CiMComp: An Energy Efficient Compute-in-Memory based Comparator for Convolutional Neural Networks

Kavitha S[1], *Student Member*, IEEE, Binsu J Kailath[1], *Member*, IEEE, B . S . Reniwal[2], *Member*, IEEE
[1]Indian Institute of Information Technology, Design and Manufacturing, (IIITDM) Kancheepuram
[2]Indian Institute of Technology, Jodhpur

***Abstract*—The utilization of large datasets in applications results in significant energy expenditures attributed to frequent data shifts between memory and processing units. In-Memory-Computing (IMC) distinguishes itself by employing computations within a memory crossbar to perform logic operations, leading to enhanced computational speed and energy efficiency. This study introduces RASA-based subtractor, strategically improved for computation, and energy consumption. Subsequently, the proposed subtractor are employed to construct a comparator and facilitate pooling operations. The comparator is developed using the proposed subtractor, achieves the comparison in n steps for a n-bit comparator. Additionally, a n-bit min pooling operation for a n $\times$ n (4 $\times$ 4) feature map requires 2^n -1 (15) steps. Energy consumption of the RASA design demonstrates hopped-up performance, showcasing an average savings of 87.42% and 89.98% compared to the ASA and Muller C based subtractor.**

***Index Terms*—SRAM, In memory computing (IMC), XOR, Subtractor, Comparator**

I. Introduction

In-memory computing (IMC) is a specialized approach that works within the context of the von Neumann architecture [1]. It optimizes memory and processor integration to reduce data movement and improve computational efficiency, making it well-suited for compute-intensive applications such as neuromorphic computing and machine learning (ML) classifiers employed in image recognition. Static Random-Access Memory (SRAM) offers rapid access times, facilitating fast data retrieval and manipulation directly within the memory. It allows for parallel access to multiple memory cells, enabling the processor to execute multiple computations simultaneously. Various computations, including Boolean logic operations (NAND, NOR, XOR), adders as well as multiply and accumulate (MAC), have been explored and incorporated using SRAM-based IMC [2]- [5].

An skewed asymmetrical sense amplifier(ASA) facilities the logic computations like AND/NAND, OR/NOR (represent A and B) [6]. Additionally, the XOR operation is derived by NOR-ing the outputs from NOR and AND gates within the SA . Employing the Muller-C element [7] to obtain XOR logic, NAND and OR operations are achieved by utilizing the discharge of RBL/RBLB and connecting them with inverters, which are then employed as inputs for external logic gates to compute the Sum and Carry signals as shown in Fig. 1 (a), (b). For obtaining Full Subtractor (FS), the Sum is equivalent to the Difference. For borrow, $\bar{A}$ is loaded into the SRAM, and the carry is determined similarly to that in the Full Adder (FA), resulting in an equivalent carry-out that serves as the borrow-out. By leveraging the Reconfigurable assist sense amplifier (RASA) [4], we construct a full adder, full subtractor, and a magnitude comparator utilizing the subtractor, and an example of comparator in max/min pooling which demonstrates superior performance is comprehensively discussed.

II. Proposed Design

A. Implementation of Full Adder and Subtractor

RASA is employed for the computation of Boolean logic, including NAND, NOR, and XOR, utilizing a MUX control to select the desired operation [4]. To achieve the XNOR/XOR operation, AS_A and AS_B are set to 0 and 1, with I_A representing RBL and I_B representing RBLB. By inputting Out and I_A into an external two-input NAND gate results in an XNOR/XOR operation. The schematic of the full adder implemented with RASA is depicted in Fig. 1 (c). The XOR operation is achieved through IMC (A $\oplus$ B), and an external XOR gate is employed to derive the Sum = (A $\oplus$ B) $\oplus$ Cin, as well as the Carry = (A $\oplus$ B).Cin + A.B. The computation of the full adder necessitates the use of three external logic gates and takes one cycle to compute a one-bit FA. This individual FA module is used to calculate an n-bit ripple carry adder (RCA) by propagating the carry from the previous stage as the carry (Cin) for the current stage through n cycles of operation.

For the full subtractor (FS), the difference is computed similarly to the sum of the FA. To obtain the borrow, an inverted Cin fed to the external gates yields Borrow

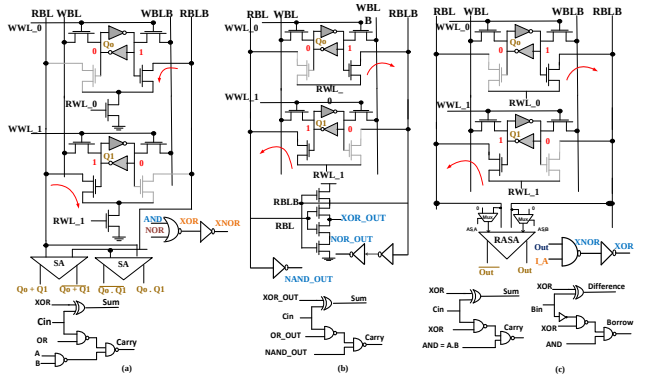

Fig. 1. Adder using Boolean computation with (a) ASA [6] (b) Muller C [7] (c) RASA [4]

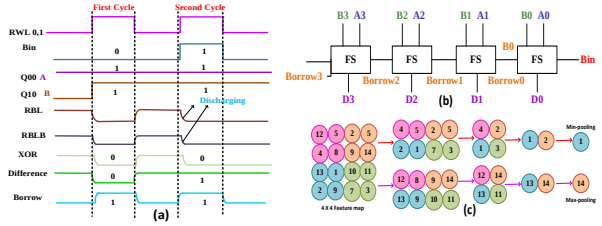

Fig. 2. (a) Control signals for one bit full subtractor (b) 4-bit Ripple borrow subtractor (c) Min pooling and Max pooling of a 4 × 4 Feature Map of CNN.

= (A ⊕ B).$\bar{Cin}$ + A.B. Thus, without the necessity of altering or loading an inverted A ($\bar{A}$) into the SRAM, using an inverter with the external logic gates enables the computation of the FS in a single cycle. The control signal is depicted in Fig. 2 (a). This individual FS module is applied to calculate an n-bit ripple borrow subtractor (RCA) by carrying forward the borrow from the preceding stage as the input borrow (Bin) for the current stage requiring n cycles of operations. 4-bit RBS is shown in Fig. 2 (b).

B. Comparator

A magnitude comparator perform a comparison between two binary numbers, determining whether one binary number is greater than, equal to, or less than the other, represented as $Z_{A>B}$ = A$\bar{B}$, $Z_{A=B}$ = $\bar{A}\bar{B}$ + AB, $Z_{A<B}$ = $\bar{A}$B. With boolean logic derived from IMC RASA, the comparison can be performed in three cycles using AND and XNOR operations. Consequently, the number of cycles for an n-bit comparator is on the order of 3*n when employing RASA. Alternatively, the proposed FS can be utilized to compute the comparator. When the difference output subtractor is zero, it indicates equal inputs, i.e., $Z_{A=B}$. When the borrow is 'One', it indicates the input A is less than B, i.e., $Z_{A<B}$; otherwise, the input A is greater than B, i.e., $Z_{A>B}$.

C. Min and Max pooling

The Min (Max) Pooling layer aggregates features within a specific region by identifying the minimum (maximum) value in that region. Min Pooling is particularly effective for images with a brighter background, accentuating the analysis of darker pixels. In CNN, Max Pooling is employed to extract the most crucial features from the feature map. Both Min and Max Pooling can be implemented using the comparator with the proposed subtractor. An illustration of Min Pooling is provided, where a 4×4 region necessitates 8 parallel comparisons in the first cycle, 4 parallel comparisons in the second cycle, two parallel comparisons in the third cycle, and one series comparison to obtain the final minimum value. For a 4×4 feature map, we require 2^n -1 steps, where 2^n/2 parallel comparison reduces the computation time by half.

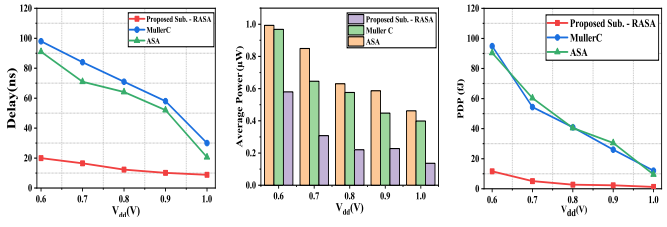

Fig. 3. (a) Delay (b) Average Power consumption and (c) PDP at different supply voltage.

TABLE I
PERFORMANCE COMPARISON TABLE

Architecture	Delay (ns)	Power (μW)	PDP (fJ)	n-bit comparator
RASA	8.82	0.136	1.199	n cycles
ASA	20.63	0.462	9.531	2n cycles
Muller C	30	0.399	11.97	2n cycles

III. SIMULATION RESULTS AND COMPARISON

The benchmarks circuits of RASA, ASA [6] and the Muller C [7] are realized in an industrial 65 nm UMC CMOS bulk technology. The analysis of these designs with different parameters such as compute delay, average power consumption, product of delay and power (PDP) and area to characterize the performance of the design.

Delay is the time it takes for the SEN signal to reach 50% of its full swing potential to the output drop to reach 50% of the full swing potential. This delay is calculated during the generation of difference and borrow outputs across various V_{dd}, employing worst-case analysis at the SS process corner. For example, at V_{dd} of 1 V, obtaining the output involves a delay of 8.82 ns, which is 70.6% and 57.24% less than Muller C and ASA respectively, is depicted in Figure 3 (a).

Power dissipation is measured as the average power consumed during both the read and write cycles. For a supply voltage of 1 V, the design consumes around 0.136 μW of power, which is 65.91% and 70.56% less than Muller C and ASA respectively. The average power is calculated for various supply voltages is showcased in Figure 3 (b). Another important parameter under consideration is the product of delay and power (PDP). The PDP for different supply voltages, is displayed in Figure 3 (c). The number of steps required to compute a n-bit comparator using n, 2n and 2n for RASA, Muller C and ASA subtractor respectively. RASA uses single SA to compute the boolean logic, while the ASA requires two SAs. The performance comparison is shown in Table I.

IV. CONCLUSION

The energy-efficient comparator presented incorporating IMC subtractor within SRAM, not only meets the rigorous requirements of contemporary computing but also lays the groundwork for forthcoming advancements in low-power and high-performance circuit design. The n-bit comparator, developed using the introduced subtractor, achieves efficient comparisons in n steps. The energy consumption analysis reveals that RASA design outperforms ASA-based (Muller C-based) subtractor, shows an average savings of 87.42% (89.98%). These findings underscore the potential of proposed approach in significantly enhancing energy efficiency and computational performance in large-scale computing applications.

REFERENCES

[1] C.J. Jhang and et.al, "Challenges and Trends of SRAM-Based Computing-In-Memory for AI Edge Devices," in IEEE TCAS I, 2021.
[2] M. Izhar and et.al, "Logic Circuit Implementation for Enabling SRAM Based In Memory Computing," 2022 5th IMPACT, Aligarh, India, 2022.
[3] M. Ali and et.al, "IMAC: In-memory multi-bit multiplication and ACcumulation in 6T SRAM array, *IEEE TCAS I*, 2020.
[4] Kavitha S and et.al, "Enabling Energy-Efficient IMC With Robust Assist-Based Reconfigurable SA in SRAM," in IEEE JETCAS, 2023.
[5] Kavitha and et.al An Approach Towards Analog In-Memory Computing for Energy-Efficient Adder in SRAM Array. VDAT 2022. Springer.
[6] Agrawal and et.al, "X-SRAM: Enabling In-Memory Boolean Computations in CMOS Static Random-Access Memories" in IEEE TCAS I.
[7] Song, S.; Kim, Y. Novel In-memory Computing Circuit using Muller C-element, *In Proceedings of the 2021 18th ISOCC*.