharge process, the change in current is insignificant, which can greatly improve the computational linearity. The CCM-based multiplier C_{OUT} collects the mirrored current and converts it to an output voltage.

When the CIM-macro conducts column-wise operations, all the SLs connect to VSS, whereas RBLs connect to the column-wise CCM-based multipliers. The other signals of the proposed circuit for column-wise multibit multiplication operation are shown in Fig. 2(b). The calculation method is given as follows.

Step 1 (Precharging Operation): The RBLs are precharged to 1/2 VDD. The voltage on C_{OUT} is initialized to zero.

Step 2 (Preparation Phase): The initialization signal is closed. The corresponding RWLs are activated to generate the discharge current.

The RBL current (I_{RBL}) caused by a DSWB is given by the following equation:

$$I_{\mathrm{RBL}} = \Delta I \sum_{i=0}^{3} 2^i W[i] \tag{1}$$

where ΔI is the discharge current corresponding to the lowest weight bit. Because it takes some time for CCM to enter steady state and accurately copy the current, it opens in advance before step 3. The BL is precharged to 1/2 VDD in step 1; therefore, CCM can quickly enter the steady state, thereby significantly reducing the preparation phase time.

Step 3 (Computing Phase): Inputs 3–0 are mapped to gate voltages of T3–T0. If input3, input2, input1, or input0 is 1, the corresponding gate voltage of T3–T0 is set to 1/2 VDD in the computing phase. In contrast, if input j is 0, the corresponding gate voltage of Tj is maintained at VDD. Finally, C_{OUT} is charged according to the calculation results of I_{OUT}, which can be expressed as follows:

$$I_{\mathrm{OUT}} = I_{\mathrm{RBL}} \sum_{j=0}^{3} \left(\frac{1}{2}\right)^j \mathrm{Input}[j] = \Delta I \sum_{i=0}^{3} 2^i W[i] \sum_{j=0}^{3} \left(\frac{1}{2}\right)^j \mathrm{Input}[j] \tag{2}$$

where $W[i]$ is the weight and Input[j] is the input. Consider the operation 1101 1001 as an example. Under this condition, RBL discharges through cells with weights of 8 and 1. Therefore, I_{RBL} becomes

$$I_{\mathrm{RBL}} = \Delta I (2^3 \times 1 + 2^2 \times 0 + 2^1 \times 0 + 2^0 \times 1) = 9\Delta I. \tag{3}$$

The inputs are then used to open the corresponding T3–T0. Current I_{OUT} proportionally mirrors I_{RBL}, which is expressed as follows:

$$\begin{aligned} I_{\mathrm{OUT}} &= I_{\mathrm{RBL}} \left[\left(\frac{1}{2}\right)^0 \times 1 + \left(\frac{1}{2}\right)^1 \times 1 + \left(\frac{1}{2}\right)^2 \times 0 + \left(\frac{1}{2}\right)^3 \times 1 \right] \\ &= 9\Delta I \left(1 + \frac{1}{2} + \frac{1}{8}\right) = \frac{117}{8} \Delta I \propto 117. \end{aligned} \tag{4}$$

Therefore, the multiplication result corresponds to the decimal number 117.

C. Principle of Row-Wise Multibit Multiplication

When the CIM-macro performs row-wise operations, the multicolumn read is realized by enabling RBLs and activating full-array RWLs. RBLs of the activated columns are connected to VSS, whereas the others keep the VDD. The principle of row-wise multibit multiplication is shown in Fig. 2(c). The row-wise CCM-based multiplier is similar to that in the column direction, except that this circuit is used to clamp the SL voltage and copy the SL current. Similarly, in the row direction, 4-bit input precision is achieved by proportionally mirroring the SL current (I_{SL}) and 4-bit weight is stored in the SRAM array.

The row-wise calculation method is similar to the column-wise calculation, including precharging, preparation, and computing phases. When the CIM-macro performs row-wise operations, full-array RWLs are activated, whereas the SLs connect to the row-wise CCM-based multiplier and are precharged to 1/2 VDD during the precharging operation. Notably, under the column-wise calculation mode, RBLs are used to control the opened number of columns where the RBLs of the activated columns are set to VSS, while the others remain at VDD. As the RWLs are maintained at a low voltage, if RBLs connect to VDD, they cannot charge the SLs that are kept at approximately 1/2 VDD. Using this method, specific columns can be selected to participate in the operation. The row-wise multiplication result of one DSWB is expressed as follows:

$$I_{\mathrm{OUT}} = I_{\mathrm{SL}} \sum_{j=0}^{3} \left(\frac{1}{2}\right)^j \mathrm{Input}[j] = \Delta I \sum_{i=0}^{3} 2^i W[i] \sum_{j=0}^{3} \left(\frac{1}{2}\right)^j \mathrm{Input}[j]. \tag{5}$$

The signals of the proposed circuit for row-wise operations are shown in Fig. 2(d); here, the waves of RWLs and RBLs are different from those in column-wise operations.

Due to the complementary configuration of the SLs and RBLs, the array has two discharge directions, thereby realizing a two-directional calculation. In addition, the linearity of the calculation is guaranteed and multibit multiplication is realized by multiplexing CCM, which improves the circuit utilization in the system and decreases the area overhead.

D. Principle of Low-Overhead Quantization Circuits

Based on current-type multiplication results, we propose a low-overhead quantization circuit involving asymmetric inverters, 9-bit counters, and a 9-8 encoder, as shown in Fig. 2(e). This structure quantifies the multiplication calculation results by detecting the flip voltage V_{Flip} of the asymmetric inverters. The output load capacitance C_{OUT} is charged by the current-type multiplication results. When the calculation starts, the enable signal is turned on, and the counter starts counting. Once the voltage C_{OUT} reaches V_{Flip}, the counter stops counting. As a result, the output time (counting cycles) on the counter is associated with the output result. This association is expressed as follows:

$$\mathrm{Cycles} = \frac{V_{\mathrm{Flip}} C_{\mathrm{OUT}}}{I_{\mathrm{OUT}} T_{\mathrm{Counting}}} \propto \frac{1}{I_{\mathrm{OUT}}}. \tag{6}$$

As per the equation, T_{Counting} is the counter counting cycle; cycles are inversely proportional to I_{OUT}, which can improve the quantization accuracy when the result value is small; in other words, a smaller result value corresponds to a lowered probability of misquantification. The quantization results can be distinguished by the different counting times that can reach an 8-bit output precision. T_{Counting} has a nonlinear relationship with I_{OUT} and the result. Therefore, the output of the counter is 9 b. The 9-b counter output is then mapped to an 8-b digital code output through the encoder containing combinational logic circuits. As shown in Fig. 2(f), when the calculating result is 15, the counter counts 50 cycles. The counter output corresponds to Q⟨8:0⟩ = 000110010. Finally, Q⟨8:0⟩ is mapped by the encoder to Out⟨7:0⟩ > = 00001111, thereby achieving 8-b quantization. Compared to the state-of-the-art time-domain voltage-to-digital converter [14] with 28-nm process, which is 143.1 fJ/conv.-step, our proposed quantization circuits have lower energy consumption, achieving an average of 134 fJ/conv.-step at VDD = 0.9 V.

III. Simulation Results and Analysis

We implemented the proposed design in a 64 × 64 SRAM array with a 0.9-V 28-nm CMOS process; M1–M6 in the proposed cells employed the minimum transistor size. Notably, the simulations discussed next are all postlayout simulations.

A. Signal Margin Analysis

In this study, 4-b weight is obtained by different transistor sizes. Input is mapped as the CCM control signals achieving proportionally mirror RBL current. However, as the number of inputs increases, the signal margin will decrease, as shown in Fig. 3(a). When the input is

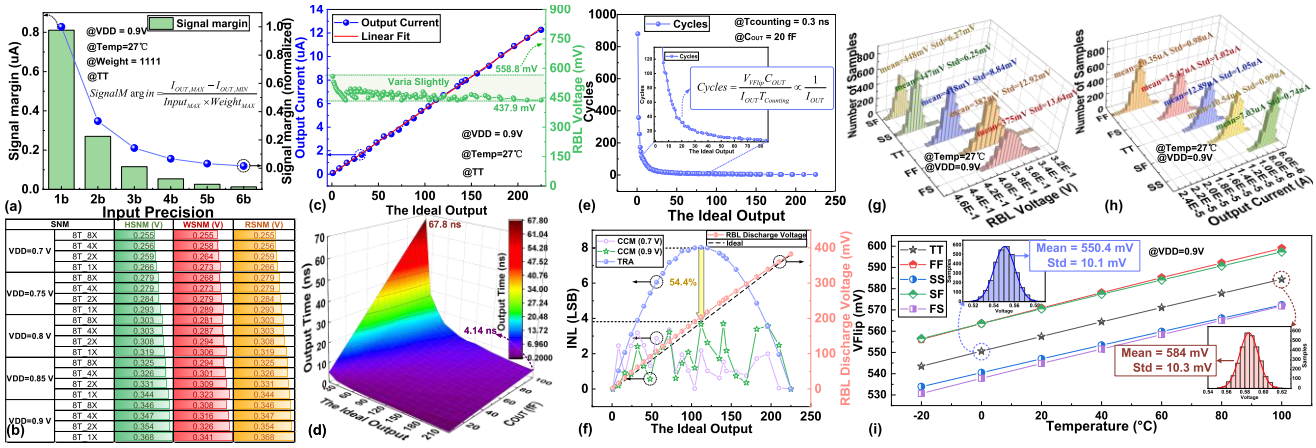


Fig. 3. (a) Limited signal margin and (b) static noise margin of proposed 8T. Simulation results of (c) output current and corresponding RBL voltage with 8-b output, (d) output time with 8-b output and various C_{OUT}, and (e) quantification cycles with 8-b output. (f) INL and discharge voltage of TRA and the proposed architecture (CCM). Monte Carlo simulations of (g) RBL voltage and (h) output current at 0.9 V. (i) Simulation of V_{Flip} in different corners and temperatures.

4 b, the signal margin is 0.054 μA; when the input is up to 6 b, the signal margin decreases to 0.013 μA. Thus, in the tradeoff between the number of inputs and signal margin, we chose 4 b as the input precision.

B. Performance of the Multiplication Operations

We evaluated the proposed CIM-macro from four aspects: static noise margin (SNM), output current with different inputs and weights, counting time with 8-b output and C_{OUT}, and computing performance.

The robustness of the 8T cell was evaluated in terms of SNM. Fig. 3(b) plots the hold SNM (HSNM), read SNM (RSNM), and write SNM (WSNM) for the proposed 8T with different sizes. DSWB includes four 8T cells of different sizes. The 1× size of 8T cell (basic 8T) exhibits the optimal SNM. As the size of the read decoupling transistor increases in the 8T cell, the corresponding parasitic capacitance will also increase, resulting in a slight reduction in the SNM. The HSNM, RSNM, and WSNM of 8× size of 8T cell are 345.8, 345.8, and 307.5 mV, respectively, at 0.9 V.

Because the calculation results were mapped as the output current, we conducted a postlayout simulation for the output current at 0.9 V with 8-b output. As shown in Fig. 3(c), the output current is linear with the calculation results. In addition, as CCM participates in the calculation, the RBL voltage is clamped; thus, the RBL fluctuation is small [Fig. 3(c)].

Fig. 3(d) shows the output time with 8-b output and various C_{OUT}. When the 8-b output is constant, the output time increases as C_{OUT} increases. When the 8-b output is 225, the output time is 0.98 and 4.14 ns at $C_{\mathrm{OUT}} = 20$ and 100 fF, respectively. To decrease the latency, $C_{\mathrm{OUT}} = 20$ fF is used in circuits. To verify the computing performance, we plotted the quantification cycles under $T_{\mathrm{Counting}} = 0.3$ ns. As shown in Fig. 3(e), when the 8-b output is larger, the counting cycles are less. When the calculated result is 1, the counting cycles are 880. However, when the count exceeds 358 cycles (the calculated result is 2), it can be regarded as the output result of 1, without waiting for the end of the count. This is because, when the weight is 0, the calculating result is 0, and the RBL is clamped at a much higher voltage. Therefore, the RBL voltage can be detected by the asymmetric inverter for disabling the counter. The average output time is 9.5 ns, which includes the operating time of CCM and is the average of the sum of calculation times corresponding to different results. The integral nonlinearity (INL) values of the CCM-based and traditional (TRA) computing architecture are shown in Fig 3(f). TRA computing is based on RBL discharge, where the RBL discharge is from 16.7 to 382.8 mV. Compared to TRA computing, the INL of CCM-based computing decreased by 54.4%. When the supply voltage is 0.7 V, the CCM-based computing still has a high linearity.

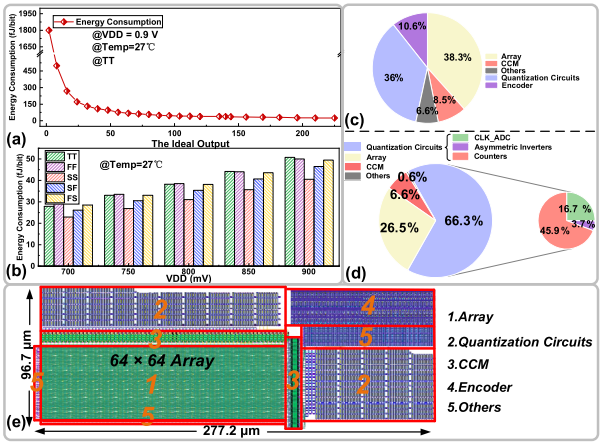


Fig. 4. Power consumption with various (a) calculating results and (b) VDDs. (c) Proportion of area and (d) energy consumption. (e) Complete layout.

C. Reliability Analysis

To verify the circuit reliability, as shown in Fig. 3(g) and (h), we performed Monte Carlo experiments for the RBL voltage and I_{OUT}. The key to improving the linearity is to clamp the RBL voltage. The results revealed that when the 8-b output was 225, RBL voltage and I_{OUT} achieved a mean value of 382 mV and 20.35 μA, respectively, with a corresponding standard deviation of 12.92 mV and 0.98 μA, respectively, at 0.9 V with FF corner. These results demonstrate that the distribution of RBL voltage or I_{OUT} is concentrated without a large deviation at extreme process corners. Fig. 3(i) shows V_{Flip} in different corners and temperatures. V_{Flip} changes from 540.5 to 571.8 mV from 0 °C SS corner to 100 °C FS corner, which ensures the stability of the counting cycles.

D. Energy Consumption and Area Evaluation

Fig. 4(a) shows the power consumption corresponding to different calculation results. Fig. 4(b) shows the average power consumption

TABLE I
COMPARISON WITH OTHER REPORTED WORKS

	This work	JSSC' 22 [1]	TCAS-II' 22 [2]	Access' 21[11]
Technology	**28 nm**	28 nm	65 nm	180 nm
Array Size	**4 Kb**	384 Kb	73 Kb	2 Kb
Cell Structure	**8T**	6T	6T	10T
Input Precision	**4b**	4/8b	4b	4b
Weight Precision	**4b**	4/8b	4b	4b
Sennsing Method	**Count-type Quantification**	SAR-SS ADC	SAR-ADC	SAR-ADC
Output Precision	**8b**	12/20b	5b	8b
Area Efficiency (GOPS/mm^2)	**999.8**	1640	NA	8.05
Throughput density (GOPS/Kb)	**6.7** [3]	0.56–2	1.28	0.64
Energy Efficency (TOPS/W)	**35.8 (0.7 V, TT)**[1,4] **19.7 (0.9 V, TT)**[1,4]	15.02–94.31 [2]	33.1 [2]	25.9 [2]
Accuracy	**97.24% (MNIST)**	67.26%–67.97% (CIFAR100)	87.9% (CIFAR10)	97.5% (MNIST)

[1]Post-layout Simulation; [2]Chip measurement;
[3]Calculated as 64×4 Mult/cycle × 1 Ops/Mult × 1/9.5 n cycles / 4Kb
[4]Calculated as 64×4 Mult/cycle × 1 Ops/Mult /(VDD × $I_{Average}$); $I_{Average}$:Mean Current in Computing Cycle

with respect to the supply voltage with different process corners. At different process corners, the power consumption of the proposed circuit varies slightly. At VDD = 0.9 V, the energy consumption is 50.7, 40.5, 50.0, 46.5, and 49.4 fJ/bit with TT, SS, FF, SF, and FS, respectively. Fig. 4(c) presents an area breakdown of the proposed SRAM CIM macro. The array of 8T occupied 38.3% of the macro area, the quantization circuits occupied 36%, the encoder occupied 10.6%, and the CCM and other circuits occupied 8.5% and 6.6% of the area, respectively. Fig. 4(d) presents the proportion of energy consumption for the proposed CCM-based CIM. Due to the use of high-speed clocks, the quantization circuits accounted for the largest proportion (66.3%) of the total power consumption, in which the $CLK_{\_ADC}$ macro accounted for 16.7% of the total power consumption. The complete layout is presented in Fig. 4(e).

E. Comparison With Other Works

Table I shows a comparison between the proposed in-memory multibit multiplication and previous works. The recognition accuracy with LeNet-5 model for the MNIST dataset was similar to [11]. The throughput density of this study was 6.7 GOPS/kb with a 4-kb array, thus exhibiting higher throughput density compared to the previous works [1], [2], [11]. The area efficiency was 999.8 GOPS/mm^2.

IV. CONCLUSION

The issues of calculation accuracy and limited read dimensions in the CIM method have been perplexing researchers. Therefore, in this study, we proposed an 8T-based DSWB that supports 2-D multibit operations. We used a CCM multiplexed as a multibit multiplier to realize multibit multiplication while guaranteeing the linearity of the operation. In addition, to achieve high-efficiency quantification, we employed low-overhead quantization circuits for each row and column. A series of experiments were conducted to analyze the performance, in which the linearity of the proposed circuits exhibited an improvement of 54.4% over TRA computing. It achieved a throughput density of 6.74 GOPS/kb. At supply voltages of 0.9 and 0.7 V, the energy efficiency achieved 19.7 and 35.8 TOPS/W, respectively.

REFERENCES

[1] J. W. Su et al., "A 8-b-precision 6T SRAM computing-in-memory macro using segmented-bitline charge-sharing scheme for AI edge chips," *IEEE J. Solid-State Circuits*, vol. 58, no. 3, pp. 877–892, Mar. 2023.
[2] X. Qiao et al., "A 65 nm 73 kb SRAM-based computing-in-memory macro with dynamic-sparsity controlling," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 69, no. 6, pp. 2977–2981, Jun. 2022.
[3] Y.-C. Chiu et al., "A 4-Kb 1-to-8-bit configurable 6T SRAM-based computation-in-memory unit-macro for CNN-based AI edge processors," *IEEE J. Solid-State Circuits*, vol. 55, no. 10, pp. 2790–2801, Oct. 2020.
[4] J. Zhang et al., "In-memory multibit multiplication based on bitline shifting," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 69, no. 2, pp. 354–358, Feb. 2022.
[5] Z. Lin et al., "Cascade current mirror to improve linearity and consistency in SRAM in-memory computing," *IEEE J. Solid-State Circuits*, vol. 56, no. 8, pp. 2550–2562, Aug. 2021.
[6] Z. Lin et al., "Two-direction in-memory computing based on 10T SRAM with horizontal and vertical decoupled read ports," *IEEE J. Solid-State Circuits*, vol. 56, no. 9, pp. 2832–2844, Sep. 2021.
[7] J.-W. Su et al., "Two-way transpose multibit 6T SRAM computing-in-memory macro for inference-training AI edge chips," *IEEE J. Solid-State Circuits*, vol. 57, no. 2, pp. 609–624, Feb. 2022.
[8] M. E. Sinangil et al., "A 7-nm compute-in-memory SRAM macro supporting multi-bit input, weight and output and achieving 351 TOPS/W and 372.4 GOPS," *IEEE J. Solid-State Circuits*, vol. 56, no. 1, pp. 188–198, Jan. 2021.
[9] K. Xiao et al., "A 28nm 32Kb SRAM computing-in-memory macro with hierarchical capacity attenuator and input sparsity-optimized ADC for 4b MAC operation," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, early access, Jan. 6, 2023, doi: 10.1109/TCSII.2023.3234620.
[10] A. Biswas and A. P. Chandrakasan, "CONV-SRAM: An energy-efficient SRAM with in-memory dot-product computation for low-power convolutional neural networks," *IEEE J. Solid-State Circuits*, vol. 54, no. 1, pp. 217–230, Jan. 2019.
[11] V. T. Nguyen, J.-S. Kim, and J.-W. Lee, "10T SRAM computing-in-memory macros for binary and multibit MAC operation of DNN edge processors," *IEEE Access*, vol. 9, pp. 71262–71276, 2021.
[12] Z. Chen et al., "CAP-RAM: A charge-domain in-memory computing 6T-SRAM for accurate and precision-programmable CNN inference," *IEEE J. Solid-State Circuits*, vol. 56, no. 6, pp. 1924–1935, Jun. 2021.
[13] A. Jaiswal, I. Chakraborty, A. Agrawal, and K. Roy, "8T SRAM cell as a multibit dot-product engine for beyond von Neumann computing," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 27, no. 11, pp. 2556–2567, Nov. 2019.
[14] Q. Chen, C. C. Boon, Q. Liu, and Y. Liang, "A single-channel voltage-scalable 8-GS/s 8-b > 37.5-dB SNDR time-domain ADC with asynchronous pipeline successive approximation in 28-nm CMOS," *IEEE J. Solid-State Circuits*, early access, Dec. 29, 2022, doi: 10.1109/JSSC.2022.3230697.