Design of RRAM-Based Integrate and Fire Neuron and Programmable Synapse for Neuromorphic Computing

A

Thesis submitted

for the award of the degree of

DOCTOR OF PHILOSOPHY

By

ASHVINIKUMAR PRUTHVIRAJ DONGRE

DEPARTMENT OF ELECTRONICS AND ELECTRICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY GUWAHATI

GUWAHATI - 781 039, ASSAM, INDIA

April 2023

TH-3339_186102005

TH-3339_186102005

Certificate

This is to certify that the thesis entitled "**RRAM Based Integrate and Fire Neuron and Reprogrammable Synapse for Neuromorphic Computing**", submitted by **ASHVINIKUMAR PRUTHVIRAJ DONGRE** (186102005), a research scholar in the *Department of Electronics and Electrical Engineering, Indian Institute of Technology Guwahati*, for the award of the degree of **Doctor of Philosophy**, is a record of an original research work carried out by him under my supervision and guidance. The thesis has fulfilled all requirements as per the regulations of the institute and in my opinion has reached the standard needed for submission. The results embodied in this thesis have not been submitted to any other University or Institute for the award of any degree or diploma.

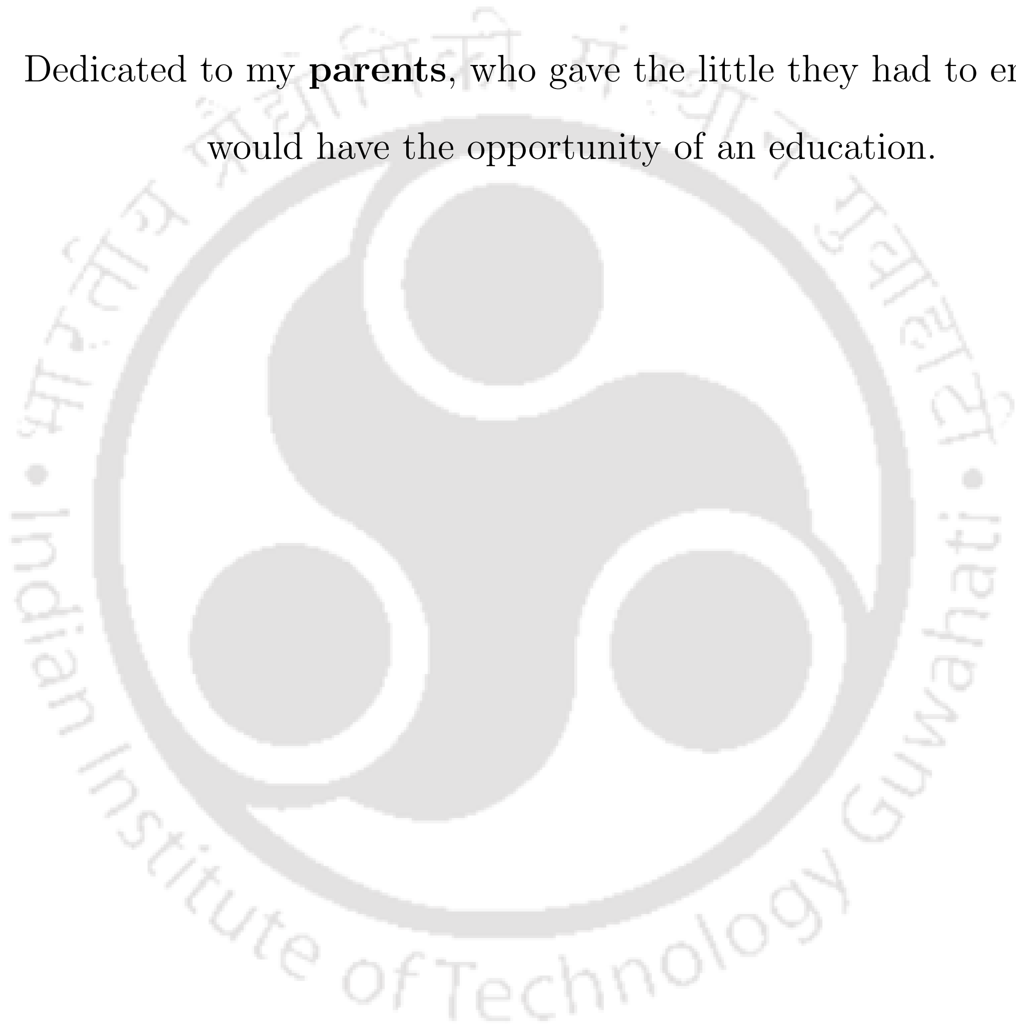
Dated:

Guwahati.

Dr. Gaurav Trivedi
Associate Professor
Dept. of Electronics and Electrical Engg.
Indian Institute of Technology Guwahati
Guwahati - 781 039, Assam, India.

TH-3339_186102005

TH-3339_186102005



Dedicated to my **parents**, who gave the little they had to ensure I would have the opportunity of an education.

TH-3339_186102005

TH-3339_186102005

Acknowledgements

A meager expression conceals a major part of my cardiac gratitude and indebtedness for my research supervisor, Dr. Gaurav Trivedi, a distinguished conceptualist and prolific trailblazer, not out of sheer imposition for mere duty's sake, but for his heraldic highlighting, extolling suggestions, constant guidance, timely advice, scrupulous scrutiny, and explicit encouragement.

I want to extend my profound sense of gratitude to Prof. Harshal B. Nemade, Dr. Hanumant Singh Shekhawat, and Dr. Aryabartta Sahu for their thorough inspiration and valuable suggestions. I am also grateful to faculty members and the office staff of the Department of Electronics and Electrical Engineering, IIT Guwahati, for their help in carrying out this research work. The work has been partially supported by the Ministry of Electronics and Information Technology's (MeitY) sponsored project Electronics and ICT Academy at IIT Guwahati. I acknowledge the infrastructure support provided by the Academy, which helped in the successful and timely conclusion of the proposed work. I sincerely thank all my colleagues and friends for their support and sympathetic encouragement.

I attribute this achievement to my father, mother, sister, and brother for their constant support and silent prayers for my success and for making me stand in this position.

ASHVINIKUMAR PRUTHIVRAJ DONGRE

TH-3339_186102005

TH-3339_186102005

Abstract

A human brain can perform compute-intensive tasks, such as multi-object recognition, reasoning, and decision-making, consuming only 20 W power. Whereas, to recognize 1000 different objects, a CPU consumes around 250 W power. Around 10^{11} neurons in the human brain are interconnected through approximately 10^{15} synapses responsible for the brain's exceptional computing capacity. The advancements in processing technology have reduced the technology nodes drastically, which further reduced the power consumption of the processors; still, they cannot match the low power consumption of the human brain. Even with the latest technological advancements, optimizing the processors with Von Neumann architectures for speed and power becomes challenging because of the memory bottleneck.

The root cause of the memory bottleneck in a processor is the separation of memory and processing units. Even though the processors can be designed to be superfast, the applications that run on these processors, especially the artificial intelligence applications, need large amounts of data to be transferred from memory to the processing unit. A simple matrix multiplication involves multiple reading, processing, and writing operations. It worsens further with an increase in the size of the matrix.

This motivated researchers to explore other paradigms, such as in-memory and near-memory computing, where computations are performed in the system's memory. Such modifications at the architectural level have improved the performance of the processing units. However, the ever-increasing demands of AI applications have forced researchers to look deep into the brain's functioning to optimize area, speed, and power. The human brain does not have separate memory and processing units. Therefore, it does not require any read or write cycles. This evolved a new neuromorphic computing era, in which brain's critical algorithmic and computational features are emulated in silicon-based hardware to improve performance at minimal power consumption. Since neurons and synapses form

TH-3339_186102005

the basic elements of a neuromorphic architecture, it is anticipated that optimizing these elements would result in area and power-efficient large-scale neuromorphic computing architectures.

The scientific community prefers neuromorphic systems realization using digital logic. Since the number of neurons required for practical applications is large, this increases the overall power and area consumption. Implementation of neurons in the analog domain is also an attractive solution. However, the large-scale realization of such architectures becomes inefficient because of its high power consumption even when implemented on the lower technology nodes. Efforts have been made to design analog neurons using CMOS transistors, but the energy consumption remains in the range of pJs. Motivated by these facts, we propose a Resistive Random Access Memory (RRAM) based integrate and fire neuron. RRAM is employed as a voltage divider for integrate and fire ($I\&F$) operation. The proposed neuron exhibits temporal integration, triggering threshold, and refractory period similar to a biological neuron making it a suitable candidate for large-scale neuromorphic systems.

Another major component of neuromorphic systems is the synapse. RRAMs have been widely explored to design synapses. However, variability in RRAMs is a major hindrance while implementing large-scale neuromorphic architectures. Multilevel cells have been extensively explored to obtain multi-bit precision in a single cell. Although it provides considerable advantages regarding area utilization and power consumption, implementing circuits for precise programming of the resistive state is a significant challenge. Therefore, this thesis proposes an RRAM-based synaptic architecture with a continuous sensing and feedback scheme to stop RRAM programming when the required conductance is achieved.

The work proposed in this thesis demonstrates that RRAM can be efficiently employed to implement energy-efficient integrate and fire neurons. We further design an RRAM-based reprogrammable synapse. The precise RRAM programming mechanism shows that the Cycle-to-Cycle and Device-to-Device variations that are pertinent to RRAM devices can be resolved effectively using circuit-level techniques. Finally, we implement a spiking neural network to evaluate the performance of the proposed integrate and fire neuron and programmable synapse.

TH-3339_186102005

Contents

List of Figures xv

List of Tables xxi

List of Acronyms xxiii

List of Symbols xxv

1 Introduction 1

1.1 Beyond Von Neumann architecture . . . 2

1.1.1 Neuromorphic computing . . . 3

1.1.2 Neuromorphic processors . . . 4

1.2 Neuron . . . 7

1.2.1 Biological neuron . . . 7

1.2.1.1 Initiation of the action potential . . . 9

1.2.1.2 The Hodgkin Huxley model . . . 9

1.2.2 Digital neuron . . . 10

1.2.3 Analog neuron . . . 11

1.2.4 Hybrid neuron . . . 12

1.2.5 Digital Vs Analog neuron . . . 13

1.3 Synapse . . . 13

1.3.1 Biological synapse . . . 14

1.3.1.1 Dynamics of a synapse . . . 15

1.3.2 CMOS synaptic circuits . . . 16

1.3.3 Non-volatile memories as synapse . . . 17

1.4 Emerging non-volatile memories . . . 18

1.4.1 Resistive Random Access Memory (RRAM) . . . 21

TH-3339_186102005

1.5 RRAM based synaptic architectures . . . 22
1.6 Artificial Neural Networks . . . 23
1.7 Spiking Neural Network . . . 25
1.8 Motivations and Contributions of the Thesis . . . 27
1.9 Organization of the Thesis . . . 28

2 Resistive Random Access Memory **31**
2.1 Introduction . . . 32
2.2 RRAM circuit simulation models . . . 33
2.3 Peking RRAM Verilog-A model . . . 35
2.3.1 Device structure . . . 35
2.3.2 Transport mechanism . . . 36
2.3.3 Model equations . . . 36
2.3.4 Simulation results . . . 38
2.3.5 Effects of individual parameter on $I-V$ characteristics . . . 39
2.4 Unimore RRAM Verilog-A model . . . 45
2.4.1 Device structure . . . 45
2.4.2 Transport mechanism . . . 45
2.4.3 Model equations . . . 46
2.4.4 Equation modeling the RTN and variation . . . 47
2.4.4.1 Simulation results . . . 50
2.4.5 Effects of individual parameter on $I-V$ characteristics . . . 50
2.5 Parallel and series combination of RRAM . . . 55
2.5.1 Parallel combination . . . 55
2.5.2 Series combination . . . 56
2.6 Cycle-to-Cycle and Device-to-Device variation . . . 58
2.7 Summary . . . 59

3 RRAM Based Integrate and Fire Neuron **61**
3.1 Introduction . . . 62
3.2 $I\&F$ operation using RRAM . . . 62
3.3 RRAM connected in series and opposite direction . . . 66

TH-3339_186102005

3.4 Digital reset control with Pulse propagation 66
3.5 Proposed I&F neuron with reset circuit 67
3.5.1 Analog pulse propagation and RRAM reset block 69
3.6 Behavioural analysis of the proposed neuron 73
3.7 Variation and Stability analysis 74
3.8 Benchmarking with state of the art $I\&F$ neuron circuits 76
3.9 Summary 77

4 RRAM based 4-bit/cell Synapse 79

4.1 Introduction 80
4.2 $4T-1R$ structure for SET/RESET operation 80
4.3 Synaptic architecture 82
4.4 CMOS peripheral circuits 85
4.4.1 Reference voltage generator 85
4.4.2 Comparator and stop logic circuit 87
4.4.3 RRAM current non-linearity 91
4.5 Variation analysis 92
4.5.1 CMOS circuit variation analysis 92
4.5.2 RRAM variation and Noise analysis 94
4.6 Power, Latency, Energy and Area estimation 96
4.7 Comparison with the contemporary architectures 98
4.8 Summary 99

5 Spiking Neural Network 101

5.1 Introduction 102
5.2 SNN Training 102
5.3 Low precision weight encoding 105
5.4 Hardware network architecture 107
5.5 Benchmarking with Non-volatile memory-based SNN 109
5.6 Summary 111

6 Conclusion and Future Work 113

6.1 Conclusion 114

TH-3339_186102005

6.2 Directions for future work 115

Bibliography **117**

List of Publications **127**

TH-3339_186102005

List of Figures

1.1 Memory Bottleneck 3
1.2 Conceptual view of a classical neuron 8
1.3 Cell membrane potential 8
1.4 Response of the axon to simulation by 2-milisecond current pulse of increasing magnitude 9
1.5 Equivalent circuit for cell membrane 10
1.6 Area and power comparison of a million neuron learning system 13
1.7 Biological synapse 14
1.8 Generic neural network connectivity between neurons through a synapse 15
1.9 (a) Timing diagram for potentiation and depression (b) Curve for spike time dependent plasticity 16
1.10 Non-volatile memories as synapse 18
1.11 Classification of semiconductor memories 19
1.12 Classification of Resistive Random Access Memories 21
1.13 Artificial Neural Network 24
1.14 Fully connected neural network 25
1.15 Spiking Neural Network 26

2.1 Schematic of conductive filament evolution 36
2.2 Physical process of resistive switching used in the model 37
2.3 RRAM $I-V$ characteristics 38
2.4 $I-V$ Characteristics of RRAM for varying L_0 40
2.5 $I-V$ Characteristics of RRAM for varying WCF 40
2.6 $I-V$ Characteristics of RRAM for varying E_a 41
2.7 $I-V$ Characteristics of RRAM for varying X_t 41

TH-3339_186102005

2.8 $I-V$ Characteristics of RRAM for varying E_i . . . 42
2.9 $I-V$ Characteristics of RRAM for varying I_0 . . . 42
2.10 $I-V$ Characteristics of RRAM for varying a . . . 43
2.11 $I-V$ Characteristics of RRAM for varying f . . . 43
2.12 $I-V$ Characteristics of RRAM for varying V_t . . . 44
2.13 $I-V$ Characteristics of RRAM for varying R_{th} . . . 44
2.14 Device structure . . . 45
2.15 Evolution of oxygen ions (blue spheres) and vacancies (red spheres) during SET and RESET . . . 46
2.16 $I-V$ Characteristics of Unimore Verilog-A model . . . 50
2.17 $I-V$ Characteristics of RRAM for varying a . . . 50
2.18 $I-V$ Characteristics of RRAM for varying b . . . 51
2.19 $I-V$ Characteristics of RRAM for varying E_{ad} . . . 51
2.20 $I-V$ Characteristics of RRAM for varying g . . . 52
2.21 $I-V$ Characteristics of RRAM for varying E_{ag} . . . 52
2.22 $I-V$ Characteristics of RRAM for varying gg . . . 53
2.23 $I-V$ Characteristics of RRAM for varying c_0 . . . 53
2.24 $I-V$ Characteristics of RRAM for varying V_0 . . . 54
2.25 $I-V$ Characteristics of RRAM for varying k_{bar} . . . 54
2.26 $I-V$ Characteristics of RRAM for varying K_{cf} . . . 55
2.27 (a) RRAM connected in parallel combination (b)RRAM parallel connection obtained $I-V$ characteristics . . . 56
2.28 RRAM parallel connection expected $I-V$ characteristics . . . 56
2.29 (a) RRAM connected in series combination (b) RRAM series connection obtained $I-V$ characteristics . . . 57
2.30 RRAM series connection expected $I-V$ characteristics . . . 57
2.31 (a) RRAM connected in series combination in opposite direction (b) RRAM series connection obtained $I-V$ characteristics . . . 58
2.32 RRAM series connection expected $I-V$ characteristics . . . 58
2.33 Switching voltage variability for 20 consecutive SET/RESET cycle . . . 58

TH-3339_186102005

2.34 Mean and standard deviation for cycle-to-cycle variation in (a) HRS and (b) LRS . . 59
2.35 Mean and standard deviation for device-to-device variation in (a) HRS and (b) LRS . 59

3.1 RRAM SET voltages at different V_{tb} 63
3.2 Current through RRAM for pulse input with different amplitude 63
3.3 Temperature effects on RRAM switching 64
3.4 (a) RRAM connected in series and opposite direction to generate the required sudden change in voltage (b) Voltage V_s across $RRAM2$, when $RRAM1$ and $RRAM2$ are connected in opposite direction in series 66
3.5 $I\&F$ Neuron with digital pulse propagation and RRAM reset block 67
3.6 Proposed neuron circuit with integrated RRAM reset circuit 68
3.7 Schematic for negative pulse generator ($Reset_En = 0V$) 70
3.8 Schematic for negative pulse generator ($Reset_En = 0.8V$) 71
3.9 Output for negative pulse generator 71
3.10 Layout of pulse propagation and RRAM reset block 72
3.11 (a) Output for digital reset control and analog reset control (b) Firing frequency with respect to initial conducting filament length and amplitude of input pulse. 72
3.12 Firing frequency for different input pulse voltage 73
3.13 Firing frequency for different input pulse width 74
3.14 Firing frequency with respect to initial conducting filament length and amplitude of input pulse. 74
3.15 The distribution of resistance of RRAM for different number of pulses 75
3.16 Neuron spiking in presence of 20% variation in resistance 75
3.17 Neuron spiking in presence of 20% variation in switching voltage 76
3.18 Neuron spiking in corner cases 76
3.19 (a) Gaussian white noise current profile (b) Neuron spiking in presence of noise . . . 76

4.1 (a) Schematic for $4T-1R$ structure (b) SET process (c) RESET process 80
4.2 Arrangement of $4T-1R$ structure in array 81
4.3 RRAM programming to SET and RESET 82
4.4 Synaptic architecture 83

TH-3339_186102005

4.5 RESET stop block for a single RRAM cell 84
4.6 RESET operation for a selected RRAM cell 84
4.7 Precise *stop_update* pulse generation for all 16 states 85
4.8 Reference voltage generator circuit 86
4.9 Output of reference voltage generator circuit 87
4.10 Power dissipation in reference voltage generator circuit 88
4.11 Schematic for comparator circuit 88
4.12 Power dissipation in comparator circuit 89
4.13 Schematic for stop logic circuit 90
4.14 Power dissipation in stop logic circuit 90
4.15 Digital output Vs RRAM current (a) Reduced non-linearity for selected range of resistance value (b) Non-linearity for full range of resistance value (HRS-LRS). 91
4.16 Monte Carlo simulation for reference voltage generator (a) 0000 (b) 0001 (c) 0010 (d) 0011 92
4.17 Monte Carlo simulation for reference voltage generator (a) 0100 (b) 0101 (c) 0110 (d) 0111 92
4.18 Monte Carlo simulation for reference voltage generator (a) 1000 (b) 1001 (c) 1010 (d) 1011 93
4.19 Monte Carlo simulation for reference voltage generator (a) 1100 (b) 1101 (c) 1110 (d) 1111 93
4.20 Output of stop logic circuit at process corners 93
4.21 Resistance distribution for LRS and HRS 94
4.22 Difference in V_{be} and *stop_update* signal timing due to (a)cycle-to-cycle variation (b) device-to-device variation 95
4.23 RTN noise current profile on RRAM 96
4.24 Power dissipation in RRAM array 97
4.25 Layout for the CMOS part of the circuit. 99

5.1 Spiking Neural Network 103
5.2 Distribution of trained weights 106
5.3 AMS architecture for Spiking Neural Network 107

TH-3339_186102005

5.4 Reconstructed weights for corresponding output neurons 110

TH-3339_186102005

TH-3339_186102005

List of Tables

1.1 Brain Vs CPU 2

1.2 Comparison of Neuromorphic Processors 6

1.3 Concentration of ions inside neural processes and in the extracellular fluid 8

1.4 Comparison between the current and emerging memory technologies 20

1.5 Switching modes for various metal-oxide based RRAM 22

2.1 Comparison of various models 35

2.2 Simulation parameters for Peking RRAM Verilog-A model 39

2.3 RRAM model parameter 48

2.4 RTN Parameters 49

3.1 Parameters used in Verilog-A model 65

3.2 Calculations for $P1$, $P2$ and $P3$ 65

3.3 $I\&F$ neuron circuit parameters 69

3.4 Comparison of energy per spike, frequency and need for external reset circuit of proposed I&F Neuron and published I&F Neurons 77

4.1 Resistive state and its corresponding Vbe for 4-bit precision 83

4.2 Reference voltage generator circuit parameters 87

4.3 Comparator circuit parameters 89

4.4 Stop Logic circuit parameters 91

4.5 Variation in the reference voltage 93

4.6 Parameters for the process variation 94

4.7 Percentage Variation in HRS and LRS 95

4.8 Comparison of RTN 96

TH-3339_186102005

4.9 Comparison with contemporary architectures 99

5.1 Parameters for SNN 104
5.2 Normalised weight 106
5.3 Confusion matrix 110
5.4 Benchmarking with contemporary SNN implementations 111

Indian Institute of Technology Guwahati

TH-3339_186102005

List of Acronyms

AI	Artificial Intelligence
SRAM	Static Random Access Memory
RRAM	Resistive Random Access memory
FeRAM	Ferroelectric Random Access Memory
STT-MRAM	Spin-Transfer Torque Magnetic Random Access Memory
PCRAM	Phase Change Random Access Memory
HRS	High Resistance State
LRS	Low Resistance State
LTP	Long Term Potentiation
LTD	Long Term Potentiation
STP	Short Term Potentiation
STD	Short Term Depression
STDP	Spike Time Dependent Plasticity
SNN	Spiking Neural Network
ANN	Artificial Neural Network
RTN	Random Telegraph Noise
CF	Conducting Filament
FPGA	Field Programmable Gate Array
VLSI	Very Large Scale Integration
$I\&F$	Integrate and Fire
CPU	Central Processing Unit
GPU	Graphical Processing Unit
MNIST	Modified National Institute of Standards and Technology database

TH-3339_186102005

TH-3339_186102005

List of Symbols

$K+$	Potassium Ion
$Na+$	Sodium Ion
$Cl-$	Chlorine Ion
V_K	Reversal potential of potassiun ion
V_{Na}	Reversal potential of sodium ion
V_{Cl}	Reversal potential of chlorine ion
x	Barrier Thickness
t_{ox}	Maximum thickness of the conducting filament
ρ	Oxide material resistivity
t_{ox}	Oxide layer thickness
S_0	Initial conductive filament section
E_a	Activation energy of the trap assisted tunneling
T_0	Ambient temperature
l	Typical tunneling length
V_0	HRS current non-linearity factor
$alpha$	Resistivity temperature coefficient
$beta$	Barrier resistance fitting parameter
c_0	Bond vibration frequency
c_{pb}	Barrier thermal capacity
C_{pcf}	CF thermal capacity
K_{bar}	Barrier thermal conductivity
k_{cf}	CF thermal conductivity
k_{ex}	Barrier/CF mutual thermal conductivity

TH-3339_186102005

E_{ad}	Diffusion activation energy of oxygen ions
g	Field enhancement factor for oxygen ions diffusion
a	RESET curve slope fitting parameter
b	RESET curve curvature fitting parameter
E_{ag}	Bond breaking activation energy
gg	Field enhancement factor for bond breaking
$xinit$	Initial barrier thickness
$Tinit$	Initial device temperature
$Tmeas$	Temperature at which RLRS is measured
$min_time_step_vpos$	Minimum time step (for positive applied voltages)
$min_time_stop_vneg$	Minimum time step (for negative applied voltages)
$tstep_param$	Adaptive time step parameter
$const0$	Capture and emission times constant
N_c	Density of states at the bottom of the conduction band
phi	Energy barrier for injected electrons
$lambda_c$	Typical tunneling length (capture)
$lambda_e$	Typical tunneling length (emission)
$max_defects$	Maximum number of defects that can be generated
$Erel0_O$	Nominal oxygen ions relaxation energy
$Et0_O$	Nominal oxygen ions thermal ionization energy
$Erelo_V$	Nominal oxygen vacancies relaxation energy
$Et0_V$	Nominal oxygen vacancies thermal ionization energy
$Delta_Erel_O$	Spread of the oxygen ions relaxation energy distribution
$Delta_Et_O$	Spread of the oxygen ions thermal ionization energy distribution
$Delta_Erel_V$	Spread of the oxygen vacancies relaxation energy distribution
$Delta_Et_V$	Spread of the oxygen vacancies thermal ionization energy distribution
$Delta_HRS_mean$	Mean of the normal distribution associated to the log Normal distribution of the R HRS due to RTN
$Delta_HRS_std$	Standard deviation of the normal distribution associated to the logNormal distribution of the RHRS due to RTN

TH-3339_186102005

RTN_ON	Parameter used to switch on the RTN module (0=OFF; 1=ON)
$rand_seed$	Initial random seed value
$dxdt_th$	Threshold on the barrier derivative to randomly reassign defects positions

TH-3339_186102005

TH-3339_186102005

1

Introduction

Contents

1.1	Beyond Von Neumann architecture	2
1.2	Neuron	7
1.3	Synapse	13
1.4	Emerging non-volatile memories	18
1.5	RRAM based synaptic architectures	22
1.6	Artificial Neural Networks	23
1.7	Spiking Neural Network	25
1.8	Motivations and Contributions of the Thesis	27
1.9	Organization of the Thesis	28

TH-3339_186102005

1.1 Beyond Von Neumann architecture

AI applications running on conventional computing systems, such as CPUs, GPUs, and FPGAs, follow the Von Neumann architecture. A comparison between the human brain and the modern CPU is shown in Table. 1.1 illustrates a massive difference between the power consumed by the CPU. This is due to the massive parallelism and excellent coordination among millions of neurons and synapses in the human brain. The functionalities performed by a human brain parallelly are significantly higher than the CPU.

Table 1.1: Brain Vs CPU

Properties	Computer	Human Brain
Basic unit	10 Billion Transistors	100 Billion Neurons
		100 Trillion Synapse
Processing mode	Serial and parallel	Massively Parallel
Power Consumption	100 Watts	20 Watts
Input output for each unit	1-3	1000
Signalling mode	Digital	Analog

Although artificial neural networks (ANNs) mimic the human brain, their hardware implementations still need to be improved to achieve energy efficiency and accuracy similar to the human brain. Implementing machine learning algorithms on the conventional Von Neumann architectures requires enormous data transfer between the processor and the memory, resulting in high power consumption and memory bottleneck, which limit the performance of highly efficient algorithms running on the most efficient computing systems. Fig. 1.1 elaborates on the bottleneck created by data transfer between CPU or GPU and memory.

The memory bottleneck is often associated with accessing the data from off-chip memory, and it is fundamental to any architecture that separates the processing unit from the memory. Therefore, even solutions, such as distributed computing, can be helpful only to a certain limit because there is a communication cost associated because of data movement from the point of storage to the point of computation outside the memory.

Specialized hardware has evolved to resolve the memory bottleneck problem. Digital accelerators and domain-specific processors have achieved $10-1000\times$ higher energy efficiency and speed than

TH-3339_186102005

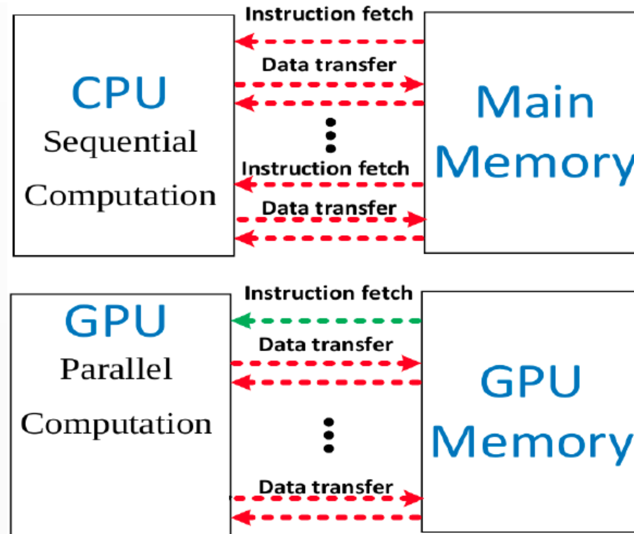


Figure 1.1: Memory Bottleneck

general-purpose processors, but efficient memory access or data movement is still an issue. The in-memory and near-memory computing provides a possible solution to this problem, but a true brain-inspired architecture can address this issue more prominently. Neuromorphic computing, also called brain-inspired computing, has shown promising results in resolving the memory bottleneck problem.

1.1.1 Neuromorphic computing

Neuromorphic computing started as a concept developed by Carver Mead in the late 1980s, describing the use of Very Large Scale Integration (VLSI) systems containing electronic analog circuits to mimic neurobiological architectures in the nervous system. It is an emerging interdisciplinary field that combines biology, physics, mathematics, computer science, and engineering to design hardware or physical models of neural and sensory systems.

Neuromorphic computing takes inspiration from the functionality of the brain. The human brain performs several impressive tasks, such as simultaneous recognition, reasoning, control, and movement, while consuming much less power than modern computers [1]. Although the brain is not yet fully explored, its remarkable capability may be attributed to three fundamental observations: extensive connectivity, structural and functional hierarchy, and time-dependent neuronal and synaptic functionality [2]. Neurons and synapses form the primary component of the brain. Neurons are the computational primitive elements that exchange or transfer information through discrete action potentials or 'spikes.' At the same time, synapses are the storage elements underlying memory and learning.

TH-3339_186102005

The human brain has a network of billions of neurons, interconnected through trillions of synapses. Spike-based temporal processing allows sparse and efficient information transfer in the brain.

Similar to the brain, neuromorphic computing systems comprise synapse and neuron circuits to emulate large-scale neural networks. Previous-generation neuromorphic systems developed using conventional design techniques could have been more area and power efficient. However, the existing neuromorphic systems designed using state-of-the-art memristive devices have shown great promise in solving major scientific problems more efficiently than conventional systems.

1.1.2 Neuromorphic processors

In the last decade, neuromorphic processors have been developed by a few industries and research institutes. Neurogrid [3], TrueNorth [4], DYNAPs [5], SpiNNaker [6], Loihi [7], and Braindrop [8] are prototypical neuromorphic processors that have recently been released. All these processors have been designed using different technologies; therefore, their working principle and capabilities vary greatly. Note that all these processors realize spiking neural networks on the hardware. All these neuromorphic processors were designed mainly for spiking neural networks that are more symmetrical and closely resemble the brain's functionality. [3], [4] initially aimed to reverse engineer the central nervous system, including the retina. Note that the spiking neural network can closely model the central nervous system and is of utmost importance for neuromorphic computing.

The digital neuromorphic prototypes, TrueNorth [4], SpiNNaker [6], and Loihi [7] exhibit the flexibility of network configuration as well as neuron model parameters and learning algorithms. The advancements in circuit fabrication techniques result in digital neuromorphic systems' high-speed and low-power operation. Field Programmable Gate Arrays (FPGAs) have also emerged as an efficient platform for implementing neuromorphic systems. To date, various architectures of synapses and spiking neurons [9], [10] and neuromorphic systems for reconfigurable spiking neural networks [11] have been realized using FPGAs.

SpiNNaker is a digital neuromorphic processor designed to simulate the spiking neural networks to meet the human brain complexity [12]. This is achieved with the help of ARM9 cores that can access the local memory of the system and shared memory across the multicore chip. The messages are processed faster as the system is optimized for small packet codes by keeping short queues. This helps SpiNNaker implement fundamental brain functionalities, such as high fan-in and fan-out connectivity, locality of information, and event-driven computation. The SpiNNaker consists of 18 ARM968

TH-3339_186102005

processors. Forty-eight chips having 864 cores are assembled on one board.

TrueNorth is a digital CMOS chip consisting of a million neurons. IBM developed TrueNorth in 2014 using 28 nm process technology [13]. It consists of arrays of 4,096 cores consisting of synapses and neurons. Each core consists of 12.75 KB of local memory to store the states and parameters of the neuron and synapse. The chip consists of 256 million synapses. Since the memory and the processing core are co-located and due to the event-driven custom design, it can perform 46 billion synaptic operations per second per watt with 26 pJ per synaptic event. The power density achieved by TrueNorth is $20mW/cm^2$, which is about 3× less than a typical CPU.

The BrainScaleS processor is a mixed-signal processor with analog circuits for neurons, synapses, and digital communication blocks [14]. A single chip consists of 512 spiking neurons and about $14,000$ synapses. The time between the pre-synaptic and post-synaptic spikes is measured with the help of dedicated sensors at the synaptic circuits. This enables learning rules that can be highly flexible to be implemented on BrainScales, which includes learning rules like reward-based learning.

Loihi is a neuromorphic processor introduced by Intel in 2018. It is developed using a 14 nm FinFET process [7]. The synapse and neurons in this processor are programmable. The chip has 128 neural cores. Each core of this processor consists of 2 MB of SRAM that stores various parameters of 1024 neurons. It is also powered by ×86 processors and 16 MB memory for synaptic operations that support a resolution of $1-9$ bits. It supports around $130,000$ neurons and 130 million synapses. Loihi supports core-to-core multicast and population-based hierarchical connectivity, enabling network connectivity similar to the biological level. The processor can implement learning rules like triplet STDP, STDP, and SRDP. The energy consumption per synaptic operation is 15 pJ while performing 30 billion synaptic operations per second (SOPS).

The NeuroGrid platform implements a large-scale neural model to implement their function in real-time [15]. Therefore, the memory and the computing resources have a time constant that matches the signals to be processed. It consists of analog-mixed signal threshold circuits to implement continuous-time neural processing elements. The functions of neurons and synapses have been efficiently implemented using the subthreshold operation of field-effect transistors. It consists of a board with 16 CMOS chips connected as a tree network. Each chip consists of a 256×256 array of neurons and 64 KB of synaptic memory. A NeuroGrid board consists of a billion synapses and around one million neurons that can efficiently model the cortical networks.

TH-3339_186102005

Table 1.2: Comparison of Neuromorphic Processors

Processor	Loihi	TrueNorth	SpiNNaker	DYNAP-SE
Developer	Intel 2021	IBM 2004	University of Manchester 2018	INI Zurich 2018
Configuration	130,000 neurons, 130 million synapses, 4096 cores	256 neurons, 268 million synapses	57,600 ARM9 processors, 1,036,800 cores and over 7 TB of RAM	1024 neurons, 64K synapses
Power consumption	23.6 pJ per synaptic operation	26 pJ per synaptic operation	100W and an air condition environment	17 pJ per synaptic operation
Advantages	Digital ASIC at 14nm FinFET	Digital ASIC at 28nm	130 nm process More flexible as inclusion of new model involves just a change of code	Mixed signal 180nm
Disadvantages	Digital implementation results in large area and power consumption	SNN emulation without on-chip learning	Energy inefficient	Area overhead

Dynamic neuromorphic asynchronous processors (DYNAP) is a mixed-signal neuromorphic processor introduced by the Institute of Neuroinformatics, Zurich. It is developed using 180 nm CMOS technology. The processor consists of $64,000$ synapses and $1,024$ neurons [16]. It consists of four cores having 256 analog neurons each. The CMOS transistors are operated in the subthreshold region to implement the neuron's temporal dynamics. Whereas to allow programmability of the network, asynchronous digital circuits are employed. The analog circuits implement a wide range of neural and synaptic features, including spike frequency adaptation that is crucial in implementing long short-term memory (LSTM)-like networks with spiking neurons. Moreover, the device variation and mismatch in CMOS analog circuits are explored to implement reservoir computing and neural sampling.

ODIN is a neuromorphic processor developed by the Catholic University of Louvain in 2019. It is fabricated in 28 nm technology. The core supports 256 programmable neurons to implement first-order LIF and second-order Izhikevich dynamics [17]. 4 KB SRAM array stores the local neuronal parameters, and the neuron logic is time multiplexed using a global controller to obtain the neuron's dynamics sequentially. The core also consists of $3-bit$ 2562 synapses implemented using 32 KB of SRAM array. The processor features on-chip learning capability, and an additional bit in each synapse is used to turn online learning on or off. Using the Modified National Institute of Standards and Technology Database (MNIST) data set, the chip demonstrated on-chip learning with an accuracy of 84.5 % while consuming 15 nJ per inference.

A comparison of various neuromorphic processors is shown in Table. 1.2. It can be observed that

TH-3339_186102005

all the processors are designed to support millions of neurons and synapses, similar to a brain. The SpiNNaker consists of 18 processor cores grouped on a chip. About 48 such chips with 864 cores are assembled on a single board. The Loihi that uses a 45 nm FinFET ASIC technology consumes 23.6 pJ of energy per synaptic operation. Whereas the TrueNorth implemented using 28nm CMOS technology consumes 26 pJ of energy per synaptic operation. The change in technology nodes can be seen in their performance. Dynap-SE used CMOS 180 nm technology node, and still, it consumes only 17 pJ of energy per synaptic operation. This is because the neurons and synaptic circuits are implemented using a mixed-signal approach. Hence, implementing low-power neurons and synapses can improve the overall performance of the system and can result in low power consumption.

An emerging strategy is to utilize non-volatile memory devices, such as phase-change memory, magnetic tunnel junction, oxide-based resistive memory, and floating-gate transistor, as synaptic devices [18]. These devices also enabled mixed analog implementation of various neuromorphic computing system building blocks, such as the synapse and the neuron. Since the number of synapses and neurons in a neuromorphic system is huge, replacing mainstream static random access memory or content-addressable memory with such non-volatile memories can remarkably improve the system's performance. The multiple states supported by these devices can further boost the system density, reducing the size of the chips by multifold [19].

1.2 Neuron

Neurons are the primary computing elements in a neuromorphic system. Researchers have been trying to mimic the essential dynamics of biological neurons using CMOS circuits. Most neuron models implemented in neuromorphic computing systems have the basic concept of accumulation of charge and fire when a threshold is reached to affect other connected neurons. Instead of mimicking all the dynamics of biological neurons, the leaky-integrate-and-fire (LIF) and integrate-and-fire ($I\&F$) neurons have sufficient dynamics to produce satisfactory results for most neuromorphic applications.

1.2.1 Biological neuron

A neuron is the fundamental anatomical unit of the nervous system. Fig.1.2 shows an artistic view of a neuron. The long and filamentary extensions of the single cell are called processes. A tree of such processes is called a dendrite. The synapses form the junction between two neurons and are responsible for information processing in the neural system. Some neurons have a specialized process

TH-3339_186102005

called an axon which helps digitize the data for local and long-distance transmission [20].

Electrical operation of a neuron: The most basic charge transfer agents in all nerve membranes are the metabolically driven pumps that actively expel sodium ions from the cytoplasm and simultaneously import potassium ions from the extracellular fluid, as depicted in Fig. 1.3. Table. 1.3 shows the concentration of ions inside and outside the cell membrane.

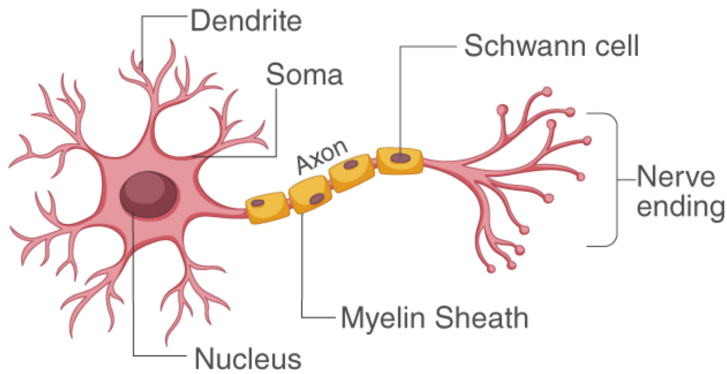


Figure 1.2: Conceptual view of a classical neuron [20]

Suppose the potential inside the cell rises above V_t. In that case, it results in a positive current flowing outwards, whereas if the potential inside the cell drops below V_t, it causes a positive current to flow inside the cell; hence V_t is called reversal potential.

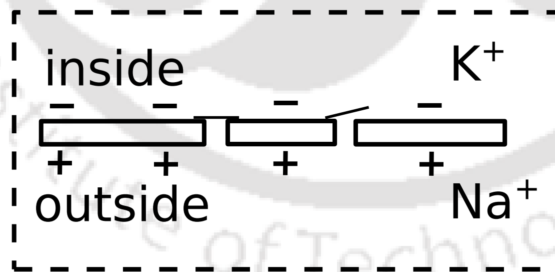


Figure 1.3: Cell membrane potential [20]

Ion	Inside	Outside	Reversal Potential (mv)
Potassium (K+)	400	10	-92
Sodium (Na+)	50	460	55
Chlorine (Cl-)	40	540	-65

Table 1.3: Concentration of ions inside neural processes and in the extracellular fluid

TH-3339_186102005

1.2.1.1 Initiation of the action potential

When a small pulse of current is injected into the cytoplasm, the potential responds, as shown in Fig.1.4, for currents that depolarize the membrane less than approximately 20 mv from its resting state. The potential shows a slow response that saturates after a few milliseconds. If the current pulse is terminated before the potential has reached approximately -40 mv, the membrane recovers, and no pulse is generated.

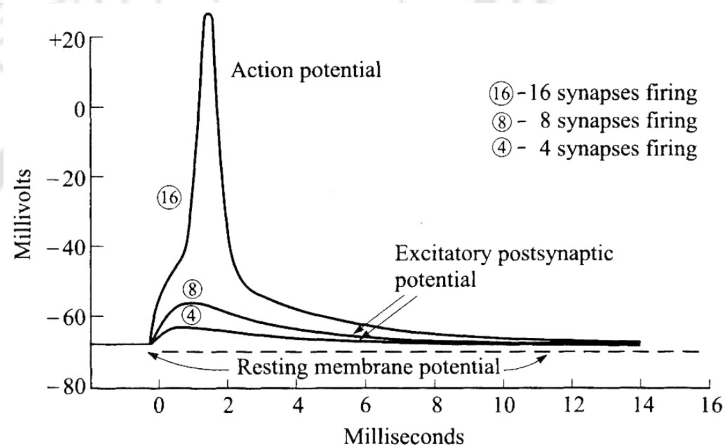


Figure 1.4: Response of the axon to simulation by 2-milisecond current pulse of increasing magnitude

An output pulse is generated once the potential becomes more positive than -40 mv. However, a pulse is generated even if the driving current is terminated. That potential is, therefore, a threshold beyond which a self-reinforcing reaction is underway, and no recovery is possible.

1.2.1.2 The Hodgkin Huxley model

The Hodgkin–Huxley model, or conductance-based model, is a mathematical model that describes how action potentials in neurons are initiated and propagated. It is a set of nonlinear differential equations approximating the electrical characteristics of excitable cells, such as neurons and muscle cells. The equivalent circuit for the above-said behaviour is shown in Fig. 1.5, which can be used to visualize the operation of the membrane over a wide range of conditions. In Fig. 1.5, all the $V's$ are the reversal potential of the ion, and the $G's$ are the conductance of the membrane current I for any given cytoplasm potential V. The membrane current can be calculated by Eq. 1.1.

$$I = (V_K - V)G_K + (V_{Na} - V)G_{Na} + (V_{Cl} - V)G_{Cl} \tag{1.1}$$

TH-3339_186102005

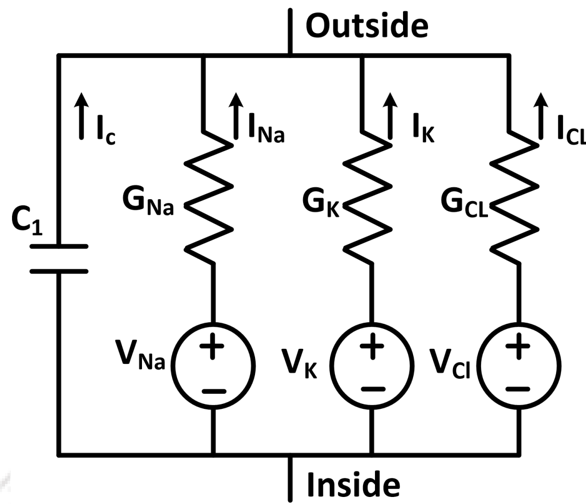


Figure 1.5: Equivalent circuit for cell membrane [20]

$$V_0 = V_K G_K + V_{Na} G_{Na} / (G_K + G_{Na}) \tag{1.2}$$

V_0 is the resting potential of the cytoplasm because it is the potential at which the cell comes to rest if left electrically undisturbed. The voltage V_0 at which the current is zero, also called resting potential, can be calculated using Eq. 1.2. In a typical neuron, G_k is approximately 20 times G_{Na}. The electrical activity in a patch of nerve membrane is achieved by making one or more ionic conductances dependent on some control quantity. That quantity can be voltage, the chemical substance's concentration, the light's intensity, and the degree of mechanical deflection.

1.2.2 Digital neuron

The functional behaviour of biological neurons is very complex. However, hardware computations only replicate some key features [21]. This behaviour includes a basic integrate-and-fire functionality in a neuron. $I\&F$ neurons are not biologically plausible but can still be used efficiently in spiking neural networks.

Digital spiking neurons help in achieving high-precision computing with ease of implementation. Instead of nonlinear differential equations, digital spiking neurons use cellular automation as basic building blocks. In [22], the authors follow a component-based approach and realize neuronal ion channel dynamics on FPGA. They use a parallel processing strategy to minimize delay and hardware efficiency, and exponential and division functions in neuronal ion channel models are calculated using a

TH-3339_186102005

hardware-efficient factoring approach. Some researchers employ Spinnaker as a platform to implement a spiking neural network; its comprehensive description can be found in [21]. In [23], the authors propose a digital ADExp neuron and use it to implement a very large-scale neuromorphic system, which can realize around 10000 synapses per neuron and use complex neural models and realistic network topologies.

A fully digital implementation of $I\&F$ neurons is shown in [24]. It consists of an 18-bit adder and an 18-bit accumulator connected to the comparator circuit, generating spikes for $I\&F$ neurons. Furthermore, leakage is added to the neuron model by using an inhibitory synapse. Authors in [25] presented a novel digital spiking neuron (DSN) that generates periodic spikes depending on the initial state of the neuron. A completely different methodology is used in [26] to design a hybrid spiking neuron consisting of shift registers. Its behaviour is more like an analog spiking neuron model. The authors in [27] encode a spike train by a digital code and propose a fast analysis method for the dynamics of the inter-spike interval and characterization of the modulation. Note that a generalized asynchronous digital spiking neuron model (GDN) elaborated in [28] is the most generalized version of asynchronous sequential logic-based neurons.

1.2.3 Analog neuron

One of the first neuron circuits proposed by Carver Mead in 1989 [29, 30] is also called a self-resetting neuron. An analog neuron with the functional characteristics of real nerve cells is proposed in [31]. This neuron can emulate the ion current dynamics and the biological neuron's discharge functionality. Later, [32] proposed an improved version of this neuron. The improved version has fewer circuitry and parameters than previous circuits, improving the spiking characteristics.

A circuit emulating the spiking behaviour of biological neurons is proposed in [33]. These neurons have been further used in [34], [35], [36] to develop large-scale neuromorphic systems. An $I\&F$ neuron with spike-based plasticity mechanism, adaptation, and spike-based learning mechanism is elaborated in [37]. In [38], the authors depict a generalized leaky integrate and fire model. It produces spiking and adaptive responses, depolarizing and hyperpolarizing after potentials. It also has the functionality of adaptive frequency and threshold. Similarly, [39] illustrates an array of LIF neuron circuits that are highly tunable and reconfigurable with accelerated dynamics. This neuron is implemented using a 65 nm CMOS process and is used to build the second-generation BrainScaleS neuromorphic hardware.

Using the basic $I\&F$ neuron architecture, the authors in [40] propose a motoneuron that exhibits

TH-3339_186102005

the properties of a mammalian motoneuron. This model is suitable for use in large arrays to analyze motor pools. [41] states an analog spiking neuron, where the current mirror configuration and the CMOS inverter are used to implement the integration and threshold functionality. In [42], a modified version of the Mihalas-Niebur neuron, a generalized version of the leaky $I\&F$ neuron, is implemented using a switched capacitor. The authors in [43] utilize a capacitor-free integrator to achieve a smaller physical area of a neuron. The feedback loop in the integrator is implemented using a low-power Schmitt trigger, thus reducing power consumption. The weights of a neural network are stored employing a floating gate MOS transistor as non-volatile memory. Further, floating gate transistors are utilized for accumulating input current [44]. Replacing the capacitor with a floating gate transistor considerably reduces the circuit's size and increases efficiency.

1.2.4 Hybrid neuron

The analog CMOS neurons need external biases to set their parameters originating from external memory. Hybrid neurons are proposed consisting of CMOS and a memristive circuit, which does not need any external bias for setting the parameters, thus easing the realization by storing neuron parameters in the variable resistance state of a memristor [45]. In [46], a LIF neuron implementation is proposed using three 1-transistor 1-resistor (1T-1R) structures.

A neuromorphic system incorporating a stochastic neuron is elaborated in [47]. Instead of integrating the input current, this neuron uses a probability function that generates a spike depending upon the weighted activity of the neurons present in the pre-layer. Further, a leaky integrate and fire spiking (IFS) neuron is implemented in [48, 49] using silicon nano-wire technology. The IFS neuron consists of four sections. The first section is for the integration of the input signals. The second section controls the neuron's threshold, and the third section realizes the leakage mechanism. Finally, the fourth section produces output, maintaining the zero level at the output.

A carbon nanotube (CNT) based spiking neuron is depicted in [50]. It consists of a crossbar architecture, where the gate terminals of transistors are connected to a row, and the source terminals are tied to a column. [51] illustrates a hybrid neuron circuit with stochastic firing behaviour. They explore the stochastic nature of the conductive bridge memory (CBRAM) to implement stochastic firing. However, implementing a spiking neural network using these neurons is not shown.

TH-3339_186102005

1.2.5 Digital Vs Analog neuron

A comprehensive study of analog and digital neurons is elaborated in [52], in which various performance metrics of a large-scale neuromorphic system are compared with digital and RRAM-based circuits. It can be observed in Fig. 1.6 that the total area and power consumption of the RRAM-based neuromorphic system are 14× and 2× less than its digital realization at 10 nm technology node.

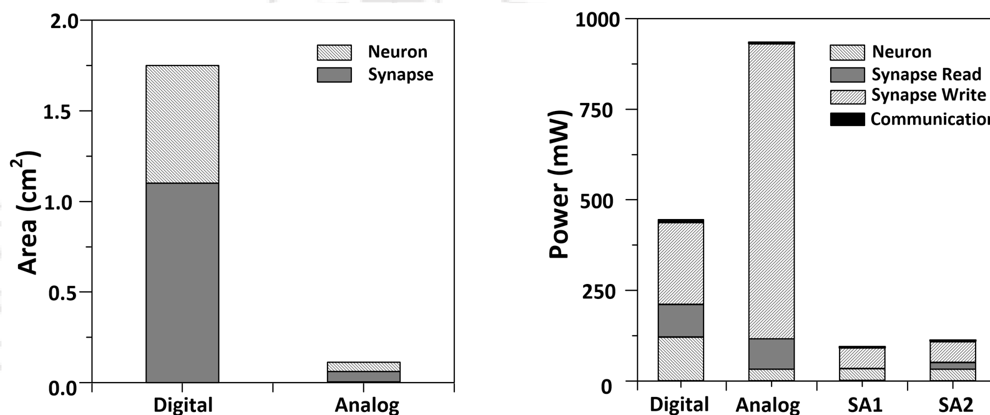


Figure 1.6: Area and power comparison of a million neuron learning system [53]

The power consumption of the analog implementation can be further reduced in two ways. One approach is to minimize the RRAM read and write currents. Another one is to shorten the duration of the programming pulse. It is estimated that a 10× reduction in the programming current of the RRAM lessens the overall system power by 6 (denoted by scaled analog-1 in Fig. 1.6). Furthermore, a 10× decrease in the current programming duration reduces the overall power by 4 (denoted by scaled analog-2 in Fig. 1.6) compared to the digital implementation of a neuron.

1.3 Synapse

Neuromorphic computing systems comprise synapse and neuron circuits arranged massively parallel to support the emulation of large-scale spiking neural networks. In neuromorphic systems, the bulk of the silicon is used by the synaptic circuits that integrate memory and computational primitives in the same area. One possible approach to save area and maximize synaptic architecture's density is implementing the basic synapse circuits in the dense crossbar arrays. However, this approach limits the role of the synapse to a basic multiplier. In biology, synapses are extremely sophisticated structures with complex and powerful computational properties, including temporal dynamics, state dependence, and stochastic learning behaviour. Details of the biological synapse and its equivalent

TH-3339_186102005

electrical realization are given below. realization are given below.

1.3.1 Biological synapse

The synapse is responsible for adaption and learning within the neural network by modifying the synaptic weight. A synapse can influence the firing of the post-synaptic neuron either as an individual or the part of several synaptic inputs. A strong synapse can cause the post-synaptic neuron to fire without additional synaptic inputs. In contrast, a weak synapse may not affect the firing of the post-synaptic neuron. However, a synapse can change its strength, and a weak synapse can eventually become strong, and vice versa. Hence, the synaptic weight can either increase (potentiation) or decrease (depression). This modification can either be short-term, STP (short-term potentiation), STD (short-term depression), long-term, LTP (long-term potentiation), or LTD (long-term depression).

There are two main types of synapses, namely electrical and chemical. The chemical synapses are abundant within a biological neural network [54]. A chemical synapse passes information from the pre-synaptic neuron cell to the post-synaptic neuron cell via the release of neurotransmitters into the synaptic cleft. As depicted in Fig. 1.7, the synaptic vesicles are released from/contained within the pre-synaptic axon by opening or closing voltage-gated calcium ion and Ca^{2+} channels. In the post-synaptic neuron, neurotransmitter receptors interact with the released neurotransmitters to change the potential of the post-synaptic cell [55].

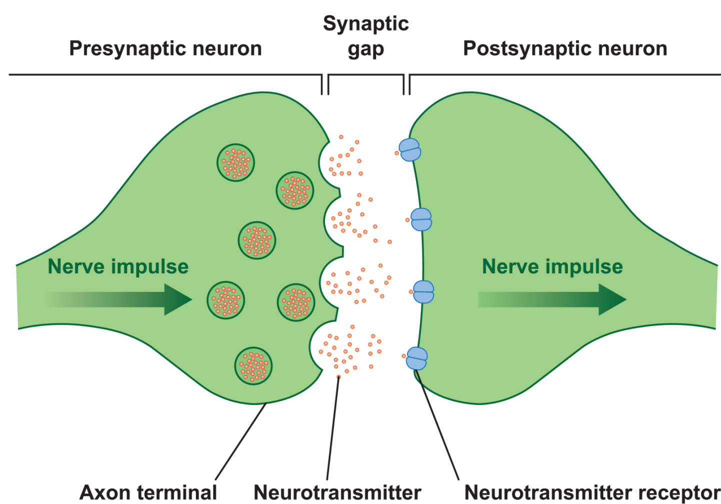


Figure 1.7: Biological synapse [55]

Hebb's theory [56] describes the change in synaptic weight based on the inputs and outputs of each neuron within the neural network. Hebb states, "When an axon of cell A is near enough to excite a cell B, and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that $A's$ efficiency as one of the cells firing B is increased." When the input from neuron A meets or exceeds the threshold of neuron B, the synaptic weight between neurons A and B is increased. Conversely, if neuron A has little or no effect on neuron B, the synaptic weight is reduced [56].

1.3.1.1 Dynamics of a synapse

A generic biological neural network is depicted in Fig. 1.8. It also shows the relationship between the spike timing difference Δt and the weight change Δw. As shown in Fig. 1.8, the synapse is located between $Neuron_{pre}$ and $Neuron_{post}$. The synapse provides variable connectivity. The weight of a synapse determines the amount of excitatory post-synaptic current into the membrane of $Neuron_{post}$ when there is a spike on $Neuron_{pre}$. The timing relationship between the pre-synaptic spike T_{pre}, and the post-synaptic spike T_{post} can modify the weight. This is called the Spike Time Dependent Plasticity (STDP) learning rule [57].

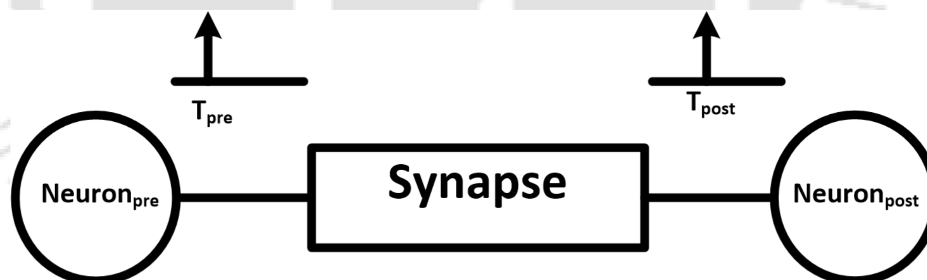


Figure 1.8: Generic neural network connectivity between neurons through a synapse

If T_{pre} is ahead of T_{post}, as depicted in Fig. 1.9 (a), the weight of the synapse increases, which is called long-term potentiation. Long-term depression happens when T_{pre} lags T_{post}, as seen in Fig. 1.9 (a). A biologically plausible STDP mechanism is shown in Fig. 1.9 (b). The closer the timing between T_{pre} and T_{post} is, the larger the weight change. According to the neurophysiological measurements, the relationship between the weight change and the spike timing difference can be approximated to have an exponential dependency [58], as illustrated in Fig. 1.9 (b).

TH-3339_186102005

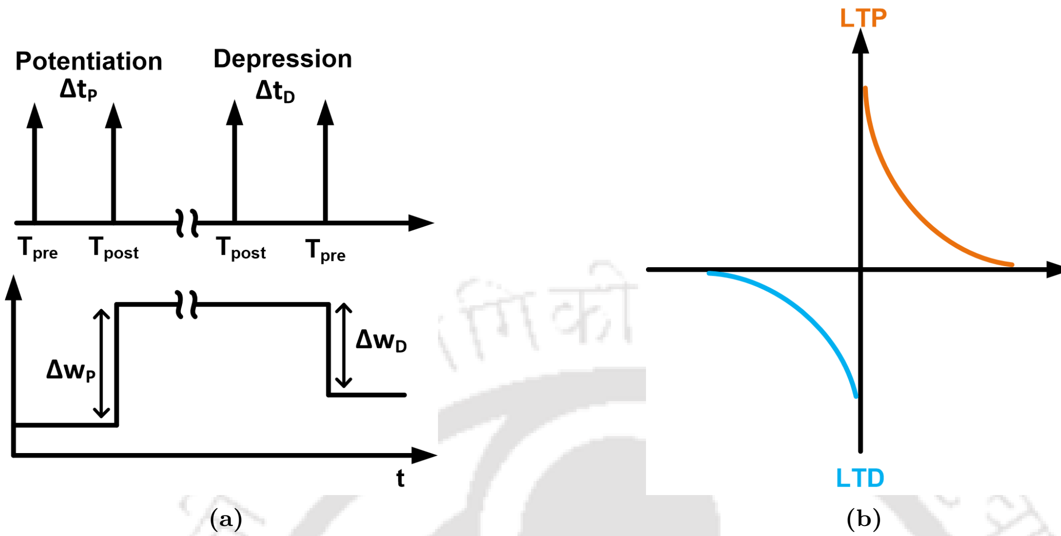


Figure 1.9: (a) Timing diagram for potentiation and depression (b) Curve for spike time dependent plasticity

1.3.2 CMOS synaptic circuits

The synapse models can be divided into two categories, the biologically-plausible implementations that include synapses for spike-based systems and the synapse implementation for traditional artificial neural networks.

Biologically plausible CMOS neuron and synapse circuits are presented in [59] and [60]. The synapse proposed in [60] can update its binary weight based on the STDP learning mechanism, and the neuron proposed in [59] operates in a leaky integrate-and-fire manner. Although these circuits very well emulate the behaviour of a biological synapse and neuron, as many as 34 and 22 transistors are used in their designs, respectively, requiring high energy consumption and area utilization. Further, it needs an analog amplifier for maintaining the binary states of the weight in the synapse, increasing energy consumption.

Another CMOS synapse and neuron circuits are designed using transconductance amplifiers [61]. However, the transconductance amplifiers that produce analog properties consume more area and energy. A cross-coupled inverter-based synapse storing the synaptic weight as a binarized value to implement long-term plasticity is presented in [62]. A CMOS symmetric/asymmetric synapse proposed in [63] consists of weight potentiation and depression circuits. These artificial synapses update the weight according to the STDP rule. Although they efficiently emulate the symmetric and asymmetric STDP mechanisms using decay pulses and switches, many transistors are required to implement the

TH-3339_186102005

circuits, resulting in higher power consumption.

A pair of simple CMOS synapse circuits, biomimetic and simplified versions composed of up to eight transistors and two capacitors, is illustrated in [64]. Although they are better than the previous designs, the refractory behaviour [59], an essential feature of a biological neuron, is not supported. Moreover, the membrane node having a heavy capacitive load should provide a full swing each time it is firing, resulting in a lower firing rate and increased switching energy consumption. Also, regular switching of the pull-up and pull-down transistors during the reset operation results in power consuming short circuit current. Variability due to these circuits' process, voltage, and temperature (PVT) variations is another concern in CMOS synaptic circuits.

1.3.3 Non-volatile memories as synapse

Since the synapses are typically the most abundant element in neuromorphic systems, they consume most of the chip area. Therefore, the focus is to optimize the synapse hardware realization. Thus, the challenge is to design neuromorphic circuits that emulate the computational properties of a synapse having optimal area utilization and power consumption.

There have been attempts to implement ANNs by fabricating a large number of synaptic element arrays and neurons based on digital and analog circuits. However, due to many technological limitations, such as device area, power consumption, and operating speed, conventional CMOS-based synapse devices cannot meet the requirements of artificial intelligence applications. As an alternative, the emerging non-volatile memories have been actively studied and developed. Non-volatile memory devices have recently emerged, providing a promising technology for addressing these problems. These devices offer a compact and efficient solution to model synaptic weights since they are non-volatile, have a nano-scale footprint, can be integrated with complementary metal-oxide semiconductor (CMOS) chips, might require little energy to change their state, and in addition, can emulate many synaptic functions observed in biological synapses.

These devices can store multi-bit information in a non-volatile manner. They can also consume very less energy in pJ regime, which enables significant improvements in integration and low-power operation compared to conventional CMOS-based synapse devices. Fig. 1.10 depicts an $I\&F$ neuron generating output pulse. The non-volatile memories are placed between the input and the output neurons, which forms a synaptic array crossbar. Each memory cell in the array can be trained by learning rules, such as Spike Time Dependent Plasticity (STDP). In the next section, a brief discussion

TH-3339_186102005

on non-volatile memories is presented.

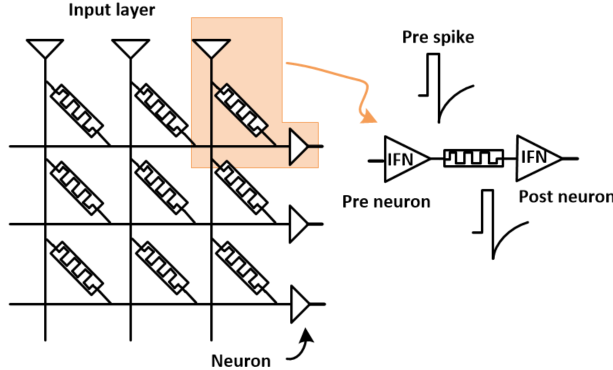


Figure 1.10: Non-volatile memories as synapse

1.4 Emerging non-volatile memories

Semiconductor memory technologies play an essential role in today's high-performance computing systems. Semiconductor solid-state memories can be found in all electronic systems, such as computers, portable electronics, automotive applications, and data centers. With an increasing demand for high-performance, large-capacity, and low-cost AI-enabled portable devices, non-volatile semiconductor memories (NVM), such as flash memory [65], are being used to realize of neuromorphic systems. The NAND flash [66], due to its lowest cost per bit, is preferred compared to other commercial NVMs, NOR flash [67] and EEPROM [68].

Due to the technological advancements in fabrication processes and aggressive scaling down of the memory cell size, the NAND flash memories provide low cost per bit. However, this would eventually end, as the planar flash memory is expected to face physical limitations. The fabrication of planar NAND flash memory in a 10 *nm* technology node requires a complicated patterning process [69], resulting in an excessive increase in the cost per bit and making these memories unproductive. On the other hand, the reliability, such as data retention and program endurance, and the cell-to-cell uniformity become worse due to the less charge storage in these devices.

Therefore, a non-planar 3D vertical structure is proposed for NAND flash [67]. In this 3D configuration, NAND cells are formed in the vertical direction, reducing the cost per bit. Based on the 3D design, NAND flash scaling is expected to be implemented on lower technology nodes [70]. However, it is also anticipated to impose issues similar to the planar NAND, facing the physical limitation of charge-based memory devices and an exorbitant increase in manufacturing costs.

TH-3339_186102005

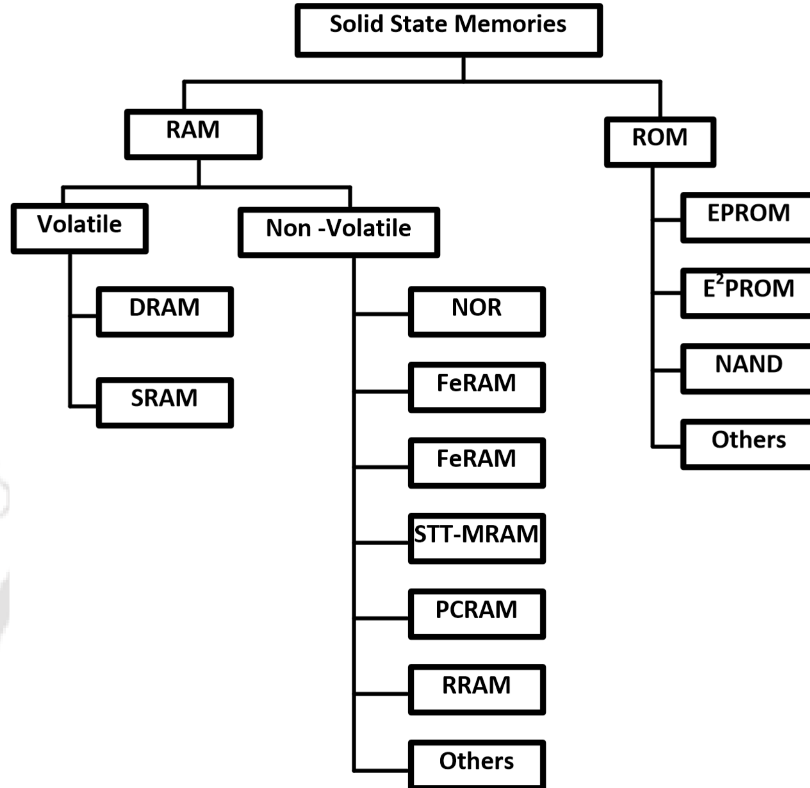


Figure 1.11: Classification of semiconductor memories

Motivated by finding a successor to existing memories, many new technologies have been explored in recent years to keep pace with the increasing demand for high-density, low-cost, and high-performance memory applications. Novel memories, such as Phase Change Random Access Memory (PCRAM), Spin-Transfer Torque Magnetic Random Access Memory (STT-MRAM) [71], Ferroelectric Random Access Memory (FeRAM) [72], Resistive Random Access Memory (RRAM) [72] are being investigated intensively.

The emerging memories are also called storage class memory (SCM). The SCM can be further classified as main memory SCM (M-SCM) and storage type SCM (S-SCM). The classification of semiconductor memories is shown in Fig. 1.11. A comparison between the current and emerging memory technologies is exhibited in Table. 1.4. All these emerging technologies demonstrate non-volatile memory behaviour and promising characteristics, such as fast program speed, low read and write energy consumption, and excellent reliability in terms of endurance and retention. Among all these memories, STT-MRAM seems to be a good candidate for the M-SCM due to its fast access speed (10 ns), endurance of (10^{16}), limited area consumption ($10F^2$), and low power operation (pJ), which

TH-3339_186102005

Table 1.4: Comparison between the current and emerging memory technologies

parameters	Volatile		Emerging non-volatile memories			
	NAND	NOR	FeRAM	PCRAM	STT-MRAM	RRAM
Configuration	1T	1T	1T1C	1S1R	1T1R	1S1R
Cell Area (F^2)	5	5	22	4	10	4
Programming Time (ns)	$10^6/10^5$	$10^6/10^7$	10/10	20/50	$<$ 10/10	$>$ 10/10
On/Off ratio	-	-	-	> 10	˜2	> 10
Endurance	$> 10^5$	$> 10^5$	$> 10^{1}4$	$> 10^8$	$> 3 \times 10^6$	$> 10^{1}0$
Retention(years)	> 10	> 10	> 10	> 10	> 10	> 10
Energy/bit	$10 - 100$ pJ	$10 - 100$ pJ	10 pJ	10 pJ	$\sim$ pJ	$\sim$ pJ
Application	Storage	Storage	Storage	Storage	Main Memory	Storage

satisfy the most stringent requirements for the high-performance memory applications. Moreover, due to its non-volatile behaviour, STT-MRAM does not require a periodic refresh process, which is mandatory for DRAM. However, RRAM seems to be the most promising candidate for the S-SCM applications due to the following reasons:

- **Simple structure:** RRAM has a very simple two-terminal Metal-Insulator-Metal (MIM) structure, which allows high geometrical scalability.

- **Good Manufacturability:** RRAM uses fully CMOS-compatible materials, which can be fabricated using fab-friendly processes. On the contrary, FeRAM, STT-MRAM, and PCRAM require dedicated processes using ferroelectric, chalcogenide, or magnetic materials.

- **Excellent scalability:** Resistive memory functional devices have demonstrated 10×10 nm^2 size [72], exceeding the physical limitations of the flash memory. Furthermore, it has better scaling potential compared to STT-MRAM and FeRAM. The latter two memory technologies require a complex material system for fabricating functional devices.

- **Low cost per bit:** Implementing RRAM in dense cross-point arrays can achieve the smallest cell footprint, i.e., $4F^2$, with F being the feature size [73].

- **Multi-level cell:** Resistive memory can provide a large on/off resistance ratio, enabling multi-level cell operation and reducing the cost per bit. On the contrary, a typical on/off resistance window for STT-MRAM is less than 2, which makes multi-level cells nearly impossible.

TH-3339_186102005

- **Fast write speed:** SET and RESET operations of resistive memory cells take less time ($> 10\ ns$) than flash memories ($> \mu s$).

1.4.1 Resistive Random Access Memory (RRAM)

Resistive Random Access Memory (RRAM) is the resistive switching memory technology class with metal-insulator-metal (MIM) structures. The concept of resistive switching is familiar to most emerging non-volatile memory technologies. For instance, the magnetic field is involved in the resistance change of STT-MRAM. For PCRAM, a thermal process controls the chalcogenide material's phase transitions between crystalline and amorphous. According to the defect type involved in the switching, resistive memory cells can be categorized into the oxygen vacancy-based RRAM and the metal ions-based conductive bridge memory (CBRAM). Furthermore, filamentary and non-filamentary switching is possible based on different ways of modulating conduction by oxygen vacancies, as shown in Fig. 1.12.

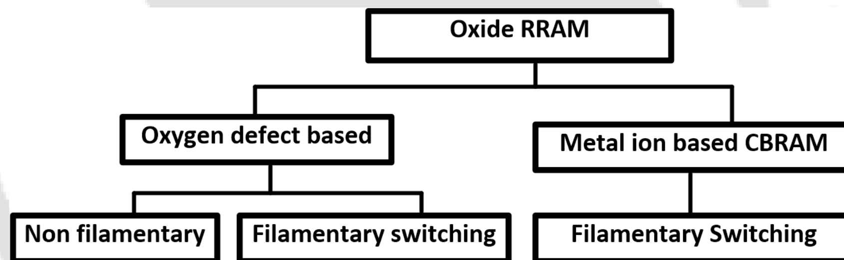


Figure 1.12: Classification of Resistive Random Access Memories [67]

Filamentary resistive switching behaviour is found among various transition metal oxides (TMO). Some examples are shown in Table. 1.5. In addition to oxide materials, metal electrodes also play an important role in the switching mode of RRAM. For instance, [74] reported ZrO_{2-} based RRAM with different electrodes (Pt and Ti). Even with the same oxide material, the switching mode can be different. In most cases, bipolar switching can be achieved by using oxidizable electrodes, such as Ti, Hf, TiN, while unipolar switching is obtained using inert electrodes (Pt) for both sides. In some cases, both unipolar and bipolar switching can be achieved using the same material, $TiN/HfO_2/Pt$ [75], depending on the polarity of the voltages that are applied to the device.

The RRAM devices can also be classified into two categories according to the switching modes: unipolar and bipolar switching RRAM. For unipolar switching devices, resistive switching depends only on the amplitude of the applied voltage, regardless of the polarity of the applied voltage (SET/RESET

TH-3339_186102005

Table 1.5: Switching modes for various metal-oxide based RRAM [67].

Unipolar switching	Bipolar switching
$Pt/NiO/Pt$	$Pt/NiO/SiRuO_3$
$Pt/TiO_2/Pt$	$Pt/TiO_2/TiN$
$Pt/ZrO/Pt$	$TiN/ZrO/Pt$
$Pt/ZrO_2/Pt$	$Ti/ZrO_2/Pt$
$TiN/HfO_2/Pt$	$TiN/HfO_2/Pt$
$Pt/Al_2O_3/Pt$	$Ti/Al_2O_3/Pt$

can be done on the same polarity). If a unipolar RRAM shows SET/RESET switching for both polarities, it is called a nonpolar RRAM [76]. On the contrary, for a bipolar RRAM, SET/RESET strongly depends on the polarity of the applied voltage. If the SET occurs on one polarity, the RESET gets triggered on the opposite polarity.

Although both unipolar and bipolar RRAM has received significant attention over the years, research has recently focused on the bipolar switching mode RRAM for the following reasons. First, the bipolar switching devices depending on the oxygen drift/migration, require less switching power compared to the unipolar devices. Unipolar switching needs thermally activated diffusion of the oxygen ions, which causes a high RESET current. Second, oxygen ions migration-based bipolar RRAMs always show better endurance than unipolar RRAMs. However, the advantage of unipolar switching RRAM is that it can work with a simpler unidirectional selector, such as a PN diode [67].

1.5 RRAM based synaptic architectures

RRAM has been widely explored for implementing synaptic architectures. Implementing binary synapse is feasible as the gap between its resistive states is large, making binary RRAMs inherently variation tolerant. However, binary synapses have very low density [77]. To increase the density, keeping RRAM variation tolerant, [78] proposes a multi-memristive synaptic architecture with a counter-based arbitration scheme. However, employing multiple memristors in the realization of synapses reduces the advantages of RRAM because it increases the overall area utilization of a single bit. Multi-level RRAMs have been extensively explored to obtain multi-bit precision in a single cell [79–82]. Although multi-level cells provide considerable advantages in terms of area and power, designing circuits to program the RRAM's resistive states accurately is a significant challenge.

The multi-level cells are programmed by varying the voltage or compliance current during SET

TH-3339_186102005

or RESET. In [83] and [84], pulses with varying magnitudes and widths are employed to program an RRAM, whereas [85] uses pulse width modulation. In the methods reported in [83] and [84], the RRAM resistance is read multiple times to verify whether it is updated correctly. If the RRAM is not appropriately programmed and the desired resistance is not obtained, then RRAM is reprogrammed until an appropriate resistive state is reached. These techniques give reliable results but increase latency due to multiple read and write cycles. Moreover, considering the variability of the synaptic devices, large neuromorphic systems become more complicated and less flexible [86].

Most state-of-the-art synaptic architectures do not have any programmability provision, limiting the architecture's applicability to a single application. This makes the synaptic arrays more prone to errors due to the variation and restricts the usage of the hardware to the training use cases. This also limits the existing architectures to inference since there is no mechanism to program the RRAM cells [87]- [88]. Off-chip training requires transmitting massive data and model parameters between the cloud and the edge devices, which is hard to deploy in complex environments while protecting the privacy of IoT applications. However, on-chip training eliminates the need for high-speed data transmission and security issues. Using RRAM for on-chip training involves challenges, such as higher weight precision and precise RRAM programmability [89]- [90]. Therefore, we design an RRAM-based synaptic architecture with continuous sensing and a feedback mechanism to address these issues. The proposed mechanism enables multi-level RRAM to be reliably programmed in a single write cycle. The $4T-1R$ structure incorporated in the architecture enables reprogramming the array multiple times, making it applicable for multiple applications.

1.6 Artificial Neural Networks

Artificial neural networks (ANNs) are mathematical or computational models that are inspired by the structure and operation of biological neural networks. ANNs consist of processing elements called activation functions or neurons, interconnected to generate a neural network. The interconnections are represented by the weight that indicates the strength of the connection between the two activation functions. Typically ANNs are used to analyze a highly complex problem or if no algorithm exists to solve a specified problem. Pattern recognition, image processing, control, and robotic systems are some examples of hardware and software applications of ANNs.

While software implementation of ANNs is comparatively easier, more computational resources are

TH-3339_186102005

required to simulate large complex networks. Additionally, current Von Neumann architecture-based computers consume a lot of power to realize these neural networks. Therefore, creating a complex, parallel, and energy-efficient system using software is difficult. Hardware ANNs are more biologically plausible and can provide better flexibility and potential in terms of parallelism and speed. Fig. 1.13 presents an example of an activation function that can be trained to find an optimal solution and solve a specified problem. This is done by a learning rule which modifies the weighted connections between each neuron in a larger neural network.

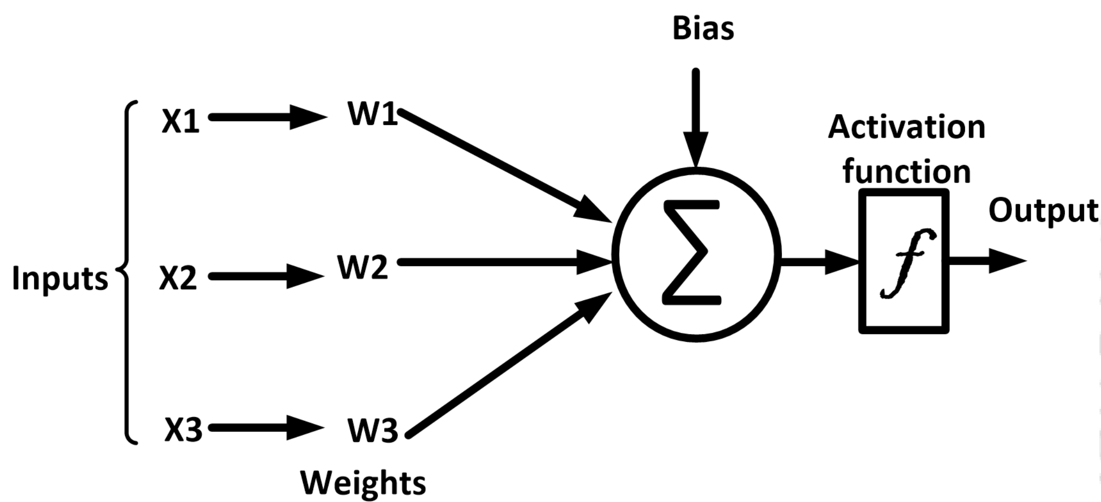


Figure 1.13: Artificial Neural Network

The first generation of ANN consisted of McCulloch-Pitts neurons. McCulloch-Pitts view neurons as computational units which use a threshold activation function. McCulloch-Pitts neurons produce output as 1 when the weighted sum of its inputs is above the threshold. If the weighted sum is below the threshold, an output of 0 is produced. Within the network, information is encoded by the presence or absence of action potentials. Even though McCulloch-Pitts neurons produce a digital output, they have been successfully applied in multilayer perceptron networks.

In the second generation of ANN, the threshold activation functions are replaced with continuous activation functions. The continuous activation functions, such as sigmoid or hyperbolic tangent functions, allow analog inputs and output. It also allows rules, such as gradient-descent algorithms for learning, which can train and change the weights connected between the neurons.

Fig. 1.14 shows a fully connected feed-forward neural network that can be implemented using second-generation artificial neurons. In this network, information is only passed forward from input to output. No feedback of information from output to input or hidden layers occurs. In more complex

TH-3339_186102005

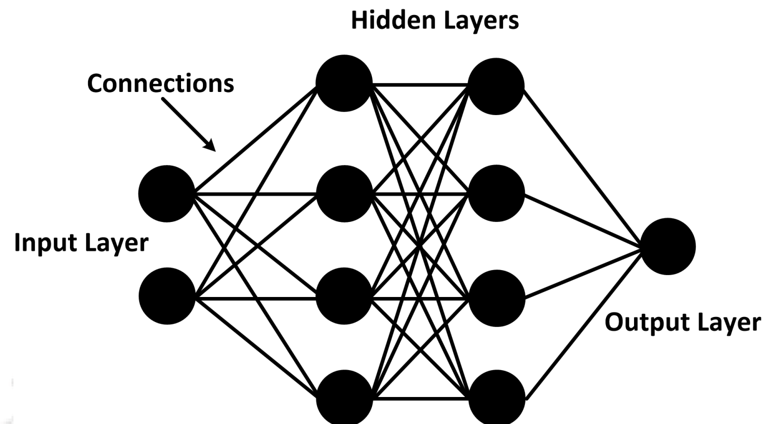


Figure 1.14: Fully connected neural network

networks, feedback paths are included as they offer greater stability and are more like biological networks. Recurrent networks and networks which require digital outputs can also be implemented using the second generation of ANN.

1.7 Spiking Neural Network

The first and second generations of ANNs emulate the main features of biological neural networks, such as plasticity, summation, and thresholding. However, they do not emulate all features of a biological neural network. Experimental results indicate that rate coding accurately describes activity in the brain. Rate coding implies that an averaging mechanism is used. However, a biological neuron operates by a "spike or no spike" mechanism with a fixed firing rate, described as an intermediate spiking frequency. While the second generation can implement this intermediate spiking frequency, it still lacks important biological features. Recent research suggests that the timing of individual action potentials, spikes, are used to code brain information. Researchers also believe the human brain's computational power is in its ability to process large numbers of these spikes in parallel. Thus, spike-based coding schemes are considered to be more efficient than rate-based coding.

The third generation of artificial neural networks, spiking neural networks (SNNs), are more biologically plausible. SNNs are designed such that the scale and connectivity observed in the brain are emulated in hardware. To obtain better efficiency, the hardware implementation of the SNN must occupy a minimum circuit area and have low power consumption. Information is computed and communicated through spatial and temporal summation of individual spikes, similar to biological neurons.

TH-3339_186102005

As depicted in Fig. 1.15, the input is encoded as spikes, and the output is also obtained as spikes.

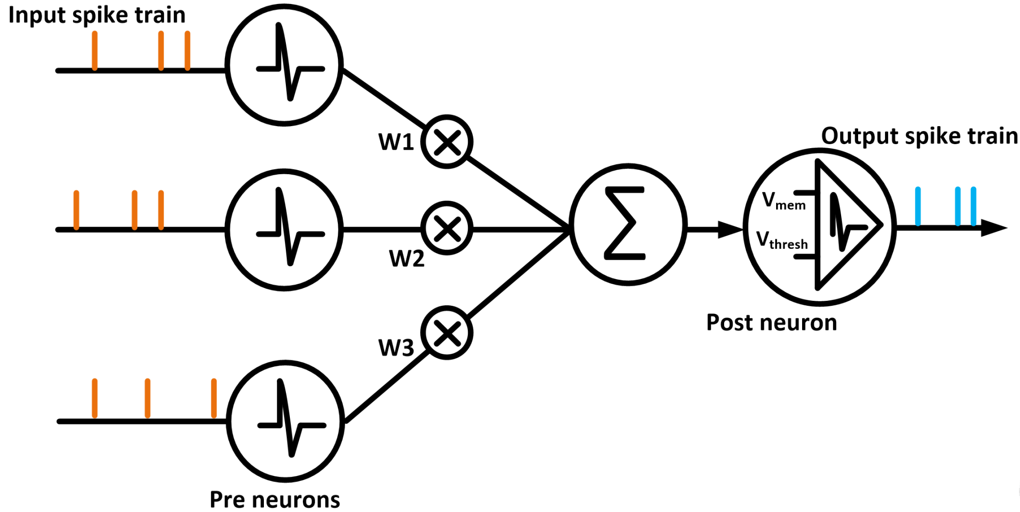


Figure 1.15: Spiking Neural Network

The SNNs use pulse coding instead of rate coding. This allows information to be encoded in the form of frequency and magnitude. SNNs do not fire at the end of each propagation cycle. Instead, they only fire when the weighted sum of their inputs causes the neuron's membrane potential to exceed its threshold value. Therefore, spiking neural networks have more computational power over equivalent networks of static neurons.

Although SNNs are theoretically more potent than second-generation networks; however, SNN training issues and hardware requirements limit their use. SNN implementation on Von Neumann architectures has proved to be very inefficient compared to second-generation neural networks. However, the advent of neuromorphic processors has proven that SNNs can be more efficient on neuromorphic hardware. Further, the advent of novel devices such as RRAM in the last decade has paved the way to further reduce neuromorphic hardware's power consumption and area utilization.

A hardware implementation of an adaptive STDP learning rule stated in [91] reports an accuracy of 94%. However, the implementation details of neurons are not discussed. A hybrid RRAM synapse with a network accuracy of 97% is reported in [92]. However, a separate synapse for the dynamic and fine-tuning phases to program the RRAM synapse accurately. This doubles the area utilization. Other works, [93], [94], [95], [96], have shown excellent realizations of synaptic architectures in different configurations and have implemented SNN with accuracies of 76%, 87%, 86%, and 87%, respectively. However, these architectures can be employed for designing large-scale neuromorphic systems only if

TH-3339_186102005

the neurons are implemented in hardware. Employing RRAM-based synapses and digital neurons in large neuromorphic systems diminishes all the advantages. [97] realizes an RRAM-based stochastic neuron and RRAM-based synapse, and a spiking neural network, but details of the synapse, i.e., the number of bits of the RRAM array, the impact of RRAM variations, random telegraph noise (RTN), and sneak current paths effects, are not discussed in it.

From the above discussion, it can be concluded that there is a huge research gap in the hardware implementation of SNNs. Most of the state-of-the-art works focus on the parts of an SNN. Therefore in this thesis, we bridge the gap in the current state-of-the-art works by implementing complete hardware for SNN and proving the efficacy of the proposed neuron and synapse to realize large-scale neuromorphic systems.

1.8 Motivations and Contributions of the Thesis

Based on the problems defined in the previous sections, the proposed work's motivation and the thesis's contributions are presented in this section.

- As we know, efficient hardware must be developed to meet the ever-increasing demands of AI applications. Knowing the limitations of the current CPUs that are based on Von Neumann architectures, the functionalities of the human brain inspired us to develop more brain-like architectures where the memory and computing are not separate but work in synchronization.
- The contemporary neuron circuits fail to meet the low power requirements of large-scale neuromorphic systems. Moreover, the analog neuron circuits employ capacitors to implement the integration operation in an $I\&F$ neuron, utilizing a large area. Certain features of novel devices, such as an abrupt SET in RRAM, can be explored to obtain the integration operation, thus, saving the area on the silicon.
- Since neurons form the basic components of neuromorphic systems and due to the limitations of the CMOS technology in achieving the low power requirements, we propose an RRAM-based $I\&F$ neuron. The proposed $I\&F$ neuron circuit consists of two RRAMs for integrate-and-fire operations, whereas the pulse propagation and reset circuit consists of 22 CMOS transistors. It consumes $1.5fJ$ per spike, 48% and 53% less than the contemporary neurons designed using nanoscale FBFET and PDSOI-MOSFET.

TH-3339_186102005

- Apart from neurons, synapses are also found in abundance in neuromorphic systems. Therefore, it is necessary to develop an optimized synaptic array. The state-of-the-art synaptic architectures have many limitations, such as they do not have programmability that limits the applicability of the synapse in many applications. Moreover, programming the RRAM array at a specific conductance level is challenging. Therefore, we present $4 - bit/cell$ synaptic array that allows programmability and a continuous feedback mechanism that helps in programming the RRAM cells to achieve a precise conductance level.

- RRAM-based systems are prone to cycle-to-cycle, device-to-device variations, and random telegraph noise (RTN). We propose a variation-tolerant neuron and a synaptic array in this thesis. It is found that the neuron spikes appropriately in the presence of variations and RTN. The proposed continuous feedback mechanism to synaptic cell programming helps achieve variation tolerance.

- The state-of-the-art hardware implementations have many limitations, such as the spiking neural networks (SNN) do not specify the details of the neurons used to implement SNN. Even if the neuron details are provided, the programming mechanism is not discussed, raising ambiguity about the RRAM cells programming for different weights. Also, it is crucial to evaluate the applicability of the proposed neuron and synaptic architecture for large neuromorphic systems. Therefore, we implemented an SNN using the proposed RRAM-based $I\&F$ neuron and the programmable synapse and validated it with the MNIST dataset.

1.9 Organization of the Thesis

This thesis work is organized into six chapters to address the issues mentioned in the previous section. The content of each chapter is summarized as follows:

- **Chapter 2:** In this chapter, we study various RRAM models available for circuit simulation and analyze the Peking Verilog-A and Unimore Verilog-A models. We plot the $I - V$ characteristics of the RRAM and perform parametric analysis to study the effects of various parameters on the $I - V$ characteristics of the device. We analyze both models for cycle-to-cycle and device-to-device variations and random telegraph noise (RTN), which is essential to verify the performance of the circuits in the presence of variations. Based on the performance of both models, we choose

TH-3339_186102005

the Unimore model for implementing the $I\&F$ neuron and synapse in this thesis.

- **Chapter 3:** In this chapter, we present the proposed scalable energy-efficient RRAM-based $I\&F$ neuron. We match the functionality of the proposed neuron with a digital $I\&F$ neuron. We incorporate the RRAM reset functionality in the neuron circuit, making it more compact. The performance of the proposed neuron at various process corners is also studied in this chapter. We further evaluate the performance of the proposed neuron in the presence of noise and RRAM variation.

- **Chapter 4:** In this chapter, we discuss a variation tolerant RRAM-based $4 - bit/cell$ synaptic architecture. A programming mechanism is proposed that helps precisely program the RRAM cells in the presence of cycle-to-cycle and device-to-device variations inherent to RRAM devices. We perform Monte Carlo analysis to verify the effects of variations on the synaptic array. The proposed synaptic array's area, power, and latency estimation are discussed in detail in this chapter.

- **Chapter 5:** This chapter presents a spiking neural network using the RRAM-based $I\&F$ neuron and the programmable synapse. A quantization method to map the learned weights to the $4 - bit/cell$ synapse is discussed in this chapter. We also elaborate on the input encoding method and training methodology for SNN implementation. Finally, we compare the proposed SNN with contemporary works incorporating non-volatile memories to implement the synapse.

- **Chapter 6:** In this chapter, we conclude the thesis and narrate the future perspective of the proposed RRAM-based $I\&F$ neuron and reprogrammable synaptic architecture. The proposed neuron and synapse exhibit low power and resource requirements and are suitable for developing large-scale neuromorphic systems.

TH-3339_186102005

TH-3339_186102005

2

Resistive Random Access Memory

Indian Institute of Technology Guwahati

Contents

2.1 Introduction 32
2.2 RRAM circuit simulation models 33
2.3 Peking RRAM Verilog-A model 35
2.4 Unimore RRAM Verilog-A model 45
2.5 Parallel and series combination of RRAM 55
2.6 Cycle-to-Cycle and Device-to-Device variation 58
2.7 Summary 59

TH-3339_186102005

2.1 Introduction

Resistive Random Access Memory (RRAM) is one of the fastest emerging memory technologies that can play a significant role in replacing conventional semiconductor memories, such as Dynamic Random Access Memory (DRAM), Static Random Access Memory (SRAM), and Embedded Dynamic Random Access Memory (EDRAM). The nonvolatile characteristics of the memory-based RRAM cells make them more attractive for nonvolatile random access memory design. Existing research on RRAM technology focuses mainly on integrating CMOS and non-CMOS devices and circuits.

RRAMs are two terminal devices that can retain their internal resistance states depending on the history of applied voltages or currents. RRAMs usually have a simple *metal/insulator/metal* structure. They have attracted special and intensive interest because of their promising properties, such as scalability, CMOS compatibility, low power consumption, and analog conductance modulation. They have recently shown outstanding characteristics like high-speed, high-density, and low-energy operation. Therefore, they are considered among the most promising memories for unconventional computing technologies.

Since these devices are still in the research and development phase, the models for circuit simulation have yet to be readily available. However, a simple and accurate model is crucial for rapid design and verification when using RRAM devices for circuit and system design. Many different models have been proposed earlier with various properties and have their limitations. Recently a few Verilog-A models have been developed that can be very efficiently employed to implement RRAM circuits. Therefore, it becomes essential to analyze these models for various characteristics, such as type of model, type of switching, genericity, complexity, compatibility with actual physical switching mechanisms, linearity, symmetry, voltage or current control, hard set or soft reset, the existence of a threshold, voltage level, timing dependence, temperature dependence, and variability. The appropriate model selection gives insight into the behaviour of RRAM and the efficient use of its properties.

In this chapter, we present various RRAM models available for circuit simulation, perform detailed simulations, and analyze the Peking Verilog-A and Unimore Verilog-A models. Based on the performance of both models, we justify using the Unimore model to implement the $I\&F$ neuron and synapse in this thesis. We plot the $I-V$ characteristics of the RRAM by varying a single parameter and keeping other parameters constant. We analyze both the models for cycle-to-cycle and device-to-device variations, and random telegraph noise (RTN), which are essential to verify the performance

TH-3339_186102005

of the circuits in the presence of variation.

2.2 RRAM circuit simulation models

Several RRAM models can be used for designing and simulating RRAM-based circuits. Some of the most commonly used models are described below.

(i) **Stanford Model:** The Stanford model is one of the earliest models proposed for RRAM devices. It is a simple model that describes the resistive switching behaviour of the RRAM cell using a single equation. The model assumes that the resistance of the cell depends on the number of oxygen vacancies in the resistive material. This SPICE-compatible model characterizes the metal-oxide RRAM bipolar switching behaviour [98].

(ii) **TEAM Model:** The threshold adaptive memristor model [99] is a simplified Simmons tunnel barrier model. This model depends on the same physics principles as the Simmons tunnel barrier model but uses simple polynomial equations instead of exponentials to relate the current to the physical device parameters. This model is simple, general, and computationally efficient.

(iii) **SPICE Model:** The SPICE model proposed in [100] can be used to simulate the behaviour of RRAM devices using SPICE simulation tools. This model assumes a memristance controlled by a voltage source. The memristive system considered in this model is a subcircuit, including a resistor, a current source, and a capacitor. The SPICE model includes subcircuits for the memory cell, programming circuit, and read circuit, which can be utilized to simulate the electrical behaviour of the RRAM device under different operating conditions.

(iv) **CBRAM Model:** The conducting bridge random access memory (CBRAM) model uses metallic filaments to store the data [101]. The CBRAM model includes several equations that describe the formation and dissolution of metallic filaments in the resistive material of the cell. It also describes the effects of voltage and current on the resistive switching behaviour. It is a physics-based compact device model that characterizes the dependence of ion migration velocity on the electric field. It also incorporates the metallic filament's vertical and lateral growth and dissolution dynamics.

(v) **IM2NP RRAM Model:** The IM2NP model is a compact model that describes the SET and RESET operations in bipolar resistive switching in Oxide-based memory devices [102]. The

TH-3339_186102005

model also considers the conductive filament (CF) electric field-induced creation and dissolution by including an electrochemical reaction and thermal mechanism equation. The model is calibrated on dynamic and quasi-static experimental data.

(vi) **Peking Verilog-A model:** The Peking Verilog-A model is physics-based. It models the formation and rupture of the conductive filament in horizontal and vertical directions. It models the generation of the oxygen-vacancies in the oxide layer. This model includes cycle-to-cyle and device-to-device variations.

(vii) **Unimore Verilog-A model:** The Unimore Verilog-A model is a physics-based model. It includes DC and pulsed characteristics. It models the current due to the formation of a conductive filament in the vertical direction. The model includes cycle-to-cyle and device-to-device variations. This is the only model that includes random telegraph noise in the DC operating mode.

Overall, these models provide a range of options for accurately modeling and simulating the behaviour of RRAM devices under different operating conditions, enabling more efficient and effective design. Table. 2.1 compares all the models based on the factors described below.

- **Type of the model:** It exhibits model type to be compact, analytical, or physics-based.
- **Type of switching (unipolar or bipolar):** It showcases the change in the state due to the application of only a positive voltage, or positive and negative voltage for switching states.
- **Efficient use in RRAM arrays:** It illustrates whether the model can be used for implementing large RRAM arrays.
- **Complexity:** A model is considered complex if the equations use hyperbolic sine and exponents rather than polynomials. This can be determined from the model equations.
- **Symmetry:** This points the symmetry in the SET/RESET processes. This feature appears in the simulated $I-V$ characteristic of the model.
- **Non-linearity:** The non-linearity in the RRAM devices is due to the nonlinear ion drift of the ionic defects accelerated by the conductive filament heating. If it is included in the model equations, it reflects in the $I-V$ characteristics.

TH-3339_186102005

Table 2.1: Comparison of various models

Model	Stanford	TEAM	CBRAM	SPICE	IM2NP	Peking	Unimore
Type of Model	Physics based	Physics based	Physics based	Analytical	Physics based	Physics based	Physics based
Efficient use in RRAM arrays	✓	✗	✗	✗	✓	✓	✓
Bipolar switching	✓	✓	✓	✓	✓	✓	✓
Low complexity	✗	✓	✗	✓	✓	✓	✓
Non linearity	✓	✓	✗	✓	✓	✗	✓
Symmetric	✗	✗	✗	✓	✗	✓	✓
Voltage controlled	✓	✗	✓	✓	✓	✓	✓
Hard SET	✓	✓	✓	✓	✓	✓	✓
Soft RESET	✗	✓	✓	✓	✓	✓	✓
Electroforming	✗	✗	✗	✗	✓	✗	✓
Temperature dependence	✓	✗	✓	✗	✓	✓	✓
Variability	✓	✗	✓	✗	✓	✓	✓
Random Telegraph Noise	✗	✗	✗	✗	✗	✗	✓

- **Hard SET/Soft RESET:** It is the ratio between the RESET and SET times. A high ratio means a hard SET and a soft RESET. This can be observed from the $I-V$ characteristics. Mostly, the SET is abrupt, and the RESET is gradual, resulting in hard SET and soft RESET.

- **Temperature dependence:** It indicates whether the model incorporates the temperature effects.

- **Variability:** It states model's cycle-to-cycle and device-to-device variability.

- **Random Telegraph Noise (RTN):** RTN is an important source of variability in RRAM devices. The Unimore model is the only one that includes RTN in the device model.

2.3 Peking RRAM Verilog-A model

2.3.1 Device structure

The Peking RRAM Verilog-A model is a physical-based analytic model of metal oxide-based RRAM. This model assumes a conductive filament (CF) evolution process described by the change in CF during SET/RESET at various inputs [103], [104]. Fig. 2.1 illustrates the structure of the modeled

TH-3339_186102005

RRAM. w is the width of conducting filament (CF), and x is the length of the gap region between the top electrode and CF. The length of the gap is initialized to x_0 before the start of the SET or RESET process. It can be observed that x reduces due to CF formation and expands in the horizontal direction indicated by width w. Thus, RRAM is SET when CF touches the top electrode, forming a low resistance path between the electrodes. The CF ruptures when a negative voltage is applied between the electrodes, illustrated by the downwards arrow in Fig. 2.1.

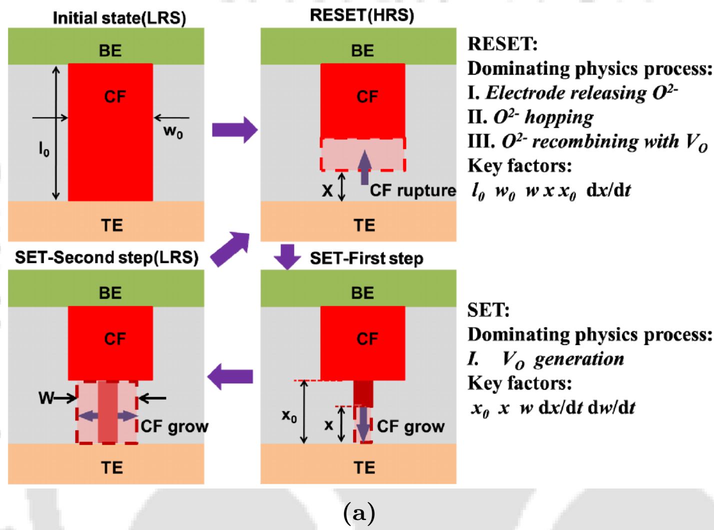


Figure 2.1: Schematic of conductive filament evolution [104]

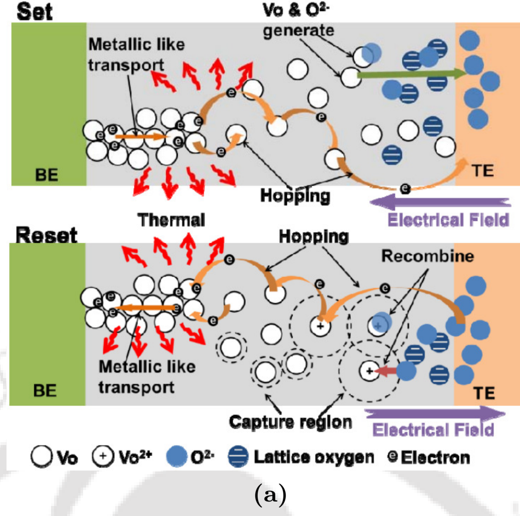
2.3.2 Transport mechanism

The switching characteristics are strongly connected with the geometry of CF that is created because of the generation and recombination of the oxygen vacancies (V_o) in the oxide layer [103], [104]. The physical process of RRAM device operation is shown in Fig. 2.2 [104]. In the SET process, the generation of V_o and the drift of oxygen ions (O^{2-}) to the top electrode forms CF, which connects both the top and bottom electrodes. It results in RRAM switching to the low resistance state (LRS) from the high resistance state (HRS). In the RESET process, the recombination between O^{2-} and V_o ruptures CF, and RRAM is switched to the high resistance state (HRS).

2.3.3 Model equations

Eq. 2.1 describes the generation probability of oxygen ions. The probability of O^{2-} ions hopping from the electrode to the dielectric layer and the probability of O^{2-} ions hopping within the dielectric layer are defined by Eq. 2.2 and Eq. 2.3. Whereas Eq. 2.4 defines the probability of recombination of

TH-3339_186102005



(a)

Figure 2.2: Physical process of resistive switching used in the model [104]

O^{2-} and V_o. The transmission probability of the electrons between two V_o is calculated using Eq. 2.5. Finally, the conductivity of the conductive filament can be calculated by Eq. 2.6. The variables used in these equations are specified in Table- 2.2 [105].

$$P_g(E,T,dt) = fdtexp^{\left(-\frac{E_a - \alpha_a ZeE}{k_B T}\right)} \tag{2.1}$$

$$P_m(V,T,dt) = fdtexp^{\left(-\frac{E_i - \gamma ZeV}{k_B T}\right)} \tag{2.2}$$

$$P_h(E,T,dt) = fdtexp^{\left(-\frac{E_h - \alpha_h ZeE}{k_B T}\right)} \tag{2.3}$$

$$P_r(T,dt) = fdtexp^{\left(-\frac{\Delta E_h}{k_B T}\right)} \tag{2.4}$$

$$W_{m \to n} = f_{ph} exp(-2\alpha R_{mn} - E_{mn}/k_B T) \tag{2.5}$$

$$\sigma = \sigma_0 exp\left(\frac{E_{AC}}{k_B T}\right) \tag{2.6}$$

Eq. 2.7-Eq. 2.9 describes the electrical behaviour of the device. Eq. 2.7 depicts the current due to free oxygen ions, whereas Eq. 2.8 and Eq. 2.9 states the current due to e^- hopping into oxygen

TH-3339_186102005

vacancies during SET and RESET operations. The total current through the RRAM is modeled by Eq. 2.10. The dies of the geometry are described using Eq. 2.11, which describes the growth of the conductive filament in the horizontal direction, and Eq. 2.12 describes the growth of the conductive filament in the vertical direction.

$$I_1 = I_0 \pi \left(\frac{WCF^2}{4} - \frac{w^2}{4} \right) exp \left(\frac{-L_0}{X_t} \right) sinh \left(\frac{V_{tb}}{V_t} \right) \tag{2.7}$$

$$I_2 = I_0 \pi \frac{w^2}{4} exp \left(\frac{-x}{X_t} \right) sinh \left(\frac{V_g}{V_t} \right) \tag{2.8}$$

$$I_2 = \frac{V_{tb}}{\frac{\rho L_0}{\pi w^2/4}} \tag{2.9}$$

$$I_{tb} = I_1 + I_2 \tag{2.10}$$

$$dx = -afexp \left(-\frac{E_a - V_g \alpha Z/x}{k_B Temp} \right) \tag{2.11}$$

$$dw = weff + pow(Weff, 2))fexp \left(-\frac{E_a - V_{tb} \alpha Z/L_0}{k_B Temp} \right) \tag{2.12}$$

2.3.4 Simulation results

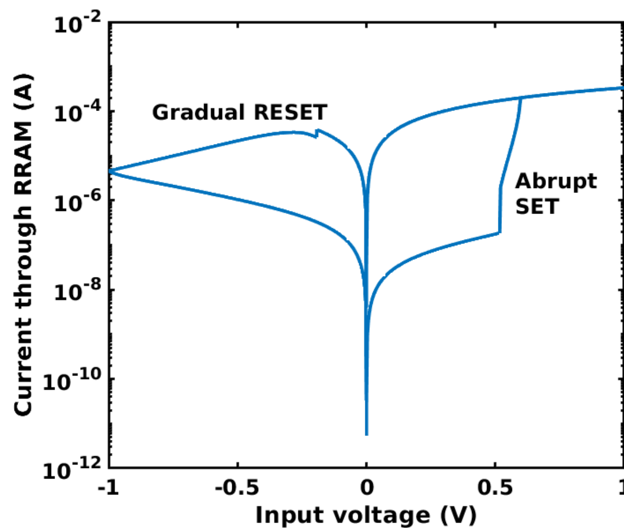


Figure 2.3: RRAM $I - V$ characteristics

TH-3339_186102005

Table 2.2: Simulation parameters for Peking RRAM Verilog-A model

Parameters	Description	Value
I_0	Hopping current density	$1 \times 10^3\ A/m^2$
ρ	Resistivity of the CF	$19.635\ \mu\Omega m$
a	Distance between oxygen vacancies (V_o)	$0.25\ nm$
f	Vibration frequency of V_o	$10^{13}\ Hz$
E_a	Average active energy of V_o	$0.7\ eV$
E_h	Hopping barrier of oxygen ion O^{2-}	$1.12\ eV$
E_i	Electrode/oxide interface	$0.82\ eV$
$\alpha_a \& \alpha_h$	Energy enhancement factor	$0.75\ nm$
γ	Voltage enhancement factor	1.5
Z	Charge number	2
R_{th}	Effective thermal resistance	$5 \times 10^5 K/W$
R_H	Oxide parasitic resistance	$2 \times 10^9 \Omega$
R_L	Electrode contact resistance	$20\ \Omega$
C_p	Electrode parasitic capacitance	$1\ fF$
L_0	Initial switching layer length	$5\ nm$
x_0	Initial gap length	$L_0, 0$
WCF	Switching layer width	$5\ nm$
$Weff$	Effective width	$0.5\ nm$
w_0	Initial CF width	$0.5\ nm, WCF$
X_t	Characteristic gap length	$0.4\ nm$
V_t	Characteristic voltage	$0.4\ V$
$T0$	Ambient temperature	$300\ K$

Fig. 2.3 exhibits $I-V$ characteristics of RRAM when it is excited with a quasi-DC voltage sweep from $0\ V \rightarrow 1\ V \rightarrow 0\ V$ for SET and $0\ V \rightarrow -1\ V \rightarrow 0\ V$ for RESET operation. A full-length conducting filament ($x \rightarrow 0$) is formed between electrodes when a particular voltage (SET voltage) is applied across RRAM. This is called SET operation and is disruptive. This leads to an abrupt flow of current in RRAM as it enters to low resistance state (LRS). Similarly, RRAM stops conducting fully when a specific voltage (RESET voltage) is applied across its electrodes, and $x \rightarrow x_0$. This steers RRAM to enter the high resistance state (HRS), having a very low current flowing between electrodes.

2.3.5 Effects of individual parameter on $I-V$ characteristics

We vary one parameter at a time to study the effects of various parameters on the $I-V$ characteristics of the Verilog-A model. The arrow in all the $I-V$ characteristics indicates the direction in which a parameter is increased.

L_0 is the initial fixed length of the RRAM switching layer. The accepted range of L_0 variation is $1\ nm$ to $10\ nm$. Fig.2.4 shows the characteristics of RRAM for different values of L_0. Parameter L_0

TH-3339_186102005

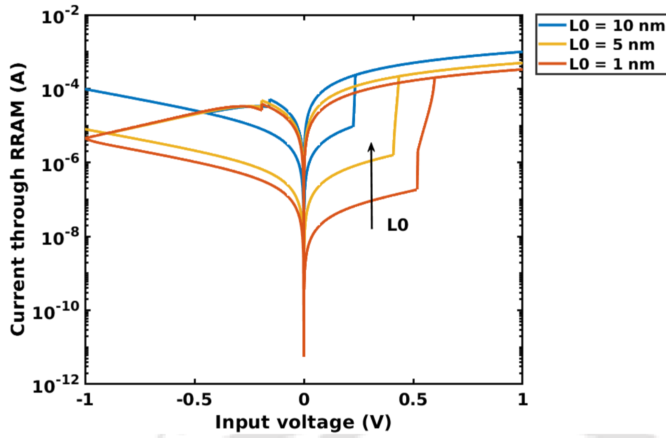


Figure 2.4: $I-V$ Characteristics of RRAM for varying L_0

contributes to the change in the dies of the geometry as well as the electrical behaviour of RRAM. Thus, as expected, there are huge variations in the characteristics with respect to L_0. It also decreases the SET voltage. For $L_0 = 10\ nm$, the SET voltage is equal to $0.2\ V$. It is observed that the SET voltage is reduced, but the ratio of on-state resistance to off-state resistance is low, which inhibits RRAM from being used in any application.

Similarly, when $L_0 = 1\ nm$, the SET voltage equals $0.5\ V$, but the on-state resistance to off-state resistance ratio is high. The on-resistance to off-resistance ratio is acceptable when $L_0 = 5\ nm$.

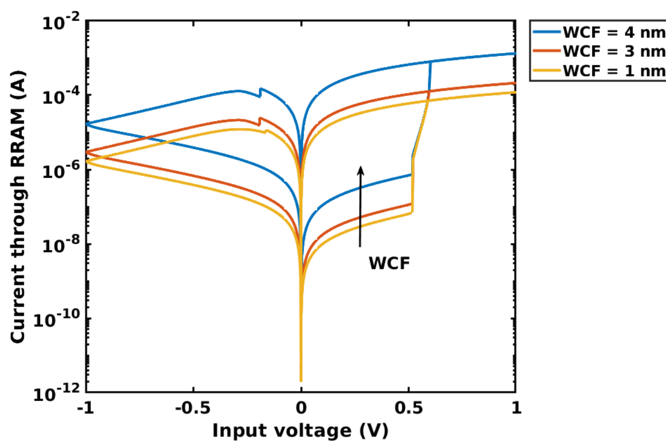


Figure 2.5: $I-V$ Characteristics of RRAM for varying WCF

WCF is the fixed width of the RRAM switching layer. Fig.2.5 shows characteristics of RRAM for different values of WCF. The parameter WCF contributes to the change in the dies of the geometry as well as the electrical behaviour of RRAM. Similar to L_0, the current increases as WCF increase.

TH-3339_186102005

Here, we can observe that the on-resistance to off-resistance ratio is almost the same for all the values of WCF, but the resultant current is less when $WCF = 1\ nm$. This parameter can be adjusted to lower the current for low-power operations.

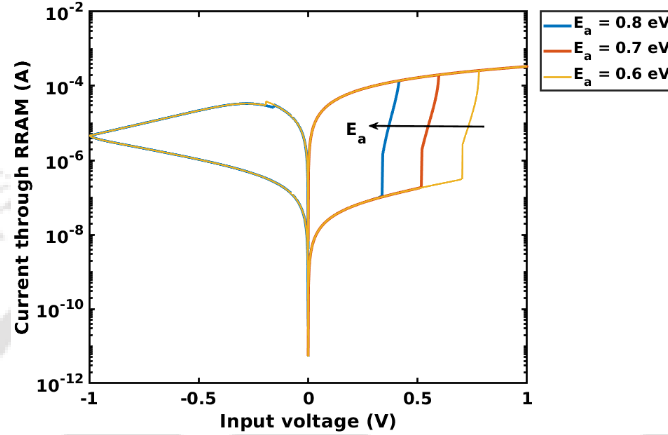


Figure 2.6: $I-V$ Characteristics of RRAM for varying E_a

Parameter E_a controls the rate of oxygen vacancy generation during the SET process. In the model, increasing E_a leads to an abrupt current increase at a lower voltage. Parameter E_a controls the rate of the electrode releasing oxygen ions O^{2-}, which is the first step of the RESET process. The first stage sustains for a short time when the voltage starts rising. Parameter E_a is critical to the recombination rate between oxygen vacancies and O^{2-}, the second stage during the RESET process, impacting the voltage and current during RESET. For $E_a = 0.6\ eV$, the SET voltage is equal to 0.6 V. Whereas, for $E_a = 0.7\ eV$, the SET voltage equals 0.5 V, and for $E_a = 0.8\ eV$, the SET voltage equals 0.4 V. Fig. 2.6 shows the $I-V$ characteristics of RRAM for different values of E_a.

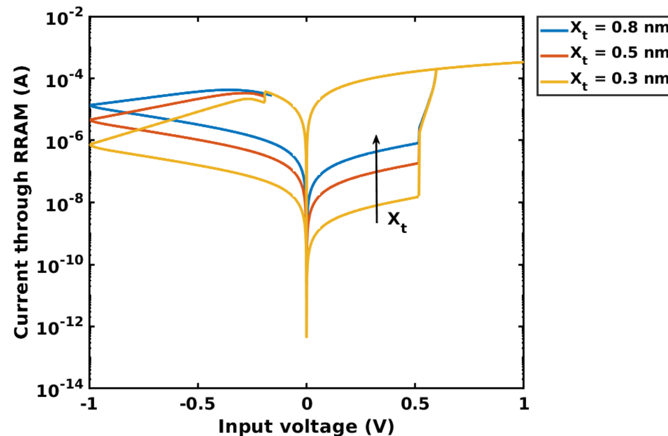


Figure 2.7: $I-V$ Characteristics of RRAM for varying X_t

TH-3339_186102005

X_t is the characteristic gap ranging between 0.3 nm to 0.8 nm. Increasing X_t results in an increased initial current, keeping the SET voltage unaffected. By varying X_t, the on-to-off resistance ratio can be controlled. The highest on-to-off resistance ratio is achieved for $X_t = 0.3\ nm$, while it is the lowest when $X_t = 0.8\ nm$. Fig. 2.7 shows the characteristics of RRAM for different values of X_t.

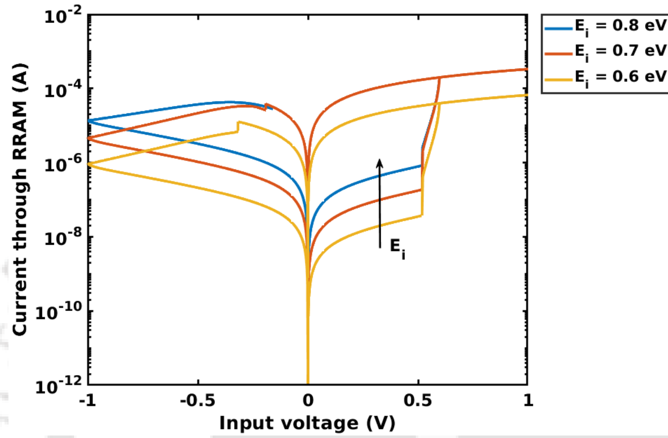


Figure 2.8: $I-V$ Characteristics of RRAM for varying E_i

E_i is the electrode-to-oxide interface. It can be observed in Fig. 2.8, as the value of E_i increases, it escalates the RRAM current in the HRS state. E_i is varied from 0.6 eV to 0.8 eV, and a significant rise in current can be seen. This parameter can be adjusted to obtain appropriate currents according to the circuit requirements.

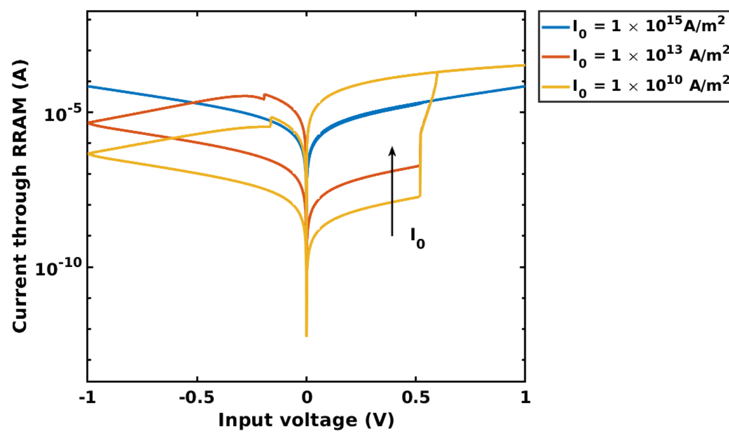


Figure 2.9: $I-V$ Characteristics of RRAM for varying I_0

I_0 is the hopping current density in the gap region, varying from $1 \times 10^{10} A/m^2$ to $1 \times 10^{15} A/m^2$. However, for larger values of I_0, attenuation is observed in the characteristics. No loop is obtained for

$I_0 = 1 \times 10^{15}$, whereas, for smaller values of I_0, the on-to-off resistance ratio is larger. For $I_0 = 1 \times 10^{10}$, the on-to-off resistance ratio is the largest, whereas, for $I_0 = 1 \times 10^{15}$, the on-to-off resistance ratio is the least. Thus, I_0 can be adjusted accordingly for an appropriate on-to-off resistance ratio. Fig. 2.9 illustrates the characteristics of RRAM for different values of I_0.

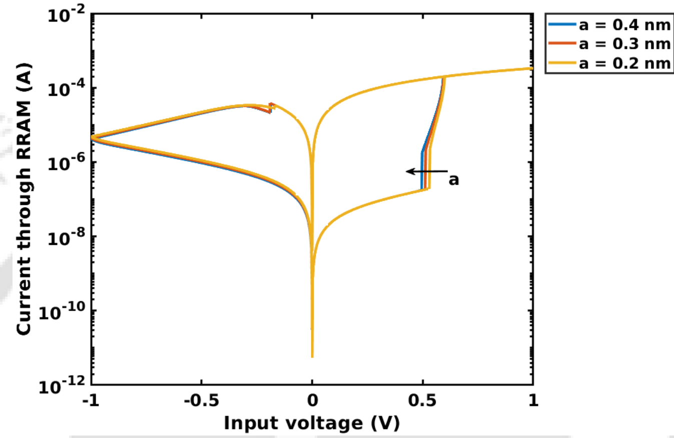


Figure 2.10: $I - V$ Characteristics of RRAM for varying a

The distance between adjacent oxygen vacancies (V_o) is denoted by a. Variation in parameter a does not cause any significant changes in the $I - V$ characteristics for a selected set of parameters, except for the minor variations in the SET voltage. Increasing a causes the SET voltage to decrease slightly. Fig.2.10 showcases the $I - V$ characteristics of RRAM for different values of a.

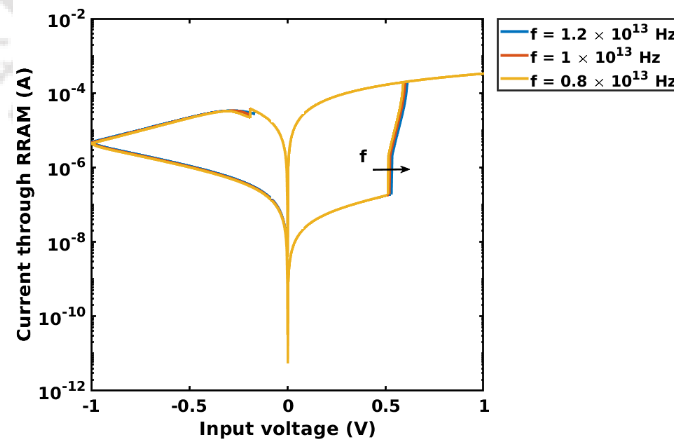


Figure 2.11: $I - V$ Characteristics of RRAM for varying f

The vibration frequency of an oxygen atom is indicated by f. For $f = 1.2 \times 10^{13}$ Hz, the SET voltage is slightly larger than the SET voltage at $f = 0.8 \times 10^{13}$ Hz. Fig. 2.11 shows the characteristics

TH-3339_186102005

of RRAM for different values of f.

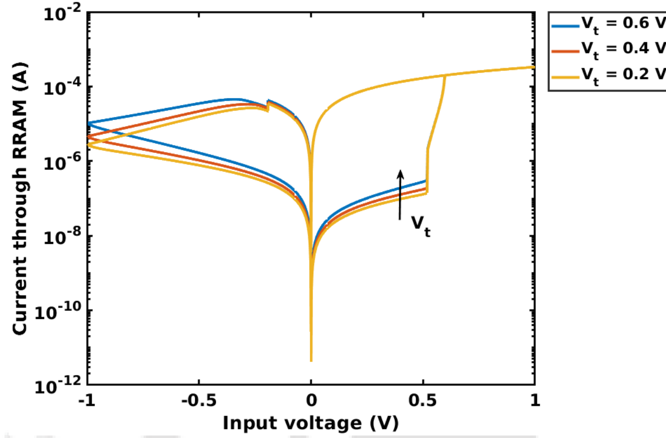
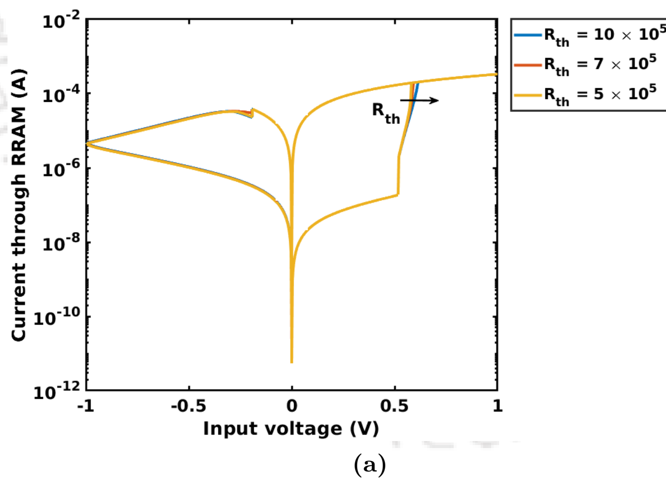


Figure 2.12: $I-V$ Characteristics of RRAM for varying V_t

V_t is the characteristic voltage. It does not contribute to the die of the geometry, therefore, it has less effect on the RRAM current. Increasing V_t from 0.2 V to 0.6 V increases the RESET current, as shown in Fig.2.12. This parameter can be adjusted accordingly to obtain less RRAM current.



(a)

Figure 2.13: $I-V$ Characteristics of RRAM for varying R_{th}

R_{th} is the effective thermal resistance. Varying R_{th} does not have much effect on the characteristics, except for a slower SET process for a higher R_{th}. Fig. 2.13 exhibits the characteristics of RRAM for different values of R_{th}.

2.4 Unimore RRAM Verilog-A model

2.4.1 Device structure

The Unimore RRAM Verilog-A model is designed to simulate the behaviour of bipolar metal oxide, $TiN/Ti/HfO_2/TiN$, RRAM devices. The model is based on the physics of these devices and includes non-ideal effects, such as cycle-to-cycle and device-to-device variations and random telegraph noise, which can be turned on or off through model and simulation parameters. This model considers the conductive filament (CF) growth and barrier components that add to the total resistance of the device [106]. The model can produce the DC and pulsed characteristics using a single set of parameters. Fig. 2.14 illustrates the structure of RRAM. Here, x is the barrier thickness initialized to 4.2 nm before the start of SET process, and t_{ox} is the maximum thickness of the CF.

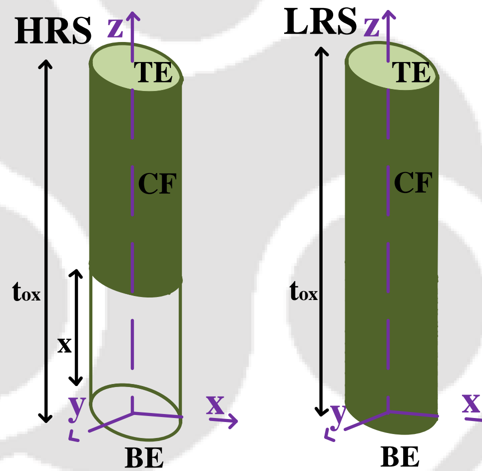


Figure 2.14: Device structure [106]

2.4.2 Transport mechanism

The transport mechanism modeled by the Unimore Verilog-A model can be explained with the help of Fig. 2.15. The RRAM is in HRS during the initial stage, generating no oxygen vacancies. When a positive ramp input is applied across the RRAM, the current slowly rises with an increase in the input voltage. As soon as the voltage reaches the SET voltage, the HfO_2 bond stretching is induced by the high electric field. This lowers the energy required to break the HfO_2 bond and creates oxygen vacancies.

The newly generated oxygen vacancies support the trap-assisted-tunneling (TAT) transport, which increases the current and the local temperature. This subsequently triggers a thermally driven positive

TH-3339_186102005

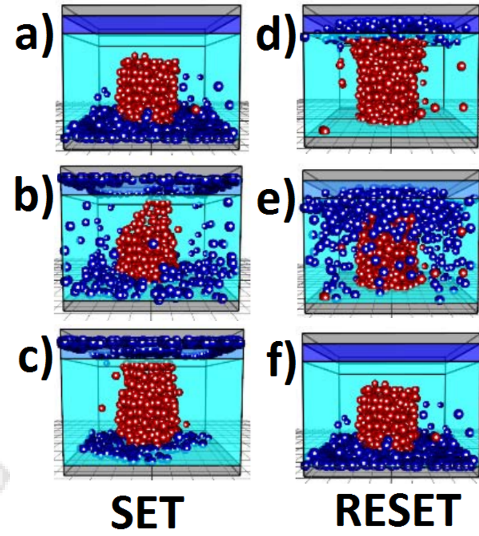


Figure 2.15: Evolution of oxygen ions (blue spheres) and vacancies (red spheres) during SET and RESET [107]

feedback process that leads to the formation of the conductive filament (CF), and the device is said to be in the low resistance state (LRS), as shown in Figs. 2.15 (a-c). A negative voltage needs to be applied across the RRAM to change the state from LRS to HRS. When a negative ramp voltage is applied, the ions accumulated in the HfO_2 layer flow back towards the CF, recombining with the oxygen vacancies. This partially oxidizes the conductive filament, which results in the formation of the oxide barrier and increases RRAM resistance. Thus, the RRAM is said to be in the HRS, as illustrated in Fig. 2.15 (d-f).

2.4.3 Model equations

The model equations consider the quasi-ohmic charge transport in the conductive filament (CF) and the trap-assisted tunneling transport in the dielectric barrier. The dielectric barrier dynamics are modeled with differential equations Eq. 2.18 and Eq. 2.19. The model considers the field-driven oxygen ion drift during RESET and the field and temperature accelerated bond breakage during SET [107]. The temperature dynamics of the CF and the barrier are modeled with two differential equations, Eq. 2.22 and Eq. 2.23, which enable accurate predictions when the device is driven with very short pulses. The effects of die structure, temperature dynamics, and the generation and rupture of the CF are defined by Eq. 2.13 - Eq. 2.23 [106], [108]. The parameters used in the proposed study are presented in Table. 2.3.

TH-3339_186102005

$$R_{LRS} = \frac{\rho \times t_{ox}}{S} \tag{2.13}$$

$$R_{cf} = R_{LRS}.\frac{t_{ox} - x}{t_{ox}}.[1 + \alpha.(T_{cf} - T_{meas})] \tag{2.14}$$

$$R_{bar} = R_{LRS}.\beta.(e^{\frac{x}{l}}).\frac{E_a}{e^{k_B}T_{bar}} \tag{2.15}$$

$$R = R_{cf} + R_{bar} \tag{2.16}$$

$$I = \frac{V_0}{R} sinh(\frac{V}{V_0}) \tag{2.17}$$

$$\frac{dx}{dt} = c_0 e^{\frac{E_{ad} - (g - ax^b)\frac{V}{t_{ox}}}{k_B T_{cf}}} \tag{2.18}$$

$$\frac{dx}{dt} = -x c_0 e^{\frac{E_{ag} - gg\frac{V}{x}}{k_B T_{cf}}} \tag{2.19}$$

$$V_{cf} = V.\frac{R_{cf}}{R_{cf} + \frac{R_b}{0.5.e^{\frac{V}{V_0}}}} \tag{2.20}$$

$$V_{bar} = V.\frac{0.5.e^{\frac{V}{V_0}}}{R_{cf} + \frac{R_B}{0.5.e^{\frac{V}{V_0}}}} \tag{2.21}$$

$$\frac{dT_{pcf}}{dt} = C_{cf}^{-1}[V_{cf}.I - k_{cf}(T_{cf} - T_0) - k_{ex}(T_{cf} - T_{bar})] \tag{2.22}$$

$$\frac{dT_{bar}}{dt} = C_{pbar}^{-1}[V_{bar}.I - k_{bar}(T_{bar} - T_0) - k_{ex}(T_{bar} - T_{cf})] \tag{2.23}$$

2.4.4 Equation modeling the RTN and variation

Random Telegraph Noise (RTN) causes fluctuations in the device current, leading to errors while reading the device state. Two physical mechanisms are responsible for RTN when the device is in HRS or LRS [109] [110]. When the device is in HRS, RTN current fluctuations are caused due to the temporary de-activation of the (V_o^+) defects. The de-activation is caused due to charge de-trapping in the slow defects that do not participate in the charge transport. In LRS, RTN is caused by charge trapping in defects located near the CF. The charges trapped at such defects disturb the potential in their surroundings, causing a screening effect on the portion of the CF close to them, which causes a change in the resistance of the device.

In the Unimore Verilog-A model, the defects that cause cycle-to-cycle and device-to-device vari-

TH-3339_186102005

Table 2.3: RRAM model parameter

Parameter	Description	Value
ρ	Oxide material resistivity	$3000\ \Omega.nm$
t_{ox}	Oxide layer thickness	$12\ nm$
S_0	Initial conductive filament section	$72\ nm^2$
E_a	Activation energy of the trap assisted tunneling	$0.12\ eV$
T_0	Ambient temperature	$300\ K$
l	Typical tunneling length	$0.42\ nm$
V_0	HRS current non-linearity factor	$0.30\ V$
α	Resistivity temperature coefficient	$0.002\ K^{-1}$
β	Barrier resistance fitting parameter	1×10^{-3}
c_0	Bond vibration frequency	$5 \times 10^{13} Hz$
C_{pb}	Barrier thermal capacity	$4 \times 10^{-9} J/K$
C_{pcf}	CF thermal capacity	$5 \times 10^{-13} J/K$
k_{bar}	Barrier thermal conductivity	$5 \times 10^{-5} W/K$
k_{cf}	CF thermal conductivity	$1 \times 10^{-6} W/K$
k_{ex}	Barrier/CF mutual thermal conductivity	$0\ W/K$
E_{ad}	Diffusion activation energy of oxygen ions	$4.4\ eV$
g	Field enhancement factor for oxygen ions diffusion	$54\ e.nm$
a	RESET curve slope fitting parameter	14
b	RESET curve fitting parameter	0.4
E_{ag}	Bond breaking activation energy	$1.2\ eV$
gg	Field enhancement factor for bond breaking	$1.75\ e.nm$
$xinit$	Initial barrier thickness	$4.3\ nm$
$Tinit$	Initial device temperature	$300\ K$
$Tmeas$	Temperature at which R_{LRS} is measured	$300\ K$
$min_time_step_vpos$	Minimum time step (for positive applied voltages)	$1 \times 10^{-13} s$
$min_time_stop_vneg$	Minimum time step (for negative applied voltages)	$1 \times 10^{-13} s$
$tstep_param$	Adaptive time step parameter	$1 \times 10^{2} s$

ations are distributed along the barrier when the CF is growing, and outside the CF when the CF collapses. Each defect is associated with a random vertical distance from the bottom electrode, the resistance variation, and the device's initial state. The RTN current contribution due to such defects is added to the RRAM current. The Unimore Verilog-A model incorporates the cycle-to-cycle and device-to-device variations using Eq. 2.24 - Eq. 2.29 [108]. The RTN parameters utilized to obtain required characteristics are illustrated in Table. 2.4.

$$E_{rel_{O_i}} = E_{rel0_O} + U(-\Delta E_{rel_O}, \Delta E_{rel_O}) \tag{2.24}$$

$$E_{t_{O_i}} = E_{t0_O} + U(-\Delta E_{t_O}, \Delta E_{t_O}) \tag{2.25}$$

$$\Delta R_i = R_{Bar}.var_{ln} \tag{2.26}$$

$$E_{rel_{V_i}} = E_{rel_{0_V}} + U(-\Delta E_{rel_V}, \Delta E_{rel_V}) \tag{2.27}$$

$$E_{t_{V_i}} = E_{t_{0_V}} + U(-\Delta E_{t_V}, \Delta E_{t_V}) \tag{2.28}$$

$$\Delta R_i = R_{Bar}.var_{ln} \tag{2.29}$$

Table 2.4: RTN Parameters

Parameter	Description	Value
$const0$	Capture and emission times constant	$4.19 \times 10^{-32} J.m^3/s$
$max_defects$	Maximum number of defects that can be generated	150
$E_{rel_{0_O}}$	Nominal oxygen ions relaxation energy	$2.67\ eV$
Nc	Density of states at the bottom of the conduction band	$2.42 \times 10^{45} Jm^3$
$E_{rel_{0_V}}$	Nominal oxygen vacancies relaxation energy	$1.19\ eV$
$E_{t_{0_V}}$	Nominal oxygen vacancies thermal ionization energy	$2.1\ eV$
ϕ	Energy barrier for injected electrons	$2.1\ eV$
λ_c	Typical tunneling length (capture)	$2 \times 10^{-10} m$
λ_e	Typical tunneling length (emission)	$2 \times 10^{-10} m$
$E_{t_{0_O}}$	Nominal oxygen ions thermal ionization energy	$2.3\ eV$
ΔE_{rel_O}	Spread of the oxygen ions relaxation energy distribution	$0.4\ eV$
ΔE_{t_O}	Spread of the oxygen ions thermal ionization energy distribution	$0.5\ eV$
ΔE_{rel_V}	Spread of the oxygen vacancies relaxation energy distribution	$0.4\ eV$
ΔE_{t_V}	Spread of the oxygen vacancies thermal ionization energy distribution	$0.5\ eV$
ΔHRS_{mean}	Mean of the normal distribution associated to the log-normal distribution of the R_{HRS} due to RTN.	$ln(0.5)$
ΔHRS_{std}	Standard deviation of the normal distribution associated to the log-normal distribution of the R_{HRS} due to RTN.	0.6
RTN_ON	Parameter used to switch on the RTN module ($0 = OFF, 1 = ON$)	0
rand_seed	Initial random seed value	1
$dxdt_th$	Threshold on the barrier derivative to randomly reassign defects positions	$1 \times 10^{-10} m/s$

TH-3339_186102005

2.4.4.1 Simulation results

Fig. 2.16 showcases RRAM's $I-V$ characteristics obtained when an RRAM is excited with a quasi-DC voltage sweep from $0\ V \rightarrow 1.2\ V \rightarrow 0\ V$ for the SET and $0\ V \rightarrow -1.2\ V \rightarrow 0\ V$ for the RESET operations. It can be seen that the SET operation is abrupt due to sudden CF forming, whereas the RESET operation is gradual due to the recombination of the oxygen ions. It can be observed that the SET voltage of the RRAM is $0.8\ V$, and the RESET voltage is $-0.8\ V$.

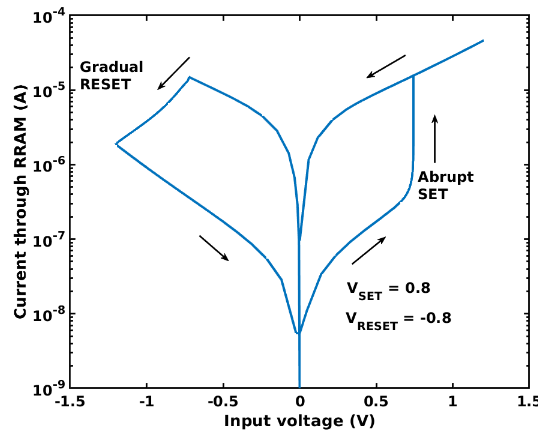


Figure 2.16: $I-V$ Characteristics of Unimore Verilog-A model

2.4.5 Effects of individual parameter on $I-V$ characteristics

To analyze the effects of all individual parameters on the behaviour of the device, the $I-V$ characteristics are studied by varying a single parameter at a time, keeping other parameters constant. The arrow indicates the direction of increasing parameter values.

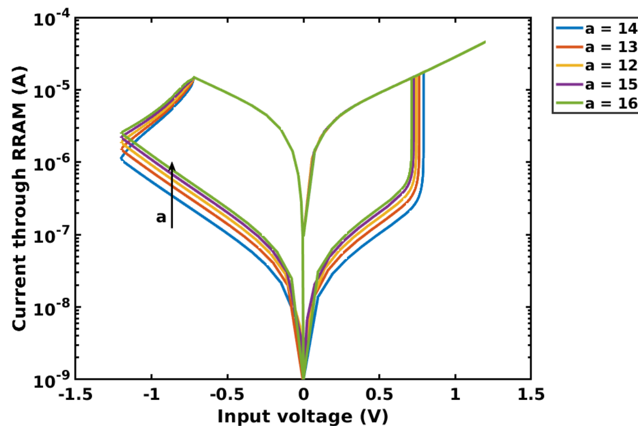


Figure 2.17: $I-V$ Characteristics of RRAM for varying a

TH-3339_186102005

Parameter a is the RESET curve fitting parameter, which models the relationship between the barrier growth rate, the applied voltage, and the barrier thickness. The slope of the RESET curve is influenced by a. It is changed from 13 to 16 and is observed that the RESET curve becomes steeper. This also results in a slight decrease in the SET voltage, as depicted in Fig. 2.17.

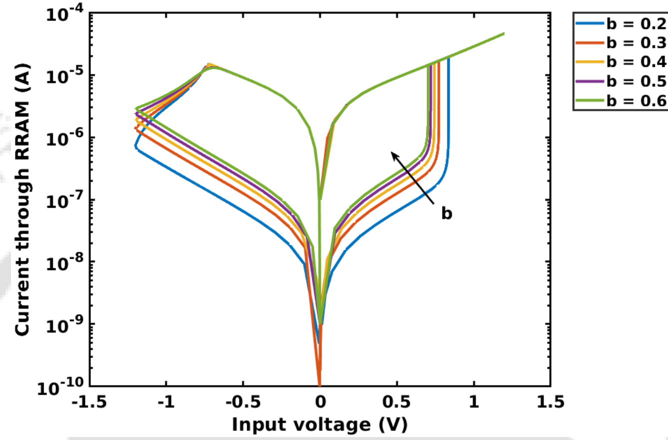


Figure 2.18: $I - V$ Characteristics of RRAM for varying b

Parameter b is the SET curve fitting parameter, which can be tuned to adjust the slope and curvature of the SET characteristics. It is varied in between $0.2 - 0.6$. It can be observed in Fig. 2.18 that increasing b changes the curvature of the $I - V$ characteristics and decreases the SET voltage.

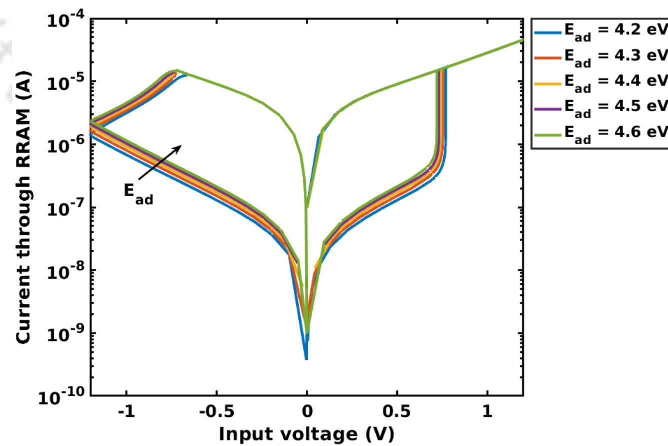


Figure 2.19: $I - V$ Characteristics of RRAM for varying E_{ad}

The parameter E_{ad} is the diffusion activation energy of oxygen ions, which can be altered from 4.2 eV to 4.6 eV. Increasing E_{ad} decreases the voltage at which the barrier starts to grow. Therefore,

TH-3339_186102005

it can be seen in Fig. 2.19 that as E_{ad} is increased, the SET voltage is reduced. This parameter can be tuned to obtain an appropriate SET voltage. To obtain the required functionality, E_{ad} is set to 4.4 eV.

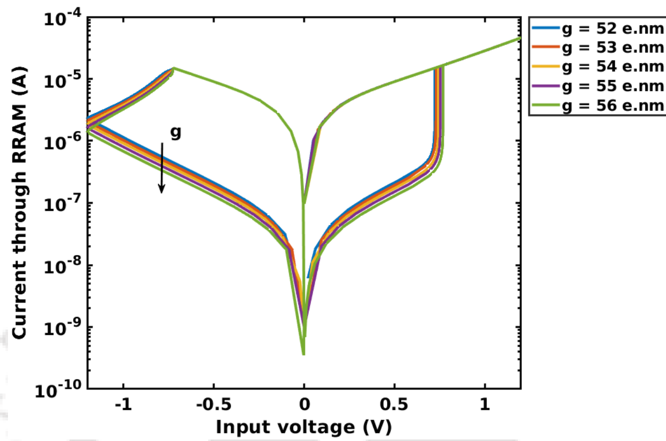


Figure 2.20: $I-V$ Characteristics of RRAM for varying g

The parameter g is the field enhancement factor for oxygen ions diffusion. It boosts the effects of the applied voltage, leading to faster barrier growth. As depicted in Fig. 2.20, raising the value of g from 52 $e.nm$ to 56 $e.nm$ results in RESET at lower RRAM currents due to faster barrier growth. However, increasing g causes a slight increase in the SET voltage.

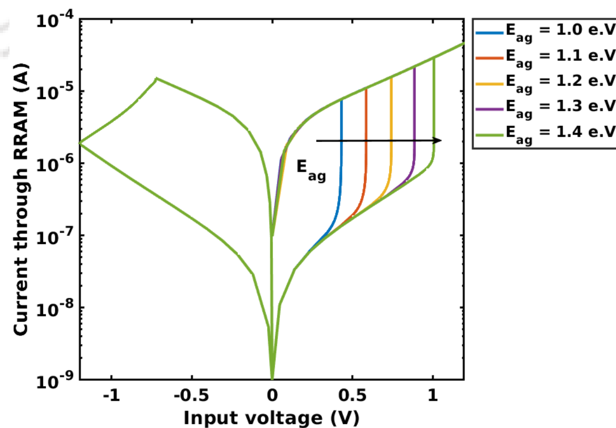


Figure 2.21: $I-V$ Characteristics of RRAM for varying E_{ag}

The parameter E_{ag} is the bond-breaking activation energy influencing the SET process. The default value set for this parameter is 1.2 eV. It can be observed in Fig. 2.21, as E_{ag} increases from 1.0 eV to 1.4 eV, the SET voltage is also increased from 0.3 $V-1$ V, but the HRS resistance remains unaffected. Hence, if it is needed to increase the SET voltage, E_{ag} can be elevated to obtain the

required SET voltage value while keeping the HRS intact.

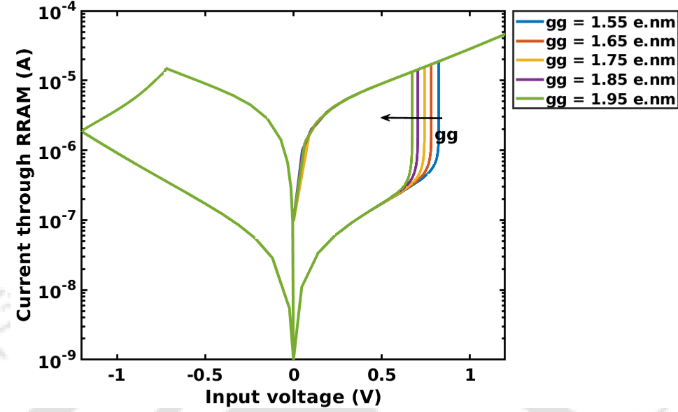


Figure 2.22: $I - V$ Characteristics of RRAM for varying gg

The parameter gg is the field enhancement factor for bond breaking that influences the SET process. The default value for this parameter is 1.75 $e.nm$. Increasing gg enhances the field dependence of the SET process. Therefore, as depicted in Fig. 2.22, increasing the value of gg from 1.55 $e.nm$ to 1.95 $e.nm$ results in the decrease of the SET voltage of the RRAM from 0.8 $V - 0.6$ V. This parameter can be effectively used to tune the SET voltage without changing the $I - V$ characteristics and the SET and RESET currents.

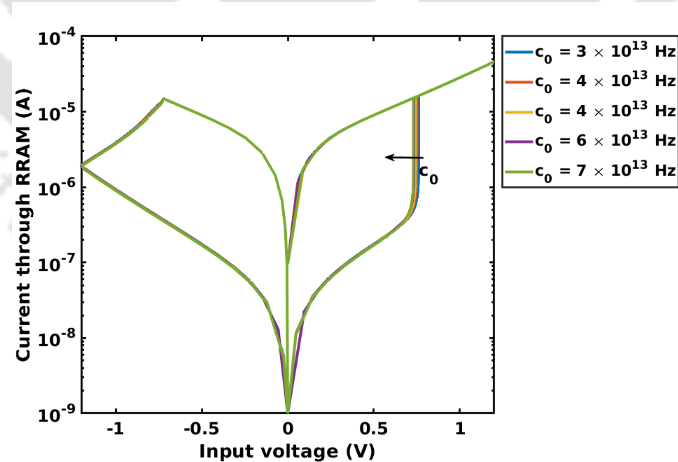


Figure 2.23: $I - V$ Characteristics of RRAM for varying c_0

The parameter c_0 is the bond vibration frequency affecting the barrier variation rate. It can be observed in Fig. 2.23 that increasing c_0 from $3\ \times 10^{13} Hz - 7\ \times 10^{13} Hz$ while keeping all the other parameters constant reduces the SET voltage slightly.

The parameter V_0 is the HRS current nonlinearity factor. As shown in Fig. 2.24, when V_0 is

TH-3339_186102005

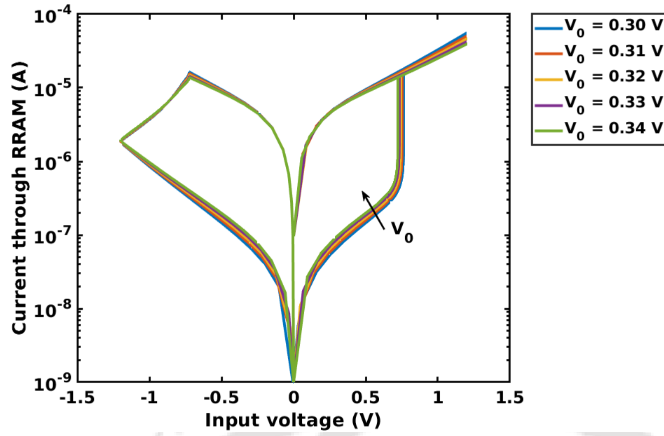


Figure 2.24: $I-V$ Characteristics of RRAM for varying V_0

increased from 0.30 V to 0.34 V, the current through the RRAM in LRS reduces, and the slope of the RESET curve becomes nonlinear. V_0 is set to 0.30 V to obtain the minimum nonlinearity in the RRAM current.

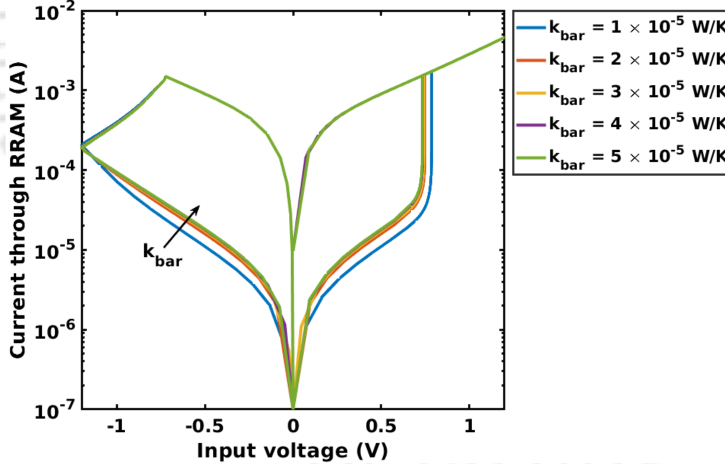


Figure 2.25: $I-V$ Characteristics of RRAM for varying k_{bar}

The parameter k_{bar} is the barrier thermal conductivity influencing the temperature dynamics of RRAM. Lower temperatures lead to smaller barriers, smoother SET transitions, and more RESET current. It can be observed in Fig. 2.25 that when k_{bar} is increased from 1 $\times 10^{-5} W/K$ to 5 $\times 10^{-5} W/K$, the SET transition becomes smoother. This parameter can be adjusted appropriately if a drastic SET operation is needed.

The parameter K_{cf} is the thermal conductivity of the conductive filament. It affects the die geometry of the conductive filament. Smooth RESET transitions can be obtained by increasing K_{cf},

TH-3339_186102005

as exhibited in Fig.. 2.26.

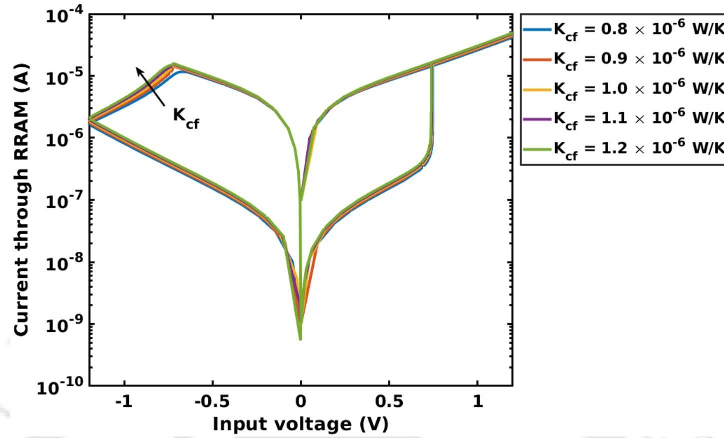


Figure 2.26: $I-V$ Characteristics of RRAM for varying K_{cf}

2.5 Parallel and series combination of RRAM

RRAM connected in different combinations can produce characteristics that can be used to obtain certain functionalities in a circuit. Therefore, we study the parallel and series combinations of RRAMs in detail and verify their characteristics with contemporary works.

2.5.1 Parallel combination

When $RRAM1$ and $RRAM2$ are connected in parallel, Fig. 2.27 (a), the resultant resistance is expected to reduce, as shown in Fig. 2.28. The $I-V$ characteristics are obtained by connecting the top electrode of both the RRAMs to the positive terminal, and the bottom electrodes to the ground terminal, as illustrated in Fig. 2.27 (a). Initially, both the RRAMs are in the RESET state. When a positive voltage is applied, these RRAMs are SET simultaneously.

Fig.2.27 (b) exhibits the $I-V$ characteristics for the parallel combination. The orange curve shows the $I-V$ characteristics when $RRAM1$ and $RRAM2$ are connected in parallel, whereas the blue curve illustrates the $I-V$ characteristics of a single RRAM. Fig.2.28 depicts the characteristics discussed in [111], where the current characteristics of the parallel memristors lie above the individual characteristics of $M1$ and $M2$ as expected.

TH-3339_186102005

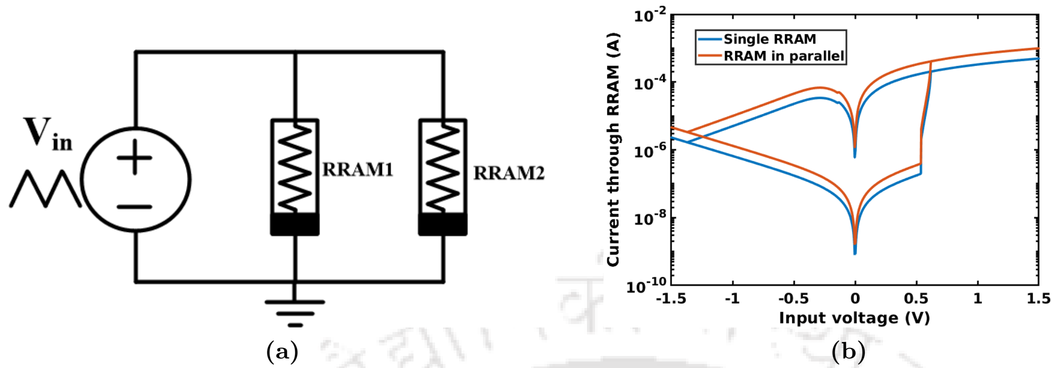


Figure 2.27: (a) RRAM connected in parallel combination (b)RRAM parallel connection obtained $I-V$ characteristics

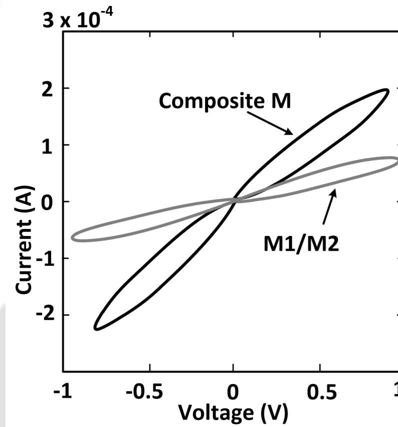


Figure 2.28: *RRAM parallel connection expected $I-V$ characteristics [111]*

2.5.2 Series combination

When RRAMs are connected in series in the same direction, as shown in Fig.2.29 (a), it acts as a voltage divider circuit. Initially, $RRAM1$ and $RRAM2$ both are RESET. If the HRSs of RRAM1 and RRAM2 are different, the voltage across $RRAM2$ can be calculated by using Eq.2.30. However, if the HRSs of $RRAM1$ and $RRAM2$ are same, then the voltage across $RRAM2$ can be obtained by Eq.2.31. The current through the series combination is expected to decrease compared to a single RRAM Fig.2.29 (b). When the input pulse is applied across the series combination, both RRAMs are SET simultaneously. The $I-V$ characteristics of this series combination match with the expected characteristics shown in Fig. 2.30.

$$V_{RRAM2} = Vin\frac{RRAM2}{RRAM1 + RRAM2} \tag{2.30}$$

$$V_{RRAM2} = Vin/2 \tag{2.31}$$

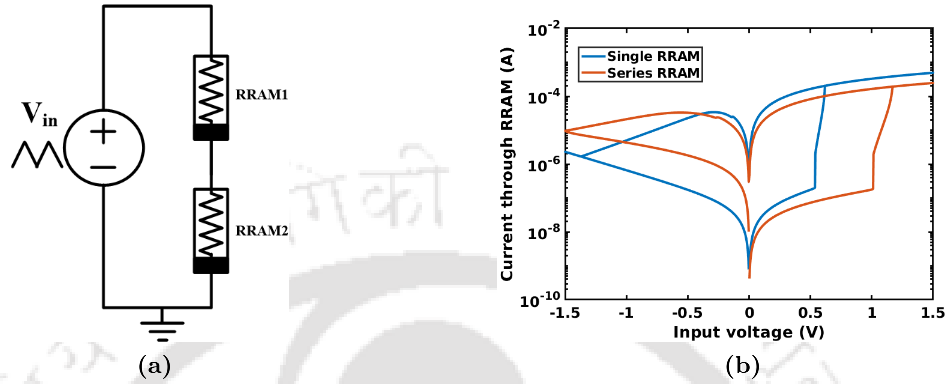


Figure 2.29: (a) RRAM connected in series combination (b) RRAM series connection obtained $I-V$ characteristics

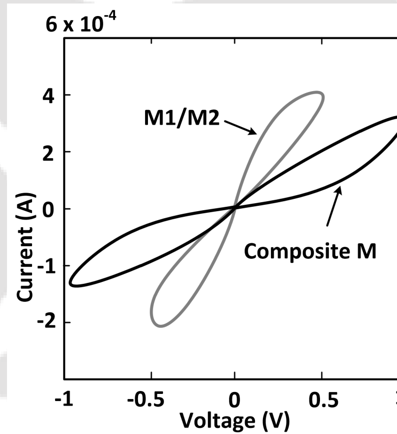


Figure 2.30: *RRAM series connection expected $I-V$ characteristics [111]*

Another way of connecting RRAMs in series is to put them together in the opposite direction, as shown in Fig.2.31 (a). When a positive voltage is applied across this combination, $RRAM1$ is SET. However, $RRAM2$ cannot be SET because it is connected in the opposite direction. Therefore, when $RRAM1$ enters the LRS abruptly, the voltage across $RRAM2$ rises suddenly. This property is explored to design the proposed $I\&F$ neuron. As shown in Fig.2.31 (b), the proposed combination results no loop, similar to [111] as exhibited in Fig. 2.32.

TH-3339_186102005

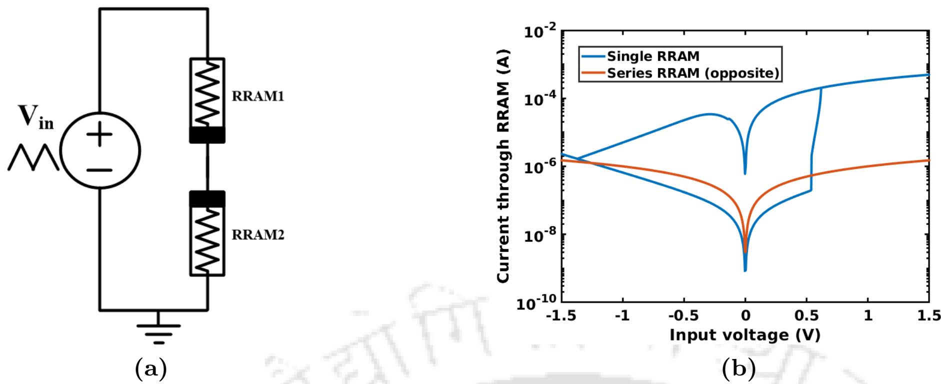


Figure 2.31: (a) RRAM connected in series combination in opposite direction (b) RRAM series connection obtained $I-V$ characteristics

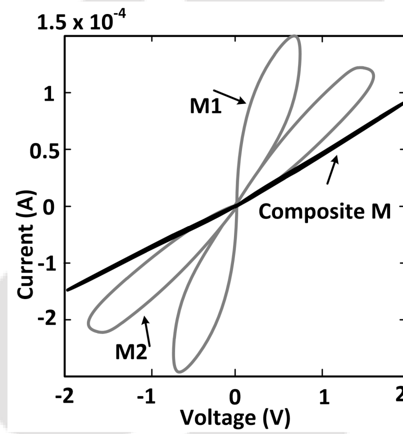


Figure 2.32: *RRAM series connection expected $I-V$ characteristics [111]*

2.6 Cycle-to-Cycle and Device-to-Device variation

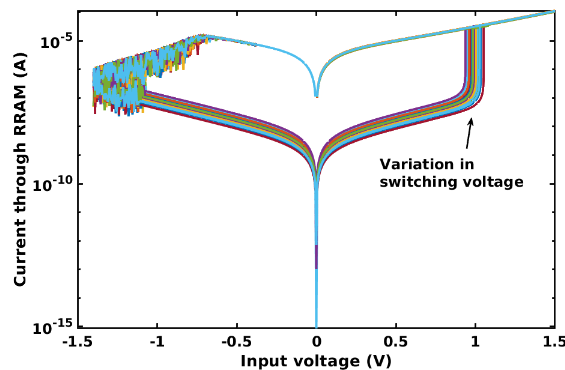


Figure 2.33: Switching voltage variability for 20 consecutive SET/RESET cycle

TH-3339_186102005

RRAM devices suffer from cycle-to-cycle and device-to-device variations, a major hindrance to developing RRAM-based large-scale systems. The Unimore Verilog-A model incorporates the cycle-to-cycle and device-to-device variations in RRAM. Fig. 2.33 shows variation in the switching voltage for 20 consecutive SET/RESET cycles. The variation in the resistances of the RRAM in LRS and HRS states is also evaluated. Fig. 2.34 shows the mean and standard deviation of the LRS and HRS for cycle-to-cycle variations. The mean and standard deviations during the LRS are 694 Ω and 122 Ω, whereas the mean and standard deviations during the HRS are 75 $K\Omega$ and 19.8 $K\Omega$. Fig. 2.35 illustrates the mean and standard deviations of the LRS and HRS for device-to-device variations. The mean and standard deviations during the LRS are 693 Ω and 124 Ω, whereas the mean and standard deviation during the HRS are 73 $K\Omega$ and 18.19 $K\Omega$.

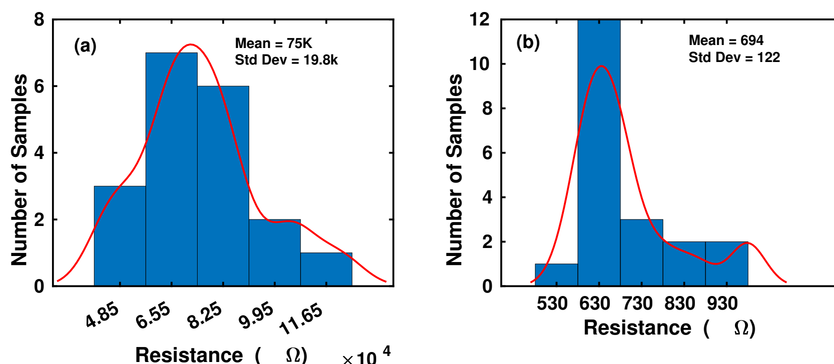


Figure 2.34: Mean and standard deviation for cycle-to-cycle variation in (a) HRS and (b) LRS

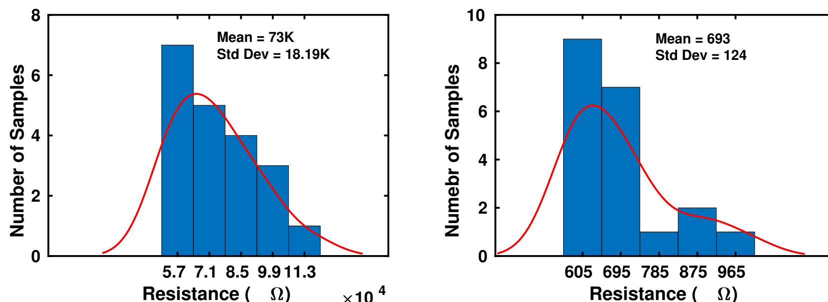


Figure 2.35: Mean and standard deviation for device-to-device variation in (a) HRS and (b) LRS

2.7 Summary

This chapter presents the detailed analysis of the Peking Verilog-A model and the Unimore Verilog-A model. We study the device structure and the model equations that define the resistive state of

TH-3339_186102005

the device. We further study the effects of different model parameters on the $I - V$ characteristics of RRAM. It is found that some parameters can be tuned to achieve faster switching, whereas the on/off ratio of the device can also be controlled by tuning the parameters. All the parameters are varied in the specified range in the Verilog-A model, ensuring that the characteristics match the fabricated devices.

We perform various experiments by connecting the devices in series and parallel to explore different behaviour and verify the functionality of the device by matching it with the existing state-of-the-art works. As the Unimore Verilog-A model supports cycle-to-cycle and device-to-device variations, we analyzed the device's performance in the presence of these variations. The Unimore Verilog-A model includes a module that can induce random telegraph noise (RTN) during circuit simulations to study the effects of RTN noise on the peripheral circuits. The various analysis discussed in this chapter helps design appropriate hybrid CMOS/RRAM circuits, as discussed in the next chapters.

TH-3339_186102005

3

RRAM Based Integrate and Fire Neuron

Indian Institute of Technology Guwahati

Contents

3.1 Introduction 62
3.2 *I&F* operation using RRAM 62
3.3 RRAM connected in series and opposite direction 66
3.4 Digital reset control with Pulse propagation 66
3.5 Proposed I&F neuron with reset circuit 67
3.6 Behavioural analysis of the proposed neuron 73
3.7 Variation and Stability analysis 74
3.8 Benchmarking with state of the art *I&F* neuron circuits 76
3.9 Summary 77

TH-3339_186102005

3.1 Introduction

Neurons are the basic building block of a neuromorphic system. Therefore, optimizing the circuits mimicking neurons lead to low-power large-scale neuromorphic systems. The state-of-the-art analog implementations of $I\&F$ neuron circuits employ capacitors and complex CMOS circuits for the integration dynamics of an $I\&F$ neuron [112]. However, capacitors consume a lot of silicon area, hindering the realization of large-neuromorphic systems.

With the advent of novel devices, such as RRAMs, multifold reduction in power and area consumption can be achieved. RRAM is considered the most mature technology [104] and has been widely explored to implement low-power synapses for SNN. The abrupt SET characteristics of an RRAM can be explored to replace the capacitors to obtain the integration dynamics for an $I\&F$ neuron.

This chapter presents an RRAM-based $I\&F$ neuron with a built-in reset circuit. The proposed circuit uses RRAM as a voltage divider for $I\&F$ operation. Since the RRAMs are connected in series in the opposite directions, one RRAM remains in HRS. Therefore, it conducts a very low current during spiking. The proposed neuron exhibits temporal integration, triggering threshold, and refractory period similar to a biological neuron making it a suitable candidate to be used in neuromorphic computing. Including a reset circuit into an RRAM-based neuron enables the implementation of a large-scale SNN, making it superior in terms of power and energy consumption.

3.2 $I\&F$ operation using RRAM

The $I-V$ characteristics of RRAM in Fig. 2.16 shows that the RESET operation is gradual, whereas the SET operation is abrupt. This abrupt operation results from the sudden conductive filament formation between the top and bottom electrodes. We explore this property of the RRAM device to generate an important functionality that integrates the incoming current and generates a spike when the threshold is reached. When a positive potential is applied across the RRAM, it transits from HRS to LRS. Initially, there is no rise in the RRAM current, but a slow increase in current can be observed as the conducting filament starts forming. A sudden rise in the current is observed when the RRAM is SET, as shown in Fig. 3.1. This behavior is similar to the $I\&F$ operation of a biological neuron.

The inputs to the neuron in spiking neural networks are in the form of voltage pulses. Therefore, verifying the above-observed characteristics when input pulses are applied across the device is impor-

TH-3339_186102005

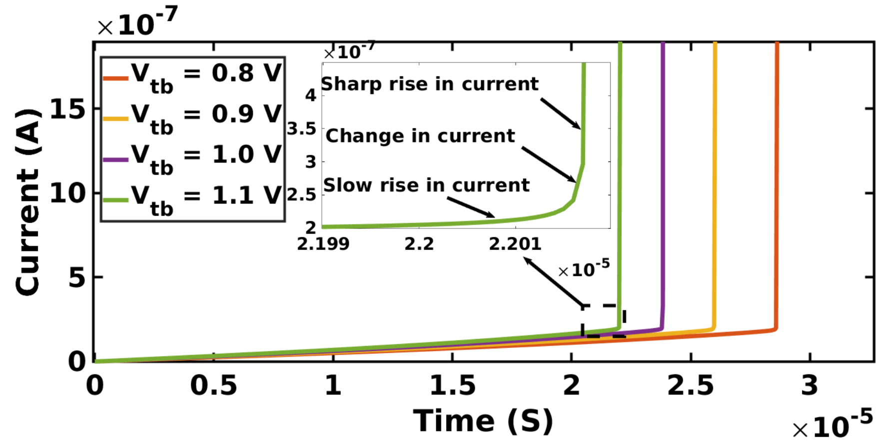


Figure 3.1: RRAM SET voltages at different V_{tb}

tant. We apply an input pulse of 1 V and 5 ns width and 10 ns period with 0.1 ns rise and fall time across the RRAM to obtain the pulsed characteristics and observe the current. The obtained characteristics are shown in Fig.3.2. The current dynamics in response to a pulse are the same as that of the quasi-DC characteristics in Fig. 3.1.

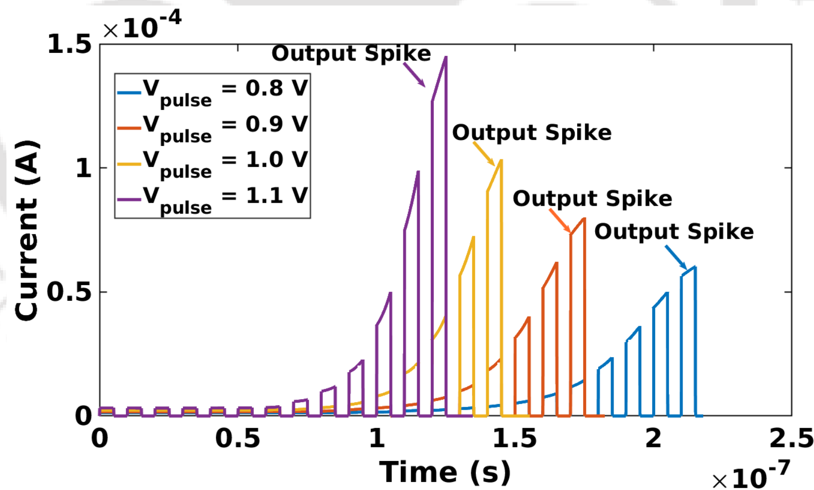


Figure 3.2: Current through RRAM for pulse input with different amplitude

The positive voltage applied at the top electrode affects the SET voltage of RRAM. As V_{tb} (voltage applied across the top and bottom electrode of RRAM) increases, the rate at which CF grows increases, making the RRAM SET process faster. Therefore it can be observed in Fig.3.2 that when the amplitude of the input pulse is increased, the output spike is obtained much earlier.

The Unimore Verilog-A model also incorporates the thermal effects. In Fig.3.2, we can observe that the current during the start of the next pulse is higher than that of the end of the previous

TH-3339_186102005

pulse. This is because of the change in temperature and x when a pulse is applied. In the absence of the pulse, the RRAM current increases due to a decrease in the temperature, reducing the overall resistance of RRAM. The Unimore RRAM Verilog-A model includes the thermal effects of RRAM switching. During the SET pulse, heating the conducting filament (CF) from $P1$ to $P2$ increases its resistance, whereas gap x is reduced, decreasing the resistance of RRAM. The increase in resistance due to the temperature is less than the decrease in the resistance due to the reduction in x. Therefore, there is an overall increase in the conductance. Conversely, when the SET pulse is absent, the change in gap temperature T_x reduces gap resistance R_{bar} because there is a minimal decrease in x. As shown in Fig. 3.3, T_{cf}, the conductivity filament temperature at $P2$ is higher than that at $P3$. The decrease in T_{cf} also lowers CF resistance R_{cf}, further reducing overall resistance R and increasing conductivity and overall current in the RRAM. Therefore, the current at $P3$ in RRAM is higher than at $P2$.

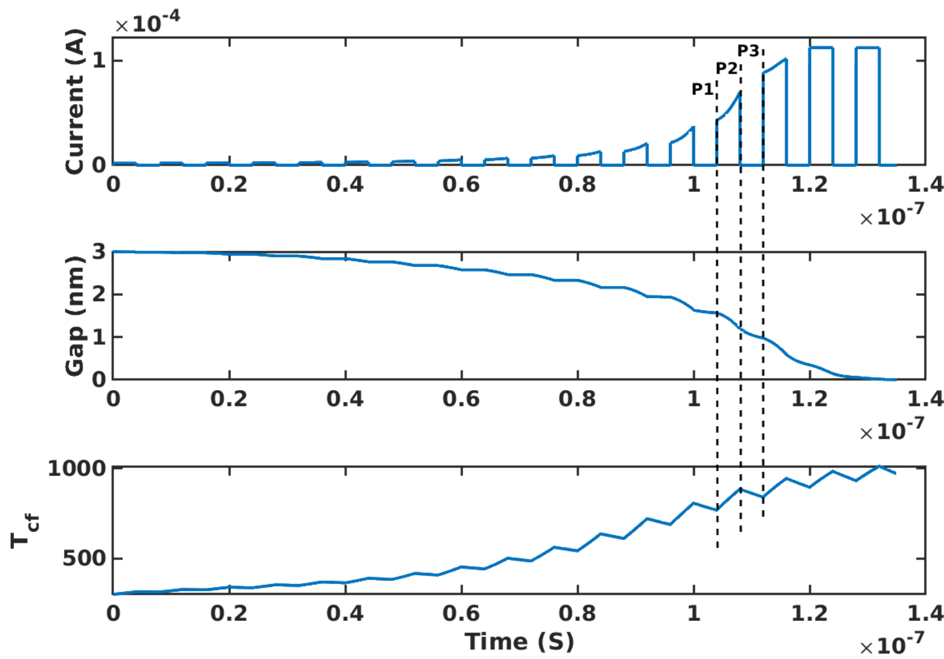


Figure 3.3: Temperature effects on RRAM switching

The device characteristics depicted in Fig. 3.3 can be explained using Eq.3.1 - Eq.3.4, which define the RRAM resistance of the Unimore Verilog-A model. As shown in Eq. 3.1, the current depends on R. R is the linear combination of R_{bar} and R_{cf}. It can be observed that, from $P1$ to $P2$, the increase in T_{cf} and decrease in x boost R_{cf}, whereas R_{bar} reduces. It can be seen that the decrement in R_{bar} is more than the increment in R_{cf}, which leads to an overall decrease in R, increasing the

TH-3339_186102005

overall current in RRAM. Similarly, when the SET pulse is absent, x is approximately constant as T_x decreases, which reduces R_{bar}. A further change in T_{cf} reduces R_{cf}. Thus, R is lowered, increasing the current in RRAM when the pulse is absent. This is showcased in Table 3.2. Table 3.1 exhibits all the parameters used for calculating the variables in Table 3.2.

$$I_{tb} = \frac{V_0 Sinh(\frac{V_{tb}}{V_0})}{R} \tag{3.1}$$

$$R = R_{bar} + R_{cf} \tag{3.2}$$

$$R_{bar} = (beta \times R_{lrs} e^{(\frac{x}{l}-1)} e^{(\frac{E_a}{K_b(T_x+C_1)})}) \tag{3.3}$$

Here, $C_1 = 1.0 \times 10^{-15}$

$$R_{cf} = R_{lrs}(\frac{t_{ox} - x}{t_{ox}})(1 + alpha(T_{cf} - T_{meas}) \tag{3.4}$$

Table 3.1: Parameters used in Verilog-A model

Parameter	Definition	Value
V_0	HRS curent non linearity factor	0.32
beta	barrier resistance fitting parameter	10^{-3}
l	typical tunneling length	0.42
$alpha$	Resistivity temperature coeff	0.002
t_{ox}	Oxide layer thickness	$12 \times 10^{-9} M$
E_a	Activation energy	$0.12 eV$
K_b	Boltzmann constant	8.6×10^{-5}
t_{meas}	Temperature at which LRS is measured	300

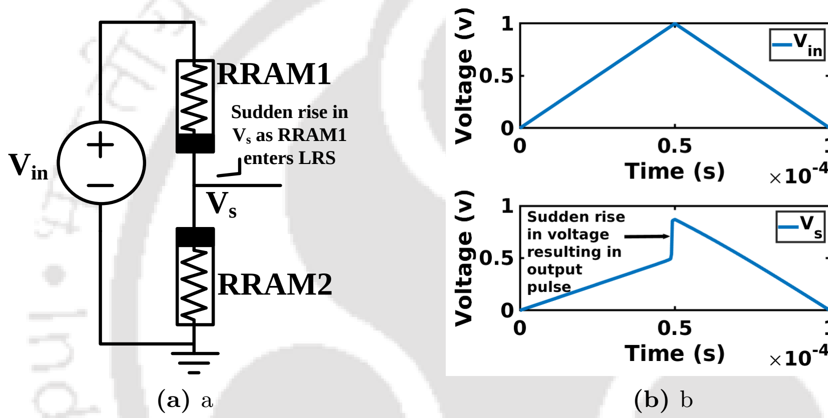
Table 3.2: Calculations for $P1$, $P2$ and $P3$

Point	V_0 (V)	V_{tb} (V)	T_{cf} (K)	R_{bar} (Ω)	R_{cf} (Ω)	R(Ω)	$I_{tb}(A)$
$P1$	0.32	1.4	776.3	2.05×10^3	850	2900	4.3×10^{-3}
$P2$	0.32	1.4	875.4	948	964	1912	6.6×10^{-3}
$P3$	0.32	1.4	846	467	962	1429	8.8×10^{-3}

TH-3339_186102005

3.3 RRAM connected in series and opposite direction

Fig. 3.4 (a) shows two RRAMs connected in series but in the opposite direction. A quasi-DC voltage sweep is applied to demonstrate $I\&F$ operation using the above-mentioned circuit. Since a positive voltage appears across $RRAM1$, it changes its state from HRS to LRS. At the same time, the voltage across $RRAM2$ is negative. Therefore it remains in HRS. The drop in $RRAM1$ resistance due to applied input voltage causes a sudden rise in the potential of V_s , which is shown in Fig. 3.4 (b).



(a) a **(b)** b

Figure 3.4: (a) RRAM connected in series and opposite direction to generate the required sudden change in voltage (b) Voltage V_s across $RRAM2$, when $RRAM1$ and $RRAM2$ are connected in opposite direction in series

Using a transistor as a resistor instead of RRAM2 is an alternative, but it needs to be operated in the linear region. This requires proper biasing and the gate terminal of the transistor to be connected with an external supply voltage to keep it in the linear region, which increases the design complexity of the neuron. Further, because of the narrow linear region, the DC operating point of the transistor would not be as stable as RRAM, in which the resistance does not change once it is set. Also, RRAM2 acts as a compliance element and limits the current to 0.4 μA, when RRAM1 sets abruptly. Thus, it is better to employ RRAM instead of a transistor for optimally designing the neuron.

3.4 Digital reset control with Pulse propagation

Fig. 3.5 exhibits an $I\&F$ neuron with digital pulse propagation and reset controller. It consists of a *Pulse Detector and Propagation*, *Timer*, and *Reset Enable* blocks. The pulse detector activates the *Reset Enable* block in response to the spike generated at terminal V_s, and, at the same time, the

TH-3339_186102005

Timer block also gets enabled. The *Timer* is switched on for 0.3 μs so that RRAM can be reset properly. Once the *Timer* is off, it deactivates *Reset Enable* so the neuron can accept the next input spikes. The above-mentioned blocks are implemented using 16 D-flip flops and one multiplexer. Total 142 transistors are employed in this implementation, consuming 210 μW power and 112×5 μm^2 area.

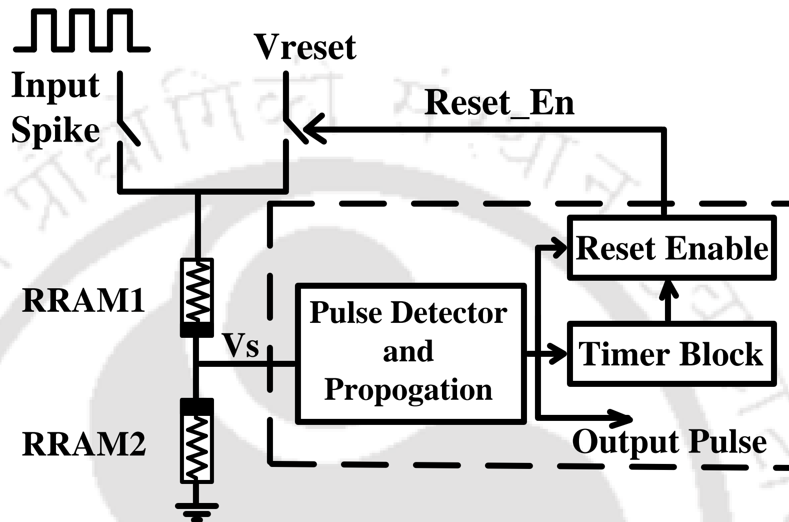


Figure 3.5: *I&F* Neuron with digital pulse propagation and RRAM reset block

Implementing a neuromorphic system involves a large number of neurons. For example, Intel *Loihi 2* [113] has one million neurons and 120 million synapses per chip. If the proposed digital neuron implemented at UMC 65 nm CMOS technology is employed to realize it on silicon, it will take 210 W power per chip as compared to less than 1 W power consumed by Intel Loihi realized at 14 nm FinFET technology. Since the proposed digital neuron is realized employing 16 D-flip flops, its power consumption is high due to its switching activity per spike. Thus, our endeavour is to reduce power at UMC 65 nm CMOS technology node itself by implementing it using novel devices, which is described below.

3.5 Proposed I&F neuron with reset circuit

The construction of the proposed *I&F* neuron circuit is illustrated in Fig. 3.6. This circuit can be divided into three parts. Part (i) consists of $RRAM1$ and $RRAM2$ in series, which perform the *I&F* operation. Part (ii) consists of transistors $M2 - M9$ propagating the spike generated to other neurons similar to the axon in a biological neuron. Part (iii) is the RRAM reset block consisting of transistors $M10 - M22$, which initializes $RRAM1$ to HRS. Using a transistor as a resistor instead of

TH-3339_186102005

$RRAM2$ seems to be an alternative, but it needs to be operated in the linear region. This requires proper biasing and the gate terminal of the transistor to be connected with an external supply voltage to keep it in the linear region, which increases the design complexity of the neuron. Further, because of the narrow linear region, the DC operating point of the transistor would not be as stable as RRAM, in which the resistance does not change once it is SET. Also, $RRAM2$ acts as a compliance element and limits the current to 0.4 μA when $RRAM1$ SETs abruptly. Thus, it is better to employ RRAM in place of a transistor for designing the neuron optimally.

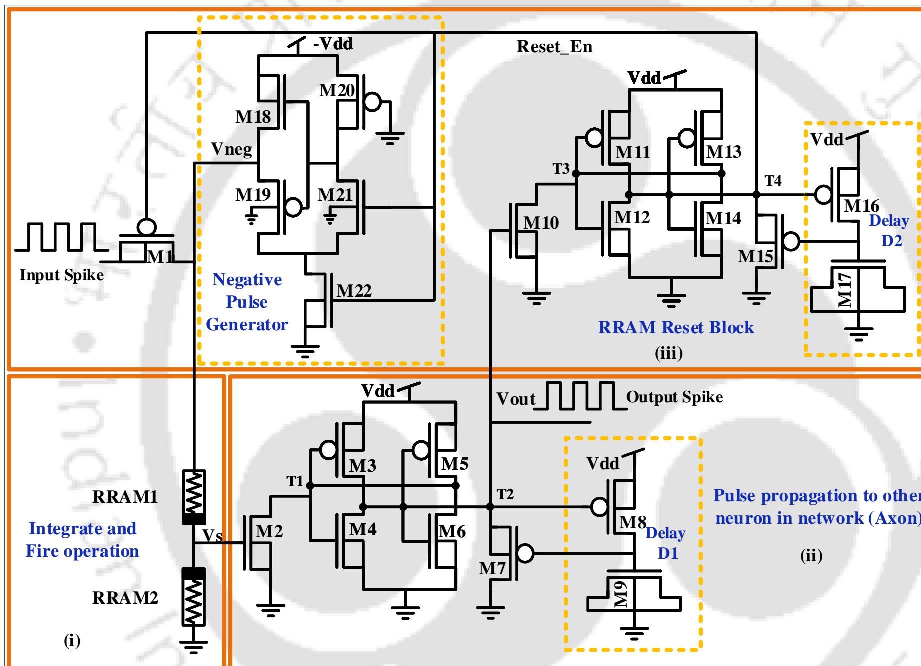


Figure 3.6: Proposed neuron circuit with integrated RRAM reset circuit

Fig. 3.6-(i) performs $I\&F$ operation. Initially, $RRAM1$ and $RRAM2$ are in HRS. For $RRAM1$, x_0 is initialized to 4 nm with a resistance of 50 $M\Omega$ in HRS. The potential across $RRAM2$ can be calculated using Eq. 3.5, where R_{RRAM1} and R_{RRAM2} are the resistances of $RRAM1$ and $RRAM2$, respectively. The initial value of V_s is found to be 0.02 V, and the initial current passing through $RRAM1$ in the RESET state is 15.5 nA. When input pulse of 0.8 V amplitude, 3 ns width, 6 ns period, and 0.1 ns rise and fall time is applied at the input, a pulse is generated at V_s when $RRAM1$ is set, and the current through $RRAM1$ increases to 0.4 μA. It can be observed that during this spiking event, the current passes through the RRAM circuit is very low. This significantly reduces the energy consumption of the proposed neuron, making it a suitable candidate to be employed in neuromorphic computing. In order to generate pulses continuously, it is imperative to reset $RRAM1$.

TH-3339_186102005

$$V_s = \frac{R_{RRAM2}}{R_{RRAM1} + R_{RRAM2}} V_{in} \tag{3.5}$$

Table 3.3: *I&F* neuron circuit parameters

Transistor	Channel length (nm)	Channel width (nm)
$M1$, $M13$	80	400
$M2$, $M10$	85	400
$M3$, $M5$, $M13$, $M11$	80	200
$M4$, $M6$, $M14$, $M12$	80	100
$M7$, $M8$, $M16$, $M15$	80	550
$M20$	70	120
$M21$, $M22$	210	120
$M18$, $M19$	120	120

3.5.1 Analog pulse propagation and RRAM reset block

The implementation of a neuron using RRAM and bulk CMOS transistors is illustrated in Fig. 3.6. Initially, terminals $T1$ and $T2$ are at V_{dd} and zero potential, respectively. It is to mention that $M2$ is OFF because V_s is at a very low potential. As soon as spikes arrive at the input terminal, the resistance of $RRAM1$ starts reducing. When $RRAM1$ is SET, the voltage at terminal V_s is found to be $\sim 0.52\ V$, sufficient to switch $M2$ on. As a result, $T1$ pulls down to $0\ V$, and the potential at $T2$ increases to V_{dd}, which further turns $M8$ off and enables the delay unit $D1$. To generate an output spike, the latch composed of $M3$, $M4$, $M5$, and $M6$ needs to be reset. This is accomplished with the help of $D1$, which provides a delay of 3 ns leading to produce an output pulse similar to the input pulse. It is to mention that the delay mostly depends on the leakage current and the width of $M9$ [114], which is set to 200 nm. $D1$ enables $M7$ after 3 ns, which brings down the potentials of $T1$ and $T2$ to V_{dd} and 0 V, respectively.

As we know, during the refractory period, a neuron should not respond to the incoming pulses. This functionality is incorporated in the proposed design, and this period is utilized to reset $RRAM1$. Thus, for the neuron to accept incoming pulses, $RRAM1$ must be reset to HRS. The RRAM reset block shown in Fig. 3.6-(iii) resets $RRAM1$ whenever a pulse is generated at $T2$. The schematic and functioning of the reset block is the same as that of the pulse propagation block except for the output pulse duration, which should be equal to the RRAM reset time and is estimated to be approximately 0.3 μs. Therefore, the delay unit $D2$ is designed accordingly, and the width of $M17$ is set to 400 nm

TH-3339_186102005

for the said purpose. Since a neuron must enter into the refractory period after spike generation, thus when the potential at $T4$ rises to V_{dd}, it switches $M1$ off and isolates the neuron from the input terminal. Further, in order to reset $RRAM1$, a negative voltage is required.

Fig. 3.6-(iii) exhibits the schematic of the negative pulse generator. To repeat the $I\&F$ cycle, the RRAM should be RESET after every SET. A negative voltage needs to be applied across the RRAM to RESET. Therefore, we need a negative pulse generator circuit to be incorporated into the neuron circuit. The schematic for the negative pulse generator for different modes of operation is depicted in Fig.3.7 and Fig.3.8. The working of the negative pulse generator is explained below with the help of an input transistor $M1$ in the proposed neuron circuit.

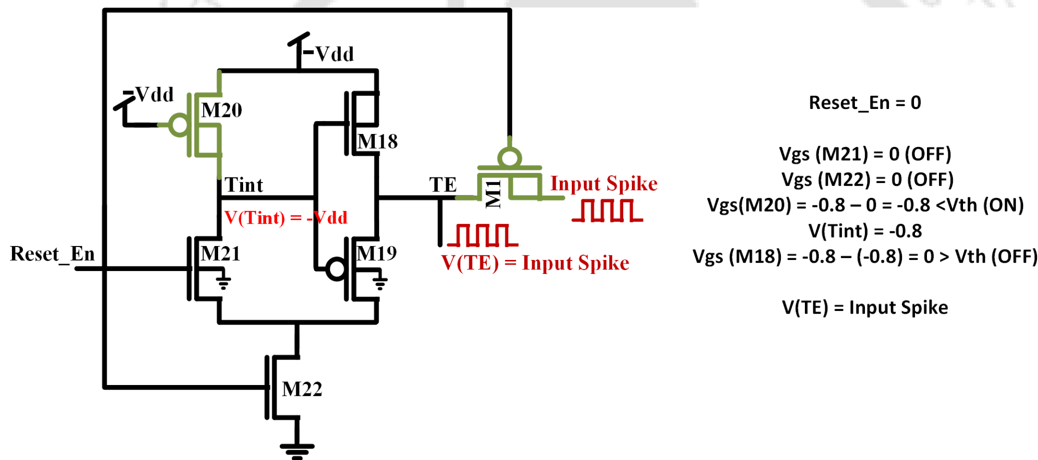


Figure 3.7: Schematic for negative pulse generator ($Reset_En = 0V$)

When $Reset_En$ is set to 0, the gate to source voltage (V_{gs}) of $M21$ and $M22$ becomes 0 V, which turns off $M21$ and $M22$. Similarly, V_{gs} for $M20$ equals $-0.8\ V$, which is less than the threshold voltage of the PMOS transistor $M20$. This sets $Tint$ at $-Vdd$ ($-0.8\ V$). Thus, it can be seen that $M18$ and $M19$ are off as their V_{gs} are less than the threshold voltages. Since the gate terminal of transistor $M1$ is at 0 V, and an input spike with a positive voltage (0.8 V) is applied to the source terminal, $M1$ acts as a switch, and the input spikes pass through TE, which is connected to the top electrode of $RRAM1$ in Fig. 3 of the revised manuscript. Fig. 3.9 illustrates the output waveform for a negative pulse generator. It may be observed that the potential at TE, i.e., V(TE), varies from ($0.8\ V - 0.1\ V$).

A transmission gate may be used in place of transistor $M1$ to get a full voltage swing at TE, but this would add no benefit to the functionality of the neuron because an RRAM sets only when a minimum set potential appears across the top electrode. Thus, using a single PMOS transistor $M1$,

TH-3339_186102005

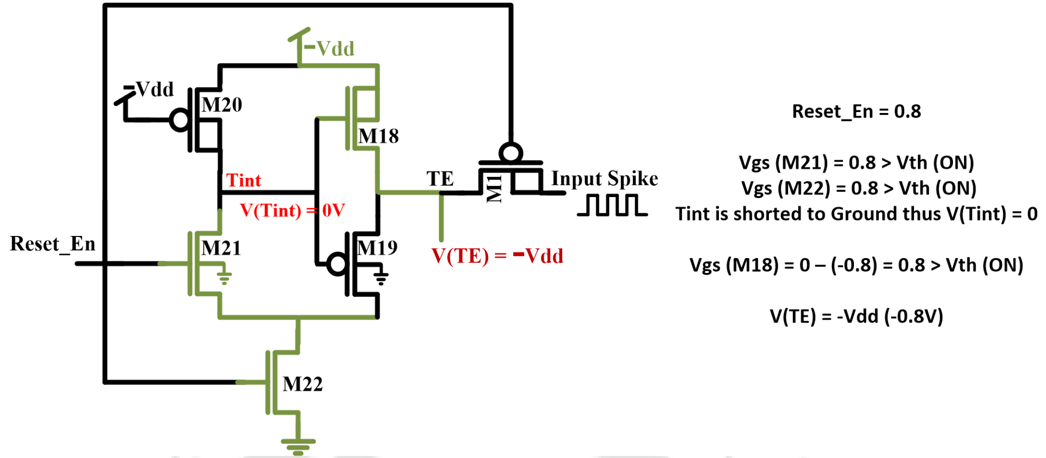


Figure 3.8: Schematic for negative pulse generator ($Reset_En = 0.8V$)

we save the area of three transistors (one NMOS transistor and two MOSFETs for an inverter required for a transmission gate), optimizing the circuit.

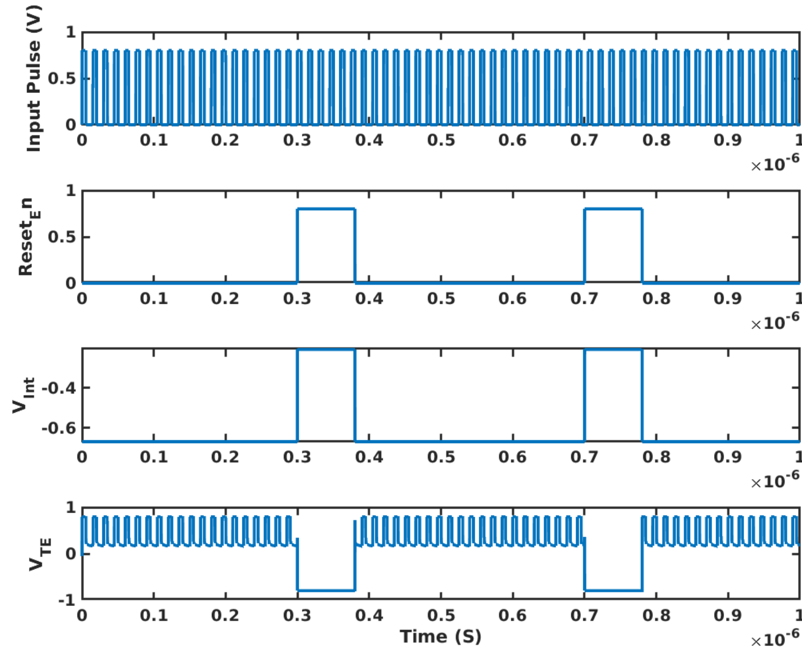


Figure 3.9: Output for negative pulse generator

When $Reset_En$ is 0.8 V, it may be seen in Fig. 3.8 that V_{gs} of $M21$ and $M22$ are 0.8 V, which is greater than the threshold voltage, turning these transistors on. When $M21$ and $M22$ are turned on, $Tint$ is short-circuited to the ground, making V($Tint$) equal to 0 V. Due to the rise in the potential of $Tint$, $M18$ switches on as $V_{gs} = 0.8\ V$. $M1$ is turned off as the potential of the gate terminal becomes

TH-3339_186102005

0.8 V. Thus, as depicted in Fig. 3.9, we get a negative potential at TE, which resets the $RRAM1$ connected at TE. It is to mention that $M18$ and $M19$ form an inverter with logic levels shifted, i.e., high $\rightarrow$ gnd and low$\rightarrow$ $-Vdd$. $M22$ is used to avoid connecting TE to the ground when $Reset_En$ is 0. If $M22$ is not employed, it may result in input pulses shorted to ground through $M19$.

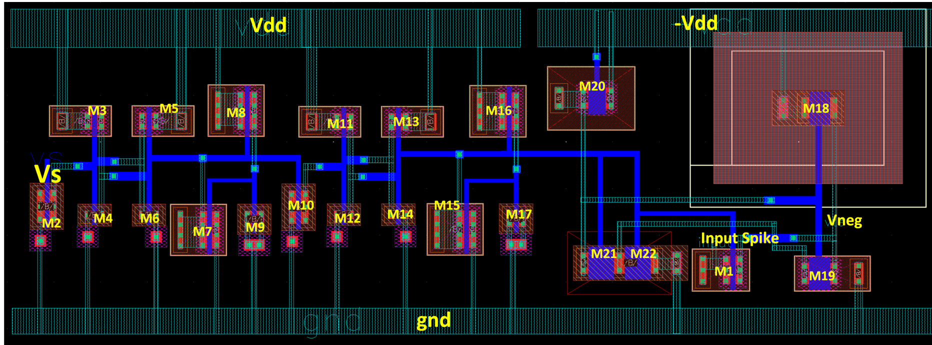


Figure 3.10: Layout of pulse propagation and RRAM reset block

Note that as the negative voltage is used in the circuit, there are chances that the transistors may break down. Thus, $M20$ and $M21$ with gate oxides with 6 nm thickness are employed in the proposed circuit, whereas other transistors with 2.6 nm gate oxide thickness are used. Once $RRAM1$ resets to HRS, the delay unit turns on $M15$, bringing down $T4$ to 0 V, which enables $M1$ and the neuron becomes ready to accept the input pulse again. The above-mentioned steps get repeated to make each $I\&F$ cycle identical. Fig. 3.6-(ii) and Fig. 3.6-(iii) are realized using 22 CMOS transistors consuming 2.3 μW power and 12×3 μm^2 area. Table 3.3 illustrates circuit parameters used in the implementation of the proposed neuron. Its layout is designed using Cadence Virtuoso and is shown in Fig. 3.10.

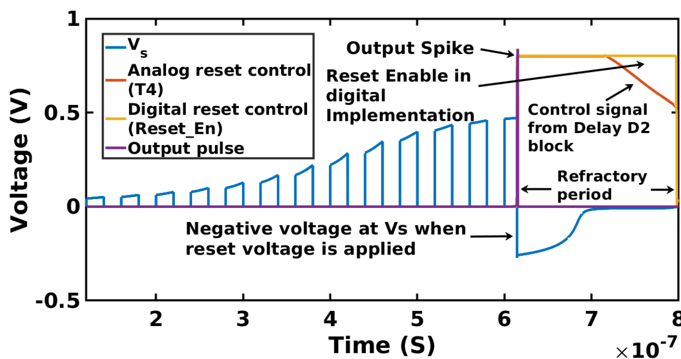


Figure 3.11: (a) Output for digital reset control and analog reset control (b) Firing frequency with respect to initial conducting filament length and amplitude of input pulse.

Fig. 3.11 exhibits the output pulse generated employing analog and digital implementations of pulse propagation and reset control block. The output spikes generated by both these implementations are the same. However, the area utilization and power consumption of analog implementation is 15.56× and 91.30× times less than that of the digital implementation.

3.6 Behavioural analysis of the proposed neuron

As we know that the firing frequency of a neuron should vary with the amplitude and width of incoming pulses. This is exhibited by the proposed neuron in Fig. 3.12, where the amplitude of the input pulse is varied from 0.8 V to 1.1 V. In Fig. 3.13, the width of the 0.8 V input pulse is varied from 3 ns to 15 ns. The initial value of x, i.e., x_0, also impacts the time required to set the RRAM and affects the output frequency of the neuron.

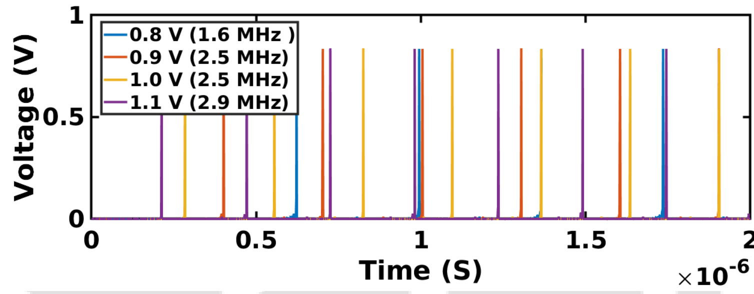


Figure 3.12: Firing frequency for different input pulse voltage

For the different values of initial conducting filament length, x_0, the range of firing frequency varies, as shown in Fig. 3.14. The neuron generates the lowest output frequency spikes for $V_{pulse} = 0.8\ V$, with a pulse width of 3 ns, and x_0 is initialized to 4 nm, whereas maximum spikes are generated for $V_{pulse} = 1.1\ V$, with a pulse width of 15 ns and $x_0 = 2.5\ nm$. This aids in setting an appropriate firing frequency of a neuron according to the application. It is to mention that Eq. 3.6 is employed to calculate the energy per spike of the neurons, which is depicted in Table 3.4, while comparing the proposed neuron with various contemporary neurons.

$$E = \sum_{i=1}^{N} V_{pulse,i} I_{pulse,i} T_{pulse,i} \tag{3.6}$$

TH-3339_186102005

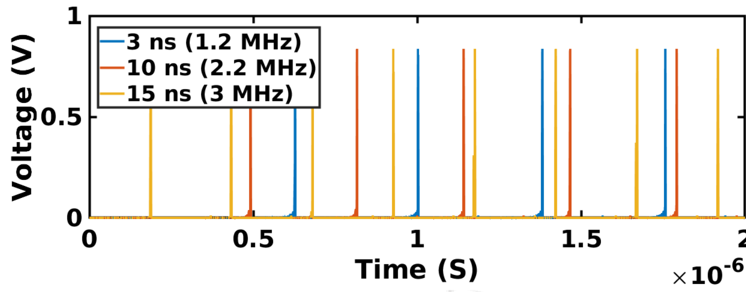


Figure 3.13: Firing frequency for different input pulse width

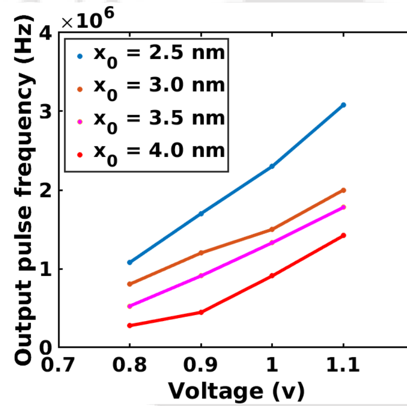


Figure 3.14: Firing frequency with respect to initial conducting filament length and amplitude of input pulse.

3.7 Variation and Stability analysis

In this section, the effect of variations of various parameters of RRAM on the performance of the proposed neuron is studied to evaluate its stability. Fig. 3.15 illustrates the cycle-to-cycle variation in the RRAM resistance, when RRAM is subjected to a different number of pulses.

As observed in Figs.3.16 and 3.17, the proposed neuron produces reliable output if the variation is limited to 20 %. As we know, the stability of the neurons is imperative for the reliable operation of a neural network; an analysis of the proposed neuron is conducted by varying temperature and supply voltage. It is found that there is 23 % variation in the output spiking frequency when the temperature changes from $-10^{\circ}\ C$ to $110^{\circ}\ C$. Further, variation in the supply voltage from $0.8\ V$ to $1.2\ V$ results in 25 % change in the spiking frequency.

Similarly, a corner case analysis of the proposed neuron is performed to evaluate its performance. Fig. 3.18 depicts output spikes generated by the proposed neuron in all the corner cases. For the proposed study, SS, FF, FNSP, and SNFP corners are used. Here, SS, FF, FNSP, and SNFP stand

TH-3339_186102005

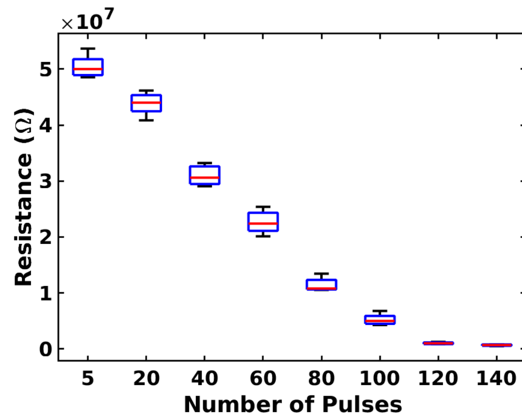


Figure 3.15: The distribution of resistance of RRAM for different number of pulses

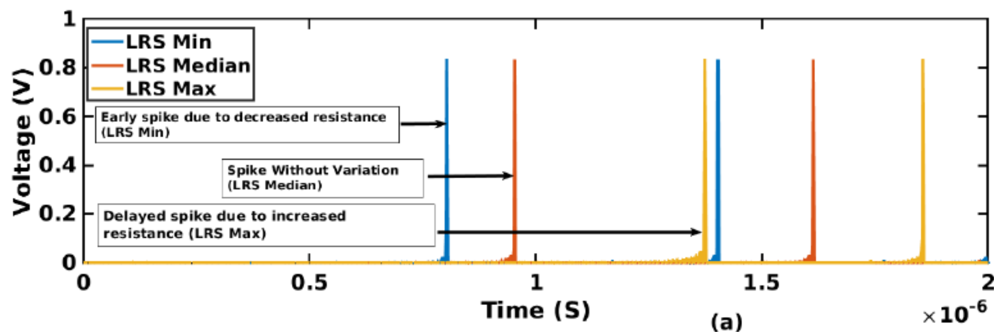


Figure 3.16: Neuron spiking in presence of 20% variation in resistance

for *Slow NMOS and Slow PMOS, Fast NMOS and Fast PMOS, Fast NMOS, and Slow PMOS,* and *Slow NMOS and Fast PMOS*, respectively. For SS and SNFP, the threshold of NMOS transistors is increased, thus, as expected, the spikes are delayed for SS and SNFP corners. On the contrary, for FF and FNSP, the threshold of the NMOS transistor is decreased, which causes the neuron to spike early. Although differences in the spike timing at different corners are observed, it can be noted that the proposed neuron generates spikes at all the corners appropriately.

The circuit of a neuron is validated in the presence of white Gaussian noise. The current profile of Gaussian white noise is depicted in Fig. 3.19 (a). It is to mention that the output spikes of a neuron are shown in Fig. 3.19 (b) when noise is absent in the input signal. It can be observed in Fig. 3.19 (c) output bursts exist in the presence of noise, and spike count increases slightly. It can be stated that the variation tolerant circuit design adds more elements to the circuit, increasing its design complexity, area utilization, and power consumption, while the incorporation of variation tolerant techniques at the architecture level produces optimal design exhibiting its better performance.

TH-3339_186102005

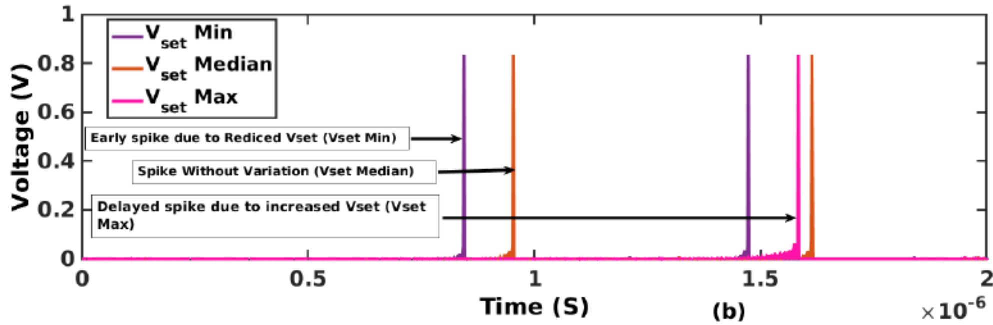


Figure 3.17: Neuron spiking in presence of 20% variation in switching voltage

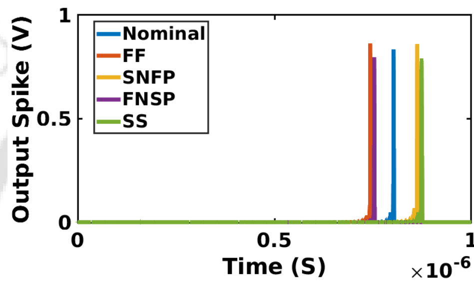


Figure 3.18: Neuron spiking in corner cases

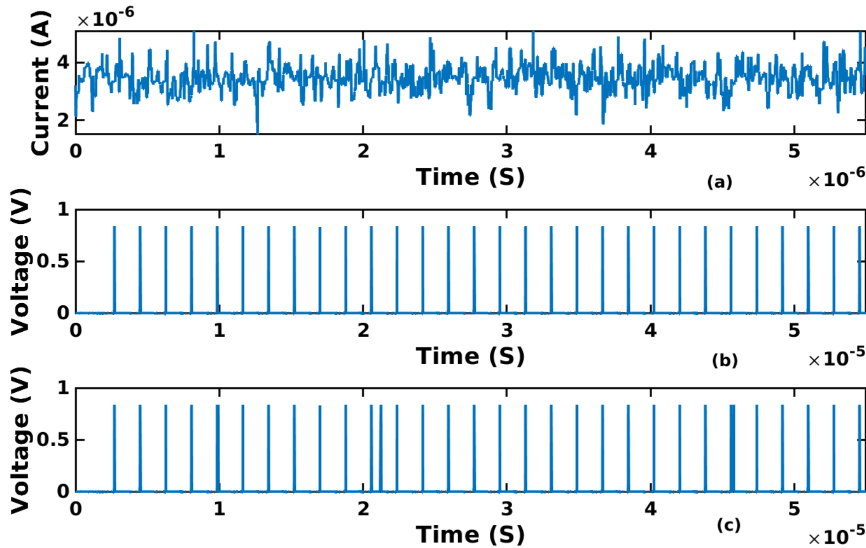


Figure 3.19: (a) Gaussian white noise current profile (b) Neuron spiking in presence of noise

3.8 Benchmarking with state of the art $I\&F$ neuron circuits

As shown in Table 3.4, [115] employs novel devices for the spike generation and exhibits a higher frequency range, but energy per spike is still in the range of $10^{-12}J$ due to device switching. [116], [118]

Table 3.4: Comparison of energy per spike, frequency and need for external reset circuit of proposed I&F Neuron and published I&F Neurons

Publication	Year	Platform	Neuron Model	Energy per spike (J)	Spiking frequency	External reset circuit/Controller
[115]	2018	PCMO	IF	4.8×10^{-12}	$0.4\ MHz - 0.8\ MHz$	Yes
[116]	2020	SGFBPF	LIF	0.25×10^{-12}	$150\ kHz$	No
[117]	2020	FBFET	IF	2.9×10^{-15}	$20\ kHz$	No
[118]	2020	PDSOI MOSFET	IF	3.2×10^{-15}	$150\ kHz$	Yes
[119]	2021	DG-JLFET	LIF	1.14×10^{-12}	$200\ MHz$	No
[120]	2021	CMOS	IF	0.135×10^{-12}	$0.2\ kHz$	No
Proposed Work	2022	RRAM	IF	1.5×10^{-15}	$277\ kHz - 3\ MHz$	No

consume less energy per spike, but they need a digital reset circuit, which increases overall power consumption. Although [119] and [120] do not use any external reset circuit, the energy per spike is in the range of $10^{-12}J$. This is because of their implementation using CMOS logic. The proposed neuron circuit has the lowest energy per spike, 1.5×10^{-15} J, and it does not need any external reset circuit. It is 2× more energy efficient than [117], which is implemented using nanoscale FBFET technology.

3.9 Summary

This chapter discusses an RRAM-based $I\&F$ neuron with a self-resetting circuit. We perform corner analysis to verify the performance of the proposed neuron at all the process corners. We also evaluate the neuron spiking in the presence of RRAM's cycle-to-cycle and device-to-device variations. The proposed neuron circuit generates reliable output spikes even in the presence of RTN, validating the robustness of the proposed circuit. The neurons realized with MOSFETs do not need external circuits but exhibit higher energy usage. Novel devices, such as PCM, and RRAM, generate spikes using less energy but require an external reset circuit. Integrating the reset block with the neuron enables area and power optimal implementation of large-scale SNN. The proposed neuron's area utilization and power consumption are 36 μm^2 and 2.3 μW. Using MOSFET as a capacitor makes the proposed neuron highly scalable. Since the synapse in SNN can also be implemented using RRAM, the proposed neuron can be easily integrated with RRAM synapse, enabling synapse and neuron to be designed using the same device technology for realizing large neuromorphic systems efficiently.

TH-3339_186102005

TH-3339_186102005

4

RRAM based 4-bit/cell Synapse

Indian Institute of Technology Guwahati

Contents

4.1 Introduction 80
4.2 $4T - 1R$ **structure for SET/RESET operation** 80
4.3 Synaptic architecture 82
4.4 CMOS peripheral circuits 85
4.5 Variation analysis 92
4.6 Power, Latency, Energy and Area estimation 96
4.7 Comparison with the contemporary architectures 98
4.8 Summary 99

TH-3339_186102005

4.1 Introduction

Resistive Random Access Memory (RRAM) has been widely explored to represent the synaptic weights in artificial neural networks. Despite significant advances in device technology, implementing multi-bit synapses that could save area utilization by increasing the density of the synaptic architectures is challenging. Moreover, precise modulation of conductance required for maintaining high accuracy remains a significant challenge. These devices suffer mainly from cycle-to-cycle and device-to-device variations, which make it even more difficult to implement reliable neuromorphic systems.

In this chapter, an RRAM-based $4 - bit/cell$ synaptic architecture with continuous sensing and feedback scheme is present to stop RRAM programming when the required conductance is achieved. Unlike contemporary architectures, which require a precise gap between different resistive states, the proposed feedback scheme to stop RESET operation provides flexibility in choosing resistive states. The variation and stability analyses of the peripheral circuits are performed to verify the robustness of the proposed programming scheme for the synaptic architecture.

4.2 $4T - 1R$ structure for SET/RESET operation

In the proposed synaptic architecture, we employ a $4T-1R$ structure, as shown in Fig. 4.1 (a). It consists of two PMOS and two NMOS transistors that can be switched accordingly to SET or RESET the RRAM. As shown in Fig. 4.1 (b), $M1$ and $M2$ are turned on during SET operation, resulting in a positive potential across RRAM, triggering the SET process. Similarly, RRAM is RESET by enabling $M3$ and $M4$, resulting in a negative voltage across RRAM that triggers the RESET process, as depicted in Fig. 4.1 (c).

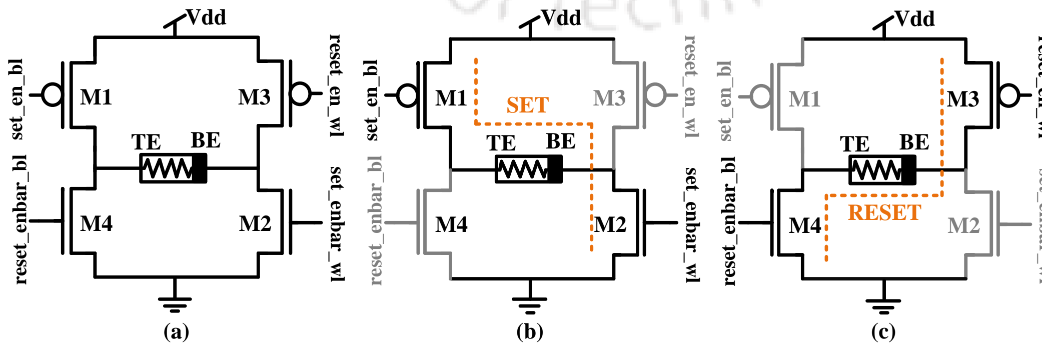


Figure 4.1: (a) Schematic for $4T - 1R$ structure (b) SET process (c) RESET process

The arrangement of the RRAM cells and the $4T - 1R$ structure in a synaptic array is shown in

TH-3339_186102005

Fig. 4.2. The NMOS and PMOS transistors are controlled using the *set_en_bl*, *set_enbar_wl*, and *reset_en_wl* and *reset_enbar_bl* indicating SET and RESET operation of a selected *wordline* and *bitline* in the synaptic array. When it is required to SET a particular cell, *set_enbar_wl* is enabled, selecting all the RRAM cells connected to output neuron $No0-Nom$. The input *set_en_bl* selects an individual RRAM cell. Similarly, if the RESET operation is to be performed, *reset_enbar_wl* is enabled, selecting all the cells connected to one particular output neuron and enabling *reset_enbar_bl* to select an individual cell to RESET.

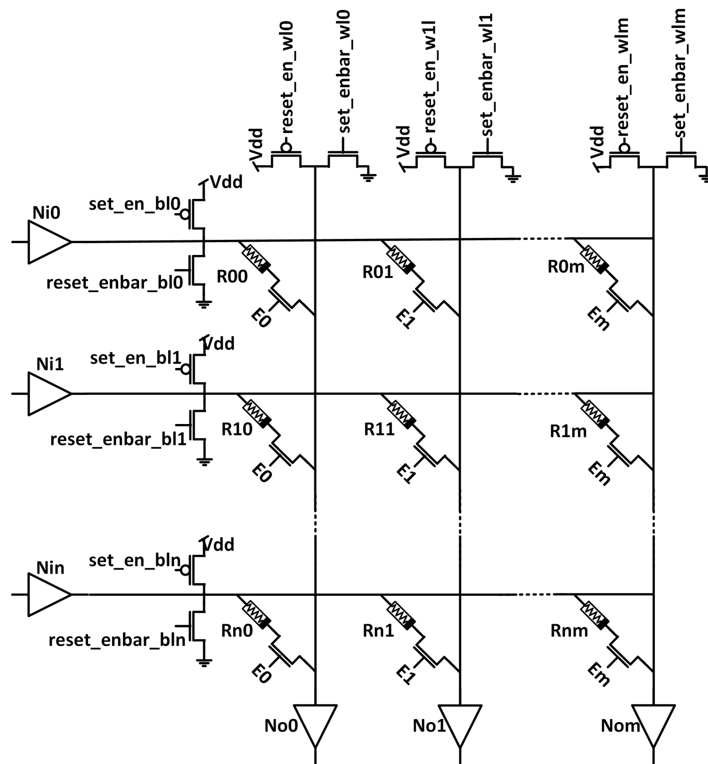


Figure 4.2: Arrangement of $4T-1R$ structure in array

The transistor in series with RRAM helps diminish sneak path current, a primary concern while designing resistive arrays. During training, appropriate signals can be applied depending on which cell is selected and whether it is to be SET or RESET. During the inference phase, all input neurons ($Ni0-Nin$) are connected to their corresponding output neurons through respective synapses $R00$-Rnm and signals $E0-Em$ control these connections. The programming of the RRAM cell is shown in Fig. 4.3. The RRAM cell is switched from SET to RESET repeatedly, showing that the $4T-1R$ structure can be efficiently used to reprogram the RRAM cells.

TH-3339_186102005

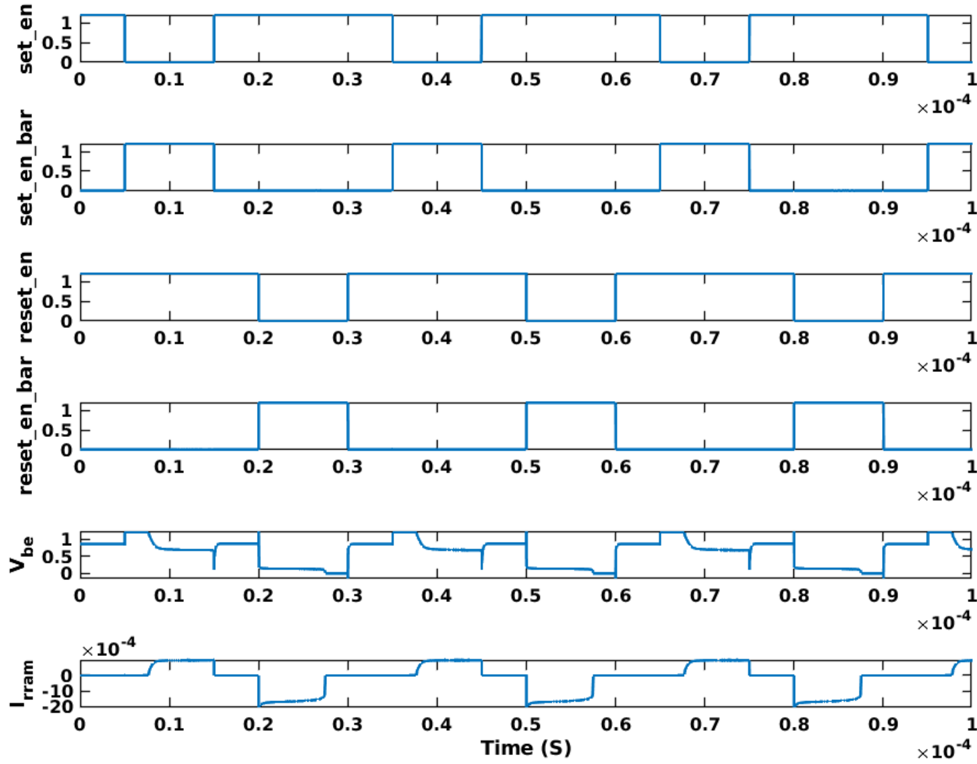


Figure 4.3: RRAM programming to SET and RESET

4.3 Synaptic architecture

The proposed synaptic architecture is depicted in Fig. 4.4. The *synapse update block* controls the SET and RESET operation while training. During inference, all the signals, namely *set_en_bl*, *set_enbar_wl*, *reset_en_wl*, and *reset_enbar_wl*, are disabled, thus, preventing any SET or RESET operation. To program the RRAM cell and precisely control its resistive state, we use a *RESET stop* block, as shown in Fig. 4.4. The purpose of the *RESET stop* block is to stop the RESET operation when a desired resistive state is achieved. During a RESET operation, a reference voltage corresponding to the required resistive state is generated by the reference voltage generator that consists of a digital-to-analog converter.

The voltage at V_{be} is compared to this reference voltage V_{ref}, and when $V_{be} = V_{ref}$, the comparator generates output signal *stop_c*. The voltage V_{be} may vary due to the switching of the transistors $M1$, $M2$, $M3$, and $M4$; this might prompt the comparator to generate false stop signals. Thus, a stop

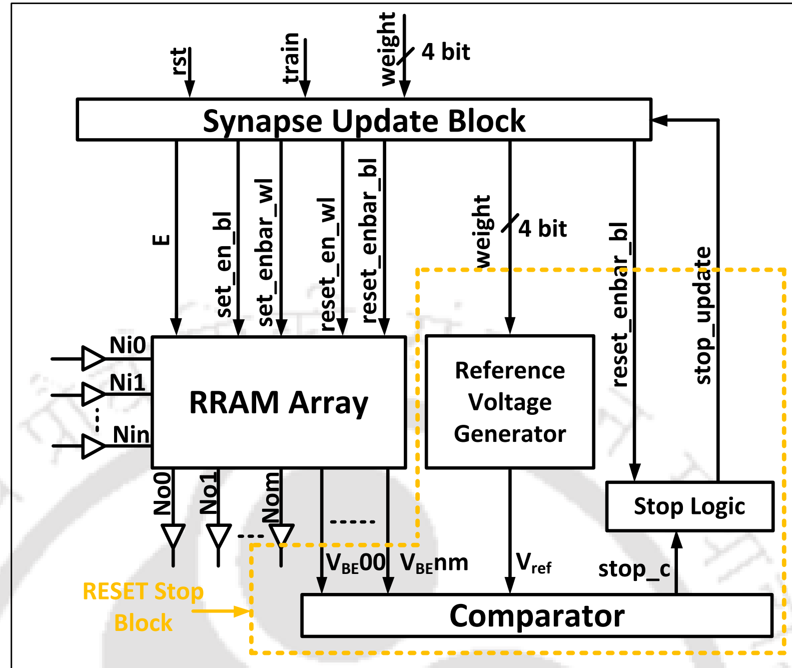


Figure 4.4: Synaptic architecture

logic circuit is designed that generates a stop pulse in response to the comparator output only when the *reset_enbar_bl* signal is enabled. This prevents false triggering in the circuit and assures reliable operation. Table 4.1 shows the allocation of the 16 resistive states ranging from 2.1 $k\Omega$ to 73 $k\Omega$. In Table 4.1, V_{be} indicates the voltage at the bottom electrode corresponding to each resistive state.

Table 4.1: Resistive state and its corresponding Vbe for 4-bit precision

Count	V_{be}	$R_{RRAM}(k\Omega)$	Count	V_{be}	$R_{RRAM}(k\Omega)$
1111	1.14	74.2	0111	0.91	32.8
1110	1.12	69.6	0110	0.88	28.2
1101	1.09	65.0	0101	0.854	23.6
1100	1.06	60.4	0100	0.82	19.0
1011	1.03	55.8	0011	0.79	14.4
1010	1	51.2	0010	0.76	9.8
1001	0.97	46.6	0001	0.73	5.2
1000	0.94	37.4	0000	0.7	1.1

The connection of an individual selected synaptic cell in the array is shown in Fig. 4.5. The functionality is explained with the help of waveforms in Fig. 4.6. As shown in Fig. 4.5, the BE of the RRAM cell is connected to the comparator to monitor the voltage drop during RESET operation. Reference voltage V_{ref} is obtained from the reference voltage generator. The output of the comparator

TH-3339_186102005

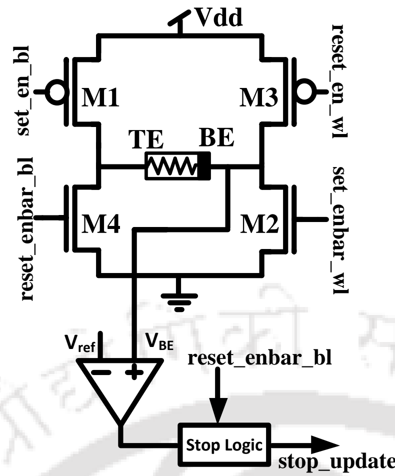


Figure 4.5: RESET stop block for a single RRAM cell

stop_c is fed to the stop logic circuit, which sends a feedback signal *stop_update* to the synapse update block to stop the RESET operation when the desired resistive state is achieved.

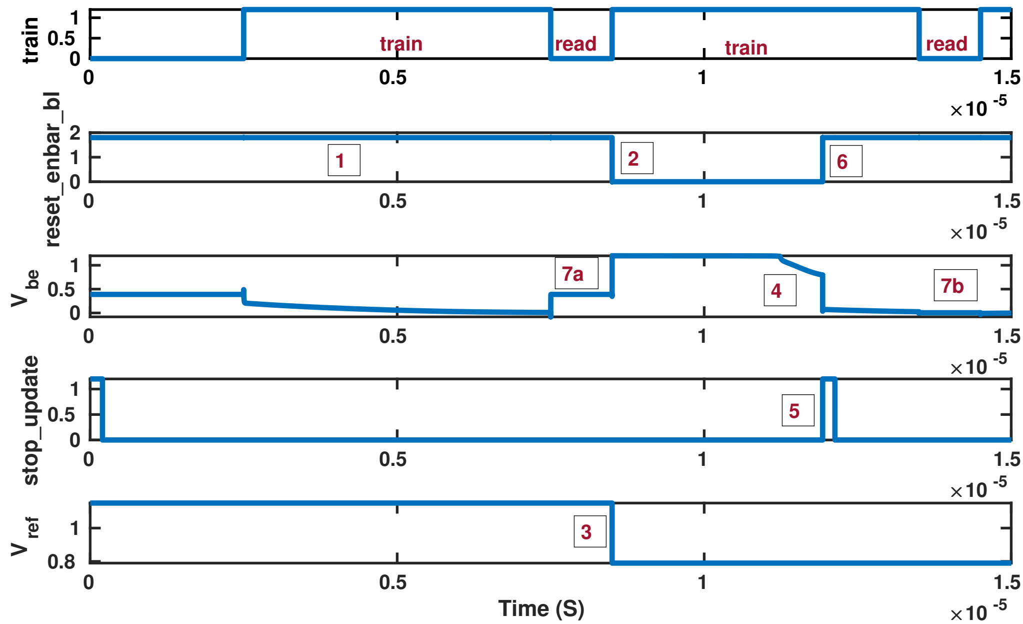


Figure 4.6: RESET operation for a selected RRAM cell

All the RRAM cells are initialized to LRS. Therefore, if the weight to be updated is 0000; no operation is performed since RRAM is already in LRS (indicated by $state-1$ in Fig. 4.6). During the next cycle, the weight to be updated is 0011; hence the *train* signal is enabled as indicated

TH-3339_186102005

by $state-2$. The synapse update block starts the RESET operation by enabling $reset_en_wl$ and $reset_enbar_bl$ signal corresponding to the selected RRAM cell. Also, the reference voltage generator provides a V_{ref} of 0.79 V, corresponding to the weight 0011, as indicated in the $state-3$. Since a negative potential appears across RRAM, the RESET process starts. As a result, the potential at BE, i.e., V_{be}, starts falling, as highlighted by $state-4$. Once the required resistive state is obtained, the potential at V_{be} reaches the corresponding value, and the comparator triggers the stop logic circuit, which further generates a positive pulse $stop_update$, as shown in the $state-5$. This, in turn, disables the $reset_enbar_bl$ signal, indicated by $state-6$.

The change in resistive state can be verified by observing the potential at V_{be} after the RESET process is completed. It can be observed that V_{be} before the RESET operation is higher than the V_{be} after the RESET operation is performed; this is highlighted by $states-7a$ and $7b$, respectively. Fig. 4.7 shows precise control of the RESET process. The $stop_update$ signal is generated precisely at every corresponding V_{be} of all resistive states shown in Table 4.1, validating our claim to achieve $4-bit/cell$ precision programming.

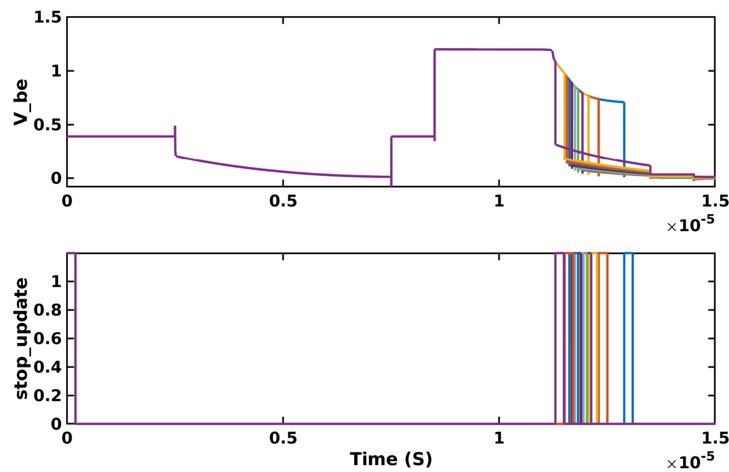


Figure 4.7: Precise $stop_update$ pulse generation for all 16 states

4.4 CMOS peripheral circuits

4.4.1 Reference voltage generator

A digital-to-analog converter (DAC), followed by a quasi-linear current-to-voltage converter shown in Fig. 4.8, is employed as a reference voltage generator. This provides a reference voltage for the comparator when a digital input is received from the weight update block of the proposed architecture.

TH-3339_186102005

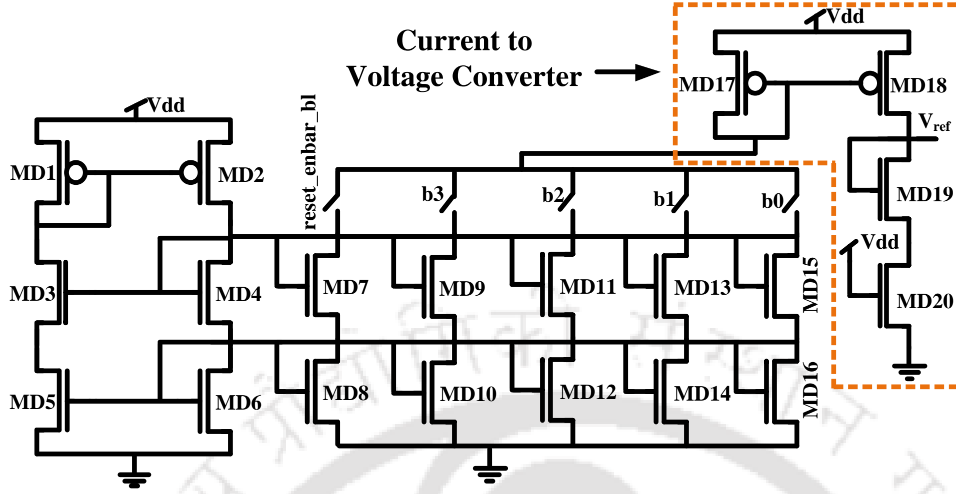


Figure 4.8: Reference voltage generator circuit

As depicted in Fig. 4.8, the DAC is followed by a quasi-linear current-to-voltage converter. Transistors $MD1 - MD16$ form the DAC, whereas transistors $MD17 - MD20$ form the current-to-voltage converter. The $MD17$ and $MD18$ transistors form a current mirror. The other two transistors, $MD19$ and $MD20$, are connected in series, creating a non-linear resistor. The output of the reference voltage generator is shown in Fig. 4.9. It shows that appropriate voltages are generated for all 16 possible input combinations.

The non-linear characteristics of the resistor help achieve a quasi-linear current-to-voltage conversion and a linear operation in a wide voltage range. For this, the transistor $MD19$ should operate in the triode region, and proper sizing of the transistor is also important. One of the conditions that must be fulfilled to achieve a quasi-linear characteristic is shown in Eq. 4.1. V_{DS} is the drain-source voltage, V_{GS} is the gate-source voltage, and V_{th} is the threshold voltage of the transistor $MD19$. We use a constant transconductance bias circuit to reduce the process variation and mismatch effects to supply constant current in the presence of variation and mismatch.

$$V_{DS} < V_{GS} - V_{th} \tag{4.1}$$

The maximum power is dissipated in the reference voltage generator when the inputs to the DAC change from 0000 to 1111 or from 1111 to 0000 because of the switching of all the current sources at once. We change the input to turn maximum transistors on and off to estimate the maximum power consumed. To achieve this, we initially apply 0000 and then change the input to 1111 so that

TH-3339_186102005

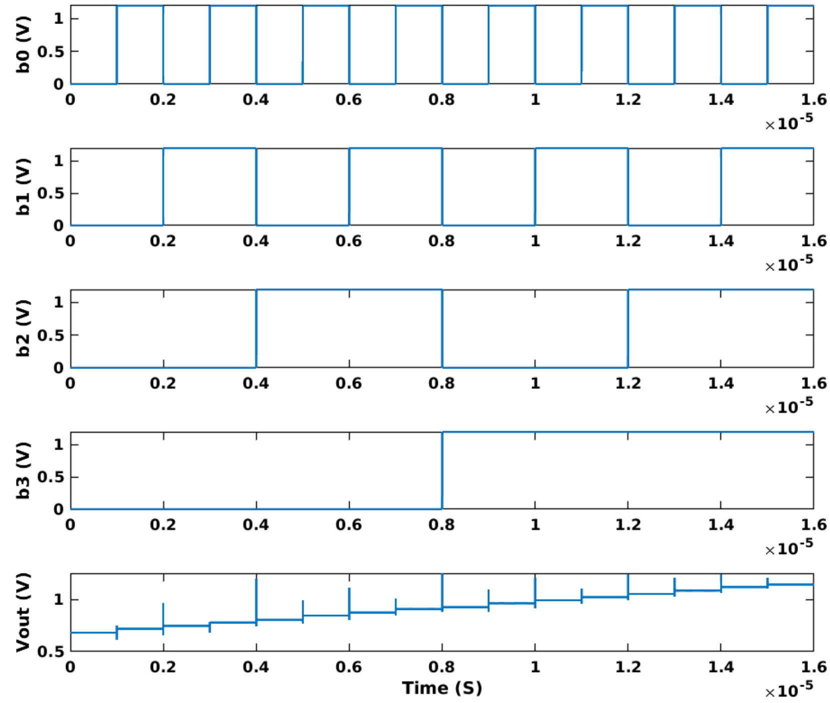


Figure 4.9: Output of reference voltage generator circuit

maximum voltage swing is obtained at the output of the reference voltage generator. Fig. 4.10 shows the output corresponding to the applied inputs and the instantaneous power consumed. Note that the power consumed by the reference voltage generator is estimated as 7.1 μW.

Table 4.2: Reference voltage generator circuit parameters

Transistor	Channel length	Channel width	Transistor	Channel length	Channel width
$MD1$	500 nm	85 nm	MD 12	1.1 μm	100 nm
$MD2$	800 nm	85 nm	$MD13$, $MD14$	1.35 μm	100 nm
$MD3$, $MD5$	600 nm	85 nm	$MD15$, $MD16$	1.5 μm	100 nm
$MD4$,$MD6$	950 nm	85 nm	$MD17$	500 nm	100 nm
$MD7$, $MD8$	850 nm	100 nm	$MD18$	1 μm	100 nm
$MD9$, $MD10$	1 μm	100 nm	$MD19$	80 nm	400 nm
$MD11$,	1.25 μm	100 nm	$MD20$	80 nm	400 nm

4.4.2 Comparator and stop logic circuit

The comparator triggers the stop logic circuit when the voltage at the bottom electrode V_{be} reaches the required voltage corresponding to the resistive state. When the required resistive state is obtained,

TH-3339_186102005

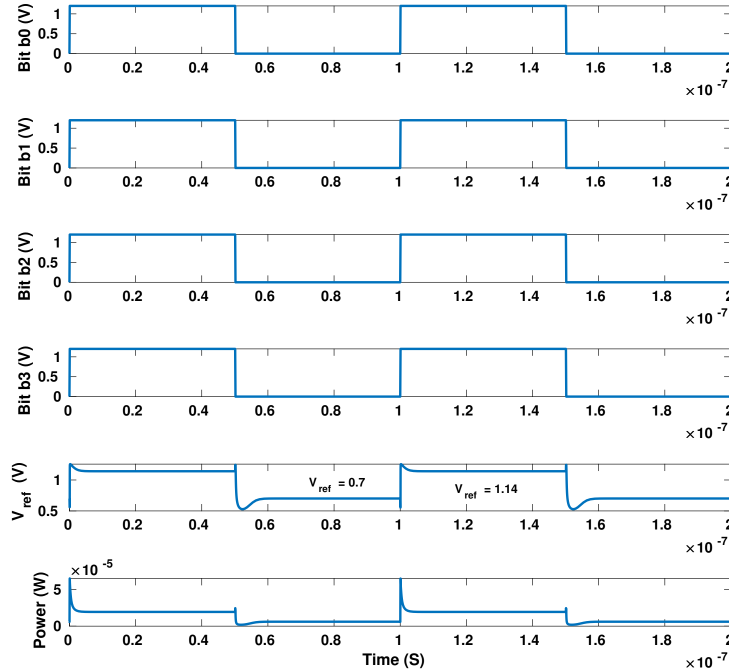


Figure 4.10: Power dissipation in reference voltage generator circuit

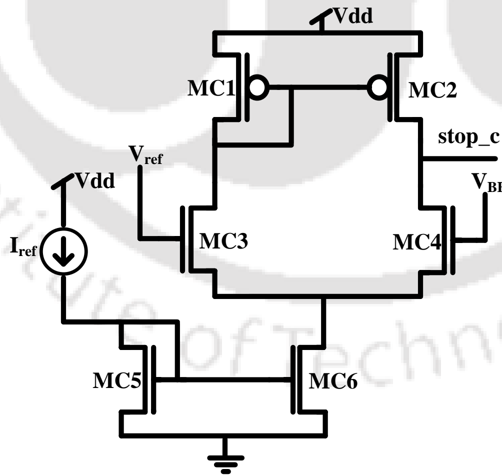


Figure 4.11: Schematic for comparator circuit

the comparator must be precise enough to trigger the stop logic circuit. Fig. 4.11 shows the schematic for the comparator circuit. We scaled the transistors appropriately such that the comparator works for the voltage range of 0.7 V to 1.14 V, similar to the voltage range of the resistive states. Transistors $MC5$ and $MC6$ form the current mirror circuit so that $MC6$ is a current source. Transistors $MC1$ –

TH-3339_186102005

$MC4$ form the differential pair. The parameters of the transistors used for the design are specified in Table. 4.3.

To estimate the maximum power consumed by the comparator circuit, we apply all the possible inputs to the comparator to measure the average power. As depicted in Fig. 4.12, we vary the input reference voltage from 0.7 V to 1.14 V as required for the synaptic array. The power consumed by the comparator circuit is 4.7 μW.

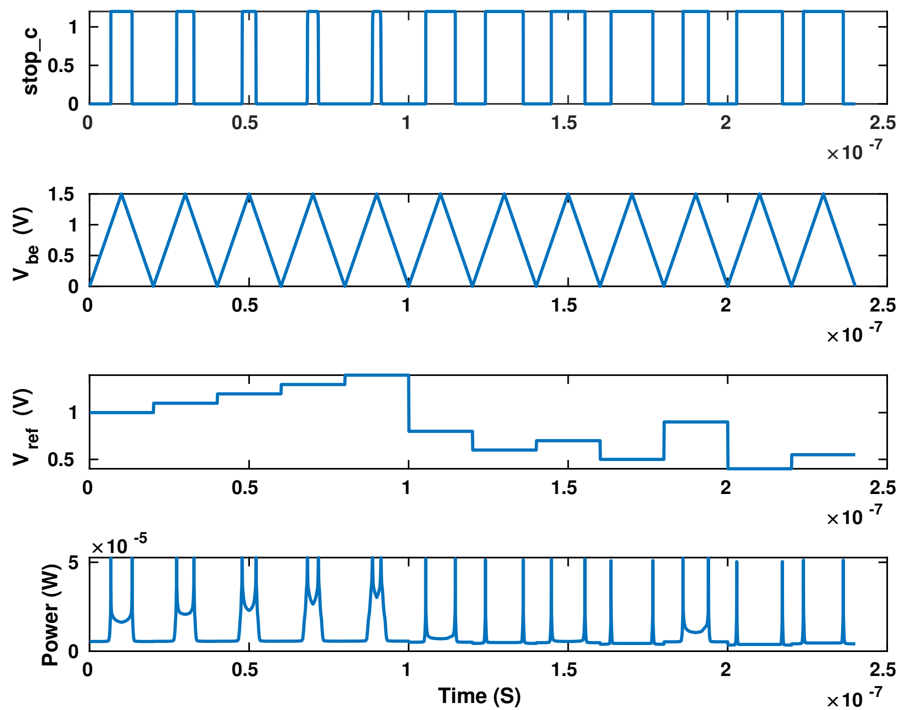


Figure 4.12: Power dissipation in comparator circuit

Table 4.3: Comparator circuit parameters

Transistor	Channel length	Channel width	Transistor	Channel length	Channel width
$MC1$	250 nm	100 nm	$MC4$	350 nm	85 nm
$MC2$	560 nm	85 nm	$MC5$	250 nm	85 nm
$MC3$	300 nm	100 nm	$MC6$	500 nm	85 nm

The stop logic circuit is shown in Fig. 4.13. This circuit generates a signal *stop_update* that stops the RESET operation. To generate a *stop_update* signal with a pulse width of 10 ns, we use transistor $ML9$ as a capacitor, minimizing the area occupied by the CMOS circuits. Fig.4.14 shows

TH-3339_186102005

the instantaneous power dissipated in the stop logic circuit. The average power dissipated is 1.7 μW, and the total energy consumption of the peripheral circuits is 3.3 pJ.

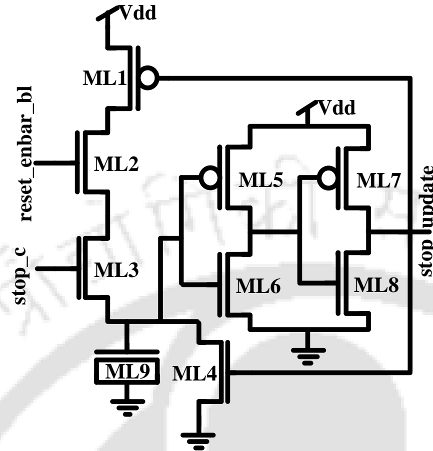
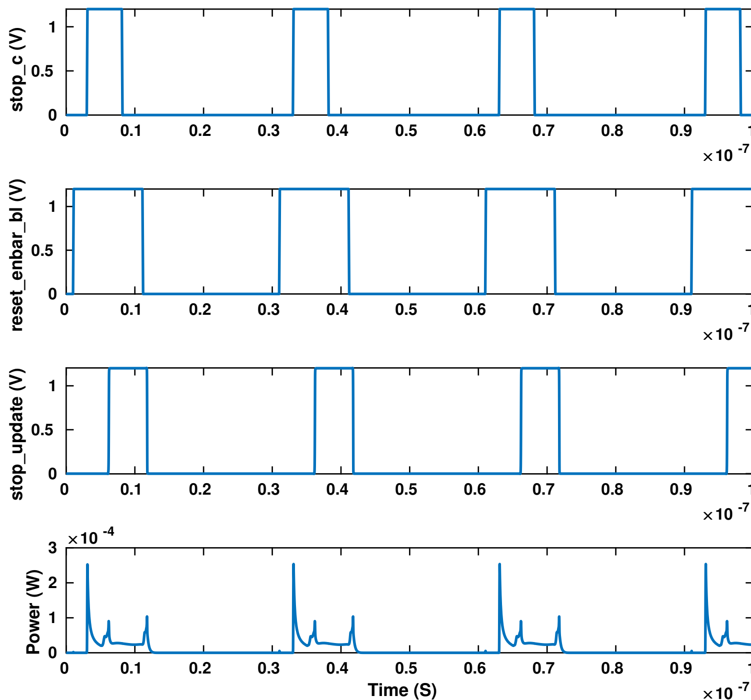


Figure 4.13: Schematic for stop logic circuit



(a)

Figure 4.14: Power dissipation in stop logic circuit

TH-3339_186102005

Table 4.4: Stop Logic circuit parameters

Transistor	Channel length (nm)	Channel width (nm)
$ML1$	400	80
$ML2$, $ML3$	200	80
$ML9$	300	150
$ML4$	200	80
$ML5$, $ML7$	500	80
$ML6$, $ML8$	250	80

4.4.3 RRAM current non-linearity

We know that non-linearity exists in the RRAM current. As shown in Fig. 4.15 (b), if the RRAM resistance is chosen from HRS to LRS, it results in a non-linear RRAM current, producing the maximum RRAM current error as 3.9 μA. The non-linearity is minimized for a reliable RRAM operation by lowering the RRAM resistance range. The error in the non-linear RRAM current can be reduced to 1.8 μA, as seen in Fig. 4.15 (a). In the proposed architecture, we have reduced the error in the RRAM current due to non-linearity by reducing the RRAM resistance range.

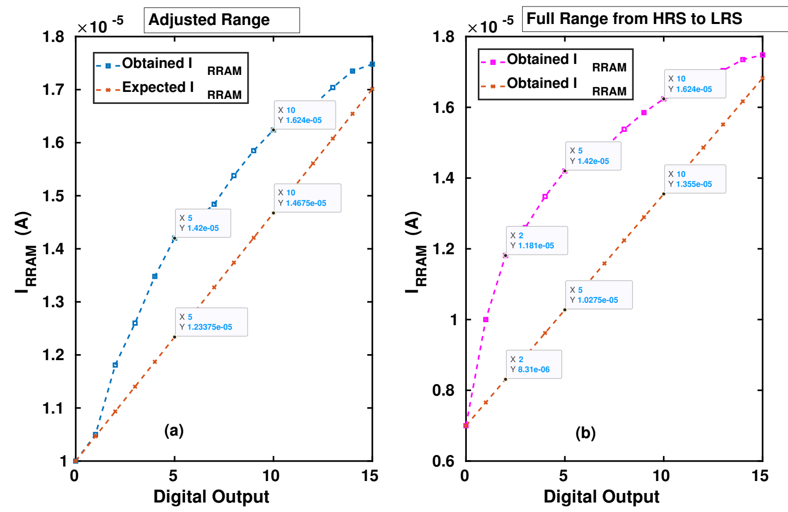


Figure 4.15: Digital output Vs RRAM current (a) Reduced non-linearity for selected range of resistance value (b) Non-linearity for full range of resistance value (HRS-LRS).

TH-3339_186102005

4.5 Variation analysis

4.5.1 CMOS circuit variation analysis

The variation in CMOS peripheral circuits significantly impacts the proper programming of the RRAM cell. Thus, checking the CMOS circuits for variation and mismatch is essential. Particularly, the reference voltage generator must be as precise as possible for accurate programming of the RRAM cell. We performed Monte Carlo and corner analysis of the CMOS peripheral circuits to evaluate the effects of process variation and mismatch of CMOS transistors. Fig. 4.16-Fig. 4.19 depicts the mean and the standard deviation for all the output voltages corresponding to the input of the reference voltage generator. It can be observed from Table 4.5, the maximum error in the output voltage in the presence of process variation and mismatch is 4.6%.

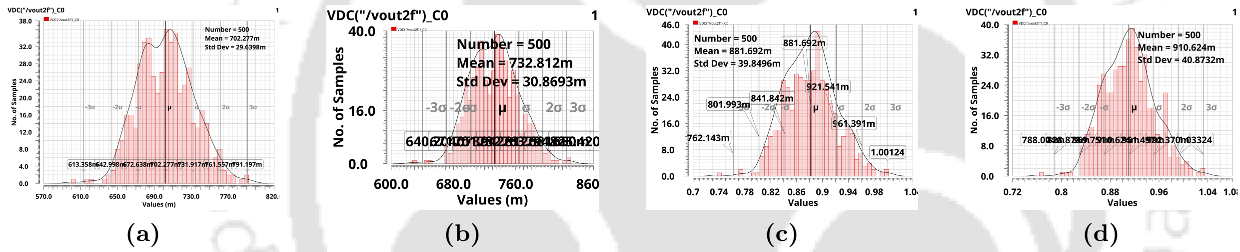


Figure 4.16: Monte Carlo simulation for reference voltage generator (a) 0000 (b) 0001 (c) 0010 (d) 0011

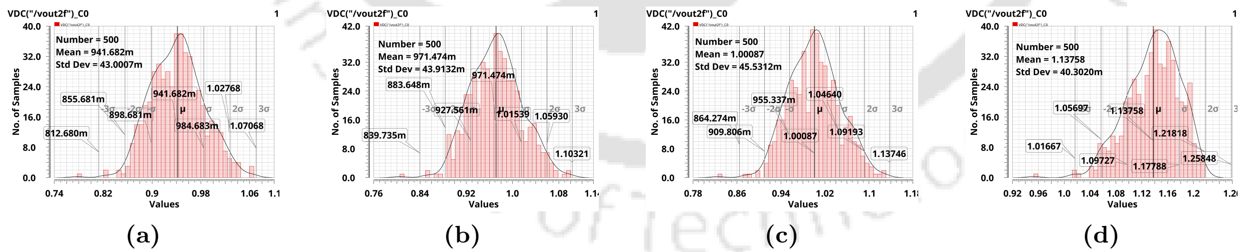
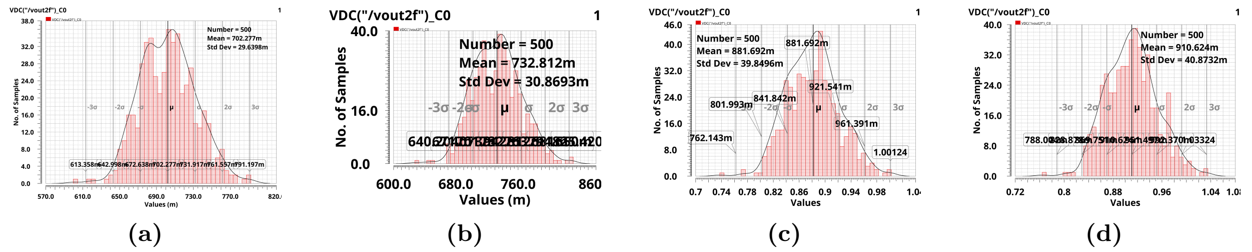


Figure 4.17: Monte Carlo simulation for reference voltage generator (a) 0100 (b) 0101 (c) 0110 (d) 0111

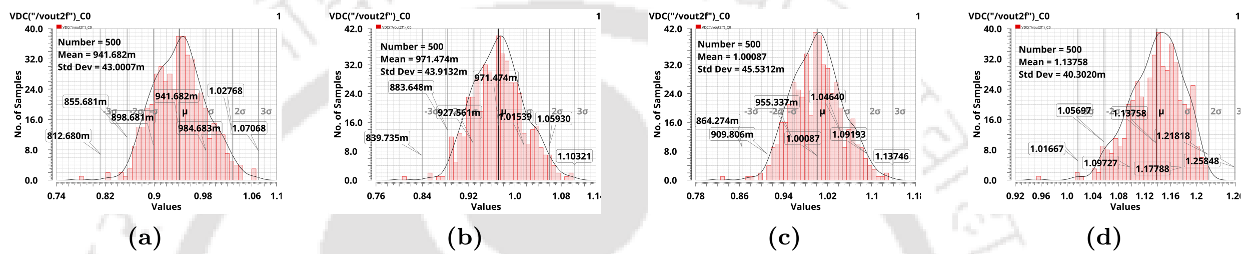
We perform process corner analysis for the stop logic circuit for all process corners, namely Typical (Typ), Slow NMOS Slow PMOS (SS), Fast NMOS Fast PMOS (FF), Slow NMOS Fast PMOS (SNFP), Fast NMOS Slow PMOS (FNSP). As depicted in Fig. 4.20, the stop logic circuit generates an output pulse for all the corners indicating that the CMOS peripheral circuits can operate in the presence of process variation and mismatch.

TH-3339_186102005



(a) (b) (c) (d)

Figure 4.18: Monte Carlo simulation for reference voltage generator (a) 1000 (b) 1001 (c) 1010 (d) 1011



(a) (b) (c) (d)

Figure 4.19: Monte Carlo simulation for reference voltage generator (a) 1100 (b) 1101 (c) 1110 (d) 1111

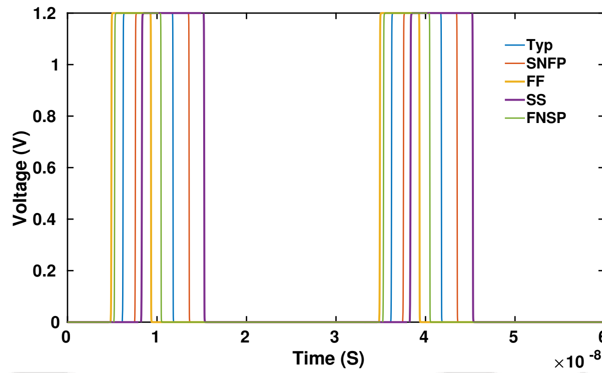


Figure 4.20: Output of stop logic circuit at process corners

Table 4.5: Variation in the reference voltage

Input	Vref (V)	Std Dev (V)	Mean (V)	% Error	Input	Vref (V)	Std Dev (V)	Mean (V)	% Error
0000	0.7	29.634 m	702 m	4.2	1000	0.94	43.2 m	942.7 m	4.5
0001	0.734	30.87 m	732.8 m	4.21	1001	0.97	43.2 m	969.5 m	4.64
0010	0.76	31.6 m	759.7 m	4.1	1010	1	44.7 m	998.5 m	4.2
0011	0.79	33.5 m	791.2 m	4.2	1011	1.03	42.6 m	1.025 m	4.48
0100	0.82	36.74 m	821.3 m	4.47	1100	1.06	45.97 m	1.061 m	4.40
0101	0.85	37.5 m	853.4 m	4.4	1101	1.09	46.712 m	1.087 m	4.18
0110	0.88	41.4 m	887.6 m	4.6	1110	1.12	45.46 m	1.117 m	3.88
0100	0.91	40.87 m	910.62 m	4.49	1111	1.14	43.38 m	1.135 m	3.54

TH-3339_186102005

4.5.2 RRAM variation and Noise analysis

The Unimore RRAM model includes the SET and RESET variability and random telegraph noise (RTN) by adding a white Gaussian noise on the CF cross section and the barrier thickness during SET and RESET events. To validate the functionality of the proposed synaptic array in the presence of cycle-to-cycle and device-to-device variation, we first set the variation and simulation parameters to achieve variations similar to the fabricated device [83]. Some of the variation and simulation parameters adapted from [121] are shown in Table 4.6. Fig. 4.21 shows the probability distribution for the LRS and HRS of the RRAM for 20 cycles.

Table 4.6: Parameters for the process variation

Variation parameter	Name	Unit
th_set	SET event detection threshold on the barrier derivative	10 nm/s
dx	Maximum variation allowed on the barrier thickness ($= 3\sigma$)	0.9 nm
ds	Maximum variation allowed on the CF cross section ($= 3\sigma$)	23 nm^2
$S0_{var}$	CF cross section at which the measured ds is obtained	72 nm^2
F_{max}	Maximum frequency of the noise for transient noise analysis	1300 Hz
$ddt_x_clip_th$	RESET event detection threshold on the barrier derivative	10^{-6} nm/s

The detailed calculation to obtain an average variation is shown in Table 4.7. As illustrated in Fig. 2.34, the mean and standard deviation for cycle-to-cycle variation during the HRS are 75 $K\Omega$ and 19.8 $K\Omega$;, therefore, the percentage variation obtained is 26.5%. Similarly, the percentage variation is calculated for cycle-to-cycle variation during LRS and device-to-device variation in HRS and LRS. The average percentage variation is obtained as 22%.

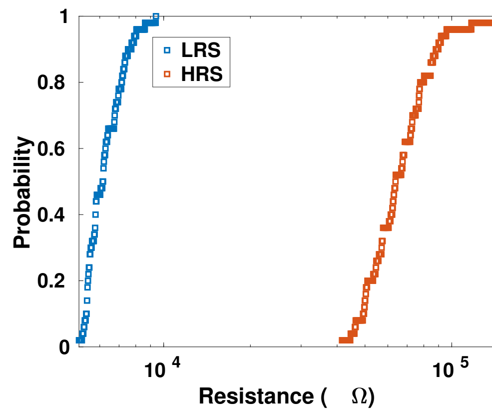


Figure 4.21: Resistance distribution for LRS and HRS

Table 4.7: Percentage Variation in HRS and LRS

Variation type	Resistance State	Mean	Standard Deviation	% Variation
Cycle to Cycle variation	HRS	$75K$	$19.9K$	26.5
Cycle to Cycle variation	LRS	694	122	17.57
Device to Device variation	HRS	$73K$	$19.2K$	26.28
Device to Device variation	LRS	693	124	17.89
			Average Variation	22 %

A single RRAM cell is programmed to evaluate the cycle-to-cycle variations, and this cycle is repeated 20 times, as shown in Fig. 4.22 (a). Similarly, to study the effects of device-to-device variations when several devices need to be programmed during the training, all the RRAM cells in a row of the RRAM array are programmed as depicted in Fig. 4.22 (b). It can be observed in both figures that despite the presence of cycle-to-cycle and device-to-device variations, the stop pulse is generated only when a critical voltage at the bottom electrode (V_{be}) is reached.

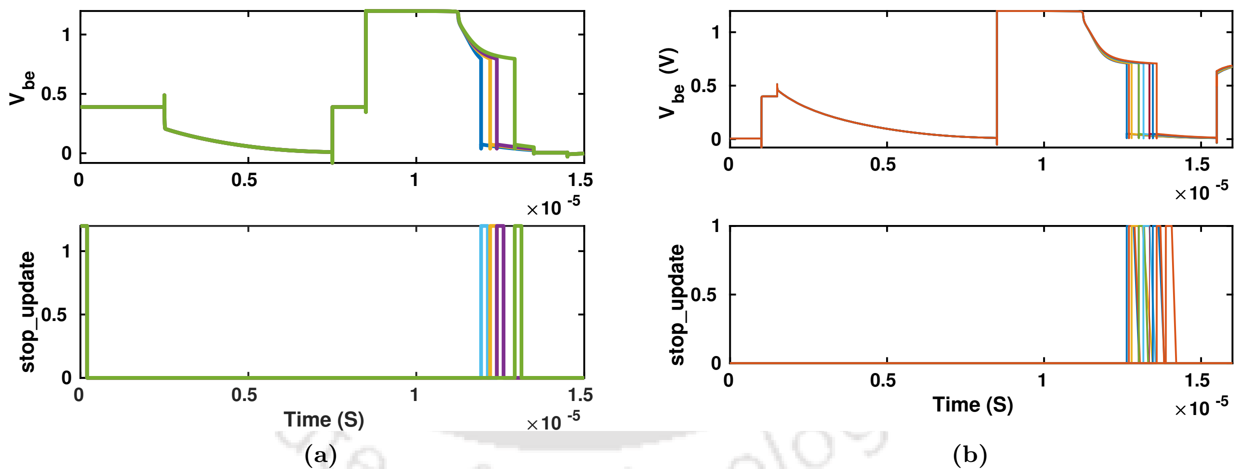


Figure 4.22: Difference in V_{be} and *stop_update* signal timing due to (a)cycle-to-cycle variation (b) device-to-device variation

The variation in the resistance of the RRAM cell is compensated by the time required to reach the desired resistive state. This makes the architecture tolerant to cycle-to-cycle and device-to-device variations in the RRAM. Fig. 4.7 depicts the *stop_update* signal for all 16 states. The RESET process is stopped at different values of V_{be}; whereas in Fig. 4.22, the RESET process is stopped at the same V_{be}, but at a different time due to cycle-to-cycle and device-to-device variation in RRAM.

RRAM devices are known to have intrinsic random telegraph noise (RTN), which interferes with the read current. Therefore, the effect of noise on the synapse should be analyzed. The current profile

TH-3339_186102005

for RTN noise in RRAM is depicted in Fig. 4.23.

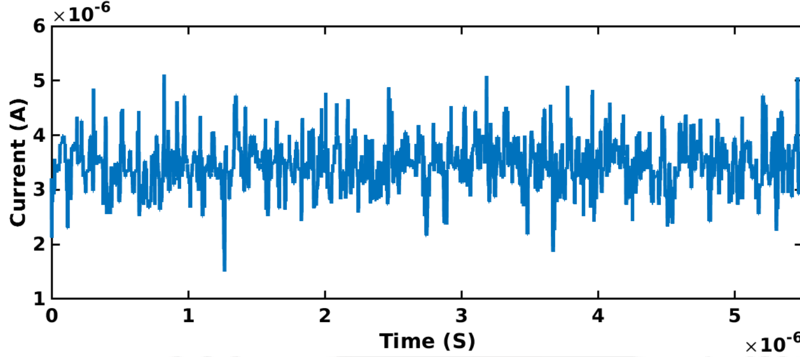


Figure 4.23: RTN noise current profile on RRAM

A comparison of RTN in the Unimore Verilog-A model with other state-of-the-art works is shown in Table 4.8. Since the Unimore RRAM Verilog-A model targets to work at a high frequency, it supports a pulsed SET and RESET operation at $10ns$. Also, the input pulse applied to the neuron and synapse cannot be less than $10ns$; therefore, we did all the measurements at a pulse width of $10ns$. It lies within the wide range of pulse width at which other state-of-the-art works operate. The range of resistances over which RTN is observed is also comparable to the state-of-the-art works.

Table 4.8: Comparison of RTN

Pramameter	[122]	[123]	[124]	[125]	Proposed work
HRS range	$1M\Omega - 5M\Omega$	$3.7M\Omega - 12M\Omega$	–	$30k\Omega - 47k\Omega$	$19.2k\Omega - 75k\Omega$
LRS range	$10\Omega - 700k\Omega$	$2k\Omega - 12k\Omega$	$20k\Omega - 40k\Omega$	$2.5k\Omega - 75k\Omega$	$122\Omega - 694\Omega$
Applied voltage (V)	3	2	0.6	0.75	0.8
No of devices	5	2	100	–	10
Pulse width	$50\mu s$	$22ns$	$1\mu s$	$0.5\mu s$	$10ns$

"–" states unavailability of the data.

4.6 Power, Latency, Energy and Area estimation

- **Power estimation**

  To estimate the maximum power dissipated in the RRAM array, we change the state of the RRAM from HRS to LRS. Fig. 4.24 depicts the instantaneous power consumed in the RRAM cell during RESET operation, which is realized using a 4T-1R cell.

  Power = Power (reference voltage generator) + Power (comparator) + Power (stop_logic) + Power (4T-1R cell)

TH-3339_186102005

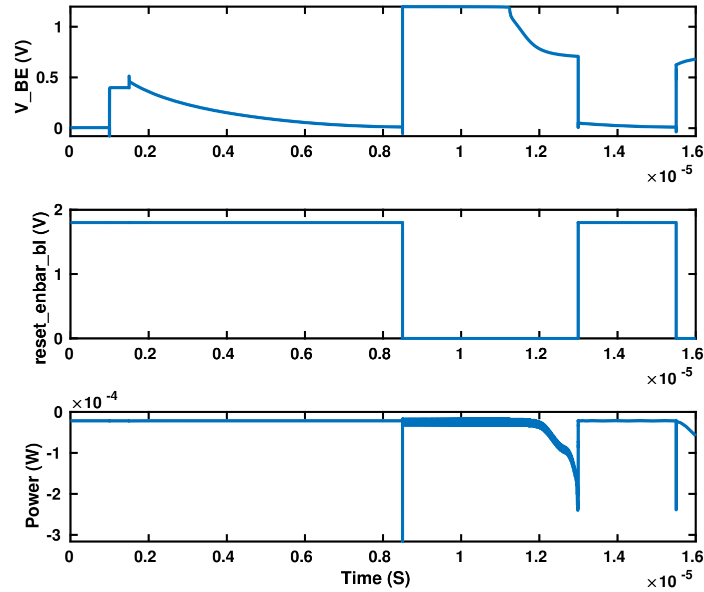


Figure 4.24: Power dissipation in RRAM array

Power $= 7.1\ \mu W + 4.7\ \mu W + 7.3\ \mu W + 0.1\ \mu W = 19.2\ \mu W$

Further, we estimate the power consumption of an RRAM array at the architecture level. In the RRAM array, a single row is programmed at a time; thus, we program each row for the maximum power dissipation (0000 to 1111) and take the average power. The power dissipated at the architectural level is 213 μW.

- **Latency estimation**

As we know, the latency is the time when the RRAM RESET operation starts until the stop pulse is generated to stop it, as shown in Fig. 4.6. Although the RRAM RESET stops after the generation of a stop pulse, the weight change block gets disabled until the RRAM RESET stop signal is disabled.

Therefore, we calculate the time till the end of the stop pulse to measure the latency of the proposed synaptic architecture. This also indicates that the current weight change process is complete, and the synapse can be programmed with a new weight. Since the state change from LRS to HRS takes the maximum time, it is considered the worst-case latency of the proposed architecture.

TH-3339_186102005

Latency = time required to reset + settling time of the reference voltage generator + delay of the comparator + delay of the stop signal generator + pulse width of the stop logic signal.

Latency = 0.98 μs + 10 ns + 4 ns + 3 ns + 10 ns

Latency = 1.07 μs

- **Energy consumed per cell**

  To compare the results with other state-of-the-art designs, we calculated the energy required for the change of RRAM state from HRS to LRS or LRS to HRS in a single cell. The energy is estimated using Eq. 4.2, where V_{TE} is the voltage across the RRAM cell, I_{TE} is the current through the RRAM, and T is the time required to program a single RRAM cell.

$$Energy = V_{TE} \times I_{TE} \times T \tag{4.2}$$

  We calculate the maximum energy dissipation in a single RRAM cell. The maximum energy is consumed when the RRAM state changes from LRS to HRS or HRS to LRS. The time required for the RESET operation is 1.07 μs, the voltage across the RRAM is $V_{TE} = 1.2\ V$, and the average current through the RRAM during RESET operation is $I_{TE} = 0.09\ \mu A$; thus, the total energy consumption per cell is 0.11 pJ.

- **Area estimation**

  The layout of the proposed architecture is designed to estimate the area of the CMOS peripheral circuits, as shown in Fig. 4.25. The area of the CMOS part of the architecture is 54 $\mu m \times 44\ \mu m$.

4.7 Comparison with the contemporary architectures

Table 4.9 summarizes the proposed design and compares it to state-of-the-art synaptic architectures. Most previous work is focused on the device level; our work is focused on circuit-level techniques to achieve $4-bit/cell$. A synaptic architecture with only $2-bit/cell$ is reported in [126], but it requires a read operation after every write to check if RRAM is programmed correctly, increasing the latency. The latency of the proposed architecture is 3.1× less than [127]. A $1T-1R$ cell is proposed in [128]. Although fewer transistors are utilized, it is more prone to RRAM variation. Similarly, $2T-1R$ and $4T-1R$ structures proposed in [129] and [130] provide multi-bit storage in a single cell, but the effects

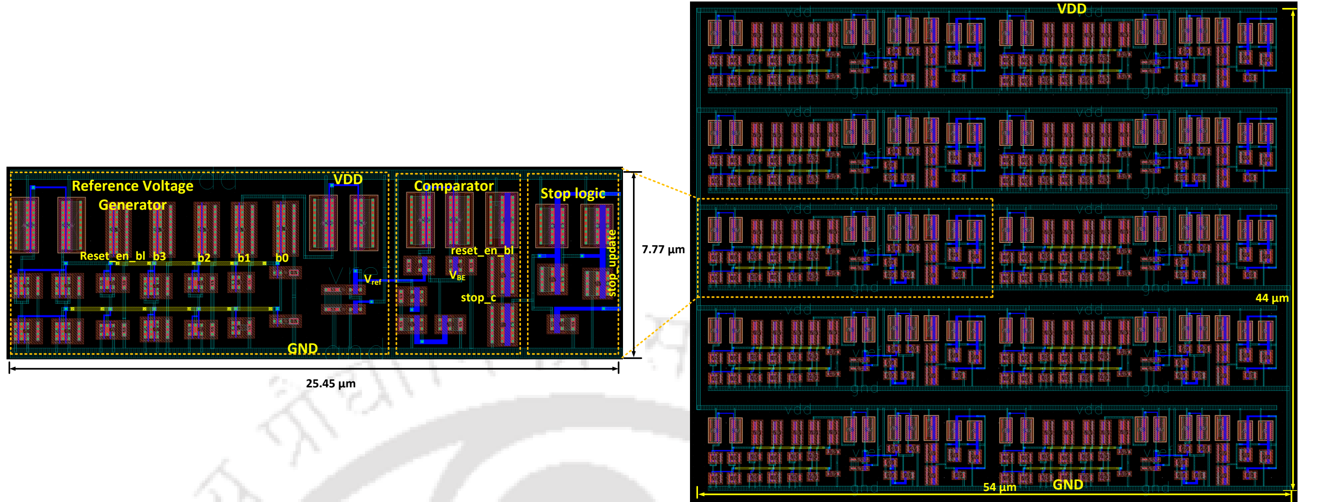


Figure 4.25: Layout for the CMOS part of the circuit.

of variation are not discussed. The proposed architecture is tolerant to 22% variation in RRAM cells, which is closer to the variations observed in a fabricated RRAM.

Table 4.9: Comparison with contemporary architectures

Publication	[84]	[131]	[127]	[126]	[132]	Proposed work
Bit/cell	3-bit	3-bit	3-bit	2-bit	2-bit	4-bit
Programming mode	IC_{set}	IC_{set}	V_{reset}	IC_{set}	IC_{set}	V_{reset}
Array Size	8×8	$2\ KB$	–	16×16 ($16.3\ KB$)	Single cell	10×10 ($25.6\ KB$)
Energy/Cell	$0.85\ pJ$	$30\ pJ$	$240\ pJ$	–	$65\ pJ$	$0.11\ pJ$
Latency	$3.39\ \mu s$	$5\ \mu s$	$3.1\ \mu s$	$15\ \mu s$	$2.5\ ms$	$1.07\ \mu s$
Design level	Device	Device	Device	Circuit	Device	Circuit

"–" states unavailability of the data.

4.8 Summary

In this chapter, a continuous sensing, feedback, and stop RESET mechanism for accurately programming the RRAM cell is presented. A $4T-1R$ structure is used to enable SET and RESET operations of the RRAM cell. Further, the numerical analysis exhibits that the proposed design works reliably in the presence of 22% cycle-to-cycle and device-to-device variations. This proves the robustness of the proposed programming scheme. The latency of the maximum weight change, i.e., the weight changing from LRS to HRS, is considered while evaluating the performance of the proposed design. It ensures occurring of no extra delay while implementing large neural networks. The peripheral circuits in the proposed architecture are implemented using UMC 65 nm CMOS technology, and

TH-3339_186102005

the overall power consumption is 213 μW. The energy consumption of the proposed design is at least 8.5× less than its nearest contemporary work available in the literature. The proposed design achieves $4 - bit/cell$, the highest among all the synapse implementations reported.

TH-3339_186102005

5

Spiking Neural Network

Indian Institute of Technology Guwahati

Contents

5.1 Introduction 102
5.2 SNN Training 102
5.3 Low precision weight encoding 105
5.4 Hardware network architecture 107
5.5 Benchmarking with Non-volatile memory-based SNN 109
5.6 Summary 111

TH-3339_186102005

5.1 Introduction

Nature has always influenced various engineering fields. For example, studies on the human visual cortex system have inspired the development of deep convolution neural networks [133]. The work on training CNNs proposed in [134] led to the development of many improved network architectures and training techniques to develop deep networks with super-human level accuracy [135].

Along with the development of artificial neural networks, third-generation neural networks, also called spiking neural networks (SNNs), have shown great potential to solve modern AI problems. Similar to the time-based information encoding in the biological nerve cells, SNNs use precise timing of the spikes transmitted between neurons. Since SNNs utilize neural dynamics and synaptic delays to realize temporal dimensions, they successfully emulate the different spike-firing dynamics in the brain [135].

In this chapter, an SNN employing the proposed RRAM-based $I\&F$ neuron and the programmable synapse is presented. We focus on the problem of handwritten digit classification and use the Modified National Institute of Standards and Technology (MNIST) dataset for training, using the unsupervised STDP algorithm. Prior implementations of SNN for the MNIST dataset have shown 90.76% [136] accuracy with a large number of 15000 output neurons compared to 10 output neurons used in the design of the proposed SNN. Another contemporary work [137] reports 75.65% accuracy with a memristive synapse; however, the synaptic array is not programmable, limiting the SNN to only one application. In contrast, the proposed synapse is programmable, enabling it to be used for multiple applications with slight changes in the architecture. The details of the proposed SNN are discussed below in the subsequent sections.

5.2 SNN Training

To evaluate the performance of SNN incorporating the proposed RRAM-based neuron and the synapse. First a two-later SNN is trained using Python [138]. As illustrated in Fig. 5.1, we design a fully connected two-layer SNN to classify handwritten digits using the MNIST dataset. The size of the images in the MNIST dataset is 28×28 pixels. Therefore, the network's input layer consists of 784 neurons, and the output layer consists of 10 neurons, one for each digit to be classified. The MNIST training dataset is divided into two parts: $50,000$ for training and the remaining $10,000$ for testing. During training, all the $50,000$ images are presented once to the network in each training epoch.

TH-3339_186102005

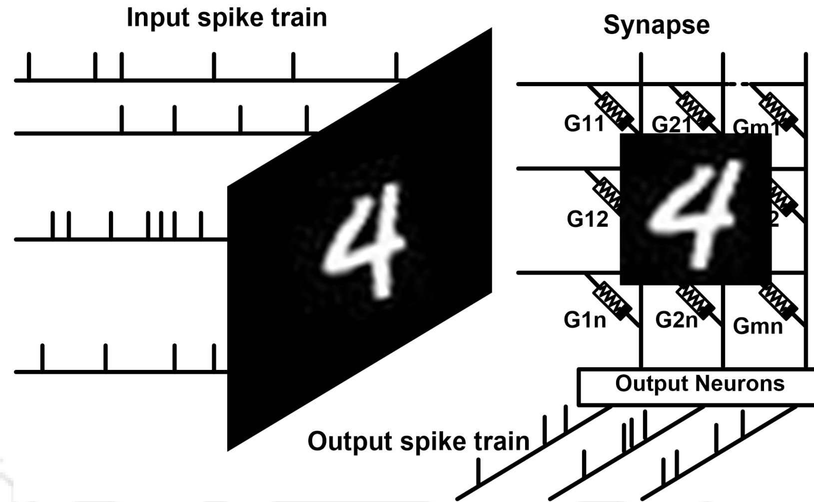


Figure 5.1: Spiking Neural Network

Before training the SNN, the static images are converted to spikes, similar to a biological neuron, where the input excitations are transmitted as time domain pulses, and the frequency of the output pulses is proportional to the input excitation. The biological neurons employ transformations, such as population, phase, rate, and time-of-spike coding [139]. Similarly, the Gaussian receptive fields or Poisson encoding directly converts the real-valued inputs to output spikes. As the dataset consists of static images, we convert each grayscale pixel value in the range of 0 to 255 to output spikes over a time of 150 ms. The firing rate of each pixel is given by Eq. 5.1, where FR is the firing rate, R_{max} is the maximum membrane potential, and RP_{min} is the minimum refractory period.

$$FR = \begin{Bmatrix} \frac{1}{RP_{min} * \frac{RF}{R_{max}}} & if \quad RF > 0 \\ 0 & if \quad RF \leqslant 0 \end{Bmatrix} \tag{5.1}$$

Spike Time Dependent Plasticity (STDP) is a learning algorithm widely used for training spiking neural networks. It is an unsupervised learning algorithm that changes the weight between two neurons based on the difference between the pre-synaptic spike and post-synaptic spike. For a given synapse, the weight is increased if the post-synaptic spike occurs after a pre-synaptic spike, whereas the weight is decreased if the post-synaptic spike occurs before a pre-synaptic spike.

$$w_{new} = \begin{cases} w_{old} + \sigma \Delta w (w_{max} - w_{old}) & if \quad \Delta w > 0 \\ w_{old} + \sigma \Delta w (w_{old} - w_{min}) & if \quad \Delta w \leq 0 \end{cases} \tag{5.2}$$

The simplified STDP rule adapted from [140], [141] is illustrated in Eq. 5.2. Here, the weights

TH-3339_186102005

lie between w_{min} and w_{max}, and the weight change rate σ controls the weight adaptation speed. The change in weight Δw is calculated using Eq. 5.3. Here, Δt is the time difference between pre-synaptic and post-synaptic spikes, $A+$ and $A-$ are the constants for positive and negative Δt values, and $\tau+$ and $\tau-$ are the steepness time constants for the weight change in both the directions. The values of all the parameters used for SNN implementation are shown in Table. 5.1.

$$\Delta w = \begin{cases} A^{-}exp(\frac{\Delta t}{\tau-}) & if \quad \Delta t \leq -2 \\ 0 & if \quad 2 < \Delta t < -2 \\ A^{+}exp(\frac{\Delta t}{\tau+}) & if \quad \Delta t \geq -2 \end{cases} \tag{5.3}$$

Table 5.1: Parameters for SNN

Parameter	R_{max}	RP_{min}	w_{max}	w_{min}	$A+$	$A-$	$\tau+$	$\tau-$
Value	50	-5.0	2	0	0.8	0.3	$+10$	-10

The spiking neural networks rely on the temporal information carried by spike trains and hence require time tracking. A time unit block is utilized to keep track of time steps during training and inference. Although the number of time units for every data sample is fixed to 150 ms in the training and inference phases, the number of clocks required to process every image is adaptive. During training, certain scenarios occur that need to be resolved carefully.

The first case is when fewer neurons are active in the first layer, the time unit employs fewer clocks. In this case, the potential of all the neurons that are not in the refractory period and the spike count is reduced, as depicted by Algorithm. 1.

Algorithm 1 No input, no output

```
1:  procedure STDP
2:      spike ← input spike
3:      synapse ← value of the weight
4:      time ← current time
5:      potential ← membrane potential of the neuron
6:      end ← total time units
7:      while time ≠ end do
8:          potential = potential - decay
9:          time ← time + 1
10:     end while
11: end procedure
```

Alternatively, it may also happen that at least one input neuron is active, but there is no activity in the output layer. In this case, the potential of the output neurons is updated according to the

TH-3339_186102005

corresponding synaptic weights, as shown in Algorithm. 2. This increases the neuron's membrane potential, producing spikes even if fewer input neurons are active.

Algorithm 2 Input, no output

```
 1: procedure STDP
 2:     spike ← input spike
 3:     synapse ← value of the weight
 4:     time ← current time
 5:     potential ← membrane potential of the neuron
 6:     end ← total time units
 7:     while time ≠ end do
 8:         potential = potential + spike * weight
 9:         time ← time + 1
10:     end while
11: end procedure
```

Finally, due to the reduction of the potential of the neurons, some spiking activity occurs in the output layer resulting in weight change. The weight change mechanism is enabled only during the training phase in which the synapses corresponding to the spiking output neuron are changed according to the STDP rule stated in Algorithm. 3.

Algorithm 3 Input, output

```
 1: procedure STDP
 2:     spike ← input spike
 3:     synapse ← value of the weight
 4:     time ← current time
 5:     potential ← membrane potential of the neuron
 6:     end ← total time units
 7:     while time ≠ end do
 8:         potential = potential + spike * weight
 9:         time ← time + 1
10:         if the neuron first to fire then
11:             Inhibit rest of the neurons
12:             synapse ← synapse + synapse
13:         end if
14:     end while
15: end procedure
```

5.3 Low precision weight encoding

The SNN training in Python does not restrict the precision of the synaptic weight, but the RRAM cells are limited to $4 - bits/cell$. Various state-of-the-art works on neural networks have shown that successful classification can be achieved with limited precision of data representation [139]. Therefore,

TH-3339_186102005

to convert the trained weights to $4-bit$ equivalent, we followed the methodology discussed in [139]. Since the synapse supports $4-bit/cell$, we perform $4-bit$ quantization of the weights obtained from training. Fig. 5.2 shows the distribution of the trained synaptic weights, where the mean of the trained weight is $\mu = 1.15$ and the standard deviation is $\sigma = 0.135$.

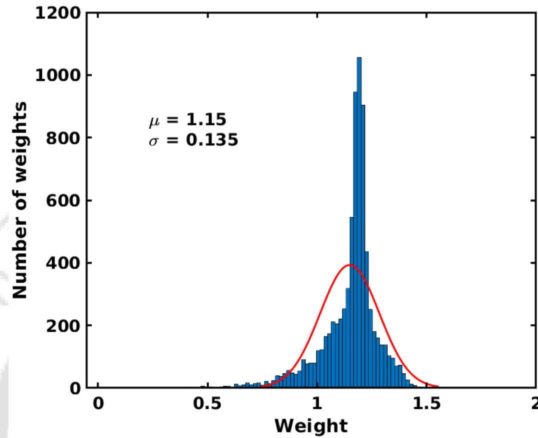


Figure 5.2: Distribution of trained weights

Following the approach discussed in [139], quantizing a selected range of 2σ from the mean value of the trained weights is more effective than quantizing the entire range. Therefore, the weights are clamped between [0,2] during learning, but while transferring the weights to the RRAM cells, the trained weights are quantized in the range [0.88,1.42]. Table. 5.2 shows the trained weights and the corresponding $4-bit$ digital weights obtained after quantization, which are used to program the RRAM synapse.

Table 5.2: Normalised weight

Trained weight	Digital equivalent	Trained weight	Digital equivalent
0.88 – 0.91375	0000	1.15 – 1.18375	1000
0.91375 – 0.9475	0001	1.18375 – 1.2175	1001
0.9475 – 0.98125	0010	1.2175 – 1.25125	1010
0.98125 – 1.015	0011	1.25125 – 1.285	1011
1.015 – 1.04875	0100	1.285 – 1.31875	1100
1.04875 – 1.0825	0101	1.31875 – 1.3525	1101
1.0825 – 1.11625	0110	1.3525 – 1.38625	1110
1.11625 – 1.15	0111	1.38625 – 1.42	1111

TH-3339_186102005

5.4 Hardware network architecture

The proposed RRAM-based hardware architecture of SNN, implemented using the analog-mixed-signal (AMS) environment, is illustrated in Fig. 5.3. The architecture can be divided into two sections, the blocks shown in blue are implemented digitally using Verilog, and the blocks illustrated in red are realized in analog. The digital-on-top approach enables the implementation of a test bench that can be employed to read multiple files having pre-trained weights and input spikes. The proposed architecture operates in two phases: the programming phase, where the RRAM-based synapses are programmed for the corresponding trained weights, and the second phase is the inference, where images in the form of input spikes are passed to the network.

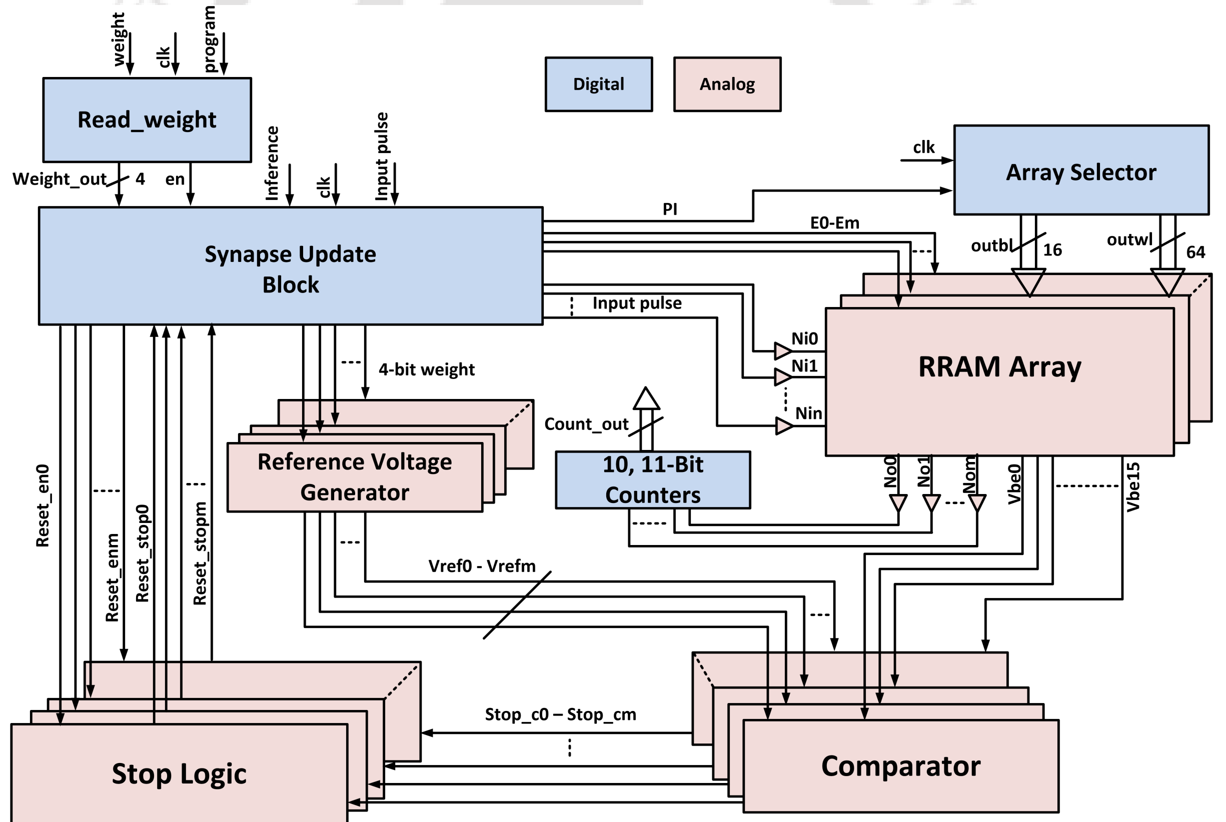


Figure 5.3: AMS architecture for Spiking Neural Network

When the *program* signal is high, the *Read_weight* block reads the trained weight one-by-one on every rising edge of the clock. As shown in Algorithm. 4, during the programming phase, the *count* is initialized to the number of synapses to be programmed (7840), and the RRAM-based synapses are programmed to the corresponding trained weights when the *program* signal is active.

Since the synapses in the proposed SNN architecture are arranged in a 784 × 10 array, the *Array Selector* is programmed to select RRAM cells column-wise, and the programming is per-

TH-3339_186102005

Algorithm 4 *Read_weight* block

```
1: procedure Read_weight
2:     Input pulse ← Spike encoded input image
3:     weight ← Trained weight
4:     weight_out ← 4-bit weight
5:     program ← Signal indicating program and inference
6:     count ← Initialized to number of synapses
7:     while count ≠ zero and program = 1 do
8:         count = count − 1
9:         weight_out = weight
10:        Reset_en = 1
11:        if count = 0 or program = 1 then
12:            start inference
13:        end if
14:    end while
15: end procedure
```

formed starting from the synapses connected to neurons $No0$ to $No9$. As shown in Algorithm. 5, every synapse in each column is programmed to the corresponding weight. The value of the *counter* is decremented by 1 after programming each synapse. Every time the *counter* becomes 0, the *column* signal is incremented by 1, and the *counter* is initialized to 784. When $column = 10$ and $counter = 0$, it indicates that the programming of the synapses is complete, and inference can be initiated. Since the maximum time required for programming the synapse, including the delay of the peripheral circuits, is 1.07 μs, the clock width is set to 1.1 μs to allow proper programming of the RRAM cells.

Algorithm 5 *Array Selector*

```
1: procedure Array_Selector
2:     counter ← Initialized to 784
3:     column ← Indicate the number of output neurons
4:     while column ≠ 10 do
5:         column = column + 1
6:         while counter ≠ 0 do
7:             program each RRAM cell
8:             counter = counter − 1
9:         end while
10:        if column = 10 then
11:            stop programming
12:        else
13:            initialise counter to 784
14:        end if
15:    end while
16: end procedure
```

TH-3339_186102005

Finally, the *Synapse Update* block controls the peripheral circuits for programming the synaptic array. During the programming phase, it takes input from the *Read_weight* block, and during the inference phase, it reads the spike-encoded images and passes it to the input neurons connected to the RRAM array. As shown in Fig. 5.3, the output of the *Synapse Update* block is connected to the *Reference Voltage Generator*, *Stop Logic*, *Array Selector*, and the RRAM array. The weights read by the *Read_weight* block are converted to equivalent binary values and transferred to the *Reference Voltage Generator* to produce equivalent analog values of the trained weight for programming the RRAM cells. During the inference phase, the input images in the form of the voltage pulse are passed through corresponding synapses, and the output of the corresponding neuron is observed for the output spike. The output of the neurons is fed to $11 - bit$ counters for keeping the track of output neurons spikes, which helps to measure the accuracy. As illustrated in Algorithm. 6, *Reset_en* is activated only when the *program* signal is high and passes either the weights or the spikes to the RRAM array, depending on the programming and inference phase. Fig. 5.4 shows the trained weights for all 10 MNIST digits, and the confusion matrix obtained after inference for the test dataset is shown in Table. 5.3.

Algorithm 6 *Synapse Update* block

1: **procedure** *Synapse Update*
2: inference ← Signal indicating program and inference
3: input pulse ← Spike encoded input image
4: *Reset_en* ← Signal enables programming of the synapse
5: **if** $program = 1$ **then**
6: *Reset_en* ← 1
7: pass binary weight to peripheral circuits
8: **else**
9: *Reset_en* ← 0
10: read input spikes and pass it to the RRAM array
11: **end if**
12: **end procedure**

5.5 Benchmarking with Non-volatile memory-based SNN

The performance of the proposed SNN is compared with the contemporary works that incorporate non-volatile memories to implement the synapse, as shown in Table. 5.4. To the best of our knowledge, no state-of-the-art works have used a combination of RRAM-based neurons and synapses to realize an SNN on hardware. We achieve an accuracy of 89% after mapping the trained weights to the

TH-3339_186102005

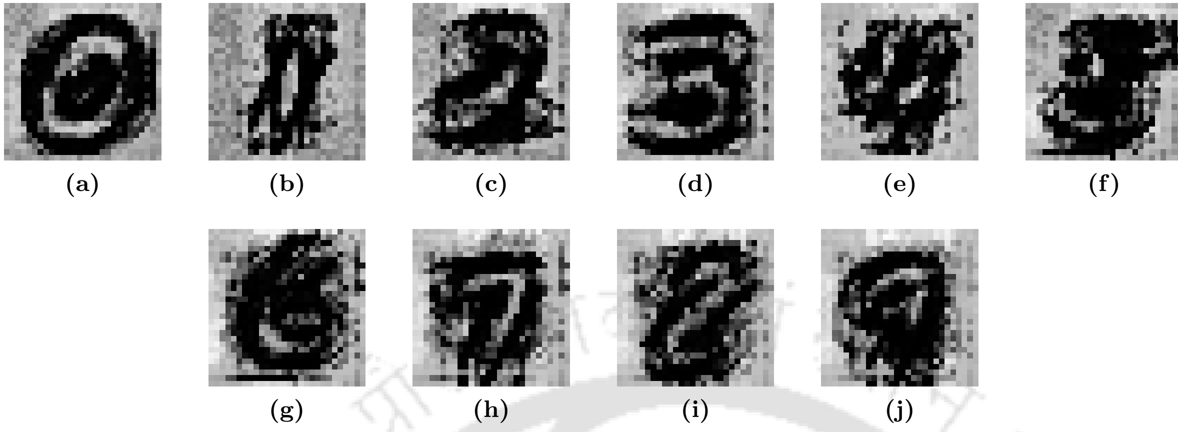


Figure 5.4: Reconstructed weights for corresponding output neurons

Table 5.3: Confusion matrix

Actual Predicted	0	1	2	3	4	5	6	7	8	9
0	920	0	3	0	2	2	37	31	4	4
1	0	721	40	0	0	0	7	4	0	4
2	2	3	872	4	1	1	0	9	1	1
3	0	2	0	940	0	7	4	11	6	4
4	0	1	2	0	886	0	1	4	5	7
5	0	1	0	32	0	845	3	0	1	3
6	16	1	1	0	41	23	891	0	1	0
7	2	139	6	2	0	1	0	931	3	7
8	17	0	1	31	1	13	1	23	947	3
9	23	63	2	1	51	0	13	14	6	975
No Spike	0	204	105	0	0	0	1	1	0	1
Total	980	1135	1032	1010	982	892	958	1028	974	1009

conductance of the RRAM synapse. By employing a $4-bit/cell$ synapse, the accuracy achieved is comparable with [142], [143], [136], which use full precision digital neurons. Although the accuracy of [136] is slightly more than the proposed architecture, the number of output neurons is much more than the proposed architecture. Multiple parameters, such as the training algorithm and precise synaptic programming, help achieve better accuracy. Moreover, the $4-bit/cell$ density leads to more accuracy than [144], [145], where $1-bit/cell$ synapses are used for synapse implementation. To compare the energy consumed by the proposed SNN architecture with the contemporary works, the energy consumed per synaptic operation of an RRAM array is calculated to be 3.3 pJ. The energy consumption of all the $I\&F$ neurons and the peripheral circuits in all the columns of the proposed architecture is 150 pJ, which is 16.6% less than [146].

Table 5.4: Benchmarking with contemporary SNN implementations

Attributes	Neuron	Synapse	Programmable	bit/cell	Input-output neuron	Energy per synaptic event	Accuracy
This work	RRAM-Based	1T-1R	Yes	$4-bit/cell$	$784-10$	$150\ pJ$	89%
[142]	Simulator	Simulator	Yes	$32bit$	$784-6400$	–	94.8%
[143]	Simulator	Simulator	Yes	$32bit$	$784-300$	–	93.5%
[144]	Digital	$1T-1R$	No	$1-bit/cell$	$784-50$	–	75%
[145]	Analog	9PCM	No	$1-bit/cell$	$784-10$	$205\ pJ$	70%
[137]	–	1 Memristor	No	–	$784-30$	$5\ nJ$	75.65%
[136]	Analog	1RRAM	No	$1-bit/cell$	$784-15000$	–	90.76%
[146]	Analog	1T-1R	No	$3-bit/cell$	$784-10$	$180\ pJ$	84%

"–" states unavailability of the data.

5.6 Summary

This chapter presents a hardware implementation of a spiking neural network using RRAM-based neurons and $4-bit/cell$ RRAM synapse. The proposed AMS architecture for SNN bridges the gap in the state-of-the-art works focusing only on neuron design, synapse design, or SNN design. We show the applicability of the proposed neuron and synapse in the realization of large neuromorphic systems. We adopt the quantization method to map the full precision trained weights to the synapse's limited conductance levels and obtain a classification accuracy comparable to the contemporary hardware implementations of SNN employing non-volatile memories for synapse implementation. The low energy and lesser area make the proposed SNN a suitable candidate for realizing large-scale neuromorphic systems.

TH-3339_186102005

TH-3339_186102005

6 Conclusion and Future Work

भारतीय प्रौद्योगिकी संस्थान गुवाहाटी • Indian Institute of Technology Guwahati •

Contents

6.1 Conclusion 114
6.2 Directions for future work 115

TH-3339_186102005

6.1 Conclusion

Developing low-power, large-scale neuromorphic systems for modern AI applications remains a significant challenge. AI's future growth depends on the optimization level that can be achieved at the hardware on which the AI applications run. In this thesis, we present a possible direction to develop low-power large-scale neuromorphic systems efficiently for the realization of AI applications.

For achieving the above-mentioned goal, a scalable, energy-efficient RRAM-based $I\&F$ neuron is proposed in this thesis. The series combination employed for the spike generation results in low energy consumption compared to the digital and analog implementation of $I\&F$ neurons. The performance of the proposed neuron is analyzed in the presence of random telegraph noise, which is prominent in RRAM devices and adversely affects the functionality of the peripheral circuits. The detailed analysis shows that the proposed neuron produces reliable output spikes even in the presence of RTN.

The RRAM suffers from cycle-to-cycle and device-to-device variations. Therefore, the functionality of the neuron is validated in the presence of these variations. Due to these variations, the switching voltage of the RRAM changes, resulting in a change in the timing of the neuron spikes. However, the spiking of the neuron indicates that the proposed circuit works satisfactorily even in the presence of cycle-to-cycle and device-to-device variations.

Next, the corner analysis of the proposed neuron is performed. It is observed that the proposed neuron behaves differently in all the corners, as expected. At the SS corner, the MOSFET in RRAM-based $I\&F$ neuron has a higher threshold, which results in a delay in spike generation, whereas, in the FF corner, the threshold of the MOSFET is reduced; hence the spikes are generated at a faster rate.

Further, a programmable RRAM array is proposed by incorporating a $4T-1R$ structure, which helps reprogram the array and increases the scope of application of the proposed synaptic architecture. Moreover, the area of the proposed synaptic array is optimized by sharing the transistors among the rows and columns. The variability in the switching behaviour changes the voltage at which RRAM switches the states from HRS to LRS or LRS to HRS, severely affecting the functionality of the peripheral circuits. To address this issue, we propose an RRAM-based synaptic architecture with continuous sensing and feedback scheme to stop RRAM programming when the required conductance is achieved. The cycle-to-cycle and device-to-device variations analyses are conducted for the individual CMOS circuit to analyze the robustness of the proposed architecture. Additionally, we perform the

TH-3339_186102005

Monte Carlo analysis to study the effects of CMOS and RRAM variations at the architecture level to analyze the reliability of the proposed architecture.

Moreover, nonlinearity is a major issue in RRAM-based synaptic architectures. With the proposed circuit-level mechanism to program the RRAM-based synapse, a $4-bit/cell$ precision is achieved while reducing the nonlinearity in the RRAM current by selecting the intermediate resistance states, lowering the nonlinearities in the RRAM current.

Finally, we design a spiking neural network by employing the proposed RRAM-based $I\&F$ neuron and synapse. The training of the SNN is performed using Python, and the weights are mapped to the RRAM array. For transferring the weights to the RRAM-based synapse, we employ a standard quantization method. For the proposed RRAM-based SNN, 89% accuracy is attained, which is comparable to the contemporary works employing non-volatile memories for the synapse implementation.

6.2 Directions for future work

From the above-mentioned discussion, it can be observed that the thesis focuses on the design of efficient neuron and synaptic circuits for large-scale neuromorphic systems. A few potential directions to which the contribution of the thesis can be extended are discussed below.

- The successful demonstration of the proposed $I\&F$ neuron and the programmable synapse for implementing SNN shows that the proposed circuits can be employed efficiently for the development of large-scale neuromorphic systems.
- Since the proposed architecture is programmable, it can be easily employed to implement on-chip training. The off-chip training requires transmitting huge data and model parameters between the cloud and the edge devices, which is hard to deploy in complex environments while protecting the privacy of IoT applications. If on-chip training is performed, it would eliminate the need for large data transfers and improve accuracy.
- RRAM-based arrays are also vulnerable to transient faults, also called soft errors. These errors occur due to high-energy particle strikes, resulting in bit flips at the hardware layer. These errors can change the weight values and neuron operations to some extent, adversely affecting the neural networks, leading to incorrect outputs and the degradation of accuracy. These non-idealities can be reduced but cannot be diminished at the device or circuit level; hence should

TH-3339_186102005

be considered while designing the training algorithms and programming techniques to make the RRAM-based array architectures more robust. The RRAM programming mechanism in the proposed architecture can be effectively explored to resolve the transient faults.

- Moreover, to achieve a high on/off resistance ratio, a high negative voltage is applied across the RRAM devices, which results in stuck-at-short faults. Since the current through the RRAM is the product of the applied voltage and the conductance of the RRAM, the stuck-at-short faults can have adverse effects on the reading process. The proposed RRAM programming mechanism can be used to identify such faults and take precautionary measures to avoid erroneous outputs.
- Furthermore, he proposed RRAM architecture can be employed to implement in-memory computing architectures and AI accelerators for implementing deep neural networks with lower resource consumption. In addition to this, an I/O interface can be developed so that the RRAM-based arrays can be easily interfaced with the processors. This would significantly reduce the power consumption of the processors designed for AI applications.
- Finally, A programmable architecture and circuit-level blocks can be developed to train the network for different datasets. The design space exploration can be done within the network to further reduce the overall system's area and power. Also, as there is still scope in the current architecture to improve the accuracy, optimizing the current training algorithm and developing a programmable architecture to train different datasets would go a long way in developing versatile neuromorphic hardware.

TH-3339_186102005

Bibliography

[1] H. Lim, V. Kornijcuk, J. Seok, S. Kim, I. Kim, C. Hwang, and D. Jeong, "Reliability of neuronal information conveyed by unreliable neuristor-based leaky integrate-and-fire neurons: a model study," in *Sci Rep*, vol. 5, no. 9776, 2015.

[2] Neumann and v. John, *The Computer and the Brain*. USA: Yale University Press, 1958.

[3] C. M. Zhang, G. C. Qiao, S. G. Hu, J. J. Wang, Z. W. Liu, Y. A. Liu, Q. Yu, and Y. Liu, "A versatile neuromorphic system based on simple neuron model," *AIP Advances*, vol. 9, no. 1, 01 2019, 015324. [Online]. Available: https://doi.org/10.1063/1.5052609

[4] P. Merolla and et.al, "Artificial brains. a million spiking-neuron integrated circuit with a scalable communication network and interface," in *Science.*, 2014 Aug.

[5] B. Rajendran, A. Sebastian, M. Schmuker, N. Srinivasa, and E. Eleftheriou, "Low-power neuromorphic hardware for signal processing applications: A review of architectural and system-level design approaches," *IEEE Signal Processing Magazine*, vol. 36, no. 6, pp. 97–110, 2019.

[6] E. Painkras, L. A. Plana, J. Garside, S. Temple, F. Galluppi, C. Patterson, D. R. Lester, A. D. Brown, and S. B. Furber, "Spinnaker: A 1-w 18-core system-on-chip for massively-parallel neural network simulation," *IEEE Journal of Solid-State Circuits*, vol. 48, no. 8, pp. 1943–1953, 2013.

[7] M. Davies, N. Srinivasa, T.-H. Lin, G. Chinya, Y. Cao, S. H. Choday, G. Dimou, P. Joshi, N. Imam, S. Jain, Y. Liao, C.-K. Lin, A. Lines, R. Liu, D. Mathaikutty, S. McCoy, A. Paul, J. Tse, G. Venkataramanan, Y.-H. Weng, A. Wild, Y. Yang, and H. Wang, "Loihi: A neuromorphic manycore processor with on-chip learning," *IEEE Micro*, vol. 38, no. 1, pp. 82–99, 2018.

[8] A. Neckar, T. C. Stewart, B. V. Benjamin, and K. Boahen, "Optimizing an analog neuron circuit design for nonlinear function approximation," in *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2018, pp. 1–5.

[9] A. Cassidy, A. G. Andreou, and J. Georgiou, "Design of a one million neuron single fpga neuromorphic system for real-time multimodal scene analysis," in *2011 45th Annual Conference on Information Sciences and Systems*, 2011, pp. 1–6.

[10] J. Li, Y. Katori, and T. Kohno, "An fpga-based silicon neuronal network with selectable excitability silicon neurons," *Frontiers in Neuroscience*, vol. 6, 2012.

[11] V. Kornijcuk and D. S. Jeong, "Pointer based routing scheme for on-chip learning in neuromorphic systems," in *2018 International Joint Conference on Neural Networks (IJCNN)*, 2018, pp. 1–6.

[12] S. B. Furber, F. Galluppi, S. Temple, and L. A. Plana, "The spinnaker project," *Proceedings of the IEEE*, vol. 102, no. 5, pp. 652–665, 2014.

[13] e. PAUL A. MEROLLA, "A million spiking-neuron integrated circuit with a scalable communication network and interface," *Science*, vol. 345, no. 5, pp. 668–673, 2014.

[14] S. Friedmann, J. Schemmel, A. Grübl, A. Hartel, M. Hock, and K. Meier, "Demonstrating hybrid learning in a flexible neuromorphic hardware system," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 11, no. 1, pp. 128–142, 2017.

TH-3339_186102005

[15] B. V. Benjamin, P. Gao, E. McQuinn, S. Choudhary, A. R. Chandrasekaran, J.-M. Bussat, R. Alvarez-Icaza, J. V. Arthur, P. A. Merolla, and K. Boahen, "Neurogrid: A mixed-analog-digital multichip system for large-scale neural simulations," *Proceedings of the IEEE*, vol. 102, no. 5, pp. 699–716, 2014.

[16] S. Moradi, N. Qiao, F. Stefanini, and G. Indiveri, "A scalable multicore architecture with heterogeneous memory structures for dynamic neuromorphic asynchronous processors (dynaps)," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 12, no. 1, pp. 106–122, 2018.

[17] C. Frenkel, M. Lefebvre, J.-D. Legat, and D. Bol, "A 0.086-mm^2 12.7-pj/sop 64k-synapse 256-neuron online-learning digital spiking neuromorphic processor in 28-nm cmos," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 13, no. 1, pp. 145–158, 2019.

[18] D. Seok Jeong, I. Kim, M. Ziegler, and H. Kohlstedt, "Towards artificial neurons and synapses: a materials point of view," *RSC Adv.*, vol. 3, pp. 3169–3183, 2013. [Online]. Available: http://dx.doi.org/10.1039/C2RA22507G

[19] Y. Kim, W. H. Jeong, S. B. Tran, H. C. Woo, J. Kim, C. S. Hwang, K.-S. Min, and B. J. Choi, "Memristor crossbar array for binarized neural networks," *AIP Advances*, vol. 9, no. 4, 04 2019, 045131. [Online]. Available: https://doi.org/10.1063/1.5092177

[20] C. Mead, "Neuromorphic electronic systems," *Proceedings of the IEEE*, vol. 78, no. 10, 1990.

[21] X. Jin, M. Lujan, L. A. Plana, S. Davies, S. Temple, and S. B. Furber, "Modeling spiking neural networks on spinnaker," *Computing in Science and Engineering*, vol. 12, no. 5, pp. 91–97, 2010.

[22] T. S. T. Mak, G. Rachmuth, K.-P. Lam, and C.-S. Poon, "A component-based fpga design framework for neuronal ion channel dynamics simulations," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 14, no. 4, pp. 410–418, 2006.

[23] J. Schemmel, D. Brüderle, A. Grübl, M. Hock, K. Meier, and S. Millner, "A wafer-scale neuromorphic hardware system for large-scale neural modeling," in *2010 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2010, pp. 1947–1950.

[24] L. Camunas-Mesa, A. Acosta-Jimenez, T. Serrano-Gotarredona, and B. Linares-Barranco, "Fully digital aer convolution chip for vision processing," in *2008 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2008, pp. 652–655.

[25] H. Torikai, H. Hamanaka, and T. Saito, "Reconfigurable digital spiking neuron and its pulse-coupled network: Basic characteristics and potential applications," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 53, no. 8, pp. 734–738, 2006.

[26] S. Hashimoto and H. Torikai, "A novel hybrid spiking neuron: Bifurcations, responses, and on-chip learning," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 57, no. 8, pp. 2168–2181, 2010.

[27] H. Torikai, Y. Shimizu, and T. Saito, "Various spike-trains from a digital spiking neuron: analysis of inter-spike intervals and their modulation," in *The 2006 IEEE International Joint Conference on Neural Network Proceedings*, 2006, pp. 3860–3867.

[28] T. Matsubara and H. Torikai, "Dynamic response behaviors of a generalized asynchronous digital spiking neuron model," in *Neural Information Processing*, B.-L. Lu, L. Zhang, and J. Kwok, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 395–404.

[29] M. I. Carver Mead, *Analog VLSI Implementation of Neural Systems.* Springer New York, NY, 1989.

[30] C. Mead, "Neuromorphic electronic systems," *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1629–1636, 1990.

[31] D. R. Mahowald M, "A silicon neuron." in *Nature 354*, 1991, p. 515–518.

[32] D. R. Rasche C, "An improved silicon neuron." in *Analog Integrated Circuits and Signal Processing 23*, 2000, p. 227–236.

[33] A. van Schaik, "Building blocks for electronic spiking neural networks," *Neural Networks*, vol. 14, no. 6, pp. 617–628, 2001. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0893608001000673

TH-3339_186102005

[34] G. Indiveri, "Modeling selective attention using a neuromorphic analog vlsi device," *Neural Computation*, vol. 12, no. 12, pp. 2857–2880, 2000.

[35] S.-C. Liu, J. Kramer, G. Indiveri, T. Delbrück, T. Burg, and R. Douglas, "Orientation-selective avlsi spiking neurons," *Neural Networks*, vol. 14, no. 6, pp. 629–643, 2001. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0893608001000545

[36] W. Maass, "Networks of spiking neurons: The third generation of neural network models," *Neural Networks*, vol. 10, no. 9, pp. 1659–1671, 1997. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0893608097000117

[37] G. Indiveri, "Synaptic plasticity and spike-based computation in vlsi networks of integrate-and-fire neurons," *Neural Inf. Process. Lett. Rev*, vol. 11, pp. 135–146, 2007.

[38] Ş. Mihalaş and E. Niebur, "A generalized linear integrate-and-fire neural model produces diverse spiking behaviors," *Neural computation*, vol. 21, no. 3, pp. 704–718, 2009.

[39] S. A. Aamir, P. Müller, A. Hartel, J. Schemmel, and K. Meier, "A highly tunable 65-nm cmos lif neuron for a large scale neuromorphic system," in *ESSCIRC Conference 2016: 42nd European Solid-State Circuits Conference*. IEEE, 2016, pp. 71–74.

[40] J. Bragg, E. Brown, P. Hasler, and S. DeWeerth, "A silicon model of an adapting motoneuron," in *2002 IEEE International Symposium on Circuits and Systems. Proceedings (Cat. No.02CH37353)*, vol. 4, 2002, pp. IV–IV.

[41] C. Yajie, S. Hall, M. Liam, O. Buiu, and K. Peter, "Analog spiking neuron with charge-coupled synapses," *Lecture Notes in Engineering and Computer Science*, vol. 2165, 07 2007.

[42] F. Folowosele, A. Harrison, A. Cassidy, A. G. Andreou, R. Etienne-Cummings, S. Mihalas, E. Niebur, and T. J. Hamilton, "A switched capacitor implementation of the generalized linear integrate-and-fire neuron," in *2009 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2009, pp. 2149–2152.

[43] L. Ming-Ze, P.-W. Po, T. Kea-Tiong, and F. Wai-Chi, "Multi-input silicon neuron with weighting adaptation," in *2009 IEEE/NIH Life Science Systems and Applications Workshop*, 2009, pp. 194–197.

[44] V. Kornijcuk, H. Lim, J. Y. Seok, G. Kim, S. K. Kim, I. Kim, B. J. Choi, and D. S. Jeong, "Leaky integrate-and-fire neuron circuit based on floating-gate integrator," *Frontiers in Neuroscience*, vol. 10, 2016. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fnins.2016.00212

[45] W. Jayawan and D. Piotr, "Compact silicon neuron circuit with spiking and bursting behaviour," *Neural Networks*, vol. 21, no. 2, pp. 524–534, 2008, advances in Neural Networks Research: IJCNN '07. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0893608007002705

[46] T. Dalgaty, M. Payvand, B. De Salvo, J. Casas, G. Lama, E. Nowak, G. Indiveri, and E. Vianello, "Hybrid cmos-rram neurons with intrinsic plasticity," in *2019 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2019, pp. 1–5.

[47] P. Wijesinghe, A. Ankit, A. Sengupta, and K. Roy, "An all-memristor deep spiking neural computing system: A step toward realizing the low-power stochastic brain," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 5, pp. 345–358, 2018.

[48] B. Ahmet and H.-H. Sotoudeh, "The design of a new spiking neuron using dual work function silicon nanowire transistors," *Nanotechnology*, vol. 18, no. 9, p. 095201, jan 2007. [Online]. Available: https://dx.doi.org/10.1088/0957-4484/18/9/095201

[49] A. Bindal and H. S, "An integrate and fire spiking neuron using silicon nano-wire technology," *TechConnect Briefs*, vol. 1, pp. 173–176, 01 2007.

[50] K. K. C L Chen, "A spiking neuron circuit based on a carbon nanotube transistor," in *Nanotechnology*, 2012.

[51] G. Palma, M. Suri, D. Querlioz, E. Vianello, and B. De Salvo, "Stochastic neuron design using conductive bridge ram," in *2013 IEEE/ACM International Symposium on Nanoscale Architectures (NANOARCH)*, 2013, pp. 95–100.

TH-3339_186102005

[52] B. Rajendran, A. Sebastian, M. Schmuker, N. Srinivasa, and E. Eleftheriou, "Low-power neuromorphic hardware for signal processing applications: A review of architectural and system-level design approaches," *IEEE Signal Processing Magazine*, vol. 36, no. 6, pp. 97–110, Nov 2019.

[53] J.Schemmel, "A wafer-scale neuromorphic hardware system for large scale neural modeling," *In Proc. IEEE International Symposium on Circuits and Systems (ISCAS)*, 2010.

[54] T. C. Sudhof, "Calcium control of neurotransmitter release," *Cold Spring Harb Perspect Biol*, vol. 4, no. 1, 2012.

[55] R. C. M. Christian Lüscher, "Nmda receptor-dependent long-term potentiation and long-term depression (ltp/ltd)," *Cold Spring Harb Perspect Biol*, vol. 4, no. 6, 2012.

[56] S. H, "Half a century of hebb." *Nat Neurosci*, vol. 1, no. 11, 2000.

[57] M. Rahimi Azghadi, N. Iannella, S. F. Al-Sarawi, G. Indiveri, and D. Abbott, "Spike-based synaptic plasticity in silicon: Design, implementation, application, and challenges," *Proceedings of the IEEE*, vol. 102, no. 5, pp. 717–737, 2014.

[58] H. Markram, W. Gerstner, and P. J. Sjöström, "Spike-timing-dependent plasticity: A comprehensive overview," *Frontiers in Synaptic Neuroscience*, vol. 4, 2012. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fnsyn.2012.00002

[59] G. Indiveri, "A low-power adaptive integrate-and-fire neuron circuit," in *Proceedings of the 2003 International Symposium on Circuits and Systems, 2003. ISCAS '03.*, vol. 4, 2003, pp. IV–IV.

[60] G. Indiveri, E. Chicca, and R. Douglas, "A vlsi array of low-power spiking neurons and bistable synapses with spike-timing dependent plasticity," *IEEE Transactions on Neural Networks*, vol. 17, no. 1, pp. 211–221, 2006.

[61] J. M. Cruz-Albrecht, M. W. Yung, and N. Srinivasa, "Energy-efficient neuron, synapse and stdp integrated circuits," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 6, no. 3, pp. 246–256, 2012.

[62] J. V. Arthur and K. Boahen, "Learning in silicon: Timing is everything," in *Advances in Neural Information Processing Systems*, Y. Weiss, B. Schölkopf, and J. Platt, Eds., vol. 18. MIT Press, 2005.

[63] H. Tanaka, T. Morie, and K. Aihara, "A cmos circuit for stdp with a symmetric time window," *International Congress Series*, vol. 1301, pp. 152–155, 2007, brain-Inspired IT III. Invited and selected papers of the 3rd International Conference on Brain-Inspired Information Technology "BrainIT 2006" held in Hibikino, Kitakyushu, Japan between 27 and 29 September 2006.

[64] I. Sourikopoulos, S. Hedayat, C. Loyez, F. Danneville, V. Hoel, E. Mercier, and A. Cappy, "A 4-fj/spike artificial neuron in 65 nm cmos technology," *Frontiers in Neuroscience*, vol. 11, 2017. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fnins.2017.00123

[65] F. Masuoka, M. Asano, H. Iwahashi, T. Komuro, and S. Tanaka, "A new flash e2prom cell using triple polysilicon technology," in *1984 International Electron Devices Meeting*, 1984, pp. 464–467.

[66] F. Masuoka, M. Momodomi, Y. Iwata, and R. Shirota, "New ultra high density eprom and flash eeprom with nand structure cell," in *1987 International Electron Devices Meeting*, 1987, pp. 552–555.

[67] Z. F. A. Z. T. Z. . K. F. A, "Resistive random access memory (rram): an overview of materials, switching mechanism, performance, multilevel cell (mlc) storage, modeling, and applications," in *Nanoscale Res Lett*, vol. 15, no. 90, 2020.

[68] F. Masuoka, M. Asano, H. Iwahashi, T. Komuro, and S. Tanaka, "A 256k flash eeprom using triple polysilicon technology," in *1985 IEEE International Solid-State Circuits Conference. Digest of Technical Papers*, vol. XXVIII, 1985, pp. 168–169.

[69] J. Hwang, J. Seo, Y. Lee, S. Park, J. Leem, J. Kim, T. Hong, S. Jeong, K. Lee, H. Heo, H. Lee, P. Jang, K. Park, M. Lee, S. Baik, J. Kim, H. Kkang, M. Jang, J. Lee, G. Cho, J. Lee, B. Lee, H. Jang, S. Park, J. Kim, S. Lee, S. Aritome, S. Hong, and S. Park, "A middle-1x nm nand flash memory cell (m1x-nand) with highly manufacturable integration technologies," in *2011 International Electron Devices Meeting*, 2011, pp. 9.1.1–9.1.4.

TH-3339_186102005

[70] S. Lai, "Non-volatile memory technologies: The quest for ever lower cost," in *2008 IEEE International Electron Devices Meeting*, 2008, pp. 1–6.

[71] M. Hosomi, H. Yamagishi, T. Yamamoto, K. Bessho, Y. Higo, K. Yamane, H. Yamada, M. Shoji, H. Hachino, C. Fukumoto, H. Nagao, and H. Kano, "A novel nonvolatile memory with spin torque transfer magnetization switching: spin-ram," in *IEEE InternationalElectron Devices Meeting, 2005. IEDM Technical Digest.*, 2005, pp. 459–462.

[72] B. Govoreanu, G. Kar, Y.-Y. Chen, V. Paraschiv, S. Kubicek, A. Fantini, I. Radu, L. Goux, S. Clima, R. Degraeve, N. Jossart, O. Richard, T. Vandeweyer, K. Seo, P. Hendrickx, G. Pourtois, H. Bender, L. Altimime, D. Wouters, J. Kittl, and M. Jurczak, "10×10nm2 hf/hfox crossbar resistive ram with excellent performance, reliability and low-energy operation," in *2011 International Electron Devices Meeting*, 2011, pp. 31.6.1–31.6.4.

[73] B. Govoreanu, A. Redolfi, L. Zhang, C. Adelmann, M. Popovici, S. Clima, H. Hody, V. Paraschiv, I. Radu, A. Franquet, J.-C. Liu, J. Swerts, O. Richard, H. Bender, L. Altimime, and M. Jurczak, "Vacancy-modulated conductive oxide resistive ram (vmco-rram): An area-scalable switching current, self-compliant, highly nonlinear and wide on/off-window resistive switching cell," in *2013 IEEE International Electron Devices Meeting*, 2013, pp. 10.2.1–10.2.4.

[74] C.-Y. Lin, C.-Y. Wu, C.-Y. Wu, T.-C. Lee, F.-L. Yang, C. Hu, and T.-Y. Tseng, "Effect of top electrode material on resistive switching properties of film memory devices," *IEEE Electron Device Letters*, vol. 28, no. 5, pp. 366–368, 2007.

[75] L. Goux, Y.-Y. Chen, L. Pantisano, X.-P. Wang, G. Groeseneken, M. Jurczak, and D. J. Wouters, "On the gradual unipolar and bipolar resistive switching of memory systems," *Electrochemical and Solid-State Letters*, vol. 13, no. 6, p. G54, apr 2010. [Online]. Available: https://dx.doi.org/10.1149/1.3373529

[76] H.-S. P. Wong, H.-Y. Lee, S. Yu, Y.-S. Chen, Y. Wu, P.-S. Chen, B. Lee, F. T. Chen, and M.-J. Tsai, "Metal–oxide rram," *Proceedings of the IEEE*, vol. 100, no. 6, pp. 1951–1970, 2012.

[77] J. Bill and R. Legenstein, "A compound memristive synapse model for statistical learning through stdp in spiking neural networks," *Frontiers in Neuroscience*, vol. 8, 2014. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fnins.2014.00412

[78] I. Boybat and et al., "Neuromorphic computing with multi-memristive synapses." *Nat Commun*, vol. 9, 2018.

[79] M.-H. Lee, Y.-H. Lin, Y.-Y. Lin, F.-M. Lee, D.-Y. Lee, and K.-Y. Hsieh, "Studies on reram conduction mechanism and the varying-bias read scheme for mlc and wide temperature range tmo reram," in *2018 14th IEEE International Conference on Solid-State and Integrated Circuit Technology (ICSICT)*, 2018, pp. 1–3.

[80] H. Aziza, S. Hamdioui, M. Fieback, M. Taouil, and M. Moreau, "Density enhancement of rrams using a reset write termination for mlc operation," in *2021 Design, Automation and Test in Europe Conference and Exhibition (DATE)*, 2021, pp. 1877–1880.

[81] J. Reuben and D. Fey, "A time-based sensing scheme for multi-level cell (mlc) resistive ram," in *2019 IEEE Nordic Circuits and Systems Conference (NORCAS): NORCHIP and International Symposium of System-on-Chip (SoC)*, 2019, pp. 1–6.

[82] P. Amit and H. Hyunsang, "Multilevel cell storage and resistance variability in resistive random access memory," *Physical Sciences Reviews*, vol. 1, no. 6, p. 20160010, 2016. [Online]. Available: https://doi.org/10.1515/psr-2016-0010

[83] Z. Fang, H. Y. Yu, X. Li, N. Singh, G. Q. Lo, and D. L. Kwong, "Hfox/tiox/hfox/tiox multilayer-based forming-free rram devices with excellent uniformity," *IEEE Electron Device Letters*, vol. 32, no. 4, pp. 566–568, 2011.

[84] H. Aziza, S. Hamdioui, M. Fieback, M. Taouil, and M. Moreau, "Density enhancement of rrams using a reset write termination for mlc operation," in *2021 Design, Automation and Test in Europe Conference and Exhibition (DATE)*, 2021, pp. 1877–1880.

TH-3339_186102005

[85] F. Tan, Y. Wang, Y. Yang, L. Li, T. Wang, F. Zhang, X. Wang, J. Gao, and Y. Liu, "A reram-based computing-in-memory convolutional-macro with customized 2t2r bit-cell for aiot chip ip applications," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 67, no. 9, pp. 1534–1538, 2020.

[86] J. Kang, B. Gao, P. Huang, L. Liu, X. Liu, H. Yu, S. Yu, and H.-S. P. Wong, "Rram based synaptic devices for neuromorphic visual systems," in *2015 IEEE International Conference on Digital Signal Processing (DSP)*, 2015, pp. 1219–1222.

[87] D. Kwon, S. Lim, J.-H. Bae, S.-T. Lee, H. Kim, Y.-T. Seo, S. Oh, J. Kim, K. Yeom, B.-G. Park, and J.-H. Lee, "On-chip training spiking neural networks using approximated backpropagation with analog synaptic devices," *Frontiers in Neuroscience*, vol. 14, 2020. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fnins.2020.00423

[88] Y. Geng, B. Gao, Q. Zhang, W. Zhang, P. Yao, Y. Xi, Y. Lin, J. Chen, J. Tang, H. Wu, and H. Qian, "An on-chip layer-wise training method for rram based computing-in-memory chips," in *2021 Design, Automation and Test in Europe Conference and Exhibition (DATE)*, 2021, pp. 248–251.

[89] H. Jiang, S. Huang, X. Peng, and S. Yu, "Mint: Mixed-precision rram-based in-memory training architecture," in *2020 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2020, pp. 1–5.

[90] Y. Geng, B. Gao, Q. Zhang, W. Zhang, P. Yao, Y. Xi, Y. Lin, J. Chen, J. Tang, H. Wu, and H. Qian, "An on-chip layer-wise training method for rram based computing-in-memory chips," in *2021 Design, Automation and Test in Europe Conference and Exhibition (DATE)*, 2021, pp. 248–251.

[91] A. Gautam and T. Kohno, "An adaptive stdp learning rule for neuromorphic systems," *Frontiers in Neuroscience*, vol. 15, 2021. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fnins.2021.741116

[92] W. Choi, M. Kwak, S. Kim, and H. Hwang, "Neural network training acceleration with rram-based hybrid synapses," *Frontiers in Neuroscience*, vol. 15, 2021. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fnins.2021.690418

[93] T. Werner, E. Vianello, O. Bichler, D. Garbin, D. Cattaert, B. Yvert, B. De Salvo, and L. Perniola, "Spiking neural networks based on oxram synapses for real-time unsupervised spike sorting," *Frontiers in Neuroscience*, vol. 10, 2016. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fnins.2016.00474

[94] W. He, S. Yin, Y. Kim, X. Sun, J.-J. Kim, S. Yu, and J.-S. Seo, "2-bit-per-cell rram-based in-memory computing for area-/energy-efficient deep learning," *IEEE Solid-State Circuits Letters*, vol. 3, pp. 194–197, 2020.

[95] S. Ambrogio, S. Balatti, V. Milo, R. Carboni, Z. Wang, A. Calderoni, N. Ramaswamy, and D. Ielmini, "Novel rram-enabled 1t1r synapse capable of low-power stdp via burst-mode communication and real-time unsupervised machine learning," in *2016 IEEE Symposium on VLSI Technology*, 2016, pp. 1–2.

[96] M. Kumar, S. S. Bezugam, S. Khan, and M. Suri, "Fully unsupervised spike-rate-dependent plasticity learning with oxide- based memory devices," *IEEE Transactions on Electron Devices*, vol. 68, no. 7, pp. 3346–3352, 2021.

[97] J. Lin and J.-S. Yuan, "Analysis and simulation of capacitor-less reram-based stochastic neurons for the in-memory spiking neural network," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 12, no. 5, pp. 1004–1017, 2018.

[98] Z. Jiang, S. Yu, Y. Wu, J. H. Engel, X. Guan, and H.-S. P. Wong, "Verilog-a compact model for oxide-based resistive random access memory (rram)," in *2014 International Conference on Simulation of Semiconductor Processes and Devices (SISPAD)*, 2014, pp. 41–44.

[99] S. Kvatinsky, E. G. Friedman, A. Kolodny, and U. C. Weiser, "Team: Threshold adaptive memristor model," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 60, no. 1, pp. 211–221, 2013.

[100] C. de S. Dias and P. F. Butzen, "A novel spice model of memristive devices with threshold current based control," in *2018 31st Symposium on Integrated Circuits and Systems Design (SBCCI)*, 2018, pp. 1–6.

TH-3339_186102005

[101] S. Yu and H.-S. P. Wong, "Compact modeling of conducting-bridge random-access memory (cbram)," *IEEE Transactions on Electron Devices*, vol. 58, no. 5, pp. 1352–1360, 2011.

[102] M. Bocquet, D. Deleruyelle, H. Aziza, C. Muller, and J.-M. Portal, "Compact modeling solutions for oxram memories," in *2013 IEEE Faible Tension Faible Consommation*, 2013, pp. 1–4.

[103] H. Li, P. Huang, B. Gao, B. Chen, X. Liu, and J. Kang, "A spice model of resistive random access memory for large-scale memory array simulation," *IEEE Electron Device Letters*, vol. 35, no. 2, pp. 211–213, 2014.

[104] P. Huang, X. Y. Liu, W. H. Li, Y. X. Deng, B. Chen, Y. Lu, B. Gao, L. Zeng, K. L. Wei, G. Du, X. Zhang, and J. F. Kang, "A physical based analytic model of rram operation for circuit simulation," in *2012 International Electron Devices Meeting*, 2012, pp. 26.6.1–26.6.4.

[105] X. Weijie, Z. Yudi, L. Haitong, K. Jinfeng, L. Xiaoyan, and H. Peng, "Peking university resistive-switching random access memory (rram) verilog-a model," Jun 2019. [Online]. Available: https://nanohub.org/publications/284/2

[106] F. M. Puglisi, L. Pacchioni, N. Zagni, and P. Pavan, "Energy-efficient logic-in-memory i-bit full adder enabled by a physics-based rram compact model," in *2018 48th European Solid-State Device Research Conference (ESSDERC)*, 2018, pp. 50–53.

[107] A. Padovani, L. Larcher, F. M. Puglisi, and P. Pavan, "Multiscale modeling of defect-related phenomena in high-k based logic and memory devices," in *2017 IEEE 24th International Symposium on the Physical and Failure Analysis of Integrated Circuits (IPFA)*, 2017, pp. 1–6.

[108] P. Francesco, Maria, Z. Tommaso, and P. Pavan, "Unimore resistive random access memory (rram) verilog-a model," Jun 2019. [Online]. Available: https://nanohub.org/publications/289/1

[109] F. M. Puglisi, N. Zagni, L. Larcher, and P. Pavan, "Random telegraph noise in resistive random access memories: Compact modeling and advanced circuit design," *IEEE Transactions on Electron Devices*, vol. 65, no. 7, pp. 2964–2972, 2018.

[110] F. M. Puglisi°, N. Zagni, L. Larcher, and P. Pavan, "A new verilog-a compact model of random telegraph noise in oxide-based rram for advanced circuit design," in *2017 47th European Solid-State Device Research Conference (ESSDERC)*, 2017, pp. 204–207.

[111] L. Luo, H. Xiaofang, and Duan.Shukai, "Multiple memristor series-parallel connection with use in synaptic circuit design," *IET Circuits,Devices Systems*, pp. 1–6, 2017.

[112] S. Dutta and et.al, "Leaky integrate and fire neuron by charge-discharge dynamics in floating-body mosfet," in *Scientific Reports*, vol. 7, no. 2045-2322, 2017.

[113] M. Davies, A. Wild, G. Orchard, Y. Sandamirskaya, G. A. F. Guerra, P. Joshi, P. Plank, and S. R. Risbud, "Advancing neuromorphic computing with loihi: A survey of results and outlook," *Proceedings of the IEEE*, vol. 109, no. 5, pp. 911–934, 2021.

[114] K. Roy, S. Mukhopadhyay, and H. Mahmoodi-Meimand, "Leakage current mechanisms and leakage reduction techniques in deep-submicrometer cmos circuits," *Proceedings of the IEEE*, vol. 91, no. 2, pp. 305–327, 2003.

[115] S. Lashkare, S. Chouhan, T. Chavan, A. Bhat, P. Kumbhare, and U. Ganguly, "Pcmo rram for integrate-and-fire neuron in spiking neural networks," *IEEE Electron Device Letters*, vol. 39, no. 4, pp. 484–487, 2018.

[116] K.-B. Choi, S. Y. Woo, W.-M. Kang, S. Lee, C.-H. Kim, J.-H. Bae, S. Lim, and J.-H. Lee, "A split-gate positive feedback device with an integrate-and-fire capability for a high-density low-power neuron circuit," *Frontiers in Neuroscience*, vol. 12, 2018. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fnins.2018.00704

[117] S. Woo, J. Cho, D. Lim, Y.-S. Park, K. Cho, and S. Kim, "Implementation and characterization of an integrate-and-fire neuron circuit using a silicon nanowire feedback field-effect transistor," *IEEE Transactions on Electron Devices*, vol. 67, no. 7, pp. 2995–3000, 2020.

[118] T. Chavan, S. Dutta, N. R. Mohapatra, and U. Ganguly, "Band-to-band tunneling based ultra-energy-efficient silicon neuron," *IEEE Transactions on Electron Devices*, vol. 67, no. 6, pp. 2614–2620, 2020.

TH-3339_186102005

[119] N. Kamal and J. Singh, "A highly scalable junctionless fet leaky integrate-and-fire neuron for spiking neural networks," *IEEE Transactions on Electron Devices*, vol. 68, no. 4, pp. 1633–1638, 2021.

[120] M. Akbari, S. M. Hussein, T.-I. Chou, and K.-T. Tang, "A 0.3-v conductance-based silicon neuron in 0.18μm cmos process," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 68, no. 10, pp. 3209–3213, 2021.

[121] Z. Fang, H. Y. Yu, X. Li, N. Singh, G. Q. Lo, and D. L. Kwong, "Hfox/tiox/hfox/tiox multilayer-based forming-free rram devices with excellent uniformity," *IEEE Electron Device Letters*, vol. 32, no. 4, pp. 566–568, 2011.

[122] K. Beckmann, J. S. Holt, N. C. Cady, and J. Van Nostrand, "Comparison of random telegraph noise, endurance and reliability in amorphous and crystalline hafnia-based reram," in *2015 IEEE International Integrated Reliability Workshop (IIRW)*, 2015, pp. 107–110.

[123] S. Vecchi, P. Pavan, and F. M. Puglisi, "A unified framework to explain random telegraph noise complexity in mosfets and rrams," in *2023 IEEE International Reliability Physics Symposium (IRPS)*, 2023, pp. 1–6.

[124] L. Reganaz, D. Deleruyelle, Q. Rafhay, J. Minguet Lopez, N. Castellani, J. F. Nodin, A. Bricalli, G. Piccolboni, G. Molas, and F. Andrieu, "Investigation of resistance fluctuations in reram: physical origin, temporal dependence and impact on memory reliability," in *2023 IEEE International Reliability Physics Symposium (IRPS)*, 2023, pp. 1–6.

[125] S. Vecchi, P. Pavan, and F. M. Puglisi, "Defects motion as the key source of random telegraph noise instability in hafnium oxide," in *ESSDERC 2022 - IEEE 52nd European Solid-State Device Research Conference (ESSDERC)*, 2022, pp. 368–371.

[126] Y. Yilmaz and P. Mazumder, "A drift-tolerant read/write scheme for multilevel memristor memory," *IEEE Transactions on Nanotechnology*, vol. 16, no. 6, pp. 1016–1027, 2017.

[127] J. Reuben and D. Fey, "A time-based sensing scheme for multi-level cell (mlc) resistive ram," in *2019 IEEE Nordic Circuits and Systems Conference (NORCAS): NORCHIP and International Symposium of System-on-Chip (SoC)*, 2019, pp. 1–6.

[128] S. Ambrogio, S. Balatti, V. Milo, R. Carboni, Z. Wang, A. Calderoni, N. Ramaswamy, and D. Ielmini, "Novel rram-enabled 1t1r synapse capable of low-power stdp via burst-mode communication and real-time unsupervised machine learning," in *2016 IEEE Symposium on VLSI Technology*, 2016, pp. 1–2.

[129] Z. Wang, S. Ambrogio, S. Balatti, and D. Ielmini, "A 2-transistor/1-resistor artificial synapse capable of communication and stochastic learning in neuromorphic systems," *Frontiers in Neuroscience*, vol. 8, 2015. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fnins.2014.00438

[130] V. Milo, G. Pedretti, R. Carboni, A. Calderoni, N. Ramaswamy, S. Ambrogio, and D. Ielmini, "Demonstration of hybrid cmos/rram neural networks with spike time/rate-dependent plasticity," in *2016 IEEE International Electron Devices Meeting (IEDM)*, 2016, pp. 16.8.1–16.8.4.

[131] A. Levisse, M. Bocquet, M. Rios, M. Alayan, M. Moreau, E. Nowak, G. Molas, E. Vianello, D. Atienza, and J.-M. Portal, "Write termination circuits for rram: A holistic approach from technology to application considerations," *IEEE Access*, vol. 8, pp. 109 297–109 308, 2020.

[132] S. R. Lee, Y.-B. Kim, M. Chang, K. M. Kim, C. B. Lee, J. H. Hur, G.-S. Park, D. Lee, M.-J. Lee, C. J. Kim, U.-I. Chung, I.-K. Yoo, and K. Kim, "Multi-level switching of triple-layered taox rram with excellent reliability for storage class memory," in *2012 Symposium on VLSI Technology (VLSIT)*, 2012, pp. 71–72.

[133] Y. Goldberg, "A primer on neural network models for natural language processing," *J. Artif. Int. Res.*, vol. 57, no. 1, p. 345–420, sep 2016.

[134] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[135] E. G. Izhikevich EM, "Large-scale model of mammalian thalamocortical systems." *Proc Natl Acad Sci U S A.*, vol. 105, 2008.

TH-3339_186102005

[136] M. Kumar, S. S. Bezugam, S. Khan, and M. Suri, “Fully unsupervised spike-rate-dependent plasticity learning with oxide- based memory devices,” *IEEE Transactions on Electron Devices*, vol. 68, no. 7, pp. 3346–3352, 2021.

[137] S. Nandakumar and B. Rajendran, “Bio-mimetic synaptic plasticity and learning in a sub-500 mv cu/sio2/w memristor,” *Microelectronic Engineering*, vol. 226, p. 111290, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167931720300782

[138] S. Gupta, A. Vyas, and G. Trivedi, “Fpga implementation of simplified spiking neural network,” in *2020 27th IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, 2020, pp. 1–4.

[139] S. Kulkarni and B. Rajendran, “Scalable digital cmos architecture for spike based supervised learning,” in *Engineering Applications of Neural Networks*, L. Iliadis and C. Jayne, Eds. Cham: Springer International Publishing, 2015, pp. 149–158.

[140] M. Ambard, B. Guo, D. Martinez, and A. Bermak, “A spiking neural network for gas discrimination using a tin oxide sensor array,” in *4th IEEE International Symposium on Electronic Design, Test and Applications (delta 2008)*, 2008, pp. 394–397.

[141] I. T. R. M. A. G. M. et al, “Simplified spiking neural network architecture and stdp learning algorithm applied to image classification,” in *J Image Video Proc*, vol. 4, 2015.

[142] P. Diehl and M. Cook, “Unsupervised learning of digit recognition using spike-timing-dependent plasticity,” *Frontiers in Computational Neuroscience*, vol. 9, 2015. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fncom.2015.00099

[143] D. Querlioz, O. Bichler, and C. Gamrat, “Simulation of a memristor-based spiking neural network immune to device variations,” in *The 2011 International Joint Conference on Neural Networks*, 2011, pp. 1775–1781.

[144] Y. Guo, H. Wu, B. Gao, and H. Qian, “Unsupervised learning on resistive memory array based spiking neural networks,” *Frontiers in Neuroscience*, vol. 13, 2019. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fnins.2019.00812

[145] I. Boybat, G. M. Le, and N. S. R. et al, “Neuromorphic computing with multi-memristive synapses,” *Nat Commun*, vol. 9, 2514, 2018.

[146] A. Valentian, F. Rummens, E. Vianello, T. Mesquida, C. L.-M. de Boissac, O. Bichler, and C. Reita, “Fully integrated spiking neural network with analog neurons and rram synapses,” in *2019 IEEE International Electron Devices Meeting (IEDM)*, 2019, pp. 14.3.1–14.3.4.

TH-3339_186102005

TH-3339_186102005

List of Publications

Journal Publications

1. **Ashvinikumar Dongre**, Gaurav Trivedi, "*RRAM-Based Energy Efficient Scalable Integrate and Fire Neuron With Built-In Reset Circuit*", in IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 70, no. 3, pp. 909-913, March 2023, doi: 10.1109/TCSII.2022.3219203.

2. **Ashvinikumar Dongre**, Gaurav Trivedi, "*Variation Tolerant RRAM Based Synaptic Architecture for On-Chip Training*", in IEEE Transactions on Nanotechnology, vol. 22, pp. 436-444, 2023, doi: 10.1109/TNANO.2023.3298962.

3. **Ashvinikumar Dongre**, Bipul Boro, and Gaurav Trivedi, "*ADC-Less Reprogrammable RRAM Array Architecture for In-Memory Computing*", in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 31, no. 12, pp. 2053-2060, 2023. doi: 10.1109/TVLSI.2023.3319578.

4. **Ashvinikumar Dongre**, Gaurav Trivedi, "*Implementation of Spiking Neural Network with RRAM-based Neuron and Reprogrammable Synapse*", in IEEE Transactions on Very Large Scale Integration (VLSI) Systems (Under Review).

Conference Publications

1. **Ashvinikumar Dongre**,Gaurav Trivedi, "*Binary Synaptic Array for Inference and Training with Built-in RRAM Electroforming Circuit*", 2023 24th International Symposium on Quality Electronic Design (ISQED), San Francisco, CA, USA, 2023, pp. 1-6.

TH-3339_186102005

TH-3339_186102005