Compute-In-Memory SRAM Design for Accelerating Low-Precision Neural Network

Chuyao Cheng, and Xuan Su
{clare.cheng, xsu52}@berkeley.edu

***Abstract*—Frequent data movement in and out the memory system has been a bottleneck for many machine learning applications. There has been a lot of active research focusing on bringing computing as close as possible to the memory array such as, Compute-In-Memory(CIM), which focuses on reducing data movement by integrating compute elements within the memory primitives. One of the active areas of CIM is re-designing SRAM Macro incorporating computation and readout circuitry. In this paper, two state-of-the-art SRAM Macro designs, one using twin 8T SRAM cell [1] and one using 8T FinFET SRAM cell [2], will be discussed and compared, including their unique techniques and performance. The comparison study will be based off 4-bit operation as they are both designed for multi-bit calculation. The design with FinFET 8T SRAM cell is likely to outperform the other one with respect to area density and latency, while the other may have higher accuracy and stability.**

I. Introduction

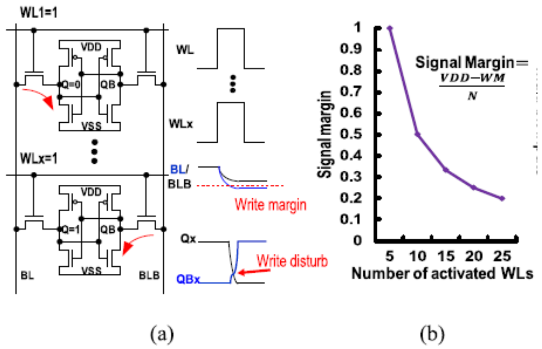
WITH data volume growing exponentially and the popularities of today's edge devices, power and performance limits are being reached due to frequent data movement in and out the memory system. For data-heavy workloads, such as convolution neural networks, data movement often dominates when implemented with today's computing architectures [3]. This has motivated spatial architectures, where the arrangement of data-storage and compute hardware is distributed and explicitly aligned to the computation dataflow. There has been a lot of active research focusing on bringing computing as close as possible to the memory array such as, Compute-In-Memory(CIM) and Compute-Near-Memory (CNM). CIM techniques focus on reducing data movement by integrating compute elements within or near the memory primitives in order to relax memory throughput, increase the performance of the applications and reduce energy. Several silicon verified SRAM based CIM devices have been developed, such as a 10T Conv-SRAM for binary weight neural networks [4], a charge-domain in-memory-computing accelerator for binarized convolutional neural network (CNN) [5] and alike. Those work already demonstrated the benefits as well as the efficiency boost they could bring.

With the data sets becoming more and more complex, multibit CNN has been developed to improve the accuracy of the inference. However, multibit SRAM CIM designs still have many challenges and trade-offs.

1) Write Disturb Issues: Rule-based 6T SRAM cells may have write disturb issues when the bitline voltage V_{BL} is lower than the worst case write margin voltage V_{WN} as shown in Figure 1(a). Keeping the V_{BL} higher than the V_{WN} reduces the voltage range for various MAC values (MACVs) and also degrades the signal margin (V_{SM}) across neighbouring MACVs [6].

2) Limited Signal Margin: In order to improve energy efficiency, multiple inputs can be loaded into the CIM macro simultaneously through multiple activated WLs to increase the number of MAC operations. However, with the increase of the activated WLs, the signal margin decreases under a given maximum V_{BL} swing range (V_{BLS_MAX}) as shown in Figure 1(b).

In this paper, we will present two state-of-the-art papers of CIM SRAM designs, discuss what design techniques they employ to overcome some challenges and compare some parts of their design and performance.



(a) (b)

Fig. 1. Challenges in multibit SRAM-CIM design (a) write disturb issues (b) limited signal margin

II. Architecture and Techniques

In this article, we will mainly focus on analyzing two state-of-the-art SRAM CIM Macro designs [1][2]. Those two designs, built upon basic 8T SRAM cells, both focused on multi-bit computation, while there were also different circuits techniques and methods. In the following sub-sections, the Macro structure, SRAM cell design, input mapping schemes and circuits techniques employed in output operation will be discussed.

A. Macro Structure

Figure 2 and 3 present the overall structure of the 55nm T8T SRAM CIM [1] and 7nm 8T SRAM-CIM [2].

1) 55nm T8T SRAM CIM: As presented in Figure 2, the macro architecture of the 55nm T8T SRAM CIM mainly contains a T8T SRAM cell array, an even–odd dual channel (EODC) array structure, two's complement processing units (C2PUs), output combiners (OCs) and a configurable global–local reference voltage generation (CGLRVG).

There are two operation modes supported by this T8T SRAM CIM unit-macro, including memory mode and CIM mode. In the memory mode, the signed weights can be written to the T8T cell array in two's complement form with a read/write peripheral circuit. While in CIM mode, in order to map multibit inputs to the T8T cell array, multiple read-WLs (RWLs) are activated simultaneously in either single channel or dual channel. Partial MACVs (pMACV) are processed separately on read-bitlines (RBLs) and then combined using a C2PU and OC. And thus, multibit MACVs can be accessed in parallel across IOs.

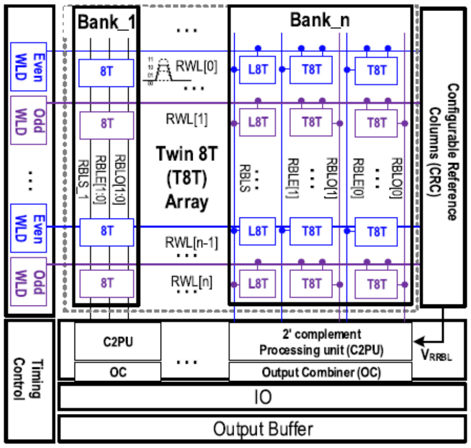


Fig. 2. Macro Architecture of the proposed 55nm T8T SRAM CIM

2) 7nm 8T SRAM CIM: The macro architecture of the 7nm 8T SRAM CIM are shown in Figure 3. Different from the 55nm T8T SRAM CIM, this structure has standard two-port compiler 8T SRAM to balance the cell stability and area overhead. In addition, row-wise RWL counters and column-wise Flash ADCs are added to perform multiply-and-average (MAV) computations. A 4b digital counter is applied for each row to convert the 4b input data into multiple RWL pulses. 4b weight is loaded through four SRAM cells on each row and each four RBLs share one 4b Flash ADC as shown in Figure 3.

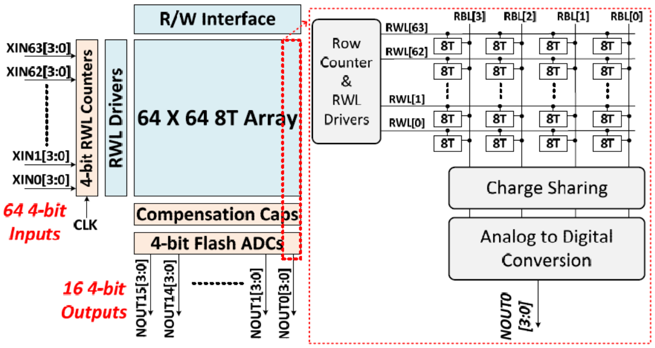


Fig. 3. Macro Architecture of the proposed 7nm 8T SRAM CIM

B. SRAM Cell

Figure 4 and Figure 5 shown below demonstrate the unit SRAM cell design of the entire Macro. Two designs both employ the 8T SRAM cell but the 55nm one customizes the 8T cell into a twin-8T cell (T8T) with two read ports.

For the T8T SRAM cell shown in Figure 4, each T8T cell comprises two read-decoupled 8T (RD8T) SRAM cells representing two bits, one for most significant bit and one for least significant bit. The transistor width of the read-port (N2 and N1) in M8T is double that in L8T in order to provide 2-bit weighted cell current [1]. The advantages of this cell design are larger voltage swing without interrupting writability compared to 6T, and the small area overhead by modifying foundry cell. There are two modes of bit precision available for input, a single bit with two voltage levels and 2-bit with four levels. The detailed method of performing computation will be discussed in the circuits technique section later.

For the 7nm 8T SRAM shown in Figure 5, the SRAM architecture is designed around a standard two-port macro using a foundry 8T SRAM in a 7nm FinFET technology. 8T cell is chosen because it has higher stability than that of a 6T cell and less overhead and modifications compared to T8T and 10T cells. With the advance 7nm FinFET technology, the cell only occupies an area of 0.053 μm^2.

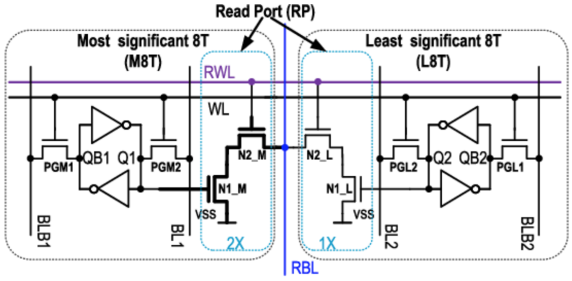


Fig. 4. SRAM Cell of the proposed 55nm T8T SRAM CIM

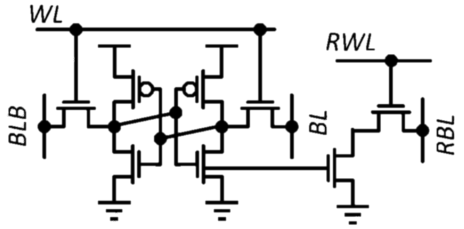


Fig. 5. SRAM Cell of the proposed 7nm 8T SRAM CIM

C. Multibit Input and Weight Mapping

1) Multibit Input Mapping: There are two common methods to load multibit inputs into WL signals. First is using a fully parallel input structure which allows multibit inputs converted into multi-level WL voltages (V_{WL}) with a digital-to-analog converter (DAC) or an input decoder with multiple power supply sources. Another method is using a fully serial input structure which has multiple inputs sent sequentially to the WL.

The T8T SRAM CIM uses a fully parallel input structure, which supports two configurable input bit precision modes, including binary input mode and 2-bit-input mode. The binary input mode has two RWL voltage levels (0 V and V_{RWLL3}), while the 2-bit-input mode has four RWL voltage level (0 V, V_{RWLL1}, V_{RWLL2}, V_{RWLL3}) as shown in Table 1. An even–odd dual-channel (EODC) scheme is also developed to reduce by half the number of cells on an RBL in a T8T cell array and thus lower the settling time of V_{RBL}.

Different from the T8T SRAM CIM, the 8T SRAM CIM has a fully serial input structure, which uses the number of RWL pulses to represent the 4b input as shown in Figure 6(a). The RWL pulses are controlled by row-wise 4b digital counters, which is more variation-tolerant and compact than row-wise DAC and analog delay line [7-8].

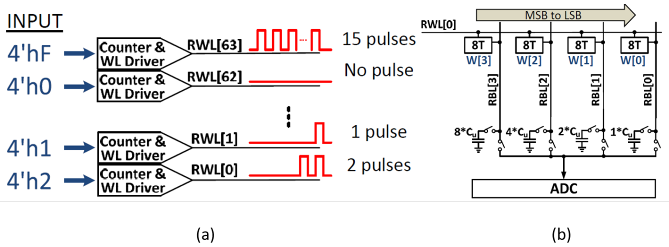
2) Multibit Weight Mapping: Many prior studies use binary-weighted capacitors to realize multibit weight mapping. Both the T8T SRAM CIM and the 8T SRAM CIM apply this technique in their designs, but the structures of them still have some differences. The T8T SRAM CIM uses M8T and L8T structure to provide 2-bit weighted cell current inside each T8T, and use binary-weighted capacitors in the weight processor (WP) to combine multiple pMACVs generated on RBLs with respect to their bit positions. While the 8T SRAM CIM has both binary-weighted computation capacitors and corresponding compensation capacitors, and uses charge sharing techniques to average out the voltage on the computation caps to generate final output.

In the T8T SRAM CIM structure, each T8T cell comprises one M8T and one L8T. The read-port in the M8T has transistor with double width than that in the L8T so that 2-bit weighted cell current can be provided. Moreover, a two's complement weight mapping (C2WP) scheme is proposed so that the j-bit signed weight can be written to the T8T cell array in two's complement form via single WL activation. Only j cells are required to store the j-bit signed weights, and thereby maximizing the memory array usage. Moreover, a two's complement processing unit (C2PU) is proposed to support the T8T cell array structure and the C2WM scheme. The weight processor in the C2PU comprises several binary-weighted capacitors in order to combine the pMACV generated on each RBL with its specific place values.

In the 8T SRAM CIM structure, the multibit weight is realized by charge sharing among computation caps inside the Flash ADC. As shown in Figure 6(b), from LSB to MSB, the RBL connects to the computation caps with 1:2:4:8 capacitance ratios. Compensation caps are added to the RBL to make sure each RBL has same amount of capacitance. Before RWL activation, all the caps on the RBL are pre-charged to VDD. Once RBL sampling starts, the voltage on each RBL will be lowered by the discharge currents if the bit-cell is storing a '1'. After RBL sampling, each binary-weighted computation cap is isolated from the RBLs and holds the voltage same as its corresponding RBL. Then, charge sharing happens among the computation caps to average out the voltage, which represents the MAV result.

Table 1.
Truth Table of T8T SRAM Cell

Input (2b)	RWL Voltage	Weight		RBL Current	IWP
		M8T	L8T		
11	WLL3	1	1	$3\times I_{MC}$	9
10	WLL2	1	1	$2\times I_{MC}$	6
01	WLL1	1	1	$1\times I_{MC}$	3
11	WLL3	1	0	$2\times I_{MC}$	6
10	WLL2	1	0	$4/3\times I_{MC}$	4
01	WLL1	1	0	$2/3\times I_{MC}$	2
11	WLL3	0	1	I_{MC}	3
10	WLL2	0	1	$2/3\times I_{MC}$	2
01	WLL1	0	1	$1/3\times I_{MC}$	1
00	VSS	1/0	1/0	0	0



(a) (b)

Fig. 6. Input (a) and weight (b) mapping of the proposed 7nm 8T SRAM CIM

D. *Circuit Techniques (Operation and Output Readout)*

The FinFET 8T SRAM array uses a 4-bit Flash ADC to convert the sampled voltage to a digital output. The voltage represents the multiply and average result of the charge sharing 64 44b array. In this case, column-wise Flash ADC uses area-efficient SA instead of analog comparator to save area and reduce energy consumption. Inherent SA cap is used as unit cap directly to sample. Unlike analog comparators that require pre-amp or offset caps to minimize kick-back effect, the proposed SA is immune to this effect because of the self-sampling on the SA internal caps [2].

The readout circuitry of T8T SRAM is much more complicated since the T8T SRAM CIM design has multiple modes including single-channel and dual-channel. The output are basically accumulated in two steps, first by multiplying and accumulating partial products of input and weight, then the output combiner comprising a 2-bit-shifter and a 7-bit-adder is used to combine two MAC values from previous calculation. The following equations explain the calculation process and are based off an example of 4-bit input and 5-bit weight multiplication. During first cycle, two MSBs of input are multiplied with weight and convert into digital using C2PU (1); during second cycle, two LSBs are used to calculate MACV (2). And they are combined by the output combiner by adding and shifting in the end (3).

$$Cycle1 : MACV1 = \sum(IN_i[3:2] \times W_i[4:0]) \qquad (1)$$

$$Cycle2 : MACV2 = \sum(IN_i[1:0] \times W_i[4:0]) \qquad (2)$$

$$\begin{aligned} OutputCombiner : FMAC &= 4 \times MACV1 + MACV2 \\ &= \sum(IN_i[3:0] \times W_i[4:0]) \quad (3) \end{aligned}$$

III. Setup for Comparison

A. Row Circuitry

Since different methods of writing inputs into SRAM cell are used in two papers, one is using different RWL levels to indicate different inputs, while the other is using multiple RWL pulses to pass in data bits, those two methods in terms of linearity, delay and accuracy shall be compared. Both designed for multi-bit operation, the two designs will be compared based on 4-bit input operation.

B. Performance

The performance of the T8T SRAM CIM and the 8T SRAM CIM should be compared based on the MAC operations with same number of input bits and weight bits. However, since the T8T SRAM CIM applies a two's complement weight mapping technique, its weight requires one more bit to function as the sign bit. And therefore, the operations would be conducted with 4-bit input for both, but 4-bit weight for 8T SRAM CIM and 5-bit weight for T8T SRAM CIM. The access time with 1V power supply for CIM operations should be analyzed and compared. Furthermore, some large datasets, for example MNIST, can be trained and tested to compare the accuracy of these two systems.

C. Area Overhead

One of the biggest drawbacks of CIM is the relatively large area and energy consumption of A/D conversion, which is the readout circuitry. One paper uses a modified SAR ADC plus a output combiner while the other uses Flash ADC. Area and energy ratio of the read circuitry compared to the entire MACRO will be compared and analyzed.

IV. Hypothesis

The method of using multiple RWL pulses is likely to outperform multi-level voltage RWL in scalability and linearity. In terms of performance, there are obvious trade-offs between the two designs. Twin 8T SRAM is designed for better signal margin and high accuracy, with extra area overhead; FinFET 8T SRAM aims for higher density and less delay. As regards to A/D conversion area and energy overhead, our hypothesis is that they are similar comparing two designs, as A/D conversion is an essential part of this analog-nature SRAM CIM design. The important question is if larger size array will amortize the cost of this conversion.

V. Conclusion

In this article, we present two SRAM CIM unit-macro: T8T SRAM CIM [1] and 8T SRAM CIM [2], and compare the unique features of these two including the macro architectures, the SRAM cells structure, the multibit input/weight mapping techniques, as well as the MAC operations and output readout methods. The T8T SRAM CIM has a twin 8T cell structure and applies an EODC scheme for multibit input mapping and a C2WM scheme for multibit signed weight mapping, while the 8T SRAM CIM is built with a conventional 8T cell structure and uses multiple RWL pulses and charge-sharing techniques to realize multibit input and weight mapping. The performance will be evaluated based on 4-bit MAC operations. It is highly likely that the 8T SRAM CIM will have smaller bitcell area and less cycle time, while the T8T SRAM CIM will demonstrate higher accuracy and stability. Although in-memory computing may have some limitations in scalability and accuracy, it still has the potential for achieving high energy efficiency and throughput, and appears to be a promising technique that deserves further study.

References

[1] X. Si et al., *A Twin-8T SRAM Computation-In-Memory Macro for MultipleBit CNN-Based Machine Learning*. ISSCC, pp. 396-397, Feb. 2019.

[2] Dong, q., Ersin Sinangil, M., Erbagci, B., Sun, D., Khwa, W., Liao, H., Wang, Y. and Chang, J. (2020). *A 351 TOPS/W and 372.4 GOPS Compute-in-Memory SRAM Macro in 7nm FinFET CMOS for Machine Learning Applications*. In: International Solid-State Circuits Conference. IEEE.

[3] Jia, H., Tang, Y., Valavi, H., Zhang, J., and Verma, N. (2018). *A microprocessor implemented in 65nm CMOS with configurable and bit-scalable accelerator for programmable in-memory computing*.arXiv preprint arXiv:1811.04047.

[4] A. Biswas and A. P. Chandrakasan, *Conv-RAM: An energy-efficient SRAM with embedded convolution computation for low-power CNNbased machine learning applications.* In IEEE ISSCC Dig. Tech. Papers, Feb. 2018, pp. 488–489.

[5] H. Valavi, P. J. Ramadge, E. Nestler, and N. Verma, *A mixed-signal binarized convolutional-neural-network accelerator integrating dense weight storage and multiplication for reduced data movement.* In Proc. IEEE Symp. VLSI Circuits, Jun. 2018, pp. 141–142.

[6] W.-S. Khwa et al., *A 65nm 4Kb algorithm-dependent computing-inmemory SRAM unit-macro with 2.3ns and 55.8TOPS/W fully parallel product-sum operation for binary DNN edge processors.* In IEEE ISSCC Dig. Tech. Papers, Feb. 2018, pp. 496–497.

[7] J. Zhang et al., *In-Memory Computation of a Machine-Learning Classifier in a Standard 6T SRAM Array.* In JSSC, vol. 52, no. 4, pp. 915-924, 2017.

[8] S. K. Gonugondla et al., *A 42pJ/Decision 3.12TOPS/W Robust In-Memory Machine Learning Classifier with On-Chip Training.* In ISSCC, pp. 490-491, Feb.2018.