International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Noise Immune Schmitt Trigger Logic Gate Based Disturb Free Scalable 9T SRAM In Memory Computing Design

Mucherla Usha Rani[1], Siva Sankara Reddy.N[2], Rajendra Naik.B[3]

Submitted:14/05/2024 **Revised**: 25/06/2024 **Accepted**: 03/07/2024

Abstract: In Von Neumann's architecture, memory serves as a storage component, whereas the arithmetic logic unit (ALU) serves as a computation device. However, when a lot of data flows between the memory and the ALU, Von Neumann bottlenecks remain a serious problem, resulting in limitations related to temporal overhead, throughput, and energy efficiency. This paper develops an In-memory computing (IMC) structure for computing ripple carry addition and multiplication (RCA-M) processes directly in the memory array. Here, Half-Select Disturb-Free Write Assist (HDFWA) Static random-access memories (SRAM) cell is designed with 9 transistors to optimize the read and write static noise margin (SNM) individually. The current compensation based sensor circuit is used to reduce leakage power and improve reading performance. In addition, the control circuits of the proposed IMC architecture, including SRAM control, IMC control, and RCA-M, are designed using Schmitt trigger (ST) logic to enable noise-immune, reliable, and efficient IMC systems. By using the ST logic, switching power is greatly reduced, and noise immunity is increased. The simulation outputs demonstrate that the suggested IMC for RCA-M outperforms other IMC architectures in terms of average energy efficiency (4.019 fJ) on CMOS 14nm technology.

Keywords: *Static random-access memories, In-memory computing (IMC), Schmitt trigger, static noise margin and Ripple carry adder and multiplication.*

1. Introduction

The field of artificial intelligence (AI) is a prestigious technology that has found extensive application across multiple domains, which demand a significant quantity of data flow and perform massive quantities of computations [1]. One fascinating method of removing the negative effects associated with data navigation is in-memory computing (IMC) [2]. Over the last few years, different IMC and Near Memory Computation (NMC) designs have been established. By performing computations inside the memory macro, computing-in-memory (CIM), also referred to as processing-in-memory (PIM) [3] introduced in order to remove the distinction between the processing and memory components.

Recent research findings suggest that Resistive Random Access Memory (RRAM) [4] is a viable option for IMCs in order to optimize the area. Yet, the marketability of RRAM-based IMC technology is hindered by several limitations, such as expensive write power, limited resilience, and small signal tolerances. The dynamic random access memory (DRAM)-based IMC [5] functions preserve a lot of storage. But, DRAM is ineffective and consumes a lot of energy because it needs to be renewed frequently. Employing RRAM, numerous IMC activities have been accomplished in recent years. Certain writers combined two 1T1R cells into a single group for performing the XNOR operations, and they used the 1T1R RRAM [6] array to perform both matrix and ternary multiplication operations. Compared to RRAM, spin-toque transfer magnetic random-access memory (STT-MRAM) [7] is favored due to its large capacity and almost negligible leakage energy. However, Compared to traditional memory types, RRAM as well as STT-MRAM are more expensive.

The SRAM is currently utilized in computing devices with various shapes and sizes, which is an economically established technology [8]. In order to increase operation time and decrease power dissipation, SRAM is typically manufactured utilizing modern techniques. However, the risk of leakage increases as technology advances to the nanoscale level [9]. Over the years, numerous SRAM macro designs have been developed for power and energy savings. The main designing strategies are Current compensation [10], Secondary supply [11], Current-mode sense amplification [12], and Split-cell supply [13]. Their applications were limited due to several problems. Although the current compensation method increased SRAM access speed, it was unable to reduce leakage current. Instead, the secondary supply approach uses an additional higher supply voltage to speed up access to the SRAM cell.

The In-Memory Boolean Computation (IMBC) methods have been carried out utilizing ordinary 6T SRAM by concurrently engaging two Word lines (WL) [14]. Nevertheless, because of its unavoidably lower frequency operation, the 6T SRAM-based IMC design encounters computational instability [15]. The usage of IMC in the 6T SRAM [16] causes serious reliability problems, including read interruption and deterioration of read distortion tolerances. Although using 8T [17] or 10T [18] units lessens these restrictions, using these cells comes with a larger area burden. In-

1Dept. of ECE, ACE Engineering College, Ghatkesar, Telangana 501301, India.
2Dept. of ECE, Vasavi College of Engineering, Ibrahimbagh, Hyderabad – 500031, Telangana, India
3Dept. of ECE,Osmania University,Hyderabad-500007,Telangana,India
** Corresponding Author Email:usharani250991@gmail.com*

memory computing often involves reading and writing data directly from and to memory cells integrated within processing units. These memory cells are susceptible to noise, which can result in errors during read and write operations. Schmitt trigger logic provides noise immunity by ensuring that signals are properly conditioned and interpreted, reducing the likelihood of errors caused by noise. This point motivated us to develop a new IMC structure for addition and multiplication operations using Schmitt logic based controller circuits. The main contributions of this research are listed as follows:

- To propose a new IMC design for performing several operations like addition and multiplication using a 9T SRAM array.
- To minimize the leakage power and enhance the speed of the reading process by using compensation-based sensor circuit in the SRAM design.
- To enable noise-immune, reliable, and efficient IMC systems by integrating Schmitt trigger logic with memory cells.
- To evaluate the effectiveness of the suggested IMC model by considering two 4-bit numbers for addition and multiplication.

The rest of the paper is structured as follows: Section 2 reviews the recent papers on IMC design for different applications. Section 3 explains the suggested design in detail. Section 4 validates the performance of the proposed model via simulation. Section 5 concludes the paper with future research directions.

2. Literature Survey

It is commonly acknowledged that using IMC may significantly increase energy efficiency. Operands in SRAM are saved in columns to achieve IMC that reverses SRAM write patterns and necessitates extra data transfer. Zhang et al. [19] presented an 8T SRAM array with programmable word lines (WL). Here, the operands are grouped in rows, as same as that of the conventional SRAM storage pattern, and hence, no extra data transfer is needed. Three distinct computing modes were supported by this configuration. The ternary multiplication mode did not require any reference voltage production column. Discharge and charging routes were employed in the unsigned multiple bit multiplication mode for enlarging the voltage deviation of the least significant bit. Various logic operations can be completed in a single cycle when using the logic operation mode.

In order to address the reliability issues, Rajput et al. [20] implemented a local bit-line Shared pass-gate Dual-Port 8T (SDP8T) SRAM-basis IMC design for achieving cost-effective IMC processes. Furthermore, to accomplish accurate recognition for IMC on 4 operands concurrently in a single cycle, the Reference-based Reconfigurable SA is offered. To examine the implemented design, a 65-nm CMOS method is used to create a 4 KB SRAM array. In the Monte-Carlo simulations, the recommended IMC design is analyzed in the partial selection problem in a writing operation, the computation disturbance, and the computational failures impact on IMC. Rajput et al. [21] implemented an 8T SRAM cell-based arithmetic circuitry collaborative design and in-memory computing approach. In this implemented model, the 80 nm CMOS method is used for illustrating the Boolean logic process and arithmetic operations via an 8T SRAM unit. In order to prove that 8T SRAM cells are capable of performing in-memory computations and Boolean logic.

Si et al. [22] introduced an architecture for IMC with the goal of enhancing the energy efficacy of edge devices that perform multiple bit multiply-and-accumulate (MAC) processes. This structure used a 6T SRAM-based IMC for performing the following functions: (i) a simple 6T local calculating cell to do multiplication with a reduced sensibility for process discrepancy; (ii) weight-bitwise MAC operations were conducted to amplify sensing margin and enhance readout exactness, particularly for Precise MAC processes; (iii) a readout approach was optimized dynamically based on MAC awareness for improving energy efficiency (iv) a strategy to increase the signal margin by selecting a bitline (BL) header; and an upgraded sensing amplifier with a tiny offset margin to provide reliable read operations in contrary to process fluctuations. Mori et al. [23] presented a low-energy 64-Kb 8T dual-port image memory with a 28-nm fully depleted SOI (FD-SOI) process technology. This SRAM used a selective source line drive mechanism in conjunction with a data write mechanism to increase active energy efficiency at lower voltage.

This analysis shows that the SRAM plays a significant role in in-memory computing due to several key attributes. Yet, there are a lot of difficulties with using an SRAM-IMC because SRAM exhibits large power leakage and poor efficiency. The usage of IMC in the 6T SRAM causes serious reliability problems, including read disturbance and deterioration of read noise margins. To tackle this issue, 8T or 9T or 10T bit cells have been used in a few researches. However, they increased write and read stability at the expense of write latency and area overhead. Also, the susceptibility of the existing IMC design to noise and electromagnetic interference is high while reducing the supply voltage and scaling of the transistors because of the deprivation in Static Noise Margin (SNM). However, Schmitt Triggers can provide the optimal solution for enhancing the noise resistance of the circuit and stabilizing the output voltage levels. These points motivated us to propose a new 9T SRAM cell and use the same to develop ST logic based IMC for RCA-M.

3. Proposed Design

In this work, a new IMC design is proposed for performing several operations like basic Boolean operations, addition, and multiplication using a 1-kb 9T SRAM array. The proposed architecture is insisted with several blocks, namely SRAM cell, current compensation circuit, IMC control circuits, row decoder, column decoder, automated Write-Back Switching unit (AWBSU), and BIST to perform ripple carry addition and multiplications (RCA-M) as shown in Figure 1. Here, the HDFWA SRAM cell with 9 transistors is designed to optimize the read and write static noise margin individually. To diminish the leakage power and upsurge the reading performance, the current compensation based sensor circuit is utilized. Also, the control circuits of SRAM, RCA-M, IMC and AWBSU are designed using Schmitt Trigger (ST) logic for enhancing the noise resistance of the circuit and stabilizing the output voltage levels.

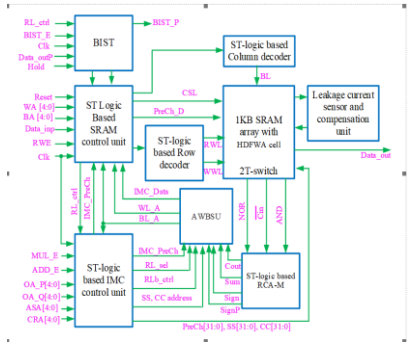


Fig.1. Proposed ST-logic based IMC design architecture

Table 1 describes the input and output signals of the suggested IMC. Also, Figure 2 shows the 2×2 IMC schematic. The proposed IMC structure utilizes a 2T Switch for detecting the values saved in 2 SRAM rows(B_L) by connecting the bit output of Q_{nn} and QB_{nn} ($n \in 0,1$) of each HDFWA SRAM cell. These bits are used to compute the operations $\overline{Cin_n}$, AND_n and NOR_n ($n \in 0,1$). After that, it does addition and multiplication using the RCA-M unit. Subsequently, it writes the computation outcome back to the appropriate SRAM using a multiplexer to complete the MAC processes in the memory.

Table 1. Input and output signals of the proposed IMC

Signal	Meaning
RWE	Read and write enable signal to enable for read/ write operation in the memory.
WA [4:0]	Word address for selecting the 5-bit WL character of memory.
BA [4:0]	Bit address for selecting the 5-bit BL of the memory.
OA_P [4:0]	Operand address for input P
OA_Q [4:0]	Operand address for input Q
ASA [4:0]	Arithmetic symbol number address line
CRA [4:0]	Computation result address line.
ADD-E	The addition process enables signal.
MUL-E	The multiplication process enables signal.
Data_inp:	Data input
Clk	Clock signal
RESET	Reset signal.
BIST_E	Built-in self-test enables signal.
Data_outp	Data output.
HOLD	Hold time mode activation signal.
BIST_P	BIST output

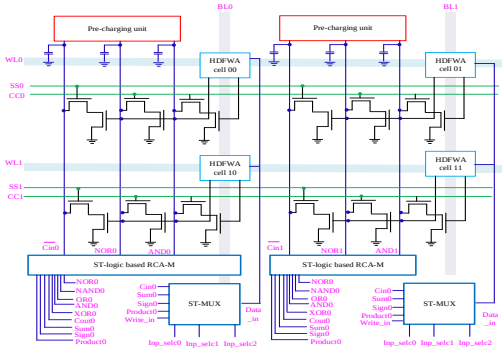


Fig.2 Schematic of 2×2 IMC for RCA-M

3.1 HDFWA SRAM:

In Figure 3, the design of an HDFWA SRAM cell is provided. Here, the inverters are constructed by combining the pull-up (P1/P2) and pull-down (N1/N2) transistors that keep the stored data. Then, an NMOS transistor (N3) is located in the middle of two inverters for cutting off the cross-coupled feedback information. This mechanism is used to improve the WSNM during write operation. This HDFWA bit cell also uses a read decoupling method to separate the storage nodes from BLs, preventing capacitive noise interference and the probability of read distresses. As a result, HDFWA bit cells have the maximum Read static noise margin (RSNMs). The entire column shares the columnar write enable (WE) which is utilized for activating and deactivating the N3. To boost the speed of operations, one NMOS transistor (N4) is utilized as the access transistor that performs read and write processes. The N5/N6 transistors are considered write/read access transistors, which divide the read and write paths to improve RSNM. The QB node activates the N7 transistor in the read path for the completion of the read process. Row-based signals WWL and RWL are used to regulate the N5 and N6 transistors, respectively. Column-based signals include the BL and column select line (CSL). The read and write latency can be decreased by using the suggested cell, which transfers the bit line voltage to the cell or the ground using a wide-access transistor. Table 2 reports the signals used for the various operations of the proposed cell.

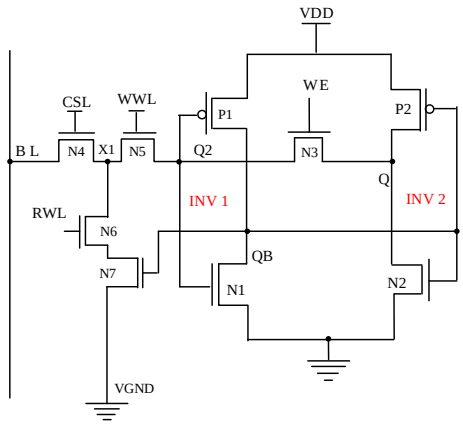


Fig. 3. Half-Select Disturb-Free Write Assist SRAM Cell

Table.2. Application of different signals for various processes

Signal	**WRITE '1'**	**WRITE '0'**	**READ**	**HOLD**
BL	V_{dd}	Ground	Pre-charged	V_{dd}
WWL	V_{dd}	V_{dd}	Ground	Ground
RWL	Ground	Ground	V_{dd}	Ground
CSL	V_{dd}	V_{dd}	V_{dd}	Ground
WE	Ground	Ground	V_{dd}	V_{dd}
V_{gnd}	V_{dd}	V_{dd}	Ground	V_{dd}

3.2 ST-logic based control circuits for HDFWA SRAM cell :

The HDFWA SRAM cell in Figure 3 uses WWL and RWL to read and write data. WL chooses a single row per instance. Nonetheless, this WL should be connected to an ST-logic based

DEMUX to allow the WWL and RWL signals in accordance with write or read process requirements. This ST-logic based DEMUX is constructed using two ST based AND gates (ST-AND) and one inverter (ST-INV). The control circuit of the HDFWA SRAM cell is shown in Figure 4. Here, ST logic is used as an effective solution for noise immunity enhancement.

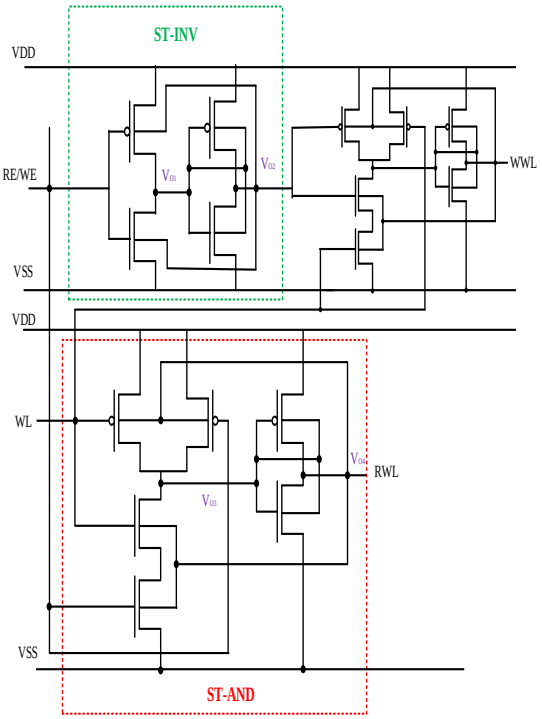


Figure 4 ST-logic based SRAM cell control circuit

The ST-INV design in Figure 4 illustrates how hysteresis is implemented by using a voltage feedback link from the outcome of the 2nd stage to the 1st-stage inverter's substrate. This ST-INV does not require additional current to ensure output stability, and hence it is well suited for low power design. Furthermore, it is possible to combine the buffer and inverter outputs (Vo1andVo2) into a single logic gate. The first-stage inverter's substrate bias is linked to Vo2 and is unaffected by its input node. As a result, the value of Vo2 changes is based on the threshold voltage of the 1st stage. This design is referred to as VTCMOS or variable threshold voltage CMOS. Every transistor in the first stage has a larger turn-on voltage than a typical DTMOS because it is configured to a zero-body bias state for the result conversion. The following formulas can be used to explain the switching threshold voltage:

$$V_{L-H} = \frac{V_{dd} - |V_{T,P}| + \varepsilon \times V_{TZ,N}}{\varepsilon + 1} \quad (1)$$

$$V_{H-L} = \frac{V_{dd} + \varepsilon \times V_{T,N} - |V_{TZ,P}|}{\varepsilon + 1} \quad (2)$$

where V_{TP} and V_{TN} denote the PMOS and NMOS transistor's threshold voltage during forward bias conditions. Likewise, $V_{TZ,N}$ and $V_{TZ,P}$ denotes the threshold voltage of NMOS and PMOS during zero-bias conditions. Also, , $\varepsilon = \frac{\gamma_N}{\gamma_P}$

which γ_N and γ_P refer to the NMOS and PMOS transistor's transconductance values. The concept of ST-INV can be extended to ST-AND gate design. The ST-AND gate needs fewer transistors to invert logic when compared with the conventional design. The suggested ST-AND logic gates demand reduced switching current since it utilizes the voltage feedback from V_{o4}.

The operating principle of these ST-AND gates resembles the operation of ST-INV design. However, the adjustment is required for the value of ε in equations (1) and (2) to ensure proper functioning. This scenario mimics the ST-INV gate if there is only one input transition required for output change. Nevertheless, if both input transitions lead to a modified output value, all four transistors (i.e., 2 PMOS and 2 NMOS) in the first stage are switched on and off simultaneously. As a result, a transistor with a dual channel width may be used to replace two parallel transistors, and the 2 sequential transistors can be treated as a single transistor with a dual channel length. Hence, the value of ε can be derived for an ST-AND gate with $\gamma_P' = 2\gamma_P$ and $\gamma_N' = 2\gamma_N$ as

$$\varepsilon' = \sqrt{\frac{\gamma_P'}{\gamma_N'}} = 2\varepsilon \quad (3)$$

3.3 Leakage Sensor and Compensation Unit

Leakage in SRAM architecture becomes a concern with the technology moving closer to the nanoscale. When threshold voltage and gate oxides decrease exponentially, leakage current will be increased. As a result, the SRAM's operating frequency and power consumption will be degraded. In this work, a new current compensation unit is proposed in order to lower the leakage and speed up operation. The new compensation design presented in this research comprises a compensation unit and a leakage current sensor, as shown in Figure 5. The voltage decrease initiated by the leakage current is detected using a leakage current sensor. If the leakage is discovered and verified, a compensation unit will accelerate the read process of SRAM. An SRAM cell with an integrated comparator makes up a leakage current sensor. An SRAM cell is utilized as a leakage observer, providing a voltage relative to the leakage current, V_{leak} for the comparator. The reason behind this is that the large leakage can reduce the operating frequency, power consumption, and status flipping of the SRAM cell. If V_{leak} is more than the reference voltage V_r, the comparator sends a warning signal to the compensating circuit.

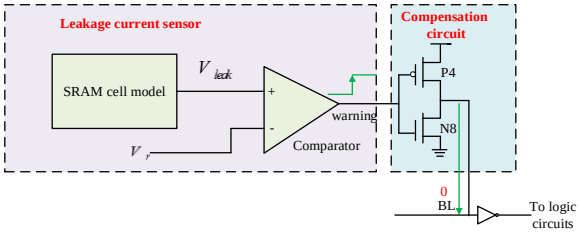


Fig. 5. Schematic diagrams of leakage current sensor

3.4 ST-logic based RCA-M

The proposed ST-logic based RCA-M is shown in Figure 6. It is made up of carry generation, sum generation, and ST-logic based control units. The source of every transistor in the carry generation unit is coupled to either supply voltage or disconnected from the ground. This results in a decrease in transistor count and power consumption while increasing computing capacity. Table 3 displays the RCA-M unit's logical functions. The input bits that need to be added are denoted as P and Q. The carry input and carry output are represented as Cin and Cout correspondingly. Also, the signed input is denoted as X and Y respectively. U and V are inputs for multiplication. In the RCA-M control circuit, the 2-input NOR is gate constructed using

conventional ST logic. In this case, the pull up network (PUN) links the input transistors in a way that creates a series combination of the inputs at the gate's output, whereas the pull down network (PDN) arranges the inputs in parallel in regard to inputs AND and NOR. The circuit's immunity to noise is provided by the feedback transistors.

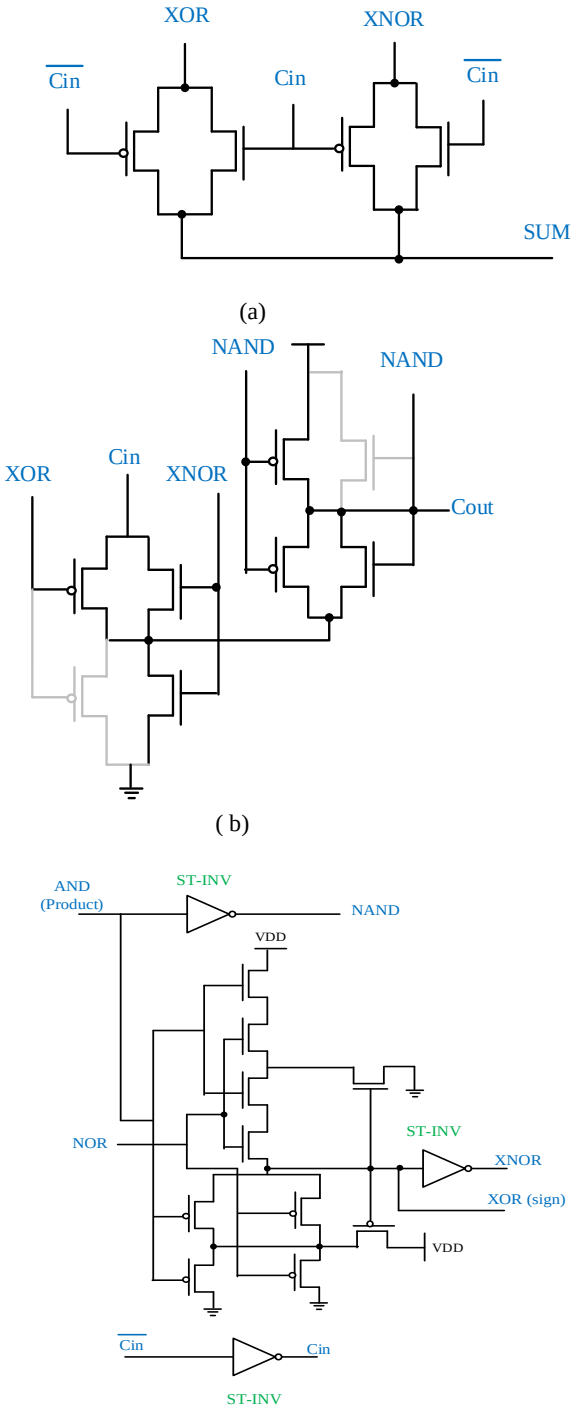


Fig.6. ST-logic based RCA-M (a) Sum generation unit (b) carry generation unit (c) Operation control circuit.

Table 3: Logical functions implemented by the RCA-M unit

Function	Description
AND	PQ
NAND	$\overline{PQ}$
NOR	$\overline{P+Q}$
Product	UV
XOR	$P \oplus Q$
Sum	$(P \oplus Q) \oplus Cin_n$
Cout	$(P \oplus Q) \cdot Cin_x + P \cdot Q$
SignP	$X \oplus Y$

The logic transitions that occur during four-bit addition are displayed in Figure 7 (a). The Pre-charging signal(PreCh)) signal tells the pre-charge circuit to set Cin_0 to low and $\overline{Cin_0}$, NOR_0 and AND_0 to high during the start-up process. The proposed IMC loads W_{L0} with data. Inp_selc[1:0] triggers ST-MUX for writing 0 as its input, and thereby, it raises Q_{00} to high. Also, a carry bit 0 has to be kept in Q_{20} to perform addition. Subsequently, $\tilde{C}_0$, $\tilde{C}_1$ and $\tilde{S}_2$ are turned on simultaneously to start computations. The high values of Q_{00} and Q_{10} perform NOR operation to output NOR_0 as zero. Also, AND operation is performed on Q_{00} and Q_{10} that stores high and low values, respectively. As a result, $NAND_0$ is turned to low. NOT is done where there is Q_{20} is low, $\overline{Cin_0}$ is high, and Cin_0 is low. This completes the Bit 0 addition by attaining high and low values for Sum_0 and $Cout_0$, respectively. After that, ST-MUX chooses WL_3 and $Inp\_selc[1:0]$ of [1] trigger to store Sum_0 into Q_{30}. Then, ST-MUX chooses WL_2 and $Inp\_selc[1:0]$ enables ST-MUX to designate the carry bit in $Cout_0$ and store it in Q_{21}. The $PreCh$ signal increases $\overline{Cin_1}$, NOR_1, and AND_1 to high quickly for enabling the computation of the subsequent bit.

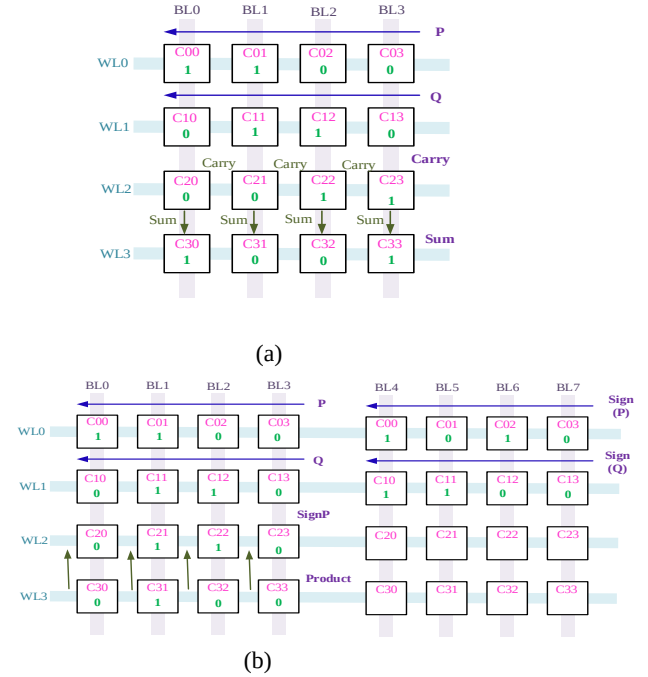


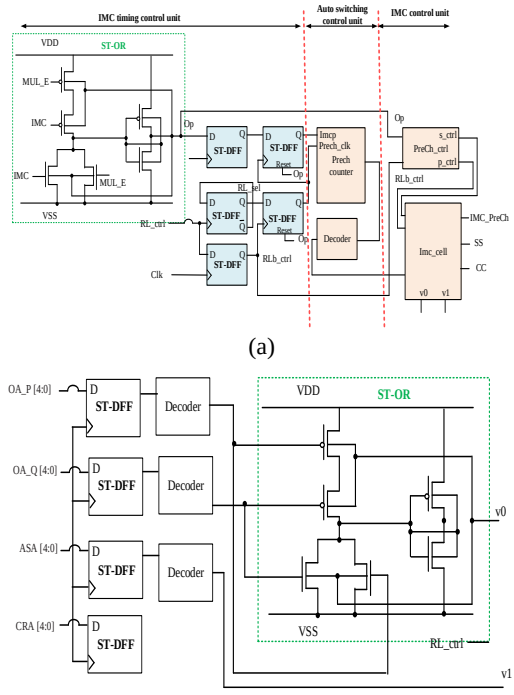
Fig.7. Data flow for (a) addition and (b) Multiplication

The multiplication operation resembles the addition operation but with the following distinct feature: it retains the sign-product (SignP) bit in a cell Q_{20} while the product bit is saved in Q_{30}. It selects WL_2 once the product bit has been identified and it is then saved into Q_{30}. Subsequently, it chooses WL_3 to store the SignP bit into Q_{20} as shown in Figure 7 (b).

3.5 ST-logic based IMC control unit

The IMC control unit is used for controlling the pre-charging stage of the particular row address, selecting the data saved in the designated row and controlling the 2T control for computation. It contains an IMC controlling component, IMC timing controller, address choosing controller, and Automatic pre-charge controller, as shown in Figure 8. The IMC timing controller decides whether to execute computations and produce timing control signals. This unit begins to generate the required computation control signal via its output signal $Oper\_Cntrl$ when one of the IMC or MUL-E signals is set high.

When the IMC signal is activated, the memory will add the given process address from BL_j ($j = 0,1,2,3$) in sequence till the IMC signal is switched off. In contrast, if the MUL-E signal is activated, the memory performs the multiplication on the designated process address from BL_j ($j = 0,1,2,3$) till the MUL-E signal is deactivated. Simultaneously, the inverted latch is triggered with Clk, and the read latch clock controlling signal(RL_ctrl)is divided by the IMC timing control circuit. It used ST-logic based D-flip flop (ST-DFF) to yield control signals RL_sel, RLb_ctrl by generating two distinct operating clocks based on operation requisite.



(a)

(b)

Figure 8 IMC control circuit (a) Timing control, auto-switching, and IMC control (b) Address selecting control unit

SRAM is susceptible to single-event upset (SEUs), which can cause temporary mistakes and interfere with regular operations. Because of this, flip-flops with strong single-event performance are crucial to ensure reliable and noise immune IMC control circuits. The SE performance can be improved by extending the feedback-loop delay of the D-FF. Schmitt-trigger gates can be used as a substitute for inverters in a standard D-flip flop (D-FF) architecture in order to increase the feedback-loop latency. Figure 9 illustrates a D-FF design in which the first Schmitt-trigger gate is used in place of the traditional D-FF latch inverters.

In this ST-DFF design, the transistors MP1 and MN1 won't get any current when in write mode. This structure limits the delay cost caused by the off-centred threshold voltage and extricates the Schmitt-trigger's direct route in write mode. It is possible to raise the critical charge during the hold operation by drawing up the sources of MN1 and MN2 and bringing down the sources of MP1 and MP2. Furthermore, the MN4 or MP4 transistor will turn on while the positive or negative transient is formed on node H2. In order to prevent a possible upset, this will draw down the CLK signal or pull up the CLKb signal by switching off the transistors. It is established that this architecture has three time enhancements in SE when compared with traditional DFF design.

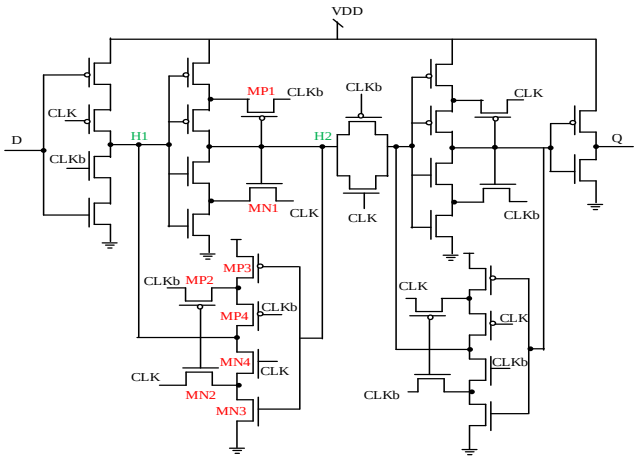


Figure 9 Schematic of ST-DFF

The auto-switching pre-charge control circuit depicted in Figure 8 produces the designated row addresses for sequential precharging in the memory. Here, RL_sel initiates the precharging switched clock signal ($PreCh\_clk$) via the ST-DFF. The pre-charge counter increases by one after every clock cycle. The timing control signals for pre-charge $PreCh\_bit[31:0]$ are produced by the 5-to-32 Decoder. The mathematic operation addresses in the memory, such as OA_P [4:0], OA_Q [4:0], ASA [4:0], and CRA [4:0], are specified by the address choosing control circuit. These addresses are read by the ST-DFF, three 5-to-32 multiplexers, and the OR gate. The IMC control unit outputs the address of the arithmetic memory. The IMC control unit employs the pre-charge control $PreCh\_ctrl$ for generating control signals s_ctrl, and p_ctrl interchangeably. Hence, the operation is executed once the pre-charge is finished. In order to operate the 2T Switch, the IMC control unit outputs IMC_PreCh signals, and the computation unit controls the lines s_ctrl, and p_ctrl.

3.6 ST-logic based AWBSU

The proposed ST-logic based AWBSU consists of an automated B_L switching unit, data switching unit, and W_L switching unit, as in Figure 10. This AWBSU writes the sum or product to an allotted address beginning from B_{L0} to B_{L3} while O_p is raised to high. If the IMC_PreCh is raised to high, the automated B_L switching unit uses an inverted RL_sel for switching the clock signal (B_L_clk). The ST-DFF triggers this clock signal, and the original memory bitline address is replaced by the output $B_L\_A[4:0]$ to perform the automatic memory B_L switching operation. The data selection is under the control of the data switching unit. It proposed two addition or two multiplication outputs because of its single-ended output. This is accomplished using four SC-logic based 32-to-1 multiplexers that are managed by B_L_A . The data selection signal is utilized for switching the operation addresses of addition and multiplication, sum and carry, product and sign product. Lastly, ADD-E and MUL-E are utilized to select data.

The output of the data switching unit produces (product, signP) pair for multiplication and (carry, sum) pair for addition. If the operation start signal (O_p) of the automated B_L switching unit is raised too high, ST-DFF will latch RLb_ctrl generated through the IMC timing control unit for giving $IMC\_datasel$ as output. This $IMC\_datasel$ is utilized for controlling 2-to-1 ST-MUX. These multiplexers allow the product and sign product addresses ($PA, SignPA$) to be switched in turn with the carry and sum addresses (CA, SA). The memory address of the W_L is switched automatically using the output $W_L\_A[4:0]$.

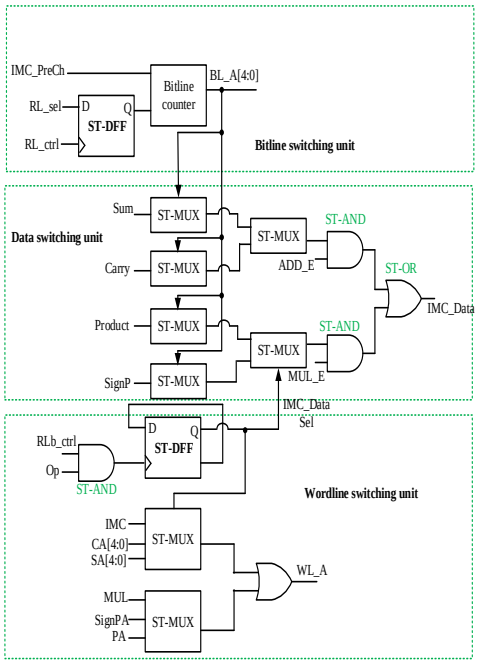


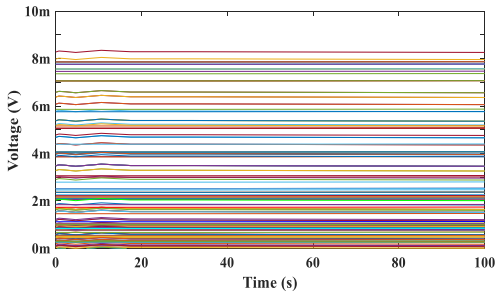
Figure 10 Schematic of ST-logic based AWBSU

3.7 Built-in Self-Test Circuit

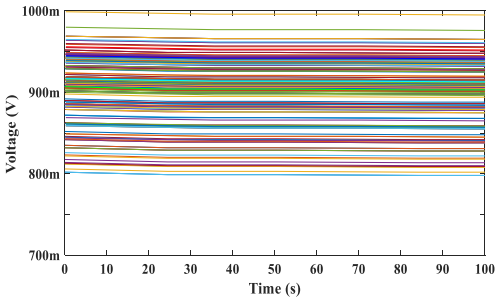
It operates in two modes: normal and Hold. The BIST controller performs a write or read operation after receiving the testing data. In normal mode, the BIST controller used a pattern generator for generating alternating switching data 0 and 1. In order to ascertain the accurate functioning of the memory's read and write, it simultaneously matches the memory with the aforementioned output response data. The self-test outcome BIST_P is turned ON while the outcome result agrees with the records produced by the BIST controller. If not, the read function of the memory is not working. To enable long-term memory reading and test distortion level of the saved data after extended reading, hold mode utilizes the BIST controller for latching the self-test data (BIST_Data) and resetting the self-test read-write signal (BIST_WR) to a lower level. BIST_P is raised to high if modification is not required on the saved data.

4.Results and discussion

This study implements the suggested IMC design using 14-nm CMOS technology. Monte Carlo simulations were run at node Q of the SRAM cell for the purposes of reading data 0 and data 1, as in Figure 11. Here, the voltage drifted between 0 to 8mV while reading data 0. Also, the voltage drifted between 800 to 1000 mV while reading data 1. The results of the Monte Carlo simulations demonstrate that the suggested IMC may successfully stop the variations in the circuit characteristics.



(a)



(b)

Fig.11. Monte Carlo simulation results for (a) Reading 0 (b) Reading 1

The SNM of the proposed cell in IMC is shown in Figure 12. To improve the RSNM, a decoupled-read approach is employed in the proposed cell design. This approach uses a read buffer made of N6 and N7 transistors to isolate the data storage nodes from the BL. The proposed cell shows a high SNM of 176 mV at which the SRAM cell will continue to function without any stored or output bit disruptions. In Figure 12, the low RSNM (approximately 140 mv) of a standard 6T-SRAM cell is caused by the read path's connection to the data storage nodes.

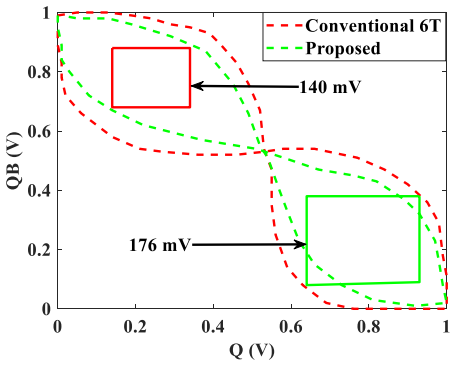
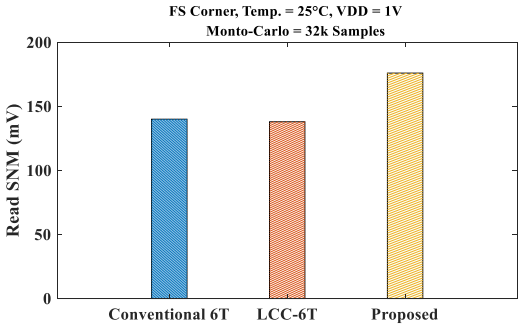
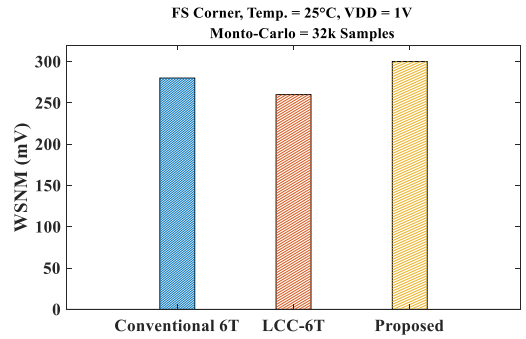


Fig.12. Read SNM analysis

Figure 13 compares the simulated write SNM and Read SNM of the proposed cell with a conventional SRAM macro and LCC-6T. According to the findings of the 300 Monte Carlo simulation displayed in Figure 13(a), the worst case RSNM in the IMC macro employing LCC-6T is less than that of a standard SRAM macro. Furthermore, the worst-case write margin of the IMC macro with LCC-6T is 1.28% smaller than a standard SRAM macro. The threshold voltage change between the transistors in the LCC causes this loss in SNM. The proposed model with 9T cells shows the best RSNM and WSNM. The proposed cell uses a feedback-cutting method in the writing mode. This method enhances the WSNM by causing the data storage node to be impacted by the bitline as in Figure 13 (b). Also, the stability of the proposed macro cell is improved due to the use of Schmitt logic in the peripheral designs.



(a)



(b)

Fig.13. Comparative analysis of SNM (a) Read SNM (b) Write SNM

Consider two inputs *P* (1100) and Q (0110) for validating the addition process of the proposed IMC. At first, the carry bit at cell 20 is low, as seen in Figure 14. Then, the values in cell 00, cell 10, and cell 20 summed up to write the resultant sum value in cell 20 along BL_0. In the meantime, it writes the carry in cell 21 of the subsequent line (BL_1). This operation is repeated till the sum is projected.

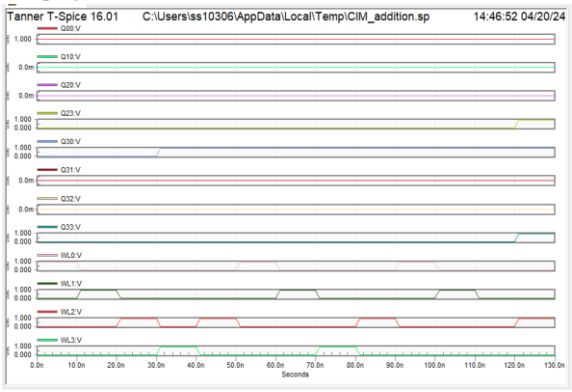


Figure 14 Simulated waveform for addition operation

Similarly, the ability of the proposed IMC for four-bit multiplication is demonstrated by performing the multiplication on a sample word. Here, the positive and negative symbols of any input are represented as 0 and 1, respectively. The multiplicand and multiplier bits are represented as $P = [(-1)\ \ (+1)\ (+0)\ (+0)]$ and $Q = [(+0)\ \ (-1)\ (+1)\ (+0)]$ respectively as shown in Figure 15. Here, W_{L3} reflects the product of the bits on W_{L0} and W_{L1} as well as W_{L2} reflects its sign.

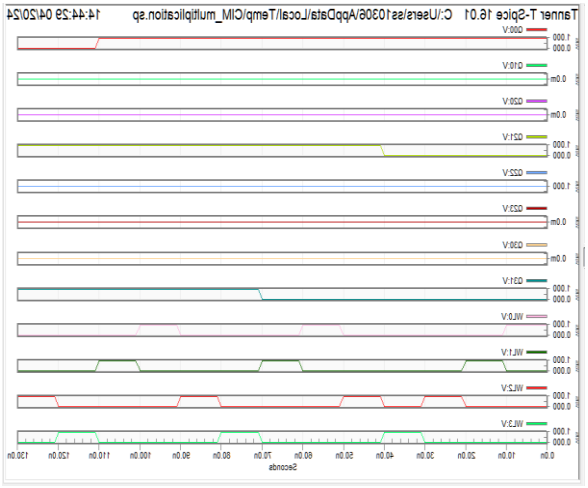


Figure 15 Simulated waveform for multiplication

The Schmitt trigger logic in the suggested IMC design has noise immunity, which can be evaluated and validated using process variability analysis. Process variability analysis examines how integrated circuit performance is impacted by manufacturing variances, such as differences in transistor sizes, threshold voltages, and parasitic capacitances. Monte Carlo simulations are developed for evaluating the statistical distribution of circuit performance indicators, including noise margins and switching thresholds by taking into account the process variations. In order to assess the proposed IMC's dependability under process variation, the parameters are varied up to 10% from their starting values using a Gaussian distribution with a 3% standard deviation. The global Monte Carlo simulations for Reading 1 are shown in Figure 16. According to this statistical graph, the standard

deviation (σ) is 2.16 V, and the average read value (μ) is 1 V, which is consistent with the intended operating voltage. Hence, Read data 1 is in line with the normal distribution. The reason behind this is that the proposed model uses Schmitt Trigger Logic Gates, D-FF, and multiplexers in the control circuitry of the proposed IMC design. This significantly improves noise immunity with reduced switching power consumption compared to CMOS Schmitt triggers but with a slight increase in latency.

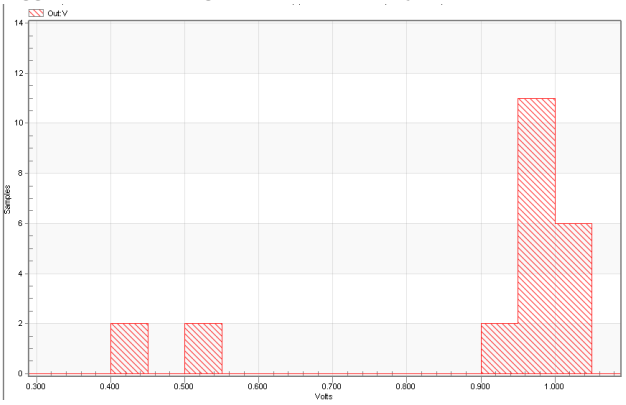


Figure 16 Impact of process variations analysis using Monte Carlo for Read 1

The performance comparison between the suggested IMC and a number of current IMC architectures is displayed in Table 4. The typical energy consumption of the suggested IMC is lesser because it makes use of noise-immune control circuits. As a result, it may result in energy savings by lowering the requirement for additional signal processing. The suggested IMC performs better in FOM than the previous works, although having the highest amount of executed operations (NAND, NOR, XOR, addition, and multiplication). The figure of merit (FoM) represents the energy per bit, and it can be computed using the following expression:

$$FOM = \frac{E_{avg}/bit}{V_{DD}} \quad (4)$$

Where E_{avg} represents the average energy.

Table 4 Comparative analysis with other reported works

Parameters	Si et al. [22]	Mori et al. [23]	Zhang et al. [19]	Proposed
Process (nm)	28	28	28	14
Supply voltage (V)	0.9	0.7	0.9	0.9
Cell type	LCC-6T	8T	8T	9T
Operation	MAC	Image recognition	Multiplication	Addition and multiplication
Array size	64kb	64kb	1kb	1kb
Power (μW)	-	9.16	-	0.3738
Average energy (fJ)	11.54	389.6	11.375	4.019
FOM (fJ/bit/MHz)	-	0.08	0.9	4.019

5.Conclusion

This work presents a new IMC for RCA-M operation using an HDFWA SRAM cell array. Also, the noise immunity of the proposed architecture has been reduced using ST logic. The proposed cell reduced read and write delays by using a wide access transistor that shifts the bitline voltage to the ground or within the cell. Also, the proposed IMC contains a compensation-based sensor circuit to reduce leakage current and speed up the reading process. The accumulating problems of existing designs and the von Neumann bottleneck are resolved by the suggested IMC. The viability of the suggested scheme was proved by the simulated 14nm IMC macro with HDFWA SRAM cell and ST-logic based control circuits, yielding an energy of 4.019fJ. In the future, this approach can significantly improve the feasibility of using SRAM-IMC for AI edge computing.

References

[1] Mittal, Sparsh, Gaurav Verma, Brajesh Kaushik, and Farooq A. Khanday. "A survey of SRAM-based in-memory computing techniques and applications." *Journal of Systems Architecture* 119 (2021): 102276.

[2] Gupta, Aman Kumar, and Abhishek Acharya. "Exploration of 9T SRAM cell for in memory computing application." In *2021 Devices for Integrated Circuit (DevIC)*, pp. 461-465. IEEE, 2021.

[3] Jhang, Chuan-Jia, Cheng-Xin Xue, Je-Min Hung, Fu-Chun Chang, and Meng-Fan Chang. "Challenges and trends of SRAM-based computing-in-memory for AI edge devices." *IEEE Transactions on Circuits and Systems I: Regular Papers* 68, no. 5 (2021): 1773-1786.

[4] Huo, Qiang, Yiming Yang, Yiming Wang, Dengyun Lei, Xiangqu Fu, Qirui Ren, Xiaoxin Xu et al. "A computing-in-memory macro based on three-dimensional resistive random-access memory." *Nature Electronics* 5, no. 7 (2022): 469-477.

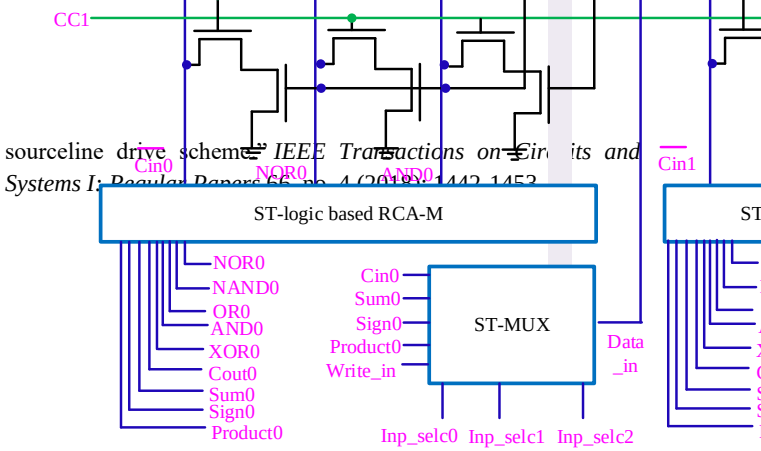
[5] Xie, Shanshan, Can Ni, Aseem Sayal, Pulkit Jain, Fatih Hamzaoglu, and Jaydeep P. Kulkarni. "eDRAM-CIM: Reconfigurable Charge Domain Compute-In-Memory Design With Embedded Dynamic Random Access Memory Array Realizing Adaptive Data Converters." *IEEE Journal of Solid-State Circuits* (2023).

[6] Jing, Zhaokun, Bonan Yan, Yuchao Yang, and Ru Huang. "VSDCA: A voltage sensing differential column architecture based on 1T2R RRAM array for computing-in-memory accelerators." *IEEE Transactions on Circuits and Systems I: Regular Papers* 69, no. 10 (2022): 4028-4041.

[7] Pham, Thi-Nhan, Quang-Kien Trinh, Ik-Joon Chang, and Massimo Alioto. "STT-BNN: A novel STT-MRAM in-memory computing macro for binary neural networks." *IEEE Journal on Emerging and Selected Topics in Circuits and Systems* 12, no. 2 (2022): 569-579.

[8] Lin, Zhiting, Zhongzhen Tong, Jin Zhang, Fangming Wang, Tian Xu, Yue Zhao, Xiulong Wu et al. "A review on SRAM-based computing in-memory: Circuits, functions, and applications." *Journal of Semiconductors* 43, no. 3 (2022): 031401.

[9] Wang, Chua-Chin, and Chien-Ping Kuo. "67.5-fJ per access 1-kb SRAM using 40-nm logic CMOS process." In *2021 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1-4. IEEE, 2021.



[10] K. Agawa, H. Hara, T. Takayanagi, and T. Kuroda, "A bitline leakage compensation scheme for low-voltage SRAMs," *IEEE J. Solid-State Circuits*, vol. 36, no. 5, pp. 726–734, May 2001.

[11] D. Kim, G. Chen, M. Fojtik, M. Seok, D. Blaauw, and D. Sylvester, "A 1.85fW/bit ultra low leakage 10T SRAM with speed compensation scheme," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, Rio de Janeiro, Brazil, May 2011, pp. 69–72.

[12] A.-T. Do, Z.-H. Kong, K.-S. Yeo, and J. Y. S. Low, "Design and sensitivity analysis of a new current-mode sense amplifier for low-power SRAM," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 19, no. 2, pp. 196–204, Feb. 2011.

[13] M. S. M. Siddiqui, Z. C. Lee, and T. T.-H. Kim, "A 16-kb 9T ultralow-voltage SRAM with column-based split cell-VSS, data-aware write-assist, and enhanced read sensing margin in 28-nm FDSOI," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 29, no. 10, pp. 1707–1719, Oct. 2021.

[14] Dhakad, Narendra Singh, Eshika Chittora, Gopal Raut, Vishal Sharma, and Santosh Kumar Vishvakarma. "In-Memory Computing with 6T SRAM for Multi-operator Logic Design." *Circuits, Systems, and Signal Processing* (2023): 1-15.

[15] Su, Jian-Wei, Xin Si, Yen-Chi Chou, Ting-Wei Chang, Wei-Hsing Huang, Yung-Ning Tu, Ruhui Liu et al. "Two-way transpose multi-bit 6T SRAM computing-in-memory macro for inference-training AI edge chips." *IEEE Journal of Solid-State Circuits* 57, no. 2 (2021): 609-624.

[16] Song, Jiahao, Yuan Wang, Xiyuan Tang, Runsheng Wang, and Ru Huang. "A 16Kb transpose 6T SRAM in-memory-computing macro based on robust charge-domain computing." In *2021 IEEE Asian Solid-State Circuits Conference (A-SSCC)*, pp. 1-3. IEEE, 2021.

[17] Ammoura, Lila, Marie-Lise Flottes, Patrick Girard, and Arnaud Virazel. "Preliminary Defect Analysis of 8T SRAM Cells for In-Memory Computing Architectures." In *2021 16th International Conference on Design & Technology of Integrated Systems in Nanoscale Era (DTIS)*, pp. 1-4. IEEE, 2021.

[18] Saragada, Prasanna Kumar, Subashish Manna, Amandeep Singh, and Bishnu Prasad Das. "A Configurable 10 T SRAM-Based IMC Accelerator With Scaled-Voltage-Based Pulse Count Modulation for MAC and High-Throughput XAC." *IEEE Transactions on Nanotechnology* (2023).

[19] Zhang, Jin, Zhiting Lin, Xiulong Wu, Chunyu Peng, Wenjuan Lu, Qiang Zhao, and Junning Chen. "An 8T SRAM array with configurable word lines for in-memory computing operation." *Electronics* 10, no. 3 (2021): 300.

[20] Rajput, Anil Kumar, Manisha Pattanaik, and Gaurav Kaushal. "Local bit-line shared pass-gate 8T SRAM based energy efficient and reliable In-Memory Computing architecture." *Microelectronics Journal* 129 (2022): 105569.

[21] Rajput, Anil Kumar, and Manisha Pattanaik. "Implementation of Boolean and arithmetic functions with 8T SRAM cell for in-memory computation." In *2020 International Conference for Emerging Technology (INCET)*, pp. 1-5. IEEE, 2020.

[22] Si, Xin, Yung-Ning Tu, Wei-Hsing Huang, Jian-Wei Su, Pei-Jung Lu, Jing-Hong Wang, Ta-Wei Liu et al. "A local computing cell and 6T SRAM-based computing-in-memory macro with 8-b MAC operation for edge AI chips." *IEEE Journal of Solid-State Circuits* 56, no. 9 (2021): 2817-2831.

[23] Mori, Haruki, Tomoki Nakagawa, Yuki Kitahara, Yuta Kawamoto, Kenta Takagi, Shusuke Yoshimoto, Shintaro Izumi, Hiroshi Kawaguchi, and Masahiko Yoshimoto. "A 28-nm FD-SOI 8T dual-port SRAM for low-energy image processor with selective

sourceline drive scheme." *IEEE Transactions on Circuits and Systems I: Regular Papers* 66, no. 4 (2019): 1442-1453.