This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

An RRAM-Based Computing-in-Memory Macro With Low-Power Readout/Hold Circuits and Activation Differential Strategy for AdderNet

Zhihang Qian, Shengzhe Yan, Zhuoyu Dai, Zeyu Guo, Zhaori Cong, Yifan He, Chunmeng Dou, Feng Zhang, Jinshan Yue, and Yongpan Liu

***Abstract*—AdderNet is an innovative neural network (NN) structure that substitutes multiplications with additions in convolutional operations, while computing-in-memory (CIM) is an efficient architecture that tackles the memory bottleneck for von Neumann architectures. Previous work has explored the SRAM-based CIM AdderNet circuits and demonstrates high energy efficiency. However, it still suffers low storage density, repetitive readout, and redundant comparisons. In this brief, an RRAM-based CIM macro is proposed for efficient AdderNet with the following innovations. First, RRAM cells are adopted to replace SRAM for high-density weight storage. A low-power readout and hold circuit is proposed to save redundant read power of weight data held for multiple cycles. Second, an 8-bit comparator with an early-stop strategy is proposed to compare 8-bit activations and weights in one cycle. Third, an activation (ACT) differential strategy is proposed to reduce redundant comparisons. The proposed 28-nm RRAM CIM macro achieves 12.8-TOPS/mm^2 peak area efficiency and 126-TOPS/W peak energy efficiency, which is 3.0× and 1.2× compared with the state-of-the-art AdderNet CIM macro.**

***Index Terms*—AdderNet, computing-in-memory (CIM), efficiency, neural network (NN), RRAM.**

I. Introduction

Convolutional neural networks (CNNs) have been widely adopted in artificial intelligence tasks such as image classification and object detection [1], [2]. However, CNNs are typically memory-intensive and computation-intensive [3], which limits the applications on low-power edge devices. The computing-in-memory (CIM) architecture is proposed to perform in situ computation by integrating computing circuits into the memory array. Previous CIM chips [4], [5], [6], [7] have demonstrated advanced energy efficiency over the conventional von Neumann accelerators.

To further reduce the computation overhead, various techniques have been proposed, including data pruning [8], low-bit quantization [9], and matrix transformation (such as Winograd [10]). Among them, one effective technique is the AdderNet [11], which removes the power-consuming multiplications to save power. As shown in Fig. 1, different from the original cross correlation (i.e., multiply-and-accumulation and the Euclidean distance) in original CNNs, AdderNet adopts the $L1$-distance, which calculates the sum of the absolute differences to measure the similarity between weights and activations (ACT). AdderNet replaces the convolution kernel with the adder kernel, and suits the same applications as traditional CNN. The accuracy loss is negligible compared to the original CNN model (0.4% on the CIFAR-10 dataset with the ResNet-20 model and 1.3% on the ImageNet dataset with the ResNet-50 model [11]).

Previous research has explored chip design for AdderNet using both conventional and CIM architectures. A straightforward AdderNet hardware architecture [12] is proposed by calculating the absolute difference between ACTs and weights ($\sum|x_i - w_i|$) through two multibit adders and one multiplexer, which is energy-inefficient and area-inefficient. Thus, an area-efficient AdderNet architecture [13] is proposed by replacing two adders and one multiplexer with one adder and an XOR gate. It computes the subtraction result of ACT and weight data using a single adder and then calculates its absolute value. In these two works, at least one multibit adder is still needed to calculate the absolute difference between ACTs and weights.

Recently, [14] proposed an SRAM-CIM-based macro for the Addernet algorithm. Following (1), it converts the absolute difference ($|x_i - w_i|$) to the addition/subtraction between the ACT (x_i), weight (w_i), and their minimal value ($\min\{x_i, w_i\}$). Though it appears to be more complex, it's noticed that the $\sum x_i$ result can be shared by multiple channels of w_i, thus the power/area overhead is negligible. Furthermore, $\sum w_i$ can be precalculated offline since the weight data are fixed. Therefore, the real computational workload is mainly $\sum \min\{x_i, w_i\}$, which requires lower power/area overhead than $\sum|x_i - w_i|$

$$\sum|x_i - w_i| = \sum x_i + \sum w_i - 2\sum \min\{x_i, w_i\}. \tag{1}$$

However, there are still several remaining problems. First, using SRAM as weight storage leads to low storage density. Second, weights stored in SRAM cells need to be repetitively read out since the readout and minimum value selection operations are bit serial. Furthermore, previous works mainly focus on the design of AdderNet operations, ignoring the co-optimization utilizing ACT/weight data characteristics. There exist numerous redundant comparisons without considering the similarity between neighboring ACTs and weights.

To address these problems, this brief proposes an RRAM-based CIM macro for AdderNet applications. First, 1T1R RRAM cells are adopted to replace SRAM cells for higher storage density. We propose a low-power readout and hold circuit to save redundant read power of weight data held for multiple cycles. Second, an 8-bit comparator with an early-stop strategy is proposed to compare 8-bit activations and weights in one cycle. Third, an ACT differential strategy (ADS) is proposed to utilize the comparison results (COMP) of the previous cycle to avoid redundant comparisons. In summary, the main contributions of this work are as follows.

1) Propose a low-power RRAM readout and hold circuit with 67.8% lower read and retention power compared to the SRAM-based design [14].
2) Design an 8-bit comparator with an early-stop strategy and an ADS reducing comparison power by 41.9%.
3) Implement an RRAM-based CIM macro with 3.0× peak area efficiency and 1.2× peak energy efficiency compared to the state-of-the-art SRAM-based AdderNet chip [14].

Received 3 November 2024; revised 7 January 2025 and 15 February 2025; accepted 25 February 2025. This work was supported in part by the National Key Research and Development Program of China under Grant 2023YFB4402400; and in part by the National Natural Science Foundation of China under Grant 92464201, Grant 92464203, Grant U2341218, Grant 92364202, and Grant 62204256. *(Corresponding author: Jinshan Yue.)*

Zhihang Qian, Shengzhe Yan, Zhuoyu Dai, Zeyu Guo, Zhaori Cong, Chunmeng Dou, Feng Zhang, and Jinshan Yue are with the State Key Laboratory of Fabrication Technologies for Integrated Circuits, Institute of Microelectronics of the Chinese Academy of Sciences, Beijing 100029, China, and also with the School of Integrated Circuits, University of Chinese Academy of Sciences, Beijing 100049, China (e-mail: yuejinshan@ime.ac.cn).

Yifan He and Yongpan Liu are with the Department of Electronic Engineering, Tsinghua University, Beijing 100084, China.

This article has supplementary downloadable material available at https://doi.org/10.1109/TVLSI.2025.3546684, provided by the authors.

Digital Object Identifier 10.1109/TVLSI.2025.3546684

1063-8210 © 2025 IEEE. All rights reserved, including rights for text and data mining, and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: KU Leuven Libraries. Downloaded on April 17,2025 at 14:50:35 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

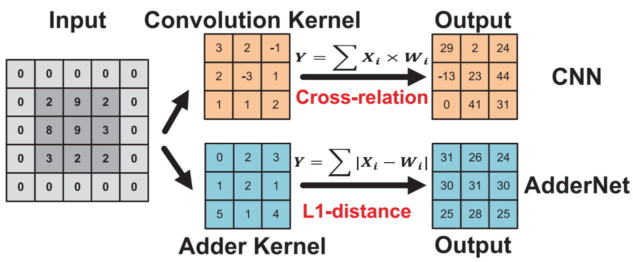

Fig. 1. Example of the difference between AdderNet and traditional CNN.

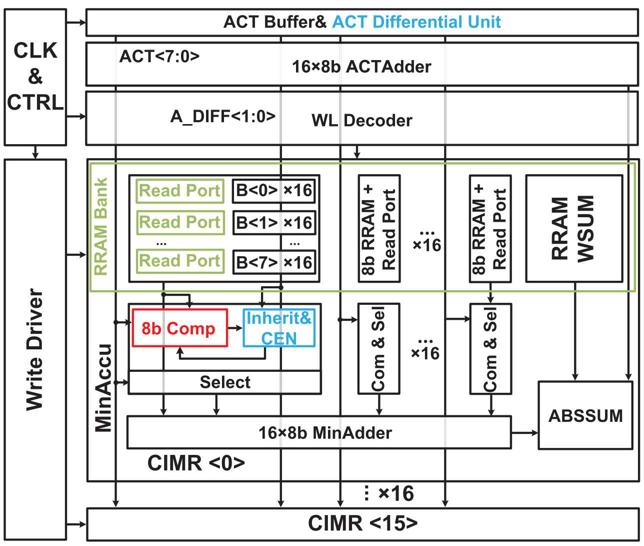

Fig. 2. Overall architecture of the proposed RRAM CIM AdderNet design.

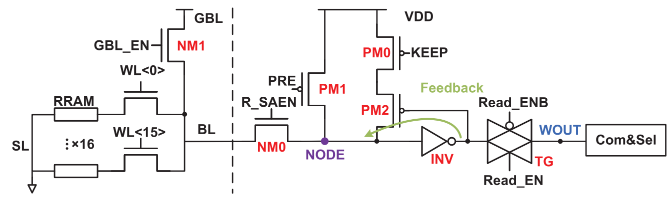

Fig. 3. Schematic of the proposed low-power readout and hold circuit.

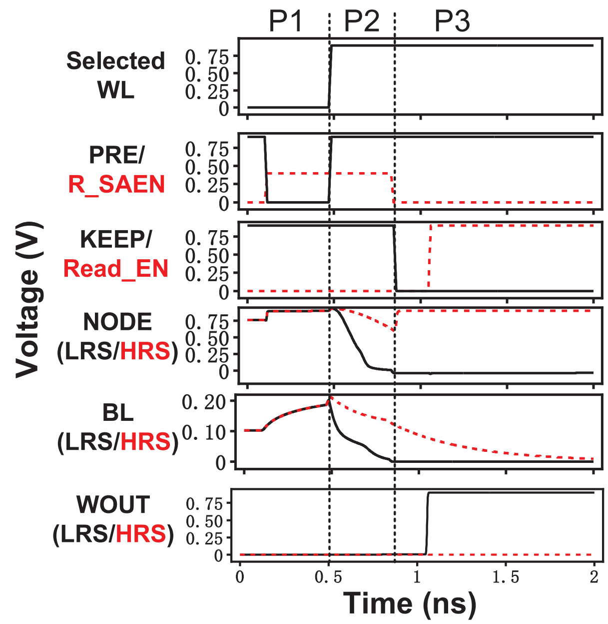

Fig. 4. Simulation waveform of the proposed readout and hold circuit.

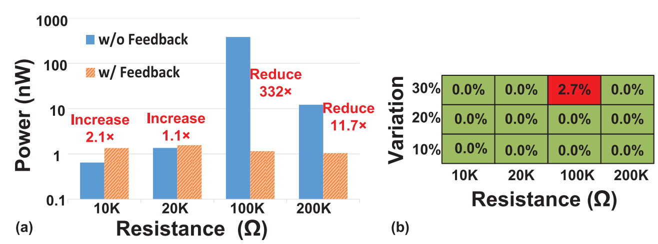

Fig. 5. (a) Retention power consumption of the proposed readout circuit. (b) Readout error rates of the proposed readout circuit.

II. Proposed RRAM CIM AdderNet Architecture

The overall architecture of the proposed RRAM CIM AdderNet design is shown in Fig. 2. It consists of 16 CIM rows (CIMRs) for the AdderNet vector computation [as shown in (1)], an activation (ACT) buffer with an ACT differential unit that receives activations and differential signals, and the timing control circuits. Each CIMR contains an RRAM bank, a minimum value accumulation unit (MinAccu), and an absolute difference summation unit (ABSSUM). The RRAM bank has 16 RRAM subarrays, while each subarray owns eight read ports to simultaneously read 8-bit weight from its 8×16 RRAM cells. The precalculated summation of weights ($\sum w_i$) is stored locally in additional RRAM cells, i.e., the weight summation unit (WSUM). The MinAccu unit consists of 16 compare and select units, coupled with a minimum adder tree (MinAdder) to accumulate the 16×8 b minimum values. Then, the ABSSUM unit performs the addition/subtraction among $\sum x_i$ from the activation adder unit (ACTAdder), $\sum w_i$ from the WSUM units, and $\sum \min(x_i, w_i)$ from the MinAccu unit according to (1).

A. Low-Power RRAM Readout and Hold Circuits

The 1T1R RRAM is adopted as weight storage in this work to achieve higher storage density than the previous 8T SRAM cells [14]. Single-level cells are used to alleviate the accuracy loss caused by process variations. However, the straightforward RRAM-based design needs sense amplifiers to read weights in each cycle, which is power consuming. In this work, a low-power readout and hold circuit is designed as shown in Fig. 3. After the readout operation, the weight data are maintained for multiple cycles to perform computations with multiple ACTs, avoiding repetitive readout operations.

The waveform of the proposed readout and hold circuit is illustrated in Fig. 4. The readout process comprises three phases: precharge, evaluate, and keep phases. In the precharge phase, PRE is pulled down and R_SAEN is pulled up to precharge BL and NODE. During the evaluation phase, the selected WL is pulled up to discharge BL and NODE. By adjusting the pulsewidth of R_SAEN, the discharging time is fine-tuned to either completely discharge NODE of the low-resistance-state (LRS) cells, or slightly discharge NODE of the high-resistance-state (HRS) cells due to different discharging rates. In the final phase, KEEP is pulled down to recharge the NODE of the HRS cells. Then, the transmission gate (TG) is opened to read weights to the subsequent compare and select units.

The nonideality of RRAM devices poses significant challenges to the circuit performance, notably impacting power consumption and readout accuracy. The dynamic voltage sense amplifier proposed in [15] distinguishes HRS and LRS based on different discharge rates of the discharging node (VINV). However, as the resistance of HRS cells varies, the voltage of VINV fluctuates. When the voltage of VINV is in the transition region of the inverter, the power consumption

Authorized licensed use limited to: KU Leuven Libraries. Downloaded on April 17,2025 at 14:50:35 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

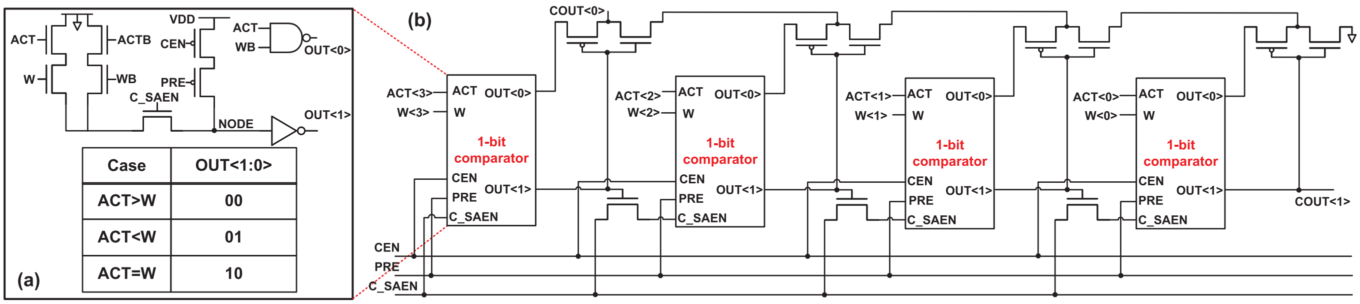

Fig. 6. (a) Schematic of the 1-bit comparator and the truth table. (b) Cascaded relationship of the 8-bit comparator (four stages shown as an example).

increases, and the readout error occurs. To mitigate these effects, the proposed readout circuit incorporates a feedback path to recharge the NODE of HRS cells, which effectively reduces readout errors and static power consumption of the inverter. In the keep phase, the selected WL maintains high to fully discharge BL, enlarging the voltage difference between NODE and BL, which can increase the discharging rate to fully discharge LRS cells even under a large resistance variation. Since this work mainly focuses on the read function, the RRAM device model is simplified as a resistance model with corresponding Gaussian variations, which are set to 10%~30% variations to reflect the device imperfections according to practical RRAM measurement results [16]. Fig. 5(a) illustrates the retention power consumption of the proposed readout circuits, demonstrating $> 11.7\times$ reduction in the retention power consumption of HRS cells and a slight increase in the retention power consumption of LRS due to the feedback path. Overall, the readout and hold circuit has 67.8% lower read and retention power consumption in subsequent simulations. Fig. 5(b) depicts the readout error of RRAM cells (LRS =10/20 kΩ and HRS =100/200 kΩ) across different variation states (10%–30% Gaussian variations). The proposed readout circuit achieves no error for variation ≤ 20% and only incurs a 2.7% error rate for HRS =100 kΩ with 30% variation, highlighting a substantial reduction in readout errors compared with those presented in [15].

B. 8-Bit Comparator With an Early-Stop Strategy

Following the readout and hold circuit, an 8-bit comparator is needed to compare the 8-bit ACT and weight in one cycle. As shown in Fig. 6(a), each 1-bit comparator is comprised of a dynamic logic circuit to determine whether the ACT is equal to the weight and a NAND gate to determine which one of them is the minimum value. Fig. 6(b) shows the cascaded eight 1-bit comparators (four stages shown as an example). With the early-stop strategy, the comparison between ACT and weight continues until ACT⟨i⟩ and W⟨i⟩ are unequal. The OUT⟨1⟩ of the former 1-bit comparator determines the C_SAEN input of the latter comparator. If ACT and weight are unequal at the current bit, the OUT⟨1⟩ of the current 1-bit comparator will be "0," stopping the C_SAEN signal and avoiding activating the next 1-bit comparator. Thus, the OUT⟨1⟩ of the next 1-bit comparator remains "0," even if ACT and weight are equal at this bit, which means the 8-bit comparator stops at the current bit. The COUT⟨1⟩ will be "0," and COUT⟨0⟩ is equal to OUT⟨0⟩ of the current 1-bit comparator. The comparators with no discharge can be reused in the next cycle, reducing the precharge power consumption of the 8-bit comparator.

Compared with the bit-serial comparator in the previous work [14], the proposed parallel 8-bit comparator shows lower latency. Fig. 7(a) shows the latency comparison between the proposed 8-bit comparator and the bit-serial comparator. The latency of the bit-serial comparator is limited by the readout latency of weights, resulting in a lower operating frequency. Fig. 7(b) depicts the power consumption of the 8-bit comparator that stops at different bits and the precharge power in the next comparison. Analyzed with a quantized ResNet-20 model on the CIFAR-10 dataset, only three bits of the 8-bit data need to be compared on average, which reduces the comparison power consumption by 36.3% and the precharge power by 57.8%.

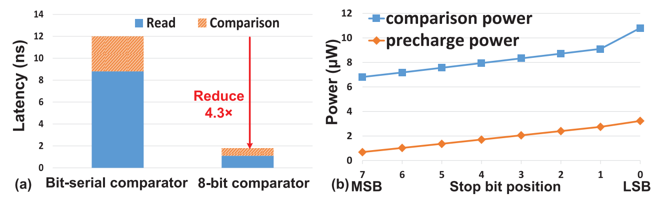

Fig. 7. (a) Latency improvement of the proposed 8-bit comparator. (b) Power savings of the early-stop strategy.

C. ACT Differential Strategy for Further Power Saving

Algorithm 1 ACT Differential Strategy

Input: input activation ACT, weight W, 8-bit comparator output $COUT$, previous comparison result $COMP_{\text{pre}}$,

Output: current comparison result $COMP_{cur}$, comparison enable CEN,

Step 1: Generate ACT differential result

A_DIFF ← compare $\left(ACT_{cur}, ACT_{\text{pre}}\right)$;

Step 2: Decide whether to compare

CEN ← cen_generate $\left(COMP_{pre}, A\_DIFF\right)$;

Step 3: Perform 8-bit comparison if CEN = '0'

$COUT$ ← 8 bit_comparator (CEN, ACT, W);

Step 4: Inherit comparison result if CEN = '1'

Inherit ← select $\left(COMP_{\text{pre}}, A\_DIFF\right)$;

Step 5: Update comparison result according to CEN

$COMP_{cur}$ ← select($COUT$, Inherit, CEN).

Motivated by the similarity of local pixels in the images, we propose an ADS to reduce redundant comparisons. When the weight is read out and held, it needs to be compared with 2-D activations through height and width, which feature data similarity in a local region. The previous COMP result with the ACT larger than the weight can be reused if the current ACT is larger than the previous activation. Details of ADS are presented in Algorithm 1. First, the relationships (larger or smaller) between adjacent activations are generated by comparing the current ACT with the previous-cycle activation. This ACT differential signal (represented as A_DIFF) is shared by 16 CIMRs. Second, the comparison-enable (CEN) signal is generated based on the previous-cycle COMP and the A_DIFF signal, with the operation logic shown in Fig. 8(a). The comparison can be skipped when either COMP⟨1⟩ or A_DIFF⟨1⟩ equals "1," or when COMP⟨0⟩ and A_DIFF⟨0⟩ are identical. Then, the COMP between the current ACT and weight can be directly obtained without

Authorized licensed use limited to: KU Leuven Libraries. Downloaded on April 17,2025 at 14:50:35 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

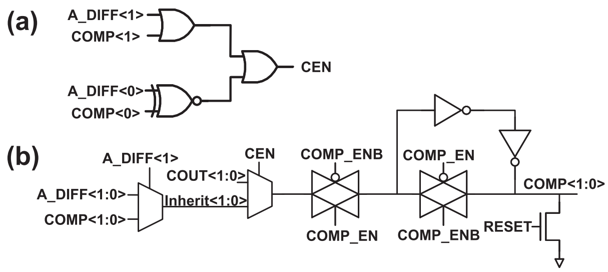

Fig. 8. Schematics of (a) CEN generate circuit and (b) Inherit circuit.

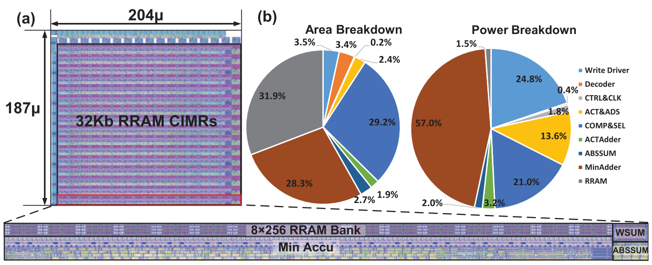

Fig. 9. (a) Layout of the proposed RRAM CIM macro. (b) Power breakdown and area breakdown.

the actual comparison operation. Then, CEN is pulled up to prevent the precharge of 1-bit comparators, and the COMP can be obtained through the circuit shown in Fig. 8(b). Otherwise, CEN is pulled down normally to compare activations and weights.

The reduction in the number of comparisons using the ADS is analyzed using a quantized ResNet-20 model on the CIFAR-10 dataset [14]. The proposed ADS effectively manages to reduce the number of comparisons by 84.85%. The area and power overhead, 7.0% and 7.9%, of the inherit and CEN circuits and the ACT differential unit are acceptable relative to the power reduction achieved in 8-bit comparators through the elimination of redundant comparisons. The energy efficiency improved by the ADS will be elaborated in Section III.

III. Experimental Results

A. Experimental Setup

The proposed RRAM CIM design for AdderNet is implemented and simulated under a 28-nm commercial CMOS technology. The digital circuits (such as the WL decoder and ACT differential unit) are designed using Verilog, then synthesized by the Synopsys Design Compiler, and placed and routed by the Cadence Innovus using high-VT standard cells. The RRAM cells are assumed at 10/100 kΩ for LRS/HRS with a 10% Gaussian variation ensured by the write verification technique. The experimental results are based on circuit simulations using the Cadence Spectre.

B. Experimental Results and Analysis

The layout of the proposed RRAM CIM macro is shown in Fig. 9(a) with an area of 0.038 mm^2. The area breakdown and power breakdown are shown in Fig. 9(b). The 32-kb RRAM cells with readout circuits take 30.8% area of the macro. Compared with [14], it achieves 2× storage capacity with only a 53.7% increase in area. Over half of the area is occupied by the MinAccu units, with 28.2% dedicated to the compare and select units, and 27.3% allocated to the adder trees. The power breakdown is evaluated at 0.9 V, 1 GHz, 90% input sparsity, and 10% toggle rate. The peak performance is 0.486 TOPS at 0.9 V, 1 GHz. The RRAM readout operations take two cycles and then the readout weight data are kept for multiple cycles of CIM operations. This work achieves 4.3× working frequency compared with the SRAM-based design [14]. The power consumption of the RRAM array and the WL decoder is effectively reduced due to the low-power readout and hold circuits and low activity of the WL decoder. The macro achieves 126 TOPS/W peak energy efficiency at 0.8 V, 667 MHz, 81% input sparsity, and 18% toggle rate.

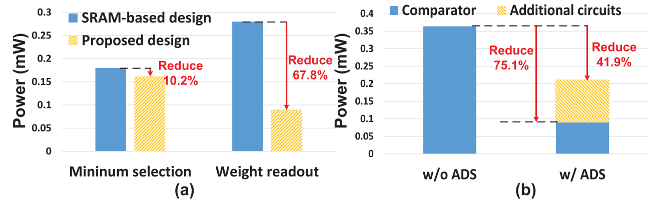

Fig. 10. (a) Power comparison of weight readout and minimum selection circuits. (b) Comparator power with/without the ADS.

TABLE I
Comparison With Existing RRAM CIM and SRAM CIM Macros

	ISSCC2023 [17]	TCAS-II2023 [15]	ISSCC2023 [14]	**This Work**
Technology	22nm	28nm	28nm	28nm
Storage	RRAM	RRAM	SRAM	**RRAM**
Verification	Fabricated	Simulation	Fabricated	Simulation
CIM Type	Analog CIM	Digital CIM	Digital CIM	Digital CIM
Operation	Multiply+ADD	Multiply+ADD	ABS+ADD	**ABS+ADD**
Capacity	4MB	16Kb	16Kb	**32Kb**
Area (mm^2)	-	0.011	0.028	0.038
Voltage (V)	0.7-0.8	0.60-0.90	0.54-0.90	0.80-0.90
Input Bit	1b-8b	1b	2b-8b	8b
Weight Bit	1b-8b	1b	2b-8b	8b
Area Efficiency (TOPS/mm^2@8b)	-	1.2 (0.9V)[1]	4.2 (0.9V)	**12.8 (0.9V)**[2]
Energy Efficiency (TOPS/W@8b)	68.9 (0.7V)	30.7 (0.6V)[1]	102 (0.54V)	**126 (0.8V)**[2,3]
CIFAR-10 Accuracy	91.9%	90.10%	-	91.55%
ImageNet Accuracy	70.9%	-	74.5%	74.5%

[1] Normalized to 8b precision.
[2] One addition and one comparison are counted as two operations (OPs).
[3] The estimation of energy efficiency excludes the RRAM program (set/reset) process.

Fig. 10(a) shows the power comparison of the weight readout and the minimum selection operations between the proposed RRAM-based design and the previous SRAM-based design [14]. This work shows 67.8% lower readout power since the repetitive readout is eliminated. The power of compare and select operations of the proposed design is 10.2% lower than the minimum-value selection of the SRAM-based design while operating in a 2.9× higher frequency. Fig. 10(b) depicts the power comparison of the comparator with or without the ADS. The power of the comparator is effectively reduced by 75.1% and the total power consumption (including the overhead of ACT differential unit) is reduced by 41.9%, demonstrating the effectiveness of the ADS.

Table I lists comparison of the proposed design with the existing RRAM-based and SRAM-based CIM designs. Note that the evaluation of the whole NN models on Cifar-10/ImageNet requires weight mapping on multiple macros. Compared with the RRAM-based analog/digital CIM designs [15], [17] for traditional CNNs, this work achieves 4.1×/1.8× higher energy efficiency thanks to multiplyless operations. Compared with the SRAM-based design for AdderNet, this work achieves 1.2× higher energy efficiency and 3.0× higher area efficiency.

IV. Conclusion

This work proposes an RRAM-based CIM macro with low-power readout and hold circuit and ADS. The proposed low-power readout and hold circuit enables weights to be read and held for multiple cycles with low power consumption. Thus, the power consumption of the RRAM array is reduced by 67.8%. The proposed 8-bit comparator achieves low latency, which improves the macro throughput. The proposed 8-bit comparator is designed with an early-stop strategy. Together with the ADS, the comparison power consumption is reduced by 4.0×. Eventually, the macro achieves 12.8-TOPS/mm^2 peak area efficiency and 126-TOPS/W peak energy efficiency, which is 3.0× and 1.2× higher than the state-of-the-art SRAM-based design.

Authorized licensed use limited to: KU Leuven Libraries. Downloaded on April 17,2025 at 14:50:35 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Except for RRAM, other emerging nonvolatile devices with a high density and a high $R_{\mathrm{high}}/R_{\mathrm{low}}$ ratio may also be adopted in the proposed design without much circuitry revision, which is worth future exploration.

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017.

[2] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[3] X. Xu et al., "Scaling for edge inference of deep neural networks," *Nature Electron.*, vol. 1, no. 4, pp. 216–222, Apr. 2018.

[4] C.-X. Xue et al., "16.1 A 22nm 4Mb 8b-precision ReRAM computing-in-memory macro with 11.91 to 195.7TOPS/W for Tiny AI edge devices," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2021, pp. 245–247.

[5] Q. Liu et al., "33.2 A fully integrated analog ReRAM based 78.4TOPS/W compute-in-memory chip with fully parallel MAC computing," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2020, pp. 500–502.

[6] H. Jia et al.., "15.1 a programmable neural-network inference accelerator based on scalable in-memory computing," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, vol. 64, Feb. 2021, pp. 236–238.

[7] H. Fujiwara et al., "A 5-nm 254-TOPS/W 221-TOPS/mm^2 fully-digital computing-in-memory macro supporting wide-range dynamic-voltage-frequency scaling and simultaneous MAC and write operations," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2022, pp. 1–3.

[8] C.-C. Chang, C.-H. Huang, and Y.-S. Chu, "A hardware-friendly pruning approach by exploiting local statistical pruning and fine grain pruning techniques," in *Proc. IEEE Int. Conf. Consum. Electron.-Asia (ICCE-Asia)*, Oct. 2022, pp. 1–3.

[9] Y. Choukroun, E. Kravchik, F. Yang, and P. Kisilev, "Low-bit quantization of neural networks for efficient inference," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW)*, Oct. 2019, pp. 3009–3018.

[10] A. Lavin and S. Gray, "Fast algorithms for convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4013–4021.

[11] H. Chen et al., "AdderNet: Do we really need multiplications in deep learning?," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 1465–1474.

[12] Y. Wang et al., "AdderNet and its minimalist hardware design for energy-efficient artificial intelligence," 2021, *arXiv:2101.10015*.

[13] G. Seo and S. Ryu, "Area-efficient AdderNet hardware accelerator with merged adder tree structure," *IEICE Electron. Exp.*, vol. 20, no. 23, 2023, Art. no. 20230427.

[14] Y. He et al., "7.3 a 28nm 38-to-102-TOPS/W 8b multiply-less approximate digital SRAM compute-in-memory macro for neural-network inference," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2023, pp. 130–132.

[15] Y. He et al., "An RRAM-based digital computing-in-memory macro with dynamic voltage sense amplifier and sparse-aware approximate adder tree," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 70, no. 2, pp. 416–420, Feb. 2023.

[16] Y. Huang, Y. He, J. Yue, H. Yang, and Y. Liu, "Accuracy optimization with the framework of non-volatile computing-in-memory systems," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 69, no. 2, pp. 518–529, Feb. 2022.

[17] W.-H. Huang et al., "A nonvolatile al-edge processor with 4MB SLC-MLC hybrid-mode ReRAM compute-in-memory macro and 51.4-251TOPS/W," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2023, pp. 15–17.

Authorized licensed use limited to: KU Leuven Libraries. Downloaded on April 17,2025 at 14:50:35 UTC from IEEE Xplore. Restrictions apply.