A Multi-bit Near-RRAM based Computing Macro with Highly Computing Parallelism for CNN Application

Kuan-Chih Lin, Hao Zuo, Hsiang-Yu Wang, Yuan-Ping Huang, Ci-Hao Wu, Yan-Cheng Guo, Shyh-Jye Jou, Tuo-Hung Hou, Tian-Sheuan Chang
Institute of Electronics, National Yang Ming Chiao Tung University, Hsinchu, Taiwan

***Abstract*—Resistive random-access memory (RRAM) based compute-in-memory (CIM) is an emerging approach to address the demand for practical implementation of artificial intelligence (AI) on resource constrained edge devices by reducing the power-hungry data transfer between memory and processing unit. However, the state-of-the-art RRAM CIM designs fail to strike a balance between precision, energy efficiency, throughput, and latency. This work merges the techniques of CIM and compute-near-memory (CNM) to deliver high precision, high energy efficiency, high throughput, and low latency. In this paper, a 256Kb RRAM based CNM macro fabricated in TSMC 40 nm process is presented featuring: 1) opposite weight mapping with variation-robust SA to mitigate the impact of RRAM device variations on MAC (Multiply-Accumulate) results; 2) switched-capacitor-based analog multiplication circuit to achieve highly parallel computing of 128 4-bit by 4-bit MAC result with low power consumption and high operation speed; and 3) joint optimization of hardware and software to compensate for the accuracy loss after considering the non-idealities of circuits. The macro achieves a low latency of 17ns and high energy efficiency of 71 TOPS/W for MAC operations with 4-bit input, 4-bit weight and 4-bit output precision. It is used to accelerate the convolution process in the Light-CSPDenseNet AI model, resulting in a high accuracy of 86.33% on Visual Wake Words dataset.**

Keywords—Artificial intelligence, autonomous driving, computing-in-memory, nonvolatile memory, resistive random access memory

I. Introduction

In recent years, Artificial Intelligence (AI) has become an important technique for many applications such as speech recognition, image processing, and autonomous driving. However, using the conventional processors based on the von Neumann structure for running AI applications face challenges in achieving low energy and short latency [1], due to the memory wall involving data movement between memory and processing elements. In response to this issue, the concept of compute-in-memory (CIM) has been developed and aims to overcome the von Neumann bottleneck by enabling highly parallel computing, suppressing the amount of intermediate data, performing multiple MAC operations within a single cycle, and avoiding excessive data movement between memory and processing elements.

Among various kinds of CIM architectures, such as [2]-[6] etc., nonvolatile-memory-based (NVM-based) CIM, such as resistive random-access memory (RRAM) based CIM, provides the benefit of nonvolatile storage, which eliminates the need for reloading data during the wake-up operation of edge devices, thereby reducing latency and energy consumption for the frequently-standby AI applications.

To support AI tasks with complicated AI models in real-time applications, RRAM-based CIM needs to be equipped with following capabilities: (1) supporting high-precision multi-bit MAC operations, (2) low computing latency, (3) and high throughput for processing large amount of data. However, there are numbers of challenges involved in designing RRAM CIM macros with aforementioned performance: (1) the variation of RRAM device and IR drop effect generated by parasitic resistance in RRAM array that may cause inaccurate MAC results and accuracy drop, in turn, limiting wordline-level parallelism [7] and precision, (2) designing highly parallel MAC computing circuits with fast operation speeds and minimizing accumulated errors caused by factors such as quantization of intermediate values and non-ideal behavior in the circuitry, (3) maintaining high energy efficiency while overcoming the above-mentioned challenges.

In this paper, we propose the first RRAM-based computing macro to integrate CIM techniques for reading out weights with the concept of computing-near-memory (CNM) by placing computing circuits near RRAM units. This work satisfies the requirements of low power consumption, short operating times, and high throughput. First, during weight reading, we adopt low pre-charging voltage operation with no DC current and bit-line clamping circuits for power and area saving, and implement opposite weight mapping method with variation-robust SA to prevent variation of RRAM device from affecting MAC results. In addition, column-wise-parallel-in scheme is adopted for equalizing IR drop effect on each weight data on the same row and shortening latency by reducing the time for inputting multi-bit data under multiple cycles. Third, to achieve highly parallel MAC operation with low power consumption and high operation speed, we put together switched-capacitor-based analog multiplication circuit with quantization-error-free accumulator and binary-place-value combiner. Finally, the analog results are sent into SAR-ADC with RELU function to be converted into digital outputs. Non-idealities of circuits are compensated by software optimization for better tradeoff between hardware overhead and accuracy of AI algorithm. Overall, this 256Kb RRAM CIM macro that is designed in 40 nm process can process 128 4-bit by 4-bit MAC operation with 4-bit output and can achieve a short latency of 17 ns and high energy efficiency of 71 TOPS/W.

II. Overall Architecture of Proposed CIM Macro

Fig. 1 shows the block diagram of overall architecture of the proposed macro. 128 sets of 4-bit input data are parallel sent into multiplication circuits and multiplied with 128 sets of 4-bit weight data stored in RRAM array in analog domain.

979-8-3503-4859-0/DATE24/© 2024 EDAA

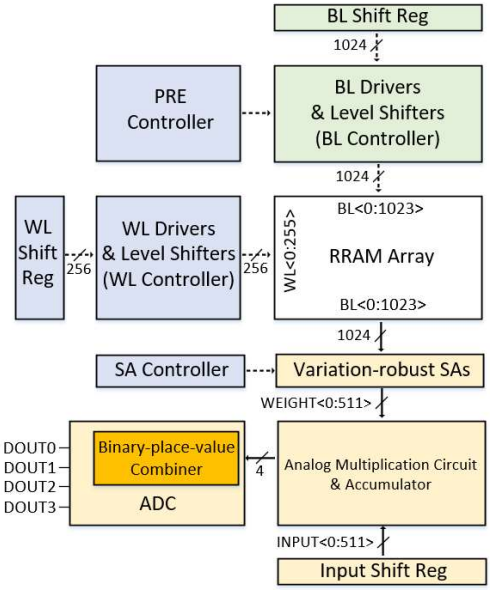

Fig. 1. Block diagram of the proposed 256 by 1024 RRAM CIM macro.

In this macro, we use voltage-sensing method to read out weight data stored in RRAM array. The signals of weight data along with the signals of input data control the switches in the analog multiplication circuit and accumulator to perform analog MAC operation. Input data is shifted into the input shift register before starting of CIM operation. The analog multiplication circuit and the accumulator generate partial MAC results and transmit the results to the binary-place-value combiner for producing final MAC results with the information of input data's binary place values. Finally, the analog to digital converter (ADC) converts analog MAC results to 4-bit digital output data.

III. Weight Mapping Method and Weight Computing Path Arrangement

A. Opposite Weight Mapping with Variation-Robust SA

The variation of RRAM cell resistance is a crucial problem in RRAM-based CIM macro designs. It limits the precision, parallel computing capability and computation accuracy of a macro. Under these limitations, it is difficult for RRAM CIM macro to support complex AI algorithms. Therefore, it is vital to find an efficient approach to restrict the variation of RRAM cell resistance without increasing too much hardware overhead.

Under this circumstance, we drew inspiration from the concept presented in [8], wherein weights are stored differentially within 2T2R architecture. Building upon this concept, we implemented similar weight storage principle with variation-robust SA to effectively manage the inherent variations observed in RRAM cell resistance. As shown in Fig. 2, two RRAM cells in opposite states are used to represent one bit of weight data, that is, if the stored value is binary one, the resistance of the memory cell in odd column will be in low resistance state (LRS), while the one in even column will be in high resistance state (HRS). After precharging and discharging all BLs, the voltage of BL connected to LRS cell will be lower, and the voltage of BL connected to HRS cell will be higher. Then, the higher BL voltage will be amplified to a high-reference voltage of 0.6V and the lower BL voltage to the low-reference voltage of 0V using variation-robust SA.

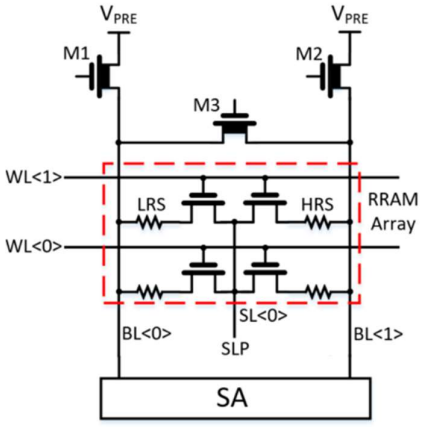

Fig. 2. One column of CSL-based RRAM array and its peripheral circuit.

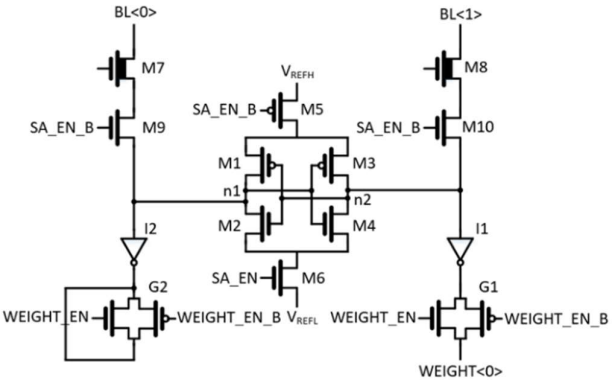

Fig. 3. Schematic of variation-robust SA.

0.6V is chosen as the high-reference voltage because it is the best tradeoff voltage between the speed and power consumption of the SA. After this operation, the unpredictable BL voltages will be settled at a constant reference voltage of 0.6V or 0V, which mitigates the randomness introduced by variations in RRAM cells.

Fig. 3 shows the schematic of variation-robust SA. To prevent damage to the SA from high voltages during RRAM cell forming, we use I/O MOS transistors M7 and M8. These transistors are turned off during weight programming to protect the SA. When performing CIM operations, M7 and M8 are turned on to connect the BLs to the SA. For power saving, M_5 and M_6 are turned off to disable SA while it is not operating. Moreover, the power consumption of SA is related to the capacitance at node n_1 and n_2. Thus, M_9 and M_{10} are used to disconnect n_1 and BL<0> as well as n_2 and BL<1> during SA operation to eliminate the need for SA to charge the large parasitic capacitance on BLs, effectively reducing power consumption.

In addition, there is a tradeoff between power consumption and accuracy of the SA. The mismatch between M_1 to M_4, resulting from process variations, can lead to an imbalance between the capacitance of n_1 and n_2, consequently affecting the amplification result. To reduce the mismatch, it is required to increase the size of transistors. However, the capacitance of n_1 and n_2 will also be increased, and thereby increasing power consumption. To achieve lower power consumption with more tolerance to mismatch, we refer to [9]-[10] for the methodology of joint hardware and software optimization.

B. Low Pre-charging Voltage Operation of Bit-line

According to the aforementioned operation of reading out weight data from memory array, the frequent charging and discharging of BLs consumes significant power. To address this power consumption issue, we propose a low pre-charging voltage operation. Fig. 2 shows the peripheral circuit and a single column of common-source-line-based RRAM array. Lower reference voltage for V_{PRE} are used to reduce charging power; however, there exists a constraint on how low V_{PRE} can be configured. If the BLs are charged to a very low voltage, the voltage difference between two BLs after discharging might not be sufficient for SA to generate accurate signals. Thus, we choose 0.2V as the reference voltage for V_{PRE}, striking a better balance from a design perspective. It's noteworthy that the voltage level of V_{PRE} is intentionally designed to be adjustable for measurement, allowing fine-tuning to maintain signal accuracy and optimize performance.

Additionally, the BL discharging time is also designed to be very short to ensure that BL voltage drop from V_{PRE} remains minimal during each CIM operation. This approach alleviates the power issue associated with the BL charging process. However, in order to maintain sufficient voltage difference between BL voltages for SA to amplify accurately, the discharging period cannot be excessively short. In our macro, we have chosen a discharging period of 200ps, which strikes a favorable balance considering the factors mentioned above. Under this configuration, the voltage difference between two BL is 53.64 mV.

Unlike current mode designs, there is no need for clamping OP to clamp BL voltages at fixed value to sense current from memory cells, thus reducing power and area of peripheral circuit. Moreover, there is no DC current flowing on the BLs during weight data readout process. Therefore, this work consumes less power compared to current mode designs.

C. High Throughput Column-Wise-Parallel-In Scheme

To process 4-bit input data, a column-wise-parallel-in scheme is proposed, shown in Fig.4. Unlike sequential schemes proposed by other works that require multiple cycles to process multi-bit input data, this method processes 4-bit input data simultaneously with 4-bit weight data in one MAC cycle, thereby shortening MAC operation time.

In current mode designs, row-wise operations accumulate current on the same BL, which can lead to limitations in parallelism due to the maximum current capacity of BLs and variations in RRAM resistance leading to sense margin degradation. On the contrary, the proposed scheme only turns on one row of RRAM cells so that the current on each BL is small, and can parallel process up to 128 sets of 4-bit data.

IV. Multi-bit Analog MAC Circuit

To achieve highly parallel MAC operating with low power consumption and high operation speed, based on the passive switched-capacitor method outlined in [11], we implement a switched-capacitor-based analog multiplication circuit along with quantization-error-free accumulator to generate four intermediate products of 128 sets of 4-bit weight and 1-bit input MAC. Unlike [11] that involves cycles to complete one inner product, switched-capacitor-based circuit is duplicated to allow the completion of 128 sets of 4-bit weight and 4-bit input MAC in a single cycle. Then, a binary-place-value combiner is designed for summing the intermediate products with corresponding binary place values of input data.

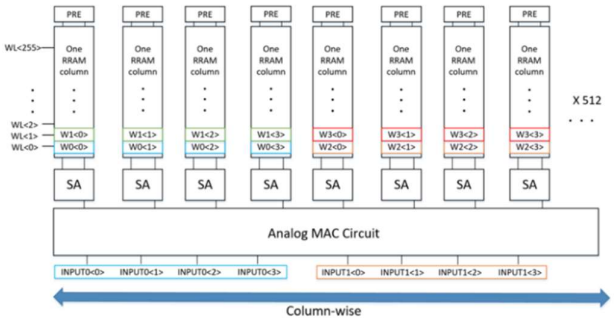

Fig. 4. Block diagram of column-wise-parallel-in scheme.

A. Switched-Capacitor-Based Multiplication Circuit

Fig. 5 shows the schematic of the switch-capacitor-based analog multiplication circuit which carries out 4-bit by 4-bit multiplication in analog domain. It includes four sets of MOS switches and capacitor arrays. The voltages of four metal lines in each array represent four partial products of 4-bit weight and 1-bit input respectively without the binary place values of input data.

Fig. 6 shows how sign magnitude representation is used for signed weight data and the reference voltages. At the beginning of the multiplying operation, the switches that reset metal lines and capacitors to VCMI are turned off to make the voltages of metal lines floating. If the voltage of the other end of capacitor is changed, the voltage difference will be coupled through capacitor to the metal line.

Weight data is read out at the output of SAs and controls the switches to connect capacitors to different reference voltages, while each bit of input data controls the switches to decide whether to pass weight data to each capacitor array or not. Only when the bit of input data is one, it turns on the switches to let weight data control the switches connected to capacitor. On the other hand, if the bit of input data is zero, there will be no voltage difference coupled to metal lines.

According to the sign magnitude representation, the MSB of weight data is the sign bit, and the other three bits are magnitude bits. The sign bit is read out at the output of SA I_4 to control S_1 to S_6 of CAP_ARRAY0. The control methodology is the same for all capacitor arrays, so CAP_ARRAY0 and LSB of weight data 0 are taken as an example. If the sign bit is zero, which means the weight data is positive, S_1, S_3, S_5 are turned on and S_2, S_4, S_6 are turned off. Therefore, C_1, C_2, C_3 are connected to VCMI+VR or VCMI according to the magnitude bits. If the LSB of weight data 0 is one, S_7 will be turned on and S_8 will be turned off, making a voltage difference of +VR between metal line and N_{C1}. In contrast, If the sign bit is one, which means the weight data is negative, S_1, S_3, S_5 are turned off and S_2, S_4, S_6 are turned on. Then, C_1, C_2, C_3 are connected to VCMI-VR or VCMI according to the magnitude bits. If the LSB of weight data 0 is one, S_7 will be turned on and S_8 will be turned off, making a voltage difference of -VR between metal line and NC1.The voltage difference couples to the metal line through C1, as expressed in Eq. (1) and Eq. (2), where the voltage change of metal line is given by VR multiplied by 1/7.

$$Q = CV \tag{1}$$

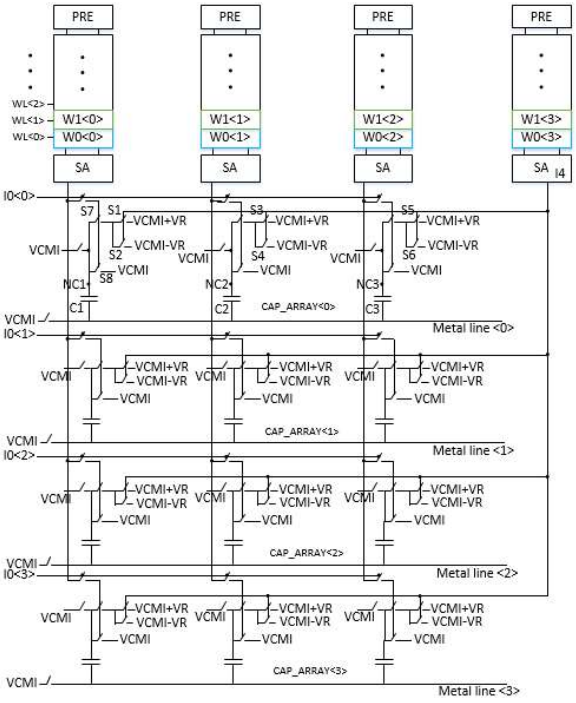

Fig. 5. Schematic of the switched-capacitor-based multiplication circuit.

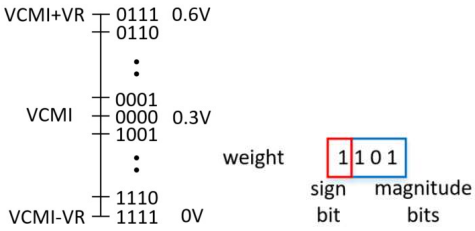

Fig. 6. Sign representation and analog-to-digital conversion table.

$$\Delta V_{metal\ line} = \frac{C_1\Delta V_1 + C_2\Delta V_2 + C_3\Delta V_3}{C_{total}} \quad (2)$$

In Eq. (2), the voltage difference between N_{C1} and VCMI to N_{C3} and VCMI are multiplied with corresponding capacitance of C_1, C_2, C_3 with the ratio of 1:2:4 which represents the place values of weight data. Finally, there are four resulting voltages representing 4-bit weight with 1-bit input partial product without the information of place values of input data at the four metal lines from CAP_ARRAY0 to CAP_ARRAY3.

B. Quantization-Error-Free Accumulator

Fig. 7 illustrates the block diagram of the quantization-error-free accumulator. In this configuration, 128 aforementioned analog multiplication circuits, each consisting of four sets of MOS switch and capacitor arrays, are used to simultaneously compute 4-bit weight by 4- bit input data. To accumulate partial products of 4-bit weight and 1-bit input multiplication, according to Eq. (3), the accumulator connects the four metal lines that are in the same row in the capacitor arrays of each multiplication circuit.

$$\Delta V_{accumulation\ line} = \frac{C_M\Delta V_1 + C_M\Delta V_2 + \ldots + C_M\Delta V_{128}}{128 \times C_M} \quad (3)$$

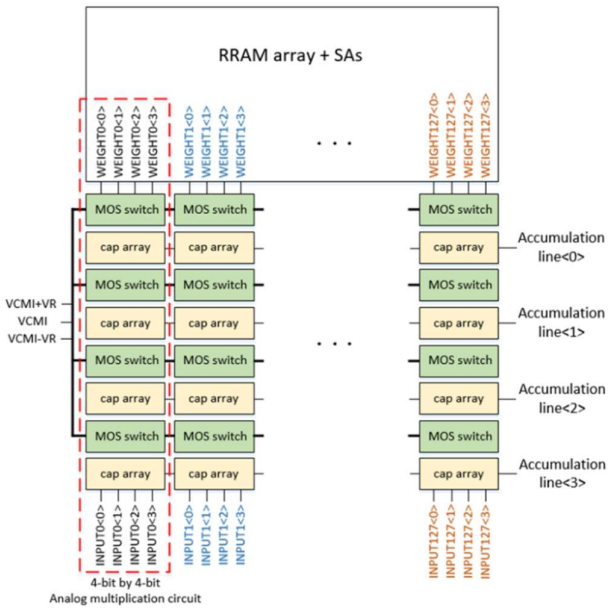

Fig. 7. Block diagram of quantization-error-free accumulator.

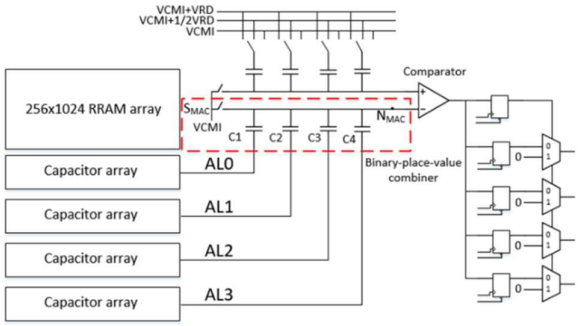

Fig. 8. Block diagram of binary-place-value combiner.

$$V_{accumulation\ line} = VCMI + \Delta V_{accumulation\ line} \quad (4)$$

In Eq. (3), C_M denotes the total capacitance of an individual capacitor array. The accumulation results are free from quantization errors, for there is no need to convert the analog results of each multiplication to digital values before the accumulation process. In addition, the utilization of the accumulator eliminates the necessity for additional hardware to convert intermediate analog results into digital values and store them, resulting in lower power consumption and a reduced area.

C. Binary-Place-Value Combiner

As shown in Fig. 8, the four accumulation lines AL0 to AL3 are connected to four capacitors with capacitance ratio of 1:2:4:8, representing the binary place values of input data. The four capacitors couple the voltage differences between AL0 and VCMI to AL3 and VCMI to net N_{MAC} according to Eq. (5). In Eq. (5), C_u represents the unit capacitance used to compose the four capacitors. The final MAC result voltage (VMAC) on net NMAC is equal to Eq. (6).

$$\Delta V_{MAC} = \frac{C_u\Delta V_{AL0} + (2C_u)\Delta V_{AL1} + (4C_u)\Delta V_{AL2} + (8C_u)\Delta V_{AL3}}{15 \times C_u} \quad (5)$$

$$V_{MAC} = VCMI + \Delta V_{MAC} \quad (6)$$

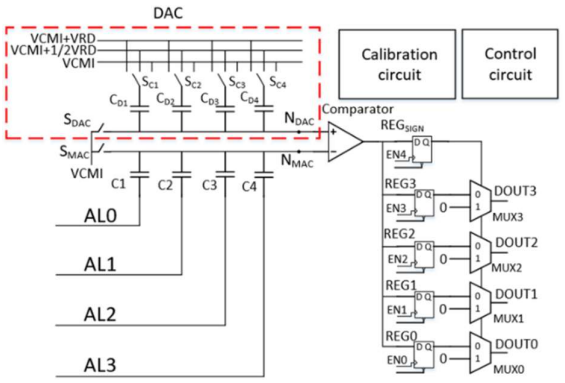

Fig. 9. Block diagram of SAR-based ADC.

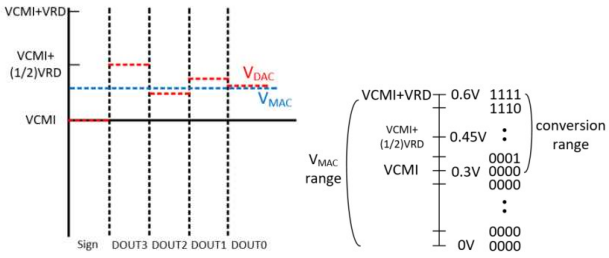

Fig. 10. Sequential comparisons between VMAC and VDAC and the conversion table of analog MAC voltage and digital output.

V. Successive-Approximation-Register-Based ADC

A successive-approximation-register (SAR)-based analog-to-digital converter (ADC) with RELU function is designed to convert analog MAC results to digital output of our RRAM CIM macro. We make reference to the design concepts of [12] when designing the SAR-based ADC. Different from the circuit structure of [12], additional circuitry is integrated to implement RELU function which is frequently used in machine learning algorithm. Fig. 9 shows the block diagram of the SAR-based ADC, comprising four key components: a digital-to-analog converter (DAC), a comparator with calibration circuit, a block containing registers and multiplexers, and finally, the control circuit.

Before CIM operations, the calibration circuit is turned on to calibrate the input offset voltage of the comparator. During the first comparison, if V_{MAC} is lower than V_{DAC}, indicating the MAC result is negative, the output of comparator, which will be stored in register REG_{SIGN}, will be high, and controls MUX_3 to MUX_0 to output zero, realizing the above-mentioned RELU function. On the other hand, if V_{MAC} is higher than V_{DAC} during the first comparison indicating the MAC result is positive, the output of comparator will control the multiplexers MUX3 to MUX0 to output the data stored in REG3 to REG0.

Following the first comparison are four sequential comparisons between V_{MAC} and V_{DAC}. Fig. 10 shows how V_{DAC} approximates V_{MAC} step by step, and generates 1-bit digital output data from MSB to LSB in each step. The conversion table of analog MAC voltage to digital output number is also shown in Fig. 10. In conclusion, the outputs of DOUT3 to DOUT0 are MSB to LSB of the 4-bit output results to 128 sets of 4-bit weight data and 4-bit input data MAC operation.

TABLE I. Comparisons of Our Macro with Works Using RRAM

	This Work	ISSCC'22[13]	JSSC'22[14]	ISSCC'20[15]			JSSC'20[16]	
Technology	40nm	22nm	40nm	22nm	22nm	22nm	55nm	55nm
Memory capacity	256Kb	8Mb	64Kb	2Mb	2Mb	2Mb	1Mb	1Mb
Sensing mode	voltage	voltage	voltage	current	current	current	current	current
Weight precision	4b	1-8b	1-8b	2b	4b	4b	3b	3b
Input precision	4b	1-8b	1-8b	1b	2b	4b	1b	2b
Output precision	4b	3-19b	20b	6b	10b	11b	4b	4b
Accumulation number	128	8	9	16	16	16	9	9
Latency (ns)	17	14.4(8b-8b-19b)	20	9.8	13.1	18.3	11.75	14.6
[1]Energy efficiency (TOPS/W)	71	21.6(8b-8b-19b)	56.67/14.1/7 for 1b/4b/8b	121.38	45.52	28.93	53.17	21.9
[2]FoM	4544	26266	1133/4512/8960 for 1b/4b/8b	1457	3642	5092	638	526
Model	CSPNet-based	ResNet-20	MobileNet	ResNet-20	ResNet-20	ResNet-20	ResNet-18	ResNet-18
Dataset	VWW	CIFAR-100	ImageNet	N/A	CIFAR-100	CIFAR-100	CIFAR-10	CIFAR-10
Accuracy	86.33%	67.11%	53.5%	N/A	64.15%	66.46%	81.83%	88.52%

[1]Energy efficiency = number of operations per second / average power
[2]FoM = (energy efficiency) x (input precision) x (weight precision) x (output precision)

VI. Chip Implementation and Results

The proposed CIM macro is fabricated by 40 nm process with embedded RRAM module provided by TSMC with application for the common scenario of identifying the presence of a person in an image. This functionality serves as a trigger for activating embedded AI devices. Light-CSPDenseNet is used as the AI model along with Visual Wake Words dataset. The model with 4-bit activation and 4-bit weight achieves a baseline accuracy of 88.17%, significantly surpassing the model with binary activation and ternary weight, which only achieves an accuracy of 78.12%. This contrast highlights the necessity of designing higher-precision macros to attain higher accuracy.

The non-ideal effects of the proposed circuits are added to simulate the impact on the inference accuracy, such as the error rate for reading out weights using SA due to RRAM cell resistance variation, the probability distribution of input offset voltage in ADC comparator, and the parasitic capacitance between the metal lines and the capacitors. After adding the input offset voltage of ADC comparator and the error rate of SA output in the model, there is a reduction of 2.43% in the inference accuracy. By retraining the AI model, the accuracy increases by 1.48%, leaving less than 1% reduction in the inference accuracy. After adding all the non-ideal effects in the AI model and retraining it, the accuracy exhibited only a slight decrease of 1.84%, resulting in a total accuracy of 86.33% from the initial baseline of 88.17%.

Table I shows the comparisons of our CIM macro with other works using RRAM. The chip is currently undergoing measurement, so the results presented in the table are based on simulation results. Our macro can achieve highly computing parallelism of 128 4-bit by 4-bit MAC operation, which is much higher than other works with accumulation number of 16. Moreover, our macro can achieve low latency of 17ns with high energy efficiency of 71 TOPS/W, which is better than that of other works with similar precisions. From Table I, we can also observe the trend that the precisions of the macros become higher and higher in recent years.

VII. Conclusion

A voltage mode 256Kb RRAM CIM macro in 40 nm process is designed, and the layout and chip photo of the macro is shown in Fig.11 and Fig.12 respectively. This macro can support the AI algorithms which need higher computation precision and higher throughput of data processing. 128 sets of 4-bit input data are parallel sent into analog multiplication

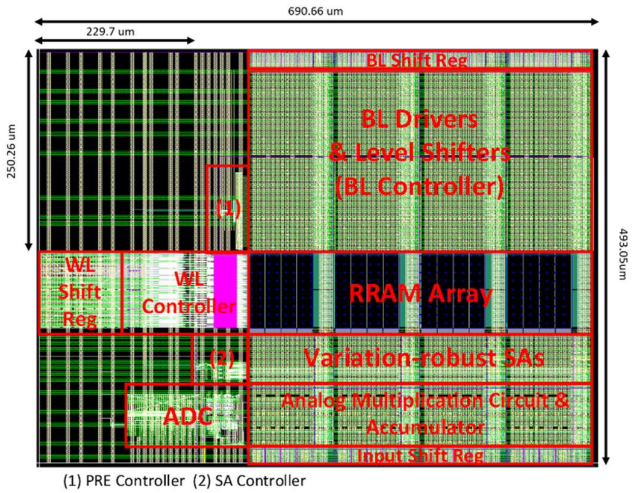

Fig. 11. Layout of the CIM macro.

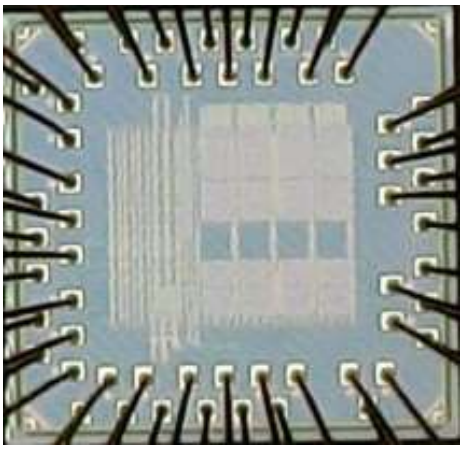

Fig. 12. Chip photo of the CIM macro.

circuits and multiplied with 128 sets of 4-bit weight data stored in RRAM array. Finally, the macro generates 4-bit output data of MAC operation and the latency of the overall computation is 17ns. The energy efficiency of the macro is 71 TOPS/W.

In this work, numbers of solutions are introduced to handle the design challenges of multi-bit RRAM CIM macro including fusing hardware and software optimization. The macro is used to accelerate the convolution computation within the Light-CSPDenseNet AI model, with the application of the Visual Wake Words dataset. After considering the non-ideal effects of the macro, we were able to achieve an accuracy of 86.33%, which is compatible with the baseline accuracy.

ACKNOWLEDGMENT

The authors would like to acknowledge chip fabrication support provided by TSMC Taiwan, NSTC 112-2622-8-A49-013-SB for financial support and Design and measurement support by Taiwan Semiconductor Research Institute, Taiwan.

REFERENCES

[1] A. Biswas and A.P. Chandrakasan, “Conv-RAM: An Energy-Efficient SRAM with Embedded Convolution Computation for Low-Power CNN-Based Machine Learning Applications,” in IEEE ISSCC Dig. Tech. Papers, Feb. 2018, pp. 488–490.

[2] X. Si, Y.N. Tu, W.H. Huang, J.W. Su, M.F. Chang, et al. “A 28nm 64Kb 6T SRAM Computing-in-Memory Macro with 8b MAC Operation for AI Edge Chips,” in IEEE ISSCC Dig. Tech. Papers, Feb. 2020, pp. 246-248.

[3] M. He, C. Song, I. Kim, C. Jeong, S. Kim, et al. “Newton: A DRAM-maker’s Accelerator-in-Memory (AiM) Architecture for Machine Learning”, in Int’l Symp. On Microarchitecture (MICRO), Oct. 2020, pp. 372-385.

[4] H. Choi, Y. Lee, J.J. Kim, S. Yoo, “A Novel In-DRAM Accelerator Architecture for Binary Neural Network”, in Proc. IEEE Symp. LowPower High-Speed Chips (COOL CHIPS), Apr. 2020, pp. 1–3.

[5] Y. Pan, P. Ouyang, Y. Zhao, W. Kang, S. Yin, et al. “A Multilevel Cell STT-MRAM-Based Computing In-Memory Accelerator for Binary Convolutional Neural Network”, IEEE Transactions on Magnetics, vol. 54, no. 11, pp. 1-5, Nov. 2018.

[6] D. Bankman, J. Messner, A. Gural and B. Murmann, “RRAM-Based In-Memory Computing for Embedded Deep Neural Networks,” in 2019 53rd Asilomar Conference on Signals, Systems, and Computers, 2019, pp. 1511-1515.

[7] Y. Park, S. Y. Lee, H. Shin, J. Heo, T. J. Ham and J. W. Lee, "Unlocking Wordline-level Parallelism for Fast Inference on RRAM-based DNN Accelerator," 2020 IEEE/ACM International Conference On Computer Aided Design (ICCAD), San Diego, CA, USA, 2020, pp. 1-9.

[8] B. Penkovsky, M. Bocquet, T. Hirtzlin, J. Klein, E. Nowak, et al. “In-Memory Resistive RAM Implementation of Binarized Neural Networks for Medical Applications,” in 2020 Design, Automation & Test in Europe Conference & Exhibition (DATE), 2020, pp. 690-695.

[9] Y.S. Chiang, C.E. Ni, Y. Sung, T.H. Hou, T.S. Chang, S.J. Jou, “Hardware-Robust In-RRAM-Computing for Object Detection,” IEEE Journal on Emerging and Selected Topics in Circuits and Systems, May 2022.

[10] C. Liu, S. T. Li, T. L. Pan, C. E. Ni, Y. Sung, C. L. Hu, K. Y. Chang, T. H. Hou, T. S. Chang and S. J. Jou, “An 1-bit by 1-bit High Parallelism In-RRAM Macro with Co-Training Mechanism for DCNN Applications,” IEEE International Symp. on VLSI Design, Automation and Test, April 2022.

[11] E. H. Lee and S. S. Wong, "Analysis and Design of a Passive Switched-Capacitor Matrix Multiplier for Approximate Computing," in IEEE Journal of Solid-State Circuits, vol. 52, no. 1, pp. 261-271, Jan. 2017, doi: 10.1109/JSSC.2016.

[12] M. van Elzakker, E. van Tuijl, P. Geraedts, D. Schinkel, E.A.M. Klumperink, et al. “A 10-bit Charge-Redistribution ADC Consuming 1.9 uW at 1 MS/s,” IEEE Journal of Solid-State Circuits, vol. 45, no. 5, pp. 1007-1015, May 2010.

[13] J.M. Hung, Y.H. Huang, S.P. Huang, F.C. Chang, M.F. Chang, et al. “An 8-Mb DC-Current-Free Binary-to-8b Precision ReRAM Nonvolatile Computing-in-Memory Macro using Time-Space-Readout with 1286.4 - 21.6TOPS/W for Edge-AI Devices,” in IEEE International Solid-State Circuits Conference - (ISSCC), Feb. 2022, pp. 182-184.

[14] J.H. Yoon, M. Chang, W.S. Khwa, Y.D. Chih, M.F. Chang, A. Raychowdhury, “A 40-nm, 64-Kb, 56.67 TOPS/W Voltage-Sensing Computing-In-Memory/Digital RRAM Macro Supporting Iterative Write With Verification and Online Read-Disturb Detection,” IEEE Journal of Solid-State Circuits, vol. 57, no. 1, pp. 68-79, Jan. 2022.

[15] C.X. Xue, T.Y. Huang, J.S. Liu, T.W. Chang, M.F. Chang, et al. “A 22nm 2Mb ReRAM Compute-in-Memory Macro with 121-28TOPS/W for Multibit MAC Computing for Tiny AI Edge Devices,” in IEEE International Solid-State Circuits Conference - (ISSCC), Feb. 2020, pp. 244-246.

[16] C.X. Xue, W.H. Chen, J.S. Liu, J.F. Li, M.F. Chang, et al. “Embedded 1-Mb ReRAM-Based Computing-in- Memory Macro With Multibit Input and Weight for CNN-Based AI Edge Processors,” IEEE Journal of Solid-State Circuits, vol. 55, no. 1, pp. 203-215, Jan. 2020.