A Non-volatile Computing-in-Memory ReRAM Macro using Two-bit Current-Mode Sensing Amplifier

Qiqiao Wu[a], Wenhao Sun[a], Junpeng Wang[a], Xuefei Bai[a], Feng Zhang[b], Song Chen[a*] and Yi Kang[a]
[a] School of Microelectronics, University of Science and Technology of China, Hefei, China
[b] Institute of Microelectronics of the Chinese Academy of Sciences, Beijing, China
*Email: songch@ustc.edu.cn

***Abstract*—The non-volatile memories such as FeRAM, PcRAM, and ReRAM afford an innovative approach to the computing in memory (CIM) architecture, which is promising to solve the memory wall problem in the traditional *Von Neumann* architecture. This paper proposes ReRAM-based CIM architecture, which does multiplication and accumulation in the ReRAM array with low power consumption and saves the bandwidth of the storage unit and the processing unit. We combine the CIM architecture with digital circuits to verify the speaker recognition function based on the Long Short-Term Memory (LSTM) network. Moreover, We propose a Two-bit Current-Mode Sensing Amplifier (2b-CSA) as an interface between analog and digital to improve throughput and energy efficiency. This work is simulated under the CMOS 180nm process for compatibility with embedded ReRAM and CMOS logic. The result shows that this work can achieve a CIM operation energy consumption of 1.6pJ per bit.**

***Index Terms*—ReRAM, Analog Compute, Computing-in-Memory, Current-mode Sensing Amplifier.**

I. Introduction

Artificial neural networks (ANNs) have achieved substantial advances in machine-learning problems such as image or speaker recognition. The most commonly used computation for ANNs forward inference is the multiply-and-accumulate (MAC) operation [1]. However, for the data-intensive ANNs inference, it is challenging to improve the energy efficiency of accelerators with traditional *Von Neumann* architecture due to the memory wall problem. Therefore, embedding MAC operations into the memory array itself is more potential [2]. As the next generation of memory, ReRAM has the feature of current accumulation to perform the computation in memory. Thus the academic has aroused great interest in ReRAM-based computing-in-memory (CIM) architecture [3].

To increase the integration density of ReRAM, researchers commonly arrange ReRAM into a crossbar structure. The one-transistor-one-resistor (1T1R) crossbar integration of resistive units can be expanded on a large scale. Based on this structure, the MAC operation can be done by sensing summed currents in the non-volatile memory array and converting them to digital data. And the studies on CIM architecture have made numerous achievements, such as CNN inference [4], binary DNN inference [5], and neural network training [6].

The main challenge in ReRAM-based hardware design is the analog-to-digital conversion part. CIM inference engine in edge computing requires a compact analog-to-digital converter(ADC) design to achieve area and power efficiency [7]. Zhang et al. proposed an embedded HfOx-ReRAM macro with an adaptive current-mode sensitive amplifier (CSA) instead of ADC [8]. However, they employ ReRAM as storage without computing in memory. Long et al. proposed a recurrent neural network (RNN) accelerator based on ReRAM Processing In-Memory (PIM) architecture [9]. Their design applied ADC/DAC as the input and output interface module of ReRAM. Compared with ADC, CSA will consume fewer area and power, thus improving the in-memory computing performance on-chip.

In this paper, we propose a non-volatile ReRAM-based computing in-memory architecture. The entire architecture includes 16 ReRAM-based CIM units with a total storage capacity of 128Kb and 512 2b-CSAs to collect the summed current of 512 bit-lines in 16 CIM units. We map a trained LSTM network [10] to this architecture and complete the forward inference in a fully hardware implementation to verify speaker recognition function. In addition, because the internal ReRAM array generates the partial sum during the neural network inference, we also add shift-and-add modules, activation function modules, and an overall control module to complete the forward inference. The main contributions of this work can be summarized as follows:

- We analyze the impact on the reading accuracy causing by the dispersion of the ReRAM device and propose 2b-CSA for sensing two ReRAM cells simultaneously without theoretical accuracy loss.
- We propose the ReRAM-based CIM architecture for bit-vector dot-product. Every CIM equips with a row decoder with extenders for inputs decoding, a normal column decoder, and 32 2b-CSAs for current sensing. ReRAM array generates bit-vector matrix multiplication sum, which is shifted and added subsequently.
- We map a trained LSTM network to the entire hardware with 16 CIMs, and the hardware evaluation accuracy for the TIMIT data set is 93.1%, then we complete the layout design.

The rest of this paper is organized as follows: In Section II, we present the basics of ReRAM, CIM, and LSTM

978-1-6654-2375-5/21/$31.00 ©2021 IEEE

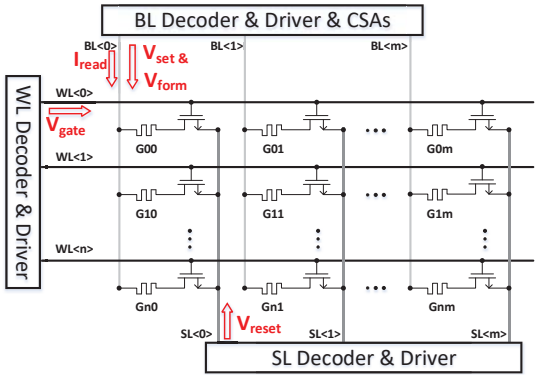
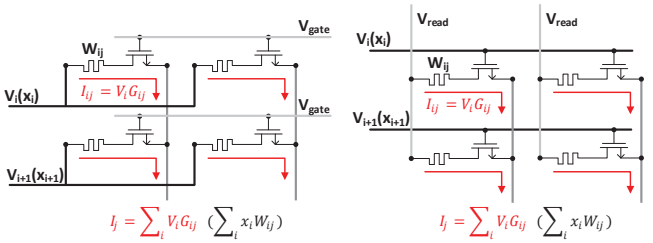


Figure 1: 1T1R architecture with decoders, drivers and CSAs



(a) conventional array topology (b) low-power array topology

Figure 2: Comparison of ReRAM array topology for CIM

acceleration. Section III illustrates the details of our overall architecture, CIM operation, and 2b-CSA circuits. Section IV describes the chip layout and gives the simulation results in comparison with related work. Section V concludes this paper.

II. PRELIMINARY

A. ReRAM device and array architecture

ReRAM is a kind of NVM, and it will not change the stored data even if the power is off. However, RRAM needs to go through a high-voltage FORM process before the ReRAM device can adjust the resistance typically. Therefore, the main difficulty in the circuit design based on ReRAM is that the FORM voltage of ReRAM is usually higher than the core voltage of the CMOS technology node. For example, The operation voltage of the core device at 28nm node is around 1V. As the FORM voltage of the selected cell higher than 2.5 V, the unselected access transistors in the same active bit-line (BL) will generate a high leakage current, causing large voltage drop on the decoder transistor, and the voltage of I/O transisitor can be as high as 6 V if the V_{FORM} reaches 2.5 V in the case of 1 k cells in bit-line [11].

Figure 1 shows the ReRAM array architecture with 1T1R cells. The decoder and driver circuits are usually needed to transmit the analog voltage value to the corresponding position in the array. In order to reduce the leakage currents of BL and SL flowing through the decoder, the typical design does not set a large number of array columns. So that the voltage of V_{FORM}, V_{SET}, and V_{RESET} signal through the IO transistor will not exceed the limit voltage range of the CMOS technology node.

B. Computing in Non-volatile Memory

The data stored in the NVM crossbar array is an analog value. According to Ohm's law (I = VG, where I is the current, V is the voltage, and G is the conductance), the NVM cells can multiply the input voltage and the conductance value of the NVM device, which is an inherent multiplication inside the NVM array. In addition, Kirchhoff's current law sums these contributions along each column line to accumulate.

Figure 2(a) shows the conventional ReRAM topology for CIM. The usual method is to map the activation value to the voltage on BL and map the weight value to 1/R of the ReRAM cell, and the partial sum is mapped to the summary current on SL. However, the conventional ReRAM topology faces two problems, (1) The interface between the ReRAM array and the digital processor requires digital-to-analog and analog-to-digital converters, increasing the chip area and power consumption; (2) All access transistors are simultaneously turned on by VDD on word-lines (WLs) to perform a CIM operation, resulting in large sneak currents and energy wastes [12]. As Figure 2(b) shows, the activation value input is mapped to the logic 0/1 applying on WL. Therefore, the DAC can be omitted from the interface of the processor to the NVM array. Furthermore, during working, transistors whose gate voltage is applied by logic 0 will turn off, reducing leakage energy consumption [12].

C. ANN and LSTM acceleration in NVM

The CIM-NVM architecture, due to its efficient analog computation, is very suitable for doing matrix multiplication (MM). As we all know, convolution and MM are the primary operations in ANNs that have obtained great success in deep learning. Moreover, the convolution can be converted to MM by input images unfolding and convolution kernel reshaping. Therefore, the CIM-NVM architecture is capable of the multi-layer perceptron (MLP) and LSTM network based on MMs and suitable for the convolutional neural network (CNN) based on convolution.

There were a lot of in-memory acceleration studies on CNN and MLP [13], [14]. However, compared with CNN and MLP, LSTM needs more memory acceleration because LSTM occupies more memory storage. And the performance of LSTM accelerators is rigorously restricted by memory bandwidth, so conventional ANN acceleration architectures exhibit poor performance for LSTM inference [15]. Therefore, this work is mainly oriented to accelerate the inference of LSTM networks to satisfy the low power requirements for edge devices.

III. ReRAM based Computing-in-Memory Circuit Architecture

In this paper, we propose a nonvolatile CIM ReRAM macro for efficiently LSTM inference. We use a low-power ReRAM array topology and design a corresponding decoder for the CIM process. In addition, we propose a 2b-CSA with a

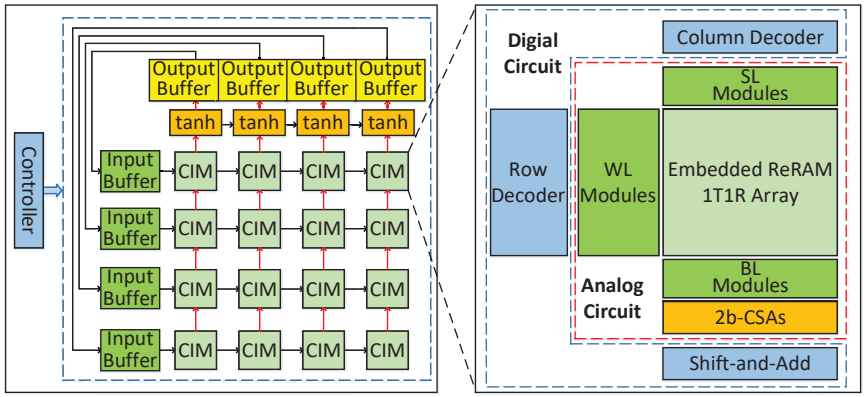


Figure 3: (a) Overall architechture with 16 CIMs. (b) CIM architechture with embedded ReRAM 1T1R array

symmetrical structure to sense the current on the bit-line of the ReRAM array.

A. Overall ReRAM-based architecture

As shown in Figure 3(a), the top-level architecture includes controller, buffers, tanh modules, and CIMs. The controller provides different working modes, such as FORM ReRAM array, weight loading, and CIM mode, sending different commands to the blue dashed box in Figure 3(a). Buffers and the internal storage of CIMs constitute a memory hierarchy. CIMs are both data storage and arithmetic unit. In this CIMs array, the data flow would be as follows: (1) first in the weight loading mode, the weights are fed from the input buffer to the left column of the CIMs array, and then move from left to right, and finally, all weights will be installed in different CIMs respectively; (2) next in the CIM mode, the input values of each LSTM recurrent cycle are fed from the input buffer to the left column of the CIMs array, similarly, moving from left to right so that each CIM will get some partial sum results; (3) the partial sum results will be moved up from the bottom row of the array, performed tanh activation which is frequently applied in LSTM, and conveyed to the output buffer after accumulation; (4) a LSTM recurrent cycle is consist of step(2) and step(3), the controller sends commands to repeat LSTM recurrent cycle until finishing LSTM inference. To deploy the LSTM network in our architecture, we quantify weight and activation and use Taylor fitting for the tanh function. In addition, the hardware evaluation accuracy for the TIMIT data set is 93.1%, whose accuracy degradation is 5.6% lower than ideal floating-point inference.

Figure 3(b) depicts the specific structure of a CIM. In the weight loading mode, the embedded ReRAM array will store external weight data by bit, and in the CIM mode, the input data is sent to the corresponding rows through WL modules. Besides, we design CIMs hierarchically. As shown in Figure 3(b), inside the red dashed box are analog circuits containing embedded ReRAM arrays and three-terminal(WL, SL, BL) read-write modules. Each column (bit-line) of the array is equipped with a 2b-CSA, generating 2-bit output data every period. In addition, 2b-CSA replaces the analog-to-digital converter as an interface between analog and digital. Moreover, Our decoders and shift-and-add digital circuits are in the blue dashed box. We have computed the bit-wise partial sum in the analog part, and it needs to be shifted and added to generate the final output gate value.

B. ReRAM array with Decoders and CIM operation

The top-level architecture contains 16 CIMs with a total storage capacity of 128Kb. Considering that LSTM is a weight-intensive neural network, we set the capacity of a single CIM to 8Kb to store the weights and bias inside CIM. On the other hand, because the FORM voltage of the ReRAM device is applied on bit-lines, which is higher than V_{gate} applied on the word-line. Therefore, in a single CIM, the number of bit-line is 32, and the number of word-line is 256. Accordingly, the I/O transistor's driving voltage will not exceed the CMOS node voltage limitation when the FORM operation works.

Like Static Random-Access Memory (SRAM), our ReRAM array also needs row and column decoders to reduce bus bandwidth and metal layer routing pressure. A 5-32 decoder is used together on the bit-line and the source-line, which has the additional function of turning on all output channels simultaneously when CIM mode works. However, the innovation in this work lies in the word-line decoder design. Figure 4 illustrates that our word-line decoder is composed of a basic 7-128 decoder and 128 extenders. The extenders controlled by the signal in the below part in Figure 4, expand the decoder's 128 outputs to 256 lines connecting to the ReRAM analog array. Particularly, when the *Mode* signal is 0, the *Sel* signal is equivalent to the lowest address signal. Thus the 7-bit address of the 7-128 decoder combined with the *Sel* signal constitutes an 8-bit address signal. The decoder selects a specific row in the ReRAM array according to the provided address. The mode talked above will be used to perform FORM, RESET, and SET operations on ReRAM. We load all weights by traversing the entire array. On the contrary, when the *Mode* signal is 1, our decoder works in the CIM mode. At this

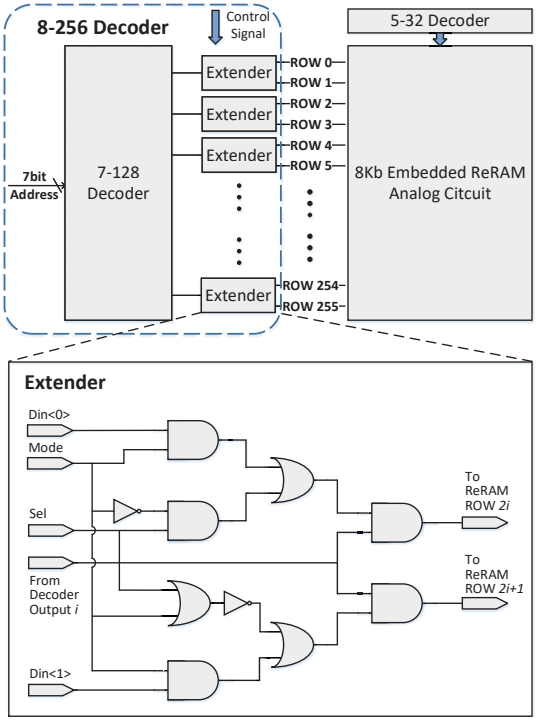


Figure 4: Decoder with extenders

TABLE I: Truth table of extender

Din<1>	Din<0>	ROW 2i	ROW 2i+1
0	0	off	off
0	1	off	on
1	0	on	off
1	1	on	on

time, assuming that the decoder output i enables extender i, the on or off of ROW $2i$ and ROW $2i$+1 will be determined by Din<1:0> signal according to Table I. Note that our 8-256 decoder with extenders in this work has more operating modes flexibly adapted to different CIM operations than the basic 8-256 decoder. In general, a ReRAM array can complete a (1×2) bit-vector matrix multiply a (2×32) bit-vector matrix, producing 32 2-bit results. Next, we will describe the CIM operation in the two-row working mode.

In Figure 5, we assume that the weight and input data are quantized in 4-bit, then 2×4 ReRAM cells are an operation unit that stores two 4-bit weights, and the input data is divided into four cycles, multiplying the weights by each bit, and then accumulate the product value. Take (3,6) dot product (10,2) as an example and get the result(3*10 + 6*2 = 42), as shown in Figure 5(a). However, in CIM architecture, we get the final dot-product result through(2^0*10 + 2^1*(10+2) + 2^2*2 + 2^3*0) in Figure 5(c). Because the value stored in the ReRAM array does not assign the significance of the bit position, so the array's output should be shifted and added to get final results, as shwon in Figure 5(b). In particular, we exploit 2b-CSAs in the CIM architecture, so the output of a 2b-CSA is represented

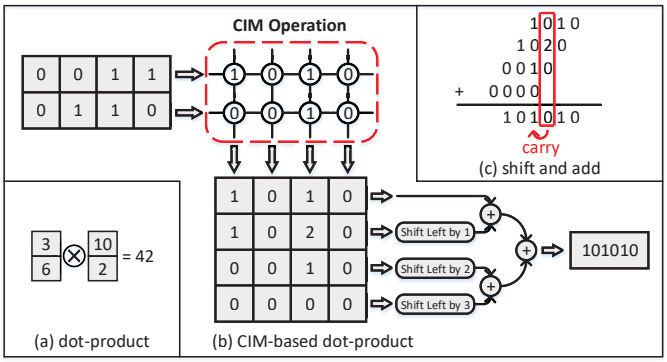


Figure 5: A computation example based on CIM operation

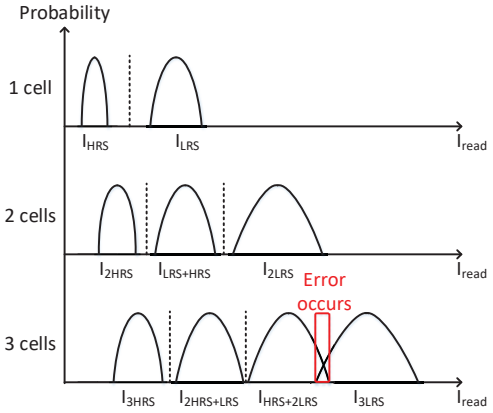


Figure 6: Read current analysis with ReRAM device dispersion

by two-bit data. Therefore, the computation process exists the intermediate number 2, and it will be carried to a higher bit in the shift-add operation.

C. Two-bit Current-mode Sensing Amplifier

In this subsection, we explain the reasons for using 2b-CSAs in CIM architecture and the circuit analysis of 2b-CSA. Traditional CSAs are used to amplify the bit-line swing in order to decrease access time in modern SRAM and some non-volatile memories (NVMs) [16]. In order to improve the reliability of ReRAM macro and reduce read errors, ReRAM usually uses a current adaptive mode sensitive amplifier(CSA), which also achieves a faster read speed and robustness to noise [8]. However, the traditional CSA is equivalent to a binary quantization of the read current, while in the CIM architecture, a higher precision quantization is required for partial sum accumulation. In this paper, we propose a 2b-CSA in place of traditional CSA. When the word-line decoder turns on two rows simultaneously, a 2b-CSA will receive a combined current of two ReRAM cells. Moreover, two-bit data can express three current situations, $I_{HRS} + I_{HRS}$, $I_{HRS} + I_{LRS}$, $I_{LRS} + I_{LRS}$, where I_{HRS} is the current of high resistance state and I_{LRS} is the current of low resistance state.

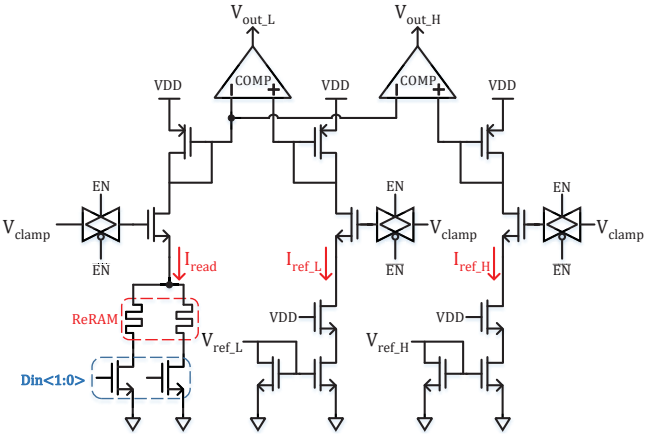


Figure 7: Sensing circuit of 2b-CSA

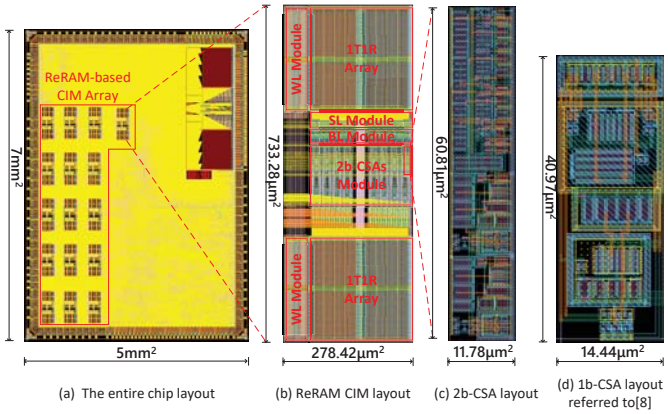


Figure 8: Layout design and area comparison between 2b-CSA with traditional CSA

However, read errors will occur when more selected ReRAM cells generate summed current for the CSA, which is caused by the inevitable dispersion of ReRAM devices. We take the embedded ReRAM test performance in [17] for analysis. The read current fluctuation of low resistance state device at 72K ohms is 22.9%, and the read current fluctuation of high resistance state device at 530K ohm will reach 43.7% [17]. As shown in Figure 6, based on the current fluctuations of the above test results, we have performed read current analysis on selecting 1, 2, and 3 ReRAM cells, respectively. When more than two cells are turned on, confusion occurs between two states of read current, which leads to errors in the read results. In the situation where three ReRAM cells are turned on in Figure 6, $I_{HRS+2LRS}$ and I_{3LRS} may have overlapping areas, causing the disturbance of sensing currents.

Figure 7 shows the sensing circuits of 2b-CSA, which is modified from the conventional CSA in [8]. Compared with the conventional CSA, 2b-CSA has two current paths, I_{ref_L} and I_{ref_H} respectively, and an additional comparator to generate one more bit output. In Figure 7, by adjusting V_{ref_L}, I_{ref_L} is between I_{2HRS} and $I_{LRS+HRS}$, and I_{ref_H} is between $I_{LRS+HRS}$ and I_{2LRS} by adjusting V_{ref_H} so that the three current states can be separated without read errors. Besides, we coded the output of 2b-CSA for linking with the digital part. As shown in Table II, Vout(1,0) is redundant, and Vout(1,1) represents the encoded data 2. We use fewer MOS transistors to optimize 2b-CSA to reduce the area and improve energy efficiency, as discussed in the next section.

TABLE II: 2b-CSA output coding scheme

V_{out_H}	V_{out_L}	Encoded data
0	0	0
0	1	1
1	0	invalid
1	1	2

IV. Layout and Simulation Results

This section will discuss the layout scheme of this work and analyze the 2b-CSA layout area and power consumption. Finally, we simulate and analyze the CIM operation.

A. Layout scheme

We use the 180nm CMOS PDK for the entire design and systematize the CIM module as an IP to adapt to the IC back-end design flow. We first complete the analog layout in the CIM module, so the analog IP area can be calculated, then mark the port coordinates of the analog IP, which is necessary while routing. Next, we place analog IPs in the entire layout, and finally finish the IC back-end design flow. Figure 8(a) shows that the entire chip layout contains 16 ReRAM-based CIM IPs.

B. 2b-CSA area and power consumption analysis

Figure 8(b) shows an 8Kb capacity ReRAM CIM analog layout, which occupies an area of $0.2mm^2$. In the 2b-CSAs module, there are 32 2b-CSAs respectively connected to the 32 bit-lines of the ReRAM array. In order to optimize the layout, we divided the 256×32 elongated array into eight 32×32 sub-arrays without breaking the logic topology.

Figure 8(c) shows the layout of 2b-CSA, and Figure 8(d) shows the layout of one-bit CSA referred to [8]. Comparing the two areas, 2b-CSA's area is 21% larger due to an additional comparator and some auxiliary transistors not in traditional CSA. However, assume the following situation: if the 1b-CSA is used for the CIM process, two different reference voltages need to be set for the CSA in the two periods obtaining two bits of output, respectively. Instead, the 2b-CSA in this work should be set two different reference voltages in the same period, thereby generating two-bit data in one period. The 2b-CSA only consumes a small extra amount of hardware cost, which significantly reduces the CIM latency. In other words, 2b-CSA doubles the throughput rate compared with 1b-CSA.

Figure 9 shows the simulation results of analog CIM IP. During the first CIM cycle, we select two low-resistance-state ReRAM devices through the decoder, which means that

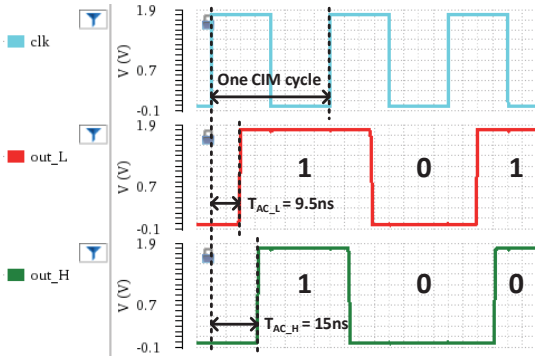


Figure 9: ReRAM-based CIM function simulation

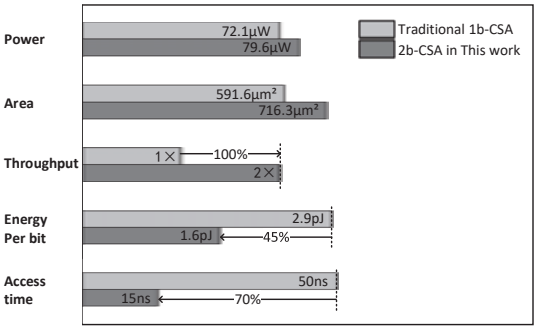


Figure 10: Power&Area&Throughput&Energy per bit analysis

(1,1) dot product (1,1), and simulation result is (1,1), which is encoded as the number 2 for further process. During the second CIM cycle, we modify the address of the decoder so that two high-resistance-state ReRAM devices are selected, which means that (1,1) dot product (0,0), the result of the simulation is (0,0). In addition, the simulation waveform indicates that the two outputs access at different times because of the distinct reference voltages of the two comparators in the 2b-CSA. The low bit output access time is 9.5ns, and the high bit output access time is 15ns. By contrast, the access time of traditional 1b-CSA in [8] is 50ns. Therefore our work's access time is reduced by 70%. Moreover, 2b-CSA can double data throughput due to higher precision sensing during the MAC period and reduce the operation energy per bit by 45% with a minor increase in power consumption and area, as shown in Figure 10.

V. Conclusion

We propose a ReRAM-based CIM architecture, including the decoder with extenders and 2b-CSA. We consider the dispersion of ReRAM devices and realize the bit-vector matrix multiplication in the two-row mode. Compared with 1b-CSA, 2b-CSA in this work achieves doubled throughput, dramatically reduces operating energy consumption per bit and access time with a minor increase in power consumption and area. In the future, we plan to apply the fault-tolerant solution for CIM operation of more selected ReRAM cells with higher energy efficiency.

Acknowledgment

This work was supported in part by the National Key R&D Program of China under grant No. 2019YFB2204800, in part by National Natural Science Foundation of China (NSFC) under grant Nos. 61874102, 61732020, 61931008, and U19A2074, in part by the Strategic Priority Research Program of Chinese Academy of Sciences, Grant No. XDB44000000. The authors would like to thank Information Science Laboratory Center of USTC for the hardware & software services.

References

[1] Y. H. Chen, T. Krishna, J. S. Emer, and V. Sze, "Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks," in *Solid-state Circuits Conference*, 2016.

[2] S. Yu, "Neuro-inspired computing with emerging nonvolatile memory," *Proceedings of the IEEE*, vol. 106, no. 2, pp. 260–285, 2018.

[3] D. B. Strukov, G. S. Snider, D. R. Stewart, and R. S. Williams, "The missing memristor found," *nature*, vol. 453, no. 7191, pp. 80–83, 2008.

[4] P. Yao, H. Wu, B. Gao, J. Tang, Q. Zhang, W. Zhang, J. J. Yang, and H. Qian, "Fully hardware-implemented memristor convolutional neural network," *Nature*, vol. 577, no. 7792, pp. 641–646, 2020.

[5] W. H. Chen, K. X. Li, W. Y. Lin, K. H. Hsu, and M. F. Chang, "A 65nm 1mb nonvolatile computing-in-memory reram macro with sub-16ns multiply-and-accumulate for binary dnn ai edge processors," in *2018 IEEE International Solid - State Circuits Conference - (ISSCC)*, 2018.

[6] S. Ambrogio, P. Narayanan, H. Tsai, R. M. Shelby, and G. W. Burr, "Equivalent-accuracy accelerated neural-network training using analogue memory," *Nature*, vol. 558, no. 7708, 2018.

[7] S. Yu, X. Sun, X. Peng, and S. Huang, "Compute-in-memory with emerging nonvolatile-memories: Challenges and prospects," in *2020 IEEE Custom Integrated Circuits Conference (CICC)*, 2020.

[8] Z. Feng, D. Fan, D. Yuan, L. Jin, and M. F. Chang, "A 130nm 1mb hfox embedded rram macro using self-adaptive peripheral circuit system techniques for 1.6x work temperature range," in *2017 IEEE Asian Solid-State Circuits Conference (A-SSCC)*, 2017.

[9] Y. Long, T. Na, and S. Mukhopadhyay, "Reram-based processing-in-memory architecture for recurrent neural network acceleration," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 26, no. 12, pp. 2781–2794, 2018.

[10] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," *computer science*, 2014.

[11] X. Xu, L. Tai, T. Gong, J. Yin, and M. Liu, "40× retention improvement by eliminating resistance relaxation with high temperature forming in 28 nm rram chip," in *2018 IEEE International Electron Devices Meeting (IEDM)*, 2018.

[12] F. Su, W. H. Chen, L. Xia, C. P. Lo, and Y. Liu, "A 462gops/j rram-based nonvolatile intelligent processor for energy harvesting ioe system featuring nonvolatile logics and processing-in-memory," in *2017 Symposium on VLSI Circuits*, 2017.

[13] R. Mochida, K. Kouno, Y. Hayata, M. Nakayama, T. Ono, H. Suwa, R. Yasuhara, K. Katayama, T. Mikawa, and Y. Gohou, "A 4m synapses integrated analog reram based 66.5 tops/w neural-network processor with cell current controlled writing and flexible network architecture," in *2018 IEEE Symposium on VLSI Technology*. IEEE, 2018, pp. 175–176.

[14] L. Ni, Z. Liu, Y. Hao, and R. V. Joshi, "An energy-efficient digital reram-crossbar-based cnn with bitwise parallelism," *IEEE Journal on Exploratory Solid-State Computational Devices and Circuits*, vol. 3, pp. 37–46, 2017.

[15] J. Han, H. Liu, M. Wang, Z. Li, and Y. Zhang, "Era-lstm: An efficient reram-based architecture for long short-term memory," *IEEE Transactions on Parallel and Distributed Systems*, vol. 31, no. 6, pp. 1328–1342, 2020.

[16] S. Ardalan, D. Chen, M. Sachdev, and A. Kennings, "Current mode sense amplifier," in *48th Midwest Symposium on Circuits and Systems, 2005.*, 2005, pp. 17–20 Vol. 1.

[17] D. Dong, L. Jing, Y. Wang, X. Xu, and M. Liu, "The impact of rtn signal on array level resistance fluctuation of resistive random access memory," *IEEE Electron Device Letters*, vol. PP, no. 99, pp. 1–1, 2018.