

In Memory Computing using ReRAM Devices

*submitted in partial fulfillment of the requirements
for the degree of*

MASTER OF TECHNOLOGY

in

Electrical Engineering

by

AYUSH YADAV EE23M114

Supervisor(s)

Dr Vikram Kumar Pudi



**DEPARTMENT OF Electrical Engineering
INDIAN INSTITUTE OF TECHNOLOGY TIRUPATI**

24 November 2024

DECLARATION

I declare that I have written this work in my own words and have properly credited and referenced the underlying sources. I declare that I followed academic integrity and honesty principles and did not alter, fabricate, or misrepresent any ideas, data, facts, or sources in my work. Violations of the above may result in disciplinary action by the Institute, as well as consequences from sources that have not been properly referenced or permission has not been obtained.

Place: Tirupati
Date: 24-11-2024

Signature
Ayush Yadav
EE23M114

BONA FIDE CERTIFICATE

This is to certify that the report titled "**In-Memory Computing Using ReRAM Devices**" , submitted by **Ayush Yadav** , to the Indian Institute of Technology, Tirupati, for the granting of the degree in **Master of Technology**, is a bona fide record of the project work done by him under my supervision. The contents of this report, in its entirety or in part, has not been submitted to any other institute or university for degree or diploma award.

Place: Tirupati
Date: 24-11-2024

Dr Vikram Kumar Pudi
Guide
Assistant Professor
Department of Electrical
Engineering
IIT Tirupati - 517501

ACKNOWLEDGMENTS

I am grateful to Dr. Vikramkumar Pudi, my project guide, for providing invaluable guidance, support, and encouragement throughout this project. Although I have only just begun, his expertise and insights have already helped shape the initial direction of this work.

Furthermore, I'd like to thank my family and friends for their constant encouragement and compassion as I set out on this journey. I am humbled to IIT Tirupati for providing an excellent academic environment and resources to support my research.

ABSTRACT

KEYWORDS: Resistive random-access memory (ReRAM), 1Transistor-1Resistor (1T-1R), Logic gates, Adders, Majority logic, In-memory computing, Sense amplifier, Von Neumann bottleneck, Memristor

In-memory computing is a novel method to computer design that addresses the slow data transfer between memory and processors in traditional systems. Instead of transporting data back and forth, in-memory computing processes data right where it is stored, making everything faster and more energy efficient. Resistive Random-Access Memory (ReRAM) is a crucial technology in this regard. ReRAM stores data even when the power is turned off, and it performs faster and consumes less energy than standard memory types. This work investigates how ReRAM can be utilised for in-memory computing, performing operations such as the Hadamard transform within the memory itself. By doing so, we hope to demonstrate how ReRAM may significantly increase the speed and efficiency of computing processes. This could be the future of computing.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS	i
ABSTRACT	ii
LIST OF FIGURES	viii
LIST OF TABLES	ix
ABBREVIATIONS	x
1 Introduction	1
1.1 Objectives and Scope	1
1.2 Relevant Figures	2
2 Literature Survey	3
2.1 Research Gap/Motivation:-	3
2.2 Literature Review:	3
2.3 Problem Statement	5
2.3.1 Recent advancements in automating complex computations	5
2.3.2 Existing 2-D transforms rely on power-intensive CPU-GPU systems.	6
2.4 Hadamard Matrix	6
2.4.1 Equation	6
2.4.2 Performing the Hadamard Transform	7
2.4.3 Applications	7
2.4.4 Properties	7
2.5 Conclusion	7
3 Methodology	9
3.1 ReRAM Basics	9
3.2 Design	10

3.3	Stanford-PKU (Peking University) RRAM Model	10
3.3.1	Overview of the Model	10
3.3.2	Physical Structure of RRAM Cells	10
3.3.3	Key Insights	11
3.3.4	Applications in ReRAM-Based Computing	11
3.4	Summarized Phenomenon: Bipolar Switching in ReRAM	12
3.4.1	SET Process	12
3.4.2	RESET Process	13
3.4.3	Key Insights:	14
3.5	1T1R Cell	15
3.5.1	Testing of Writing and Reading data from ReRAM cells. . .	15
3.5.2	ReRAM Operation Table for Write-1	16
3.5.3	Testing of Writing and Reading data from ReRAM cells. . .	16
3.5.4	ReRAM Operation Table for Write-0	16
3.5.5	Overall Table for Different Operations	17
3.6	Testing of Majority Logic using Sense Amplifier from ReRAM cells.	17
3.7	Binary Addition Flowchart and Details	18
3.7.1	Testing of Binary Arithmetic using Sense Amplifier from ReRAM cells.	19
3.8	Circuit Overview	19
3.9	Flow-Chart	20
3.9.1	Cycle-by-Cycle Operations	20
3.9.2	Counter	21
3.9.3	Demultiplexer	21
3.9.4	Triple Row Decoder	21
3.9.5	ReRAM Cells	22
3.10	Sense Amplifiers	22
3.10.1	Components of Sense Amplifier	22
3.10.2	Key Concepts in Time-Based Sensing	24
3.10.3	Role of W/L Ratios in CMOS Design	25
3.10.4	Timing Characteristics	25
3.10.5	Operation Summary (Cycle by Cycle)	25
3.11	Circuit	25

4 Results and Discussions	26
4.1 Results	26
4.1.1 Circuit Diagram for Test 0	26
4.1.2 Waveform for Test 0	27
4.1.3 Results Table for Test 0	27
4.1.4 Circuit Diagram for Test 1	28
4.1.5 Waveform for Test 1	28
4.1.6 Results Table for Test 1	28
4.1.7 Circuit Diagram for Test 2	29
4.1.8 Waveform for Test 2	29
4.1.9 Results Table for Test 2	30
4.1.10 Circuit Diagram for Test 3	30
4.1.11 Waveform for Test 3	31
4.1.12 Results Table for Test 3	31
4.1.13 Circuit Diagram for Test 4	32
4.1.14 Waveform for Test 4	32
4.1.15 Results Table for Test 4	33
4.1.16 Circuit Diagram for Test 5	33
4.1.17 Waveform for Test 5	33
4.1.18 Results Table for Test 5	34
4.2 Problems	34
4.2.1 Problem 1: Synchronization issues in <i>cob</i> writing:	34
4.2.2 Problem 2: Lack of proper delay in circuit operations:	34
4.3 Conclusion	34
4.3.1 Conclusion 1: Automated full adder design using ReRAM cells is feasible and effective for in-memory computation.	34
4.3.2 Conclusion 2: Synchronization between WRITE and READ operations is critical:	34
4.3.3 Conclusion 3: Counter-driven automation reduces the need for manual cycle management:	35
4.3.4 Conclusion 4: Improvements needed for real-time scalability:	35
5 Future Scope	36
5.1 Future Work	36

5.1.1	Scalability:	36
5.1.2	Integration:	36
5.1.3	Applications:	36

LIST OF FIGURES

1.1	Diagram of Von-Neumann bottleneck	2
1.2	Von-Neumann architectural schemes in contrast to in-memory computing	2
2.1	Various Applications of ReRAM	8
3.1	Structure of 1T1R Cell	9
3.2	1T-1R Cell	9
3.3	Layer structure of the Stanford-PKU RRAM Model (Not to Scale)	11
3.4	Set Operation	13
3.5	ReSet Operation	14
3.6	1T1R Cell	15
3.7	Symbol	15
3.8	Write 1	15
3.9	Write 0	16
3.10	Flow Chart for Binary Addition[7]	18
3.11	Testing of Binary Arithmetic	19
3.12	Flow-Chart for the Circuit	20
3.13	Sense Amplifier Schematic Diagram	23
3.14	Example Circuit Overview	25
4.1	Circuit for Test 0: Testing of correctly turning ON the respective word lines.	26
4.2	Waveform for Test 0: Testing of correctly turning ON the respective word lines.	27
4.3	Circuit for Test 1: Testing of correctly sensing a,b,cin, and later sensing cob,a,b.	28
4.4	Waveform for Test 1: Testing of correctly sensing a,b,cin, and later sensing cob,a,b.	28
4.5	Circuit for Test 2: Testing of correctly sensing a,b,cin, and writing 1.3V in cob.	29

4.6	Waveform for Test 2: Testing of correctly sensing a,b,cin, and writing 1.3V in cob.	29
4.7	Circuit for Test 3: Testing of sensing a,b,cin, writing 1.3V in cob and again sensing cob,a,b.	30
4.8	Waveform for Test 3: Testing of sensing a,b,cin, writing 1.3V in cob and again sensing cob,a,b.	31
4.9	Circuit for Test 4: Testing of writing 1.3V in cob with 2nd Clock Pulse.	32
4.10	Waveform for Test 4: Testing of writing 1.3V in cob with 2nd Clock Pulse.	32
4.11	Circuit for Test 5: Testing of writing 1.3V in cob with 2nd Clock Pulse with a delay element.	33
4.12	Waveform for Test 5: Testing of writing 1.3V in cob with 2nd Clock Pulse with a delay element.	33

LIST OF TABLES

3.1	ReRAM Set and Reset Processes: Voltage, Current, and Resistance State Transitions	16
3.2	ReRAM Reset Operation and Resistance State Transitions	16
3.3	Table of Word Line, Bit Line Voltages, and Resistance States for Different Operations	17
3.4	Truth Table for a 3-Input Majority Gate	18
3.5	Cycle-by-cycle operations for the full adder design using ReRAM cells.	21

ABBREVIATIONS

ReRAM	Resistive Random Access memory
CPU	central processing unit
DRAM	Dynamic Random Access memory
LRS	Low-resistance State
HRS	High-resistance State
1T-1R	1 Transistor-1 Resistor
STT-MRAM	Spin-Transfer Torque Magneto resistive Random-Access Memory
IITT	Indian Institute of Technology, Tirupati

CHAPTER 1

Introduction

By addressing the inefficiencies of the conventional von Neumann model, which separates memory and processing units, in-memory computing is transforming computer design. The frequent data transfers between memory and the CPU caused by this separation, sometimes known as the "memory wall," result in considerable delays and excessive energy consumption. By integrating computational operations directly within the memory units, in-memory computing seeks to address these issues and greatly improves processing speed as well as energy efficiency.

Resistive Random-Access Memory (ReRAM) is a game-changing technology that propels computing forward. ReRAM is a type of non-volatile memory that stores information by changing the resistance of a substance, usually a metal oxide. ReRAM outperforms existing memory technologies such as DRAM and Flash, with faster switching rates, lower energy usage, and increased endurance. Furthermore, ReRAM preserves data even when power is turned off, making it extremely reliable. ReRAM's distinct properties make it a great choice for in-memory computing, allowing for more efficient data processing while reducing the performance bottlenecks associated with traditional memory architectures.

1.1 Objectives and Scope

Applying ReRAM-Based In-Memory Computing to the Hadamard Transform Implementation: The main goal is to create a fast way to use ReRAM devices to perform the Hadamard transform. This entails using ReRAM's in-memory computing capabilities to carry out the necessary matrix operations right inside the memory, saving on data transfer costs and enhancing computational effectiveness.

1.2 Relevant Figures

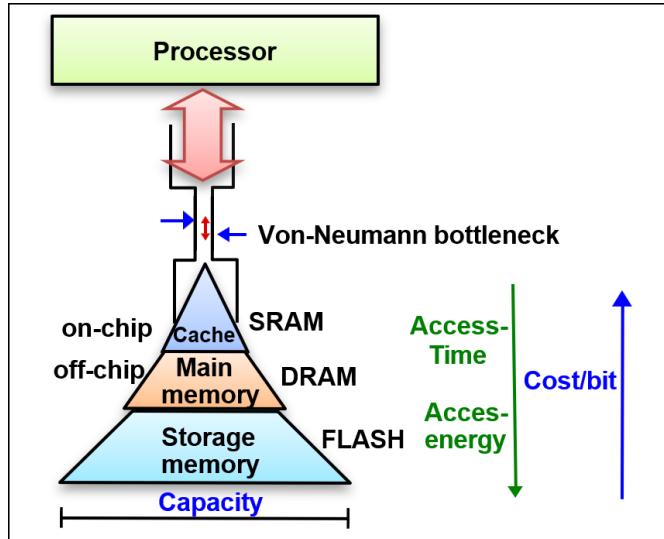


Figure 1.1: Diagram of Von-Neumann bottleneck.

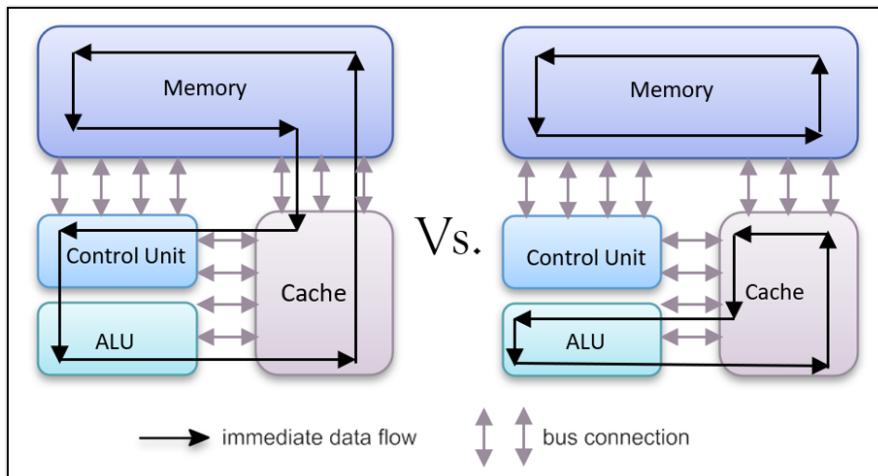


Figure 1.2: Von-Neumann architectural schemes in contrast to in-memory computing.

Figure 1.1 tells us that, "The von Neumann bottleneck is the slowdown that occurs because data has to move back and forth between the CPU and memory, causing delays in processing. Memory hierarchies involve trade-offs between cost/bit, energy, and access time. While slower access and higher energy consumption can be achieved at a cheaper cost, faster access and lower energy consumption come at a higher expense. "

Figure 1.2 is about the Von-Neumann architectural schemes in contrast to in-memory computing.

CHAPTER 2

Literature Survey

Literature Survey on In-Memory Computing Using ReRAM Devices:

2.1 Research Gap/Motivation:-

However, a lot of research has been done on in-memory computing using ReRAM devices, but there is a lack of research on its vast variety of applications. The thorough investigation of ReRAM devices leads to the possibility of their application in vast areas or fields.

2.2 Literature Review:

After getting familiar with the basics of in-memory computing using ReRAM, the focus was shifted to its wide variety of applications.

I have read around eight research papers based on such materials. Important terms, their description, and findings from each have been mentioned below.

This paper [1] introduces a unique in-memory multiplier sign that combines the Wallace Tree architecture with majority logic and is specifically optimized for ReRAM devices. This method focuses on optimizing the addition steps of the Wallace Tree, a famous method for achieving rapid multiplications, by running numerous majority gates in parallel within the ReRAM array. This parallelism reduces latency dramatically when compared to standard memory multipliers, resulting in faster and more efficient computations. Furthermore, the architecture is energy-efficient, making use of the low-power properties of ReRAM technology, and it can be implemented in ordinary memory arrays without requiring significant changes to existing peripheral circuits. Overall, this study offers a substantial leap in in-memory computing, displaying how high-performance multipliers may be effectively realized using ReRAM, with prospective applications in

fields such as matrix operations and signal processing.

This paper [2] emphasizes on improving the addition operation in Re-RAM arrays. The authors offer a unique strategy for reducing addition latency, which is an important component of many computational jobs, that makes use of parallel-friendly majority gates. The suggested solution achieves significant additional performance gains by employing Re-RAM's intrinsic features for the bulk of logic operations.

This paper [3] explores the use of distributed arithmetic for efficient in-memory inner product computations. This approach involves representing the input values as pulse trains and performing the computations using a series of additions and shifts. By leveraging the inherent capabilities of Re-RAM for these operations, the authors demonstrate a significant reduction in latency and energy consumption compared to traditional von Neumann architectures.

This paper [4] addresses an improved modelling methodology for ReRAM devices, based on the Stanford-PKU model and adding multiple-layer capabilities. Traditionally, the Stanford-PKU model mimics ReRAM's binary resistance states; however, this expansion improves simulation accuracy by integrating numerous intermediate resistance levels. This advancement is critical for accurate modelling of ReRAM-based circuits, allowing researchers to better understand the impacts of device changes and optimise designs for in-memory computing. The paper makes an important contribution to the advancement of ReRAM technology and the performance of ReRAM-based systems.

This paper [5] focuses on improving the read operation in STT-MRAM memory by proposing a time-based sensing approach that eliminates the need for a reference voltage. This approach strengthens the read process, making it less sensitive to fluctuations in device characteristics. Although the system is intended for STT-MRAM, the notion of translating voltage signals into time measurements could be extended for usage in ReRAMs. This modification may increase the reliability and performance of ReRAM-based memory systems, making them more appropriate for critical applications that require high data integrity and reliability.

This paper [6] utilizes majority logic, which is essential for activities such as sorting, searching, and decision-making. By leveraging ReRAM's intrinsic capabilities for majority logic operations, the comparator efficiently executes in-memory comparisons, removing the demand for external data transfer. This approach not only simplifies the comparator architecture but also decreases circuit complexity, resulting in higher per-

formance and lower power consumption than standard methods. Overall, the research stresses the potential of this approach to speed up computing jobs that rely on comparison operations, making it an important contribution to the area.

This paper [7] studies the implementation of binary addition directly within a ReRAM array using majority logic, which is a basic operation for many computational applications. By leveraging ReRAM’s intrinsic features, the technique enables efficient in-memory binary addition without the need for additional hardware. The author shows that by properly structuring the ReRAM array and programming the input values, binary addition may be performed directly within memory. This technology has the potential to substantially enhance performance and lower energy usage by eliminating the requirement for data transfer between the processor and memory. The integration of addition operations within the memory array simplifies the computational pipeline, resulting in more efficient and energy-saving in-memory computing designs.

This paper [8] reveals ReRAM devices’ ability to construct functionally complete Boolean logic, indicating that ReRAM can perform any logical operation, making it a very versatile computing element. The authors effectively build basic logic gates such as AND, OR, and NOT by carefully managing the resistance state and gate voltage in 1T1R ReRAM cells, which are composed of a single transistor and a resistive element. This invention has important significance for in-memory computing since it enables complicated logical operations to be carried out directly within the memory array, minimizing the need for data transfer between the processor and memory. As a result, our work paves the door for energy-efficient computational architectures, laying the foundation for future advances in ReRAM-based in-memory computing systems.

2.3 Problem Statement

2.3.1 Recent advancements in automating complex computations

Recent advancements in automating complex computations, such as Binary Addition implementation and other in-memory operations, open new possibilities for efficient architectures.

Target

To make automatic Circuits for performing different-different Computations.

2.3.2 Existing 2-D transforms rely on power-intensive CPU-GPU systems.

ReRAM offers a low-power alternative, but challenges exist:

- Device non-idealities affecting accuracy and reliability.
- Scalability for large-scale transformations in real-time applications.
- Efficient integration of sense amplifiers for precise read and write operations.

Target

To make Circuits for performing 2-D Transforms.

2.4 Hadamard Matrix

Understanding the Basics and Applications : The Hadamard Transform is a mathematical operation used in various fields like signal processing, image compression, and quantum computing. It transforms a set of values into a new set of values using orthogonal basis vectors, simplifying many computational problems.

The Hadamard matrix is an $N \times N$ matrix, where N is a power of 2. It contains only +1 and -1 entries and is defined recursively:

2.4.1 Equation

$$H_1 = [1] \quad H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad H_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

2.4.2 Performing the Hadamard Transform

The Hadamard Transform of a vector x is given by multiplying it by the Hadamard matrix H :

$$y = H_N \times x$$

This operation converts the input vector into a new vector, which is a combination of the Hadamard basis vectors.

2.4.3 Applications

The Hadamard Transform has a wide range of applications, including:

Signal Processing: Image compression, filtering, and analysis. Quantum Computing: Creating superposition states and implementing quantum algorithms. Error Detection and Correction: Constructing error-correcting codes.

2.4.4 Properties

The Hadamard Transform possesses several key properties:

Orthogonality: The Hadamard matrix is an orthogonal matrix, preserving the length of vectors. Fast Computation: The Fast Walsh-Hadamard Transform (FWHT) algorithm allows for efficient computation in $O(N \log N)$ time. Binary and Orthogonal Basis Vectors: The Hadamard matrix uses binary (+1, -1) orthogonal basis vectors.

The Hadamard Transform is a powerful tool in computational mathematics with diverse applications. Its properties make it efficient and effective for various tasks.

2.5 Conclusion

According to the literature, ReRAM-based in-memory computing shows great promise for implementing computationally intensive tasks such as the Hadamard transform. The studies examined show the potential for increased speed, energy efficiency, and scal-

ability. However, challenges with device reliability, integration, and accuracy persist, highlighting the need for additional research and development in this area. As ReRAM technology matures, it is expected to play an important role in advancing in-memory computing architectures for a variety of applications, including signal processing and machine learning.

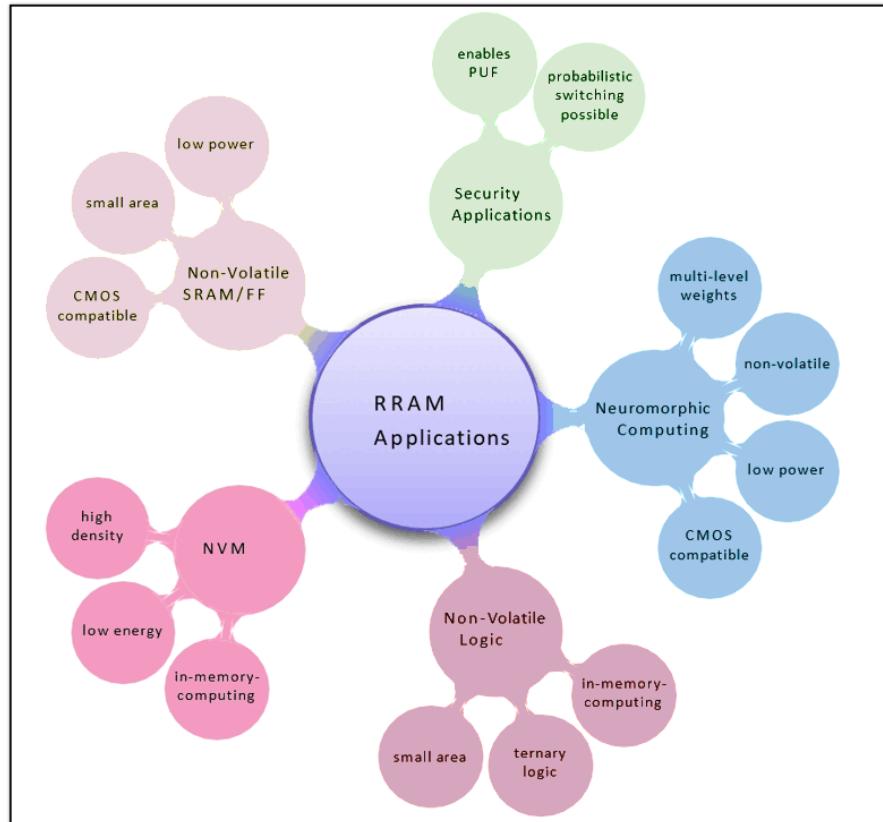


Figure 2.1: Various Applications of ReRAM

CHAPTER 3

Methodology

3.1 ReRAM Basics

A memory device belonging to the memristor family, resistive RAM (ReRAM) has numerous advantages, particularly low write energy, resilient scalability, non-volatility, high density, and a favorable resistance window. It is possible to reversibly shift the resistance of a ReRAM cell between LRS and HRS through putting the proper voltage across it. The formation or rupture of a conductive filament between the cell's two electrodes induces this resistance transformation.

ReRAM cells are normally organized in a memory array with one access transistor per cell, often known as a 1 Transistor-1 Resistor (1T-1R) array. The access transistor enables individual cell selection within the array, preventing sneak path currents, and restricting the current traveling through the ReRAM cell.

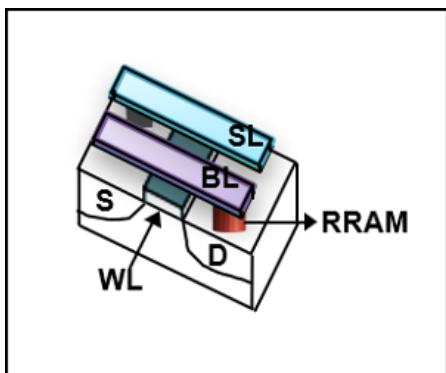


Figure 3.1: Structure of 1T1R Cell

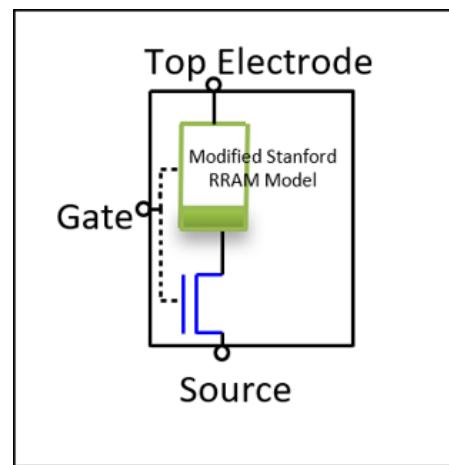


Figure 3.2: 1T-1R Cell

The ITIR cell consists of a transistor with a source, drain, and gate, as well as a resistive RAM (RRAM) element. The transistor functions as a selector, directing the flow of current between the source and drain. The gate regulates the transistor's conductivity, enabling or preventing the flow of current. The RRAM element stores data by changing

its resistance in response to the applied voltage. The 1T-1R cell is being studied as a potential substitute for regular DRAM memory cells due to its increased density and reduced power consumption.

3.2 Design

Design:

- Using Stanford PKU model of RRAM
- Testing of Writing and Reading data from ReRAM cells.
- Testing of Majority Logic using Sense Amplifier from ReRAM cells.
- Testing of Binary Arithmetic using Sense Amplifier from ReRAM cells.
- Designed using Cadence Virtuoso.

Simulation:

- Evaluated performance using Cadence Spectre simulations.

3.3 Stanford-PKU (Peking University) RRAM Model

3.3.1 Overview of the Model

We used the Stanford PKU model of RRAM for our simulations and testing. This model offers the following features:

The Stanford-PKU RRAM Model is a SPICE-compatible compact model describing the switching performance of bipolar metal oxide RRAM. It is scalable and imposes no inherent size limitations on RRAM cells. The model simplifies the complex process of ion and vacancy migration into the growth of a single dominant filament, preserving the essential switching physics. The primary variable determining the device resistance is the tunneling gap (g), which represents the distance between the tip of the filament and the opposite electrode.

3.3.2 Physical Structure of RRAM Cells

The physical structure of the RRAM cell in this model comprises the following layers:

- TiN: 50 nm

- TiO_x : 5 nm
- HfO_2 : 1.7, 2.1, and 3.3 nm (varies depending on the configuration)
- Pt: 50 nm

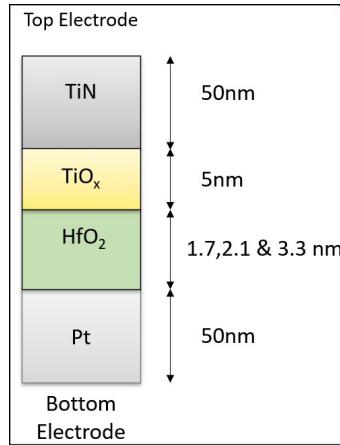


Figure 3.3: Layer structure of the Stanford-PKU RRAM Model (Not to Scale)

3.3.3 Key Insights

The model effectively captures the switching behavior, including the set and reset processes, by tracking the formation and rupture of the conductive filament. It provides accurate predictions of resistance states for use in computational tasks such as binary addition and logic gate implementations.

3.3.4 Applications in ReRAM-Based Computing

Using this model, we simulated ReRAM cell behavior for binary arithmetic operations and logic gate design. The simplified filament growth dynamics enable efficient testing of in-memory computing operations in a scalable and reliable manner.

3.4 Summarized Phenomenon: Bipolar Switching in ReRAM

3.4.1 SET Process

RESET Operation in ReRAM:

During this process, the ReRAM device transitions to a High Resistance State (HRS), becoming non-conductive. A positive voltage is applied to the top electrode (TE) and a negative voltage to the bottom electrode (BE). Oxygen ions (O_2) migrate towards the TE, creating oxygen vacancies (V_o) in the insulating layer. This breaks the conductive filament, reducing free electrons and causing high resistance.

Transition from RESET to SET in ReRAM:

Applying a higher positive voltage to the TE and BE generates an electric field that drives electrons from the BE towards the TE. This prompts downward migration of oxygen vacancies (V_o), initiating filament reconstruction between the BE and TE.

SET Operation (Low Resistance State - LRS):

With further voltage increase, sufficient energy is provided to fully form a conductive filament composed of oxygen vacancies and free electrons. This filament enables efficient current flow, switching the device to a low resistance state (LRS). The alignment and bonding of oxygen vacancies with electrons complete the conductive path, programming the ReRAM cell to LRS.

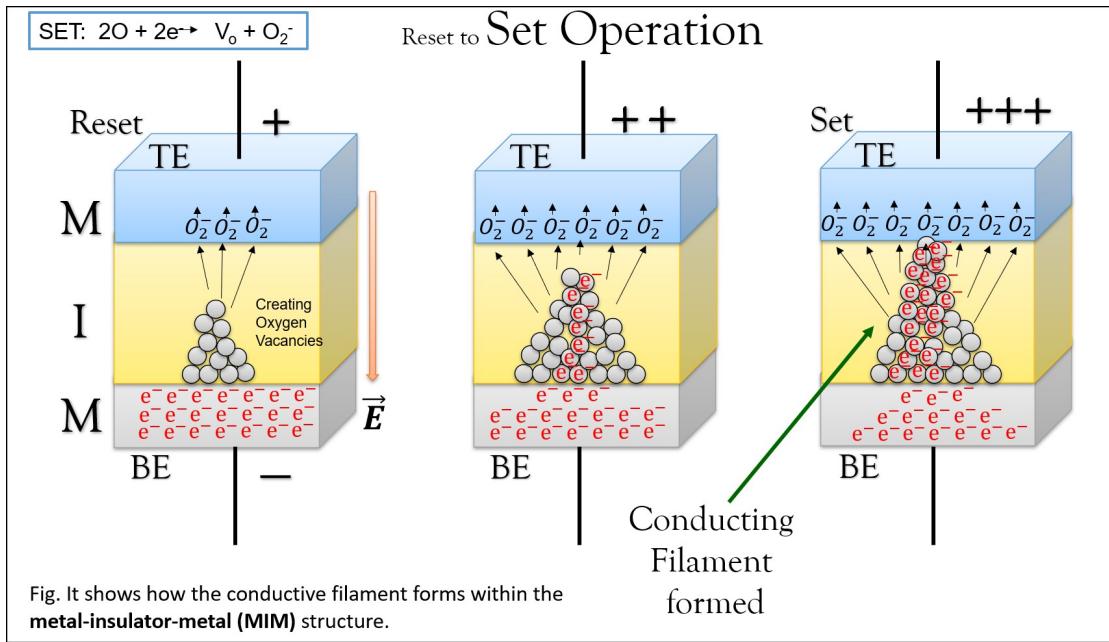


Figure 3.4: Set Operation

3.4.2 RESET Process

SET Operation in ReRAM:

In the Low Resistance State (LRS), the ReRAM cell is in a conductive state. A filament composed of oxygen vacancies (V_o) and free electrons is formed between the top electrode (TE) and the bottom electrode (BE). A negative voltage is applied to the TE, and a positive voltage to the BE. The LRS represents the programmed state, where the memory cell stores the binary data "1."

Transition from SET to RESET in ReRAM:

A reverse polarity voltage is applied to the TE and BE, generating an electric field that drives oxygen ions (O^{2-}) back into the conductive filament region. These oxygen ions combine with oxygen vacancies (V_o), reducing the number of free electrons and weakening the conductive path.

RESET Operation (High Resistance State - HRS):

In the RESET state, the conductive filament is completely destroyed, and the device transitions to a high resistance state (HRS). The annihilation of oxygen vacancies continues until the conductive filament is fully broken, making the path between the TE and BE highly resistive. With no conductive filament, the current flow through the memory cell is minimal, resulting in a high resistance state. This effectively "programs" the ReRAM cell to HRS.

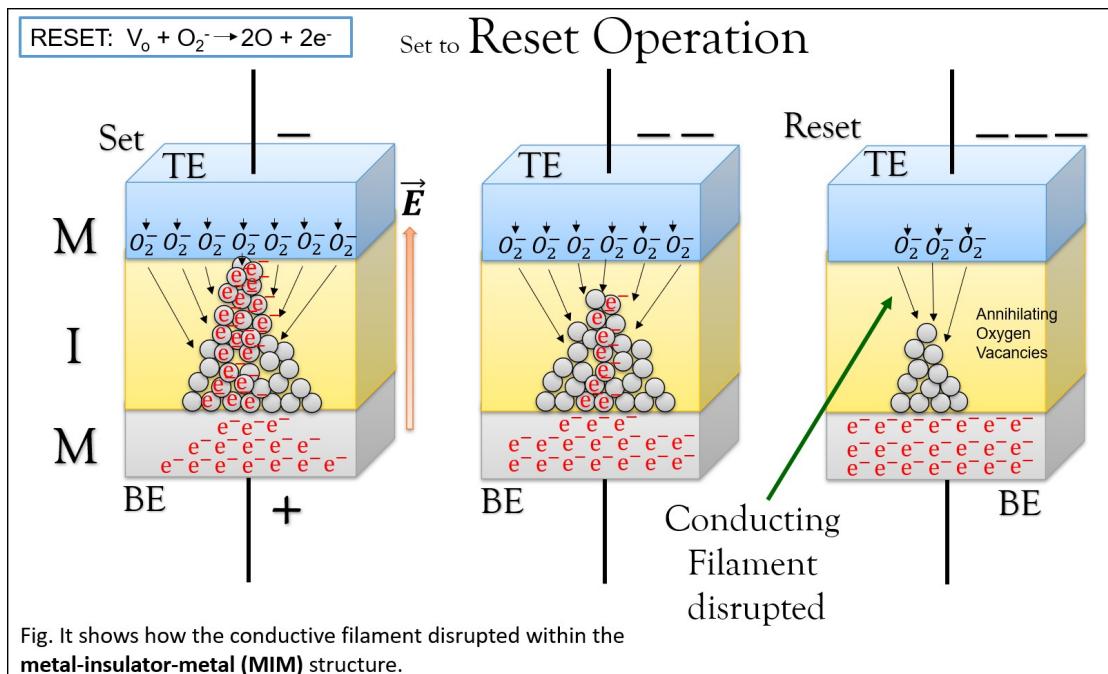


Figure 3.5: ReSet Operation

3.4.3 Key Insights:

- Switching between HRS and LRS is controlled by the movement of oxygen vacancies and filament formation.
- Typical in oxide-based ReRAM (e.g., HfO_2).
- Data storage achieved by manipulating the resistance state of the memory cell.

3.5 1T1R Cell

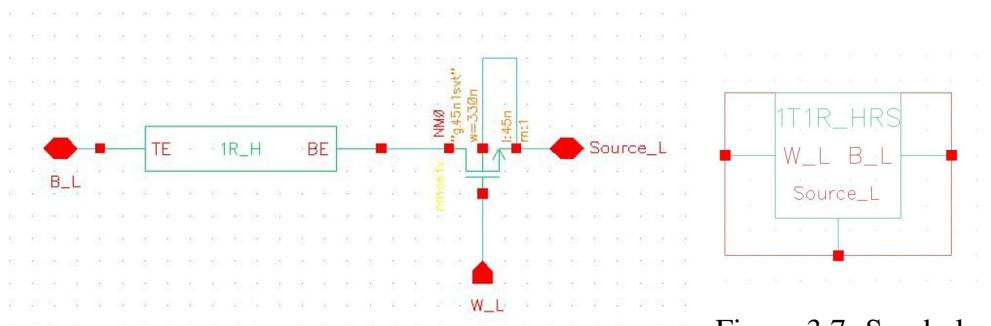


Figure 3.6: 1T1R Cell

A 1T1R memory cell consists of a single transistor and a single resistive element, typically a ReRAM device, which is used for memory storage. The cell architecture has the following components:

- **Word Line (WL):** Controls the gate of the transistor.
- **Bit Line (BL):** Connected to the drain of the transistor. Used to read and write data.
- **Source Line (SL):** Connected to the source of the transistor. Typically tied to ground during read and write operations.
- **ReRAM Element (R):** Represents the resistive memory element, which stores data based on its resistance (Low Resistance State - LRS or High Resistance State - HRS).

3.5.1 Testing of Writing and Reading data from ReRAM cells.

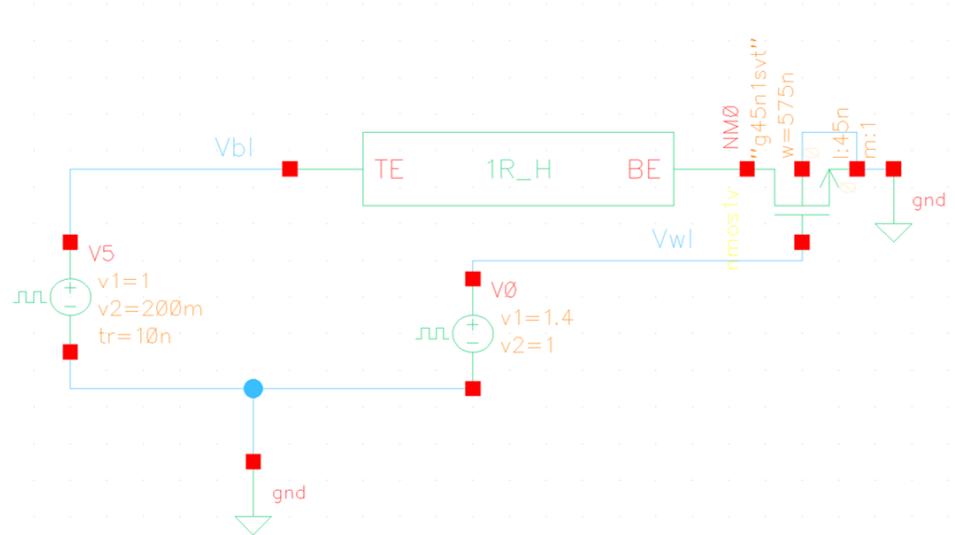


Figure 3.8: Write 1

3.5.2 ReRAM Operation Table for Write-1

Time (μs)	Word Line Voltage (V_{wl})	Bit Line Voltage (V_{bl})	Source Line	Operation	Current (μA)	Resistance State / Value (kΩ)
0 – 2	1 V	200 mV	GND	Initial Read	~1.5	~133.3 (HRS)
Description: Safe sensing of the ReRAM's resistance without altering its state.						
2 – 12	1.4 V	1 – 1.3 V	GND	Set Operation	~250	LRS
Description: Applying 1 V to the bit line initiates the set process, forming a conductive filament (LRS).						
12 – 14	1 V	200 mV	GND	Final Read	~20.5	~9.76 (LRS)
Description: Confirms that the set operation took place in the ReRAM cell without disturbing the state.						

Table 3.1: ReRAM Set and Reset Processes: Voltage, Current, and Resistance State Transitions

3.5.3 Testing of Writing and Reading data from ReRAM cells.

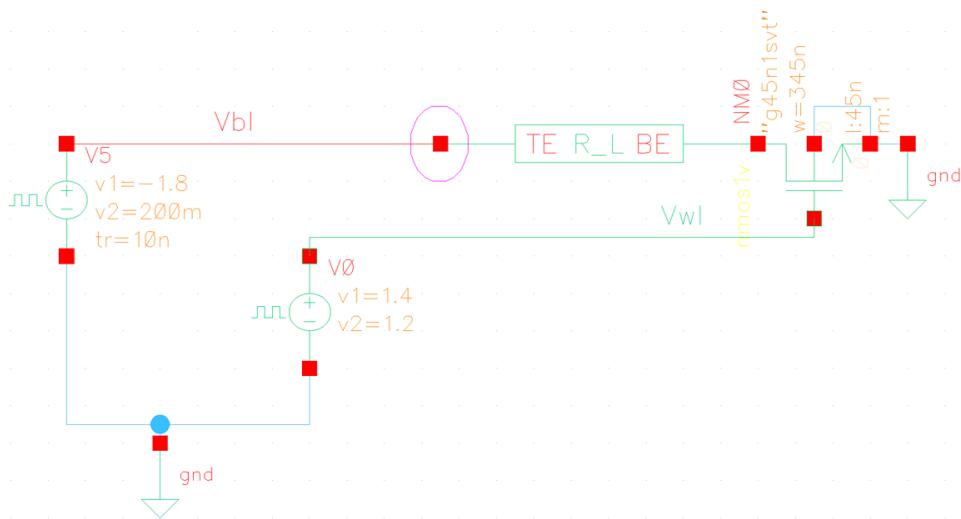


Figure 3.9: Write 0

3.5.4 ReRAM Operation Table for Write-0

Time (μs)	Word Line Voltage (V_{wl})	Bit Line Voltage (V_{bl})	Source Line	Operation	Current (μA)	Resistance State / Value (kΩ)
0 – 2	1 V	200 mV	GND	Initial Read	~20.5	~9.76 (LRS)
Description: Safe sensing of the ReRAM's resistance without altering its state.						
2 – 12	1.4 V	[-1.3, -1.8] V	GND	Reset Operation	~250	HRS
Description: Applying -1.8 V to the bit line initiates the reset process, disrupting the conductive filament (HRS).						
12 – 14	1 V	200 mV	GND	Final Read	~1.5	~133.3 (HRS)
Description: Confirms that the Reset operation took place in the ReRAM cell without disturbing the state.						

Table 3.2: ReRAM Reset Operation and Resistance State Transitions

3.5.5 Overall Table for Different Operations

Table 3.3: Table of Word Line, Bit Line Voltages, and Resistance States for Different Operations

Word Line Voltage (V_{wl})	Bit Line Voltage (V_{bl})	Source Line	Operation	Description	Resistance State
1 V	200 mV	GND	Read Operation	Bit-line at 200mV and word-line at 1V allows for safe sensing of RRAM's resistance without changing its state.	Unknown (Assumed LRS if no reset operation has occurred)
1.4 V	-1.8 V	GND	Reset Operation	Wordline at 1.4V and bit-line at -1.8V create a large potential difference, breaking the conductive filament and switching RRAM to high-resistance state (HRS).	HRS
1 V	200 mV	GND	Read Operation	Again, set to 1V and 200mV, this read operation measures the RRAM resistance to confirm it is in HRS without disturbing the cell's state.	HRS
1.4 V	1–1.2 V	GND	Set Operation	Bit-line at 1.2V and word-line at 1.4V initiate the Set operation, forming a conductive filament and switching RRAM to LRS.	LRS
1 V	200 mV	GND	Read Operation	Again, set to 1V and 200mV, this read operation measures the RRAM resistance to confirm it is in LRS without disturbing the cell's state.	LRS

3.6 Testing of Majority Logic using Sense Amplifier from ReRAM cells.

Explanation:

- The output is 1 if the majority (two or more) of the inputs are 1.
- The output is 0 if the majority of the inputs are 0.

A	B	C	$M_3(A, B, C)$	V_{bl} (mV)	Digital Logic
0	0	0	0 (means LRS here)	86.2261	1
0	0	1	0	125.5681	1
0	1	0	0	125.5681	1
0	1	1	1	228.212	0
1	0	0	0	125.5681	1
1	0	1	1	228.212	0
1	1	0	1	228.212	0
1	1	1	1 (means HRS here)	648.261	0

Table 3.4: Truth Table for a 3-Input Majority Gate

3.7 Binary Addition Flowchart and Details

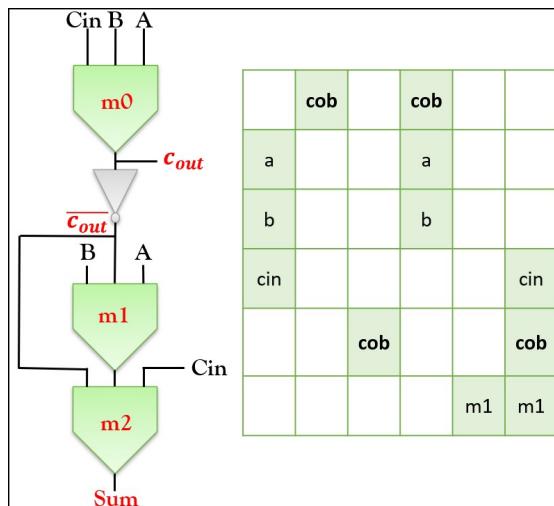


Figure 3.10: Flow Chart for Binary Addition[7]

Details of the Operation:

- **Cycle 1:** Read a , b , and c_{in} .
- **Cycle 2:** Write to c_{ob} .
- **Cycle 3:** Write to c_{ob} again.
- **Cycle 4:** Read a , b , and c_{ob} to compute m_1 .
- **Cycle 5:** Write to m_1 .
- **Cycle 6:** Read c_{in} , c_{ob} , and m_1 to compute the sum.

References:

[7] John Reuben, *Micromachines*, 2020, 11(5), 496. DOI: <https://doi.org/10.3390/mi11050496>

3.7.1 Testing of Binary Arithmetic using Sense Amplifier from ReRAM cells.

Steps for the Operation

1. **Step-1:** Making $A = \text{HRS}, B = \text{HRS}, C_{\text{IN}} = \text{HRS}$.
2. **Step-2:** Reading $A = \text{HRS}, B = \text{HRS}, C_{\text{IN}} = \text{HRS}$.
3. **Step-3:** Making/Reading $\overline{C_{\text{OUT}}}(1)$ and $\overline{C_{\text{OUT}}}(5)$ as LRS.
4. **Step-4:** Reading $A = \text{HRS}, B = \text{HRS}, \overline{C_{\text{OUT}}} = \text{LRS}$.
5. **Step-5:** Making $m_1(6) = \text{HRS}$.
6. **Step-6:** Reading $\overline{C_{\text{OUT}}} = \text{LRS}, m_1 = \text{HRS}, C_{\text{IN}} = \text{HRS}$.

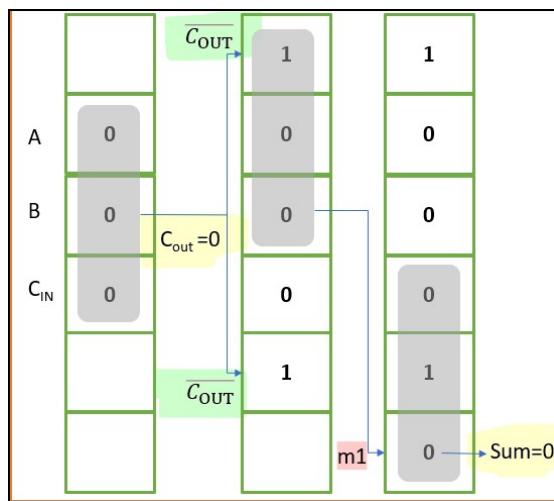


Figure 3.11: Testing of Binary Arithmetic

3.8 Circuit Overview

- Detailed circuit design for automated ReRAM operations.
- Use of Demultiplexer to give Inputs to Triple Row Decoder for turning on the Cells as per our wish.
- Use of Triple Row Decoder to turn on single/three Cells at a time which is totally different from Conventional Decoder.
- Use of Sense Amplifiers to sense (Read) the state of three cells at a time.
- Integration of control logic for turning ON the Word-Lines.
- Integration of control logic for write/read cycles.
- Integration of control logic for write/read from same Cell.
- Doing all the cycles with the help of counter where Counts = Cycles.

3.9 Flow-Chart

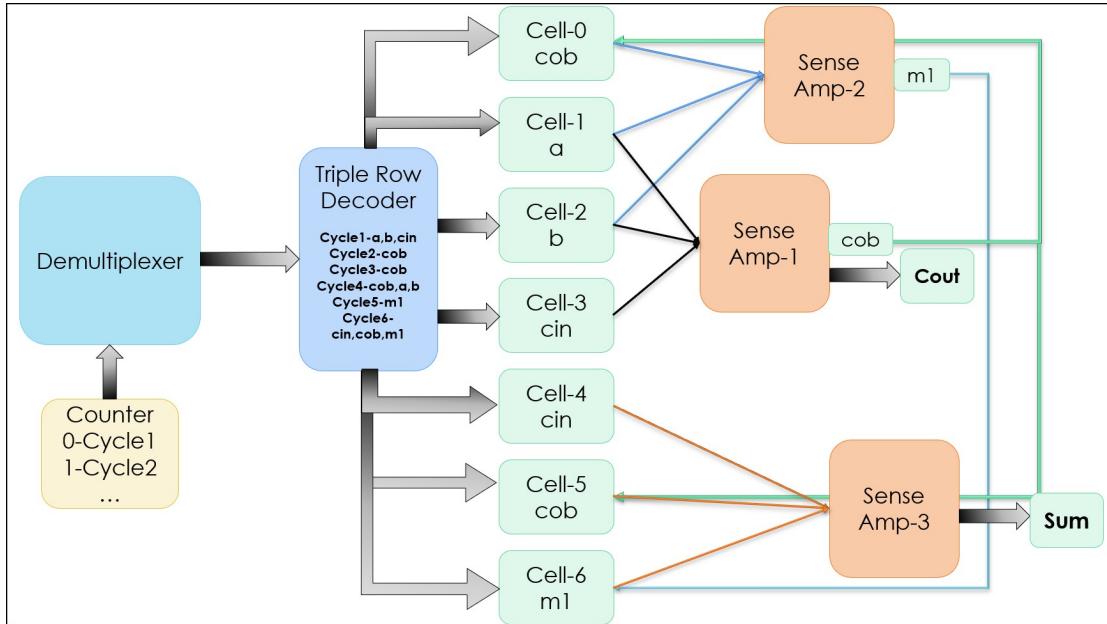


Figure 3.12: Flow-Chart for the Circuit

Key Points:

1. The flowchart operates sequentially, with the counter managing cycles.
2. **Sequential Operation:** The flowchart illustrates a sequential operation, where a counter controls the progression through different cycles. Each cycle activates specific cells and sense amplifiers to perform a particular logic operation.
3. **ReRAM-Based Processing:** The core of the operation relies on ReRAM cells. These cells store information in resistive states, enabling memory-based processing of logic operations. This approach potentially offers advantages in terms of energy efficiency and speed.
4. **Sense Amplifier Integration:** Sense amplifiers are crucial for converting the stored resistive states in ReRAM cells into readable digital outputs. They play a vital role in accurately detecting the state of each cell.
5. **Demultiplexer and Decoder:** The demultiplexer and decoder work together to select the appropriate ReRAM cells for each cycle. This ensures that the correct data is accessed and processed.

3.9.1 Cycle-by-Cycle Operations

Inputs are as follows: $a = 0, b = 0, c = 0$.

Cycle	Counter Output	DEMUX Output	MAJ	Triple Row Decoder Output	Operation	Output of Action
1	000	I_0	1	0001	Compute $cout, cob$ (Majority of a, b, cin)	$cout = 0, cob = 1$
2	001	I_1	0	0000	Write cob to Cell-0	$cob = 1$ stored
3	010	I_2	0	0010	Write cob to Cell-5	$cob = 1$ stored
4	011	I_3	1	0101	Compute $m1$ (Majority of a, b, cob)	$m1 = 0$
5	100	I_4	0	0100	Write $m1$ to Cell-6	$m1 = 0$ stored
6	101	I_5	1	0111	Compute Sum (Majority of $cin, cob, m1$)	Sum=0

Table 3.5: Cycle-by-cycle operations for the full adder design using ReRAM cells.

3.9.2 Counter

- The counter output determines the specific operation to perform in the circuit.
- Each count is passed to the demultiplexer to activate one output line at a time.
- The activated demultiplexer output is connected to the triple row decoder.
- The triple row decoder decodes the input signal to simultaneously activate the required rows in the memory array.

3.9.3 Demultiplexer

- The demultiplexer receives input from the counter.
- It maps the counter's output to one of its output lines (e.g., Cycle 1 to Cycle 6).
- Each demux output corresponds to a specific operation:
- Cycle 1: Read rows for a, b , and cin .
- Cycle 2: Write cob to Cell-0.
- Cycle 3: Write cob to Cell-5.
- Cycle 4: Compute $m1$ by reading rows for a, b , and cob .
- Cycle 5: Write $m1$ to Cell-6.
- Cycle 6: Compute the sum by reading rows for cin, cob , and $m1$.

3.9.4 Triple Row Decoder

- The decoder receives demux output and controls word-lines for ReRAM cells and supports:
- Single-row WRITE operations for storing cob and $m1$.
- Multi-row READ operations for computing majority results.

- Example operations are as follows:
- Cycle 1: Activates rows for a , b , and cin (Cells 1, 2, 3).
- Cycle 2: Activates row for Cell-0 to write cob .

3.9.5 ReRAM Cells

- ReRAM cells act as storage and computational elements:
- Cell-0: Stores cob .
- Cell-1: Stores a .
- Cell-2: Stores b .
- Cell-3: Stores cin .
- Cell-4: Stores another cin for computations.
- Cell-5: Stores cob .
- Cell-6: Stores $m1$, the intermediate majority result.
- Computations and storage occur simultaneously in memory for energy efficiency.

3.10 Sense Amplifiers

- The selected rows interface with sense amplifiers, which read or compute the data stored in ReRAM cells.
- Outputs from the sense amplifiers represent computed values (e.g., carry-out, majority, sum) or signals to be written back to the memory cells.
- The entire operation is synchronized with the clock to ensure seamless automation across cycles.
- Sense amplifiers perform READ operations to compute outputs:
- Sense Amp-1: Computes co and cob (from a , b , cin).
- Sense Amp-2: Computes $m1$ (from a , b , cob).
- Sense Amp-3: Computes the sum (from cin , cob , $m1$).
- Amplifiers ensure accurate read operations by detecting ReRAM resistance levels.

3.10.1 Components of Sense Amplifier

Common Drain MOS (BL Booster)

The PMOS transistor (M1) boosts the bit-line voltage for time-based comparison. It operates as a voltage follower, buffering the bit-line voltage without significant loading. The output of M1 is the sum of the bit-line voltage (V_{BL}) and the source-gate voltage (V_{SG}).

Current Starved Inverters

The current-starving transistor (M2) controls the current through the M3-M4 inverter stage. The gate of M2 is controlled by the boosted bit-line voltage ($V_{BL} + V_{SG}$).

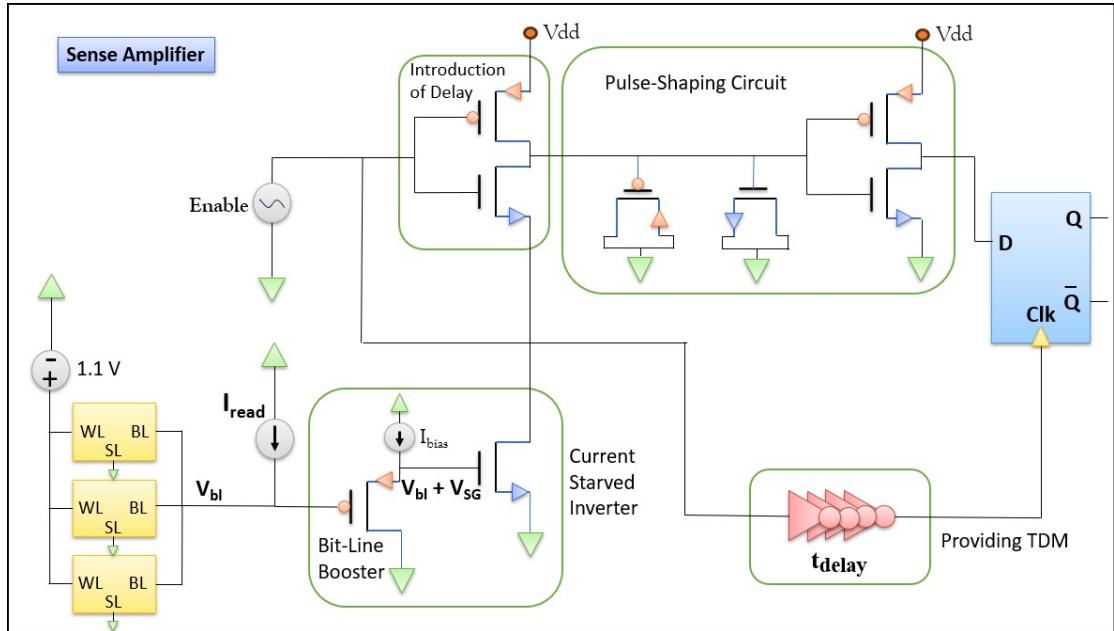


Figure 3.13: Sense Amplifier Schematic Diagram

MOS Capacitors (M5, M6)

MOS capacitors exploit the inherent gate and drain-source capacitance of MOSFETs. These capacitors add load capacitance to shape the timing characteristics of the circuit and smooth signal transitions.

Pulse-Shaping Inverter Stage (M7, M8)

The pulse-shaping inverter stage sharpens and cleans up the signal before it reaches the D-flip flop. It also reduces noise and distortion from earlier stages, ensuring a clean output.

Chain of Inverters

The chain of inverters amplifies and stabilizes signals to ensure reliable operation of the circuit.

D Flip Flop (D-FF)

The D-flip flop captures the state of the signal at the critical Time Decision Moment (TDM). It ensures accurate memory state detection based on the timing of the input signal.

Programmable Delay Line (PDL)

The programmable delay line adjusts the signal propagation time for precise operation. It compensates for variations in bit-line voltage and resistances, ensuring proper timing.

Time-Domain Comparator (D-FF)

The time-domain comparator samples the input signal at TDM for decision-making. It outputs a low signal for low-resistance states (LRS) and a high signal for high-resistance states (HRS).

3.10.2 Key Concepts in Time-Based Sensing

Time Decision Moment (TDM)

The TDM refers to a specific point in time for decision-making in time-based sensing circuits.

Voltage-to-Time Conversion

Current-starved inverters are used to convert voltage differences into timing signals.

Decision Process

For low-resistance states (LRS), higher current results in shorter delays and earlier TDMs.

For high-resistance states (HRS), lower current causes longer delays and later TDMs.

3.10.3 Role of W/L Ratios in CMOS Design

The W/L ratio affects a transistor's electrical characteristics. A larger W/L ratio improves current-carrying capacity and switching speed, but may lead to higher power consumption. It also enhances noise immunity.

3.10.4 Timing Characteristics

The Time Decision Moment (TDM) typically occurs between $4.8k\Omega$ and $8.7k\Omega$, ensuring accurate resistance state sensing.

3.10.5 Operation Summary (Cycle by Cycle)

- **Cycle 1:** Compute co and cob (majority of a, b, cin).
- **Cycle 2:** Write cob to Cell-0.
- **Cycle 3:** Write cob to Cell-5.
- **Cycle 4:** Compute $m1$ (majority of a, b, cob).
- **Cycle 5:** Write $m1$ to Cell-6.
- **Cycle 6:** Compute sum (majority of $cin, cob, m1$).

3.11 Circuit

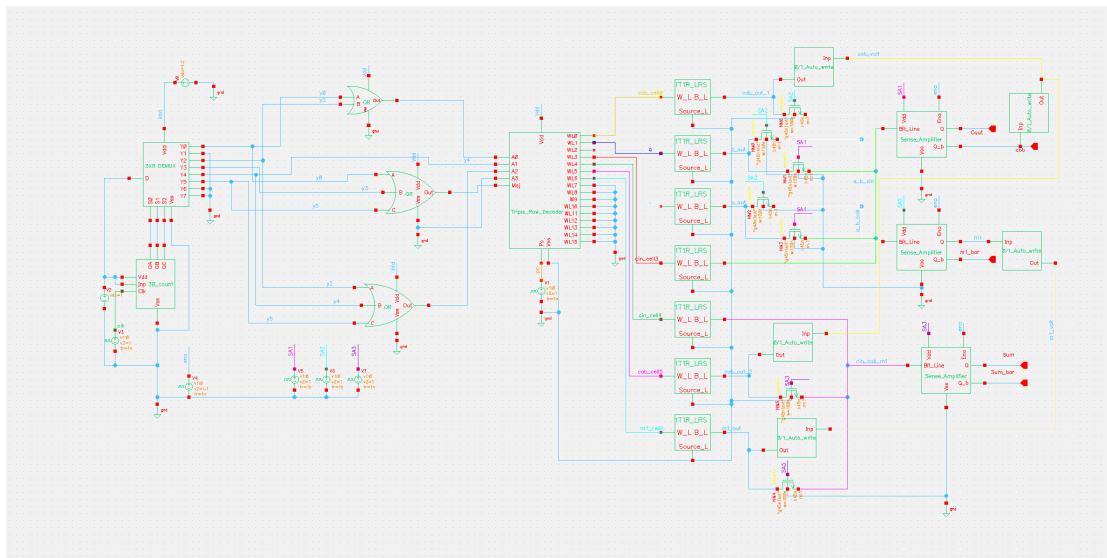


Figure 3.14: Example Circuit Overview

CHAPTER 4

Results and Discussions

4.1 Results

- **Test 0:** Testing of correctly turning ON the respective word lines.
- **Test 1:** Testing of correctly sensing a,b,cin, and later sensing cob,a,b.
- **Test 2:** Testing of correctly sensing a,b,cin, and writing 1.3V in cob.
- **Test 3:** Testing of sensing a,b,cin, writing 1.3V in cob and again sensing cob,a,b.
- **Test 4:** Testing of writing 1.3V in cob with 2nd Clock Pulse.
- **Test 5:** Testing of writing 1.3V in cob with 2nd Clock Pulse with a delay element.

4.1.1 Circuit Diagram for Test 0

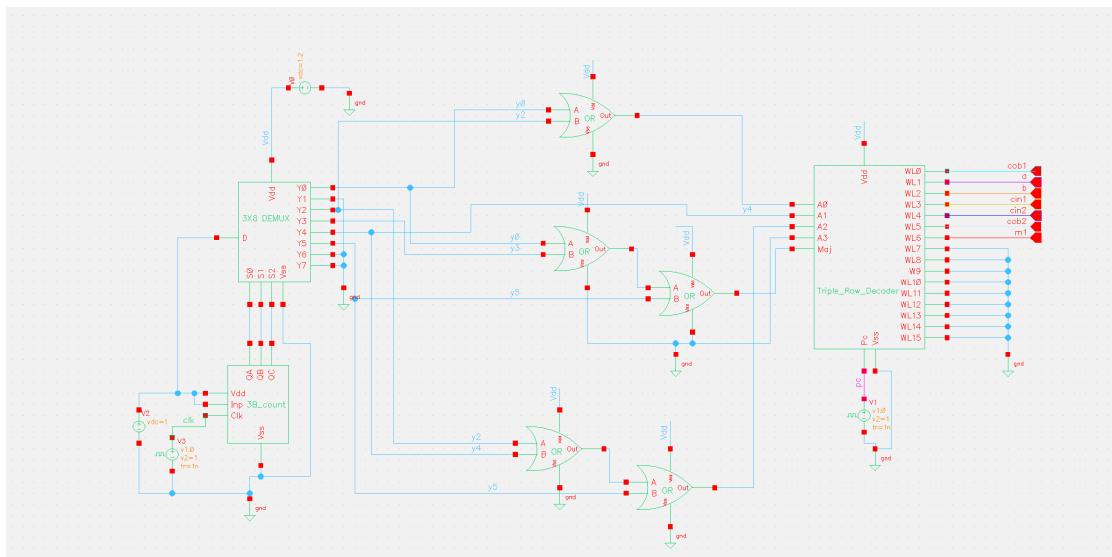


Figure 4.1: Circuit for Test 0: Testing of correctly turning ON the respective word lines.

4.1.2 Waveform for Test 0

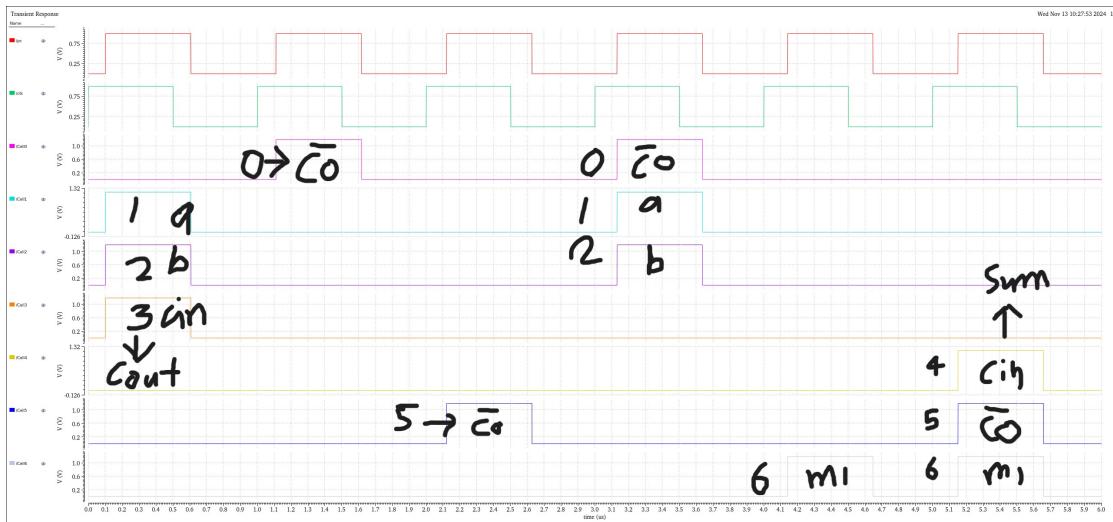


Figure 4.2: Waveform for Test 0: Testing of correctly turning ON the respective word lines.

4.1.3 Results Table for Test 0

- **Test Condition:** Turning ON respective word lines based on clock pulses.
- **Cycle 1:** Cells for a , b , and cin are activated (Cell-1, Cell-2, and Cell-3).
- **Cycle 2:** $Cell-0$ (cob) is activated for writing.
- **Cycle 3:** $Cell-5$ (cob) is activated for writing.
- **Cycle 4:** Cells for a , b , and cob (Cell-1, Cell-2, and Cell-5) are read to compute $m1$.
- **Cycle 5:** $Cell-6$ ($m1$) is activated for writing.
- **Cycle 6:** Cells for cin , cob , and $m1$ (Cell-3, Cell-5, and Cell-6) are read to compute the sum.

4.1.4 Circuit Diagram for Test 1

Figure 4.3: Circuit for Test 1: Testing of correctly sensing a,b,cin, and later sensing cob,a,b.

4.1.5 Waveform for Test 1

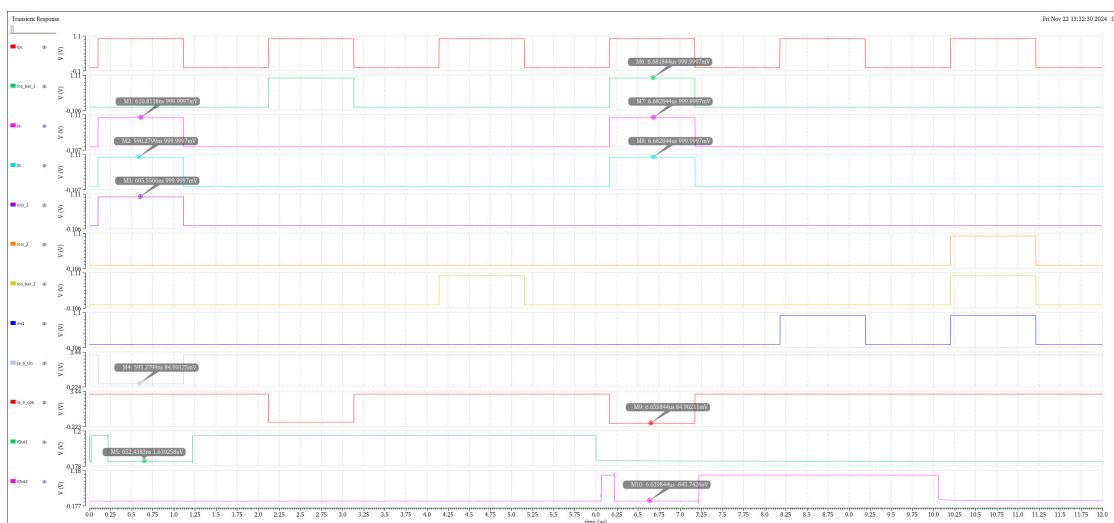


Figure 4.4: Waveform for Test 1: Testing of correctly sensing a,b,cin, and later sensing cob,a,b.

4.1.6 Results Table for Test 1

- **Test Condition:** Sensing a, b, cin, and cob in sequential cycles.
 - **Cycle 1:** Sensed a, b, cin with Sense Amp Input readings around 100 mV.
 - **Cycle 2:** Sensed cob, a, and b with cob at around 100 mV.
 - **Data Integrity:** Values of a, b, and cin correctly retained across cycles.

- **Circuit Behavior:** Sequential read operations completed successfully with no data corruption with the help of Pass Transistors.

4.1.7 Circuit Diagram for Test 2

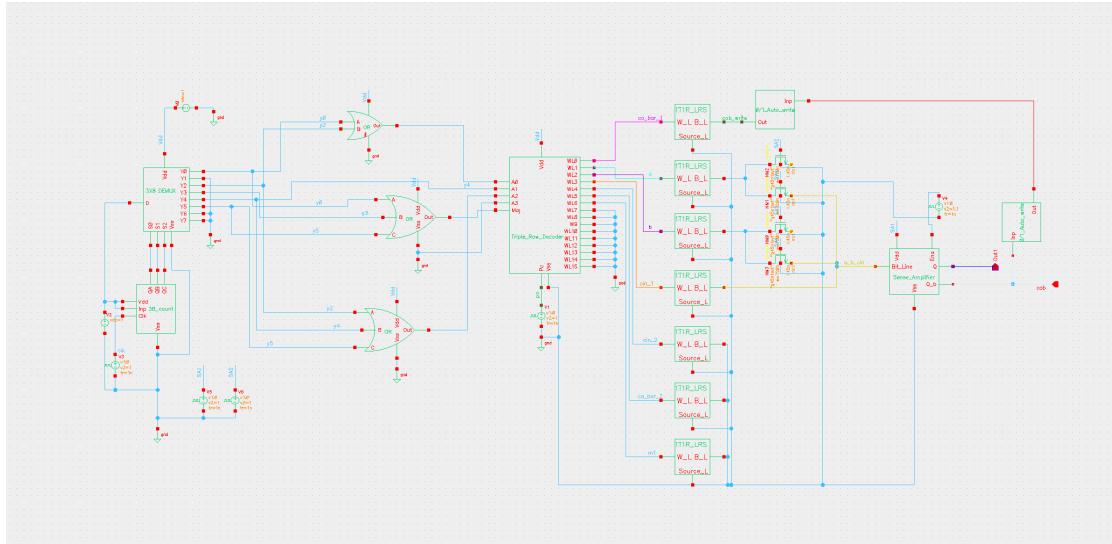


Figure 4.5: Circuit for Test 2: Testing of correctly sensing a,b,cin, and writing 1.3V in cob.

4.1.8 Waveform for Test 2

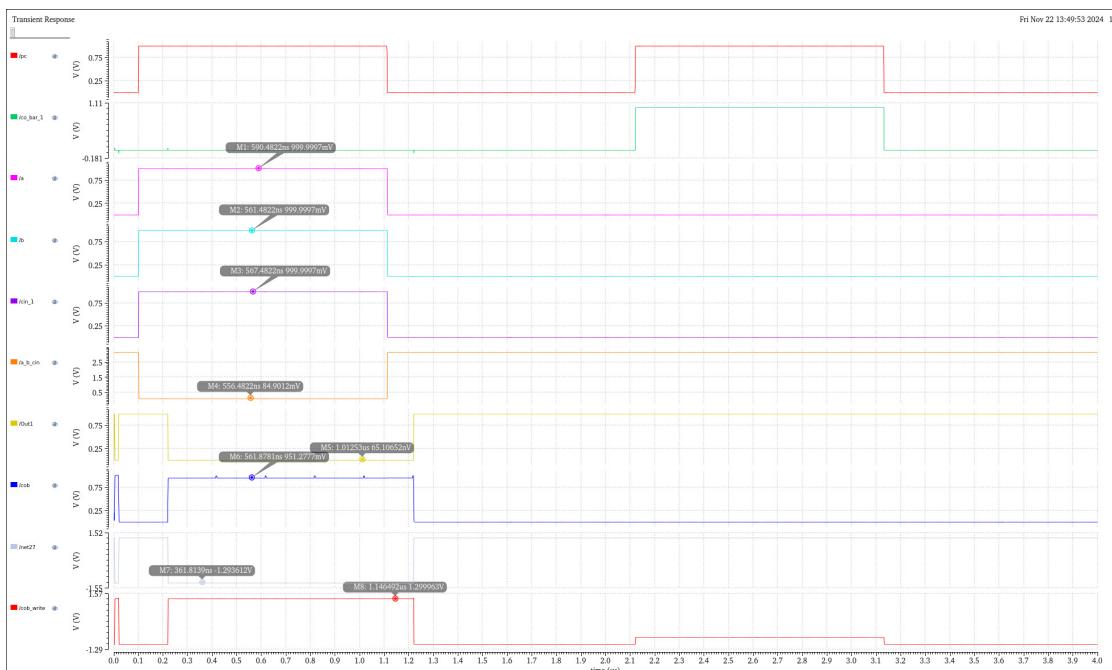


Figure 4.6: Waveform for Test 2: Testing of correctly sensing a,b,cin, and writing 1.3V in cob.

4.1.9 Results Table for Test 2

- **Cycle 1:** Cells for a , b , and cin (Cell-1, Cell-2, and Cell-3) are sensed, and $Cell-0$ (cob) is written back with 1.3V in the same cycle.
- If $cob = 1$, you write 1.3V into cob . If $cob = 0$, you write -1.3V into cob .
- I am successfully doing that with the help of two inverters in series.
- **Cycle 2:** $cob\ 0$ became high but 1.3V at bit-line reached earlier only in cycle 1.
- **Cycle 3:** $cob\ 5$ became high but 1.3V at bit-line reached earlier only in cycle 1.
- This is the problem associated that as soon as the Sense Amplifier is sensing the output, our circuit is sending 1.3/-1.3V at cob in same cycle.
- Key Point: We are able to send 1.3/-1.3V at cob bit-line.

4.1.10 Circuit Diagram for Test 3

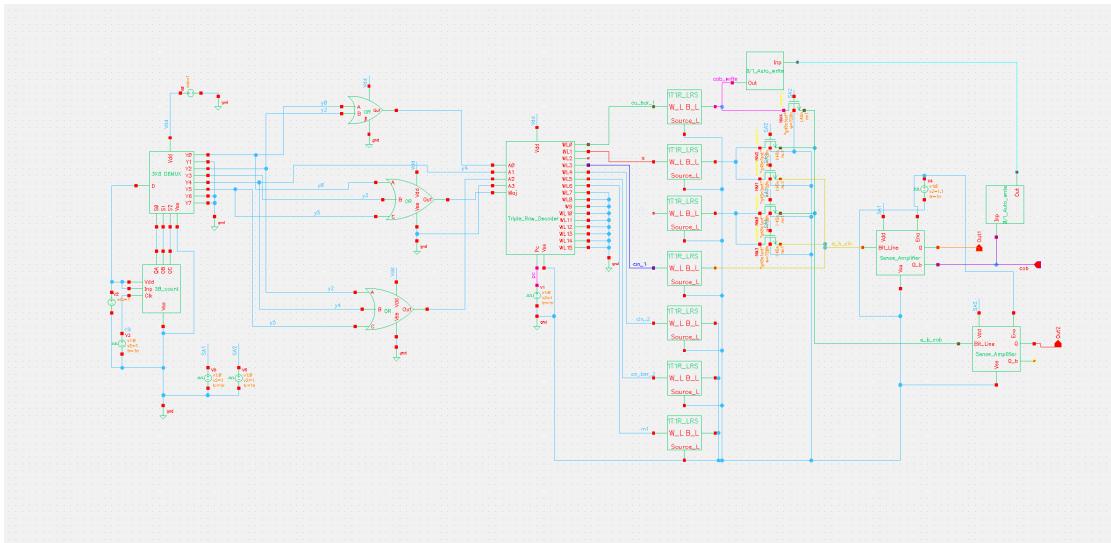


Figure 4.7: Circuit for Test 3: Testing of sensing a, b, cin , writing 1.3V in cob and again sensing cob, a, b .

4.1.11 Waveform for Test 3

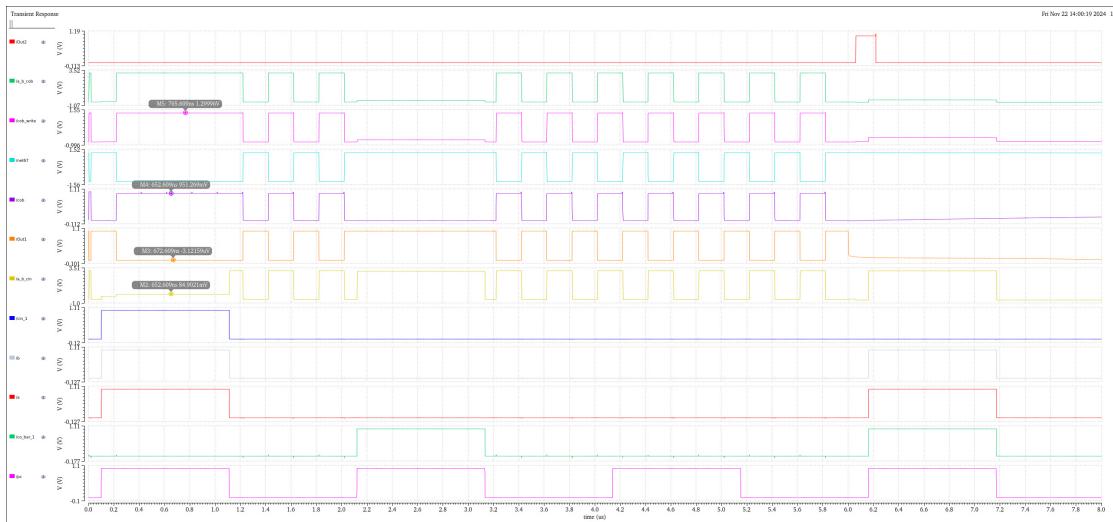


Figure 4.8: Waveform for Test 3: Testing of sensing a, b, cin , writing 1.3V in cob and again sensing cob, a, b .

4.1.12 Results Table for Test 3

- **Cycle 1:** Cells for a , b , and cin (Cell-1, Cell-2, and Cell-3) are sensed, and *Cell-0* (cob) is written back with 1.3V in the same cycle.
- If $cob = 1$, you write 1.3V into *cob*. If $cob = 0$, you write -1.3V into *cob*.
- I am successfully doing that with the help of two inverters in series in test-2 already.
- **Cycle 2:** *cob 0* became high but 1.3V at bit-line reached earlier only in cycle 1.
- **Cycle 4:** We are trying to sense a , b , and *cob* but *cob* is not written properly.
- This is the problem associated that as soon as the Sense Amplifier is sensing the output, our circuit is sending 1.3/-1.3V at *cob* in same cycle.
- Key Point: We need to synchronize the writing of *cob* in cycle 2 and cycle 3 so that in cycle 4 we can read them to generate m1.

4.1.13 Circuit Diagram for Test 4

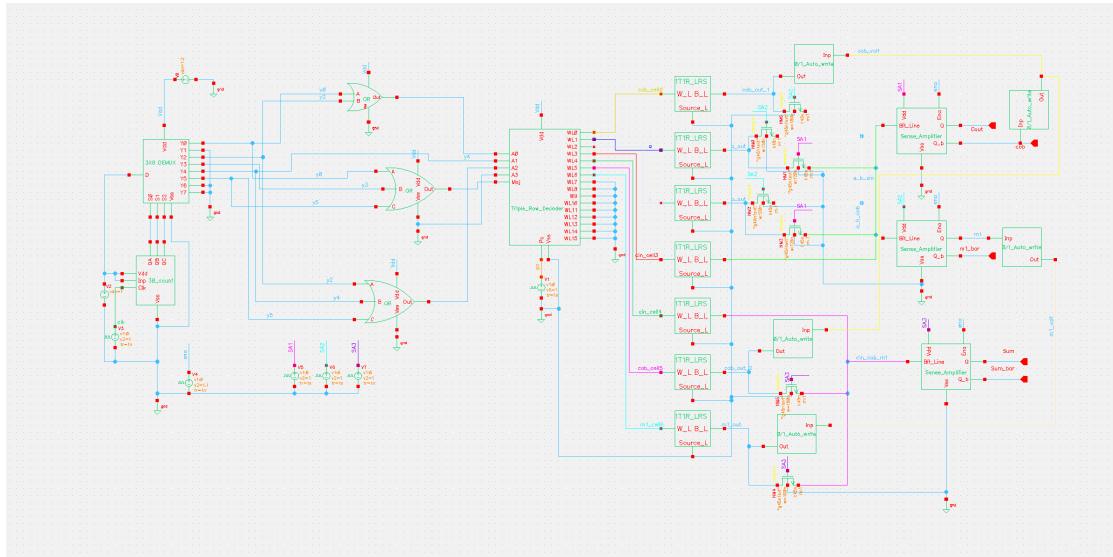


Figure 4.9: Circuit for Test 4: Testing of writing 1.3V in cob with 2nd Clock Pulse.

4.1.14 Waveform for Test 4

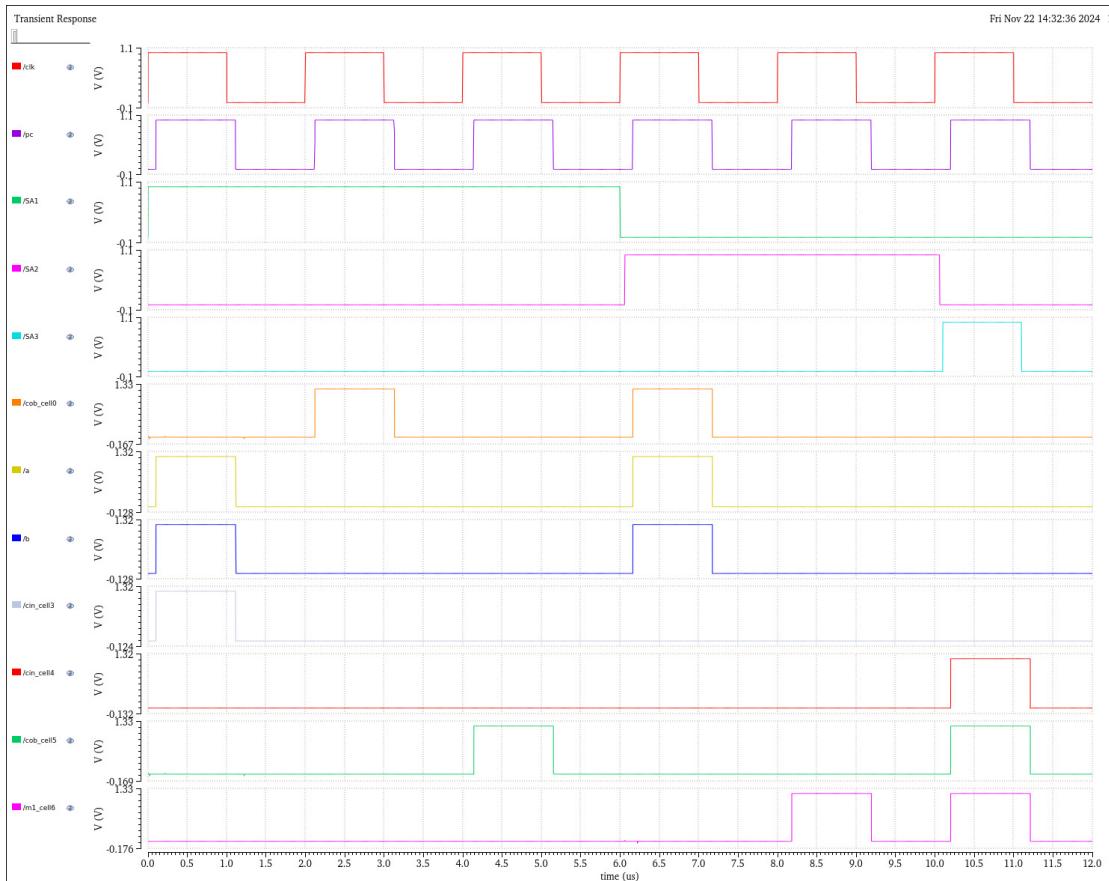


Figure 4.10: Waveform for Test 4: Testing of writing 1.3V in cob with 2nd Clock Pulse.

4.1.15 Results Table for Test 4

- This is the complete Circuit which should work as automated Full Adder using ReRAM Cells.
 - Individually, we are able to turn ON the respective word-line in different cycles.
 - Individually, we are able to sense a,b,cin or cob,a,b before the writing of cob.
 - Individually, we are able to do the writing of cob but in cycle 1 only and that is the problem.
 - **If we can do the writing in the correct cycle then the circuit will work properly and we will get Cout and Sum for any set of inputs.**

4.1.16 Circuit Diagram for Test 5

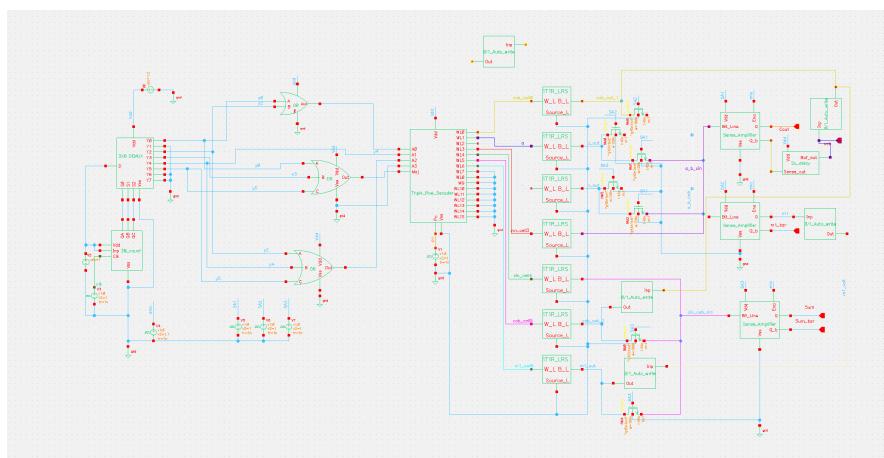


Figure 4.11: Circuit for Test 5: Testing of writing 1.3V in cob with 2nd Clock Pulse with a delay element.

4.1.17 Waveform for Test 5

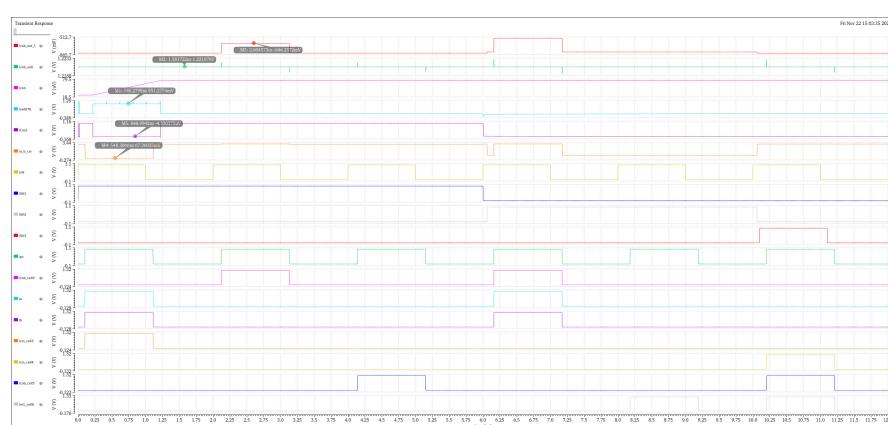


Figure 4.12: Waveform for Test 5: Testing of writing 1.3V in cob with 2nd Clock Pulse with a delay element.

4.1.18 Results Table for Test 5

- If $cob = 1$, you write 1.3V into cob . If $cob = 0$, you write -1.3V into cob .
- I am successfully doing that with the help of two inverters in series in test-2 already.
- I tried to add a buffer with 2-microsecond delay, which is the duration of one clock pulse at the output of Sense Amplifier.
- I added a RC circuit at the end of buffer but not getting the correct response.
- **If we can do the writing in the correct cycle then the circuit will work properly and we will get Cout and Sum for any set of inputs.**

4.2 Problems

4.2.1 Problem 1: Synchronization issues in cob writing:

- Writing 1.3V/-1.3V to cob overlaps with Sense Amplifier READ operations in cycle 1.
- Output sensed incorrectly due to timing mismatches. I need this writing voltage at bit-line of cob in cycle 2.

4.2.2 Problem 2: Lack of proper delay in circuit operations:

- Bit-line voltage for cob reaches prematurely in cycle 1 only, causing data corruption in intermediate steps.
- I need to send the output of sense amplifier cob in cycle 2 and not in cycle 1 with the help of proper delay at output of sense amplifier.

4.3 Conclusion

4.3.1 Conclusion 1: Automated full adder design using ReRAM cells is feasible and effective for in-memory computation.

4.3.2 Conclusion 2: Synchronization between WRITE and READ operations is critical:

- Introduce delay elements to separate Sense Amplifier READ and WRITE operations.
- Ensure precise clock pulse management for stable operation.

4.3.3 Conclusion 3: Counter-driven automation reduces the need for manual cycle management:

- Each cycle automatically controls corresponding word-lines and operations.

4.3.4 Conclusion 4: Improvements needed for real-time scalability:

- Address timing mismatches and voltage control for large-scale circuits.

CHAPTER 5

Future Scope

5.1 Future Work

5.1.1 Scalability:

- Extend the automated full adder design to multi-bit binary addition.
- Design larger automated arithmetic units like multipliers and accumulators using ReRAM arrays.

5.1.2 Integration:

- Develop hybrid architectures combining ReRAM with other emerging memory technologies.
- Explore integration with machine learning accelerators for in-memory AI computations.

5.1.3 Applications:

- Implement advanced digital signal processing tasks like Fourier Transforms and convolution after completing Hadamard Transform.
- Explore real-world use cases such as cryptography, image processing, and neural network inference.

REFERENCES

- [1] V. Lakshmi, J. Reuben, and V. Pudi, "A Novel In-Memory Wallace Tree Multiplier Architecture Using Majority Logic," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 69, no. 3, pp. 1148-1158, March 2022. DOI: <https://doi.org/10.1109/TCSI.2021.3129827>
- [2] J. Reuben and S. Pechmann, "Accelerated Addition in Resistive RAM Array Using Parallel-Friendly Majority Gates," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 29, no. 6, pp. 1108-1121, June 2021. DOI: <https://doi.org/10.1109/TVLSI.2021.3068470>
- [3] V. Lakshmi, V. Pudi, and J. Reuben, "Inner Product Computation In-Memory Using Distributed Arithmetic," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 69, no. 11, pp. 4546-4557, Nov. 2022. DOI: <https://doi.org/10.1109/TCSI.2022.3193678>
- [4] J. R. Reuben, D. Fey, and C. Wenger, "A Modeling Methodology for Resistive RAM Based on Stanford-PKU Model With Extended Multilevel Capability," *IEEE Transactions on Nanotechnology*, vol. 18, pp. 647–656, 2019. DOI: <https://doi.org/10.1109/TNANO.2019.2922838>
- [5] Q.-K. Trinh, S. Ruocco, and M. Alioto, "Time-Based Sensing for Reference-Less and Robust Read in STT-MRAM Memories," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 65, no. 10, pp. 3338-3348, Oct. 2018. DOI: <https://doi.org/10.1109/TCSI.2018.2828611>
- [6] V. Lakshmi, J. Reuben, and V. Pudi, "Majority Logic Based In-Memory Comparator," in *2022 IEEE International Conference on Semiconductor Electronics (ICSE)*, Kuala Lumpur, Malaysia, pp. 105-108, 2022. DOI: <https://doi.org/10.1109/ICSE56004.2022.9863155>
- [7] J. Reuben, "Binary Addition in Resistance Switching Memory Array by Sensing Majority," *Micromachines*, vol. 11, no. 5, p. 496, 2020. DOI: <https://doi.org/10.3390/mi11050496>
- [8] Z.-R. Wang *et al.*, "Functionally Complete Boolean Logic in 1T1R Resistive Random Access Memory," *IEEE Electron Device Letters*, vol. 38, no. 2, pp. 179-182, Feb. 2017. DOI: <https://doi.org/10.1109/LED.2016.2645946>

ORIGINALITY REPORT



PRIMARY SOURCES

1	Submitted to American University of Beirut Student Paper	1%
2	www.freepatentsonline.com Internet Source	1%
3	Submitted to IIT Delhi Student Paper	1%
4	Submitted to Indian Institute of Technology Tirupati Student Paper	1%
5	opus4.kobv.de Internet Source	<1%
6	Submitted to Khalifa University of Science Technology and Research Student Paper	<1%
7	Submitted to Hong Kong University of Science and Technology Student Paper	<1%
8	1library.co Internet Source	<1%

- 9 "Handbook of Memristor Networks", Springer Science and Business Media LLC, 2019 **<1 %**
Publication
-
- 10 Simon Deleonibus. "Emerging Devices for Low-Power and High-Performance Nanosystems - Physics, Novel Functions, and Data Processing", CRC Press, 2018 **<1 %**
Publication
-
- 11 [fdocuments.net](#) **<1 %**
Internet Source
-
- 12 [www.diplomarbeiten24.de](#) **<1 %**
Internet Source
-
- 13 Bai, Yue, Huaqiang Wu, Riga Wu, Ye Zhang, Ning Deng, Zhiping Yu, and He Qian. "Study of Multi-level Characteristics for 3D Vertical Resistive Switching Memory", Scientific Reports, 2014. **<1 %**
Publication
-
- 14 [erepository.uonbi.ac.ke:8080](#) **<1 %**
Internet Source
-
- 15 Jeong, Doo Seok, Reji Thomas, R S Katiyar, J F Scott, H Kohlstedt, A Petraru, and Cheol Seong Hwang. "Emerging memories: resistive switching mechanisms and current status", Reports on Progress in Physics, 2012. **<1 %**
Publication
-

- 16 Submitted to Rochester Institute of Technology **<1 %**
Student Paper
-
- 17 Submitted to Sim University **<1 %**
Student Paper
-
- 18 J.-L. Auriault, C. Geindreau, P. Royer, J.-F. Bloch, C. Boutin, J. Lewandowska. "Poromechanics II", A.A.Balkema Publishers Lisse, 2020 **<1 %**
Publication
-
- 19 Submitted to Liverpool John Moores University **<1 %**
Student Paper
-
- 20 Vijaya Lakshmi, John Reuben, Vikramkumar Pudi. "A Novel In-Memory Wallace Tree Multiplier Architecture Using Majority Logic", IEEE Transactions on Circuits and Systems I: Regular Papers, 2022 **<1 %**
Publication
-
- 21 course.ece.cmu.edu **<1 %**
Internet Source
-
- 22 Submitted to urjc **<1 %**
Student Paper
-
- 23 Vijaya Lakshmi, John Reuben, Vikramkumar Pudi. "Majority Logic Based In-Memory Comparator", 2022 IEEE International **<1 %**

Conference on Semiconductor Electronics (ICSE), 2022

Publication

-
- 24 adoc.pub <1 %
Internet Source
-
- 25 www.researchgate.net <1 %
Internet Source
-
- 26 Gan Fuxi, Wang Yang. "Data Storage at the Nanoscale - Advances and Applications", Pan Stanford, 2019 <1 %
Publication
-
- 27 M Scherff, B Meyer, J Hoffmann, C Jooss, M Feuchter, M Kamlah. " Pulse length and amplitude dependent resistive switching mechanisms in Pt-Pr Ca MnO -Pt sandwich structures ", New Journal of Physics, 2015 <1 %
Publication
-
- 28 Run Ni, Ling Yang, Xiao-Di Huang, Sheng-Guang Ren, Tian-Qing Wan, Yi Li, Xiang-Shui Miao. "Controlled Majority-Inverter Graph Logic With Highly Nonlinear, Self-Rectifying Memristor", IEEE Transactions on Electron Devices, 2021 <1 %
Publication
-
- 29 arxiv.org <1 %
Internet Source
-
- soar.suny.edu

30	Internet Source	<1 %
31	vdoc.pub Internet Source	<1 %
32	Alex James, Alon Ascoli, Bhaskar Choubey. "Intelligence in Chip: Integrated Sensors and Memristive Computing", River Publishers, 2024 Publication	<1 %
33	Nima Eslami, Mohammad Hossein Moaiyeri. "A Flexible and Reliable RRAM-Based In-Memory Computing Architecture For Data-Intensive Applications", IEEE Transactions on Emerging Topics in Computing, 2023 Publication	<1 %
34	dokumen.pub Internet Source	<1 %
35	lucris.lub.lu.se Internet Source	<1 %
36	patents.justia.com Internet Source	<1 %
37	researchonline.ljmu.ac.uk Internet Source	<1 %
38	www.nature.com Internet Source	<1 %

39

John Reuben. "Binary Addition in Resistance
Switching Memory Array by Sensing
Majority", *Micromachines*, 2020

<1 %

Publication

Exclude quotes On

Exclude matches < 7 words

Exclude bibliography On