


Article

# Novel In-Memory Computing Adder Using 8<sup>+</sup>T SRAM

Soonbum Song and Youngmin Kim \* 

School of Electronic and Electrical Engineering, Hongik University, Seoul 04066, Korea;  
c0192607@mail.hongik.ac.kr

\* Correspondence: youngmin@hongik.ac.kr

**Abstract:** Von Neumann architecture-based computing systems are facing a von Neumann bottleneck owing to data transfer between separated memory and processor units. In-memory computing (IMC), on the other hand, reduces energy consumption and improves computing performance. This study explains an 8<sup>+</sup>T SRAM IMC circuit based on 8<sup>+</sup>T differential SRAM (8<sup>+</sup>T SRAM) and proposes 8<sup>+</sup>T SRAM-based IMC full adder (FA) and 8<sup>+</sup>T SRAM-based IMC approximate adder, which are based on the 8<sup>+</sup>T SRAM IMC circuit. The 8<sup>+</sup>T SRAM IMC circuit performs SRAM read and bitwise operations simultaneously and performs each logic operation parallelly. The proposed IMC FA and the proposed IMC approximate adder can be applied to a multi-bit adder. The two adders are based on the 8<sup>+</sup>T SRAM IMC circuit and thus read and compute simultaneously. In this study, the 8<sup>+</sup>T SRAM IMC circuit was applied to the adder, leveraging its ability to perform read and logic operations simultaneously. According to the performance in this study, the 8<sup>+</sup>T SRAM IMC circuit, proposed FA, proposed RCA, and proposed approximated adder are good candidates for IMC, which aims to reduce energy consumption and improve overall performance.

**Keywords:** von Neumann bottleneck; memory wall; SRAM; in-memory computing (IMC); Process-in-Memory (PIM)



**Citation:** Song, S.; Kim, Y. Novel In-Memory Computing Adder Using 8<sup>+</sup>T SRAM. *Electronics* **2022**, *11*, 929. <https://doi.org/10.3390/electronics11060929>

Academic Editors:  
Mariagrazia Graziano,  
Fabrizio Riente and  
Giovanna Turvani

Received: 25 January 2022

Accepted: 15 March 2022

Published: 16 March 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

The current computing system is based on the von Neumann architecture that is based on physically separated memory and processor units. Currently, processor unit performance has rapidly progressed while memory access performance has not. This results in large energy consumption during data transfer between memory and processor units, thus reducing the computing performance [1–3]. This computing system throughput limitation due to the inadequate rate of data transfer between the memory and the CPU is called the von Neumann bottleneck or memory wall [4–6]. To address this problem, in-memory computing (IMC), which performs computation by embedding logic in the memory array, has been studied recently [7,8]. IMC reduces memory–processor data transfers and thus improves performance by reducing energy consumption [9,10].

This paper explains an 8<sup>+</sup>T SRAM IMC circuit [11] based on 8<sup>+</sup>T differential static random-access memory (8<sup>+</sup>T SRAM) [12] and proposes an 8<sup>+</sup>T SRAM-based IMC full adder (FA) and 8<sup>+</sup>T SRAM-based IMC approximate adder, which are based on the 8<sup>+</sup>T SRAM IMC circuit. The 8<sup>+</sup>T SRAM IMC circuit reads and computes simultaneously. Moreover, it performs logic computations parallelly when two words are selected simultaneously. The proposed IMC FA and IMC approximate adder can be applied to a multi-bit ripple carry adder (RCA). The proposed IMC FA and IMC approximate adder are based on the 8<sup>+</sup>T SRAM IMC circuit and thus read and compute simultaneously without SRAM read access. The SRAM-based IMC adder proposed in this study provides not only basic SRAM operations (data storage and reading) but also parallel Boolean functions and allows easy bitwise addition with minimal additional logic gates. An IMC approximate adder based on this mechanism is also proposed for better energy efficiency. 8<sup>+</sup>T SRAM, which has separate word lines for write and read, is the SRAM cell proposed in [12], and additional gates are

connected to the  $8^+T$  SRAM to take the advantage of two read bit lines (i.e., RBL and RBLB) for Boolean functions and addition. Though additional area overhead is required in the proposed IMC adder, the IMC adder proposed in this study has a simple structure and enables fast computation with low power as the operation unit is physically connected right next to the SRAM array.

Simulations in 65 nm technology show that the  $8^+T$  SRAM IMC circuit is faster and consumes less energy than the IMC circuit proposed in [7]. The proposed IMC FA reads and computes simultaneously without SRAM read access and thus consumes much less energy because it does not require data to be loaded into the processor for computation. The proposed 8-bit IMC RCA, which consists of the proposed IMC FA, is 25% faster and consumes 53% less total energy than the  $8^+T$  SRAM read + 8-bit RCA. The proposed 8-bit IMC approximate adder, which consists of the proposed IMC FA in the upper 4 bits and the proposed IMC approximate adder in the lower 4 bits, is 43% faster and consumes 15% less total energy than the proposed 8-bit accurate IMC RCA with error values comparable with those of other approximate adders.

Our main contributions are as follows:

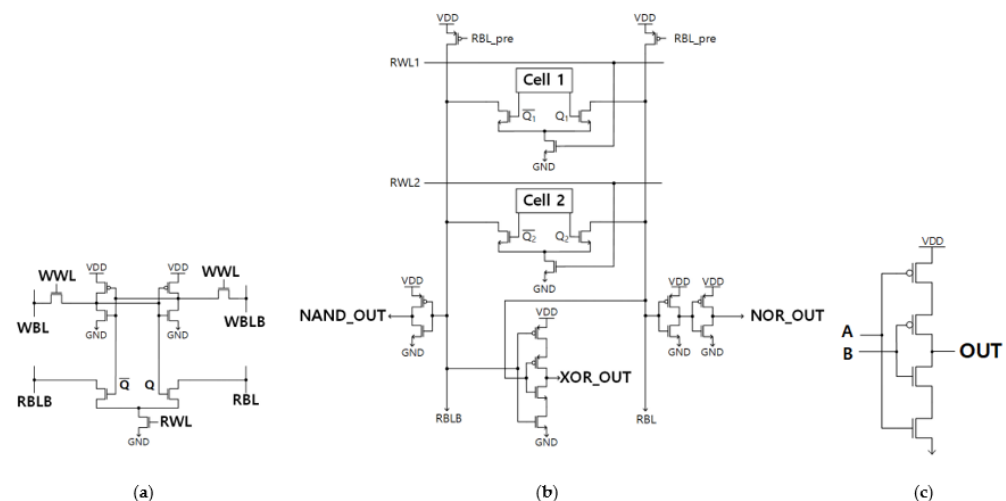
- We propose novel IMC units based on the  $8^+T$  SRAM cells. The proposed IMC units are extensively studied on various design parameters.
- We propose a novel IMC adder based on the IMC units.
- We propose a novel IMC approximate adder.
- We perform extensive studies on various design parameters for the proposed accurate and approximate adders.

The remainder of this paper proceeds as follows. Section 2 describes the mechanism for bitwise computation of the  $8^+T$  SRAM IMC circuit and its improved performance compared with the other IMC circuit. Section 3 explains the proposed IMC FA and IMC 8-bit RCA. An approximate adder based on the proposed IMC adder is described in Section 4. Finally, Section 5 concludes the paper.

## 2. $8^+T$ SRAM IMC Circuit

### 2.1. Structure

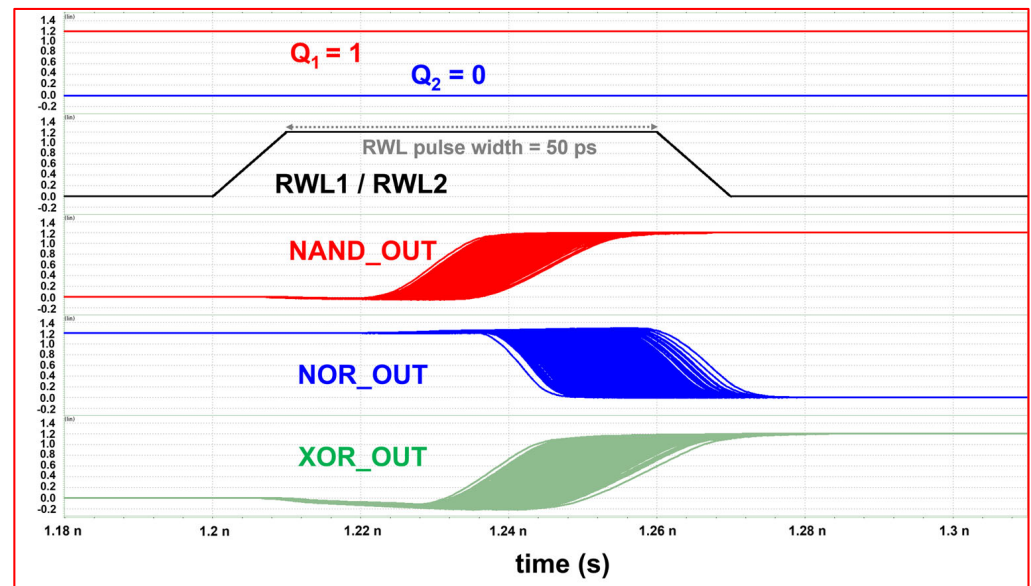
This section explains the  $8^+T$  SRAM IMC circuit in [11]. Figure 1b shows that the  $8^+T$  SRAM IMC circuit is based on  $8^+T$  Differential SRAM [12] and consists of inverters and a 2-input Muller C-element. Each inverter, buffer, and 2-input Muller C-element in node RBL and RBLB performs different logic computation. Node RBL and RBLB are initially pre-charged to '1'. Two words are selected, the inverter, buffer, and the Muller C-element output NAND, NOR, and XOR operation, respectively, as explained in [11].



**Figure 1.** (a)  $8^+T$  Differential SRAM Cell [12]; (b)  $8^+T$  Differential SRAM-based IMC circuit [11]; (c) the schematic of 2-input Muller C-element.

## 2.2. Impact of Process Variations

To quantify the impact of process variations on the IMC operations, Monte Carlo simulations are run under global and local mismatch variations. Figure 2, which is simulated in [11], shows SPICE transient simulations of global Monte Carlo (i.e., global + 3-sigma local mismatch variations) using a commercial 65 nm technology for NAND/NOR/XOR computations in the case of input “10/01.” Figure 2 shows that the 8<sup>+</sup>T SRAM IMC circuit performs NAND/NOR/XOR computations well in the case of input “10/01” and shows how the RWL pulse width is set to 50 ps. As RWL1/RWL2 rises simultaneously, the stored SRAM cell is read, and the logic computations are completed until RWL1/RWL2 falls.



**Figure 2.** Global Monte Carlo simulation when  $Q_1, Q_2$  is “10” or “01,” [11].

In all input cases, the minimum RWL pulse width for the 8<sup>+</sup>T SRAM IMC circuit to perform logical computations was 15 ps. However, to allow the logic computations to be completed until the negative edge of the RWL, the RWL pulse width in Figure 2 was set to 50 ps, and the rise and fall times of RWL were set to 10 ps. Moreover, this paper only shows Monte/Carlo simulation in the case of input “10/01” because the minimum RWL pulse width in cases of input “10/01” is the worst case [11].

## 2.3. Performance

Table 1 compares the logic computation delay of the 8<sup>+</sup>T SRAM IMC circuit, 8<sup>+</sup>T SRAM Read + logic computation, and 8T SRAM skewed inverters [7]. The 8<sup>+</sup>T SRAM Read + Logic computation is the virtual case used as a comparison target, as shown in Table 1. The difference between the 8<sup>+</sup>T SRAM Read + Logic computation and the 8<sup>+</sup>T SRAM IMC circuit is that conventional logic gates are connected to the 8<sup>+</sup>T SRAM cell in the case of the 8<sup>+</sup>T SRAM Read + Logic computation; however, the inverters and Muller C-element are connected to the 8<sup>+</sup>T SRAM cell in the case of the 8<sup>+</sup>T SRAM IMC circuit. Moreover, the 8<sup>+</sup>T SRAM IMC circuit performs logic computations when two words are selected simultaneously, but the 8<sup>+</sup>T SRAM Read + Logic computation is performed one by one. 8T SRAM skewed inverters also perform logic computations when two words are selected simultaneously. All simulation about delays and power consumptions were measured at the TT corner at room temperature (25 °C) with Hspice, the transistor-level simulation tool. We calculated the total energy by integrating the power consumption (the current and VDD) from the pre-charge of the SRAM cell for the read operation to the completion time of the adder computation using a built-in tool of the waveform viewer (i.e., Custom Waveview of Synopsys).

**Table 1.** Comparison of logic computation delay in different circuits.

| Circuits                                       | NAND       |            | NOR        |            | XOR        |            |
|--|------------|------------|------------|------------|------------|------------|
|  | Delay [ps] | Normalized | Delay [ps] | Normalized | Delay [ps] | Normalized |
| 8T SRAM skewed inverters [7]                   | 46.66      | 1.00       | 54.94      | 1.00       | 81.34      | 1.00       |
| 8 <sup>+</sup> T SRAM Read + Logic computation | 45.04      | 0.97       | 45.35      | 0.83       | 100.32     | 1.23       |
| 8 <sup>+</sup> T SRAM IMC circuit [11]         | 28.57      | 0.61       | 44.37      | 0.81       | 39.87      | 0.49       |

Table 1 shows that the 8<sup>+</sup>T SRAM IMC circuit has better NAND, NOR, and XOR performance than the 8T SRAM skewed inverters and the 8<sup>+</sup>T SRAM Read + Logic computation. The 8<sup>+</sup>T SRAM IMC circuit performs NAND, NOR, and XOR computations 40%, 20%, and 50% faster than the 8T SRAM skewed inverters, respectively. Moreover, the 8<sup>+</sup>T SRAM IMC circuit is faster than the 8<sup>+</sup>T SRAM Read + Logic computation.

Table 2 compares the logic computation performance and energy consumption of the 8<sup>+</sup>T SRAM IMC circuit, 8<sup>+</sup>T SRAM Read + Logic computation, and 8T SRAM skewed inverters. The 8<sup>+</sup>T SRAM IMC circuit has a 55% lower power-delay product (PDP) and uses 60% less total energy than the 8T SRAM skewed inverters. Its average power consumption is higher than that of the 8<sup>+</sup>T SRAM Read + Logic computation; however, its PDP is 29% less, and the total energy consumption is 72% less than that of the 8T SRAM Read + Logic computation. Table 2 shows that the 8<sup>+</sup>T SRAM IMC circuit is faster and consumes less energy than the other circuit.

**Table 2.** Comparison of logic computation performance and energy consumption in different circuits.

| Circuits                                       | Delay  |       | Average Power Consumption |       | PDP  |       | Total Energy Consumption |       |
|--|--------|-------|---------------------------|-------|------|-------|--------------------------|-------|
|  | [ps]   | Norm. | [μW]                      | Norm. | [fj] | Norm. | [fj]                     | Norm. |
| 8T SRAM skewed inverters [7]                   | 81.34  | 1.00  | 62.1                      | 1.00  | 5.05 | 1.00  | 15.85                    | 1.00  |
| 8 <sup>+</sup> T SRAM Read + Logic computation | 100.32 | 1.23  | 31.78                     | 0.51  | 3.19 | 0.63  | 22.68                    | 1.43  |
| 8 <sup>+</sup> T SRAM IMC circuit [11]         | 44.37  | 0.55  | 51.35                     | 0.83  | 2.28 | 0.45  | 6.30                     | 0.40  |

### 3. Proposed IMC Full Adder

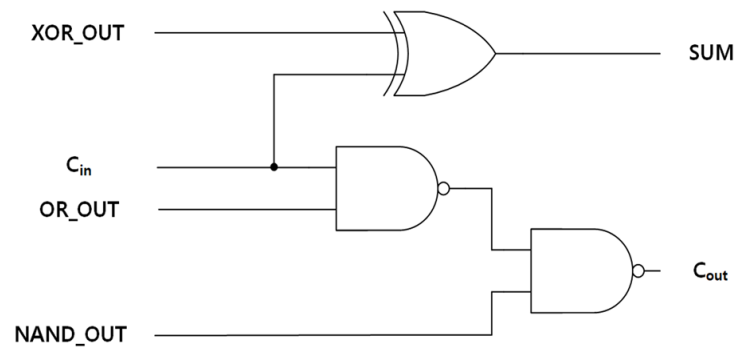
This section discusses an IMC FA and 8-bit RCA, which are based on the 8<sup>+</sup>T SRAM IMC circuit explained in the previous section. The proposed IMC FA can be applied to multi-bit RCA; thus, this section also compares the proposed IMC 8-bit RCA with other adders. The performances of the proposed IMC FA, SRAM read access [13], and 8<sup>+</sup>T SRAM Read + Full Adder were used for the comparison.

#### 3.1. 1-bit Full Adder

Figure 3 shows additional gates for the proposed IMC FA. This adder is implemented by connecting logic gates to the logic outputs (NAND\_OUT, OR\_OUT, XOR\_OUT) in the 8<sup>+</sup>T SRAM IMC circuit. Equations (1) and (2) represent the computations of the proposed IMC FA. The proposed IMC FA is based on the 8<sup>+</sup>T SRAM IMC circuit and thus operates when two words are selected simultaneously.

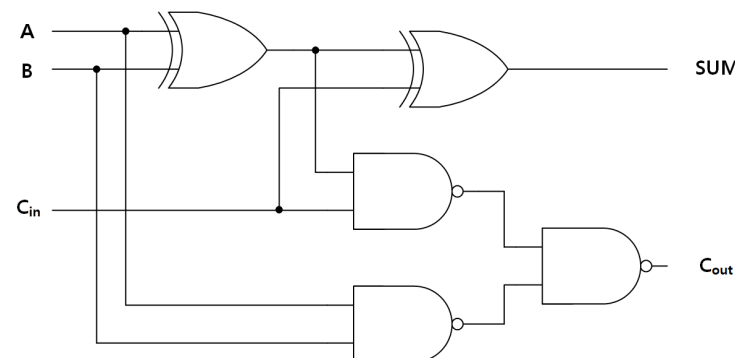
$$Sum = A \oplus B \oplus C_{in} \quad (1)$$

$$\begin{aligned} Carry &= \overline{(A \oplus B) \cdot C_{in} + A \cdot B} \\ &= \overline{(A \oplus B) \cdot C_{in} \cdot A \cdot B} \\ &= \overline{(A + B) \cdot C_{in} \cdot A \cdot B} \end{aligned} \quad (2)$$



**Figure 3.** Additional gates for the proposed IMC full adder.

Except for the  $8^+T$  SRAM cell, which is the same for both cases, the proposed IMC FA uses eight fewer transistors than the conventional FA (as shown in Figures 3 and 4). The conventional FA uses two XOR gates (conventional XOR gate using 12 transistors) and three NAND gates, thus using a total of 36 transistors. By contrast, the proposed IMC FA replaces one XOR gate with a two-input Muller C-element and one NAND gate with an inverter. Moreover, as shown in Equation (2), this adder requires an OR operation; therefore, it uses the OR\_OUT node and requires two more transistors. Therefore, the proposed IMC FA uses 28 transistors in the adder except for the  $8^+T$  SRAM cell and eight fewer transistors than the conventional FA [14].



**Figure 4.** Conventional full adder [14].

### 3.2. Performance and Energy Consumption

Table 3 compares the performance and energy consumption of the proposed IMC FA, the proposed IMC approximate adder, SRAM read access, and  $8^+T$  SRAM Read + FA. The proposed IMC approximate adder is explained in Section 4, and the performance is compared in that section. The  $8^+T$  SRAM Read + FA, which consists of  $8^+T$  SRAM and the conventional full adder, is the virtual case used as a comparison target, as shown in Table 3. It first selects two words one by one from the  $8^+T$  SRAM Cell, reads the cell, and then transfers it to the input of the connected conventional FA to compute. It selects words one by one, whereas the proposed IMC FA selects all at once. SRAM read access only shows the operation of the processor accessing SRAM in the von Neumann architecture. Thus, SRAM read access is required in the conventional systems to perform logic and arithmetic operations in the processor. In this paper, the  $8^+T$  SRAM Read + FA is the virtual case used as a reference to compare the performance of the proposed IMC adder. Therefore, the delay and energy consumption of the SRAM read access should be added to the  $8^+T$  SRAM Read + FA case in Table 3 to indicate a practical operation.

According to Table 3, the proposed IMC FA was 36% faster than the  $8^+T$  SRAM Read + FA. This is because the proposed IMC FA uses simple gates (Muller C-element, inverters) compared to the  $8^+T$  SRAM Read + FA; thus, the propagation delay in the critical path is shorter in the proposed IMC FA. By contrast, the average power consumption of the

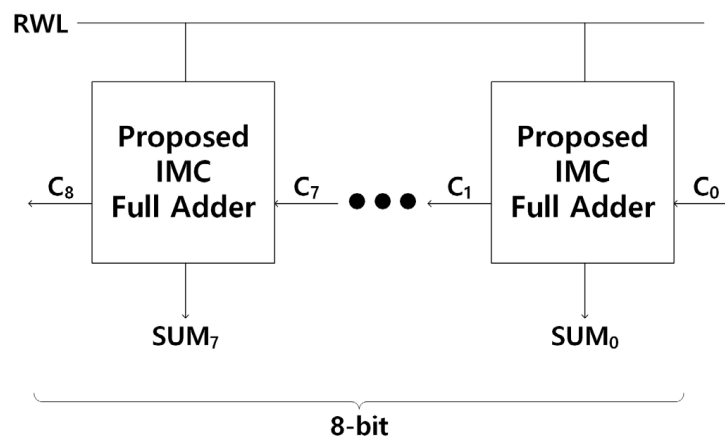
proposed IMC FA is higher than that of the 8<sup>+</sup>T SRAM Read + FA because the proposed IMC FA consumes power within a shorter time than the 8<sup>+</sup>T SRAM Read + FA. In this study, the power consumption of the 8<sup>+</sup>T SRAM Read + FA and the SRAM read access are separated to compare the performance and energy consumption only for the SRAM read and computation of the adder. Therefore, the actual average power consumption of the 8<sup>+</sup>T SRAM Read + FA should be added to the average power consumption of the SRAM read access (a few milliwatts), which is eventually higher than that of the proposed IMC FA.

**Table 3.** Comparison of the performance and energy consumption in different full adders.

| Full Adders                             | Delay  |       | Average Power Consumption |                      | PDP    |                      | Total Energy Consumption |       |
|---|--------|-------|---------------------------|----------------------|--------|----------------------|--------------------------|-------|
|   | [ps]   | Norm. | [μW]                      | Norm.                | [fj]   | Norm.                | [fj]                     | Norm. |
| SRAM read access [13]                   | 1152   | 1.00  | 11,277                    | 1.00                 | 12,991 | 1.00                 | -                        | -     |
| 8 <sup>+</sup> T SRAM Read + Full Adder | 138.39 | 0.12  | 45.50                     | $4.0 \times 10^{-3}$ | 6.30   | $4.8 \times 10^{-4}$ | 13.60                    | 1.00  |
| Proposed IMC Full Adder                 | 88.13  | 0.08  | 75.85                     | $6.7 \times 10^{-3}$ | 6.68   | $5.1 \times 10^{-4}$ | 14.31                    | 1.05  |
| Proposed IMC Approximate Adder          | 88.35  | 0.08  | 69.45                     | $6.2 \times 10^{-3}$ | 6.14   | $4.7 \times 10^{-4}$ | 13.97                    | 1.03  |

3.3. 8-bit Ripple Carry Adder

Figure 5 shows a diagram of the proposed 8-bit IMC RCA. This adder was implemented as an 8-bit RCA using the proposed IMC FA. It is based on the 8<sup>+</sup>T SRAM IMC circuit; thus, it reads and computes simultaneously and operates when two words are selected at the same time. Table 4 compares the performance of the proposed 8-bit IMC RCA, proposed 8-bit IMC approximate adder, SRAM read access, and 8<sup>+</sup>T SRAM Read + 8-bit RCA. The proposed 8-bit IMC approximate adder is explained in Section 4, and the performance is compared in that section.



**Figure 5.** Diagram of the proposed 8-bit IMC Ripple Carry Adder.

**Table 4.** Comparison of the performance and energy consumption in different 8-bit Ripple Carry Adders.

| 8-bit Ripple Carry Adders              | Delay  |       | Average Power Consumption |       | PDP    |       | Total Energy Consumption |       |
|--|--------|-------|---------------------------|-------|--------|-------|--------------------------|-------|
|  | [ps]   | Norm. | [μW]                      | Norm. | [fj]   | Norm. | [fj]                     | Norm. |
| SRAM read access [13]                  | 1152   | 1.00  | 11,277                    | 1.00  | 12,991 | 1.00  | -                        | -     |
| 8 <sup>+</sup> T SRAM Read + 8-bit RCA | 450.62 | 0.39  | 308.3                     | 0.027 | 138.93 | 0.011 | 324.20                   | 1.00  |
| Proposed 8-bit IMC RCA                 | 337.87 | 0.29  | 447.5                     | 0.040 | 151.20 | 0.012 | 153.69                   | 0.47  |
| Proposed 8-bit IMC Approximate Adder   | 193.08 | 0.17  | 470.5                     | 0.042 | 90.84  | 0.007 | 131.37                   | 0.40  |

In this study, the 8<sup>+</sup>T SRAM Read + 8-bit RCA and the SRAM read access are separated to compare the performance and energy consumption only for the SRAM read and computation of the adder. From this viewpoint, in Table 4, the proposed 8-bit IMC RCA is 25% faster and consumes 53% less total energy than the 8<sup>+</sup>T SRAM Read + 8-bit RCA. As with the case of 1-bit FA, the proposed 8-bit IMC RCA is faster than the 8<sup>+</sup>T SRAM Read + 8-bit RCA, but its average power consumption and PDP are higher. By contrast, in reality, the integrated operation of the 8<sup>+</sup>T SRAM Read + 8-bit RCA and the SRAM read access run as a real processor. Thus, when comparing in a practical operation, the delay and power required for SRAM read access should be added to those of the 8<sup>+</sup>T SRAM Read + 8-bit RCA case in Table 4, since the SRAM read access is still required in the conventional system. On the other hand, since the proposed IMC RCA does not require SRAM read access, the delay and power of the proposed IMC RCA are much smaller than conventional.

**4. Proposed Approximate Adder**

This section discusses an IMC approximate adder, which is based on the 8<sup>+</sup>T SRAM IMC circuit. The proposed IMC approximate adder is implemented by connecting the approximate adder AFA<sub>3</sub> [15] (shown in Figure 6) to the 8<sup>+</sup>T SRAM IMC circuit. The proposed IMC approximate adder also operates as an 8-bit adder consisting of the proposed IMC FA in the upper 4 bits and the proposed IMC approximate adder in the lower 4 bits. Moreover, since it is based on the 8<sup>+</sup>T SRAM IMC circuit, it operates when two words are selected simultaneously.

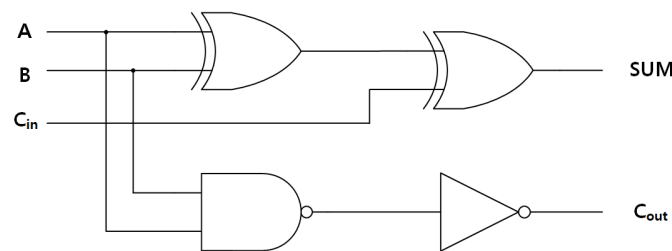


Figure 6. Schematic of AFA<sub>3</sub> [15].

Figure 7 shows additional gates for the proposed IMC approximate adder. The IMC approximate adder is implemented by connecting the output of the logic gates (NAND\_OUT, XOR\_OUT) in the 8<sup>+</sup>T SRAM IMC circuit. Equations (3) and (4) represent the computations of the proposed IMC approximate adder and AFA<sub>3</sub>. According to Equation (4), the carry computation of the proposed IMC approximate adder and AFA<sub>3</sub> is different from that of the conventional FA. Table 5 shows the truth table of the accurate full adder and the proposed IMC approximate adder. The carry of the proposed IMC approximate adder output errors in certain cases.

$$Sum = A \oplus B \oplus C_{in} \tag{3}$$

$$Carry = A \cdot B \tag{4}$$

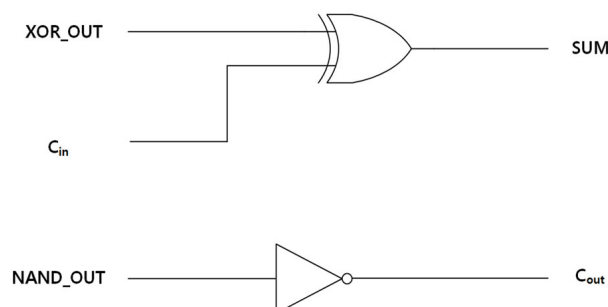
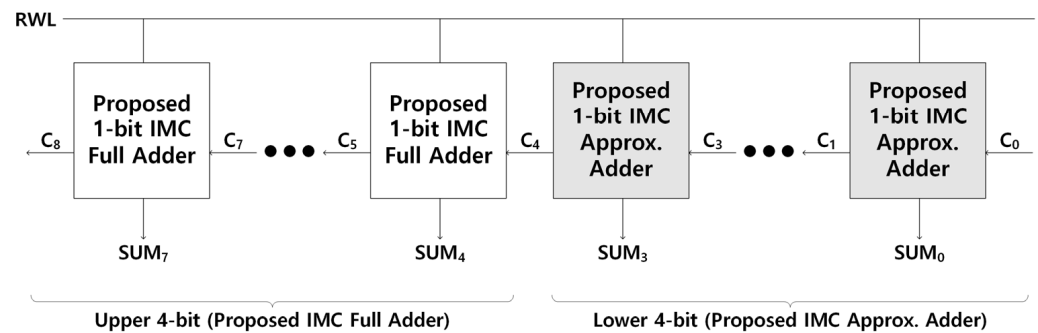


Figure 7. Additional gates for the proposed IMC approximate adder.

**Table 5.** Truth table of the accurate full adder and the proposed IMC approximate adder.

| A | B | C <sub>in</sub> | Carry    |             | Sum      |             |
|---|---|-----------------|----------|-------------|----------|-------------|
|   |   |                 | Accurate | Approximate | Accurate | Approximate |
| 0 | 0 | 0               | 0        | 0           | 0        | 0           |
| 0 | 0 | 1               | 0        | 0           | 1        | 1           |
| 0 | 1 | 0               | 0        | 0           | 1        | 1           |
| 0 | 1 | 1               | 1        | 0           | 0        | 0           |
| 1 | 0 | 0               | 0        | 0           | 1        | 1           |
| 1 | 0 | 1               | 1        | 0           | 0        | 0           |
| 1 | 1 | 0               | 1        | 1           | 0        | 0           |
| 1 | 1 | 1               | 1        | 1           | 1        | 1           |

Figure 8 shows a diagram of the proposed 8-bit IMC approximate adder. It consists of the proposed IMC FA in the upper 4 bits and the proposed IMC approximate adder in the lower 4 bits. In the lower 4 bits, it can output an error. By contrast, since each lower 4 bit independently computes, it is faster than the proposed 8-bit IMC RCA.



**Figure 8.** Diagram of the proposed 8-bit IMC approximate adder.

4.1. Performance and Energy Consumption

According to Table 3, comparing the performance and energy consumption of the proposed IMC FA, proposed IMC approximate adder, SRAM read access, and the 8<sup>+</sup>T SRAM Read + FA, the proposed IMC approximate adder has no significant improvement in computation delay and energy consumption compared to the proposed IMC FA. By contrast, according to Table 4, which compares the performance and energy consumption of the proposed 8-bit IMC RCA, proposed 8-bit IMC approximate adder, SRAM read access, and 8<sup>+</sup>T SRAM Read + 8-bit RCA, the proposed 8-bit IMC approximate adder is 43% faster, and the total energy consumption is approximately 15% lower than that of the proposed 8-bit IMC RCA. The average power consumption is higher for the proposed 8-bit IMC approximate adder, as it consumes power within a shorter time than the proposed 8-bit IMC RCA.

Equation (5) represents the computation for the case where both the proposed 8-bit IMC RCA and the proposed 8-bit IMC approximate adder compute the worst case; the computation results of the adders are the same. Figures 9 and 10 show the computation results for Equation (5) of the proposed 8-bit IMC RCA and the proposed 8-bit IMC approximate adder, respectively. Comparing Figures 9 and 10 explains why the proposed 8-bit IMC approximate adder is faster than the proposed 8-bit IMC RCA.

$$\begin{array}{r}
 0000\ 0000 \\
 + 0111\ 1111 \\
 \hline
 0111\ 1111
 \end{array} \tag{5}$$



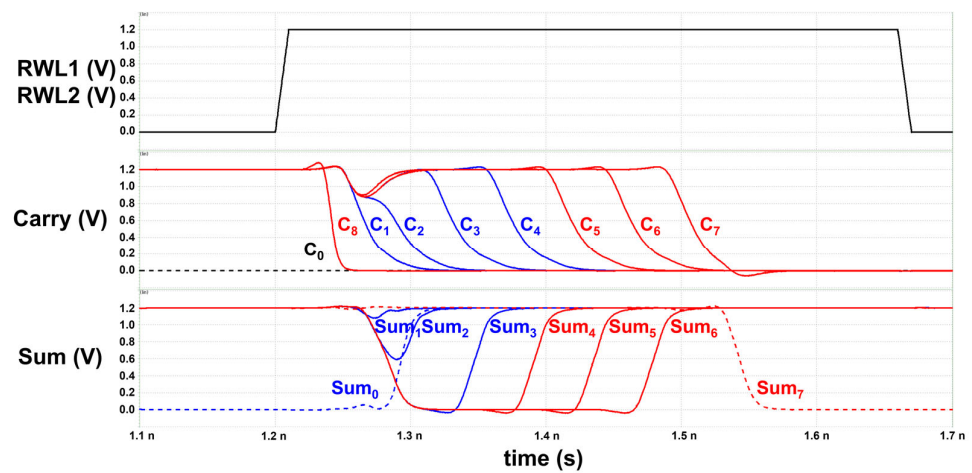


Figure 9. Timing graph of the proposed 8-bit IMC Ripple Carry Adder.

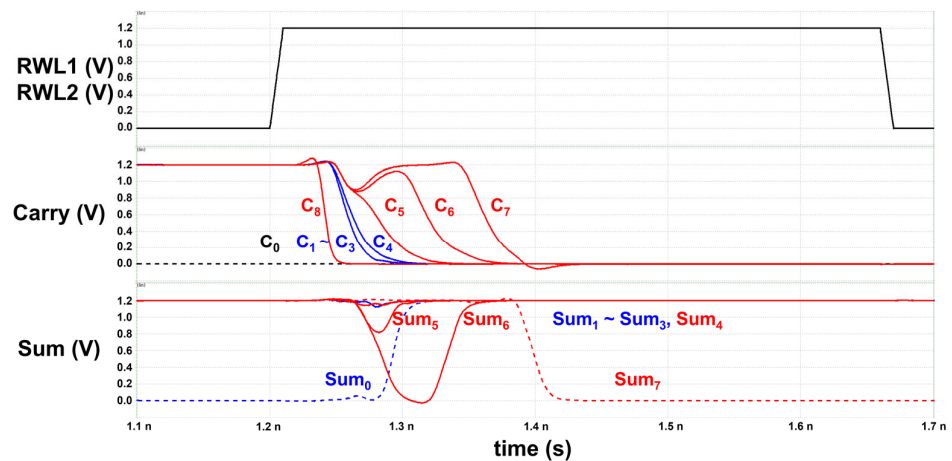


Figure 10. Timing graph of the proposed 8-bit IMC approximate adder.

In Figures 9 and 10, the black line indicates the RWL, the red line indicates the carry and sum of the upper 4 bits, and the blue line indicates the carry and sum of the lower 4 bits. The red dotted line indicates the sum of the highest bit; the sum of the highest bit is marked as a dotted line because its output is different from that of the other bits. The blue dotted line indicates the sum of the lowest bit; the sum of the lowest bit is marked as a dotted line because its initial value is different from that of the other bits. The black dotted line indicates the carry-in of the lowest bit; the carry-in of the lowest bit is marked as a dotted line because its initial value is different from that of the other bits. After the node RBL and RBLB are pre-charged, the initial carry-in of the 2nd to 8th bit ( $C_1 \sim C_7$ ) is “high” for both the proposed 8-bit IMC RCA and the proposed 8-bit IMC approximate adder. By contrast, since the initial carry-in of the lowest bit ( $C_0$ ) is “low,” the initial sum of the lowest bit ( $Sum_0$ ) is different from that of the other bits.

According to Figures 9 and 10, the proposed 8-bit IMC approximate adder consisting of the proposed IMC approximate adder in the lower 4 bits is faster than the proposed 8-bit IMC RCA. This is because, in the upper 4 bits where the carry ripples, the computation mechanisms of the two adders are the same; however, in the lower 4 bits, the carry of the proposed 8-bit IMC approximate adder is independently computed. The adder that independently computes without being affected by the carry-out of the previous bit is faster than the RCA.

In Figures 9 and 10, as RWL1/RWL2 increases, the IMC adders read and compute, and all the adder operations are completed until RWL1/RWL2 falls. Section 2 explains that the minimum pulse width of the RWL for the  $8^+T$  SRAM IMC circuit to perform the logic

operation is set to 15 ps, and the minimum pulse width of RWL for all logic operations to be completed until RWL falls is set to 50 ps. By contrast, the minimum pulse width of RWL for the 8-bit adders compared in this study (the proposed 8-bit IMC RCA, the proposed 8-bit IMC approximate adder, 8<sup>+</sup>T SRAM Read + 8-bit RCA) to be completed until RWL falls was set to 450 ps; rise and fall times were set to 10 ps.

#### 4.2. Error Metrics Comparison

Table 6 compares the errors of the proposed 8-bit IMC approximate adder, BCSA [16], SARA [17], and RAP-CLA [18] for 8-bit, block size 4. The Block-based Carry Specific Approach Adder (BCSA) corrects the errors with an additional error recovery unit [16]. The simple accuracy-reconfigurable adder (SARA) operates through the error correction stage [17]. The reconfigurable approach carries a look-ahead adder (RAP-CLA) based on the carry-look-ahead adder (CLA) operates in two modes: approximate adder mode and accurate adder mode [18]. To compare the errors of the approximate adders, the normalized mean error distance (NMED), average relative error distance (MRED), and error rate (ER) were used as indicators. According to Table 6, the proposed 8-bit IMC approximate adder has a similar NMED to other approximate adders. By contrast, it has a higher ER because it does not correct errors, unlike the other approximate adders.

**Table 6.** Comparison of errors in different approximate adders (8-bit, block size of 4).

| 8-bit Approximate Adders             | Block Size | NMED (10 <sup>-3</sup> ) | MRED (10 <sup>-3</sup> ) | ER (%) |
|--------------------------------------|------------|--------------------------|--------------------------|--------|
| BCSA [16]                            | 4          | 3.4                      | 5.2                      | 5.46   |
| SARA [17]                            | 4          | 6.8                      | 9.3                      | 5.46   |
| RAP-CLA [18]                         | 4          | 13.7                     | 14.7                     | 2.34   |
| Proposed 8-bit IMC Approximate Adder | 4          | 6.8                      | 18.2                     | 35.8   |

## 5. Conclusions

In this paper, the 8<sup>+</sup>T differential SRAM-based IMC circuit (8<sup>+</sup>T SRAM IMC circuit) is explained, and the IMC FA and approximate adder based on the 8<sup>+</sup>T SRAM IMC circuit are proposed. The 8<sup>+</sup>T SRAM IMC circuit, FA, and approximate adder operate when two words are selected simultaneously. They also read and compute simultaneously without SRAM read access. The 8<sup>+</sup>T SRAM IMC circuit was 45% faster and had a 17% lower average power consumption, 55% lower PDP, and 60% lower total energy consumption than the 8T SRAM skewed inverters. Moreover, it was 56% faster, had 29% lower PDP, and consumed 72% less total energy than the 8<sup>+</sup>T SRAM Read + Logic computation. The proposed 8-bit IMC RCA consisting of the proposed IMC FA was 25% faster and consumed 53% lower total energy than the 8<sup>+</sup>T SRAM Read + 8-bit RCA. The proposed 8-bit IMC approximate adder consisting of the proposed IMC FA in the upper 4 bits and the proposed IMC approximate adder in the lower 4 bits has similar NMED but higher ER than other 8-bit approximate adders compared in this study. In contrast, it was 43% faster and consumed 15% less total energy than the proposed 8-bit IMC RCA.

The 8<sup>+</sup>T SRAM IMC circuit was applied to the adders, and its performance and energy consumption were measured in this study. The adders proposed herein are consistent with the purpose of IMC, which aims to use reduce the energy consumption and improve the overall performance.

**Author Contributions:** Conceptualization, S.S. and Y.K.; methodology, S.S.; software, S.S.; validation, S.S. and Y.K.; investigation, S.S.; resources, Y.K.; writing—original draft preparation, S.S.; writing—review and editing, S.S. and Y.K.; visualization, S.S.; supervision, Y.K.; project administration, Y.K.; funding acquisition, Y.K. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by Ministry of Education under grand number NRF-2020R1F1A1055251.

**Acknowledgments:** This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) and funded by the Ministry of Education (NRF-2020R1F1A1055251). This work was supported by the National Research Foundation (NRF), Korea, under project BK21 FOUR. The EDA tool was supported by IC Design Education Center (IDEC), Korea.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Yeswanth, C.; Acharya, A. In-memory Computing based Boolean and logical Circuit Design using 8T SRAM. In Proceedings of the 2021 Devices for Integrated Circuit (DevIC), Kalyani, India, 19–20 May 2021; pp. 430–434.
2. Chen, Y.; Lu, L.; Kim, B.; Kim, T.T.-H. Reconfigurable 2T2R ReRAM with Split Word-Lines for TCAM Operation and In-Memory Computing. In Proceedings of the 2020 IEEE International Symposium on Circuits and Systems (ISCAS), Seville, Spain, 12–14 October 2020; pp. 1–5.
3. Reuben, J. Rediscovering Majority Logic in the Post-CMOS Era: A Perspective from In-Memory Computing. *J. Low Power Electron.* **2020**, *10*, 28. [\[CrossRef\]](#)
4. Gauchi, R.; Kooli, M.; Vivet, P.; Noel, J.-P.; Beigné, E.; Mitra, S.; Charles, H.-P. Memory Sizing of a Scalable SRAM In-Memory Computing Tile Based Architecture. In Proceedings of the 2019 IFIP/IEEE 27th International Conference on Very Large Scale Integration (VLSI-SoC), Cuzco, Peru, 6–9 October 2019; pp. 166–171.
5. Gupta, A.K.; Acharya, A. Exploration of 9T SRAM Cell for In Memory Computing Application. In Proceedings of the 2021 Devices for Integrated Circuit (DevIC), Kalyani, India, 19–20 May 2021; pp. 461–465.
6. Lue, H.-T.; Hu, H.-W.; Hsu, T.-H.; Hsu, P.-K.; Wang, K.-C.; Lu, C.-Y. Design of Computing-in-Memory (CIM) with Vertical Split-Gate Flash Memory for Deep Neural Network (DNN) Inference Accelerator. In Proceedings of the 2021 IEEE International Symposium on Circuits and Systems (ISCAS), Daegu, Korea, 22–28 May 2021; pp. 1–4.
7. Agrawal, A.; Jaiswal, A.; Lee, C.; Roy, K. X-SRAM: Enabling In-Memory Boolean Computations in CMOS Static Random Access Memories. *IEEE Trans. Circuits Syst. I Regul. Papers* **2018**, *65*, 4219–4232. [\[CrossRef\]](#)
8. Luo, T.; Zhang, W.; He, B.; Liu, C.; Maskell, D. Energy Efficient In-memory Integer Multiplication Based on Racetrack Memory. In Proceedings of the 2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS), Singapore, 29 November–1 December 2020; pp. 1409–1414.
9. Chen, H.-C.; Li, J.-F.; Hsu, C.-L.; Sun, C.-T. Configurable 8T SRAM for Enabling in-Memory Computing. In Proceedings of the 2019 2nd International Conference on Communication Engineering and Technology (ICCET), Nagoya, Japan, 12–15 April 2019; pp. 139–142.
10. Rajput, A.K.; Pattanaik, M. Implementation of Boolean and Arithmetic Functions with 8T SRAM Cell for In-Memory Computation. In Proceedings of the 2020 International Conference for Emerging Technology (INCET), Belgaum, India, 5–7 June 2020; pp. 1–5.
11. Song, S.; Kim, Y. Novel In-memory Computing Circuit using Muller C-element. In Proceedings of the 2021 18th International SoC Design Conference (ISOCC), Jeju Island, Korea, 6–9 October 2021; pp. 81–82.
12. Kulkarni, J.P.; Goel, A.; Ndai, P.; Roy, K. A Read-Disturb-Free, Differential Sensing 1R/1W Port, 8T Bitcell Array. *IEEE Trans. VLSI Syst.* **2011**, *19*, 1727–1730. [\[CrossRef\]](#)
13. Wu, S.; Zheng, X.; Gao, Z.; He, X. A 65nm embedded low power SRAM compiler. In Proceedings of the 13th IEEE Symposium on Design and Diagnostics of Electronic Circuits and Systems, Vienna, Austria, 14–16 April 2010; pp. 123–124.
14. Weste, N.; Harris, D. *CMOS VLSI Design: A Circuits and Systems Perspective*, 4th ed.; Pearson Education Korea: Seoul, Korea, 2011; pp. 432–438.
15. Dutt, S.; Nandi, S.; Trivedi, G. Analysis and Design of Adders for Approximate Computing. *ACM Trans. Embed. Comput. Syst.* **2018**, *17*, 1–28. [\[CrossRef\]](#)
16. Ebrahimi-Azandaryani, F.; Akbari, O.; Kamal, M.; Afzali-Kusha, A.; Pedram, M. Block-Based Carry Speculative Approximate Adder for Energy-Efficient Applications. *IEEE Trans. Circuits Syst. II Express Briefs* **2020**, *67*, 137–141. [\[CrossRef\]](#)
17. Xu, W.; Sapatnekar, S.S.; Hu, J. A Simple yet Efficient Accuracy-Configurable Adder Design. *IEEE Trans. VLSI Syst.* **2018**, *26*, 1112–1125. [\[CrossRef\]](#)
18. Akbari, O.; Kamal, M.; Afzali-Kusha, A.; Pedram, M. RAP-CLA: A Reconfigurable Approximate Carry Look-Ahead Adder. *IEEE Trans. Circuits Syst. II Express Briefs* **2018**, *65*, 1089–1093. [\[CrossRef\]](#)