SRAM-based Computation In Memory Architecture to Realize Single Command of Add-Multiply Operation and Multifunction*

Chua-Chin Wang[1]
Department of Electrical Engineering
National Sun Yat-Sen University
Kaohsiung, Taiwan 80424
Email: ccwang@ee.nsysu.edu.tw

Chia-Yi Huang
Department of Electrical Engineering
National Sun Yat-Sen University
Kaohsiung, Taiwan 80424
Email: mark584967@vlsi.ee.nsysu.edu.tw

Chia-Hung Yeh[2]
Department of Electrical Engineering
National Taiwan Normal University
Taipei, Taiwan
Email: yeh@mail.ee.nsysu.edu.tw

2021 IEEE International Symposium on Circuits and Systems (ISCAS) | 978-1-7281-9201-7/20/$31.00 ©2021 IEEE | DOI: 10.1109/ISCAS51556.2021.9401561

***Abstract*—**
This paper presents a computation in memory (CIM) architecture and circuit design featured with single command to execute addition, signed multiplication, and multi-function to resolve poor computation throughput caused by von Neumann bottleneck. The proposed CIM takes advantage of 2T-Switch circuit which needs only 2 switches to select the required computation units such that the area on silicon is reduced. RCAM (ripple carry adder and multiply) unit realized with full swing gate diffusion input (FS-GDI) in a single-ended disturb-free 7T SRAM further reduces the power consumption and active circuit area. Auto-switching write-back circuit consisting of BL auto-switching circuit, Data switching circuit, and WL auto-switching circuit facilitates the automatic restore of addition and multiplication to designated memory addresses. The proposed CIM is realized using 40-nm CMOS process to demonstrated 12.18/28.19 fJ/bit normalized write/read energy at 100 MHz system clock rate.

***Index Terms*—computation in memory (CIM), auto-switching write-back, single-ended SRAM, FS-GDI, AI**

I. Introduction

The pursue of speed in computing system development has never been changed. However, almost all computing architecture used for computation intensive applications such as artificial intelligence (AI), biological systems, neural networks, are based on von Neumann machines, which separates the storage units (memory) with arithmetic logic units (computation). Thus, despite the advanced CMOS technology, it still has a well-known issue called von Neumann bottleneck [1]. Due to large amount of dataflows between memory and CPU, which cause overhead limitations, many researches have been developed, including in-memory computing (IMC), also known as computation in memory (CIM), [2] [3] [4] [5]. The aim of CIM is to bypass von Neumann bottleneck and realize computation in memory arrays directly.

Considering the speed and reliability requirements for AI usage (including CNN, DNN, etc.), SRAM has the edges over DRAM. However, SRAM has very poor area efficiency on the other hand. With reference to [6], a 4T load-less SRAM was proposed to reduce the area and the power consumption simultaneously, which shows the potential to be the bedstone of CIM for AI applications. However, the disturbance of the bitline when read/write data has been pointed out to lower static noise margin (SNM) [7]. Therefore, a write-assist loop with multi-Vth transistors are presented to ensure the disturb-free feature [8]. Nevertheless, when read/write in a long period, the leakage current will destroy the stored data, which needs to be resolved.

This investigation explores the feasibility of CIM realization using load-less single-ended SRAM. More importantly, the proposed CIM is featured with single command to execute addition, multiplication, and multi-function to increase throughput by routing around von Neumann bottleneck. The core of CIM, i.e., RCAM (ripple carry adder and multiply) unit, is realized with full swing gate diffusion input (FS-GDI) to further reduces the power consumption and active circuit area [9] [10].

II. Low-energy CIM Design

The proposed CIM architecture based on single-ended 7T SRAM is shown in Fig. 1, including 1-kb 7T SRAM array, SRAM Control Circuit, Row Decoder, Column Decoder/Selector, FS-GDI RCA, CIM Control Circuit, Auto-Switching Write Back Circuit, 2T-switch Current Compensation Circuit, BIST, and Output Buffer. All the well known circuits in prior works are skipped. The contribution of this investigation is given in the following sections.

A. CIM circuit design

A 4×4 array prototype to demonstrate CIM is given in Fig. 2. The operation to carry out an addition is as follows.

[1]Prof. C.-C. Wang is also with Inst. of Undersea Tech., National Sun Yat-Sen Univ, Kaohsiung, Taiwan.
[2]Prof. C.-H. Yeh is also with Dept. of Elec. Eng., National Sun Yat-Sen Univ., Taiwan.

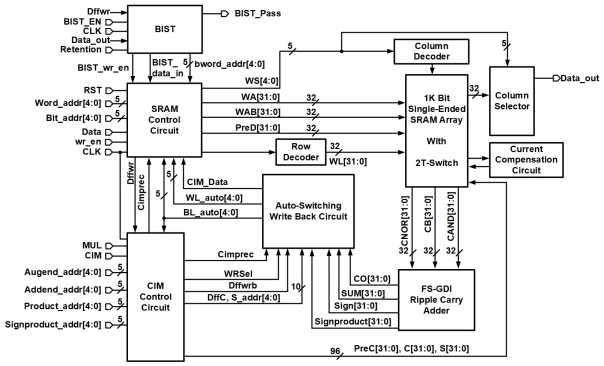


Fig. 1. System view of the proposed CIM

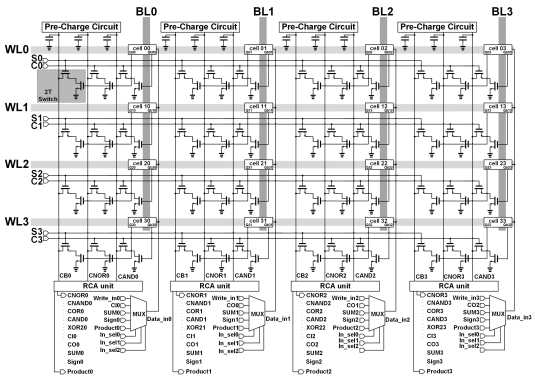


Fig. 2. Schematic of the proposed CIM

1). 2T-switch (e.g., upper left corner) is the monitor to sense which row is used to be the operands. Any two rows can be used as the operands for the addition.
2). CBi, CNORi, and CANDi, for i=0. . .3, are generated, respectively, which are coupled as inputs to the corresponding RCA unit at the bottom of each column.
3). RCA unit carries out the addition.
4). MUX is in charge of writing back the sum to a designated memory address provided the corresponding selection signals are given.

B. 7T SRAM cell

Referring to Fig. 3, the 7T SRAM cell and the associated control signal generator are given. The SRAM cell is composed of conventional 6T cell and one access NMOS such that only one bit line, namely BLB, is needed for R/W operations. Certainly, another bit line (BL) can be added if needed. The R/W operations of the 7T cell are as follows.

• Read : WA and WL_x are asserted to couple QB_{xx} with BLB. Q_{xx} is then generated by the inverter and coupled to BL if needed.
• Write 0 : PreD is pulled high to ground BLB_x. WL_x and WAB are then asserted, while WA is shut off. Q_{xx} is then grounded.

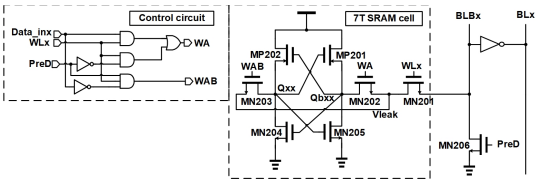


Fig. 3. Schematic of the single-ended 7T SRAM cell

• Write 1 : PreD is pulled high to ground BLB_x at the beginning. WA is then asserted to pull down QB_{xx}. At the same time, VDD will charge Q to high through MP202.

The above operations are tabulated in Table I.

TABLE I
7T SRAM CELL SIGNAL CONTROL VS. OPERATION

	$\mathrm{Data\_in}_x$	PreD	WL_x	WA	WAB
Standby	X	X	0	0	0
Read	X	0	1	1	0
Write 0	0	1	1	0	1
Write 1	1	1	1	1	0

C. 2T switch and computation operation

One of the features in the proposed CIM is disclosed in Fig. 4, where 2T-switch circuit and its auxiliary circuitry are shown. The computation operation is summarized as follows.
• initialization : PreC is reset low before the operation. CB_x, $CNOR_x$, and $CAND_x$ are all precharged high at the same time.
• start up : PreC is pulled high. S_x and C_x ($x = 0, 1, 2, 3$) drive 2T-Switch units to couple Q_{xx} or Qb_{xx} to CB_x, $CNOR_x$, $CAND_x$, respectively.
• computation phase I : only one of S_x signals is asserted to select the memory address to store the carry. If Q_{xx} is high, CB_x will be low. (Refer to left hand side of Fig. 4)
• computation phase II : two of C_x signals are asserted to select two operands for computation. NOR or AND operations are determined by the combination of Q_{xx} and Qb_{xx} (Refer to next page of Fig. 4).

D. RCAM

RCAM (ripple carry add and multiply) unit is the core the CIM computation. With reference to Fig. 5, the RCAM of the proposed CIM is a degenerated FS-GDI logic (right side) compared to the original version (left side). Namely, all the transistors with the source directly coupled to VDD or GND are removed such that not only the transistor count is reduced, the power dissipation is also dropped.

E. Control circuit

All the operations of the proposed CIM is governed by the core of the circuit, namely Control Circuit, consisting of CIM Timing Control Circuit, Auto-switching Pre-charge Control

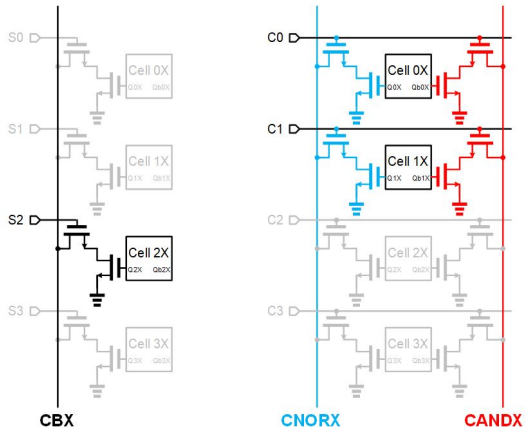


Fig. 4. R/W operations of 2T-switch

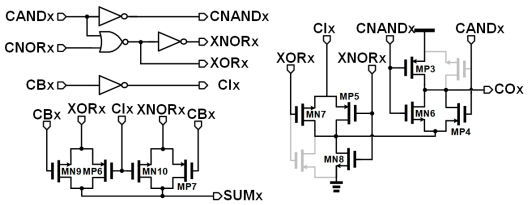


Fig. 5. Schematic of RCAM

circuit, Address selecting Control Circuit, and CIM Control Circuit unit(Refer to Fig. 6). When either CIM or MUL is asserted, which means OP is pulled high, corresponding actions will be triggered. Firstly, if OP is asserted, CIM Timing Control Circuit is activated to generated corresponding control signals to carry out required computations. If CIM is pulled high, selectors are activated to select the addresses of addend and summand for the addition operation. Similarly, if MUL is pulled high, the multiplication is executed.

For any ADD or MUL operations, the most critical action is the write back. That is, either the sum or the product shall be written back to a designated address. Referring to Fig. 7,

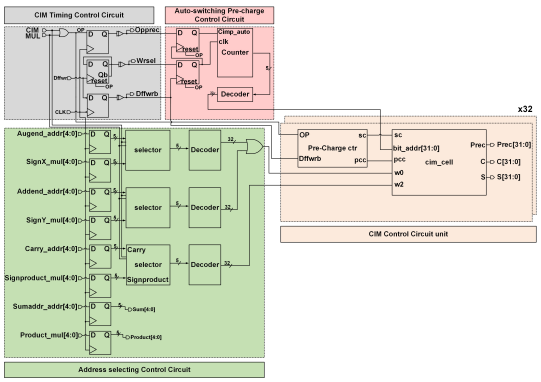


Fig. 6. Control circuit of the proposed CIM

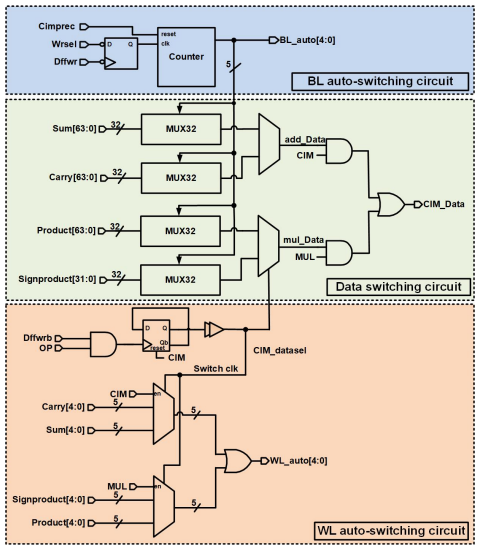


Fig. 7. Schematic of automatic write back

a total of 2 blocks are shown. The top block (in blue) is BL (bit line) auto-switching circuit. The middle block (in green), if asserted when OP is high, is in charge of data selection. If it is an addition operation, carry and sum will be selected to be CIM_Data. By contrast, the product will be selected and output. The write back operation starts from LSB (BL0) to MSB.

III. Simulation and Analysis

The proposed 1-kb CIM is realized using TSMC 40-nm CMOS process, as shown in Fig. 8. The chip area is 833.91×867.31 μm^2, where the core area is 510.265×432.81 μm^2.

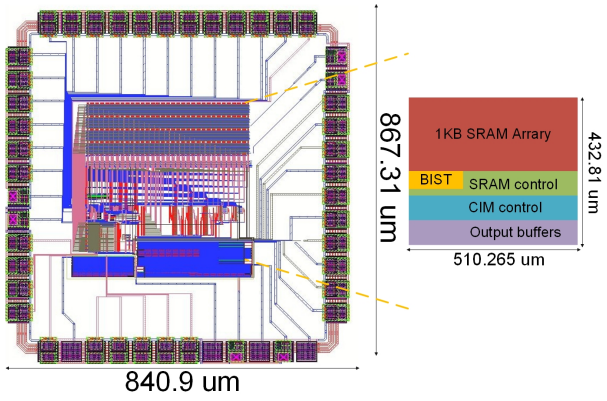


Fig. 8. Layout of the proposed CIM

A. Post-layout simulations

Referring to Fig. 9, the correct post-layout simulations of X[(-1)0(-1)0] × Y[(-1)00(+1)] = product[(+1)000] is demonstrated at all corners (0.9V±10%, [FF, FS, TT, SF, SS], [0, 25,

75]^{o}C) given 100 MHz clock. This formula can be separated into several equation. For example, (-1)×(-1) = (+1) is the operation in the bit 3 and (0)×(0) = (0) is the operation in the bit 2. To ensure the reliability of the proposed CIM, random testing simulations over 10,000 vectors, namely Monte Carlo simulation, is also carried out as shown in Fig. 10, and the results are all correct.

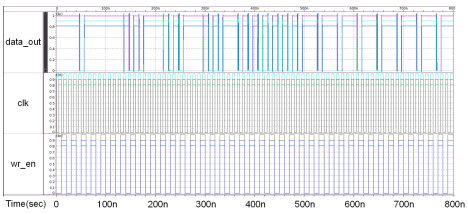
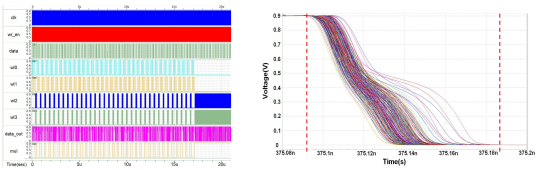


Fig. 9. Add-and-multiply operations



(a) (b)

Fig. 10. Monte Carlo simulation: (a) Execute 1000 times multiplication; (b) Execute 1000 times writing in data low

Table II summarizes the KPI (key performance index) comparison of recent two CIM designs and ours. Though our capacity is smaller, the proposed CIM is the only one to demonstrate the prototypical design fully executing signed addition and multiplication in memory arrays. The worst normalized write energy per operation is also lower than those of the prior two designs. It is believed, hence, that our CIM is a superior solution to date.

IV. Conclusion

A low-power CIM with signed addition and multiplication functions is demonstrated in this investigation. It is mainly featured with single-ended 7T cells associated with 2T-switch and auto-write back control. Based on all-PVT-corner simulation results, the proposed CIM not only shows more computation function variety, but also demonstrates low energy features, which will benefit future IoT applications with AI demand. In the future, we will have multiplication and addition operating together and finishing the matrix operations to complete the complex caculations in AI demand.

Acknowledgment

This investigation was partially supported by Ministry of Science and Technology, Taiwan, under grant MOST 108-2218-E-110-002 and 109-2218-E-110-007. The authors would like to express our deepest appreciation to TSRI (Taiwan Semiconductor Research Institute) in NARL (National Applied Research Laboratories), Taiwan, for the assistance of EDA tool support.

TABLE II
Performance comparison of CIM designs

	[11] TVLSI	[12] TCAS-I	this work
Year	2017	2018	2020
Process	65 nm	45 nm PTM	40 nm CMOS
Verification	Meas.	Simul.	Simul.
Operations	A-SRAM	NAND NOR XOR RCS	NAND NOR XOR 4-bit ADD 4-bit MUL
Size	1 kb	N/A	1 kb
Norm. Write Energy (fJ/bit)	39.5 (@ 1.2V)	N/A	7.4 (@ 0.9V, worst case)
Norm. Read Energy (fJ/bit)	4.1 (@ 1.2V)	N/A	17.2 (@ 0.9V, worst case)
Norm. Avg. Energy (fJ/bit)	21.8	14.6	12.3 (@ 0.9V, worst case)

$$\text{Norm. Write Energy} = \frac{\text{Write Energy}}{\text{Process}^2} 10^3$$

$$\text{Norm. Read Energy} = \frac{\text{Read Energy}}{\text{Process}^2} 10^3$$

References

[1] J. Backus, Can programming be liberated from the von Neumann style?: A functional style and its algebra of programs, *Commun. ACM,* vol. 21, no. 8, pp. 613-641, Aug. 1978.

[2] Y. Wang, H. Yu, L. Ni, G. Huang, M. Yan, C. Weng, W. Yang, and J. Zhao, An energy-efficient nonvolatile in-memory computing architecture for extreme learning machine by domain-wall nanowire devices, *IEEE Trans. on Nanotechnology,* vol 14, no. 6, pp. 998-1012, Nov. 2015.

[3] S. Jain, A. Ranjan, K. Roy, and A. Raghunathan, Computing in memory with spin-transfer torque magnetic RAM, *IEEE Trans. on Very Large Scale Integration Systems (TVLSI),* vol. 26, no. 3, pp. 470-483, Mar. 2018.

[4] S. Jeloka, N. B. Akesh, D. Sylvester, and D. Blaauw, A 28 nm configurable memory (TCAM/BCAM/SRAM) using push-rule 6T bit cell enabling logic-in-memory, *IEEE Journal of Solid-State Circuits (JSSC),* vol. 51, no. 4, pp. 1009-1021, Apr. 2016.

[5] D. Fan, and S. Angizi, Energy efficient in-memory binary deep neural network accelerator with dual-mode SOT-MRAM, in *Proc. IEEE Inter. Conf. on Computer Design (ICCD),* pp. 609-612, Nov. 2017.

[6] C.-C. Wang, Y.-L. Tseng, H.-Y. Leo, and R. Hu, A 4-Kb 500-MHz 4-T CMOS SRAM using low- VTHN bitline drivers and high- VTHPlatches, *IEEE Trans. on Very Large Scale Integration Systems (TVLSI),* vol. 12, no. 9, pp. 901-909, Sep. 2004.

[7] C.-C. Wang, C.-L. Lee, and W.-J. Lin, A 4-Kb low power SRAM design with negative word-line scheme, *IEEE Trans. Circuits Syst. I, Reg. Papers (TCAS-I),* vol. 54, no. 5, pp. 1069-1076, May 2007.

[8] C.-C. Wang, and C.-L. Hsieh, Disturb-free 5T loadless SRAM cell design with multi-vth transistors using 28 nm CMOS process, in *IEEE Inter. SoC Design Conf. (ISOCC),* pp. 103-104, Oct. 2016.

[9] A. Morgenshtein and A. Fish and I. A. Wagner, Gate-diffusion input (GDI): a power-efficient method for digital combinatorial circuits, *IEEE Trans. on Very Large Scale Integration Systems (TVLSI),* vol. 10, no. 5, pp. 566-581, Oct. 2002.

[10] M. A. Ahmed, and M. A. Abdelghany, Low power 4-bit arithmetic logic unit using full-swing GDI technique, in *Proc. Inter. Conf. on Innovative Trends in Computer Engineering (ITCE),* pp. 193-196, Feb. 2018.

[11] J. Lee, D. Shin, Y. Kim, and H. Yoo, A 17.5-fJ/bit energy-efficient analog SRAM for mixed-signal processing, *IEEE Trans. on Very Large Scale Integration Systems (TVLSI),* vol. 25, no. 10, pp. 2714-2723, Oct. 2017.

[12] A. Agrawal, A. Jaiswal, C. Lee, and K. Roy, X-SRAM: Enabling in-memory boolean computations in CMOS static random access memories, *IEEE Trans. Circuits Syst. I, Reg. Papers (TCAS-I),* pp. 1-14, Jul 2018.