Addressing the RRAM Reliability and Radiation Soft-Errors in the Memory Systems

by

Amr Mohamed Samir Tosson Abdelwahed

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2018

© Amr Mohamed Samir Tosson Abdelwahed 2018

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Examiner	NAME: Mohamad Sawan Title: Professor at cole Polytechnique, University of Montreal
Supervisor(s)	NAME: Catherine Gebotys Title: Professor at ECE
	NAME: Lan Wei Title: Assistant Professor at ECE
	NAME: Mohab Anis Title: Professor at ECE
Internal Member	NAME: Peter Levine Title: Assistant Professor at ECE
Internal-external Member	NAME: Patricia Nieva Title: Professor at MME
Other Member(s)	NAME: Vincent Gaudet Title: Professor at ECE

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

With the continuous and aggressive technology scaling, the design of memory systems becomes very challenging. The desire to have high-capacity, reliable, and energy efficient memory arrays is rising rapidly. However, from the technology side, the increasing leakage power and the restrictions resulting from the manufacturing limitations complicate the design of memory systems. In addition to this, with the new machine learning applications, which require tremendous amount of mathematical operations to be completed in a timely manner, the interest in neuromorphic systems has increased in recent years. Emerging Non-Volatile Memory (NVM) devices have been suggested to be incorporated in the design of memory arrays due to their small size and their ability to reduce leakage power since they can retain their data even in the absence of power supply.

Compared to other novel NVM devices, the Resistive Random Access Memory (RRAM) device has many advantages including its low-programming requirements, the large ratio between its high and low resistive states, and its compatibility with the Complementary Metal Oxide Semiconductor (CMOS) fabrication process. RRAM device suffers from other disadvantages including the instability in its switching dynamics and its sensitivity to process variations. Yet, one of the popular issues hindering the deployment of RRAM arrays in products are the RRAM reliability and radiation soft-errors. The RRAM reliability soft-errors result from the diffusion of oxygen vacations out of the conductive channels within the oxide material of the device. On the other hand, the radiation soft-errors are caused by the highly energetic cosmic rays incident on the junction of the MOS device used as a selector for the RRAM cell. Both of those soft-errors cause the unintentional change of the resistive state of the RRAM device. While there is research work in literature to address some of the RRAM disadvantages such as the switching dynamic instability, there is no dedicated work discussing the impact of RRAM soft-errors on the various designs to which the RRAM device is integrated and how the soft-errors can be automatically detected and fixed.

In this thesis, we bring the attention to the need of considering the RRAM soft-errors to avoid the degradation in design performance. In addition to this, using previously reported SPICE models, which were experimentally verified, and widely adapted system level simulators and test benches, various solutions are provided to automatically detect and fix the degradation in design performance due to the RRAM soft-errors. The main focus in this work is to propose methodologies which solve or improve the robustness of memory systems to the RRAM soft-errors. These memories are expected to be incorporated in the current and futuristic platforms running the advanced machine learning applications. In more details, the main contributions of this thesis can be summarized as:

- Provide in depth analysis of the impact of RRAM soft-errors on the performance of RRAM-based designs.
- Provide a new SRAM cell which uses the RRAM device to reduce the SRAM leakage power with minimal impact on its read and write operations. This new SRAM cell can be incorporated in the Graphical Processing Unit (GPU) design used currently in the implementation of the machine learning platforms.
- Provide a circuit and system solutions to resolve the reliability and radiation soft-errors in the RRAM arrays. These solution can automatically detect and fix the soft-errors with minimum impact on the delay and energy consumption of the memory array.
- A framework is developed to estimate the effect of RRAM soft-errors on the performance of RRAM-based neuromorphic systems. This actually provides, for the first time, a very generic methodology through which the device level RRAM soft-errors are mapped to the overall performance of the neuromorphic systems. Our analysis show that the accuracy of the RRAM-based neuromorphic system can degrade by more than 48% due to RRAM soft-errors.
- Two algorithms are provided to automatically detect and restore the degradation in RRAM-based neuromorphic systems due to RRAM soft-errors. The system and circuit level techniques to implement these algorithms are also explained in this work.

In conclusion, this work offers initial steps for enabling the usage of RRAM devices in products by tackling one of its most known challenges: RRAM reliability and radiation soft-errors. Despite using experimentally verified SPICE models and widely popular system simulators and test benches, the provided solutions in this thesis need to be verified in the future work through fabrication to study the impact of other RRAM technology shortcomings including: a) the instability in its switching dynamics due to the stochastic nature of oxygen vacancies movement, and b) its sensitivity to process variations.

Acknowledgements

First and foremost, all praise is due to Allah, the almighty, for giving me the blessings and strength to complete this work.

I would like to express my sincere gratitude to Prof. Lan Wei, Prof. Mohab Anis, and Prof. Catherine Gebotys who helped me, encouraged me, believed in me, and supported me through the various situations that I have faced throughout the tough PhD journey. I have learned a lot from their vast experience and technical knowledge.

I would also like to thank Prof. Shimeng Yu from Arizona State University whom, without his technical knowledge, guidance, and support, we would not have been able to complete and publish the various ideas discussed in thesis.

I owe a large debt of gratitude to my professors from whom I have learned invaluable lessons about research. Special thanks go to Prof. Hossam A. Fahmy, from Faculty of Engineering, Cairo University, my undergraduate and master supervisor who has always been my role model and who always encouraged me to move forward with research.

I would like to also express my deepest gratitude to all my childhood friends in Egypt and my friends and colleagues at the University of Waterloo, Mentor Graphics, GlobalFoundries for their understanding, continuous help, and technical and moral support.

Writing this thesis would not have been possible without the encouragement, dedication, and love of my family. My deepest gratitude goes to my mother, father, and sister for their never-ending support, for sharing their academic knowledge and experience with me, and for their endless love no matter what happens in my life. I really can not find enough words to express my gratitude for everything they have done for me. I thank Allah everyday for blessing me by having you in my life.

Dedication

This is dedicated to my family, friends, and all the people who helped me to get this work done.

Table of Contents

List of Tables xii

List of Figures xiii

List of Abbreviations xxv

1 Introduction 1

2 Review of Literature 5

2.1 RRAM Device Theory and Characteristics 5

2.2 RRAM Device Operations: Electroform, SET, and RESET Processes . . . 8

2.3 Physics of Resistive Switching 11

2.4 Summary of the RRAM Features 14

2.5 Comparison to Other NVM Devices 16

2.5.1 Phase-Change Random Access Memory (PCRAM) 17

2.5.2 Magnetic Random Access Memory (MRAM) 18

2.6 Applications Using RRAM Devices 21

2.6.1 Use in Crossbar Random Access Memory Arrays 21

2.6.2 Use in Low-power SRAM Designs 25

2.6.3 Use in Low-power Sequential Circuits 26

2.6.4 Use in Neuromorphic Systems 27

2.7 RRAM Soft-Errors 28
2.7.1 Reliability Soft-Errors 29
2.7.2 Radiation Soft-Errors 30
2.8 RRAM SPICE Models 32
2.9 Organization of the Research Work in this Thesis 35

3 8T1R: Optimizing the Design of the RRAM-based Non-Volatile SRAM Design to Reduce the Effect of RRAM Soft-Errors 37
3.1 Introduction 38
3.2 Proposed 8T1R cell 43
3.2.1 Read/Write Operation 44
3.2.2 Store Operation 44
3.2.3 Restore Operation 48
3.3 Simulation Results 53
3.3.1 Read and Write Operations 54
3.3.2 Store and Restore Operations 59
3.4 Summary 60

4 Resolving the RRAM Reliability Soft-Errors in 1T1R RRAM Memory Arrays 62
4.1 Introduction 63
4.2 The Concept of the Refresh Methodology 63
4.3 Refresh Circuit Schematic and Operation 66
4.3.1 Sense Amplifier of the Refresh circuit 68
4.3.2 Error Detection Unit 70
4.4 Simulation Results 73
4.4.1 SPICE Level Simulation Results 75
4.4.2 System Level Simulation Results 76
4.5 Summary 79

5 Resolving Single-Event Upsets in 1T1R RRAM Memory Arrays 80

5.1 Introduction 81

5.2 Proposed Methodology for Detecting and Fixing Single-Event Upset 81

5.2.1 Impact of Heavy-ions Strikes in Cases 1 and 2 84

5.2.2 Impact of Heavy-ions Strikes in Cases 3 and 4 85

5.2.3 Impact of the Proposed Methodology on the Write Operation . . . 89

5.2.4 Impact of the Proposed Methodology on the Read Operation 90

5.3 Required Modifications to the Read Circuitry 91

5.3.1 UDU Circuit 94

5.3.2 WGU Circuit 95

5.4 Simulation Results 98

5.4.1 Selection of the High Voltage of the WE Signal (VDD_{WE}) 100

5.4.2 Simulation Results for the Write Operation 102

5.4.3 Simulation Results for the Read Operation 103

5.4.4 System Level Simulation Results 105

5.5 Summary 107

6 Addressing the RRAM Reliability Soft-Errors in Neuromorphic Systems 109

6.1 Introduction 110

6.2 Modeling the RRAM Reliability Soft-Errors on the System Level 116

6.2.1 Computing A_{exp} 118

6.2.2 Computing A_{act} 120

6.3 Analysis of the Neuron Pulses 123

6.3.1 Effect of Changing the Pulse Frequency 124

6.3.2 Effect of Changing the Pulse Width 126

6.3.3 Effect of Changing the Pulse Amplitude 127

6.3.4 Combining the Effect of Changing the Pulse Amplitude and Width 128

6.4 Proposed Framework to Detect and Fix the RRAM Reliability Soft-Errors 128

6.4.1 Detection Step 130
6.4.2 Restore Step 130
6.4.3 Impact of the Proposed Framework on the System Performance . . 132
6.5 Improving the Proposed Framework to Detect and Fix the RRAM Reliability Soft-Errors 135
6.6 Comparative Analysis Between the Two Proposed Frameworks 138
6.7 Modifications of the Read and Write Circuits 139
6.7.1 Write Circuit Modifications 139
6.7.2 Read Circuit Modifications 140
6.8 Simulation Results for the Modified Read Circuit 145
6.9 Summary 146

7 Conclusion and Future Work **148**

References **151**

APPENDICES **167**

A Brian Code for the RRAM-based Neuromorphic System **168**

B List of Publications **173**

List of Tables

2.1 RRAM devices comparison . . . 15
2.2 NVM cells comparison . . . 20
3.1 Transistor sizes of the 8T1R NV-SRAM cell . . . 54
3.2 Comparison results for the read and write operations of the different RRAM-based NV-SRAM 128x128 arrays . . . 55
3.3 Comparison of store and restore operations of various 128x128 RRAM-based NV-SRAM arrays . . . 60
4.1 Sensing circuit performance for a 128x128 1T1R macro . . . 75
4.2 Read operation in 8 Gbit 1T1R RRAM array . . . 76
5.1 Summary of the different scenarios for the heavy-ions strikes in the 1T2R cell 90
5.2 Summary of the voltage across the senseRRAM and cellRRAM for the different bias scenarios of the half-selected cells . . . 90
5.3 WE signal different bias voltage for the various write operations initiated on the fully-selected cells . . . 91
5.4 Comparison of the write performance for 1T1R and 1T2R 128x128 arrays . 102
6.1 SPICE simulation results for the maximum percentages of change in the RRAM resistive state . . . 120
6.2 BRIAN simulation settings and results . . . 121
6.3 Comparison between the basic and modified frameworks . . . 138

List of Figures

2.1 Basic I-V characteristics of RRAM. This graph is generated from running SPICE simulations using the model described in [41] for the HfO_x RRAM device. The behavior matches the hysteresis loop shape predicted for the conceptual memristor. 7

2.2 Structure of the first TiO_2 RRAM device. The parameter “D” in the figure describes the thickness of oxide layer while the parameter “x” defines the gap distance separating the top electrode from TiO_{2-y} layer which is the part of oxide material containing the most of oxygen vacancies (i.e., current carriers). The “y” index is used to indicate the existence of oxygen vacancies in the oxide material. 8

2.3 Ion hopping illustration. The figure shows how the oxygen atoms (black dots) hop randomly at each time instance to fill the oxygen vacancies in the crystal structure (white dots). 9

2.4 Simple illustration of resistive switching in TiO_x RRAM devices. The parameters “D” and “x” define the thickness of oxide material and the gap distance separating the top electrode from the TiO_{2-y} layer. When a negative voltage is applied on the top electrode (Case I), the oxygen vacancies drift towards the top electrode under the effect of high electric field E_{drift} causing the gap distance “x” to decrease. Oppositely, if a positive voltage is applied (Case II), the direction of E_{drift} changes and the oxygen vacancies are pushed away from the top electrode increasing the gap distance “x”. . . 10

2.5 Effect of stochastic oxygen vacancies movement on generating different paths for the SET and RESET process with each switching cycle [32]. Permission granted to use the figure. 11

2.6 Switching mechanisms in oxide-based RRAM devices described [47] which change depending on the RRAM oxide and electrode materials. Permission granted to use the figure. 12

2.7 PCRAM cell structure in which the heater layer is used to cause enough heat within the chalcogenide material to change from amorphous to crystalline structure and vice versa. 17

2.8 MRAM cell structure. The pinned layer is the thick magnetic layer which has only one direction of magnetization shown by the arrow in figure. The free layer is the other magnetic layer whose direction of magnetization can be programmed. 19

2.9 Crossbar RRAM structure where the RRAM device is used to connect between the row and column data lines. 21

2.10 Illustration of the sneak-path issue [86]. The RRAM at row 4 and column 1 is the memory cell that is intended to be read, while the other RRAM devices on the dotted line are the LRS RRAM memory cells on the sneak-path. Permission granted to use the figure. 22

2.11 1T1R cell proposed in [87] which requires the existence of both negative and positive high potentials. Permission granted to use the figure. 23

2.12 The alternative 1T1R cell proposed in [88]. To overcome the need to use both negative and positive high potential voltages, the signals "BL" and its inverted version "BLB" are applied on the terminals of RRAM device. Permission granted to use the figure. 24

2.13 The basic read/write voltage configuration for RRAM arrays proposed in [91]. The unselected control lines are connected to V/2 where V represents the voltage level required to be applied on RRAM device to trigger either SET/RESET process. Only for the RRAM cell that is meant to be programmed, the voltage drop across its terminals will be V. Permission granted to use the figure. 25

2.14 Architecture of a power-gated sequential circuit. Before enabling the "sleep" mode of operation, the sequential circuit data is pushed first on the retention latches which use high-Vth MOS transistors to reduce the subthreshold leakage power. 26

2.15 RRAM usage in neuromorphic circuits [98]. Left part shows how the RRAM device can mimic the functionality of synapses in the biological neural system. Right part of the figure illustrates how the biological neural system is implemented in circuits by having pre- and post-neurons communicating pulses depending on the status of RRAM devices connecting them. Permission granted to use the figure. 28

2.16 Oxygen vacancies (V_{ox}) diffusion out of the conduction filaments containment [36]. The parameters "width" and "length" define the dimensions of conductive filaments which are affected by the reliability soft-errors. Permission granted to use the figure. 29

2.17 An example of SEE scenario in a half-selected 1T1R cell sharing the same BL bias voltage as fully-selected cell undergoing a SET operation. The current source in the figure models the SEE effect caused by the electron-hole pairs generated by the heavy-ions strikes [41]. 31

2.18 I-V characteristics curve for the RRAM HfO_x SPICE model described in [33]. I-V characteristics curve generated from the model fits that resulting from the experimental data. The figure also shows how the LRS and HRS of the device changes with the various programming conditions expressed by the different compliance current levels. Permission granted to use the figure. 33

2.19 Modeling of RRAM reliability soft-errors for HfO_x RRAM device. Curves generated from the SPICE model in [33] fits the experimental data obtained by various research groups for different operating temperatures. Permission granted to use the figure. 34

2.20 Simulation and experimental results for the current generated due to the highly energetic incident charged particles on the junction of the access transistor of 1T1R RRAM array [41]. The figure shows the matching between the computed current levels and those obtained from the experimental data for different energy levels for the incident charged particles. Permission granted to use the figure. 35

3.1 The 6T2R SRAM cell [17] in which the two RRAM devices are connected together through the extra control line 'CL'. Permission granted to use the figure. 39

3.2 The 8T2R SRAM cell proposed in [93]. Two control signals, 'SW' and 'CL', are used to program the resistive state of RRAM devices. Permission granted to use the figure. 40

3.3 The Rnv8T SRAM cell proposed in [92]. Other than being used in the store/restore operations, the transistors "RSWL" and "RSWR" are used during the write operation to enhance the noise margin. Permission granted to use the figure. 41

3.4 The 7T SRAM cell [94]. The transistor "RPG" is the NMOS device used only during the read operation and it is connected to dedicated control lines for read operation 'RBL' and 'RWL'. The same structure is used for the cell 9T2R in [94]. Permission granted to use the figure. 43

3.5 The structure of 8T1R NV-SRAM cell. 44

3.6 The store operation waveform for the case when QB is at logic '1' and Q is at logic '0'. The numbers in circles correspond to the sequence of store operation steps. In our experiments, VDD = 1.1 V and VDDH = 2.0 V. . 45

3.7 Store operation waveforms generated from running SPICE for the case when node 'QB' and 'Q' are at logic '1' and '0', respectively. RRAM state is programmed to LRS in this scenario as indicated by the decrease in "GAP" value which describes the distance separating the top electrode from the tip of conductive filaments. 46

3.8 Store operation waveforms generated from running SPICE for the case when node 'QB' and 'Q' are storing logic '0' and '1', respectively. RRAM state is programmed to HRS in this case as indicated by the increase in "GAP" value causing less current to pass between the RRAM device terminals. . . 47

3.9 Illustration of restore operation waveforms for the case when RRAM is at LRS. In this scenario, node 'Q' is precharged to a voltage less than Vth of NMOS device causing the node 'QB' to charge to high voltage when power signal is reactivated. At the end of restore operation, the voltage of nodes 'QB' and 'Q' is set to VDD and ground, respectively. 49

3.10 The restore power supply circuit. The delay chain is added to guarantee that the power supply of "INV1" is activated before that of "INV2". 50

3.11 Restore operation waveforms generated from SPICE simulation for the case when RRAM is at LRS. In this scenario, at the end of restore operation, voltage of node 'Q' is at ground while that of node 'QB' is at VDD. 51

3.12 Restore operation waveforms generated from running SPICE for the case when RRAM device is at HRS. In this case, at the end of restore operation, the voltage of node 'Q' and 'QB' are set to VDD and ground, respectively. 52

3.13 Block diagram for the simulation runs conducted to evaluate the performance of various modes of operations (i.e., read, write, store, and restore operations) for the 8T1R NV-SRAM cell. 53

3.14 Modeling of the noise sources to compute the read/write noise margins of back to back inverters used in CMOS SRAM and DFF designs [115, 116]. All possible combinations of polarities for the noise sources in the figure have to be tried and the resulting minimum value of noise signal voltage is considered as the noise margin of SRAM cell. The same concept is applied to the NV-SRAM cells studied in this work. 56

3.15 Waveforms for computing the noise margin of 8T1R cell. "V_noise" in figure describes the noise signal voltage level which results in the write failure. In this case, with a noise signal of 337 mV level, the write operation fails. . . 57

3.16 Layout of (a) Rnv8T cell and (b) 9T2R cell. The names of the transistors are aligned with those in fig. 3.5. **MRead** is the transistor connected to the dedicated read port as in [94]. 58

4.1 Division of the RRAM resistance range. "Region II" defines the resistance range where the refresh operation is triggered. In "region I" and "region IV", the RRAM state is considered at LRS and HRS, respectively. Accordingly, no refresh operation is required. When the resistance of RRAM is in "region III", the device is considered suffering from hard-errors since its resistance could not be refreshed earlier to LRS. 64

4.2 Rth_1 threshold value selection. The optimum value of Rth_1 is chosen as a compromise between increasing the difference between V(Rth_1) and V(Rth_2) and reducing the amount of refresh cycles required. 66

4.3 Block diagram of refresh circuit. Compared to the normal read circuitry, "SA2" and "error detection unit" blocks are added to sense the four regions of RRAM resistance range illustrated in fig. 4.1. 67

4.4 "SA2" circuit schematic. Compared to the structure of latch-type voltage SA in [123], the transistors "M13", "M14", and "M15" are added to support the "two-cycle" comparisons required to define whether the RRAM resistance is in region I, II, or III. 68

4.5 Illustration of waveforms of "SA2" circuit operation for the case when $V(Rth_1) < V_{sense} < V(Rth_2)$. In this scenario, the output from "first cycle" of comparison indicates that $V_{sense} > V(Rth_1)$ by having the signal 'OUTP' set to VDD. Accordingly, the signals 'OUTN' and 'OUTP' are discharged to ground by disabling the signal 'SE' (and enabling 'SEN'). After the "second cycle" of comparison, the signal 'OUTN' is raised to VDD indicating that $V_{sense} < V(Rth_2)$. 70

4.6 SPICE waveforms of "SA2" circuit operation for the case when $V(Rth_1) < V_{sense} < V(Rth_2)$. The number sequences in this figure are consistent with those in fig. 4.5. 71

4.7 Circuit schematic of the "error detection unit": a) SE2 generation circuit which is responsible of enabling the 'SE2' signal when $V(Rth_1) < V_{sense} < V(Rth_3)$, b) SE/SE1 generation circuit which is responsible of enabling the signals 'SE' and 'SE1' used by "SA1" and "SA2", and c) refresh detection unit which compares the output from "SA1" and "SA2" to decide whether the selected 1T1R cell is suffering from either soft-error or hard-error. . . . 72

4.8 Block diagram for the simulation runs and the tools used to evaluate the proposed refresh methodology. The ASU model described in [33] is used for the SPICE simulation runs while the CACTI C++ [42] is used to estimate the impact of read circuit modifications on large memory arrays. 74

4.9 Refresh circuit impact on read energy and delay for different array sizes. The change percentages in figure are computed by comparing the read circuit delay and energy consumption with the case when the refresh circuit is not integrated. Since the delay and energy consumption of read operation is mainly governed by other memory components, the higher the capacity of RRAM arrays, the lower is the impact of modified read circuit on the increase of energy and delay of read operation. 77

4.10 Refresh circuit effect on increasing the immunity of the RRAM 1T1R arrays to reliability soft-errors. Since the refresh circuit can detect and fix the reliability soft-errors generated from any source, referring to the experimental data in [38], the proposed methodology increases the resilience of RRAM 1T1R arrays to those soft-errors by 80%. 78

5.1 Schematic of the 1T2R cell. Compared to the normal 1T1R cell design, two main updates are added: a) extra RRAM device (i.e., senseRRAM) whose state indicates whether radiation soft-errors have occurred or not, and b) extra control line (i.e., WE) to correctly bias senseRRAM to track unintentional changes in cellRRAM state. 82

5.2 Layout of the proposed 1T2R cell. Using a 65 nm PDK, the RRAM device is integrated between metal levels 4 and 5 (i.e., M4 and M5 in the figure). Lower layers of metal (M1-M2) are omitted from figure to simplify the illustration. 83

5.3 The four possible scenarios for the half-selected cells bias during the write operation. a) Case 1, b) Case 2, c) Case 3, and d) Case 4. In cases 1 and 2, the half-selected cells share the same BL voltage as the fully-selected cells, while the other control signals are connected to ground. In cases 3 and 4, the half-selected cells share the same voltage of WL and SL as the fully-selected cells, while the BL is connected to VDD/2 to prohibit modifying the RRAM state of half-selected cells. 84

5.4 Biasing potentials on the half-selected cells in cases 1 and 2. Since the SL is always connected to ground in those cases, the maximum voltage drop across the RRAM terminals is not high enough to cause any change in its resistive state. 85

5.5 Waveforms of the control signals for the half-selected cells in case 3. The numbers in the figure represent the sequence of operations performed. A read operation is required before initiating the RESET process on the fully-selected cells to properly bias WE signal. In this figure, since the cellRRAM device of the half-selected cells is at LRS, WE is connected to high voltage (i.e., 1.5V). 86

5.6 Waveforms of the control signals for the half-selected cells in case 4. The numbers in the figure represent the sequence of operations performed. Since the cellRRAM device of the half-selected cells is already at HRS, even if high energetic heavy-ions are incident on the cell, cellRRAM state remains at HRS. 87

5.7 SPICE waveforms of the control signals for the half-selected cells in case 3. The SPICE simulation results demonstrate that, in case 3, if heavy-ions strikes occur, the senseRRAM and cellRRAM states will unintentionally change from HRS to LRS and from LRS to HRS, respectively. The SPICE waveforms related to setting the right voltage of WE signal is illustrated in fig. 5.14 in section 5.3.2. 88

5.8 SPICE waveforms of the control signals for the half-selected cells in case 4. The SPICE simulation results demonstrate that, in case 4, the heavy-ions strike will not cause changes to the HRS of senseRRAM and cellRRAM. . 89

5.9 Waveforms for the modified read operation. The read process consists of two regions of operation: "Upset detection" and "normal read" regions. In this case, since the senseRRAM is at HRS, this indicates that no SEU has occurred and hence, the device proceeds to the "normal read" region. . . . 92

5.10 SPICE waveforms for the modified read operation in case if the senseRRAM and cellRRAM are at HRS and LRS, respectively. Since the senseRRAM is at HRS, the read process proceeds to the "Normal read region" and hence, the BL voltage discharges due to the LRS of cellRRAM. 93

5.11 Architecture of the modified read circuit. Compared the normal read circuitry, the UDU and WGU units are added. RU is the normal read circuit, discussed in chapter 4, used to read the cellRRAM state. UDU is used to read the state of senseRRAM and trigger RU if senseRRAM is at HRS. WGU is responsible of setting up the right bias for WE signal. 94

5.12 Schematic of the UDU circuit [123] which is basically a normal latch-type voltage SA. 95

5.13 Schematic of the WGU circuit. It consists mainly of two parts: "cellRRAM detection unit" which, if cellRRAM is at LRS, the output from this unit is connected to ground, and "control part" which, based on the input from "cellRRAM detection unit" and BL voltage on the fully-selected cells, its output WE signal is either connected to ground or high voltage. 96

5.14 SPICE waveforms describing the WGU operation. Since the voltage of BL is at low voltage to enable the RESET process and the read state of cellRRAM indicates that it is at LRS, the WE voltage is raised to 1.5 V. 97

5.15 Block diagram for the simulation runs and the tools used to evaluate the performance of the 1T2R RRAM arrays. The ASU model [33] together with the methodology described in [41] are used to run SPICE simulations, while CACTI C++ files [42] are used for system level simulations. 98

5.16 Effect of VDD_{WE} voltage on the correctness of our proposed methodology in detecting the upset events. Proper voltage of WE is chosen such that the impact on RESET operation is minimized, and at the same time, the senseRRAM state can track the changes in cellRRAM state. 100

5.17 Effect of VDD_{WE} voltage on the performance of RESET operation. Increasing VDD_{WE} increases the delay and energy consumption of RESET process exponentially. 101

5.18 Impact of increasing the T_{UD} on detecting the change in senseRRAM state. "Read Energy" curve is calculated assuming worst case when the senseRRAM is at LRS. Increasing T_{UD} enhances the difference between the read voltage by SA of UDU when the senseRRAM is at HRS and when it is at LRS (i.e., V_{sense} in the figure). However, it also negatively affects the ability of SA of RU in correctly reading the state of cellRRAM. 104

5.19 System level simulation for the increase in the energy consumption of 1T2R memory arrays with different capacities. The "percentage of increase" axis in figure refers to the change in the energy consumption of read and write operations in comparison to those of 1T1R arrays with the same capacity. . 105

5.20 System level simulation for the impact of modified read circuit on the chip area of 1T2R memory arrays with different capacities. The "percentage of increase" in the figure refers to the change in chip area compared to the case when large 1T1R arrays are used instead. 107

6.1 General structure of a neuromorphic system where pre-neuorns are connected to post-neurons through a dense network made of synapses. The RRAM device is used in the synaptic network implementation due to its small size, low programming requirements, and its ability to be programmed to intermediate states depending on the pre- and post-neurons pulses. . . . 110

6.2 Structure of the neuromorphic system used in our studies [142] which classifies the handwritten digits defined in MNIST dataset [153]. 112

6.3 Illustration of the STDP rule and how the resistive state of the RRAM device changes accordingly [22]. 114

6.4 Simple implementation of WTA methodology. The first neuron generating pulses discharges the control line 'RESET_CTRL' which raises the 'RESET' control signal prohibiting any other neurons from generating pulses. 115

6.5 Modeling framework for computing the impact of RRAM reliability soft-errors on the system accuracy. "Phase I" and "Phase II" describe the SPICE and system level simulations run without taking into account the RRAM soft-errors. "Phase III" and "Phase IV" describe the SPICE and system level simulations run while the RRAM soft-errors are being considered. . . 116

6.6 Block diagram for the simulation runs and the tools used to compute the degradation in RRAM-based neuromorphic system. BRIAN is chosen as the system level simulator since the code for the system in [142] is already written using this python package. In our case-study system, the classification of the handwritten digits of MNIST dataset [44] is used to estimate the degradation in system performance due to RRAM soft-errors. 122

6.7 Properties of the action potential (neuron pulse) [159]. There are basically three main properties: pulse amplitude, pulse width, and pulse frequency which is defined through neuron threshold voltage, resting potential, and refractory time. 123

6.8 Impact of changing the various parameters of the neuron pulses on the actual system accuracy A_{act}. a) Changing the pulse frequency, b) Changing the pulse width, c) Effect of changing the pulse amplitude, and d) combining the changes in the pulse amplitude and frequency to restore the system accuracy with minimum impact on its energy consumption. 125

6.9 Decomposing the 784x400 network into 32 units (31 units consist of 25x400 neurons and the last unit contains 9x400 neurons). Using CACTI C++ files, the increase in power consumption due to the split of the neural network is about 11x. 127

6.10 Flowchart of the suggested methodology for detecting and fixing the system performance drop caused by RRAM reliability soft-errors. The proposed framework consists of two main phases: "Detection Phase" which monitors the number of generated pulses with each input pattern to detect if the RRAM array is suffering from reliability soft-errors , and "Restore Phase" which is responsible of fixing the degradation in system performance by first decreasing the resistance of all RRAM devices by ΔP_R and then re-apply the last N_{pat} patterns to reprogram the RRAM states correctly. 129

6.11 N_{pat} study. a) Impact of changing N_{pat} on restoring the system accuracy, b) the required $N_{pat,opt}$ patterns when the RRAM reliability soft-errors occur during the various tiers of the training sequence. 1st Tier, 2nd Tier, and 3rd Tier in fig. 6.11b refers to the cases when soft-errors occur in the first, second, and third 20,000 patterns of the MNIST training dataset, respectively. 132

6.12 Effect of incorporating the suggested detection and fix algorithm on the delay and energy consumption of the system [142]. When the number of $N_{pat,opt}$ grows, the training cycle duration increases and by consequence, the energy consumption of the system. The "percentage of increase" in the figure represents the increase in the energy and delay of the training cycle in comparison to the case when the suggested methodology is not integrated to the system. 133

6.13 Flowchart of the modified methodology for detecting and fixing the RRAM reliability soft-errors. The algorithm is very similar to the one described in fig. 6.10. The main difference is eliminating the need to re-apply patterns by: 1) increasing the amplitude of input pattern causing a decrease in the RRAM resistance, and 2) decreasing the amplitude of input patterns otherwise. This change is only applied for next N_{amp} patterns of the training cycle. 135

6.14 Effect of the restore step parameters (i.e., ΔP_{LRS} and N_{amp}) on recovering the degraded system performance: a) N_{amp} with fixed ΔP_{LRS}=5%, b) $N_{amp,opt}$ for various ΔP_{LRS} values. 137

6.15 Effect of the restore step parameters on the system energy consumption increase. With smaller ΔP_{LRS}, the value of $N_{amp,opt}$ increases as well as the energy consumption of the system. However, the total increase is not high due to small value of $N_{amp,opt}$, for which amplitude is increased, in comparison to the total number of training patterns. 138

6.16 Schematic of the modified read circuit. The "scan chain for pulse detection" and "detection unit" are added to the normal I&F circuit of the neuron. "Scan chain" unit is added to collect the pulses generated from the various neurons at different time instances and send them to "detection unit". The "detection unit" counts the pulses sent from the "scan chain" in response to the applied input pattern to determine whether the RRAM array is suffering from soft-errors or not. 140

6.17 "Scan chain" structure. The XOR gates are used to insert delays between the stages of the chain. 141

6.18 XOR gate output signals in case when the pulses observed at post-neurons occur simultaneously. Due to the delay of XOR gate, even if pulses from adjacent neurons are generated simultaneously, they are still going to be counted separately at "detection unit". 142

6.19 "Detection unit" structure. The unit consists basically of two counters: "3-bit counter", which checks whether more than 5 pulses have been generated for each input pattern, and "6-bit counter" which checks, in case if the same input pattern is re-applied (due to generating less than 5 pulses), the number of retrying the same pattern is higher than 50 to trigger the restore phase of the algorithms described in sections 6.4 and 6.5. 143

6.20 Waveforms for the operation of "detection unit" when the restore step of the framework is initiated. In this case, it is assumed that the input pattern was reapplied 49 times. 144

7.1 Contributions of the work presented in the thesis. 149

List of Abbreviations

1T1R one-Transistor-one-RRAM device 4, 22–24, 30, 32, 34, 36, 62, 63, 65–70, 73–76, 79–81, 83, 90, 98, 99, 101, 102, 104–108, 147

1T2R one-Transistor-two-RRAM device 80–83, 89, 90, 98–100, 102, 105, 106

BEOL Back-End Of Line 58

BL Bitline 31, 38, 41–44, 48, 51, 56, 59, 63, 66, 69, 75, 81, 84, 85, 87, 90, 91, 93, 94, 96, 103, 104, 126

BLB Bitline Bar 38, 41–44, 48, 56, 59

CBRAM Conductive Bridge Random Access Memory 8, 13

CMOS Complementary Metal Oxide Semiconductor 1, 2, 14, 20, 21, 28, 96, 103

CNN Convolutional Neural Network 139

CVDD Cell VDD 42, 45, 46, 48, 59

DAC Digital-to-Analog Converter 139, 140

DC Direct Current 7

DFF D Flip-Flop 27, 32

DL D Latch 95

DNM Dynamic Noise Margin 55

DNN Deep Neural Network 150

DRAM Dynamic Random Access Memory 22, 63, 81

DVS Dual Voltage Supply 1, 38

ECC Error Correction Codes 106

FDSOI Fully-Depleted Silicon On Insulator 106

FPGA Field Programmable Gate Array 63

GAA Gate-All-Around 48

GPU Graphics Processing Unit 3, 36, 37, 61, 148

HRS High Resistive State 7, 9, 10, 12–16, 19–22, 24, 28, 31, 42, 45, 47, 48, 52, 59, 60, 64, 65, 67, 69, 80, 81, 83, 87–91, 93, 95, 99, 101–104, 113, 118, 120, 121, 126, 128, 130, 131, 134

I&F Integrate and Fire 114, 140, 146

ICs Integrated Circuits 1, 30, 150

LET Linear Energy Transfer 85, 100–102

LRS Low Resistive State 7, 9, 12–16, 19–22, 24, 28, 31, 33, 40, 42, 45, 46, 48–51, 59, 62, 64, 65, 67, 69, 71, 78, 80, 81, 83, 87, 90–96, 99, 101–103, 113, 118, 120, 121, 130, 131, 134, 136

MEU Multiple Event Upsets 30, 32, 62, 80, 91, 92, 108

MLCs Multi-Level Cells 16

MOS Metal Oxide Semiconductor 16, 26, 42, 47, 68

MOSFET Metal Oxide Semiconductor Field Effect Transistor 2, 31, 81, 82

MRAM Magnetic Random Access Memmory 18, 19, 38

NMOS N-type Metal Oxide Semiconductor 41, 42, 48, 52, 82, 85, 96, 99, 102, 103, 114, 139

NV-SRAM Non-Volatile Static Random Access Memory 26, 37, 38, 42–45, 48, 54–60, 148

NVM Non-Volatile Memory 1, 2, 5, 16, 19, 28, 29, 38, 63, 76, 81, 111

PCM Phase Change Memory 38

PCRAM Phase Change Random Access Memory 17–20, 111

PDK Process Design Kit 48, 82, 83

PMOS P-type Metal Oxide Semiconductor 41, 42, 48, 49, 99, 114, 139

PTM Predictive Technology Model 74, 102

RDNM Read Dynamic Noise Margin 55–58

Rnv8T Resistive Nonvolatile 8T2R 43, 55–57, 59, 60

RRAM Resistive Random Access Memory 2–16, 19–52, 54, 56, 58–67, 69–76, 78–83, 87, 91, 92, 94, 98, 99, 103, 107, 109, 111, 113–124, 126–128, 130, 131, 133, 134, 136, 137, 139–141, 146–150

RU Read Unit 92–95

SA sense-amplifier 42, 57, 63, 65, 66, 68, 75, 76, 94, 99, 105, 106

SEE Single-Event Effects 30–32, 80, 81, 99, 106

SEU Single Event Upsets 30, 80, 81, 85, 88, 90–92, 95, 98–103, 106–108

SL Select Line 30, 81, 82, 84, 85, 93, 101

SPICE Simulation Program with Integrated Circuit Emphasis 3, 7, 32–35, 37, 46, 62, 65, 74–76, 80, 99, 104, 107, 109, 115, 117–120, 133, 134, 139, 145–147, 150

SRAM Static Random Access Memory 1, 3, 21, 25, 26, 32, 36–39, 41–45, 48, 49, 51, 52, 54–58, 60, 61, 81, 148

STDP Spike-Timing-Dependent Plasticity 113, 126

STT-MRAM Spine Torque Transfer Magnetic Random Access Memory 19, 20, 111

UDU Upset Detection Unit 91, 92, 94, 103, 104, 106

WDNM Write Dynamic Noise Margin 55, 56, 58

WE Write Enable 81, 82, 85, 87, 89–94, 96, 97, 99–103

WGU WE Generation unit 92–97, 99, 103, 104, 106

WL Wordline 44, 48, 66, 84, 85, 91

WTA Winner-Takes-All 113, 114

Chapter 1

Introduction

For the past four decades, scaling of Complementary Metal Oxide Semiconductor (CMOS) transistors has made it possible to integrate more than one billion transistors on a single chip in the state-of-the-art Integrated Circuits (ICs) and to have a wide range of products with very high levels of integration [1]. However, aggressive dimensional scaling of CMOS technology in sub-90nm nodes, specially for memory designs, has created significant design and technology challenges:

- Manufacturing tolerances in the fabrication process do not scale at the same pace as the transistor channel [2, 3]. Even with advanced fabrication techniques including double patterning [4, 5], shrinking the size of the memory cells to increase the memory density and capacity becomes a complicated and challenging task [6, 7].

- As the size of the devices scales down, the leakage power increases exponentially (up to 60% of the total power of Static Random Access Memory (SRAM) arrays fabricated using 65 nm technology [8]). The benefit of using the conventional power saving techniques, such as Dual Voltage Supply (DVS) [9, 10, 11] and power gating [12, 13], is diminishing with the continuous technology scaling due to the non-scalable leakage currents, supply voltage, and parasitics.

With the emergence of scalable Non-Volatile Memory (NVM) devices, the interest of incorporating them in various designs is growing in recent years to significantly decrease circuit leakage power [14, 15, 16, 17]. By pushing the data of less-frequently accessed blocks of the memory onto NVMs, the power signals of these blocks can be safely disconnected without losing the saved data. Due to their small sizes, the new emerging NVM devices can be

integrated to achieve higher energy savings with potentially less area penalty compared to when conventional power reduction techniques are used.

In addition to this, with the need to execute more complex computational operations in an energy-efficient and timely manner, the interest in neuromorphic systems has increased in recent years for machine learning applications. The neuromorphic systems structure is inspired by human brain where the information is transmitted between pre- and post-neurons depending on the states of synapses connecting them. NVMs are used to mimic the function of synaptic devices of neuromorphic systems due to the ease of arranging them in a 2-D array and their ability to save various states within the same memory cell [18, 19, 20, 21, 22, 23, 24, 25, 26].

Compared to other NVM devices, Resistive Random Access Memory (RRAM) devices have many advantages including:

- Compatibility with CMOS manufacturing technology and the possibility of their integration in 3D fashion which enables their fabrication within the metal layers or within the contact vias to the source and/or drain of a Metal Oxide Semiconductor Field Effect Transistor (MOSFET) [27, 28, 29, 30] (i.e., this means there is no need to use special materials or high temperature processes).
- Low-programming requirements ($\approx$ 1V as indicated in [31]).
- High ratio between their high and low resistance states.
- Fast switching speed between its high and low resistance values (in the order of 10 ns [32, 33]).
- Analog behavior due to the ability of the device to have many intermediate resistive states resulting from the high ratio between its lowest and highest resistance.

While the RRAM device has many advantages, it also has many disadvantages including the instability in its switching dynamics and the increased parasitic capacitances [32, 33, 34]. In particular, RRAM arrays suffer from reliability soft-errors due to the stochastic nature of oxygen vacancies movement within the RRAM oxide material [35, 36, 37, 38, 39, 40, 41]. Results reported in literature focus on the advantages of using RRAM in various designs. However, it is important to address the RRAM soft-errors as this can negatively impact the performance of the designs incorporating RRAM. Accordingly, the extend of RRAM arrays usage in products, despite its various attractive parameters, will be limited. Through the work in this thesis, we would like to bring the attention of how the RRAM soft-errors

can significantly reduce the performance of the systems to which they are integrated. In addition to this, we provide detailed analysis of how to detect and resolve RRAM soft-errors in current and futuristic platforms which can be used to run advanced machine learning applications.
A hierarchy of models and simulation tools have been used to validate our work:

- At the device level, physics-based RRAM Simulation Program with Integrated Circuit Emphasis (SPICE) models, which were previously reported in the literature with experimental verification [32, 33], are adopted. These models describe the physical behavior of the RRAM device including the drift of its resistive state due to the diffusion of the device oxygen vacancies.

- At the circuit level, HSPICE simulations are performed with memory arrays to verify the circuit functionalities and extract the critical information (e.g., delay, power, and resistive values of the RRAM devices) to be fed to the system level simulator.

- At the system level, various widely-adapted system level simulation tools and test benches are used including CACTI [42] (for estimating of power and energy consumption of high capacity memory arrays), BRIAN [43] (for estimating the accuracy of the neuromorphic system in classifying input patterns), and MNIST dataset [44].

While our suggested methodologies have been extensively verified through different SPICE and system simulations, future steps related to fabricating the proposed solutions is required in order to: a) verify their validity through silicon data, and b) enhance the suggested solutions to account for other effects that impact the performance of RRAM device including its sensitivity to process variations and the instability in its switching properties. In addition to this, some of the qualitative results are obtained from running simulations on certain system architectures (e.g., the architecture of neuromorphic system in chapter 6). The concepts behind the proposed methodologies should apply when the detailed implementation of the system changes. However, it is expected that the qualitative results for other system architectures may change.

The remaining chapters of the thesis are organized as follows: in chapter 2, a short summary is provided on the RRAM device physics, SPICE models, and various applications in which they are used. Following that, in chapter 3, a new design for integrating the RRAM device in SRAM arrays is discussed. This is basically to enable the incorporation of RRAM memories in platforms which are currently used to run machine learning applications (e.g., Graphics Processing Unit (GPU)-based platforms). In addition to this, the new design reduces the energy required to store and restore the SRAM data on RRAM

devices to increase the resilience of RRAM arrays to reliability soft-errors. Chapters 4 and 5 explain how the RRAM reliability and radiation soft-errors are detected and fixed in the one-Transistor-one-RRAM device (1T1R) arrays used in RRAM memories. After this, in chapter 6, a detailed analysis is provided on how the RRAM soft-errors can impact the performance of RRAM-based neuromorphic systems and how they can be detected and resolved.

Chapter 2

Review of Literature

This chapter summarizes the research efforts in literature on RRAM device. First, in section 2.1, the theory of RRAM device is explained. Then, in section 2.2, the basic operations of RRAM device (electroform and SET/RESET processes) are described. Following this, in section 2.3, the physics behind resistive switching in different RRAM devices is discussed. Then, in section 2.5, we provide a brief review of other popular NVM devices and compare their properties with those of the RRAM device. After this, in section 2.6, the different applications using the RRAM device are briefly summarized. The sources of RRAM soft-errors are discussed in section 2.7. A summary of the HfO_x RRAM model used in the subsequent chapters is introduced in section 2.8. Finally, we conclude the discussions in this chapter by clarifying the organization of the remaining chapters in section 2.9.

2.1 RRAM Device Theory and Characteristics

The basic theory of the memristor device was proposed by professor Chua in 1971 [45]. It was suggested that, theoretically, there should exist a fourth basic passive element which links the flux to electric charge in the same way as the resistor relates the voltage to current, the capacitor relates the voltage to charge, and the inductor relates the flux to current. The flux ($\phi(t)$) can be defined as the accumulation of voltage changes with time as described by equation 2.1.

$$\phi(t) = \int_0^t v(t)dt \tag{2.1}$$

Here, v(t) is the voltage signal applied to the device which is also a function of time t. Since the memristor device can accumulate the changes of applied voltage signal with time,

the device is said to have a "memory" of its previous state, hence the name memristor. The basic operation of the device is described by equation 2.2.

$$M(\phi, q, x) = \frac{d\phi(t)}{dq} = \frac{d\phi/dt}{dq/dt} = \frac{v(x,t)}{i(x,t)} \tag{2.2}$$

where:

- $M(\phi, q, x)$ is defined as the memristor value for a given flux $\phi(t)$ and charge "q" and state variable "x".
- x: is the state variable which defines the information stored on the memristor. Basically, when x=0, this means that the memristor has a high resistance.

The memristor device remained theoretical until 2008, when scientists fabricated the first memrsitor device. It was made of a TiO_2 layer with different oxygen atoms concentration which is sandwiched between two Platinium (Pt) electrodes [46]. This physical device uses the resistance as a variable to describe its state, the device is also known as RRAM device. Random Access part in the name of the RRAM device refers to the fact that the device will be used in memory arrays where the content of any location can be accessed directly without the need to go through any kind of special sequence. It is worth mentioning that there has been arguments about whether the RRAM is indeed a memristor device [47]. However, that is beyond the scope of this thesis. We will focus on the physical RRAM devices from now on.

In a more general way, as presented in [48], the conceptual memristors or physical RRAM devices could be represented by equations 2.3 and 2.4 which account for the dependence of state variable (i.e., "x" in eq. 2.2) change with time.

$$\frac{dx}{dt} = f(x, i, t) \tag{2.3}$$

$$v(i, x, t) = M(x, i, t)i(t) \tag{2.4}$$

Since the state variable "x" is a function in the applied current (Eq. 2.3) and the current by itself is also a function in "x" (Eq. 2.4), the authors in [48] predict that, for any periodic input signal, the change in "x" is also going to be periodic and dependent on the frequency and shape of the input signal. In addition to this, since, in general, the functions governing the change in current and state variable "x" can be different, the

authors predicted that the I-V characteristics of a memristor, which match those of the physical RRAM devices, should have a pinched hysteresis loop shape as shown in fig. 2.1. The line with higher current levels in fig. 2.1 represents the Low Resistive State (LRS) of the memristor, while the line with smaller current values represents the High Resistive State (HRS) of the device. The process of changing the state of the memristor from LRS to HRS is called RESET, while the opposite operation is referred to as SET process. The voltage and current curves in fig. 2.1 are generated from running Direct Current (DC) analysis using the SPICE model for HfO_x RRAM device described in [41].

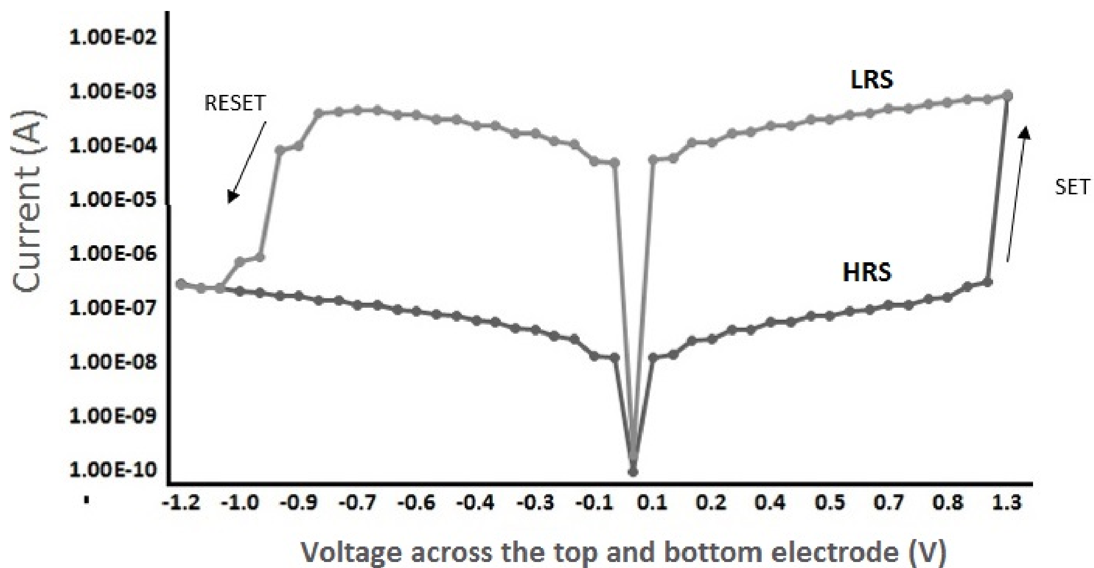

Figure 2.1: Basic I-V characteristics of RRAM. This graph is generated from running SPICE simulations using the model described in [41] for the HfO_x RRAM device. The behavior matches the hysteresis loop shape predicted for the conceptual memristor.

The structure of the first fabricated TiO_2 RRAM device, discussed in [46], is shown in fig. 2.2.

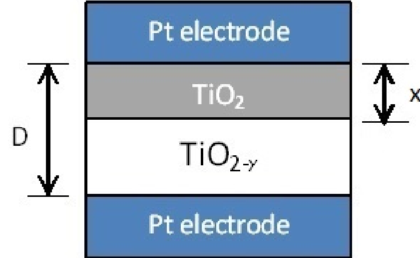

Figure 2.2: Structure of the first TiO_2 RRAM device. The parameter "D" in the figure describes the thickness of oxide layer while the parameter "x" defines the gap distance separating the top electrode from TiO_{2-y} layer which is the part of oxide material containing the most of oxygen vacancies (i.e., current carriers). The "y" index is used to indicate the existence of oxygen vacancies in the oxide material.

2.2 RRAM Device Operations: Electroform, SET, and RESET Processes

In this section, the first developed TiO_2 RRAM device is used to explain the basic operation of RRAM devices. The switching mechanism of the device varies depending on the materials used in its fabrication as explained in section 2.3 where we show that, in addition to the oxide-based RRAM devices, there is another group of materials (i.e., ion-based materials) which can be used to form RRAM devices (Conductive Bridge Random Access Memory (CBRAM)). Instead of using oxygen vacancies in the oxide material to conduct the current between the top and bottom electrodes, CBRAM devices form conductive channels through the diffusion of ions from the top and bottom electrodes. Unlike oxide-based RRAM devices, CBRAM requires the use of specific materials in their fabrication as explained in section 2.3. In this thesis, we use oxide-based RRAM device (in specific the HfO_x RRAM device for the reasons explained in section 2.8). Unless specified, the term RRAM in this thesis refers to the oxide-based RRAM device. However, the methodologies and circuit designs, discussed in this work, could be easily adapted to any other type of RRAM device including CBRAM, while the qualitative conclusions should be applicable without loss of generality.

The TiO_{2-y} layer in fig. 2.2 describes the part of TiO_2 material which contains oxygen

vacancies. To create the TiO_{2-y} layer, an one-time electroforming process must be applied [49]. By applying a voltage sweep from 0 to 6 V over 5 ms on the device, the electroforming process causes local heating and high electrical potential in the oxide material. This produces an irreversible decrease in the resistance from the as-fabricated GΩ range to kΩ/MΩ range due to the generation of conductive paths made of oxygen vacancies within the oxide material [50]. To switch between the LRS and HRS states of the RRAM device, a voltage or current signal needs to be applied across the device terminals to induce the motion of the generated oxygen vacancies. A more detailed insight of what happens in the device after electroforming could be explained as follows:

- **Applying negative voltage after electroforming**: When a negative voltage is applied to the device after electroforming, the oxygen vacancies from the TiO_{2-y} layer are attracted to the top electrode. Under high electric field, the oxygen vacancies react with the molecules of the TiO_2 material forming conductive channels of the Ti_4O_7 which extends to the top electrode. The motion of the oxygen vacancies can be described by the ion hopping model [51]. Fig. 2.3 shows an example of the movement of oxygen ions for 3 time samples. The idea is that, under high electrical field, the oxygen atoms tend to jump to the nearest vacancy positions. Then, they keep on hopping from one lattice to another position until they reach the electrode with positive voltage leaving behind oxygen vacancies in TiO_2 layer which results in creating a low resistance path between the two electrodes. Hence, the resistance of the device switches to its LRS. This operation is known as SET process.

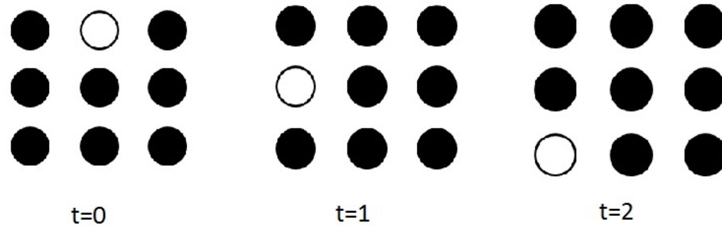

Figure 2.3: Ion hopping illustration. The figure shows how the oxygen atoms (black dots) hop randomly at each time instance to fill the oxygen vacancies in the crystal structure (white dots).

- **Applying positive voltage after electroforming**: A positive potential on the top electrode pushes the oxygen vacancies away. Accordingly, the previously created

channels between the TiO_{2-y} layer and the top electrode disappears due to recreation of the TiO_2 layer in the midst. Hence, the resistance of the device switches to its HRS. This operation is called RESET process.

The resistive switching of TiO_2 RRAM device can be summarized as in fig. 2.4. It is

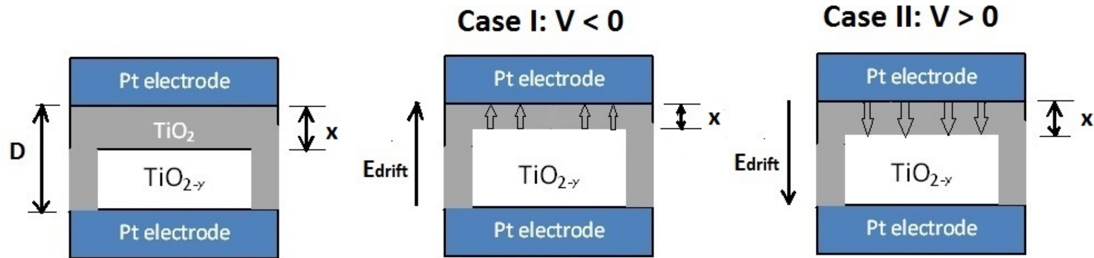

Figure 2.4: Simple illustration of resistive switching in TiO_x RRAM devices. The parameters "D" and "x" define the thickness of oxide material and the gap distance separating the top electrode from the TiO_{2-y} layer. When a negative voltage is applied on the top electrode (Case I), the oxygen vacancies drift towards the top electrode under the effect of high electric field E_{drift} causing the gap distance "x" to decrease. Oppositely, if a positive voltage is applied (Case II), the direction of E_{drift} changes and the oxygen vacancies are pushed away from the top electrode increasing the gap distance "x".

worth mentioning that, due to the stochastic nature of oxygen vacancies movement, the switching characteristics for SET/RESET process are not deterministic. This is explained by the numerous gray I-V characteristics curves in fig. 2.5 obtained through cycling between SET/RESET process multiple times. Fig. 2.5 shows that the path, through which the RRAM device changes its state, can be different with each switching cycle.

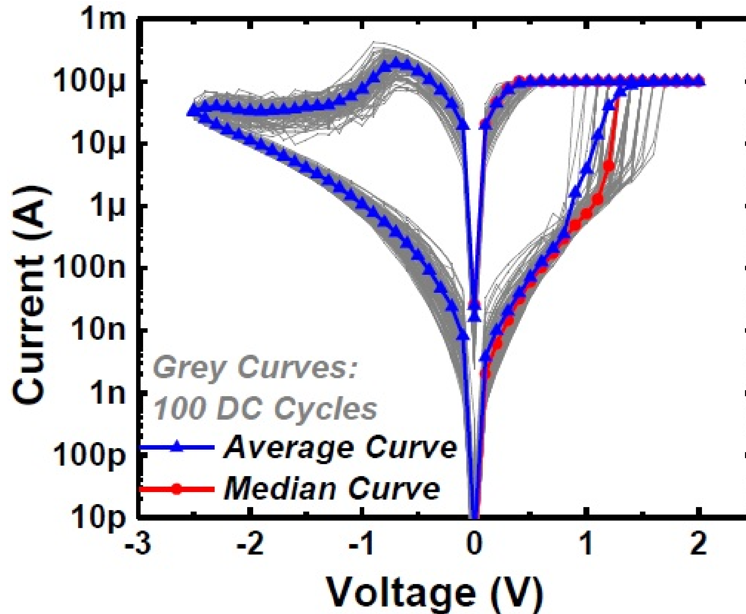

Figure 2.5: Effect of stochastic oxygen vacancies movement on generating different paths for the SET and RESET process with each switching cycle [32]. Permission granted to use the figure.

2.3 Physics of Resistive Switching

Section 2.2 explains the switching mechanism of TiO_x RRAM. When different materials are used, other resistive switching mechanisms in the RRAM device have been observed. The physics behind these variations of I-V characteristics are discussed in this section. All the switching dynamics in the different RRAM devices happen due to chemical reactions (redox) which take place in the oxide material. The main contributor in all these chemical reactions is the oxygen vacancies movement (oxide-based RRAM). These mobile species move under the effect of a high external electric field and/or under the effect of Joule heating inside the oxide material due to the high current density passing through the small dimensions of the device. For example, in oxide-based RRAMs switching due to Joule heating, the current density is in the order of $\geq 10^6$ A$/cm^2$ as reported in [52, 53]. Electric field and Joule heating generally coexist in all memristive switching, although their relative importance depends on the device stack, materials, and electrical operation history. In all cases, there are four main driving forces that work independently or together which are:

a) electric potential gradient (field), b) electron kinetic energy, c) species concentration gradient and d) temperature gradient. In other words, the switching mechanism in oxide-based RRAM devices could be categorized in four main categories as shown in fig. 2.6. These various mechanisms lead to different I-V characteristics for the devices as illustrated in the insets of fig. 2.6.

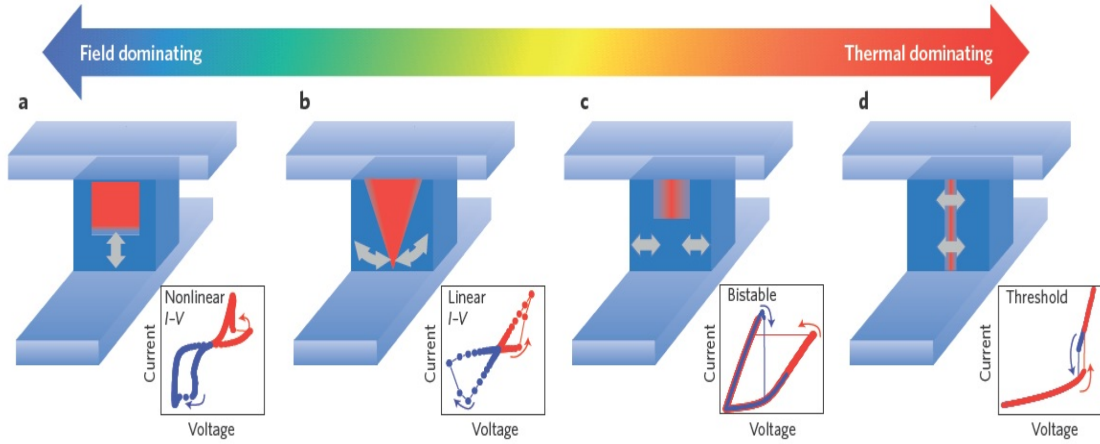

Figure 2.6: Switching mechanisms in oxide-based RRAM devices described [47] which change depending on the RRAM oxide and electrode materials. Permission granted to use the figure.

- **Bipolar nonlinear switching**: Fig. 2.6a presents schematically a device driven by an electric field. The vertical growth and retraction of the conduction channels under the electric field results in the typical switching I-V characteristics shown in the inset of fig. 2.6a. The term "bipolar" refers to the fact the device switches from LRS to HRS and vice versa using positive and negative voltages.

- **Bipolar linear switching**: Fig. 2.6b shows another type of bipolar switching, which has a linear IV curve in both the ON and OFF states as shown in the inset of that figure. In this case, there is a conduction channel connecting the top and bottom electrodes all the time in the entire switching cycle. The resistance is mainly changed because of the change in composition, volume or geometry inside the channel, which is a result of the combined effect of the vertical drift resulting from the high electric field and the thermal lateral diffusion as discussed in [47].

- **Unipolar bistable switching**: Unlike bipolar switching, the RRAM device, which have unipolar I-V characteristics, switch between their LRS and HRS and vice versa using only positive voltages. Generally speaking, the bipolar switching is prevalent when the electric field effect is more dominant than the thermal effect. Whereas, the unipolar switching is happening when the thermal effect has a more dominant role in the switching mechanism. The main cause of unipolar switching shown in fig. 2.6c is still controversial. Yet, as discussed in [54], the popular theory is related to the soft breakdown of the dielectric material caused by the applied electric field at the beginning of the switching. Then, this leads to a high current followed by heat-assisted ionic motion. The reset switching is normally described as a thermal disappearance of the conduction channel which could be caused by: a) thermal diffusion driven by concentration gradient. b) reduction of free surface energy of the filament or c) a phase-change process induced by heat and/or electric field.

- **Unipolar threshold switching**: The switching in fig. 2.6d has a different kind of state switching. With the increase of the applied current, the insulating device suddenly becomes metallic at a certain current level. Yet, after reducing the current level, the device becomes insulating again. This kind of switching is reported in the NiO_x RRAM device [53, 55] due to the spontaneous rupture of the filament channel at high temperature.

Unlike oxide-based RRAMs, the switching mechanism in CBRAM devices is caused by the ion diffusion from the electrodes. CBRAM has a similar structure as oxide-based RRAM. However, one of the CBRAM electrodes is made of materials, such as Copper (Cu), silver (Ag) or an alloy of these metals (e.g., CuTe). Also, the oxide material between the electrodes is replaced by electrolyte like amorphous Si. Unlike oxide-based RRAM devices, the majority of CBRAMs are switched by the electric field. The electroforming step is also distinguished from that of oxide-based RRAM device such that a positive high voltage on the electrode oxidizes its atoms. This generates ions which can later penetrate in the electrolyte material when a bias is applied on the top electrode. These ions drift across the electrolyte material under the effect of high electric field until they reach the bottom electrode. The injected ions are then recombined and deposited on the surface of bottom electrode. During the SET process, the metal ions grow until they reach the top electrode switching the device ON (LRS of the device).During the RESET process, a positive voltage is applied to the electrochemically inert bottom electrode to dissolve the previously created filaments and switch the device OFF (HRS of the device). In this process, the electric field is the only driving force and Joule heating is negligible given the small current usually involved in these devices.

Table 2.1 summarizes the different RRAM devices reported in the literature and their associated switching mechanisms.

2.4 Summary of the RRAM Features

The RRAM device has many advantages including:

- **Small size**: The RRAM device could be scaled down to 10nm x 10nm dimensions. This enables it to be used as high density memory blocks.
- **Low power consumption**: In order to change its resistance state, the RRAM device requires voltages as low as 1V.
- **Non-linear I-V characteristics**: Due to their non linear characteristics, RRAM devices can be programmed in durations in the range of tens of nanoseconds.
- **Non-volatility feature**: RRAM devices can retain their state in the absence of power supply for a long period of time (in the range of years). This is mainly because the oxygen vacancies movement is very slow and it gets only accelerated by several orders of magnitude when an electrical field is applied on the device terminals as described in [51].
- **Compatibility with CMOS process**: Due to its very simple structure and the usage of materials already utilized in the CMOS fabrication process, the RRAM device could be easily integrated in the various CMOS designs.
- **Analog behavior**: Due to the high ratio between their LRS and HRS, RRAM devices can be programmed to other intermediate resistive states.

However, the RRAM device needs to be carefully incorporated in the various designs due to its pitfalls including:

- **Instability in the device switching behavior**: Stochastic nature of oxygen vacancies movement in the oxide causes the instability in the device switching characteristics as explained in section 2.2.
- **RRAM soft-errors**: RRAM devices suffer from reliability soft-errors and, if deployed in high radiation environment, the data saved on the RRAM arrays are subject to radiation soft-errors as discussed in section 2.7.

Table 2.1: RRAM devices comparison

RRAM structure	**Current transport mechanism**	$R_{off}/R_{on} ratio$	**Programming signals**	**Similar devices**
Ta/TaO_x/Pt [31]	Joule heating (Bipolar linear swicthing)	45	+0.7 and -1.25 V signals	TaO_x/Cu RRAM [56] [57]
Ag/a-Si/PolySi [58]	Drift (Bipolar non-linear switching)	$20 * 10^3$	4V and -4V	TaN/SiO_x/n++ Si RRAM [59], AlN RRAM [60], Cu/CuO RRAM in [61] [62], TiO_2/ITO [63] , HfO_x RRAM [64] [65], Pt/ZnO/Pt RRAM [66], TiN/ZnO/Au RRAM [67]
Au/ Nb_2O_5 /Nb/Si RRAM [68]	Joule heating (Unipolar bistable switching)	10^3	0.9V/2.8V	NA
Silver Chalcogenide RRAM [69]	Ion-based switching	NA (this RRAM tends to switch to its LRS once set to HRS)	0.24V/-0.32V	NA
NiOx RRAM [53, 55]	Joule heating (Unipolar thershold switching) after certain Temperature	2.5	2 and 5 V	NA

- **Manufacturing costs**: Since the device can be integrated as a via between the routing levels on top of the Metal Oxide Semiconductor (MOS) transistors, this might require using more masks in the device fabrication process and hence, increasing the price per chip. This is in addition to increasing the parasitic capacitance to MOS junction.

In order to use RRAM devices as Multi-Level Cells (MLCs) despite the instability in its switching characteristics, the authors in [70] proposed the integration of multiple RRAM devices in parallel and then use each device in binary mode (i.e., each RRAM cell is only programmed to be either in its LRS or HRS which corresponds to saving logic '0' or '1', respectively).
Moreover, for the possible problem of increased parasitic capacitances resulting from having the RRAM device fabricated as a via layer between higher metal levels, the authors in [34] proposed integrating RRAM device directly on the substrate right next to MOS transistor. However, there are still technical difficulties in enabling this solution related to the increase in programming voltages and the reduction in resistance range of RRAM device.
In addition to this, a lot of research nowadays is investigating the usage of other selector device than MOS transistors [71, 72, 73, 74]. This is basically needed to further decrease the footprint of RRAM memory cells. However, the operation of selector devices has limited endurance ($\approx 10^3$ to around 10^6 cycles [71, 72]). This can hence significantly reduce the endurance of RRAM arrays. Moreover, more studies need to be conducted regarding the tolerance of these device to various soft-errors effects.

One of the main contributions of this thesis is providing circuit and system solutions for RRAM reliability and radiation soft-errors which can speed the adaptation of RRAM technology in products. Moreover, in this work, we demonstrate how the RRAM device can be reliably integrated in current and futuristic platforms running machine learning applications as discussed in chapters 3 and 6.

2.5 Comparison to Other NVM Devices

RRAM is among the various types of NVM emerging in recent years. There are two other famous technologies which have been widely investigated [75]. Each of them is discussed in a separate section.

2.5.1 Phase-Change Random Access Memory (PCRAM)

Phase Change Random Access Memory (PCRAM) cells refer to the cells that use the change of state of a chalcogenide material (typically $Ge_2Sb_2Te_5$ alloy) by temperature as a mean to store data on the cells. In more specific terms, when the chalcogenide material is at the crystalline state, the PCRAM resistance is quite small. When it is at the amorphous state, the device resistance is high. The ratio between the two states is about 4 to 5 orders of magnitude. To change between the amorphous and crystalline states, a large current must be applied to heat the material. In order to change from crystalline to amorphous state, the material must be first brought to its melting point. Then, the programming current is abruptly stopped to allow for the amorphous state to be formed (i.e., melt-quench the material). The basic structure of PCRAM cell is shown in fig. 2.7.

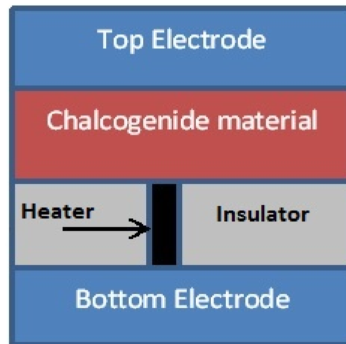

Figure 2.7: PCRAM cell structure in which the heater layer is used to cause enough heat within the chalcogenide material to change from amorphous to crystalline structure and vice versa.

Other than the need for high current as discussed in [76], the PCRAM cell suffers from other disadvantages:

- Increasing memory cell density can lead to an increase in the thermal disturbance caused by the temperature rise in the adjacent cells during the programming of neighboring selected cell. The accumulated effect of temperature rise can result in retention failures by crystallizing the PCRAM cells in amorphous state [77, 78].

- The amorphous state of the chalcogenide material drifts with the repeated programming of the PCRAM cell. This means that the high resistance value as well as the value of the required programming current changes with time [79, 80].

2.5.2 Magnetic Random Access Memory (MRAM)

Magnetic Random Access Memmory (MRAM) cells refer to the cells that use the magnetization direction of two magnetic materials (e.g., "CoFe") as a mean to store data. These devices use the concept of electron spin which is introduced in quantum mechanics to explain how two electrons can coexist in the same energy level. Accordingly, the idea behind these devices is to use ferromagnetic materials, such as "CoFe", since their electrons have already a preferred spin direction. One of the magnetic layer of the device is usually thick with large magnetization barrier to form a "pinned" layer where the electrons never change their spin direction. The other magnetic layer is usually thin which can be programmed by changing the polarization direction of its electrons. The two magnetic layer of MRAM are integrated together through a thin insulator. The device works such that:

- When the two ferromagnetic layers have the same polarization directions, the probability of the electron passing through the insulator (i.e., the magnetic tunnel) and finding an empty state in the other layer with the same polarization direction is high. Accordingly, the current can pass much easier and hence the MRAM device is said to be in the low resistance state.
- When the two ferromagnetic layers has opposite polarization directions, the probability of the electron passing through the tunnel and finding an empty state in the other layer with the same polarization direction is low. Accordingly, the current passing through the device is small and hence the MRAM device is said to be in the high resistance state.

The basic architecture of MRAM cell is shown in fig. 2.8.

There are two possible ways to change the magnetization direction:

- **Current induced magnetic field**: The idea is to pass a current in a nearby connection. Accordingly, a magnetic field is created and this changes the magnetization direction of the free layer according to the direction of current flow. This method is not commonly used as it requires high programming current (1.5 mA - 4 mA) which does not scale down when decreasing the size of MRAM cell [81].

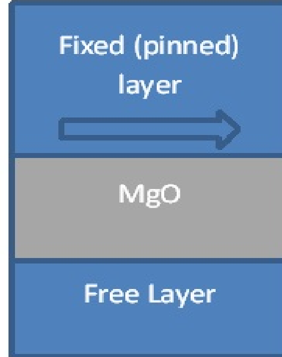

Figure 2.8: MRAM cell structure. The pinned layer is the thick magnetic layer which has only one direction of magnetization shown by the arrow in figure. The free layer is the other magnetic layer whose direction of magnetization can be programmed.

- **Spin polarization current induced magnetic field**: In this case, a high-amplitude current pulse is applied through the pinned layer to polarize the current. Then, this spin-polarized current passes through the free layer. Depending on the direction of the current, the spin polarity of the free layer can be made either parallel or anti-parallel to that of the pinned layer. The MRAM arrays programmed using this technique are called Spine Torque Transfer Magnetic Random Access Memory (STT-MRAM) arrays. This is the most commonly used MRAM cell because its programming current is small ($\approx$ 0.2 mA) [82, 83]. In addition to this, STT-MRAM cells solve the scalability problem faced previously in other MRAM cells.

Despite the various advantages of STT-MRAM (i.e., high endurance ($\geq 10^{15}$), fast programming and reading times (tens of nanseconds), and good retention features), the STT-MRAM suffers from a major disadvantage. The ratio between LRS and HRS of the device is quite small (around 10 at room temperature [82] [83]). Accordingly, the STT-MRAM arrays require using more complex peripheral circuitry to sense the small resistance difference between the HRS and LRS of its cells. Table 2.2 summarizes the comparison results between PCRAM, MRAM, and RRAM cells. It is worth mentioning that there are variations of each type of these devices. For example, for RRAM device, this includes STI-RRAM [34] and Vertical-RRAM [70]. However, in table 2.2, we focus on comparing the different NVM technologies and not focus on variations of the same technology.

Table 2.2: NVM cells comparison

Parameter Name	**PCRAM** [76, 77, 78]	**STT-MRAM** [82, 83]	**RRAM** [37, 38, 39]
Definition of HRS, LRS	Crystallization of chalcogenide material	Magnetization direction of two ferromagnetic materials	Oxygen vacancies move in the oxide material
Programming time	few hundred nanoseconds	tens nanoseconds	tens nanoseconds
HRS/LRS ratio	Very high (4-5 orders of magnitude)	Very small (< 1 order of magnitude)	5 order of magnitude
Endurance	Moderate (10^9 cycles)	Very high (10^{15} cycles)	high (10^{12} cycles)
Temperature Stability	Unstable	Stable	Unstable
Scalability	Proven to be scaled [78]	can be scaled but not peripheral circuits	scalable
Compatibility with CMOS	Requires the use of special materials	Requires the use of special materials	Does not require the use of special materials

2.6 Applications Using RRAM Devices

In this section, the various applications incorporating the RRAM devices are listed. RRAM arrays are broadly used to either: a) reduce the leakage power of CMOS designs by backing up the state of infrequent blocks of SRAMs or register arrays on RRAM memories before cutting off their power signals, or b) implement a highly scalable neural synaptic network used in the advanced neuromorphic systems.

2.6.1 Use in Crossbar Random Access Memory Arrays

The main idea is to add RRAM devices at the intersection point of every vertical and horizontal data lines of the crossbar structure to form memory arrays as shown in fig. 2.9.

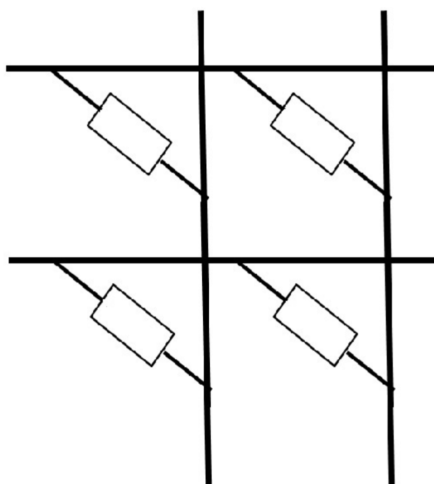

Figure 2.9: Crossbar RRAM structure where the RRAM device is used to connect between the row and column data lines.

Depending on the state of RRAM, the horizontal and vertical data lines are either connected (i.e., RRAM is at LRS) or disconnected (i.e., RRAM is at HRS).
One of the most common problems known with this architecture is the sneak-path issue which is defined as the error in reading the state of selected RRAM cells due to the current coming from half-selected cells [84]. This occurs when a RRAM cell in HRS is read, while there is another parallel path formed by a series of RRAMs in LRS. This causes an error in reading the HRS of RRAM device. This problem also affects the write operation as

discussed in [85]. Fig. 2.10 shows an example of how sneak-path could occur in RRAM memory arrays. The black boxes in fig. 2.10 represent the RRAM devices in LRS, while the white boxes represent RRAM cells in HRS. A lot of research efforts have been employed

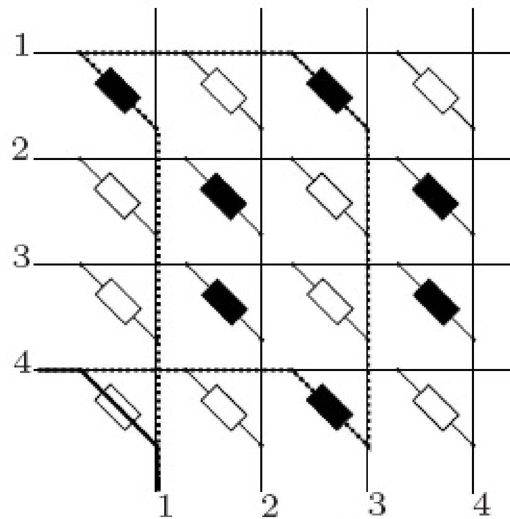

Figure 2.10: Illustration of the sneak-path issue [86]. The RRAM at row 4 and column 1 is the memory cell that is intended to be read, while the other RRAM devices on the dotted line are the LRS RRAM memory cells on the sneak-path. Permission granted to use the figure.

to overcome this problem which can be categorized as follows:

- **Using a 1T1R cell**: One of the proposed solutions is to integrate a transistor with each RRAM device to prevent the sneak-path [85]. Before this, it was proposed in [87] to have a one diode in series instead. Yet, due to the need to program the RRAM device using both voltage polarities, this architecture could not persist. The basic architecture of the 1T1R memory cell, shown in fig. 2.11, is similar to that of Dynamic Random Access Memory (DRAM) cell.

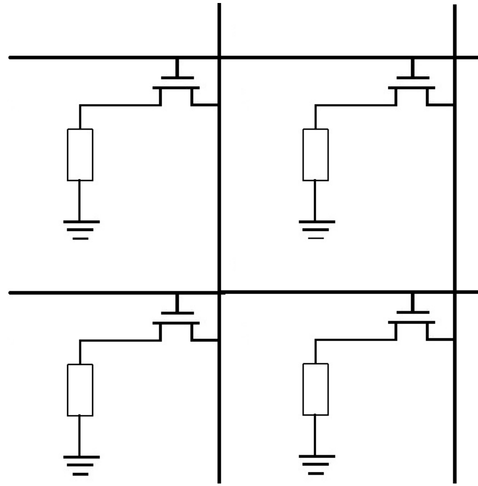

Figure 2.11: 1T1R cell proposed in [87] which requires the existence of both negative and positive high potentials. Permission granted to use the figure.

The main disadvantage of this solution is that it increases the cell size which by consequence reduces the density of RRAM array in addition to the need of having positive and negative potentials to program the RRAM cells.
To overcome the need to use positive and negative potentials, the authors in [88] suggested having two complementary bitline signals connected to the 1T1R cell as shown in fig. 2.12.

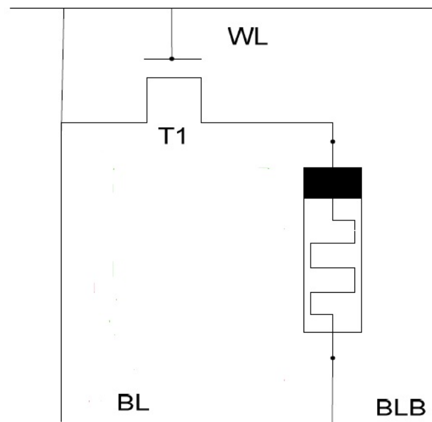

Figure 2.12: The alternative 1T1R cell proposed in [88]. To overcome the need to use both negative and positive high potential voltages, the signals "BL" and its inverted version "BLB" are applied on the terminals of RRAM device. Permission granted to use the figure.

This proposal solved the problem of using two opposite high voltage signals, but it requires increasing the area of 1T1R memory cell to account for the extra bitline/cell.

- **Encode the patterns saved in the crossbar array**: Another suggestion is proposed in [89], where the authors suggested forcing the numbers of LRS and HRS RRAMs in each row and column to be the same in order to reduce the effect of sneak-path. This is mainly because of the existence of the HRS RRAMs in multiple sneak-paths. This technique does not eliminate the effect of sneak-path but rather it reduces its occurrence. In [86] and [90], the authors present a mathematical model of how to store data in a RRAM crossbar memory in a way that can eliminate the sneak-path issue. The basic idea is to isolate the RRAM cells at LRS by surrounding them with RRAM devices at HRS. Although this can eliminate the need to have extra access devices (e.g., transistors, diodes,..) with RRAM devices, it limits the storage capacity of the memory array in addition to complicating the control of the memory operations.

- **Using Read/Write periphery circuits**: The basic architecture for the read and write voltage configuration, which can combat the sneak-path effect, is explained in [91] and shown in fig. 2.13. The idea is that any unselected row/column is at half of the programming voltage (V/2). Only the selected row is connected to the programming voltage (V) and its corresponding column is grounded. Same thing

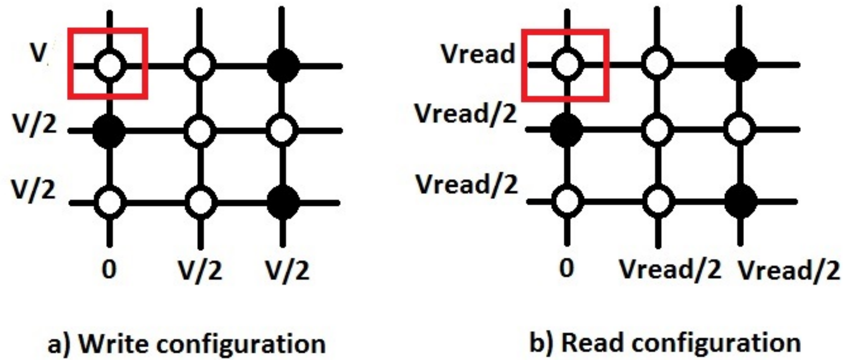

Figure 2.13: The basic read/write voltage configuration for RRAM arrays proposed in [91]. The unselected control lines are connected to V/2 where V represents the voltage level required to be applied on RRAM device to trigger either SET/RESET process. Only for the RRAM cell that is meant to be programmed, the voltage drop across its terminals will be V. Permission granted to use the figure.

applies to the read operation but the assigned potential voltage (i.e., Vread) is much less than programming voltage. The main problem with this architecture is that, during the read operation, the current from half-selected cells can cause read failures.

2.6.2 Use in Low-power SRAM Designs

RRAM device is also used in the SRAM design. Using RRAMs in SRAM arrays adds many advantages to SRAM cell:

- Reduce the leakage power of the SRAM cell by saving its state on a RRAM which gives the chance to turn off its power without the risk of losing the data.
- Reduce the total memory size on the chip level as it removes the need to have a separate flash memory block.
- Enables fast turn-on and turn-off processes for SRAM arrays. Currently, the data of SRAM cells is saved on a separate module made of flash devices during power-off and data-backup operations [92]. This scheme requires long store/restore time due to the word-by-word (i.e., serial) SRAM read/write operations resulting in extended power on/off time.

This actually makes the integration of RRAMs in SRAM cells more attractive as the resulting Non-Volatile Static Random Access Memory (NV-SRAM) arrays can easily fit in the mobile applications which require low voltage operation and extremely low leakage power. Examples of such work include those explained in [17, 93, 92, 94]. All those designs require high energy to store and restore the SRAM data on RRAMs. A detailed analysis of those designs is discussed in chapter 3 together with our proposed methodology for integrating the RRAM device with SRAM cells.

2.6.3 Use in Low-power Sequential Circuits

The main reason for using RRAMs in sequential circuits is to reduce the leakage power by introducing the RRAM device as a non-volatile latch. Previously, to reduce the power dissipated during the inactive times of the sequential circuits, the power gating methodology was used as shown in fig. 2.14. The concepts behind using power gating technique are:

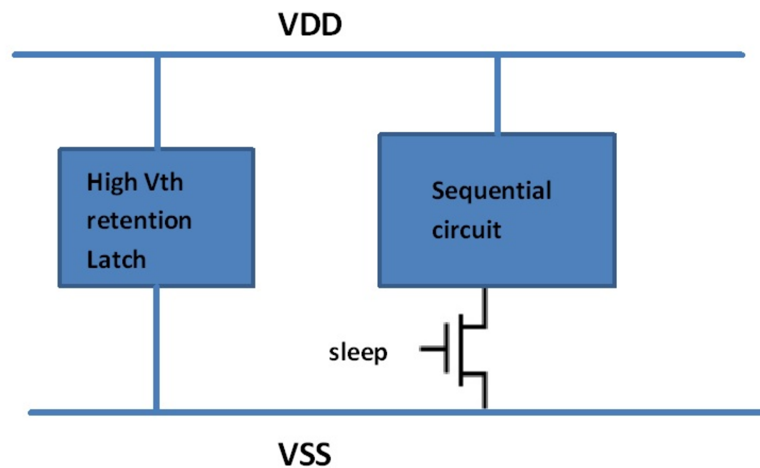

Figure 2.14: Architecture of a power-gated sequential circuit. Before enabling the "sleep" mode of operation, the sequential circuit data is pushed first on the retention latches which use high-Vth MOS transistors to reduce the subthreshold leakage power.

- The sequential circuit is disconnected from the power supplies during the sleep times.
- To prevent losing the state of the sequential circuit, another high-Vth retention latch, that is always connected between the power supplies, is used. The reason of using high-Vth MOS transistors in this latch design is to reduce the subthreshold currents.

Although this methodology reduces the leakage power significantly ($\approx$ 2x as per [95]), there are two main disadvantages associated with the power gating technique:

1. The use of high-Vth retention latches can only reduce the subthreshold leakage power. Yet, other leakage sources (e.g., gate leakage) are not eliminated by this technique.

2. The reduced power is still high for low power circuits such as wireless network sensors or biomedical implant systems which require the usage of very limited amount of power during their long sleep times.

Due to its capability of holding its state even in the absence of supply power in addition to its other advantages including its small size, RRAM devices are used instead of the conventional retention latches [14, 15, 16]. Despite the advantage of removing the need to have retention latches, the circuits in [14, 15, 16] suffer from many disadvantages:

- **Non-optimized usage of RRAM arrays**: This is because, in all those designs, the data is written on the RRAM cell no matter whether the value to be saved is the same as the one already stored on the RRAM device or not. This actually causes a large power consumption dissipation.

- **Negative impact on the circuit characteristics**: The addition of extra RRAM circuitry affects the normal operation of the D Flip-Flop (DFF). For example, in case of the circuit in [14], the propagation delay of DFF is increased by 14x.

2.6.4 Use in Neuromorphic Systems

Neuromorphic system is built to mimic the functionality of neuro biological system which has a very high processing speed and low power consumption. Neuromorphic systems are characterized by:

- Large connectivity network between its different components which can offer highly parallel processing power. The connections in these systems are called synapses.

- Ability to learn frequent patterns and enhance the connection between the components involved in learning these patterns.

- Adaptation to local changes by easily resetting the unused connections.

Due to its small size, compatibility with CMOS technology, high HRS/LRS ratio, and analog behavior between its HRS/LRS, the RRAM device is one of the famous NVM candidates which can represent the functionality of synapses in neuromorphic systems [96, 97]. A typical representation of the RRAMs in the neuromorphic systems is illustrated in fig. 2.15. Such 2-D configuration of the RRAM array can complete complex operations,

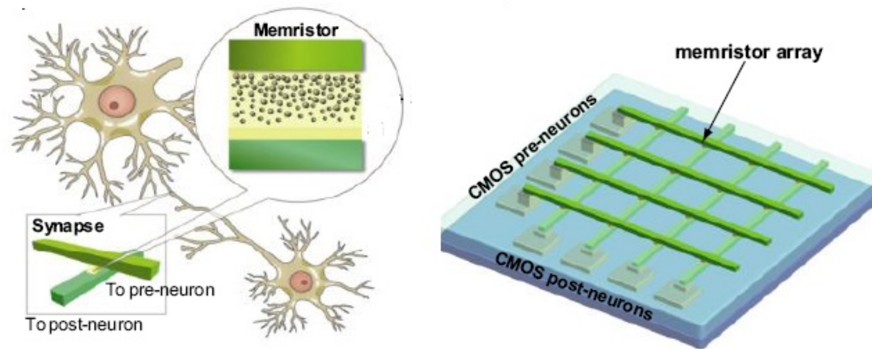

Figure 2.15: RRAM usage in neuromorphic circuits [98]. Left part shows how the RRAM device can mimic the functionality of synapses in the biological neural system. Right part of the figure illustrates how the biological neural system is implemented in circuits by having pre- and post-neurons communicating pulses depending on the status of RRAM devices connecting them. Permission granted to use the figure.

such as matrix multiplication, in fast and energy efficient manners. What makes the RRAM device capable of mapping the functionality of synapses is related to the fact that its resistance is dependent on: a) the voltage amplitude applied on its terminals, and b) for how long the voltage is applied. This mimics exactly the required learning feature of synapses in the neuromorphic systems as discussed in chapter 6.

2.7 RRAM Soft-Errors

Although the RRAM device is a promising NVM technology which can be used in various applications, the device suffers from reliability and radiation soft-errors. These soft-errors, and particularly the reliability soft-errors, are the main challenges in adopting RRAM in mass production. In this section, we describe these physical phenomena and the reason of their occurrences.

2.7.1 Reliability Soft-Errors

Reliability soft-errors are the revertible errors seen in the data stored in RRAM arrays over the course of their usage. While the RRAM device, as all the other NVM cells, suffers also from hard-errors which puts a boundary on the number of read/write cycles, the soft-errors can significantly reduce this limit (10^4 cycles instead of 10^{10} cycles as in [38, 37, 36]). As detailed in [35, 36, 37, 38, 39], the RRAM reliability soft errors results from:

- The diffusion of oxygen vacancies out of the conductive filament containment due to the concentration gradient of the vacancies within and outside the conductive filaments [38, 39, 36]. Fig. 2.16 illustrates the oxygen vacancies diffusion phenomenon.

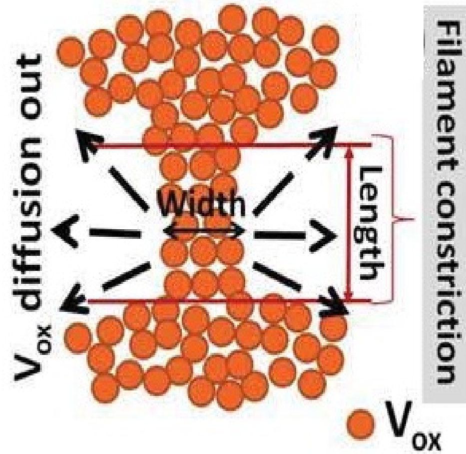

Figure 2.16: Oxygen vacancies (V_{ox}) diffusion out of the conduction filaments containment [36]. The parameters "width" and "length" define the dimensions of conductive filaments which are affected by the reliability soft-errors. Permission granted to use the figure.

With more stimulus applied to the device, more heat is generated within the conductive filaments which speeds up the diffusion process and hence the loss of RRAM resistive state. One possible reason for this is using unbalanced programming pulses [35, 38]. Depending on the voltages levels used, either extra generation or recombination of oxygen vacancies can occur. This impairs the capability of the device to switch properly and can even reduce the number of switching cycles of the device from 10^{10} to almost 10^4 cycles [35].

- Introducing manufacture defects during the fabrication process of the RRAM device which react with the oxygen vacancies in the conductive channels [37].

With the continuous improvement in the fabrication process, the soft-errors due to the manufacture defects can be suppressed. As explained in [37], by introducing a final annealing step in the manufacturing process, the soft-errors resulting from manufacture defects induced can be significantly reduced. Hence, we focus in chapters 4 and 6 on providing circuit and system solutions to address the reliability soft-errors generated from diffusion of oxygen vacancies out of the filament containment.

2.7.2 Radiation Soft-Errors

Event upsets are generally caused by highly energetic particles (i.e., protons, neutrons, alpha particles, and heavy-ions) striking the sensitive locations in the memory array which results in unintended changes in the saved data. Those charged particles come from various sources such as: a) packaging materials used in the ICs [99, 100], and b) cosmic rays that can produce heavy-ions with high-energy [101]. Depending on the energy of striking particles, the saved information in the memory cell can either switch directly to the opposite logic state (i.e., from logic '1' to '0' or vice versa) or change to an intermediate logic level. For the case causing intermediate changes in the saved data, multiple strikes have to occur to completely flip the logic state of the memory cell from one state to the other. These soft-errors are classified as Multiple Event Upsets (MEU) [40, 41]. Single Event Upsets (SEU) define the cases when the energy of striking particles is high enough to directly toggle the memory cell data to the other state.

Although the RRAM device by itself is immune to Single-Event Effects (SEE) [40, 41], the access transistor added in 1T1R arrays to prevent the sneak-path issue (discussed in section 2.6) can unintentionally change the RRAM resistive state. Based on the detailed analysis and experimental results in [40, 41], 1T1R cells can be divided into: a) fully-selected cells which are intended to change through the write operation, b) half-selected cells which are not intended to change by the write operation but share one or more of the control lines with the fully-selected cells, and c) unselected cells which are not intended to change by the write process and they do not share any of the control signals with the fully-selected cells. Unselected 1T1R cells are not susceptible to soft-errors induced by heavy-ion strikes since the biasing voltages on the terminals of their RRAM devices are not high enough to cause any changes in their states. As for the fully-selected cells, the heavy-ion strikes do not cause soft-errors. This is because the access transistor is turned on and one of the terminals of the RRAM device is connected to ground (i.e., Select Line

(SL) is connected to ground during SET process, while Bitline (BL) is connected to ground during RESET operation). Hence, even if heavy-ion strikes occur, the generated charge flow will have a direct path to the ground without accumulating at the MOSFET junction. Thus, the voltage across the RRAM device will not change.

In case of the half-selected cells, one of the RRAM device terminals is connected to a high potential while the other is left floating. Fig. 2.17 illustrates the bias voltages of half-selected cell whose RRAM device is at HRS. This cell shares the same BL as a fully-selected cell undergoing a SET operation. If heavy-ions strikes occur with enough

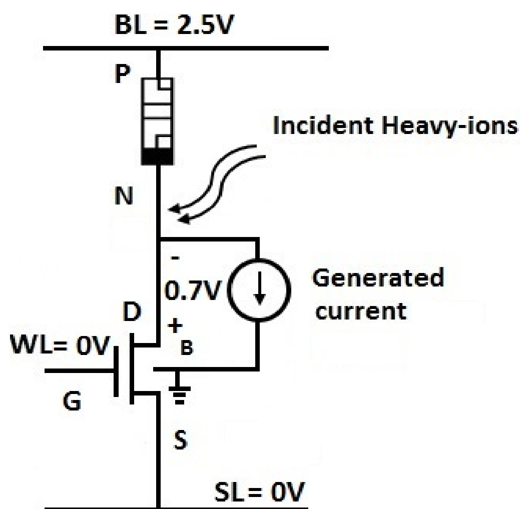

Figure 2.17: An example of SEE scenario in a half-selected 1T1R cell sharing the same BL bias voltage as fully-selected cell undergoing a SET operation. The current source in the figure models the SEE effect caused by the electron-hole pairs generated by the heavy-ions strikes [41].

energy at node 'N', the generated electron-hole pairs at the MOSFET junction will cause a current to flow through the substrate reducing the potential at node 'N' to -0.7V (the threshold voltage of the drain-substrate PN junction) [40, 41]. Hence, the RRAM device of the half-selected cell switches unintentionally to LRS since the voltage applied on it is higher than the SET operation threshold voltage (i.e., for HfO_x RRAM device, V_{SET} = 1.4 V while the applied voltage on the device is as high as 3.2V).

During the read operation, SEE can not occur. This is because, by pre-charging the BL to a relatively small voltage (i.e., $\approx$ 0.5V), even if heavy-ion strikes occur, the voltage drop across the RRAM device is not high enough to cause changes in its resistive state.

In case of MEUs, a refresh circuit can be used to sense the change in the RRAM state and restore it to its original value as discussed in chapter 4. As for SEEs, a new methodology is proposed in chapter 5 to detect and fix them in half-selected cells.

2.8 RRAM SPICE Models

There are many SPICE models in literature which describe the I-V characteristics of the various RRAM devices (TiO_x, TaO_x,...). It is worth mentioning that the subscript "x" in the oxide material name denotes the existence of oxygen vacancies in the oxide crystal structure. In the various experiments conducted in this work, the HfO_x RRAM device is used in our analysis for several reasons:

- **Validation of the SPICE model**: The SPICE models for the HfO_x RRAM device have been verified by various research groups by conducting multiple experiments on many fabricated devices. This includes the work done by the research groups at IMEC [36, 37, 38], Stanford [32, 102], and Vanderbilt University [40]. This increases our confidence in designing and simulating our proposed circuits using these models.
- **Completeness of the model**: In [33], the authors present the experimental results validating their SPICE model which describes the reliability soft-errors effects for the HfO_x RRAM device. In addition to this, the authors in [41] clarify, with the aid of experimental results, a methodology through which the radiation soft-errors in 1T1R can be simulated.
- **Open source model**: The fully calibrated SPICE models in [32, 33] for the HfO_x RRAM device are available for download at [103, 104].
- **Usage in multiple designs**: In addition to the existence of calibrated SPICE models, the HfO_x RRAM device is already integrated in multiple fabricated designs including low-power SRAM arrays [92], crossbar arrays [105], zero-leakage DFF [16].
- **Small device dimensions**: Due to the high bandgap of the HfO_2 oxide, the oxide thickness of the HfO_x RRAM device is significantly reduced compared to other RRAM devices (3 nm in HfO_x RRAM device compared to 20 nm in case of TiO_x RRAM device).

The SPICE models in [32, 33] describe the basic I-V characteristics of the HfO_x RRAM device. The authors in [33] illustrate how the SPICE model fits the I-V curve generated

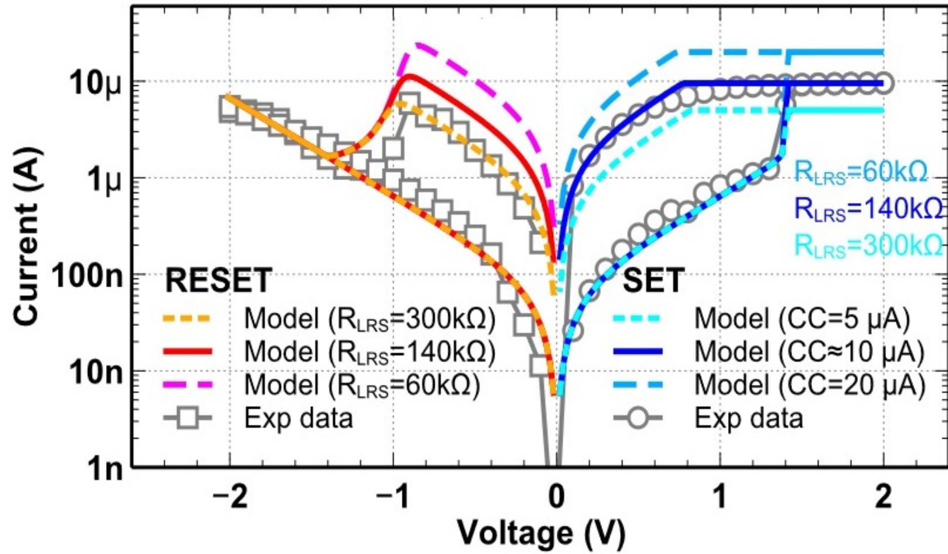

Figure 2.18: I-V characteristics curve for the RRAM HfO_x SPICE model described in [33]. I-V characteristics curve generated from the model fits that resulting from the experimental data. The figure also shows how the LRS and HRS of the device changes with the various programming conditions expressed by the different compliance current levels. Permission granted to use the figure.

from the fabricated device as shown in fig. 2.18. To represent the reliability soft-errors effects, discussed in section 2.7.1, the authors in [33] modified the original I-V equations in [32] by:

1. Introducing a positive feedback between the change in the temperature of the conductive filaments of the RRAM device and its I-V equations. This is basically to describe the effect of applied field on increasing the temperature of the filaments which by consequence increases the oxygen vacancies diffusion out of the filament containment as explained in section 2.7.1.

2. To describe the oxygen vacancies diffusion due to the difference in their concentration in the conductive filaments and in the oxide material, the model in [33] reduces the activation energy for oxygen vacancies recombination process in comparison to that of the generation operation.

The exact model equations are explained in more details in chapter 6. The authors in [33] illustrate how the SPICE model can accurately track the deviation in the LRS of RRAM

device resulting from oxygen vacancies diffusion for different operating temperatures as shown in fig. 2.19.

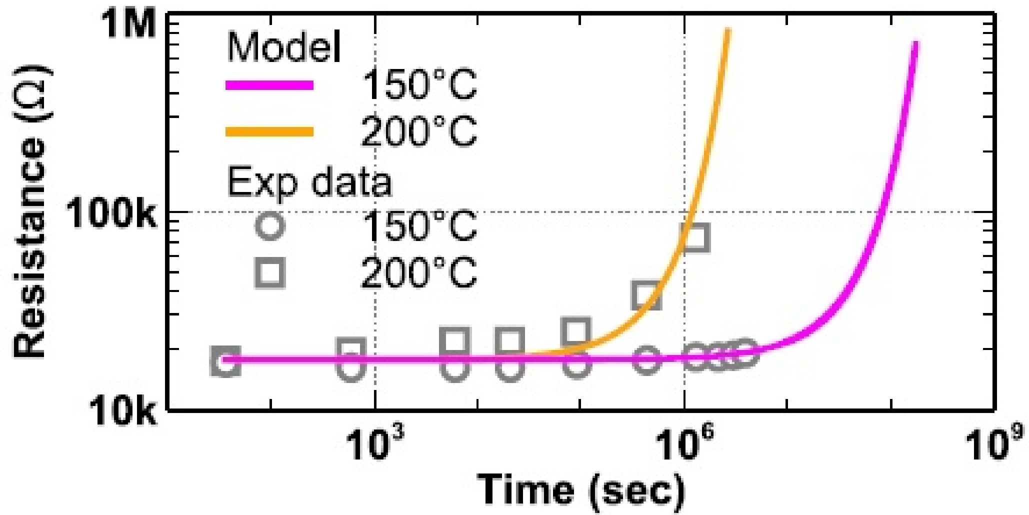

Figure 2.19: Modeling of RRAM reliability soft-errors for HfO_x RRAM device. Curves generated from the SPICE model in [33] fits the experimental data obtained by various research groups for different operating temperatures. Permission granted to use the figure.

As for the radiation effects, the authors in [41] provided a technique to include the radiation soft-errors in the RRAM 1T1R arrays. Fig. 2.17 in section 2.7.2 illustrates the proposed methodology in [41]. The authors suggested adding a current source whose amplitude depends on the energy of incident charged particles. The authors in [41] prove the validity of their proposed technique by showing that the computed current using the suggested SPICE methodology matches that obtained from the experimental data for various energy levels for the incident charged particles.

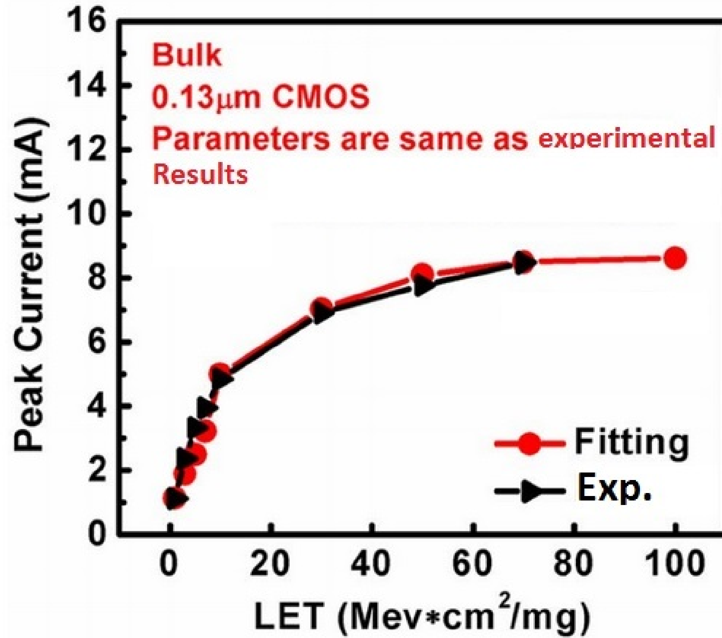

Figure 2.20: Simulation and experimental results for the current generated due to the highly energetic incident charged particles on the junction of the access transistor of 1T1R RRAM array [41]. The figure shows the matching between the computed current levels and those obtained from the experimental data for different energy levels for the incident charged particles. Permission granted to use the figure.

Since the SPICE models in [33, 41] have been experimentally verified, in this thesis, we provide multiple circuit and system solutions using simulation results based of those models. Experimental verification of the proposed designs is required for the future work as discussed in section 7. Moreover, although our studies are conducted using the HfO_x RRAM device, all the concepts and techniques discussed in the subsequent chapters are applicable to any other RRAM device.

2.9 Organization of the Research Work in this Thesis

With the increasing interest in RRAM devices and their widely usage in many applications as described in section 2.6, most of the designs ignore any degradations in RRAM arrays performance due to RRAM soft-errors. There has not been much systematic study on

the impact of RRAM soft-errors on the circuits and systems design in which the device is incorporated. Our focus in this work is to bring the attention to the need to take into account such imperfections during the design stage. In particular, we provide techniques to evaluate and reduce the effect of RRAM reliability and radiations soft-errors. Deploying the proposed methodologies eases the integration of the RRAM devices in various products including GPU and neuromorhic systems which can run machine learning applications.

We start first in chapter 3 by providing a new methodology for integrating the RRAM device in SRAM-based designs (i.e., GPU) with the aim to lower the power required to store/restore the data of SRAM cells on RRAM devices. This can reduce the temperature generated in the conductive filaments of the device exponentially and hence increase its resilience to reliability soft-errors as described in chapter 3.

Then, in chapters 4 and 5, a detailed analysis is provided on how to reduce the effect of reliability and radiation soft-errors found in the 1T1R arrays used in the design of neuromorphic systems.

After this, a novel methodology is discussed in chapter 6 on how to estimate the impact of RRAM soft-errors on the performance of RRAM-based neuromoprhic systems. In addition to this, in chapter 6, a system level solution for detecting and fixing the RRAM soft-errors is provided.

Finally, in chapter 7, the various discussions in the chapters of this thesis are concluded and a set of recommended directions for the future work is provided.

Chapter 3

8T1R: Optimizing the Design of the RRAM-based Non-Volatile SRAM Design to Reduce the Effect of RRAM Soft-Errors

In this chapter, a new design for the RRAM-based NV-SRAM cell is discussed. NV-SRAM cells are suggested to be incorporated in the SRAM-dominated designs, such as GPU, used currently to run machine learning applications to decrease their leakage power consumption. Another main advantage for our proposed NV-SRAM cell is minimizing the energy required for storing and retrieving data from RRAM devices. This, by consequence, reduces the effect of RRAM reliability soft-errors due to the decrease in the generated heat inside the RRAM conductive filaments. In section 3.1, a brief introduction about the previous work on integrating the RRAM device to SRAM cells is provided. Then, in section 3.2, the structure and operation of the new 8T1R NV-SRAM is discussed. The chapter is concluded in section 3.3 by presenting the simulation results for the newly suggested 8T1R cell which is done using the SPICE model in [33]. In addition to this, in section 3.3, the performance of 8T1R cell is compared with previously reported RRAM-based NV-SRAM cells to demonstrate its major advantages. The main contribution from this work is suggesting a new NV-SRAM cell which: a) has minimal impact on the basic SRAM read and write operations, and b) takes into account the RRAM reliability soft-errors.

3.1 Introduction

With mobile chips and many other power constrained integrated circuits applications, the demand for reducing the chip power becomes a major challenge. On one hand, device leakage has increased significantly due to continuous technology scaling. On the other hand, practical requirements, such as extending battery life, have been prioritized in recent years. SRAM cells are used extensively in many chip designs to provide on-chip storage. Despite their high-speed read/write operations and their various low-supply voltage (VDD) designs (e.g., 7T [106], 8T [107], and 10T [108] cells), the SRAM cells suffer from high leakage power. DVS [9, 10, 11] has been a popular approach to reduce leakage power by adjusting the transistor gate-source and substrate-source voltages depending on whether the SRAM cell is in active or stand-by mode. Power gating [12, 13] is another widely used approach to reduce leakage power by adjusting the supply voltage level during the inactive periods. Yet, the power saving from these techniques is diminishing with the latest technology nodes due to non-scalable leakage power components.

NVMs are suggested to be integrated with the SRAM cells to further suppress the stand-by power by switching off the power supply of the “less-frequently” used SRAM blocks without losing their data. SRAM data is first stored into the NVM device before switching off the power of SRAM cell. This configuration consumes much less power and area compared to using high threshold voltage (High-Vth) retention latches which rely on having “always-on” flip-flops to save the data of inactive SRAM blocks before switching off their power supply [109]. When access to “previously-inactive” SRAM cells is required later on, the data saved in NVM cells are written back to the storage nodes of the SRAM through a restore operation. Such SRAM cells are often referred to in literature as NV-SRAM. Several types of NVM devices (e.g., MRAM, Phase Change Memory (PCM), etc.) are proposed to be used. Yet, due to its technology advantages discussed in section 2.5, incorporating RRAM devices in NV-SRAM is preferred.

In previous works, such as those in [17, 93, 92, 94], the RRAM device has been integrated with SRAM cells but each of those designs either introduces significant degradation in the SRAM read/write operations, or consumes a large amount of energy to store/restore data to/from the RRAM device as listed in table 3.2 in section 3.3.
In [17], two RRAM devices are integrated with the conventional 6T SRAM cell at the two storage nodes as shown in fig. 3.1. The control line ‘CL’ is added to provide the needed programming voltage on the second terminal of RRAM devices. In other words, to store a ‘0’ on the RRAM device, a positive voltage is applied on the BL, while ‘CL’ is set to VDD. This way, the RRAM on Bitline Bar (BLB) side changes its state, while the one on BL side remains as is.

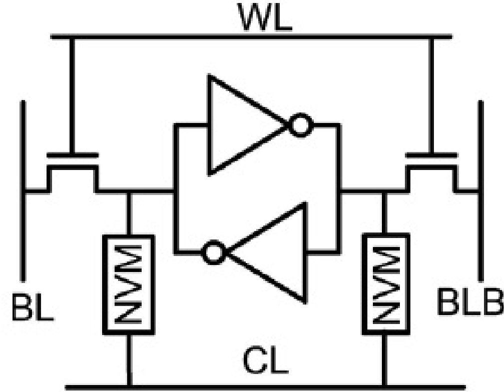

Figure 3.1: The 6T2R SRAM cell [17] in which the two RRAM devices are connected together through the extra control line 'CL'. Permission granted to use the figure.

This design has many drawbacks: a) the write operation speed is lowered due to the existence of extra resistors at the storage nodes; b) the circuit stability is compromised due to the existence of a path between the storage nodes of the SRAM cell; c) the leakage power is large since there is no isolation between the RRAM devices and the SRAM saved data; d) this design does not account for the RRAM limited endurance since a large current is passing through RRAM devices with each write operation; and e) the 'CL' line increases the SRAM cell area.
To tackle these problems, the authors in [93] suggested the 8T2R cell structure shown in fig. 3.2.

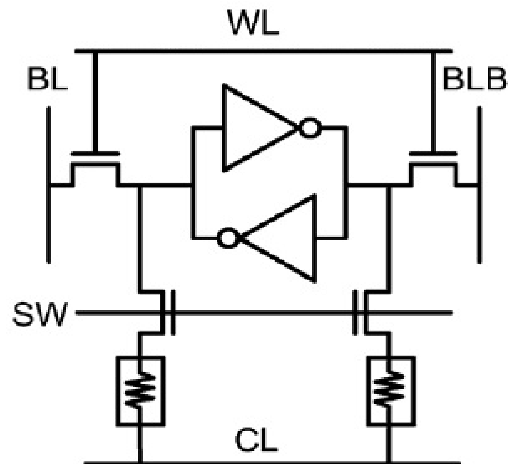

Figure 3.2: The 8T2R SRAM cell proposed in [93]. Two control signals, 'SW' and 'CL', are used to program the resistive state of RRAM devices. Permission granted to use the figure.

The cell isolates the RRAM device by adding two transistors which are disabled during the read/write operations. To store values, the signal 'SW' must be first set to VDD and the proper programming voltage is applied on 'CL'. The main disadvantage of this design is that it requires the addition of two extra control lines, 'SW' and 'CL', which requires the addition of more control circuitry and increases the cell and chip area. Also, as discussed in section 3.3.2, the restore delay and energy consumption are quite high due to the need to re-program the states of both RRAM devices to LRS once the restore operation is completed.
A similar 8T2R cell was proposed in [92]. This circuit has the schematic shown in fig. 3.3.

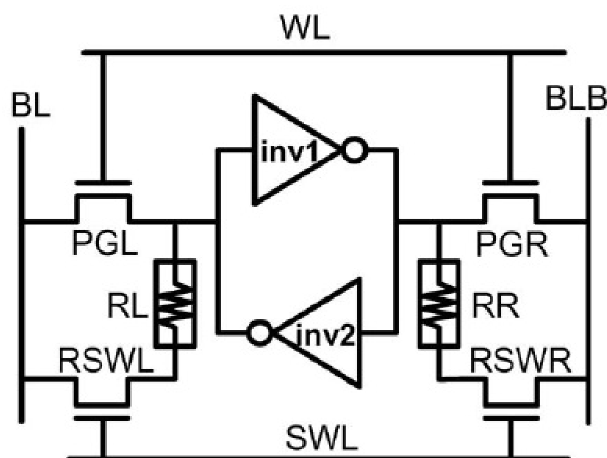

Figure 3.3: The Rnv8T SRAM cell proposed in [92]. Other than being used in the store/restore operations, the transistors "RSWL" and "RSWR" are used during the write operation to enhance the noise margin. Permission granted to use the figure.

The difference between the design in [92] and that in [93] is that the extra added transistors, "RSWL" and "RSWL", act as extra drivers for the write operation which increases the write margin and by consequence allows resizing the SRAM transistors in favor of the read operation (i.e., remove the constraint on the ratio between the sizes of P-type Metal Oxide Semiconductor (PMOS) pull-up and N-type Metal Oxide Semiconductor (NMOS) pass transistors). This actually allowed to decrease the SRAM VDD_{min} parameter to below 600mv (i.e., 450mv).
In more detailed words, the SRAM proposed in [92] works as follows:

- **Read Operation**: This operation runs exactly as in the normal 6T SRAM cell with the addition of that the line 'SWL' is connected to ground to deactivate disturbing the value saved on the RRAM.

- **Write Operation**: This operation is run as in the conventional 6T-SRAM cell but with the exception of activating the "RSWR" and "RSWL" transistors. This actually provides another path for the write operation.

- **Store operation**: This operation saves the data the SRAM storage nodes cell on the RRAM devices. During the write operation, the RRAM device is actually used as a resistor without being concerned about the value being written to it. The store procedure is run in two steps: First, the BL and BLB are set to high voltage ($\approx$ 3V)

needed to program the RRAM devices used in the design (i.e., HfO_x RRAM) by switching its state from HRS to LRS. This enables the part saving logic '0' on the RRAM to work properly. The second part of the store operation is done by setting the BL and BLB lines to 0 V, while raising the VDD signal of the SRAM cell (i.e., Cell VDD (CVDD)) to high voltage. This enables saving logic '1' (i.e., HRS) on the RRAM device.

- **Restore operation**: This operation is related to the procedure followed to retrieve the data saved on the RRAM device after restoring the SRAM power signal. During this step, only 'SWL' is turned ON and the BL and BLB lines are pulled to ground. Then the CVDD is raised to normal VDD. Accordingly, the PMOS transistors of the back to back inverters provide current to pass through the RRAM devices. If the RRAM device is at LRS, then the discharge current is larger which will turn ON the NMOS transistor of the other inverter stage. This NMOS transistor will discharge the output node of its inverter to logic '0' which increases the discharging current passing through the LRS RRAM. This positive feedback loop keeps on going until the other node (Q/QB) charges to VDD.

Although this design has decreased the VDD_{min} of the SRAM cell, it has a major disadvantage of not considering the RRAM endurance issue and its reliability soft-errors. This is in addition to increasing the energy and delay of the store operation as described in section 3.3.2.
The same authors provided another NV-SRAM design which consists of nine transistors and two RRAM cells (i.e., 9T2R cell) in [94]. The schematic of this cell looks like the normal 7T SRAM cell is shown in fig. 3.4. The authors aim to decrease even more the VDD_{min} of the 9T2R cell by decoupling completely the read and write operations. This is done by having separate read bitline (RBL) and wordline (RWL). Accordingly, the sizes of MOS transistors used in the SRAM design do not even need to take into account the read operation which enables lowering the VDD_{min}. This comes at the expense of increasing the SRAM cell area due to the extra transistor "RPG" and control lines 'RBL' and 'RWL'. This is in addition to complicating the read operation by making it single-ended which imposes more constraints on the sense-amplifier (SA) used in the read circuitry to overcome the effect of supply noise. The single-ended SA also limits the number of cells/column due to the need to increase the sensing margin.

In this chapter, we are proposing a novel NV-SRAM design, which reduces the energy required for the store/restore operations without impacting the basic read/write processes of the SRAM cell. By reducing the energy required for writing and reading the data stored on the RRAM device, the current passing through the RRAM is decreased. This reduces

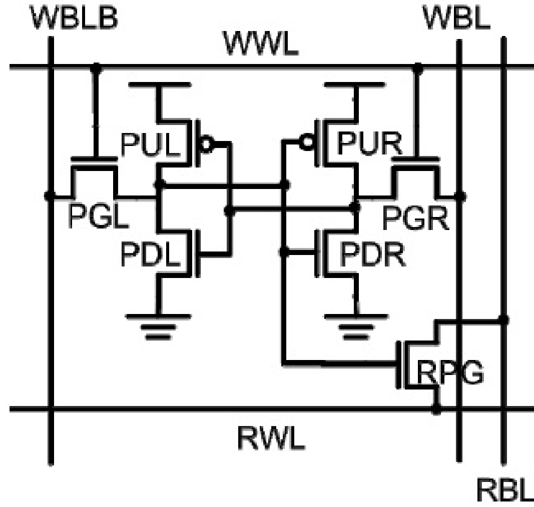

Figure 3.4: The 7T SRAM cell [94]. The transistor "RPG" is the NMOS device used only during the read operation and it is connected to dedicated control lines for read operation 'RBL' and 'RWL'. The same structure is used for the cell 9T2R in [94]. Permission granted to use the figure.

the generated heat inside the conductive filaments of the RRAM device which is the main contributor to its reliability soft-errors as discussed in chapter 4.

3.2 Proposed 8T1R cell

Fig. 3.5 shows the schematic of our proposed 8T1R NV-SRAM cell. The nodes named 'P' and 'N' in fig. 3.5 denote the anode (top) and the cathode (bottom) electrodes of the RRAM device, respectively. In comparison to the previously proposed RRAM-based NV-SRAM designs, the 8T1R cell uses only one RRAM device to save the SRAM data which simplifies the store operation. Moreover, the 8T1R cell does not add extra capacitances to the BL/BLB lines, unlike other RRAM-based NV-SRAM designs (e.g., the Resistive Nonvolatile 8T2R (Rnv8T) cell [92]), which improves the speed and noise immunity of the read operation. In the subsequent subsections, we explain each of the 8T1R cell operations.

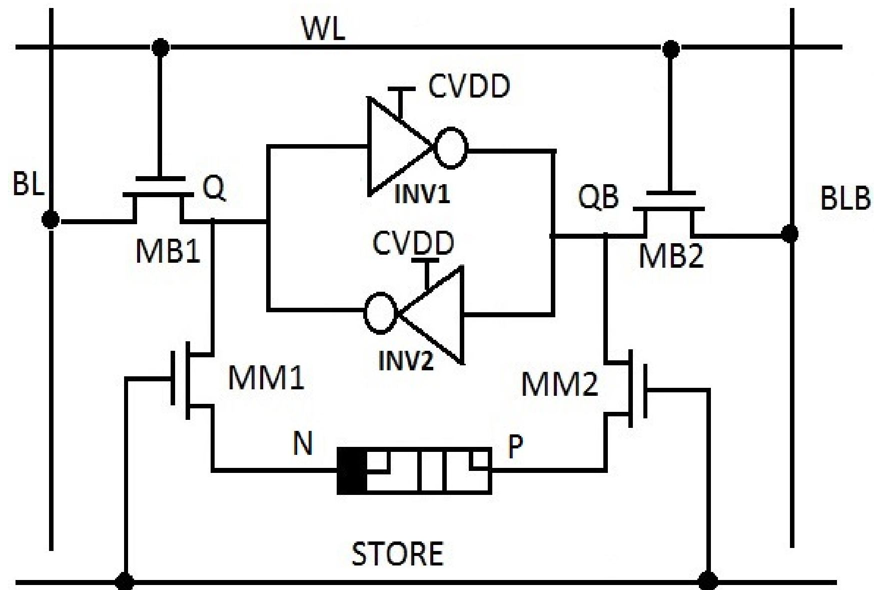

Figure 3.5: The structure of 8T1R NV-SRAM cell.

3.2.1 Read/Write Operation

During the read/write operation, the 'STORE' signal is grounded disconnecting the RRAM cell and its access transistors (i.e., MM1 and MM2) from the storage nodes (i.e., Q and QB). Accordingly, the 8T1R NV-SRAM cell works similarly to the conventional 6T SRAM design in this mode of operation. The only difference is the additional source/drain capacitances at the storage nodes 'Q' and 'QB' coming from the access transistors of the RRAM device. This causes a slight increase in the energy and delay of the 8T1R read/write operations compared to those of the 6T SRAM design as discussed in section 3.3.1.

3.2.2 Store Operation

To save the data on the RRAM device before cutting off the power if "less-frequently" SRAM blocks, a sequence of steps is required:

1. Wordline (WL) is grounded and the access transistors "MB1" and "MB2" disconnect the SRAM storage nodes 'Q' and 'QB' from the BL/BLB lines.

2. CVDD is raised to VDDH which is determined by the SET/RESET voltages of the RRAM device used.

3. Connect the 'STORE' signal to VDDH to allow the RRAM cell to change its state depending on the SRAM data.

For example, let us assume that the storage nodes 'QB' and 'Q' are at logic '1' (i.e., VDD) and '0' (i.e., 0 V), respectively, before starting the store operation. Raising CVDD to VDDH causes the voltage of 'Q' to be at VDDH, while that of the node 'QB' remains grounded. Hence, after subtracting the voltage drop across "MM1" and "MM2" transistors, the net voltage applied on the RRAM device is high enough to change its state to LRS. Oppositely, if the nodes 'QB' and 'Q' are initially at logic '0' and '1', respectively, the net voltage drop across the RRAM pushes its resistance to HRS. Fig. 3.6 shows the waveforms explaining the 8T1R NV-SRAM store operation when the nodes 'QB' and 'Q' are at logic '1' and '0', respectively (i.e., RRAM changing its state to LRS).

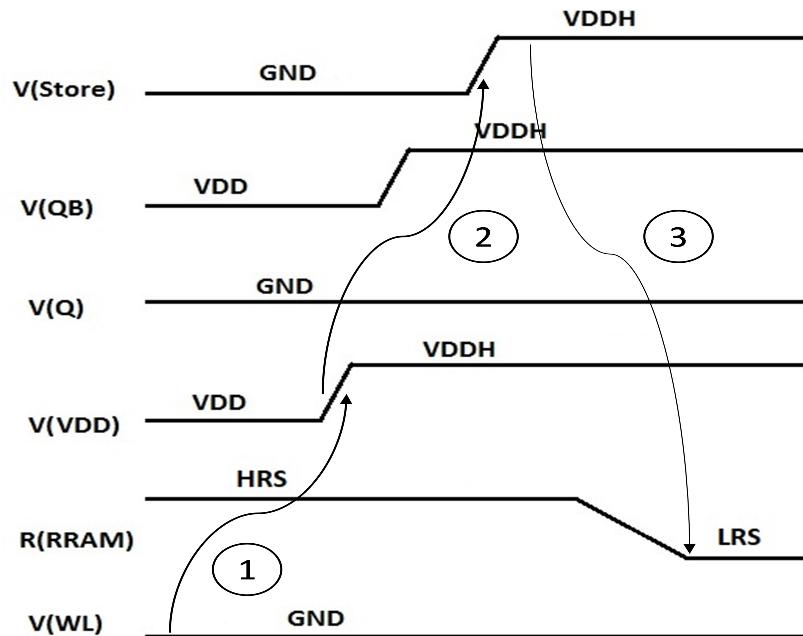

Figure 3.6: The store operation waveform for the case when QB is at logic '1' and Q is at logic '0'. The numbers in circles correspond to the sequence of store operation steps. In our experiments, VDD = 1.1 V and VDDH = 2.0 V.

Fig. 3.7 illustrates the waveforms generated from running SPICE simulations for the same store operation. The number sequence in fig. 3.6 is consistent with that in fig. 3.7.

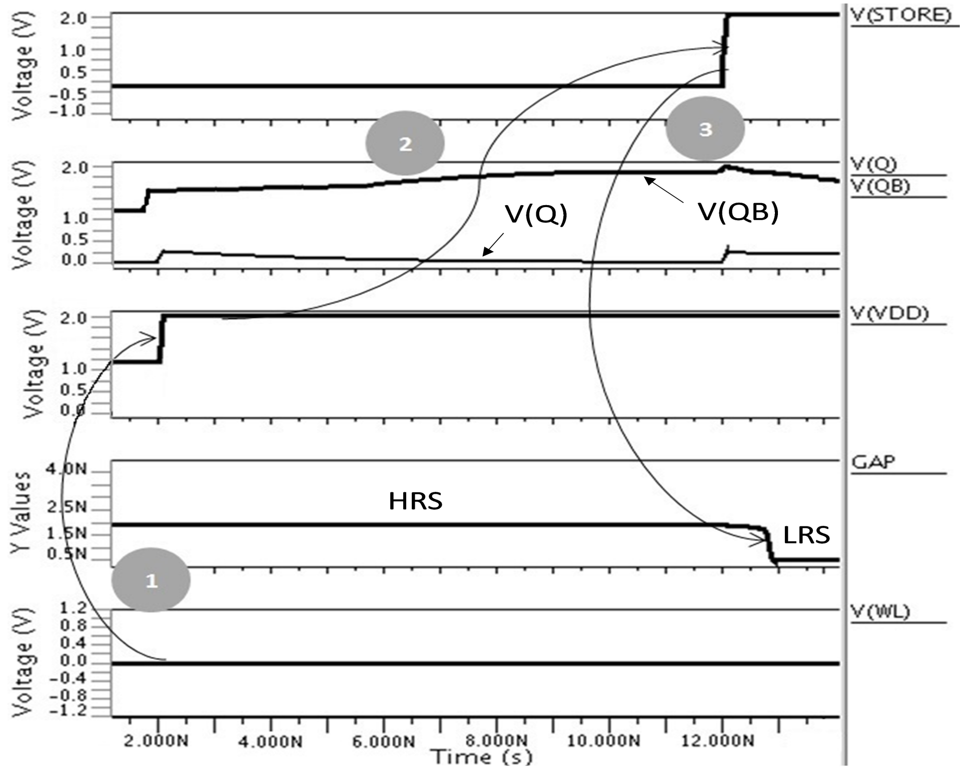

Figure 3.7: Store operation waveforms generated from running SPICE for the case when node 'QB' and 'Q' are at logic '1' and '0', respectively. RRAM state is programmed to LRS in this scenario as indicated by the decrease in "GAP" value which describes the distance separating the top electrode from the tip of conductive filaments.

The "GAP" graph in fig. 3.7 describes the state of gap distance separating the top electrode from the conductive filaments (i.e., variable "X" discussed in section 2.1). The smaller the gap distance, the easier for current to pass through the oxide material of the RRAM device (i.e., LRS). Since the node 'QB' is storing logic '1', when CVDD is connected to VDDH (i.e., 2 V in our experiment), the voltage of node 'QB' is set to VDDH before enabling the 'STORE' signal. Hence, when the 'STORE' signal is activated, the gap distance of the RRAM device switches to a low value (i.e., 0.2 nm) indicating that the device has switched to its LRS. The reason why 'STORE' signal is connected to VDDH instead of VDD (i.e., 2 V instead of 1.1 V in our experiment) is to speed up the store operation by increasing

the voltage drop across the RRAM device. This, by consequence, reduces the possibility of errors that can result from sudden power loss during the store operation. Oppositely, fig. 3.8 shows that the RRAM state is programmed to HRS (i.e., the "GAP" value increases to 1.2 nm when 'STORE' signal is activated) in case if logic '1' is saved on node 'Q'.

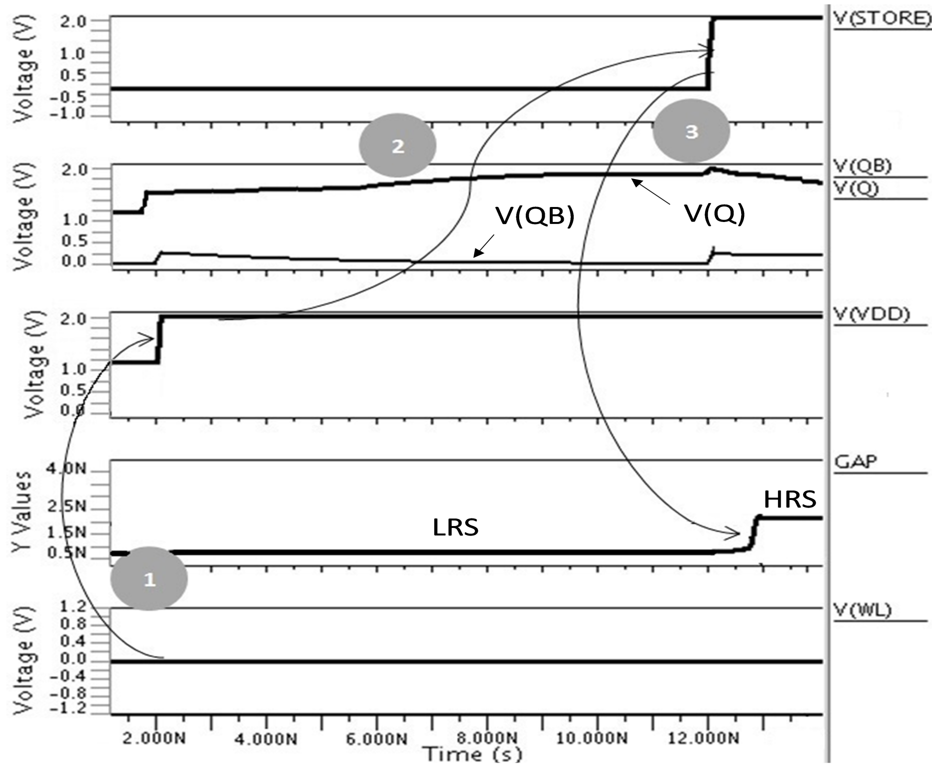

Figure 3.8: Store operation waveforms generated from running SPICE for the case when node 'QB' and 'Q' are storing logic '0' and '1', respectively. RRAM state is programmed to HRS in this case as indicated by the increase in "GAP" value causing less current to pass between the RRAM device terminals.

One point worth mentioning is related to the 8T1R reliability during the store operation. Given that a high voltage is applied on the access transistor, this can result in oxide breakdown. However, this should not have a big impact on our 8T1R cell due to:

- The duration of the store operation is short (i.e., few nanoseconds as shown in fig. 3.7 and fig. 3.8) which limits the stress effect on the MOS oxide material.

- According to the data sheet of the 40 nm Process Design Kit (PDK) used in our simulation work, there are Input/Output transistors which can handle voltage levels up to 2.5 V due to their thick gate oxide material and special fabrication process. In more advanced technologies, other access devices can be used which can operate under high voltage conditions to deliver the high current drive required on the RRAM device terminals. This includes the Gate-All-Around (GAA) transistor [110, 111] and selector devices described in [73, 74]. Those devices can further decrease the footprint of SRAM memory cells however the research of incorporating them is still ongoing [71, 72].

In comparison to the other RRAM-based NV-SRAM cells, the 8T1R store operation is simpler and faster as summarized in section 3.3.2.

3.2.3 Restore Operation

The restore operation retrieves the data stored into the RRAM cell to the SRAM storage nodes when it is reconnected to the power supply signals. To complete the restore operation, it is required to:

1. Precharge the BL to VDD, while BLB is connected to ground.
2. Activate the access transistors "MM1", "MM2", "MB1", and "MB2" by connecting the WL and 'STORE' signals to VDD.
3. Reconnecting CVDD signals of the SRAM block to VDD with the CVDD signal of the feed-forward inverter (i.e., "INV1" in fig. 3.5) reactivated before that of the feedback inverter (i.e., "INV2" in fig. 3.5).

The basic concept behind this sequence of operations is to precharge the storage node 'Q' to a voltage level depending on the RRAM resistive state. If the RRAM device is at its HRS, then voltage of the node 'Q' is higher than the threshold voltage of the NMOS transistors (i.e., $V_{th,NMOS}$). Accordingly, when the power signal of "INV1" is reactivated, the current coming from the supply through the PMOS transistor, tries to charge the node 'QB' to VDD. Yet, since the NMOS transistor of "INV1" is ON, the charges, previously built up on the 'QB' node, discharge to ground keeping the voltage of node 'QB' close to 0 V. Oppositely, if the RRAM cell is at its LRS , the voltage of the node 'Q' is less than $V_{th,NMOS}$ and the current coming from the power signal of "INV1" charges the node 'QB' to a high voltage (i.e., higher than $V_{th,NMOS}$ of the feedback inverter "INV2"). Hence,

when “INV2” is reconnected to the power signal, based on the charge collected on ‘QB’ by “INV1”, the SRAM data is correctly restored by having the nodes ‘QB’ and ‘Q’ at logic ‘1’ and ‘0’, respectively. Fig. 3.9 shows the waveforms describing the restore operation sequence when the RRAM device is at its LRS.

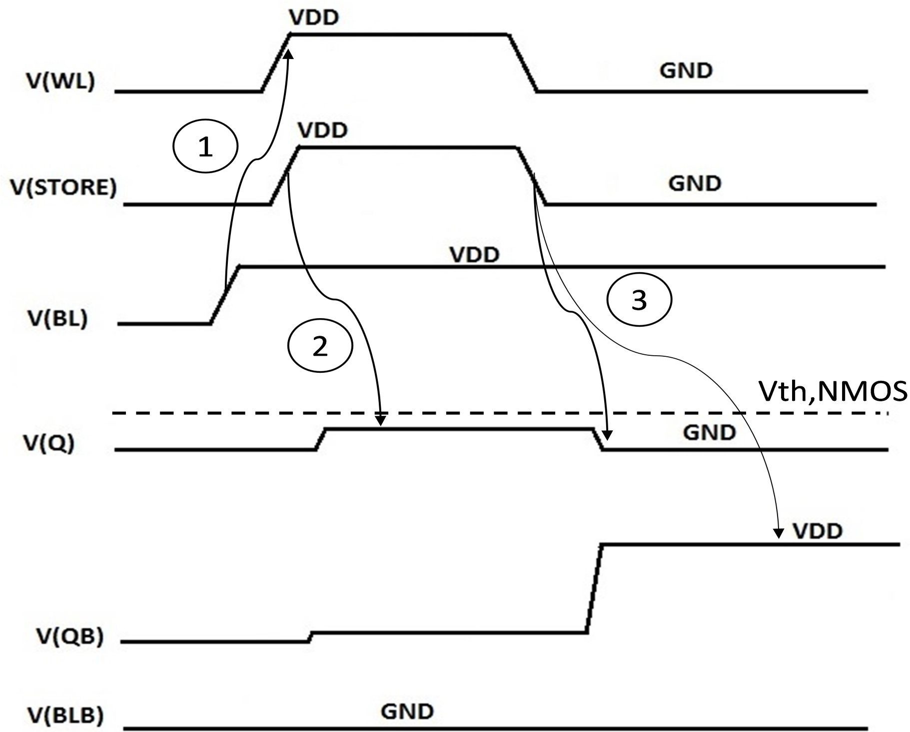

Figure 3.9: Illustration of restore operation waveforms for the case when RRAM is at LRS. In this scenario, node ‘Q’ is precharged to a voltage less than Vth of NMOS device causing the node ‘QB’ to charge to high voltage when power signal is reactivated. At the end of restore operation, the voltage of nodes ‘QB’ and ‘Q’ is set to VDD and ground, respectively.

In order to guarantee a delay between the reactivation of power supply signals of “INV1” and “INV2”, a delay chain is added between a pair of PMOS transistors connecting the power supply to the inverter storage nodes of each SRAM cell as shown in fig. 3.10.

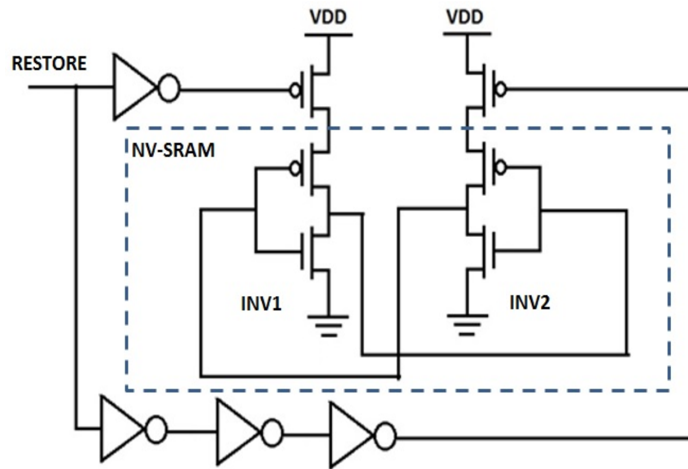

Figure 3.10: The restore power supply circuit. The delay chain is added to guarantee that the power supply of "INV1" is activated before that of "INV2".

The 'RESTORE' signal in fig. 3.10 is the control pulse issued by the memory controller to initiate the restore operation. Fig. 3.11 illustrates the waveforms generated from running SPICE simulations for the restore operation in the case when the RRAM device is at LRS.

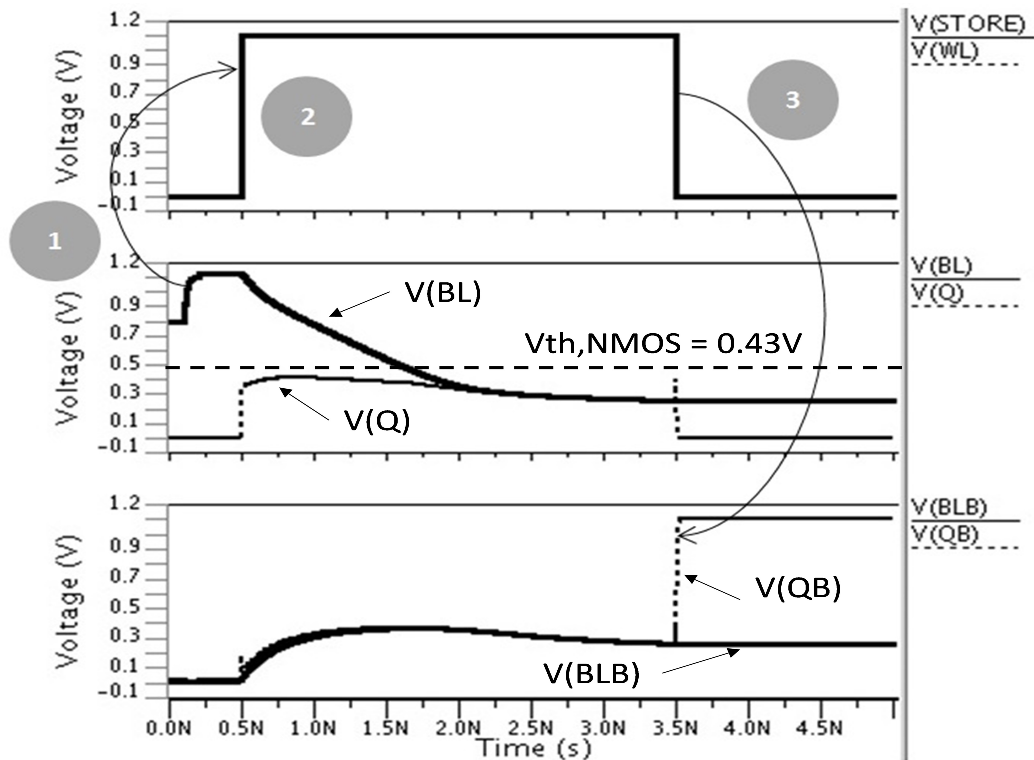

Figure 3.11: Restore operation waveforms generated from SPICE simulation for the case when RRAM is at LRS. In this scenario, at the end of restore operation, voltage of node 'Q' is at ground while that of node 'QB' is at VDD.

The sequence of numbers in fig. 3.11 are consistent with the one in fig. 3.9. Since the RRAM is at LRS, the BL voltage, originally pre-charged to VDD (i.e., 1.1 V in our experiment), discharges quickly to a low voltage (i.e., 0.3 V) during the period when signals 'WL' and 'STORE' are activated. Accordingly, when the SRAM power is restored (by restoring the power of "INV1" first as explained in fig. 3.9), the positive feedback of the back-to-back inverters causes the voltage of node 'QB' to be set to VDD, while that of node 'Q' is pulled to ground.

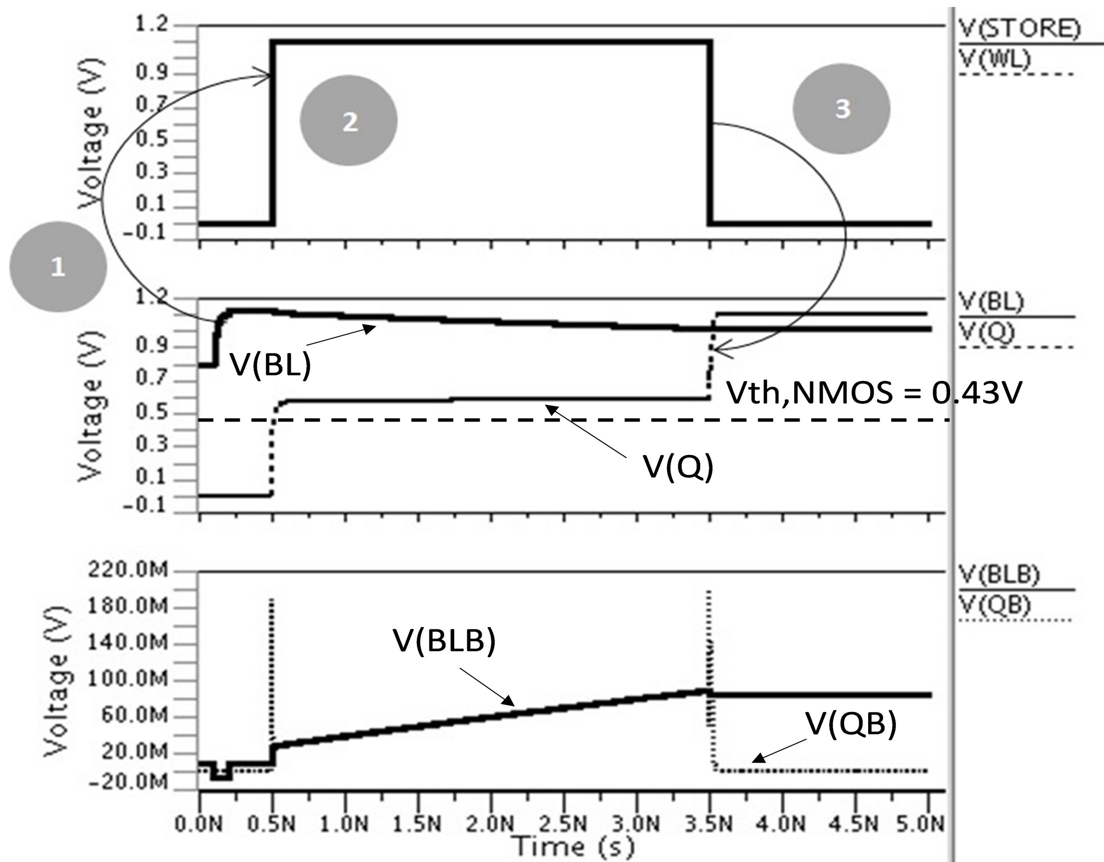

Figure 3.12: Restore operation waveforms generated from running SPICE for the case when RRAM device is at HRS. In this case, at the end of restore operation, the voltage of node 'Q' and 'QB' are set to VDD and ground, respectively.

Oppositely, fig. 3.12 demonstrates that, in case if RRAM is at HRS, the voltage of node 'Q', before reconnecting the power signal of the SRAM cell, will be around 0.56 V which is already above the threshold voltage required to turn on the NMOS transistor of "INV1" (i.e., 0.43 V for the TSMC 40 nm technology used). By consequence, when the SRAM is reactivated, the voltage of node 'Q' is set to VDD while that of node 'QB' is at 0 V. The spikes in fig. 3.11 and fig. 3.12 are caused by the sudden current change in the circuit due to the switching of 'WL' and 'STORE' signals.

3.3 Simulation Results

Fig. 3.13 illustrates the block diagram describing the flow of our simulation runs and the models used. All the circuit simulations are done using Eldo simulator from Mentor

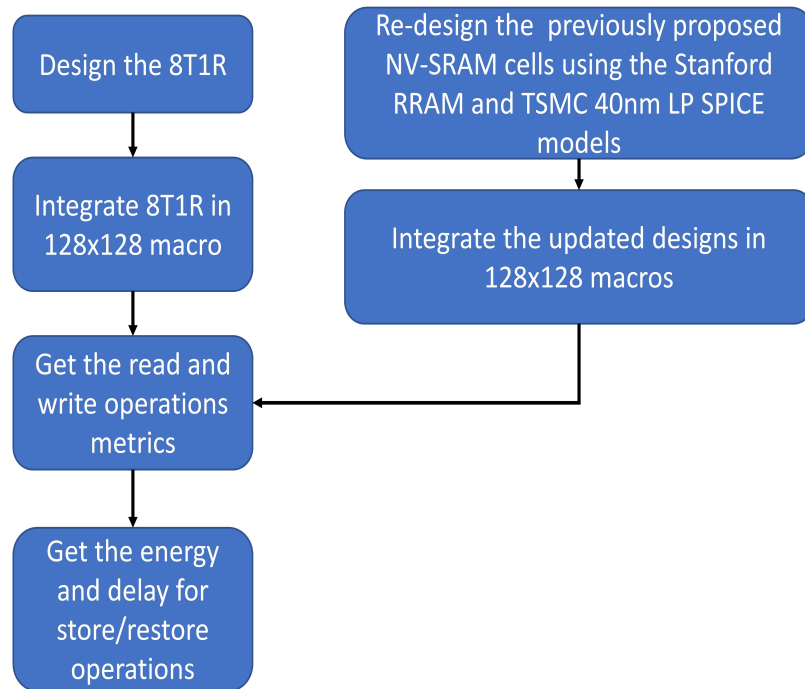

Figure 3.13: Block diagram for the simulation runs conducted to evaluate the performance of various modes of operations (i.e., read, write, store, and restore operations) for the 8T1R NV-SRAM cell.

Graphics [112] with a TSMC Low Power (LP) 40 nm CMOS model. As for the RRAM model, the Stanford model described in [32] has been used. The first step in the block diagram in fig. 3.13 is to run various simulations on the 8T1R cell to verify the correctness of its output in the different modes of operation (i.e., read, write, store, and restore operations). After this, a 8T1R 128x128 array is formed and simulated in SPICE. Various performance metrics for the read, write, store, and restore operations are computed. The

obtained simulation results are then compared with the other 128x128 RRAM-based NV-SRAM proposed in literature. The width and operating voltages of the transistors of the previously reported NV-SRAM cells are adjusted to fit TSMC 40 nm LP CMOS model. Moreover, the Standford RRAM model in [32] is used in the simulation runs of those cells. Table 3.1 lists the transistor sizes of the 8T1R NV-SRAM cell. The transistor names listed in table 3.1 are the same as those shown in fig. 3.5. MN_{inv} and MP_{inv} are the transistors used in the cross-coupled inverters of the SRAM cell.

Table 3.1: Transistor sizes of the 8T1R NV-SRAM cell

Transistor Name	**W(nm)/L(nm) value**
MN_{inv}	315/40
MP_{inv}	173/40
$MB1$, $MB2$	275/40
$MM1$, $MM2$	275/40

The same transistor sizes and RRAM device model [32] of the 8T1R NV-SRAM cell are used to compare its performance versus the other RRAM-based NV-SRAM cells discussed in literature. In the next subsections, we describe the simulation results for each of the operations of a 128x128 array made of our proposed 8T1R cell. The 128x128 array is a typical size of the SRAM bank as described in [113].

3.3.1 Read and Write Operations

Table 3.2 summarizes the simulation results for the read and write operations metrics for the RRAM-based SRAM cells reported in the literature. Items in red in table 3.2 mark the read/write metric whose value is worse in the RRAM-based NV-SRAM cell when compared with its value in the conventional 6T SRAM design. Table 3.2 demonstrates that only the 8T1R and 8T2R cells have minimal impact on the various metrics for the read and write operations of the SRAM. However, for the 8T2R cell, it has other disadvantages which are:

- The structure of 8T2R cell introduces an extra control line in SRAM cell which increases its area as discussed in section 3.1.
- The store and restore energy consumption and delay for the 8T2R are much higher than those for 8T1R cell as explained in details in section 3.3.2.

Table 3.2: Comparison results for the read and write operations of the different RRAM-based NV-SRAM 128x128 arrays

Parameter Name	**6T2R** [17]	**8T2R** [93]	**Rnv8T** [92]	**9T2R** [94]	**7T2R** [114]	**8T1R (new cell)**	**6T SRAM**
Non-volatile feature	Yes	Yes	Yes	Yes	Yes	Yes	No
Write Energy (fJ)	850	74.7	76.5	76.7	139	74.8	74
Write latency (ps)	27.6	26.4	24.7	30.3	33.8	26.3	25.6
Read Energy (fJ)	1190	1.14	1.20	0.356	55.3	1.16	1.1
Read latency (ps)	35	50.6	93.1	26.9	72.9	50.6	50
Read Dynamic Noise Margin (RDNM) (V)	0.0038	0.224	0.204	0.411	0.193	0.224	0.224
Write Dynamic Noise Margin (WDNM) (V)	0.258	0.337	0.35	0.349	0.338	0.337	0.34
Leakage currents (μA)	18.1	0.0364	0.0383	0.0934	1.22	0.0365	0.036

To calculate the noise immunity of the different NV-SRAM designs in table 3.2, we use the Dynamic Noise Margin (DNM) which accounts for the noise effect on the SRAM storage nodes 'Q' and 'QB' for the pulse duration of the read/write operation [113]. The method, described in [115, 116], is used to compute the noise voltage levels at the storage nodes 'Q' and 'QB' which can prevent data from being correctly read/written to SRAM cell. Fig. 3.14 illustrates how the noise sources are attached to the storage nodes 'Q' and 'QB' [115, 116].

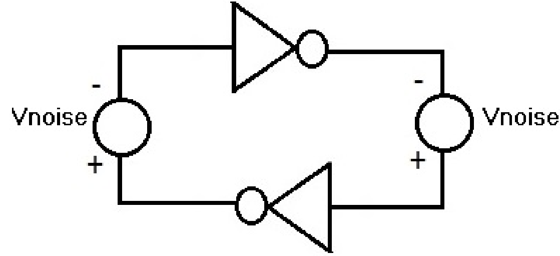

Figure 3.14: Modeling of the noise sources to compute the read/write noise margins of back to back inverters used in CMOS SRAM and DFF designs [115, 116]. All possible combinations of polarities for the noise sources in the figure have to be tried and the resulting minimum value of noise signal voltage is considered as the noise margin of SRAM cell. The same concept is applied to the NV-SRAM cells studied in this work.

Fig. 3.15 shows the simulation results for calculating the WDNM of 8T1R cell. From fig. 3.15 and using TCL scripts to parse the waveform database generated from running Eldo, it is found that the WDNM of 8T1R cell is 337 mV.
We can divide the listed results in table 3.2 into three categories:

Comparison with the conventional 6T SRAM design: Comparing the various read/write metrics values listed in column **6T SRAM** with those in column **8T1R**, it can be seen that the performance of the 8T1R array is comparable to that of the conventional 6T array. The slight change in the delay and energy consumption of the 8T1R write operation is due to: a) the existence of low-leakage current passing between the nodes Q and QB through the transistors "MM1" and "MM2" in fig. 3.5, and b) the small extra capacitances at the storage nodes 'Q' and 'QB' due to the source/drain junction capacitances of "MM1" and "MM2".

Comparison with low-power NV-SRAM designs: In this part, we focus on discussing the results of 8T1R design versus those of the 8T2R [93], Rnv8T [92], and 9T2R [94] designs. Those are considered as low-power NV-SRAM arrays, since they do not have the high leakage currents found in the 6T2R [17] and 7T2R [114] designs. **The Rnv8T cell** improves the write operation performance compared to that of the conventional 6T cell (i.e., delay is improved by 4.5% , while WDNM in enhanced by 3%). However, there is a slight increase in the energy consumption by 4%. This is resulting from using the RRAM devices and their access transistors to create a parallel path for the write operation. Also, due to increasing the capacitance of BL and BLB, the read delay and noise margin are worsened (i.e., delay is almost doubled and RDNM is lowered by 10%).

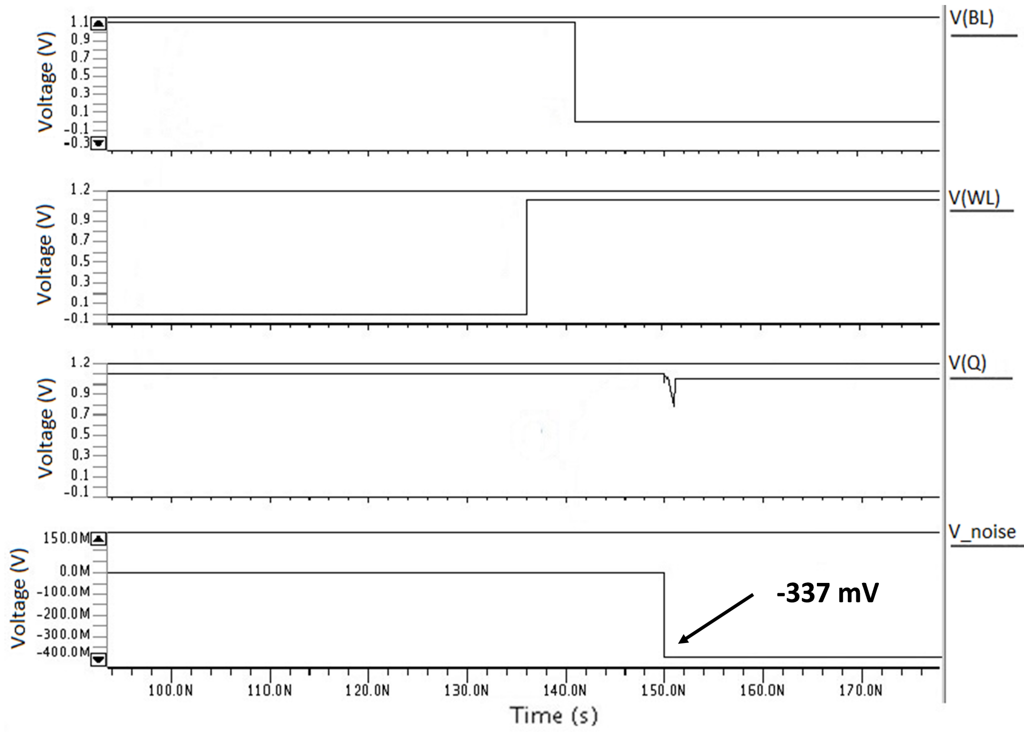

Figure 3.15: Waveforms for computing the noise margin of 8T1R cell. "V_noise" in figure describes the noise signal voltage level which results in the write failure. In this case, with a noise signal of 337 mV level, the write operation fails.

The 9T2R cell uses the same structure as that of the Rnv8T cell but with the addition of isolating the read and write operations by having an extra transistor which acts as a dedicated read-port for the cell. This comes at the expense of increasing the write delay by 23% compared to the results of the Rnv8T cell due to the imbalance in the capacitances at nodes 'Q' and 'QB'. Also, despite the fact that the 9T2R improves the read operation performance (i.e., lowering the delay and increasing the RDNM by 50%), the cell has other disadvantages:

- The read operation becomes single-ended which introduces challenges such as: a) increasing the sensing range which limits the number of cells per bitline, and b) reducing the SA immunity to supply noise as in [113].
- The SRAM cell area increases by about 30% compared to the Rnv8T cell. Fig. 3.16a and fig. 3.16b show the layout of the Rnv8T and the 9T2R NV-SRAM cell. For

simplicity, the routing metal layers are omitted from the figures. The RRAM device

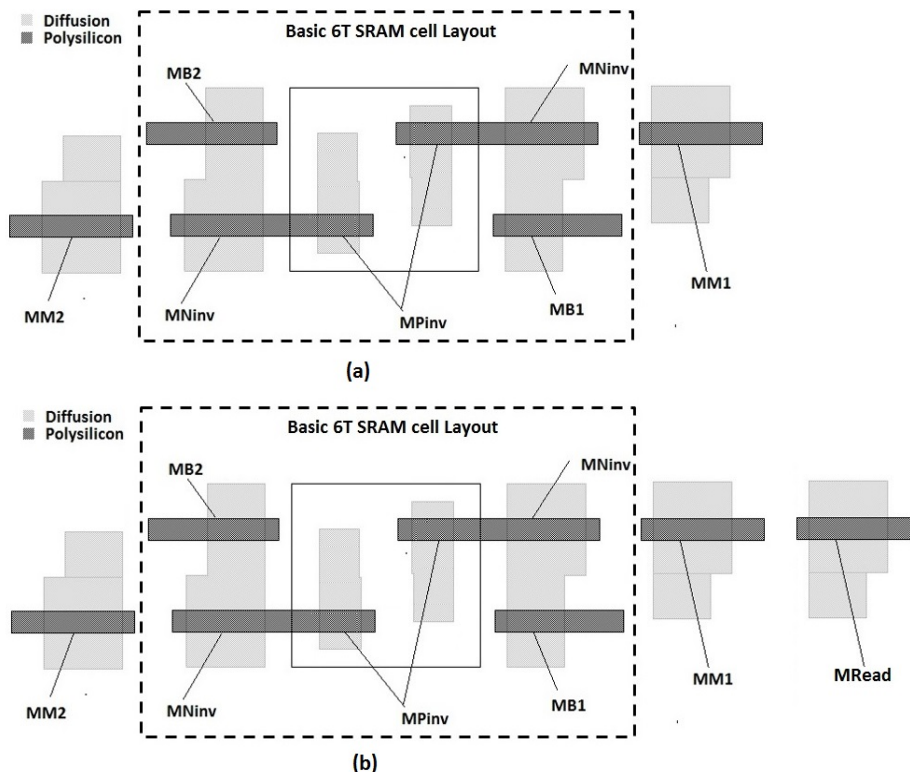

Figure 3.16: Layout of (a) Rnv8T cell and (b) 9T2R cell. The names of the transistors are aligned with those in fig. 3.5. **MRead** is the transistor connected to the dedicated read port as in [94].

in fig. 3.16a and fig. 3.16b is vertically stacked in the Back-End Of Line (BEOL) connectivity layers above the access transistors "MM1" and "MM2". Hence, its area does not affect that of the NV-SRAM cell.

.

Comparison with dense NV-SRAM designs: In this part, we discuss the results of the 8T1R array in table 3.2 with those of the 6T2R and 7T2R arrays which add less transistors to the conventional 6T SRAM cell. The 6T2R and 7T2R NV-SRAM arrays suffer from high-leakage currents due to the existence of direct paths through the RRAM devices to ground. This degrades the performance of both the read and write operations specially in terms of WDNM and RDNM. **The 7T2R cell** has a better performance than

the 6T2R cell since the path, which connects the RRAM devices together with the storage nodes, has always one RRAM device in HRS state. This decreases the leakage current by almost 90% compared to that in the 6T2R design. However, the energy is still significantly higher than that of the low-power NV-SRAM designs (e.g., 8T2R and 9T2R). As suggested in [114], keeping both of the RRAM devices in HRS could further decrease the leakage current but this comes at the expense of increasing both the delay and energy consumption of the restore operation as discussed in section 3.3.2.
In summary, the 8T1R design provides the best trade-off among all the previously proposed RRAM-based NV-SRAM designs in terms of having good write and read energy consumption and delay, while keeping the cell area as small as possible and not significantly increasing its leakage power.

3.3.2 Store and Restore Operations

In comparison with the other RRAM-based NV-SRAM designs, the most significant improvement of the proposed 8T1R cell is that it greatly reduces the energy consumption for the store/restore operations. Table 3.3 summarizes the simulation results for the store and restore operations of the various 128x128 RRAM-based NV-SRAM arrays. The 9T2R and the Rnv8T designs are using the same store and restore methodology [92, 94]. Hence, the listed results for the Rnv8T array in table 3.3 are also applicable for the 9T2R array. **For the store operation**, the time to change the state of the RRAM device is about 9 ns. The reason why the Rnv8T design needs double this time is because the store operation in this NV-SRAM cell consists of two distinct operations as explained in [92]: a) increase the voltage of the BL and BLB to VDDH and then b) increase CVDD to VDDH and grounding the BL and BLB lines. In 6T2R [17], 8T2R [93], and 7T2R [114] arrays, the RRAM devices connected to the storage nodes 'Q' and 'QB' are required to be at HRS after the completion of the restore process. Accordingly, the store operation for those designs only changes the state of the RRAM device connected to the storage node at logic '0' to LRS. Due to using only one RRAM device, the 8T1R cell reduces the store energy of the 128x128 array by at least 60 % when compared to the store energy of the 7T2R design which is the lowest value for the previously proposed RRAM-based NV-SRAM arrays. For instance, in the 8T2R array [93], the control lines voltage is raised to VDDH. Hence, although only one RRAM device is expected to be programmed, the current passing through the other RRAM cell, causes more energy dissipation.

For the restore operation, as explained in section 3.2.3, our 8T1R requires more time (2.6x compared to the Rnv8T design) to retrieve the data saved on the RRAM device due to the multiple steps involved in this operation. However, for the 6T2R, 7T2R, and

Table 3.3: Comparison of store and restore operations of various 128x128 RRAM-based NV-SRAM arrays

Parameter Name	**6T2R**	**8T2R**	**Rnv8T**	**7T2R**	**8T1R**
Store Energy (pJ)	1.17	1.04	2.08	0.954	0.387
Store Speed (ns)	9	9	18	9	9
Restore Energy (fJ)	18 + 1170*	13.6 +1040*	13.6	13.6 + 954 *	4.1
Restore Speed (ps)	63.4 + $9x10^{3*}$	63.8 + $9x10^{3*}$	63.8	63.8 + $9x10^{3*}$	170

- For all the items with *, the second number describes the RESET operation energy and delay requirements which is needed after each restore operation. The Rnv8T and 8T1R designs do not require a RESET operation after the completion of the restore process.

8T2R designs, there is an extra 'RESET' operation which is needed after the SRAM data is retrieved. The 'RESET' operation re-programs the two RRAM devices to their HRS and it requires the same delay ($\approx$ 9 ns) and energy as that of the store operation. Accordingly, the total delay results of the restore process for those designs in table 3.3 are higher by 2 order of magnitude. Compared to the Rnv8T design, the 8T1R consumes almost 70% less energy for the same duration of the restore operation. This is because the RRAM access transistors, "MM1" and "MM2" in fig. 3.5), are only activated for a portion of the restore process duration. In the Rnv8T cells, the RRAM access transistors, which connect the RRAM devices to ground, are turned ON throughout the restore operation period and hence, their energy consumption is increased in comparison to that of 8T1R.

3.4 Summary

A new RRAM-based NV-SRAM cell is presented in this chapter. The different modes of operations of 8T1R cell are explained, and its simulation results are demonstrated. Compared to the previously proposed RRAM-based NV-SRAM designs, the 8T1R cell adds non-volatility to the conventional 6T SRAM with marginal impact on the performance

of write and read operations. The proposed cell also decreases the impact of reliability soft-errors by reducing the store and restore operations energy by more than 60% and 70 %, respectively. Accordingly, the proposed technique can be easily integrated in nowadays SRAM-based designs (e.g., GPU designs [117, 118]) used to run machine learning applications to lower their power consumption and reduce the probability of having faults introduced by RRAM reliability soft-errors.

Chapter 4

Resolving the RRAM Reliability Soft-Errors in 1T1R RRAM Memory Arrays

In this chapter, a proposed methodology for detecting and fixing RRAM reliability soft-errors in the 1T1R array is discussed. The main concept is to detect the reduction in the LRS of the RRAM device of each 1T1R cell due to the diffusion of oxygen vacancies out of the conductive filament containment. Then, using a suggested refresh circuit, the LRS for the cells is restored. In section 4.1, a brief introduction is provided about the interest in the 1T1R array and its reliability soft-errors. The proposed methodology for detecting the RRAM reliability soft-errors is explained in section 4.2. Following this, in section 4.3, the details of the refresh circuit are presented. We conclude this chapter by presenting the SPICE and system level simulations for the modified read circuit incorporating the suggested refresh methodology in sections 4.4.1 and 4.4.2. The SPICE simulations are run using the HfO_x *RRAM experimentally-verified SPICE model [33] which takes into account the RRAM reliability soft-errors. CACTI C++ files [42] are used for system level simulations to estimate the effect of the modified read circuit on the performance of high-capacity memory arrays. The main contribution of the work presented in this chapter is proposing for the first time a methodology to address the RRAM reliability soft-errors with minimum impact on the basic operations of the 1T1R arrays. Also, since the refresh circuit can detect the drift in LRS of RRAM devices, it can be used to detect MEU which cause intermediate change in the RRAM state.*

4.1 Introduction

Due to their ability to retain data in the absence of power supply, NVMs were first used as a mean to protect important system data from being erased when the power supply is cut-off (e.g., Field Programmable Gate Array (FPGA) configuration bits [119] and the processor boot-code [120]). Although flash memory is the dominant NVM technology in today's market, the geometrical and voltage scaling requirements in sub-20 nm technologies make the advancement of flash memory quite challenging due to its charge-based floating gate structure [121]. New structures are actively explored in seek of the next generation of low-cost, low-power, high-speed, and high-capacity NVM technologies. Due to its attractive characteristics, discussed in section 2.4, RRAM is one of the most promising NVM technologies.

The 1T1R cell suffers, however, from reliability soft-errors as discussed in section 2.7.1. In this chapter, we propose a novel refresh circuit which detects and fixes the reliability soft-errors by analyzing the drift in RRAM resistive state to determine whether it can be recovered or not. In case of soft-errors, re-programming the device restores its correct state. Our proposed refresh circuit has little impact on the memory design since it is not attached to each cell individually, instead it is integrated into the SA of each BL. Moreover, the refresh operation in our suggested methodology is triggered when: a) the cell read data indicates that a refresh operation is required, or b) the cell data has not changed after a predefined refresh period which is in the order of of days to months [38] for RRAM arrays instead of few 10^{-6} secs for DRAM. This preserves the advantage of using the RRAM-based arrays as low-power memory technology.

4.2 The Concept of the Refresh Methodology

The failures in RRAM-based 1T1R arrays are extensively studied in [35, 36, 37, 38, 39] and they can be classified into two main categories:

- **Hard-Errors**: These failures are mainly caused by the fact that the RRAM device, like all other NVM devices, has an endurance limit (in the order of 10^{10} cycles for HfO_x RRAM device). If the device reaches this limit, its state can not be recovered due to the depletion of oxygen vacancies in the oxide material.
- **Soft-Errors**: These failures, which cause the drift in the RRAM resistive state, are discussed in details in section 2.7. These errors can be recovered by re-programming

the device to its original state. We focus in this chapter on the reliability soft-errors discussed in details in section 2.7.1, while we will explain how the radiation soft-errors can be addressed in chapter 5.

In this work, to reduce the power consumption of the memory array, low-programming conditions are used to alter the RRAM resistive state (i.e., $\leq 40\mu$A). Under this circumstance, the reliability soft-errors in these arrays mainly cause the drift of the RRAM state from LRS to HRS due to the reduction in oxygen vacancies within the conductive filaments and, hence the RRAM resistance tends to increase [36]. The process of LRS drift towards the HRS is gradual and, if detected early, it can be fixed by refreshing (i.e., rewriting) the LRS of the device.
The proposed refresh methodology is based on the concept of dividing the RRAM resistance range between its HRS and LRS into four regions shown in fig. 4.1: **Region I** defines

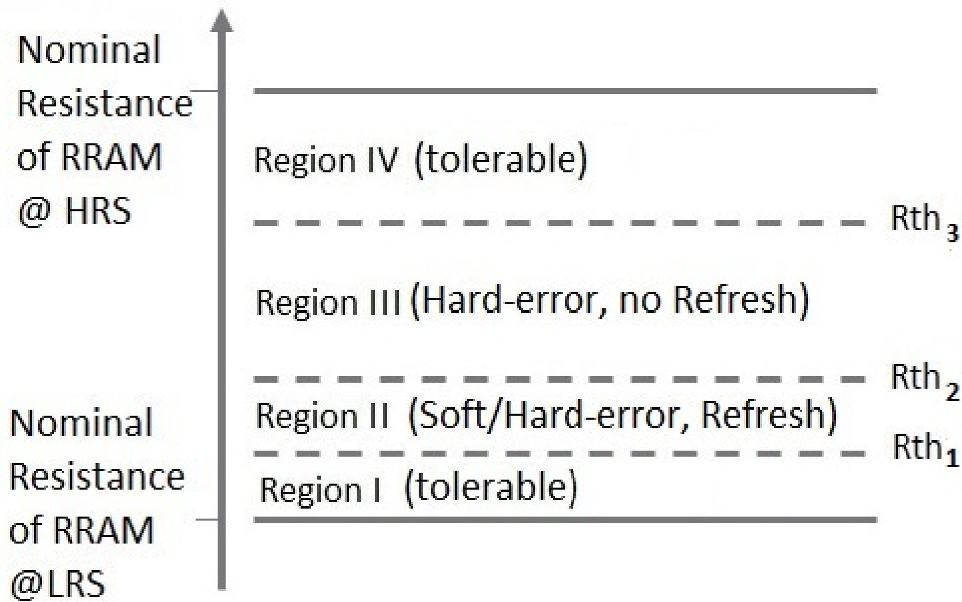

Figure 4.1: Division of the RRAM resistance range. “Region II” defines the resistance range where the refresh operation is triggered. In “region I” and “region IV”, the RRAM state is considered at LRS and HRS, respectively. Accordingly, no refresh operation is required. When the resistance of RRAM is in “region III”, the device is considered suffering from hard-errors since its resistance could not be refreshed earlier to LRS.

the zone where the RRAM resistance is considered at LRS and hence, the refresh operation is not triggered. The purpose of this region is to:

- Allow for minor changes in the resistance of RRAM devices coming from the variation in the fabrication process and the read/write operating conditions.

- Limit the amount of refresh operations needed in order to save energy and to reduce the impact of the refresh operation on the device endurance since it is basically a write operation.

"Region I" is bounded by the threshold resistance value Rth_1 which is selected based on another resistance threshold value (i.e., Rth_2 value).

Region II defines the zone where the refresh operation is triggered to restore the LRS of RRAM devices. This region is bounded by the two threshold resistance values: Rth_1 and Rth_2. Rth_2 is the resistance value that maps to the RRAM state above which it is no longer considered as LRS. Since the resistance change due to reliability soft-errors is gradual, as long as the cell is refreshed frequently, the RRAM LRS resistance does not drift beyond Rth_2 unless in case of hard-errors. Based on [36, 37, 38], Rth_2 is chosen to be 10 times of the nominal LRS resistance value.
To determine the value of Rth_1, there are two contradicting requirements to consider. On one hand, Rth_1 needs to be as close as possible to Rth_2 to reduce the frequency of refresh operation. On the other hand, the larger the difference between Rth_2 and Rth_1, the better is the performance of the SA due to the larger difference between the read voltages for the 1T1R cell when its resistance is at Rth_1 and when it is at Rth_2 (i.e., V(Rth_1) and V(Rth_2), respectively). Using the SPICE models built based on the data in [36, 37, 38], the graph in fig. 4.2 is created. This graph shows that, as we move away from the Rth_2 (i.e., the value of $|R - Rth2|$ increases), the difference between V(Rth_1) and V(Rth_2) increases but the refresh cycle duration decreases. The Rth_1 represents the optimal threshold value after which any change in the resistance increases the refresh operation energy consumption due to the exponential reduction in its cycle duration.

Region III defines the zone where the read operation is unable to differentiate whether the original state of the RRAM was LRS or HRS. When the RRAM resistance is in this region, it indicates that the device is suffering from a hard-error. Since the drift of the RRAM state is a gradual process, if the boundaries Rth_1 and Rth_2 are properly selected, soft-errors are fixed before the RRAM resistance reaches this region. The upper bound of this domain is Rth_3 which is defined as 3 times Rth_2 as explained in [36].

Region IV defines the zone where the RRAM state is considered at HRS. In this region, the refresh operation is not initiated since the soft-errors in low-programming conditions mainly cause the drift of the LRS to HRS (i.e., the device in HRS tends rarely to drift to LRS).

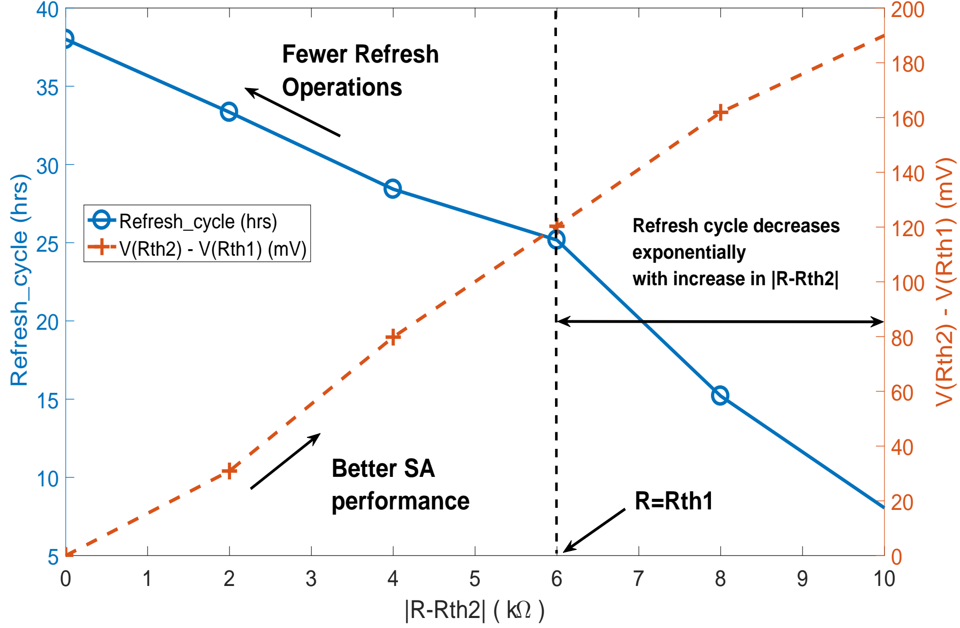

Figure 4.2: Rth_1 threshold value selection. The optimum value of Rth_1 is chosen as a compromise between increasing the difference between V(Rth_1) and V(Rth_2) and reducing the amount of refresh cycles required.

Based on the type of RRAM device used in the 1T1R array, the region boundaries might require to be re-calculated but the general concept should still be applicable.

4.3 Refresh Circuit Schematic and Operation

The block diagram of the circuit implementing the refresh methodology, discussed in section 4.2, is shown in fig. 4.3. The refresh circuit consists of:

- **SA1 (1-input) and SA2 (2-input)**: These are the voltage SAs used to detect the RRAM state of the 1T1R cell. At the beginning of the read operation, the BL is precharged to VDD. Then, a pulse is applied to the WL to enable the transistor of the 1T1R cell for a certain period of time (3 ns in our experiments as explained in section 4.4.1). During the WL pulse, the BL voltage changes depending on the state of the RRAM device. At the end of the WL pulse, the voltage of the BL (i.e., V_{sense})

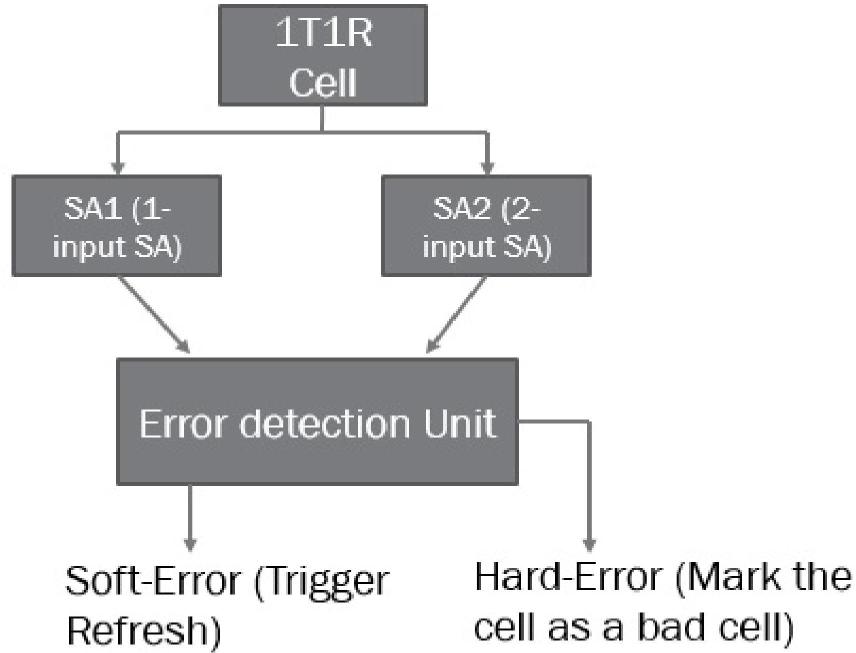

Figure 4.3: Block diagram of refresh circuit. Compared to the normal read circuitry, "SA2" and "error detection unit" blocks are added to sense the four regions of RRAM resistance range illustrated in fig. 4.1.

is sensed and compared to reference voltages corresponding to the different threshold resistance values discussed in section 4.2 (i.e., V(Rth_1), V(Rth_2), V(Rth_3)). "SA1 (1-input)" compares the V_{sense} to V(Rth_3) in order to detect whether the RRAM state is at HRS (Region IV of fig. 4.1) or not. "SA2 (2-input)" is used for checking whether the RRAM resistance is in region I/II or region III. The schematic and operation of "SA1" and "SA2" are discussed in section 4.3.1.

- **Error detection unit**: This is a logic circuit which combines the results from "SA1" and "SA2" to decide: a) whether a refresh operation is needed or not, and b) whether the RRAM device is suffering from hard-error (by checking if its resistance is in region III in fig. 4.1). The circuit operation is explained in section 4.3.2. In case if a refresh operation is needed, the unit sends a signal (i.e., 'Soft-Error' signal in fig. 4.3) to the memory controller to initiate a write operation on the currently selected cell to restore its LRS. In case if the RRAM device of the selected 1T1R cell is suffering from hard-error, the "error detection circuit" sends another signal (i.e., 'Hard-Error'

signal in fig. 4.3) to the memory controller to mask the address to this erroneous cell by mapping it to another 1T1R cell which is still functional as explained in [122].

4.3.1 Sense Amplifier of the Refresh circuit

The "SA1 (1-input)" and "SA2 (2-input)" circuits in fig. 4.3 uses latch-type voltage SA design similar to that in [123]. The two SA designs are basically the same with the difference that "SA2" has two input reference voltages while "SA1" has only one. Fig. 4.4 shows the circuit schematic for "SA2". "SA1" has the same design as that of "SA2" without the

Figure 4.4: "SA2" circuit schematic. Compared to the structure of latch-type voltage SA in [123], the transistors "M13", "M14", and "M15" are added to support the "two-cycle" comparisons required to define whether the RRAM resistance is in region I, II, or III.

marked transistors "M13", "M14", "M15" in fig. 4.4. These MOS transistors are added to the "SA2" circuit in order to have four input voltages: two input nodes connected to

reference voltages and the other two are connected to the voltage that needs to be sensed. This is basically because "SA2" needs to compare the BL voltage (i.e., V_{sense}) versus two independent reference voltages (i.e., V(Rth_1) and V(Rth_2)) in two separate sub-operations. The voltages V(Rth_1) and V(Rth_2) are connected to the nodes INN and INN2 in fig. 4.4 while the V_{sense} is connected to nodes INP and INP2. All the control signals in fig. 4.4 ('SE1', 'SE2', 'SE', 'SEN') are generated by the "error detection unit" as explained in section 4.3.2. The "SA2" works in the following sequence:

1. Signals 'SE', 'SE1', 'SE2' are first disabled (i.e., connected to ground) to discharge the output nodes 'OUTN' and 'OUTP' to ground (i.e., logic '0').

2. Only the signals 'SE' and 'SE1' are raised to VDD (logic '1') to compare V_{sense} versus V(Rth_1). If V_{sense} < V(Rth_1), the signal 'OUTN' charges up to VDD while 'OUTP' remains at logic '0'. This indicates that RRAM device of the selected 1T1R cell is at LRS (i.e., region I) and hence no refresh operation is needed. Oppositely, if V_{sense} > V(Rth_1), the node 'OUTP' changes to logic '1' while node 'OUTN' remains at logic '0' which means that a second comparison cycle between V_{sense} and V(Rth_2) is needed (i.e., steps 3 and 4 below).

3. All the control signals are disabled again (i.e., 'SE', 'SE1', 'SE2' are connected to ground, while 'SEN', which is the inverted version of 'SE', is set to logic '1') to discharge both of the output nodes 'OUTN' and 'OUTP' to ground.

4. Only the signals 'SE' and 'SE2' are set to logic '1' to compare V_{sense} with V(Rth_2). If V_{sense} < V(Rth_2), the output node 'OUTN' charges to VDD, while 'OUTP' remains at logic '0'. This indicates that the RRAM resistance is in region II and hence a refresh operation is needed. If V_{sense} > V(Rth_2), this means that RRAM resistance is either in region III or IV. The output nodes of "SA1" are hence used to indicate whether the device is in HRS or not as described in section 4.3.2.

Fig. 4.5 shows the waveforms describing the "SA2" circuit operation for the case when $V(Rth_1) < V_{sense} < V(Rth_2)$. The "SA1" design works exactly as "SA2" with the exception that it uses only one reference voltage to compare the V_{sense} with V(Rth_3). Hence, only the steps 1 and 2 of the "SA2" operation sequence are applicable for "SA1" circuit. Fig. 4.6 illustrates the waveforms obtained from running SPICE simulations for the "SA2" circuit.

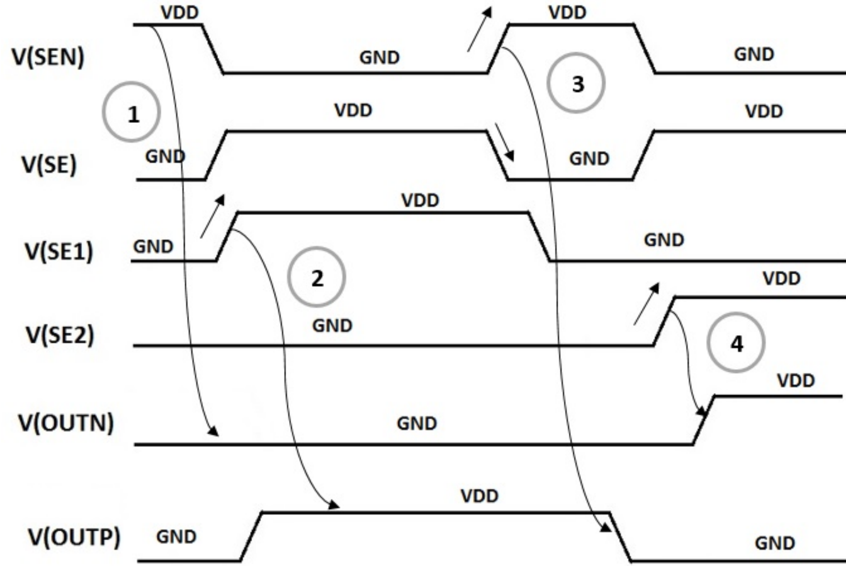

Figure 4.5: Illustration of waveforms of "SA2" circuit operation for the case when $V(Rth_1) < V_{sense} < V(Rth_2)$. In this scenario, the output from "first cycle" of comparison indicates that $V_{sense} > V(Rth_1)$ by having the signal 'OUTP' set to VDD. Accordingly, the signals 'OUTN' and 'OUTP' are discharged to ground by disabling the signal 'SE' (and enabling 'SEN'). After the "second cycle" of comparison, the signal 'OUTN' is raised to VDD indicating that $V_{sense} < V(Rth_2)$.

4.3.2 Error Detection Unit

The "error detection unit" has two main functions:

- Control the generation of the enable signals (i.e., 'SE1', 'SE2', 'SE', 'SEN' in fig. 4.4) for "SA1" and "SA2" circuits.
- Decide whether to raise the 'HARD_ERROR' or 'SOFT_ERROR' signals, shown in fig. 4.3, based on the output from the "SA1" and "SA2" designs.

The circuit structure of this unit is shown in fig. 4.7. The signal '$READ$' is generated by the memory controller to initiate the read operation. 'REF_VERIFY' signal is set by the memory controller at the end of refresh cycle duration to check the RRAM state in the 1T1R array independent of whether there is a read operation initiated or not. The 'REF_VERIFY' pulse is also used to verify that the RRAM state has successfully been

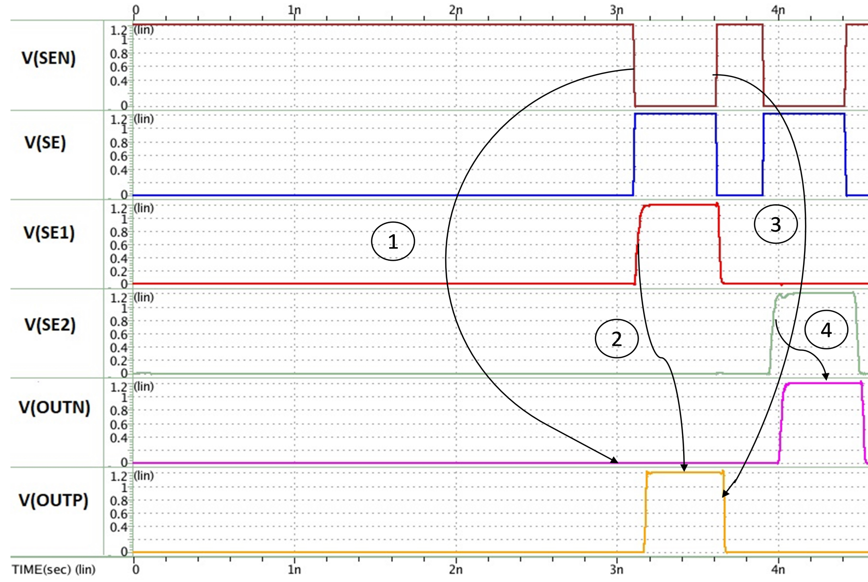

Figure 4.6: SPICE waveforms of "SA2" circuit operation for the case when $V(Rth_1) < V_{sense} < V(Rth_2)$. The number sequences in this figure are consistent with those in fig. 4.5.

restored to its LRS after the refresh operation. This is done by starting a read cycle using a 'REF_VERIFY' pulse after the refresh process is completed. If the RRAM state is not at its LRS (i.e region I in fig. 4.1) after refresh, the '$HARD_ERROR$' pulse is generated to the memory controller to prevent any future access to this cell.

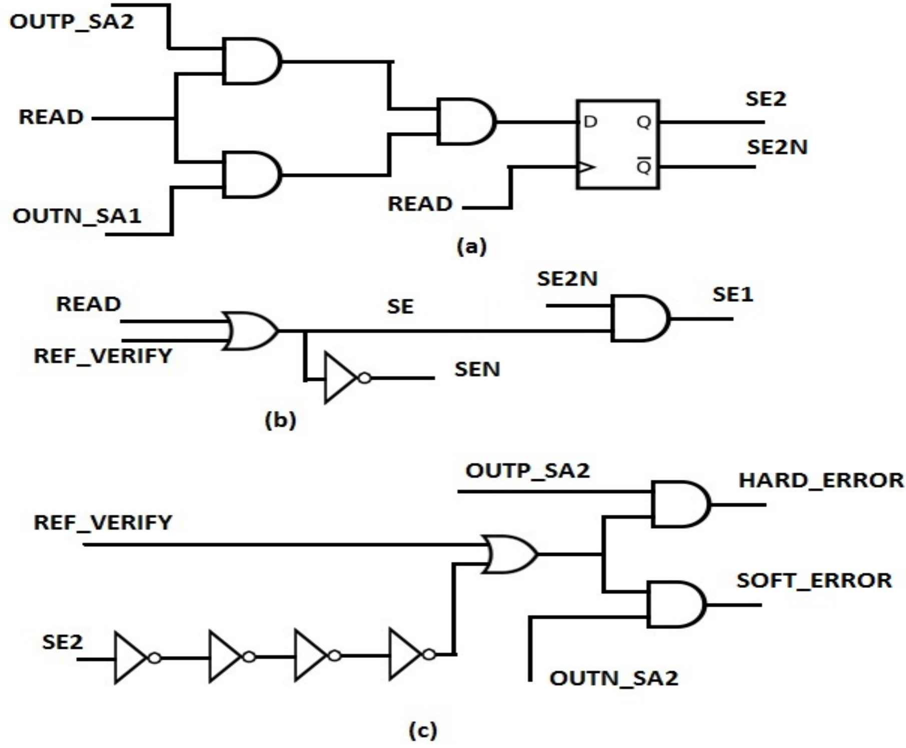

Figure 4.7: Circuit schematic of the "error detection unit": a) SE2 generation circuit which is responsible of enabling the 'SE2' signal when $V(Rth_1) < V_{sense} < V(Rth_3)$, b) SE/SE1 generation circuit which is responsible of enabling the signals 'SE' and 'SE1' used by "SA1" and "SA2", and c) refresh detection unit which compares the output from "SA1" and "SA2" to decide whether the selected 1T1R cell is suffering from either soft-error or hard-error.

There are three sub-circuits in the "error detection unit":

- **SE2 generation circuit** in fig. 4.7a creates the 'SE2' pulses needed for comparing V_{sense} versus V(Rth_2) as explained in section 4.3.1. If $V_{sense} > V(Rth_1)$, the node 'OUTP' of "SA2" circuit (i.e., '$OUTP_SA2$' in fig. 4.7) goes to logic '1'. Also, if $V_{sense} < V(Rth_3)$, the node 'OUTN' of "SA1" (i.e., '$OUTN_SA1$' in fig. 4.7) charges to logic '1'. This means that the resistance of the RRAM is either in region II or III of fig. 4.1. In this case, a 'SE2' pulse needs to be generated to compare V_{sense} versus V(Rth_2).

- **SE/SE1 generation circuit** in fig. 4.7b creates the 'SE', 'SEN', and 'SE1' pulses needed to read the RRAM state by the "SA1" and "SA2" circuits as described in section 4.3.1. These signals are generated in response to '$READ$' or 'REF_VERIFY' pulses.

- **Refresh detection circuit** in fig. 4.7c determines whether the RRAM needs a refresh operation or not based on the output from "SA2". In response to 'SE2' pulse, if $V_{sense} < V(Rth_2)$, the output node 'OUTN' of "SA2" (i.e., '$OUTN_SA2$' in fig. 4.7) rises to VDD. This means that the resistance of the RRAM device is in region II and a refresh operation is needed. Hence, the signal '$SOFT_ERROR$' is raised. Oppositely, if $V_{sense} > V(Rth_2)$, the output node 'OUTP' of "SA2" (i.e., '$OUTP_SA2$' in fig. 4.7) goes to logic '1' since the RRAM resistance is in region III. In this case, the signal '$HARD_ERROR$' is raised to VDD and the cell is marked by the memory controller as a bad cell to mask its address.

4.4 Simulation Results

Fig. 4.8 illustrates the block diagram for the flow of simulation runs conducted to evaluate the proposed methodology and its impact on the read and write operations of 1T1R arrays.

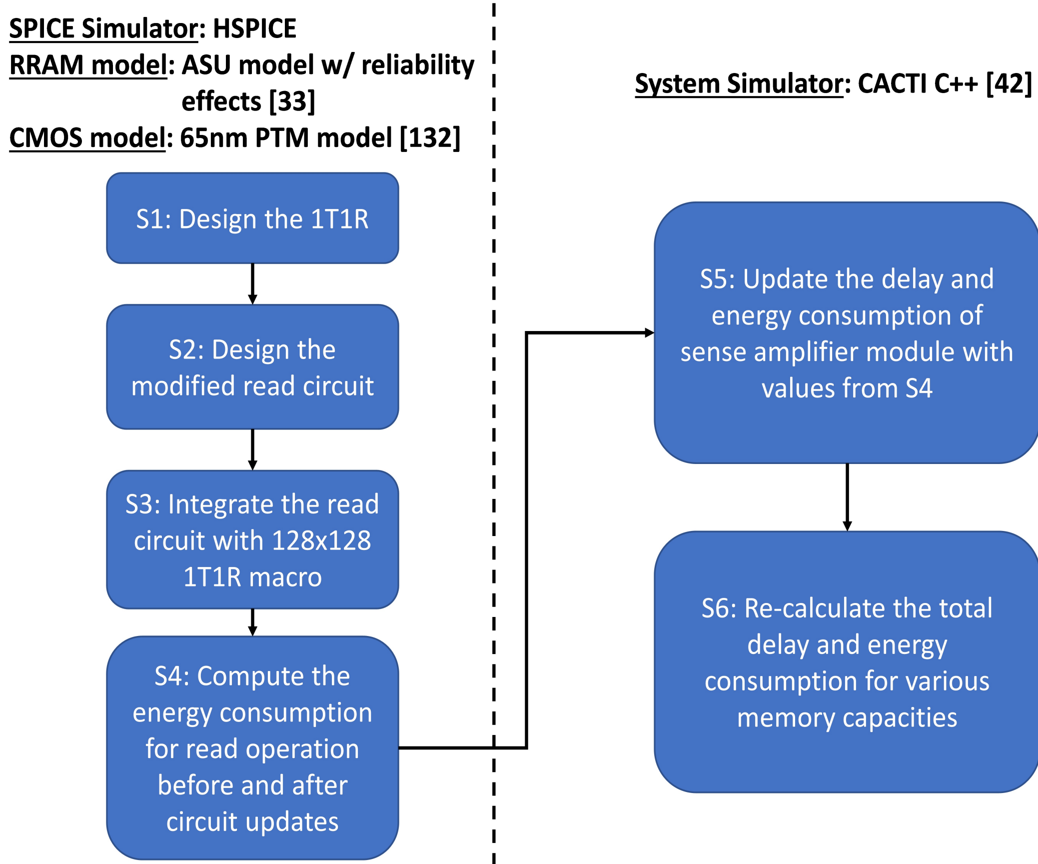

Figure 4.8: Block diagram for the simulation runs and the tools used to evaluate the proposed refresh methodology. The ASU model described in [33] is used for the SPICE simulation runs while the CACTI C++ [42] is used to estimate the impact of read circuit modifications on large memory arrays.

To fully analyze the suggested refresh circuit, SPICE and system level simulations are required. Unlike the SPICE simulation runs discussed in section 3.3, HSPICE [124] is used together with a 65 nm Predictive Technology Model (PTM) to align with the simulation data in [33] which fit the experimental studies for the RRAM reliability soft-errors in [36, 37, 38]. As for the system level simulations, the CACTI C++ files are used to assess the modified read circuit impact on the whole memory array performance similar to what is done in [125]. The first step of the simulation flow in fig. 4.8 is to verify the correct operation of 1T1R cells with the modified read circuit after integrating the refresh methodology discussed in section 4.3 (i.e., steps S1 and S2 in fig. 4.8). This includes

selecting the right pulse duration for the read operation to guarantee the ability of SA to detect the smallest change in the RRAM state. After this, a 128x128 array is formed and the read delay and energy consumption are computed and compared with the case when the refresh methodology is not added to the read circuit (i.e., steps S3 and S4 in fig. 4.8). The results from SPICE simulations on the 128x128 array are then integrated to CACTI C++ files by changing the delay and energy consumption values defined in the SA C++ module (i.e., step S5 in fig. 4.8). After this, multiple system level simulations are run on different capacity memory arrays made of 128x128 macros to estimate the percentage of increase in energy consumption and delay of the read operation due to adding the refresh circuit (i.e., step S6 in fig. 4.8).

4.4.1 SPICE Level Simulation Results

In this subsection, we mainly focus on the sensing circuit performance after modifying it as discussed in section 4.3. Table 4.1 shows the SPICE simulation results of the sensing circuit in section 4.3 incorporated with a 128x128 1T1R macro. The SA reference voltages (i.e., V(Rth_1), V(Rth_2), V(Rth_3)), discussed in sections 4.2 and 4.3, are chosen assuming that the read pulse used to discharge the BL is 3 ns. This discharging time is chosen in order to create a minimum 0.15 V change in the SA output for every 0.1 nm drift in the RRAM gap distance between its conductive filament tip and its top electrode. The

Table 4.1: Sensing circuit performance for a 128x128 1T1R macro

Metric	**Reference SA**	**SA1 + SA2 + Error detection unit**	
		Region I/IV	**Region II/III**
Delay (ps)	56.2	61.1	143.7
Energy (pJ)	18.2	47.4	72.5

column **Reference SA** in table 4.1 describes the case when only the regular two-tail SA is attached to BL while the other column refers to the proposed read circuitry in fig. 4.3. The sub-column **Region I/IV** lists the simulation results for the case when the RRAM resistance of is in region I or IV (i.e., no need to check for hard/soft-errors), while the

sub-column **Region II/III** describes the case when the RRAM resistance is in region II or III (i.e., hard/soft-errors need to be checked). Due to the overhead of the various circuits in fig. 4.3, the delay and energy consumption of the sensing design has increased. The results listed in column **Region II/III** are higher than those in column **Region I/II** because of the need to trigger a "second cycle" of comparison to determine whether the RRAM resistance is in region II or III as detailed in section 4.3.

4.4.2 System Level Simulation Results

Effect on Read Delay and Energy

Using the SPICE simulation results in section 4.4.1, the CACTI files [42] are modified (basically modifying the delay and energy consumption of the SA module as specified in table 4.1) to calculate the estimated impact of the proposed refresh methodology on the read operation of a 65 nm technology 8Gb memory. It is worth mentioning that the chosen capacity for the 1T1R array is consistent with the current NVM capacities which is in the range of Gb. Table 4.2 summarizes the comparison results of the read operation before and after integrating the refresh circuit. Table 4.2 shows that the other memory modules (e.g.,

Table 4.2: Read operation in 8 Gbit 1T1R RRAM array

Metric	**1T1R Array**	**1T1R Array + Refresh Circuit**	
		Region I/IV	**Region II/III**
Delay (ns)	5.37	5.39	5.47
Energy (nJ)	4.16	4.4	4.61
ΔDelay (%)	-	0.4	1.8
ΔEnergy (%)	-	5.74	10.68

address decoders and repeaters) have significant impact on the read energy and delay. The refresh circuit impact on the read delay is very marginal. As for the read energy increase, in most cases where no refresh operation is needed (i.e., region I/IV in fig. 4.1), the penalty for an 8 Gb memory is about 6%. In case when a "second cycle" of comparison is needed (i.e., region II/III in fig. 4.1), an 10.68% increase in the read energy is observed. However, it is worth mentioning that the refresh rate in the RRAM 1T1R array is in the range of

days to months as explained in [38]. Hence, the increase in memory power consumption due to the refresh circuit is ≈ 6% on average for a 8 Gb memory array. To prove the small overhead of the proposed refresh methodology, the circuit is integrated to other various memory capacities. The percentage of change in the read delay and energy, in comparison to the case when the refresh methodology is not integrated, is plotted in fig. 4.9.

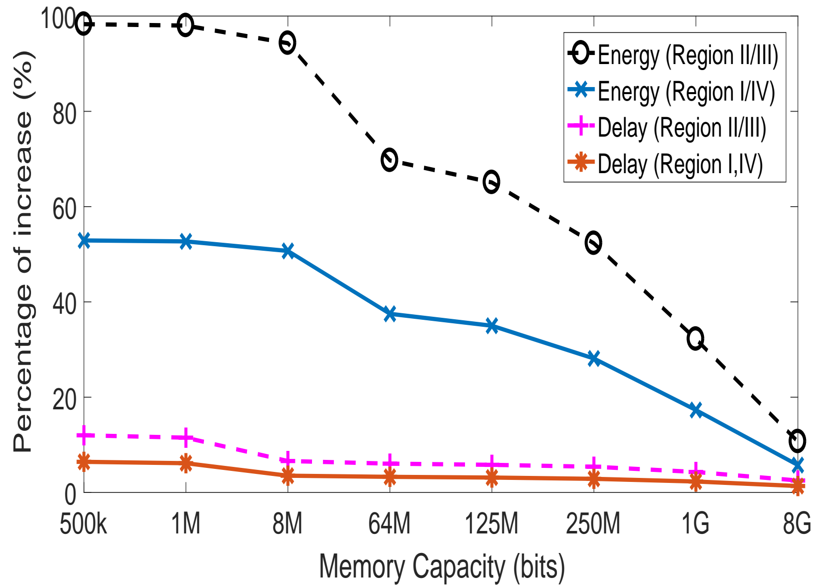

Figure 4.9: Refresh circuit impact on read energy and delay for different array sizes. The change percentages in figure are computed by comparing the read circuit delay and energy consumption with the case when the refresh circuit is not integrated. Since the delay and energy consumption of read operation is mainly governed by other memory components, the higher the capacity of RRAM arrays, the lower is the impact of modified read circuit on the increase of energy and delay of read operation.

Fig. 4.9 shows that decreasing the memory capacity increases the impact of the refresh circuit on the energy consumption of the system. This is because the contribution from other components in the memory system, such as address decoders, to the read energy consumption decreases with smaller memory capacity. Accordingly, the increased energy consumption of the modified sensing circuit becomes more dominant. The change in the read delay in fig. 4.9 and table 4.2 is negligible since the read operation latency is mostly determined by the other sub-operations like the time required to precharge and discharge the bitlines (3 ns as in section 4.4.1) rather than the extra delay added by the refresh circuit

(in the order of picosecond as in table 4.1). It is worth mentioning that the sudden increase in the energy curve when the memory capacity is changed from 64 Mb to 8 Mb is caused by the significant decrease in the number of memory sub-banks (i.e., > 8x). Accordingly, the impact of the newly proposed sensing circuit on the performance of the read operation is higher. This is different from the case when the memory capacity is reduced from 125 Mb to 64 Mb which results in decreasing the number of memory sub-banks by only 2x. This sudden increase is not seen in the case when the memory size is further reduced from 8 Mb to 1 Mb because the number of sub-banks in this case is kept the same in order to optimize the power consumption by reducing the number of address decoders and multiplexers. Alternatively, the number of 128x128 macros / bitline has increased causing more delay for the read operation as illustrated in fig. 4.9.

Effect on the Resilience of the RRAM Array to Soft-Errors

Using the data in [38], where RRAM cells are programmed to identify the different sources of RRAM reliability soft-errors, fig. 4.10 shows the percentage of the cells which retained their LRS state (i.e., "Good devices" in fig. 4.10) before and after adding the refresh circuit.

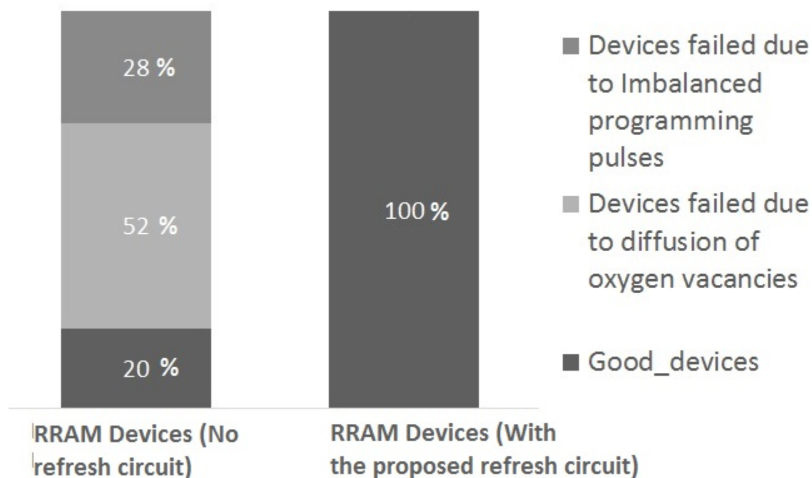

Figure 4.10: Refresh circuit effect on increasing the immunity of the RRAM 1T1R arrays to reliability soft-errors. Since the refresh circuit can detect and fix the reliability soft-errors generated from any source, referring to the experimental data in [38], the proposed methodology increases the resilience of RRAM 1T1R arrays to those soft-errors by 80%.

The proposed refresh methodology can improve the immunity of the RRAM 1T1R array to reliability soft-errors by about 80%.

4.5 Summary

In this chapter, a new refresh circuit is proposed to resolve the reliability soft-errors encountered in the RRAM 1T1R memories. Simulation data shows that, for an 8 Gb 1T1R memory, the refresh circuit has a small impact on the energy and delay of the read operation (i.e., 6% and 0.4%, respectively), while it improves the memory resilience to reliability soft-errors by 80%. Since the refresh circuit can detect the drift in the RRAM state, the proposed methodology can be also used to detect and fix radiation soft-errors in 1T1R RRAM arrays as explained in chapter 5.

Chapter 5

Resolving Single-Event Upsets in 1T1R RRAM Memory Arrays

In this chapter, a new proposed methodology is discussed for detecting and fixing the radiation soft-errors in 1T1R arrays. The refresh circuit, discussed in chapter 4, can be used to detect the intermediate change in the RRAM state due to MEU. Hence, the focus in this chapter is on detecting and resolving the complete change in RRAM state due to SEU. By adding an extra RRAM device to the basic 1T1R cell with a proper bias, if its state changes from HRS to LRS, this indicates that SEU has occurred. The read and write circuits modifications, required to support the proposed one-Transistor-two-RRAM device (1T2R) cell, are also discussed in this chapter. First, in section 5.1, a brief introduction about the SEE in 1T1R arrays is presented. After this, the suggested structure of the 1T2R cell is discussed in section 5.2. The required read circuit modifications to support the new cell are explained in section 5.3. Finally, the SPICE and system level simulation results, obtained using the experimentally-verified SPICE model in [33] and CACTI C++ files [42], are summarized in section 5.4. The main contribution of the work presented in this chapter is proposing a new cell structure and its related circuit modifications detect the shift in the RRAM state due to SEU. The work in this chapter and in chapter 4 offer an overall solution to detect and resolve the reliability and radiation soft-errors in RRAM arrays. This provides initial solutions for incorporating RRAM devices in products with advanced structures such as the neuromorphic systems discussed in chapter 6.

5.1 Introduction

As discussed in chapters 2 and 3, by storing the data of less-frequently accessed blocks on the NVMs and cutting off their power supply, the overall power consumption of the system can be significantly lowered [14, 92, 94, 126, 127]. Superior to charged-based memory cells (e.g., SRAM, DRAM), non-charge-based memories (e.g., RRAM arrays) are intrinsically immune to SEE [128, 129, 130, 131]. Accordingly, the intrinsic RRAM arrays are intended for use in high radiation environments [40]. Despite its intrinsic immunity to SEE, the data of 1T1R cell can be unintentionally changed due to the heavy-ions strikes on the junction of the MOSFET access device [40, 41]. Heavy-ions can create enough voltage drop across the RRAM devices in the half-selected 1T1R cells triggering the change in their saved data. Once detected, those soft-errors can be fixed by rewriting the original data. In this chapter, we propose a novel methodology for: a) detecting when the heavy-ions strikes cause a change in the 1T1R data, and b) recovering the information originally saved on the affected cells.

5.2 Proposed Methodology for Detecting and Fixing Single-Event Upset

Our proposed methodology to detect and fix SEU in the 1T1R RRAM arrays consists of two parts: the first part is related to modifying the memory cell itself and the second part is about the required changes in the read circuit to work with the newly suggested cell design. In this section, we explain the proposed structure of the memory cell. Then, in section 5.3, the associated modifications to the read circuit are discussed. Fig. 5.1 illustrates the schematic of the proposed 1T2R cell. Other than the RRAM device of 1T1R cell storing the data (i.e., cellRRAM in fig. 5.1), another RRAM device (i.e., senseRRAM in fig. 5.1) is added to detect SEU. To control the change in the senseRRAM state, an extra column control signal Write Enable (WE) is connected to its "P" terminal as shown in fig. 5.1 (i.e., all the cells in the same columns share the same extra WE control signal). Initially, the senseRRAM is at its HRS to minimize its impact on the write and read operations as detailed in sections 5.4.2 and 5.4.3, respectively. In our methodology, the senseRRAM switches to LRS only when SEU occurs. Also, compared to the 1T1R cell in fig. 2.17, the cellRRAM in fig. 5.1 is connected to the SL control line instead of BL. This is basically to read the state of the senseRRAM independently from that of cellRRAM during the same read operation as detailed in section 5.2.4.

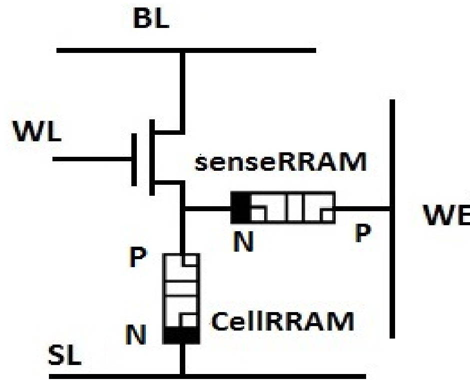

Figure 5.1: Schematic of the 1T2R cell. Compared to the normal 1T1R cell design, two main updates are added: a) extra RRAM device (i.e., senseRRAM) whose state indicates whether radiation soft-errors have occurred or not, and b) extra control line (i.e., WE) to correctly bias senseRRAM to track unintentional changes in cellRRAM state.

Since the RRAM device can be integrated between the metal layers [27, 28, 29] for 3D integration, the senseRRAM increases the footprint of the memory, determined by the transistor size, by a small percentage. Fig. 5.2 illustrates the typical layout for the 1T2R cell. Using W/L ratio of 3, to lower the required programming voltages for the SET operation of the 1T2R [41], the width and length of the channel of NMOS access transistor (i.e., W_{NMOS} and L_{NMOS} in fig. 5.2) are chosen to be 195 nm and 65 nm, respectively. Also, the size of the RRAM device is 105 nm x 120 nm. The increase in the cell area depends on three main foundry fabrication constraints (i.e., design rules) which are indicated in fig. 5.2 by the markers "d1", "d2", and "d3". Those design rules are not special for the RRAM device but they are related to the basic foundry design rules guaranteeing the correct fabrication of the given vias patterns (i.e., RRAM device) [132]. "d1" represents the minimum space between the metal layers used for routing the control signals SL and WE. "d2" describes the minimum area of the via layer, while "d3" determines the minimum overlap of the diffusion layer around the via. Depending on the available routing levels in the technology used to design the memory cell, the metal levels, between which the RRAM device is integrated, are chosen. For an open source NanGate 65 nm PDK with 7 routing metal-layers [133], the RRAM device is integrated between the fourth and fifth metal routing levels, respectively. The lower-level metal layers are not shown in fig. 5.2 to keep the figure clear. Also, in order to establish the opposite connectivity to the source of MOSFET device for the senseRRAM and cellRRAM devices,

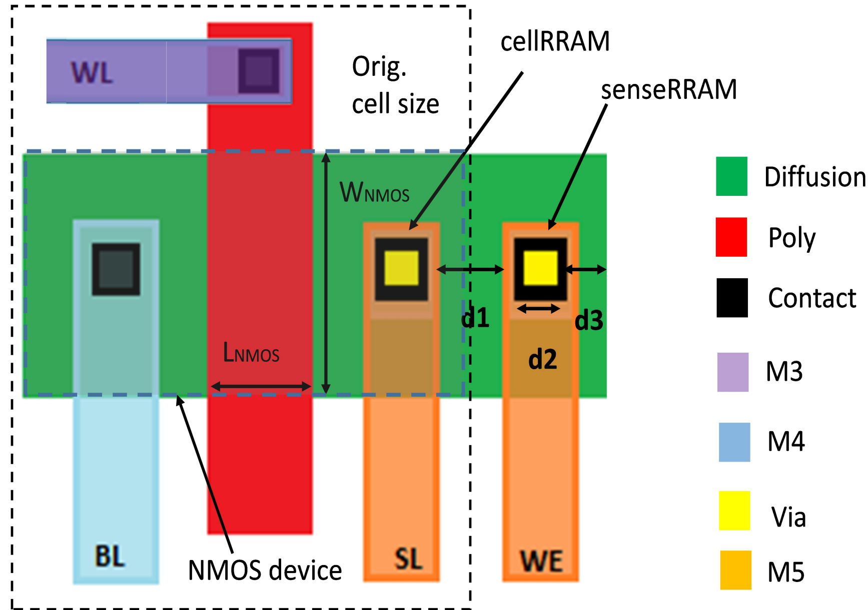

Figure 5.2: Layout of the proposed 1T2R cell. Using a 65 nm PDK, the RRAM device is integrated between metal levels 4 and 5 (i.e., M4 and M5 in the figure). Lower layers of metal (M1-M2) are omitted from figure to simplify the illustration.

instead of depositing HfO_2 then Hf layer as in the case of cellRRAM, the Hf layer for the senseRRAM is deposited before the HfO_2 layer [27, 40]. Using the design rules available in the PDK, the area of the 1T2R cell is 18% larger compared to the 1T1R cell.

Depending on the RRAM state of the half-selected cells and the write operation (SET/RESET) on the corresponding fully-selected cell, there are four cases by which the heavy-ions strikes can change the voltage across the RRAM device of the half-selected cells:

- **Case 1:** RRAM of the half-selected cell is at LRS during SET operation on the corresponding fully-selected cell.
- **Case 2:** RRAM of the half-selected cell is at HRS during SET operation on the corresponding fully-selected cell.
- **Case 3:** RRAM of the half-selected cell is at LRS during RESET operation on the corresponding fully-selected cell.
- **Case 4:** RRAM of the half-selected cell is at HRS during RESET operation on the corresponding fully-selected cell.

Fig. 5.3 illustrates the four possible cases for the half-selected cell with the different write operation modes (SET, RESET) and the corresponding biasing voltages for the WL, BL, and SL in the various scenarios.

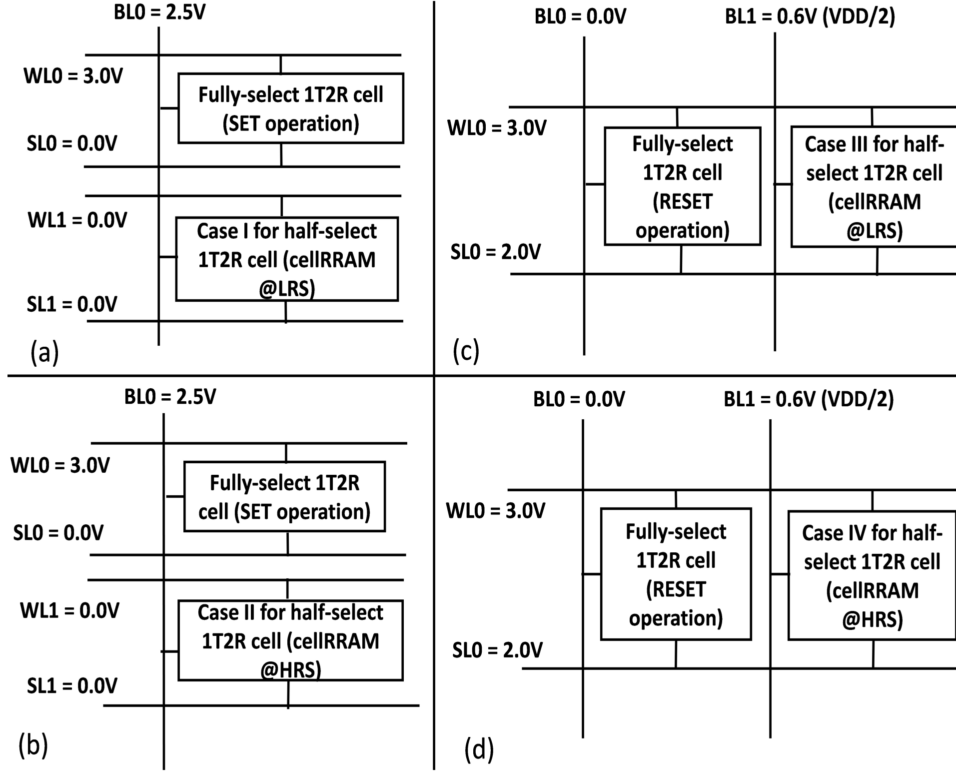

Figure 5.3: The four possible scenarios for the half-selected cells bias during the write operation. a) Case 1, b) Case 2, c) Case 3, and d) Case 4. In cases 1 and 2, the half-selected cells share the same BL voltage as the fully-selected cells, while the other control signals are connected to ground. In cases 3 and 4, the half-selected cells share the same voltage of WL and SL as the fully-selected cells, while the BL is connected to VDD/2 to prohibit modifying the RRAM state of half-selected cells.

5.2.1 Impact of Heavy-ions Strikes in Cases 1 and 2

In cases 1 and 2, the half-selected cell shares the same high potential voltage of the BL as the fully-selected cell which undergoes a SET operation. The biasing conditions of the

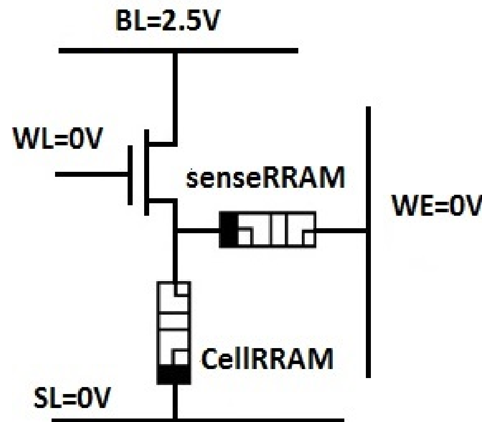

Figure 5.4: Biasing potentials on the half-selected cells in cases 1 and 2. Since the SL is always connected to ground in those cases, the maximum voltage drop across the RRAM terminals is not high enough to cause any change in its resistive state.

half-selected cell are as shown in fig. 5.4. As both WE and SL are grounded, even if highly energetic heavy-ions (i.e., Their Linear Energy Transfer (LET) is $\geq$ the threshold value of 4 $MeV.cm^2/mg$ [41]) are incident on the NMOS transistor source, the voltage drop across the cellRRAM and senseRRAM ($\approx$ 0.7V) is not high enough to trigger any change in their states (i.e., no SEU can occur).

5.2.2 Impact of Heavy-ions Strikes in Cases 3 and 4

In cases 3 and 4, the half-selected cell shares the same high potential voltage of WL and SL as those for the fully-selected cells undergoing RESET operation. As for the BL voltage of the half-selected cells, it is connected to VDD/2 to prevent them from being programmed. In those scenarios, a read operation is required for all the cells sharing the WL and SL lines to properly bias the WE signal.

Fig. 5.5 and fig. 5.6 illustrate the waveforms describing the sequence of operation for cases 3 and 4, respectively.

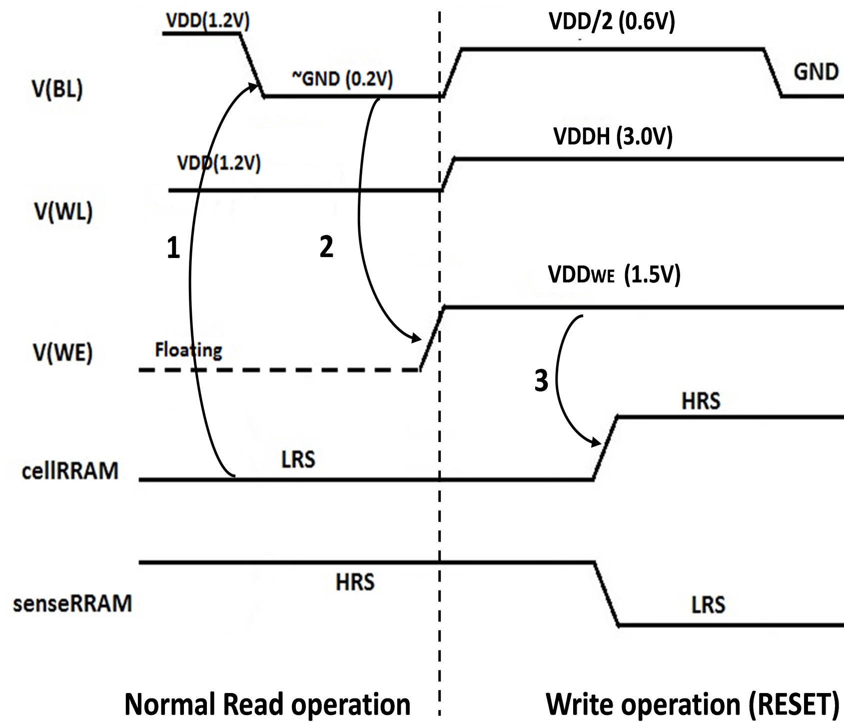

Figure 5.5: Waveforms of the control signals for the half-selected cells in case 3. The numbers in the figure represent the sequence of operations performed. A read operation is required before initiating the RESET process on the fully-selected cells to properly bias WE signal. In this figure, since the cellRRAM device of the half-selected cells is at LRS, WE is connected to high voltage (i.e., 1.5V).

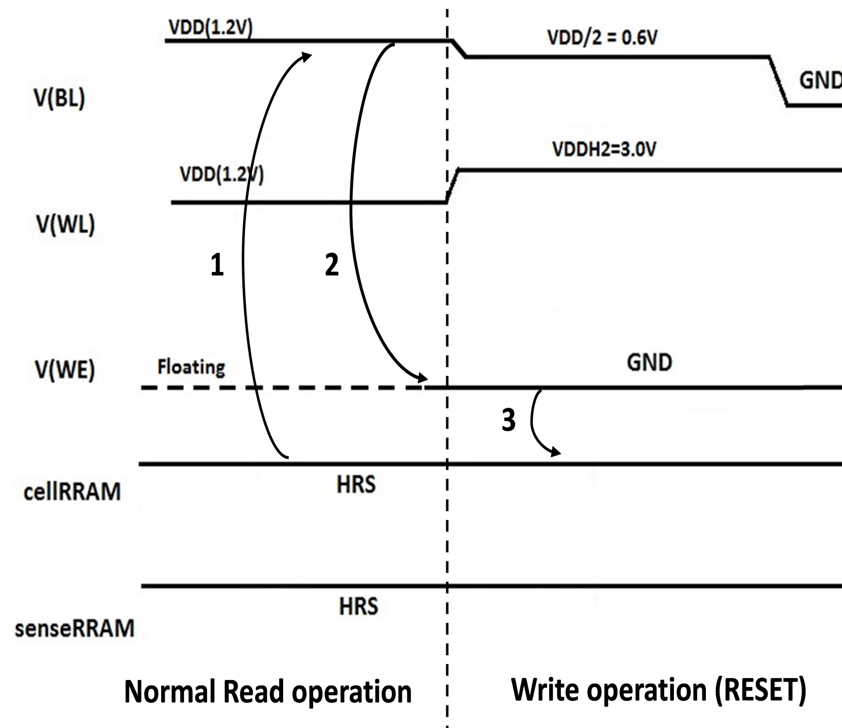

Figure 5.6: Waveforms of the control signals for the half-selected cells in case 4. The numbers in the figure represent the sequence of operations performed. Since the cellRRAM device of the half-selected cells is already at HRS, even if high energetic heavy-ions are incident on the cell, cellRRAM state remains at HRS.

The sequence of steps in fig. 5.5 and fig. 5.6 are:

1. A read operation is initiated by charging the BL to VDD while the WE signal is disconnected from the cell and kept floating. Since the cellRRAM in case 3 is at LRS, the BL discharges to a small voltage close to ground ($\approx$ 0.2V).

2. The proper voltage of WE signal is set as the inverted version of the BL voltage after the read operation. Section 5.4.1 describes how, in case 3, the high voltage of the WE can be selected.

3. If heavy-ions strikes occur in case 3, the cellRRAM switches from its LRS to its HRS while the senseRRAM changes from its HRS to its LRS. This is because the voltage drop across both of the RRAM devices is high enough to change their states. Yet,

in case 4, the voltage drop across the senseRRAM is around 0.7V, which is not high enough to alter its HRS. As for the cellRRAM in case 4, since it is already at HRS, any heavy-ions strike causes no change in its state.

Fig. 5.7 and fig. 5.8 illustrate the SPICE waveforms demonstrating the effect of heavy-ions strikes in case 3 and 4 of the half-selected cells, respectively. The part related to setting the right voltage of WE signal based on the state of the cellRRAM is discussed in details in section 5.3.2.

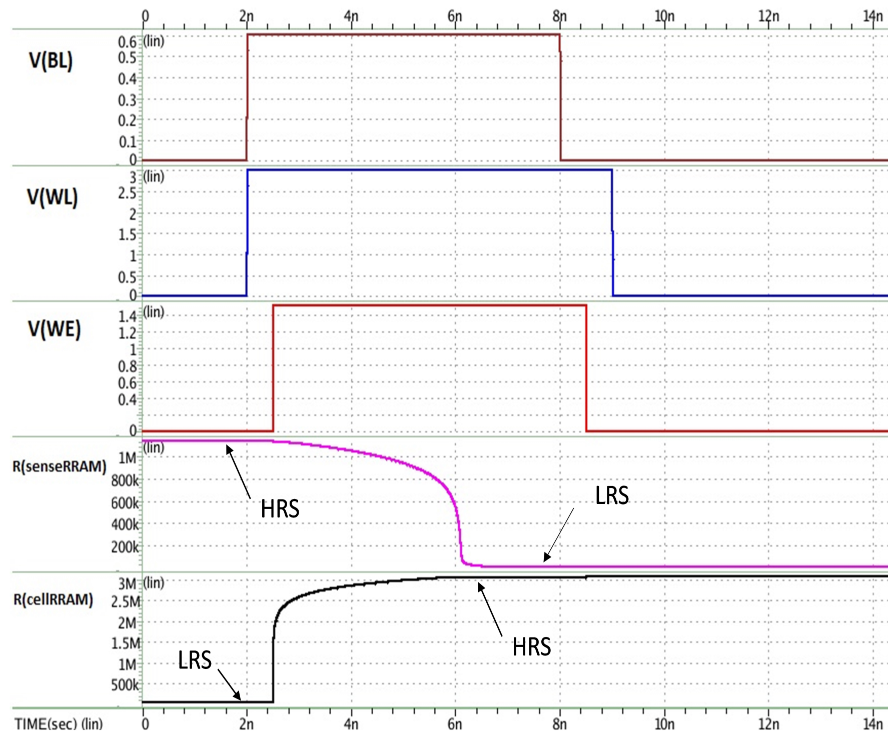

Figure 5.7: SPICE waveforms of the control signals for the half-selected cells in case 3. The SPICE simulation results demonstrate that, in case 3, if heavy-ions strikes occur, the senseRRAM and cellRRAM states will unintentionally change from HRS to LRS and from LRS to HRS, respectively. The SPICE waveforms related to setting the right voltage of WE signal is illustrated in fig. 5.14 in section 5.3.2.

Accordingly, case 3 is the only case which is affected by SEU resulting from heavy-ion strikes.

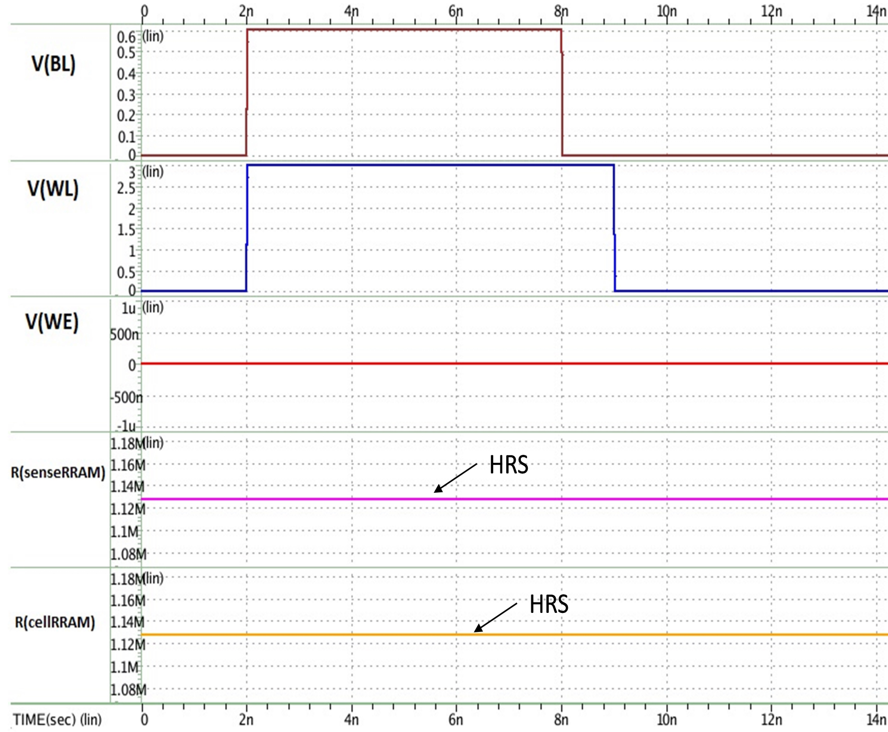

Figure 5.8: SPICE waveforms of the control signals for the half-selected cells in case 4. The SPICE simulation results demonstrate that, in case 4, the heavy-ions strike will not cause changes to the HRS of senseRRAM and cellRRAM.

Tables 5.1 and 5.2 summarize the different scenarios for the heavy-ions strikes impact on the states of the cellRRAM and senseRRAM.

The results in tables 5.1 and 5.2 demonstrate that, only in case 3 (highlighted in bold in table 5.1 and table 5.2), the senseRRAM and cellRRAM will change their resistive states.

5.2.3 Impact of the Proposed Methodology on the Write Operation

As illustrated in fig. 5.5 and fig. 5.6, the RESET operation is preceded by a normal read process to properly set the voltage of WE signal for the fully- and half-selected cells. Table 5.3 summarizes the possible voltages of WE during the write operation. Due to the HRS of the senseRRAM, the delay and energy consumption of the 1T2R memory remains

Table 5.1: Summary of the different scenarios for the heavy-ions strikes in the 1T2R cell

Scenario	**Without Heavy ions strikes**	**With Heavy ions strikes**	**SEU**
Case 1	cellRRAM=LRS , senseRRAM=HRS	cellRRAM=LRS , senseRRAM=HRS	No
Case 2	cellRRAM=HRS, senseRRAM=HRS	cellRRAM=HRS, senseRRAM=HRS	No
Case 3	**cellRRAM=LRS , senseRRAM=HRS**	**cellRRAM=HRS, senseRRAM=LRS**	**Yes**
Case 4	cellRRAM=HRS, senseRRAM=HRS	cellRRAM=HRS, senseRRAM=HRS	No

Table 5.2: Summary of the voltage across the senseRRAM and cellRRAM for the different bias scenarios of the half-selected cells

Scenario	**V(cellRRAM)**	**V(senseRRAM)**	**SEU**
Case 1	0.7V	0.7V	No
Case 2	0.7V	0.7V	No
Case 3	**2.6V**	**2.2V**	**Yes**
Case 4	2.6V	0.7V	No

practically the same as that of the 1T1R array. However, by increasing the voltage of WE during the RESET process in case 3, the delay and energy consumption of the RESET operation also increases as explained in section 5.4.2.

5.2.4 Impact of the Proposed Methodology on the Read Operation

The read operation for 1T2R memory is different from that of the 1T1R array since, not only the data stored in cellRRAM is read, but also the senseRRAM state is read to determine when SEU occurs. Hence, the read cycle is divided into two regions as illustrated in fig. 5.9. In the first region (i.e., "Upset Detection" region in fig. 5.9), only the senseRRAM state is read. If the senseRRAM is at LRS, the BL discharges to a low

Table 5.3: WE signal different bias voltage for the various write operations initiated on the fully-selected cells

Operation	**Read operation before write**	**WE bias voltage**
SET Operation, case 1 and 2	No	0V
RESET Operation, case 3	Yes	1.5V
RESET Operation, case 4	Yes	0V

voltage indicating that SEU has occurred. To restore the correct states of the senseRRAM and cellRRAM (i.e., LRS for cellRRAM and HRS for the senseRRAM), a SET operation is triggered by connecting the WE to ground and the BL and WL to high potential voltage as explained in section 5.3.
If the senseRRAM is at HRS, the BL voltage remains at VDD and the read operation proceeds to the "normal read" region, where the cellRRAM resistance is sensed. The BL in "normal read" region in fig. 5.9 drops to a low potential voltage since the cellRRAM, in this example, is at LRS. Fig. 5.10 illustrates the SPICE waveforms for the modified read operation.

If MEU occurs, it might not be observed through changes in the senseRRAM, but it can be easily detected and fixed through reading the state of the cellRRAM. Using the refresh methodology explained in chapter 4, which periodically reads the cellRRAM state, the partial drift in its resistance can be detected and fixed.

5.3 Required Modifications to the Read Circuitry

To implement the concepts discussed in section 5.2, a modified read circuit is required in order to: a) detect SEU, and b) interface with the write circuit to set the voltage of the WE signal before initiating the RESET operation. Fig. 5.11 shows the schematic for the proposed read circuit modifications. The suggested read circuit consists of three components:

- **Upset Detection Unit (UDU)**: This unit checks the state of the senseRRAM. The signal 'EN_SENSE' is used to enable the sense amplifier of this unit.

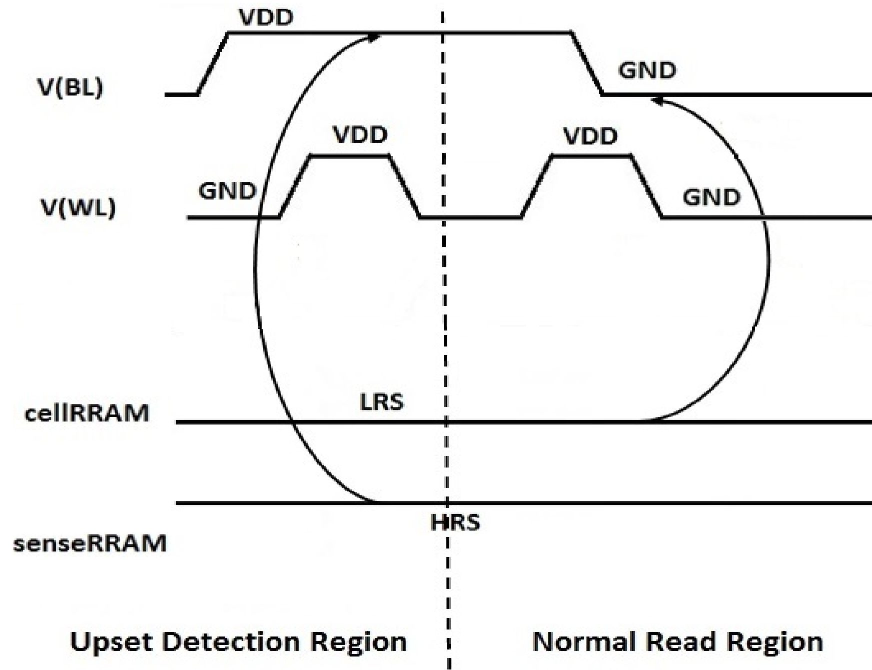

Figure 5.9: Waveforms for the modified read operation. The read process consists of two regions of operation: "Upset detection" and "normal read" regions. In this case, since the senseRRAM is at HRS, this indicates that no SEU has occurred and hence, the device proceeds to the "normal read" region.

- **Read Unit (RU)**: This is the standard read circuit for RRAM arrays with the refresh functionality, discussed in chapter 4, which detects and fixes the intermediate drift in the cellRRAM state caused by MEU or RRAM reliability soft errors.
- **WE Generation unit (WGU)**: This part sets the appropriate bias for the WE signal during the write operation as explained in section 5.2.3. The 'connect_WE' signal is used to control when to connect/disconnect the WE signal during the read operation as explained in section 5.2.4.

The sequence of operations for the suggested read circuit in fig. 5.11 can be summarized as follows:

1. The senseRRAM state is read by the UDU. If the senseRRAM is at LRS, the 'UPSET' signal is raised to VDD indicating that SEU has occurred. In this case, the

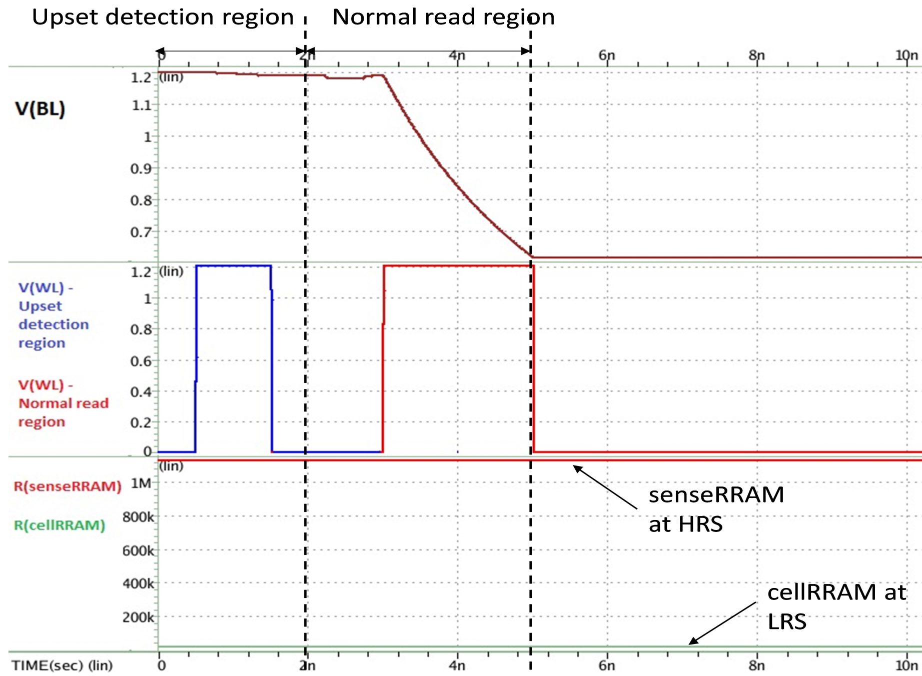

Figure 5.10: SPICE waveforms for the modified read operation in case if the senseRRAM and cellRRAM are at HRS and LRS, respectively. Since the senseRRAM is at HRS, the read process proceeds to the "Normal read region" and hence, the BL voltage discharges due to the LRS of cellRRAM.

read operation ends and a SET operation is initiated to restore the states of the senseRRAM and cellRRAM (i.e., senseRRAM state is set to HRS, while the cellRRAM state is set to LRS). Since BL is at high potential voltage (i.e., 2.5 V) and WE and SL are connected to ground, the states of the cellRRAM and senseRRAM can be reverted back to their LRS and HRS, respectively.

2. If the senseRRAM is at HRS, the enable signal 'EN_RU' is set to VDD to activate the RU. The RU has two modes of operations:

   - **Normal read operation**: In this mode, the read process is terminated after sensing the state of the cellRRAM.
   - **RESET read operation**: As explained in section 5.2.2, the cellRRAM state is read first before initiating the RESET process to set the right bias of the WE signal. In this case, the signal 'LRS_CELLRAM' in fig. 5.11 is used to enable the WGU unit. If the cellRRAM is at LRS, the 'LRS_CELLRRAM' signal is

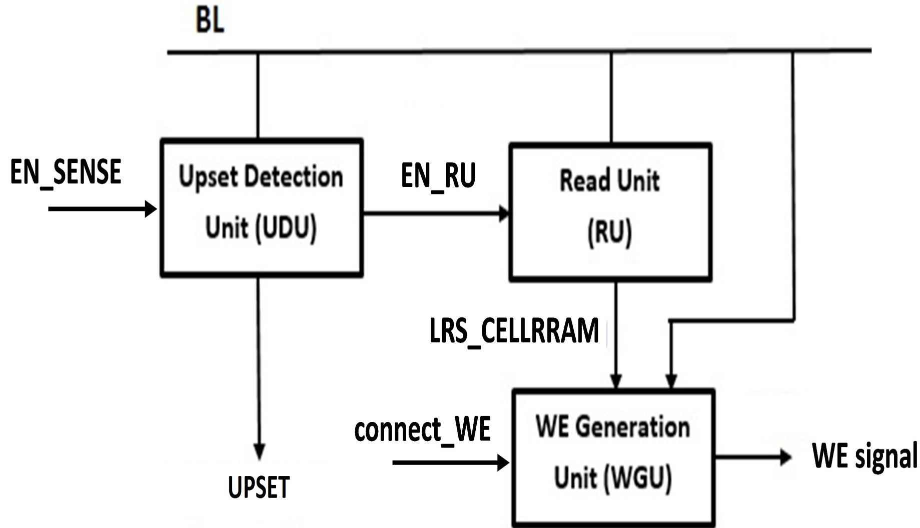

Figure 5.11: Architecture of the modified read circuit. Compared the normal read circuitry, the UDU and WGU units are added. RU is the normal read circuit, discussed in chapter 4, used to read the cellRRAM state. UDU is used to read the state of senseRRAM and trigger RU if senseRRAM is at HRS. WGU is responsible of setting up the right bias for WE signal.

raised to logic '1' and the WGU sets the voltage of WE to VDD_{WE} (i.e., 1.5V as explained in details in section 5.4.2). Otherwise, the WE signal is grounded for any other write operation.

.

In the next sections, we focus on discussing the structure of the new components, UDU and WGU, added to the standard RU unit.

5.3.1 UDU Circuit

The UDU circuit is a latch-based SA which has the structure shown in fig. 5.12 [123]. The reference voltage (V_{ref}) is chosen to correspond to the voltage of the RRAM device when it is at the upper resistance limit of its LRS. Based on the studies conducted in [38, 37, 36] for the HfO_x RRAM device, V_{ref} is chosen to be 0.65V. To enable the SA, the 'EN_sense' signal in fig. 5.12 should be raised to VDD (i.e., 1.2V) to compare the BL voltage (i.e.,

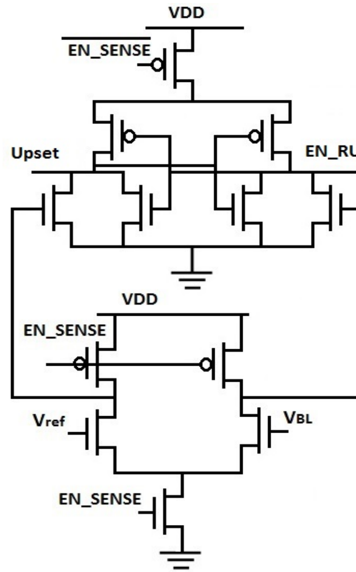

Figure 5.12: Schematic of the UDU circuit [123] which is basically a normal latch-type voltage SA.

V_{BL}) with V_{ref}. If $V_{BL} > V_{ref}$ (which means that the senseRRAM resistance is higher than that of its LRS), the signal 'EN_RU', which enables the RU unit, is set to VDD while the signal 'UPSET' is connected to ground. Oppositely, if $V_{BL} < V_{ref}$, the signal 'UPSET' is raised to VDD indicating the occurrence of SEU since the senseRRAM is at LRS. In this case, the read operation is suspended until the cellRRAM and senseRRAM states are restored to LRS and HRS, respectively.

5.3.2 WGU Circuit

Fig. 5.13 illustrates the schematic of the WGU circuit.

The circuit structure in fig. 5.13 consists of two main parts:

- **cellRRAM detection part**: This is a D Latch (DL) which stores the state of 'LRS_cellRRAM' signal from the RU unit. The DL is enabled through the control signal 'R/$\bar{W}$', which is set to VDD/ground whenever a read/write operation is initiated, respectively. Due to the architecture of the "control part" of the WGU unit, if the cellRRAM is at LRS, the output signal is at 0 V and vice versa.

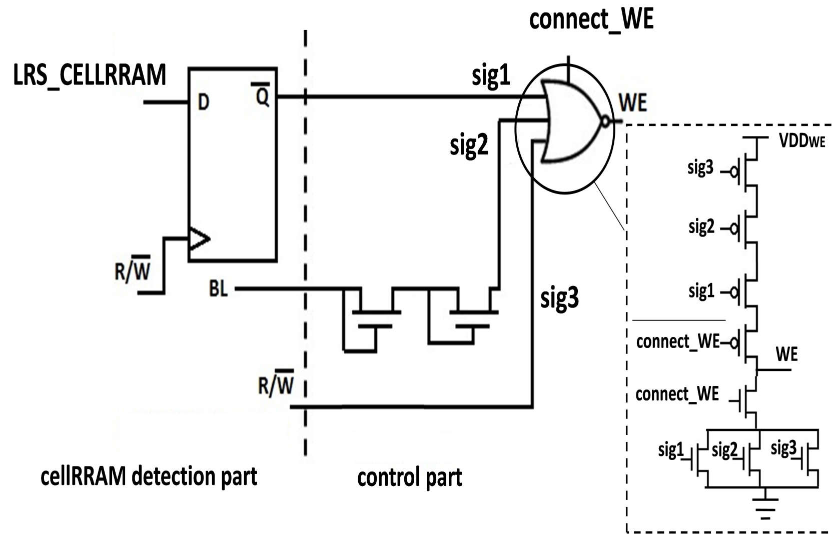

Figure 5.13: Schematic of the WGU circuit. It consists mainly of two parts: "cellRRAM detection unit" which, if cellRRAM is at LRS, the output from this unit is connected to ground, and "control part" which, based on the input from "cellRRAM detection unit" and BL voltage on the fully-selected cells, its output WE signal is either connected to ground or high voltage.

- **Control part**: This part of the WGU circuit is responsible of raising the voltage of the WE signal to VDD_{WE} (i.e., 1.5V), if the cellRRAM is at LRS and a RESET process is initiated. It is basically a 3-input NOR gate whose inputs are: a) the output signal from "cellRRAM detection part", b) a down-graded version of the BL voltage through a two diode connected NMOS devices to scale the BL voltage during the SET operation from 2.5 V range to VDD_{WE} range ($\approx$ 1.5V), and c) the same 'R/$\bar{W}$' control signal used for the "cellRRAM detection part" to distinguish between read and write operations. The enable signal 'connect_WE' connects the WE signal during the read operation. The inset in fig. 5.13 shows the CMOS structure of the 3-input NOR gate. For the write operation, the signal 'connect_WE' is set to VDD_{WE} to attach the pull-down and pull-up networks of the NOR gate. Hence, in case if: a) the cellRRAM device is at LRS (i.e., $\bar{Q}$ is at logic '0'), b) the BL is not at high potential voltage indicating a RESET operation is initiated, and c) a write operation is initiated (i.e., 'R/$\bar{W}$' is at logic '0'), the WE voltage is raised to VDD_{WE}. Otherwise, the WE signal is connected to 0 V during the write operation. As for the read operation, when the cellRRAM state is read, the 'connect_WE' signal

is grounded to set the WE voltage to high-impedance state (i.e., floating state) during the “Normal Read region” in fig. 5.9.

Fig. 5.14 illustrates the SPICE waveforms for the WGU unit when the cellRRAM of the half-selected cell is at LRS and a RESET operation is being initiated.

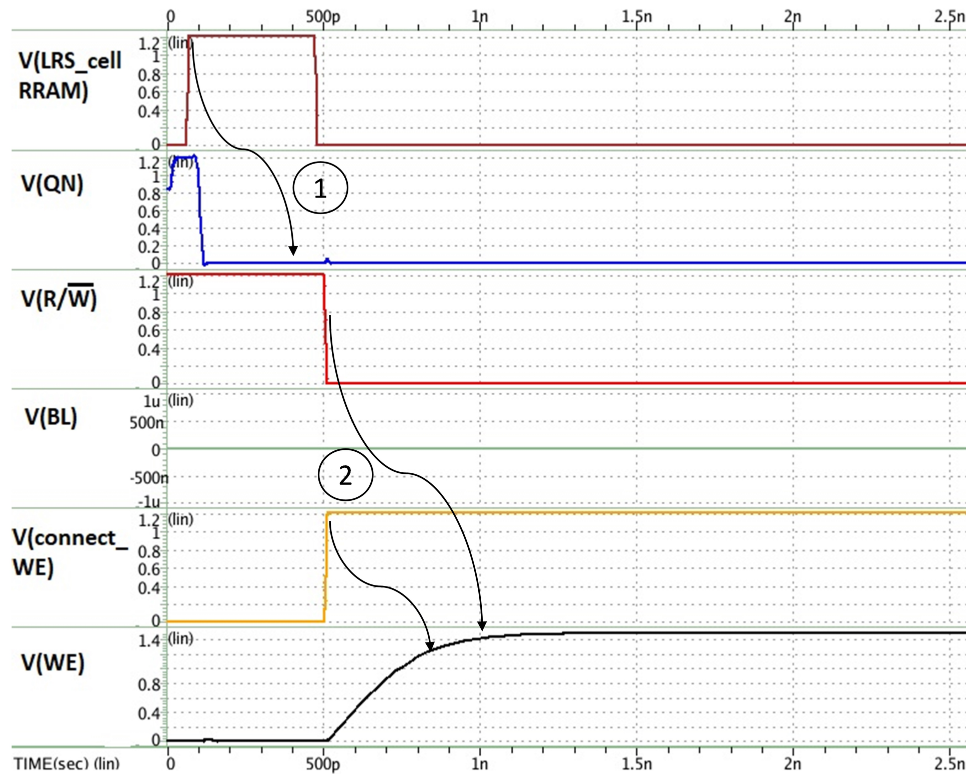

Figure 5.14: SPICE waveforms describing the WGU operation. Since the voltage of BL is at low voltage to enable the RESET process and the read state of cellRRAM indicates that it is at LRS, the WE voltage is raised to 1.5 V.

The number sequence in fig. 5.14 represents the basic two steps of WGU operation:

1. The cellRRAM state is sensed first by initiating a read operation (i.e., ‘R/$\bar{W}$’ is at logic state ‘1’). Since the cellRRAM is at LRS, the signal “LRS_CELLRRAM” is raised to VDD causing the voltage of the node ‘QN’ to be set to 0 V.

2. Since a RESET process is initiated by having the voltages of BL and ‘R/$\bar{W}$’ connected to ground, the signal ‘connect_WE’ is raised to VDD to connect the WE signal to high voltage (i.e., 1.5 V) as discussed in section 5.2.

5.4 Simulation Results

Fig. 5.15 illustrates the flow of simulation performed to evaluate the functionality of 1T2R cell and the performance of its read and write operations in comparison to those in 1T1R RRAM arrays.

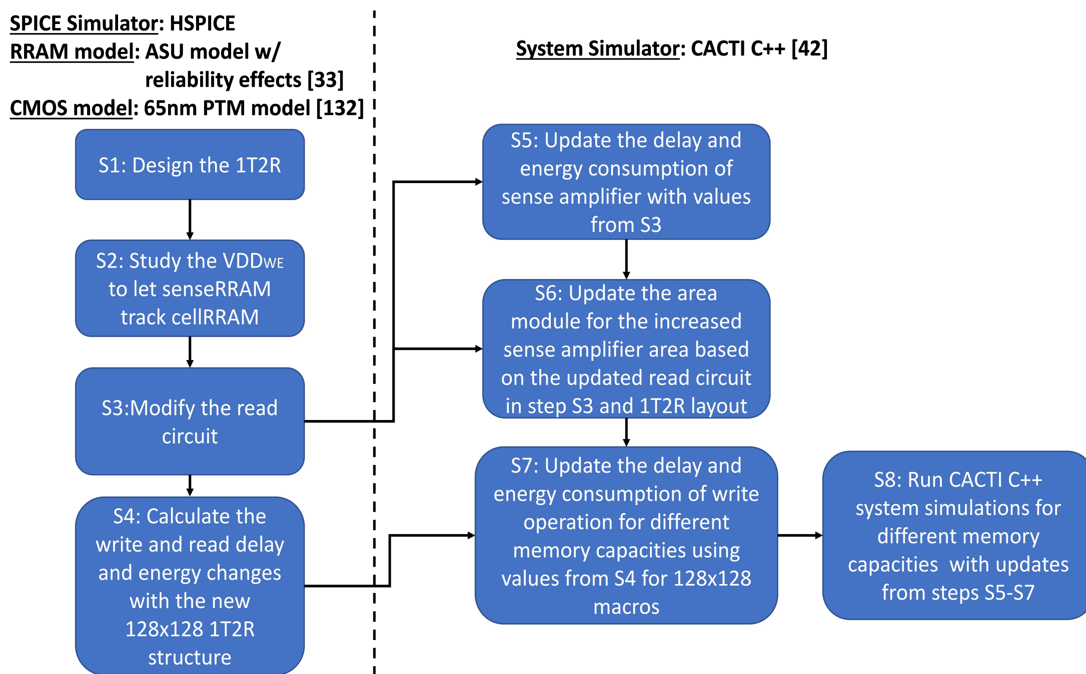

Figure 5.15: Block diagram for the simulation runs and the tools used to evaluate the performance of the 1T2R RRAM arrays. The ASU model [33] together with the methodology described in [41] are used to run SPICE simulations, while CACTI C++ files [42] are used for system level simulations.

The first step of the simulation flow is to validate the correct operation of the proposed 1T2R cell in its different modes of operations (i.e., read, write, and when SEU occur during write operations) through multiple runs of HSPICE [124] (i.e., step S1 in fig. 5.15). Then,

the study of WE high voltage bias is conducted to: a) reduce its impact on increasing the write operation delay and energy consumption, and b) make sure that, when SEU occurs, the senseRRAM state can track the changes in cellRRAM state (i.e., step S2 in fig. 5.15). After this, in step S3, the read circuit is modified as described in section 5.3 and its operation is verified through HSPICE simulations. Also, during step S3, the appropriate duration for the "upset detect" and "normal read" regions is chosen to guarantee that the senseRRAM state is correctly read, while the impact on read operation delay is minimized. Following this, a 128x128 array of 1T2R is formed to compute the delay and energy consumption of the write and read operation of the new RRAM array and compare it with the results of 1T1R arrays (i.e., step S4 in fig. 5.15). Using the simulation results for the modified read operation in step S3, the delay and energy consumption of the SA module in the CACTI C++ files are modified (i.e, step S5 in fig. 5.15) to estimate the expected increase in the read delay and energy consumption of large 1T2R arrays in comparison to those of the 1T1R memories. Moreover, using the sizes of PMOS and NMOS transistors for the newly added WGU from the HSPICE netlist in step S3 and using the 1T2R cell layout illustrated in fig. 5.2, the area module of the CACTI C++ files is modified by increasing the size and number of PMOS and NMOS devices used (i.e., step S6 in fig. 5.15). This is needed to estimate the increase in the area of large 1T2R arrays due to the modified read circuit in comparison to that of 1T1R arrays. After that, using the HSPICE results for the 128x128 macro in step S4, the bitlines parameters during the write operation in the CACTI C++ files are modified by: a) reducing the pulse duration from 10 ns to 6.4 ns as a result of using 128x128 macros compared to 1024x1024 macro in [33, 41], and b) account for energy and delay of read operation before the RESET process. This is basically done to estimate the increase in the write energy and delay of large 1T2R arrays in comparison to those of the 1T1R arrays. Finally, multiple system simulations for memories of various capacities are run in step S8 using all the modifications described in steps S5-S7.

The HSPICE simulation setup in this section is based on the results published in [40, 41]. In [40], various lab experiments and simulations are run to understand the root causes for the SEE in 1T1R memory arrays. Using the experimental results in [40], the authors in [41] propose a SPICE simulation technique, illustrated in fig. 2.17, which describes how to model the SEE in circuit simulations. In our analysis, the SPICE modeling technique in [41] is used while considering the LRS and HRS range of the HfO_x RRAM device to be < 20 kΩ and > 200 kΩ, respectively. Changing the RRAM device integrated in the 1T2R array, or changing the voltages used in the memory system, could alter the HRS-LRS resistance range. However, this does not change the main concepts behind the design proposed in this chapter or the generality of the simulation results discussed in this section. The detailed results for the modified memory array can be obtained by repeating

the experiments in this work with the adjusted voltage levels.

5.4.1 Selection of the High Voltage of the WE Signal (VDD_{WE})

The high voltage of the WE signal (i.e., VDD_{WE}) has to be properly chosen so that: a) the change in senseRRAM state can track that in the cellRRAM state when SEU occurs, and b) minimize the impact of WE voltage on the write energy consumption and delay of the new 1T2R memory cell.
Fig. 5.16 demonstrates how the VDD_{WE} voltage impacts the change in senseRRAM state in response to SEU. The "Min cellRRAM LET" and "Min senseRRAM LET" curves in fig.

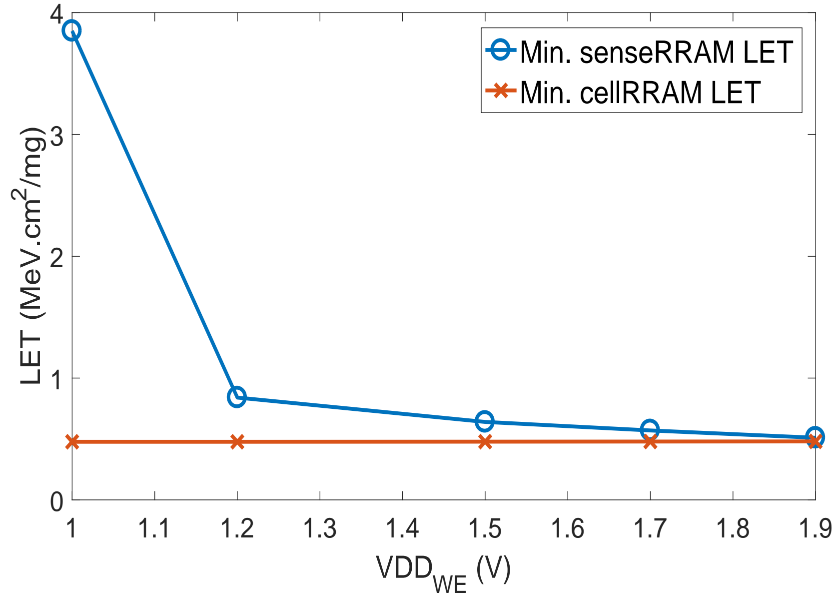

Figure 5.16: Effect of VDD_{WE} voltage on the correctness of our proposed methodology in detecting the upset events. Proper voltage of WE is chosen such that the impact on RESET operation is minimized, and at the same time, the senseRRAM state can track the changes in cellRRAM state.

5.16 refer to the minimum threshold LET required for the cellRRAM and senseRRAM to change their states, respectively. If the LET of the incident heavy-ions is $\geq 4\ MeV.cm^2/mg$, both the senseRRAM and cellRRAM of the half-selected cells will unintentionally change their state. However, for heavy-ions strikes with LET as low as 0.5 $MeV.cm^2/mg$ and if

$VDD_{WE} = 1.0$ V, the cellRRAM state of the half-selected cell switches from LRS to HRS, while the senseRRAM state remains at HRS. Only when the LET of heavy-ions strikes is $\geq 3.85\ MeV.cm^2/mg$, the senseRRAM state of the half-selected cells will switch from its HRS to LRS, indicating that SEU has occurred. By increasing VDD_{WE} to a voltage close to that of SL (i.e., 1.9 V in our example), the difference between the minimum threshold LET causing the cellRRAM and senseRRAM to switch their states becomes smaller.

However, fig. 5.17 shows how increasing VDD_{WE} can negatively impact the delay and energy consumption of the RESET process. The "Orig.Write Delay" and "Orig.Write

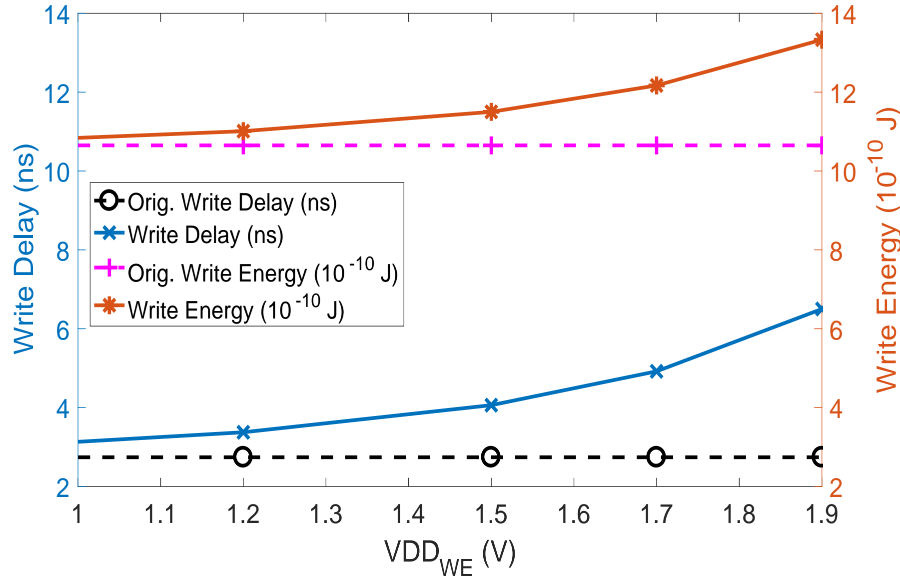

Figure 5.17: Effect of VDD_{WE} voltage on the performance of RESET operation. Increasing VDD_{WE} increases the delay and energy consumption of RESET process exponentially.

Energy" curves refer to the delay and energy consumption of the RESET process for the conventional 1T1R cell. Fig. 5.17 demonstrates that the RESET delay and energy consumption increase with VDD_{WE}. The increase in the delay is caused by the current injected by WE signal which combats the programming current from the SL signal. The higher the WE current the longer is the time required to complete the RESET process. Due to: a) the extension in the duration of the RESET operation, and b) the increase in the total current drawn from the control signals WE and SL, the average energy consumed during RESET also increases. Based on the results shown in fig. 5.16 and fig. 5.17, the WE high voltage is set to 1.5 V as a compromise between minimizing the impact of

VDD_{WE} on the RESET process and guaranteeing a correct operation of the proposed SEU detection technique. Choosing the VDD_{WE} voltage to be at 1.5 V results in reducing the difference between the minimum required LET for the incident heavy-ions to switch the states of cellRRAM and senseRRAM to ≈ 0.09 $MeV.cm^2/mg$. It is worth mentioning that, if the LET the heavy-ions strikes is within the 0.09 $MeV.cm^2/mg$ gap difference, the senseRRAM state still changes towards LRS (≈ 25 kΩ compared to 200 kΩ at HRS) but not to the minimum LRS (10 kΩ)).

5.4.2 Simulation Results for the Write Operation

Using a 65 nm PTM model [134] and W/L ratio of 3, to lower the required programming voltages for the SET operation of the 1T2R [41], the SET and RESET biasing conditions are as follows:

- **SET Operation**: WL = 3.0 V, BL = 2.5 V, SL = 0 V.
- **RESET Operation**: WL = 3.0 V, BL = 0 V, SL = 1.9 V.

The SET operation uses a higher voltage biasing conditions to account for the voltage drop across the NMOS access transistor [41]. For a RESET process, a read operation precedes initiating the write operation to properly set the WE voltage. In this section, we focus on the write operation performance after the voltage of WE signal is properly set (i.e., 0 V during the SET process and 1.5 V during the RESET process). Then, in section 5.4.3, the read operation performance is discussed. Table 5.4 summarizes the comparison results between the write delay and energy consumption of 128x128 1T1R [33] and 1T2R arrays.

Table 5.4: Comparison of the write performance for 1T1R and 1T2R 128x128 arrays

Operation	**1T1R cell**	**1T2R cell**	**Percentage of Change**
SET Energy (pJ)	665.75	644.42	-3.3%
SET delay (ns)	3.173	3.59	13.1%
RESET Energy (pJ)	106.514	115.11	8.1%
RESET delay (ns)	1.4	1.56	11.43%

Table 5.4 shows that the delay for SET and RESET operations increases due to the extra WE control signal. For the SET operation, since the WE voltage is connected to ground, the programming current drawn from BL is divided between the cellRRAM and senseRRAM devices, which both are at HRS. This also increases the voltage of the NMOS source terminal lowering the current drive of the transistor. Hence, for a fixed programming pulse (i.e., 10 ns in [40, 41, 33]), the portion of this pulse, during which both of the RRAM devices are at HRS, is increased. By consequence, the amount of current drawn from the BL during the fixed programming pulse has decreased and accordingly the energy consumption is also reduced. The fixed pulse duration is required to account for the delay introduced by the wire capacitance in the memory array [40, 41, 33]. Oppositely, for the RESET operation, since the time, during which the cellRRAM is at LRS, is prolonged, the energy consumption increases.

In order to minimize the impact of the read process initiated before the RESET operation, the size of the memory macro is reduced from 1024x1024, as in [41], to 128x128. This helps decreasing the required write pulse width to almost 6.4 ns and accordingly, the write process delay, together with the extra read operation, can remain as is (i.e., 10 ns). Taking into account the read operation, the write energy for a 128x128 array is increased by 19% due to the extra WGU and UDU units added to the read circuit. However, when the 128x128 macro is integrated in the design of 8 Gb memory, the net increase in the energy consumption for the write operation is less than 0.2% due to the overhead from the other memory components such as the address decoders and the repeaters.

5.4.3 Simulation Results for the Read Operation

As discussed in section 5.2.4, the read cycle in our proposed methodology is divided into: “upset detection period” (i.e., T_{UD}) and “normal read period” (i.e., T_{norm}). Longer T_{UD} is required in order to have better separation between the read voltage levels for the different senseRRAM states. Yet, fig. 5.18 shows that increasing T_{UD} can rapidly increase the energy consumption specially if the senseRRAM is at LRS (i.e., when SEU has caused the senseRRAM to switch its HRS to LRS). The V_{sense} curve in fig. 5.18 defines the difference between the read voltage on the BL when the senseRRAM is at HRS and when it is at LRS. Based on the simulation results data illustrated in fig. 5.18, the duration of T_{UD} is set to 1 ns since the minimum sensing voltage difference should be in the range between 0.02V and 0.1V for a bulk CMOS technologies [135].

There are three main differences in our suggested read circuitry compared the one in [136]:

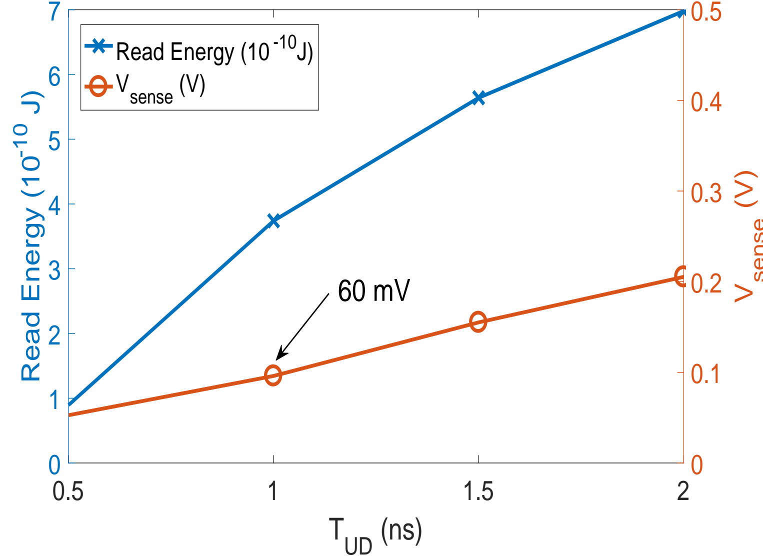

Figure 5.18: Impact of increasing the T_{UD} on detecting the change in senseRRAM state. "Read Energy" curve is calculated assuming worst case when the senseRRAM is at LRS. Increasing T_{UD} enhances the difference between the read voltage by SA of UDU when the senseRRAM is at HRS and when it is at LRS (i.e., V_{sense} in the figure). However, it also negatively affects the ability of SA of RU in correctly reading the state of cellRRAM.

- Adding one extra sense amplifier for the UDU unit.
- Reducing the pulse width for reading the cellRRAM state from 3 ns to 2 ns (i.e., T_{norm}).
- Adding the WGU unit at the output of the read circuit as discussed in section 5.3.2.

Our SPICE simulation results show that the energy consumption of the read operation is increased by only 18% compared to the one used for the 1T1R array. This is due to: a) the reduction in the duration of T_{norm} from 3 ns to 2 ns, which reduces the current discharging the BL during the read cycle, b) the small current passing through the senseRRAM during T_{UD} since it is at HRS, and c) the impact of the WGU on the increase in read energy consumption is minimal (i.e., $< 2\%$).

5.4.4 System Level Simulation Results

Using the CACTI C++ files [42] and the results for the write and read operations for the 128x128 memory block, the energy consumption of the bitlines and SA are modified to assess the impact of the newly proposed methodology on the performance of various memory arrays with different capacities. The main modifications to the C++ files are:

- Change the write pulse 6.4 ns instead of 10 ns.
- Account for the energy consumption and delay for the read operation before the RESET process.
- Split the read operation into two separate regions (i.e., “upset detection” and “normal read” regions as explained in section 5.2.4) and compute the energy consumption of each process separately then combining them.

Fig. 5.19 shows that, for a 1T2R memory array as large as 8 Gb, the increase in the energy consumption is only 0.2% and 0.1% for the read and write operations, respectively in comparison to the results of 8 Gb 1T1R memory.

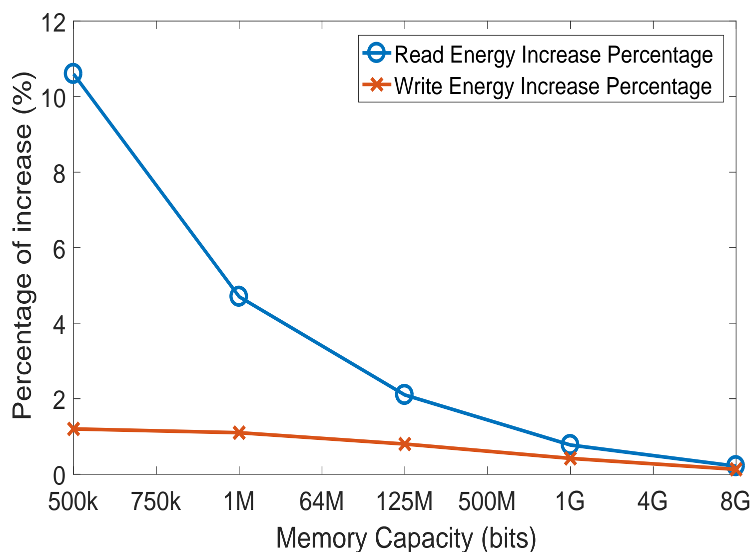

Figure 5.19: System level simulation for the increase in the energy consumption of 1T2R memory arrays with different capacities. The “percentage of increase” axis in figure refers to the change in the energy consumption of read and write operations in comparison to those of 1T1R arrays with the same capacity.

This is because the energy increase caused by the overhead of the other peripheral circuits (i.e., address decoders, multiplexers, and repeaters) of the memory system is much higher than that resulting from the suggested adjustments in the read and write circuits. Hence, the smaller the size of the memory array, the greater is the increase in the energy consumption. Although more peripheral circuits are used compared to the case of 1T1R array, due to the decrease in macro block size from 1024x1024 to 128x128, the increase in energy is minimal. This is mainly because of the fact that the decrease in write pulse duration from 10 ns to 6.4 ns reduces the write energy of the 1T2R 8 Gb memory array by more than 27% compared to that of the 8 Gb 1T1R array.

It is worth mentioning that having more sensitive peripheral circuits to SEE require deploying more radiation hardening techniques. On the system level, using Error Correction Codes (ECC) to detect and fix the induced soft-errors is required [137, 138]. On the circuit level, many techniques can be incorporated including the usage of radiation tolerant fabrication technologies such as Fully-Depleted Silicon On Insulator (FDSOI) [139]. Although using FDSOI technology can increase the 1T2R cell tolerance to radiation soft-errors to about 50 $MeV.cm^2/mg$ [139], if the 1T2R memory array is deployed in applications subject to higher radiations, our proposed methodology for detecting and fixing SEU will still be useful. As for the the peripheral circuits, their layout maybe modified to account for SEE as discussed in [140, 141].

By modifying the area calculation for the SA and for the cell dimensions in the area module of the CACTI C++ files to take into account the extra added blocks (i.e., UDU and WGU circuits), fig. 5.20 shows the CACTI simulation results for the estimated percentage of increase in the chip area.

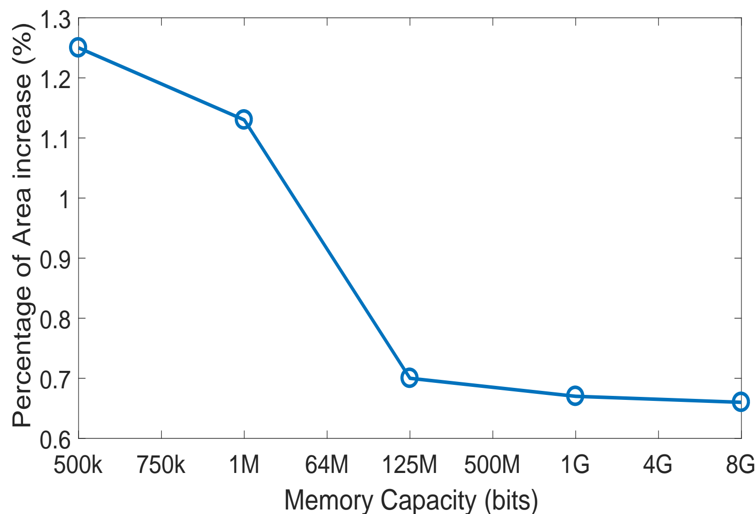

Figure 5.20: System level simulation for the impact of modified read circuit on the chip area of 1T2R memory arrays with different capacities. The "percentage of increase" in the figure refers to the change in chip area compared to the case when large 1T1R arrays are used instead.

Fig. 5.20 demonstrates that the chip area increase is about 0.66% for an 8 Gb array since the contribution from the other memory components (i.e., decoders, repeaters, and multiplexers) in the area calculation is much higher than that caused by the read circuit modifications. Similar to the results in fig. 5.19, the chip area increases with the decrease in the memory capacity due to the reduced effect of components, such as decoders and multiplexers, on the total chip area. The sudden increase in the chip area when the memory capacity is reduced from 125 Mb to 1 Mb is due to the sudden decrease in the number of sub-banks used in the memory array similar to what is illustrated in fig. 4.9 described in section 4.4.2.

5.5 Summary

In this chapter, a novel methodology is discussed to detect and fix radiation soft-errors resulting from SEU in 1T1R RRAM memory arrays. The SPICE and system level simulation results show that, for an 8 Gb 1T1R memory array, implementing the suggested technique increases the energy consumption of the read and write operations by only +0.2% and

+0.1%, respectively. Moreover, the increase in the chip area of 8 Gb memory, deploying the suggested modifications, is as low as +0.66%. With the integration of the refresh circuit discussed in chapter 4, the radiation soft-errors, causing either MEU or SEU in 1T1R arrays incorporated in neuromorphic systems, can be detected and fixed.

Chapter 6

Addressing the RRAM Reliability Soft-Errors in Neuromorphic Systems

In this chapter, the effect of RRAM soft-errors on the operation of RRAM-based neuromorphic systems is discussed. Without considering RRAM soft-errors, the actual performance of neuromorphic systems can be significantly degraded as discussed in section 6.2. For correct and accurate operation of the neuromorphic systems, this degradation has to be detected and fixed. Hence, in sections 6.4 and 6.5, we provide two possible techniques for doing this. The required modifications of the read and write circuits are also discussed in sections 6.7 and 6.8. It is worth mentioning that, although the proposed methodologies are verified with the multi-perceptron system architecture discussed in [142], the concepts behind the suggested solutions should still apply for another system architecture. However, it is expected that the qualitative results may change. Moreover, we use a generic way for studying the effect of RRAM soft-errors on the neuromorphic system performance by assuming the worst-case scenario where all the RRAM cells are suffering from soft-errors during the training cycle. A more detailed analysis is required in future work to study a less pessimistic case which accounts for the exact sequence of patterns applied to the system. In summary, the main contributions from the work presented in this chapter are: a) for the first time, a systematic framework is suggested to assess the effect of RRAM soft-errors on the overall RRAM-based neuromorphic systems, b) developing two new system level algorithms, using widely-used MNIST test benches [44], to detect and fix the degradation in neuromorphic system performance due to RRAM soft-errors, and c) providing the required circuit modifications to support the proposed system level methodologies and verifying their proper operations using experimentally-verified SPICE models [32, 33]. Those contributions provide an initial step of how the futuristic RRAM-based neuromorphic system can

be reliably used in advanced platforms running machine learning applications.

6.1 Introduction

With the increasingly aggressive need to perform more complex operations (i.e., cognitive operations) in an energy-efficient manner, the interest in neuromorphic systems has greatly boosted in recent years. A neuromorphic system consists of a large number of pre- and post-neurons which are connected together through a large network made of synapses as shown in fig. 6.1. Depending on the strength of the connection (i.e., status of the synapses),

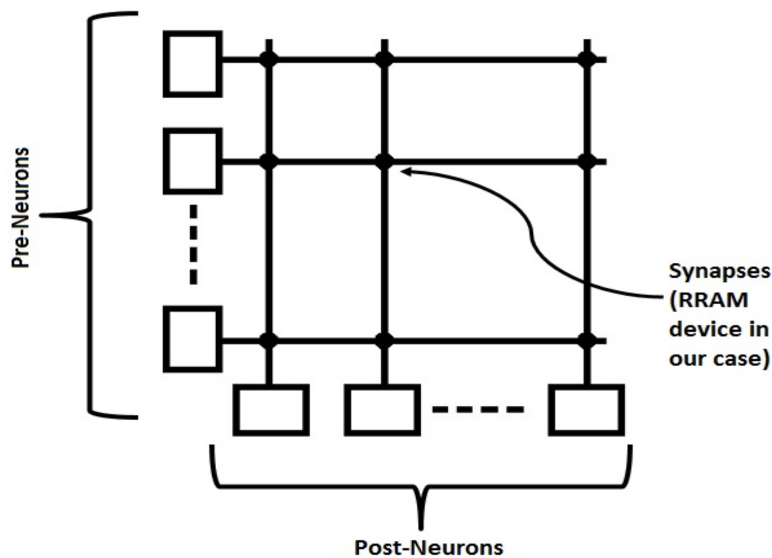

Figure 6.1: General structure of a neuromorphic system where pre-neuorns are connected to post-neurons through a dense network made of synapses. The RRAM device is used in the synaptic network implementation due to its small size, low programming requirements, and its ability to be programmed to intermediate states depending on the pre- and post-neurons pulses.

the signal can be either transmitted from the pre-neurons to the post-neurons or blocked. The neuromorphic system has two modes of operation:

- **Training (Learning) cycle**: During this mode of operation, known input patterns are applied to the system according to a learning algorithm to program the synaptic device to recognize those patterns.

- **Testing cycle**: After the training cycle, other unknown patterns are introduced to system to evaluate its performance based on the correctness in recognizing the newly applied patterns.

In order to reduce the power consumption and area of the neuromorphic systems, NVM devices, such as PCRAM [143], STT-MRAM [144], and RRAM [29, 46], have been proposed to be used in the design of synapses cells. All those devices are small in size, have non-volatile capability, and have a high and low resistive states which can be programmed according to the applied input patterns. Due to its advantages explained in section 2.4, RRAM device has been extensively used in the design of crossbar memory arrays incorporated in neuromorphic systems [18, 19, 20, 21, 22, 23, 24, 25, 26].

Depending on the RRAM device used, proper learning algorithm and system structure are chosen. For example, in order to use the sparse coding algorithm, the incorporated RRAM device needs to have a more linear I-V characteristics and hence the TaO_x device is preferred [23, 105]. Despite the linearity of the TaO_x RRAM device, due to the lack of having access to the reliability models of TaO_x RRAM devices and the availability of those for the HfO_x RRAM device, the multi-perceptron neuromorphic systems, which incorporate the HfO_x RRAM device as synapse, is used in our various analysis in this chapter. The structure of those systems rely on having multiple layers of neurons and large number of connections to accommodate the non-linear characteristics of the HfO_x RRAM device [22, 145]. This is in addition to the fact that the HfO_x RRAM device uses lower programming voltages ($\approx$ 1.4 V [33]) compared to those needed by the TaO_x RRAM ($\geq$ 2 V) [146, 147, 148], which makes the HfO_x RRAM devices more attractive to use in low-power neuromorphic systems. Moreover, the multi-perceptron systems are the simplest, most efficient, and commonly used neuromorphic structure to classify various input patterns [149, 150, 151, 152].
Yet, all the concepts and methodologies discussed in this chapter can be applied to any other RRAM-based neuromorphic system whenever the reliability models for the incorporated RRAM device are available.

In this chapter, a multi-perceptron system, described in [142], is used as a case-study to conduct our various experiments and analysis. The system structure is illustrated in fig. 6.2. This is an unsupervised multi-perceptron spiking neuromorphic system used to recognize the MNIST dataset [44]. Other than its ability to tolerate the non-linear characteristics of the HfO_x RRAM device, this system is chosen for multiple reasons:

- It achieves high accuracy in recognizing the MNIST dataset with only one hidden layer which simplifies our soft-errors reliability analysis.

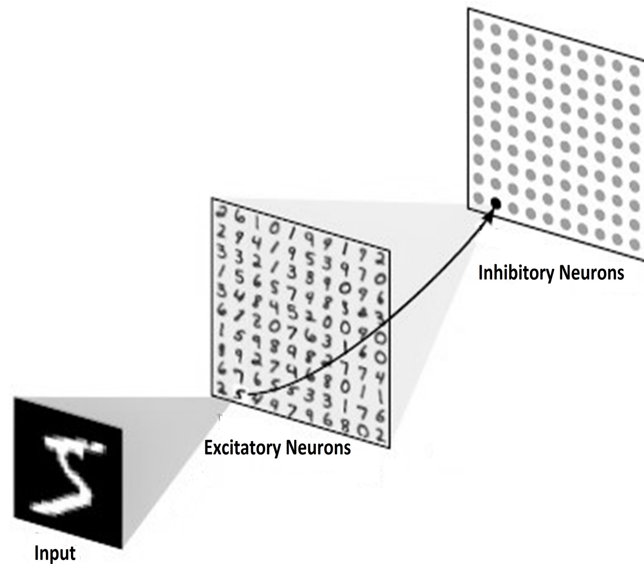

Figure 6.2: Structure of the neuromorphic system used in our studies [142] which classifies the handwritten digits defined in MNIST dataset [153].

- Unlike the other unsupervised neuromorphic systems, which use supervised methodology to classify the trained patterns during the testing phase [154], the system in [142] dynamically assigns a class to each neuron based on the highest spike response to the training set. Hence, the testing phase in the system in [142] is also done in an unsupervised manner.
- Due to the unsupervised classification process, the system is highly scalable and can achieve up to 96% accuracy in recognizing the MNIST dataset by just increasing the neurons count [142].

As shown in fig. 6.2, the system consists of two main layers of excitatory neurons. The first layer is the input layer and contains 28x28 neurons (one neuron per image pixel of MNIST dataset). The second layer is the processing layer and it is composed of 400 excitatory and inhibitory neurons. Each input digit is introduced to the system as a Poisson spike-train for 350 ms window where the rate of pulses depends on the intensity of each pixel of the applied image. The excitatory neurons of the second layer are connected in a one-to-one fashion to inhibitory neurons, which means that a spike on an excitatory neuron triggers a spike on its corresponding inhibitory neuron. Each of the inhibitory neurons is connected to all excitatory ones, except for the one from which it receives a connection. Hence, when

a pulse from an excitatory neuron occurs, the inhibitory neuron, to which it is connected, triggers pulses to all the other neurons to prohibit them from responding to the applied pattern. This method is called the Winner-Takes-All (WTA) and it is used to localize the learning process to the "most-likely" neurons, which are those who respond first to the input pattern.

In case if the number of pulses generated by the excitatory neurons of the system in response to an input pattern is less than a threshold value (i.e., 5 spikes as described in [142]), this specific input pattern is re-applied while reducing the rate of the Poisson input pulses. As explained by the authors in [142], this increases the chance of having a time overlap between the input and excitatory neurons, which leads to better programming of the synapses connecting them.

RRAM devices are used in the connection between the input neurons (i.e., neurons of the first layer or pre-neurons) and the excitatory neurons of second layer (i.e., post-neurons) in fig. 6.2. The RRAM state changes according to Spike-Timing-Dependent Plasticity (STDP) rule [155], which can be summarized as follows:

- If the post-neurons triggers a pulse (i.e., spike with 20 ms duration) in response to a pulse in the input pattern, such that, the net voltage drop across the RRAM is positively high enough to trigger a change in its state, the connection between those two neurons is enhanced (i.e., synapse weight is increased which corresponds to decreasing the RRAM resistance towards its LRS).

- If input pulses occur while spikes on the post-neurons are still applied, such that, the net voltage drop across the RRAM is negatively high enough to trigger a change in its state, the connection is weakened (i.e., synapse weight is decreased which corresponds to the increase in the RRAM resistance towards its HRS).

The exact change in the synapse weight (i.e., shift in the RRAM resistive state) depends on the time difference between the pulses on the pre- and post-neurons. Fig. 6.3 [22] illustrates the STDP rule and how the RRAM resistive state changes based on the overlap between the pre- and post-neurons pulses (i.e., spikes). As described in fig. 6.3, the neurons exponential spikes are approximated using a sequence of rectangular pulses with decaying amplitudes.

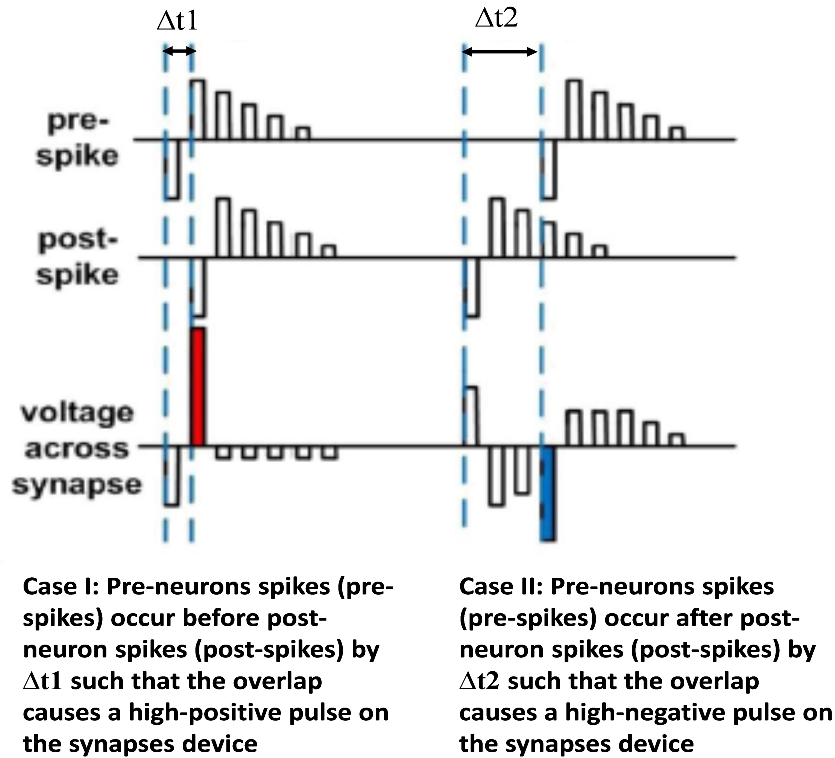

Figure 6.3: Illustration of the STDP rule and how the resistive state of the RRAM device changes accordingly [22].

One efficient way for implementing the WTA method is discussed in [156]. The proposed WTA circuit schematic is illustrated in fig. 6.4. The main idea behind the circuit in fig. 6.4 is to combine the inhibitory functionality with the excitatory neurons to save layout area and reduce the delay and energy overhead [156]. Each of the excitatory Integrate and Fire (I&F) neurons in fig. 6.4 is attached to a strong NMOS pull-down transistor. The control line 'RESET_CTRL' is kept at high potential voltage through an "always-on" weak PMOS device. Whenever any of the neurons produces an output pulse, the NMOS transistor, connected to it, discharges the control line 'RESET_CTRL'. Accordingly, the inverter in fig. 6.4 raises the 'RESET' signal to logic '1' causing all the excitatory neurons to reset their voltages. Hence, all the neurons, other than the one which has generated the original spike, are prohibited from producing any further pulses.

As explained in section 2.7.1, due to the stochastic nature of the oxygen vacancies movement in the oxide material, the RRAM state suffers from reliability soft-errors caused

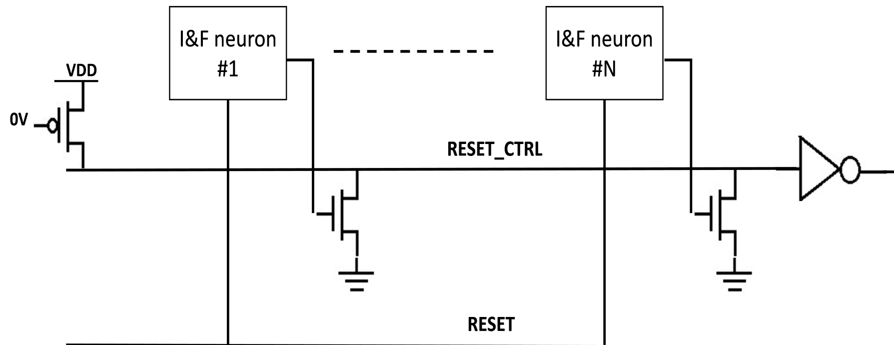

Figure 6.4: Simple implementation of WTA methodology. The first neuron generating pulses discharges the control line 'RESET_CTRL' which raises the 'RESET' control signal prohibiting any other neurons from generating pulses.

by: a) the diffusion of oxygen vacancies out of the filament containment, and b) the manufacturing defects introduced during the fabrication process [38, 35, 39, 125]. Up to the moment when this chapter is written, we are not aware of any previous work which assessed how much the impact of RRAM soft-errors on the neuromorphic system performance can be. In this work, a systematic modeling framework is provided to compute the effect of RRAM reliability soft-errors on the system performance. Applying this methodology to the system in [142], the accuracy of our case-study system in classifying the input patterns is reduced by more than 48%. Due to the nature of the case-study neuromorphic system, where the resistive state of the RRAM device changes by maximum 1% with each applied input pattern [142], the RRAM reliability hard-errors, caused by endurance limit of the RRAM device, can rarely occur during the operation of the system [36, 157]. To restore the system accuracy in classifying the input patterns, the various possible modifications to the neurons signals, which affect the change in the RRAM state, are analyzed. Furthermore, two novel algorithms are proposed to automatically detect and restore the loss in the output accuracy whenever the RRAM reliability soft-errors occur. Using a combination of SPICE and python-based neural network system simulator (i.e., BRIAN [43]) on the case-study system, the newly proposed techniques can recover the lost accuracy with minimal increase in the delay and energy consumption of the system.

6.2 Modeling the RRAM Reliability Soft-Errors on the System Level

Generally speaking, if the RRAM-based neuromorphic system is designed without taking into account the RRAM reliability soft-errors, discussed in section 2.7.1, the actual accuracy of the system (i.e., A_{act}) will be less than the expected accuracy by design (i.e., A_{exp}). This is because the actual system is manufactured with RRAM devices, some of which will encounter reliability soft-errors. Fig. 6.5 illustrates our suggested modeling framework aiming to quantify the degradation in the system performance caused by RRAM reliability soft-errors during the training phase.

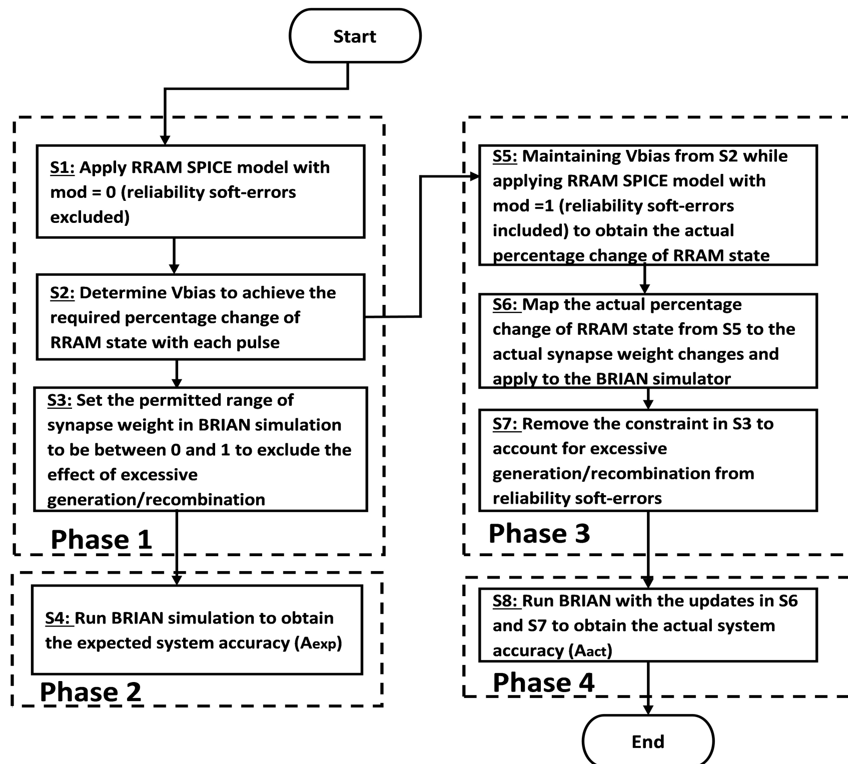

Figure 6.5: Modeling framework for computing the impact of RRAM reliability soft-errors on the system accuracy. “Phase I” and “Phase II” describe the SPICE and system level simulations run without taking into account the RRAM soft-errors. “Phase III” and “Phase IV” describe the SPICE and system level simulations run while the RRAM soft-errors are being considered.

The framework in fig. 6.5 mainly focuses on the soft-errors generated during the training phase for the following reasons:

1. Only the training phase of the system can be controlled as the patterns applied to the system are already known. During the testing phase, which mimic the case when the system is deployed in a real application, the applied patterns are completely unknown to the system.

2. The training cycle is responsible for a significant portion of the system energy and delay, since all the RRAM devices are being programmed for 350 ms for each of the 60,000 patterns of the MNIST dataset. In case if not enough pulses are generated, the input pattern is re-applied to the system for another 350 ms in an attempt to re-program the resistive state of the RRAM array as explained in section 6.1. This actually results in significant increase in the delay and hence, the energy consumption of the system.

3. If soft-errors occur during the testing phase (i.e., after the system is deployed in its application), the system needs to be re-trained again to restore the resistive states of the RRAM device. Hence, focusing on resolving soft-errors during the training cycle of the system is necessary.

More details of the framework implementation including the change in models and tools used are explained in subsequent sections. The framework described in fig. 6.5 consists mainly of two stages:

- **Computing** A_{exp}: In this stage, the system accuracy is computed excluding the RRAM reliability soft-errors, i.e., assuming that no RRAM device will suffer from reliability soft-errors. This is done through two phases:
  - **Phase 1 (i.e., S1-S3 in fig. 6.5): SPICE simulations excluding soft-errors**: The RRAM reliability soft-errors are excluded from the SPICE model in this phase to obtain the required voltage to change the RRAM resistive state by the percentages described in the system specs in [142].
  - **Phase 2 (i.e., S4 in fig. 6.5): BRIAN simulations excluding soft-errors**: In this phase, the BRIAN simulation is run to obtain A_{exp} without adding any parameters describing the RRAM reliability soft-errors.
- **Computing** A_{act}: In this stage, the actual system accuracy A_{act} is evaluated when the RRAM reliability soft-errors occur. Two phases are invoked in this stage:

- **Phase 3 (i.e., S5-S7 in fig. 6.5): SPICE simulations including soft-errors**: During this phase, the voltage biases, obtained in phase 1, are re-run using a SPICE model containing equations which describe the change in the I-V characteristics of the device due to the RRAM reliability soft-errors. Hence, the new percentages of change in the resistive state of the RRAM, including the impact of the RRAM reliability soft-errors, are obtained.
- **Phase 4 (i.e, S8 in fig. 6.5): BRIAN simulations including soft-errors**: Using the new percentages of change in the RRAM resistive state in phase 3, the BRIAN simulation for the system is re-run to compute A_{act}.

6.2.1 Computing A_{exp}

The changes in the synapses weights are mapped to the RRAM resistive state such that: the RRAM HRS corresponds to state '0' of the synapse weight, while the RRAM LRS maps to state '1'. Hence, in step S1, the biasing conditions, causing the maximum percentages of synapse weight changes, are obtained using the RRAM HfO_x SPICE model in [33] excluding the reliability soft-errors. This is done by modifying the SPICE model in [33] to be able to include (i.e., with mod = 1) or exclude (i.e., with mod = 0) the reliability effects from its equations. The RRAM reliability soft-errors are described in the SPICE model in [33] through :

- **Activation energy for the generation and recombination processes**: As described in [52, 33], the SET/RESET process, which changes the resistive state of the RRAM device from HRS/LRS to LRS/HRS, is initiated whenever the applied voltage exceeds the activation energy for generating (i.e., E_{ag}) /recombining (i.e., E_{ar}) the oxygen vacancies around the conductive filaments, respectively. To model the reliability soft-errors attributed to the diffusion of the oxygen vacancies out of the conductive filament containment (i.e., mod = 1), the E_{ar} in [33] is lower than the E_{ag} to favor the recombination process over the generation process. In mod = 0 (i.e., excluding the effect of the RRAM reliability soft-errors), E_{ag} and E_{ar} are equalized. Equations 6.1 and 6.2 describe the rate of change in the resistive state of the RRAM during mod = 0 and mod = 1, respectively.

$$\begin{aligned} \frac{dr}{dt} = K_1 * [exp(-\frac{qE_{ag}}{kT}) * exp(\frac{K_2 * V}{kT}) \\ -exp(-\frac{qE_{ar}}{kT}) * exp(-\frac{K_2 * V}{kT}))] \end{aligned} \tag{6.1}$$

$$
\begin{aligned}
\frac{dr}{dt} = K_1 * [exp(-\frac{qE_{ag}}{kT}) * exp(\frac{K_2 * V}{kT}) \\
-exp(-\frac{qE_{ag}}{kT}) * exp(-\frac{K_2 * V}{kT}))]
\end{aligned}
\tag{6.2}
$$

The constants K_1 and K_2 are described in more details in [33]. k and q are the Boltzmann constant and electron charge in Coulomb, respectively. V and T describe the voltage across the RRAM device and the temperature of its conductive filaments, respectively.

- **Device temperature and its I-V characteristics**: When the RRAM state changes, the move of oxygen vacancies in and out of the filament containment, under the effect of the applied field, increases the temperature inside the conductive filament [33, 158]. This also speeds up the out diffusion of the oxygen vacancies [36, 38, 39]. To describe the temperature change with the frequency of change of the RRAM state, the SPICE model in [33] introduced a positive feedback loop between the temperature of the filament and the voltage and current of the RRAM device. With each time step of the simulation, the filament temperature changes exponentially according to: a) the voltage applied on the RRAM device and the resulting output current, and b) the period of time during which the voltage on the RRAM device is applied. This effect is suppressed (i.e., with mod = 0) using the temperature variation equations in [32, 102] which linearly relates the change in the filament temperature with the applied stimulus. Equations 6.3 and 6.4 describe the rate of temperature change with the applied stimulus for mod = 1 and mod = 0, respectively.

$$
T = K_3 + \frac{|V * I|}{C_{th}} + C_1 * exp(-T_{(t-1)}) \tag{6.3}
$$

$$
T = T0 + \frac{|V * I|}{C_{th}} \tag{6.4}
$$

K_3 and C_{th} are constants defined in the SPICE model described in [33]. C_1 is a constant resulting from resolving the differential equations for the temperature change with time as detailed in [33]. $T_{(t-1)}$ is basically, the conductive filaments temperature of the RRAM device at a previous time instance. V and I are the voltage and current of the RRAM device. Due to the exponential dependency on precedent temperature values, Equation 6.3 introduces also an exponential function to the voltage and current of the RRAM device.

It is worth mentioning that, with mod = 0, the model equations 6.2 and 6.4 are identical to those in model [32]. Hence, with mod = 0, we use directly the model in [32]. SPICE

simulations, with the modifications to include and exclude the reliability soft-errors, are applied to obtain the required voltages on the RRAM devices to cause the maximum percentage of change in the synapses weights described in [142] (i.e, step S2 in fig. 6.5). The change in the RRAM resistive state is maximized when:

- The overlap between the pre- and post-neurons pulses is equal to the duration of the pulses (i.e., 20 ms as described in [142]).
- The RRAM resistive state is near its LRS which results in a large current to pass through the RRAM device maximizing the change in its state.

As described in [35, 38], using unbalanced programming pulses can cause extra generation/recombination for the oxygen vacancies resulting in having smaller/larger RRAM resistive state than its minimum LRS/maximum HRS. To exclude the consideration of this effect, the synapses weights in the BRIAN simulation are not allowed to change beyond the [0,1] boundaries (i.e., step S3 in fig. 6.5). By excluding the RRAM reliability soft-errors (steps S1-S3) and running the BRIAN simulations for the system in [142] (i.e., step S4), the results demonstrates that the accuracy of the system in recognizing the MNIST dataset is $A_{exp} = 91.6\%$.

6.2.2 Computing A_{act}

A_{act} is calculated by repeating the steps S1-S4 while including the RRAM reliability soft-errors effect. In step S5 in fig. 6.5, the same voltages, obtained in step S2, are applied to the RRAM device but with mod = 1 of our modified SPICE model (i.e., the RRAM reliability soft-errors are taken into account). Applying this to our case-study system to estimate the impact of the RRAM reliability soft-errors, table 6.1 summarizes the biasing voltages and the related maximum percentage of change in the synapses weights. The RRAM reliability

Table 6.1: SPICE simulation results for the maximum percentages of change in the RRAM resistive state

Biasing voltages	**mod = 0 (excluding soft-errors)**	**mod = 1 (including soft-errors)**
-0.85V/+0.95V	-0.01%/+1%	-4.3%/+3.5%

soft-errors cause a larger change in the RRAM device state when mod = 1 compared to the case when mod = 0. The percentage of decrease in the RRAM resistive state (i.e., -4.3%)

is higher than that of the increase in its resistive state (i.e., +3.5%) due to the low-power operating conditions which results in less number of oxygen vacancies in the conductive filaments [35, 39]. Accordingly, the diffusion of the oxygen vacancies out of the conductive filaments has a bigger impact on the decrease in the RRAM resistive state. To compute A_{act}, the maximum percentages of change for the synapses weights in the BRIAN simulator are modified (i.e., -4.3%/+3.5% instead of -0.01%/+1% in our case-study system) (i.e., step S6). Also, to model the possible extra generation/recombination of the oxygen vacancies, the synapse weights of the system are allowed to change beyond the [0,1] boundaries (i.e., step S7). This way, if extra generation/recombination events occur, the incoming pulses from the applied pattern have to first compensate for the extra change in the synapses weight before moving it back to be within the [0,1] range (i.e., normal range for the change between maximum HRS and minimum LRS resistance values, respectively). By applying the steps S5-S7 to the BRIAN simulations, the A_{act} is calculated in step S8. Table 6.2 summarizes the BRIAN simulation settings and results of our proposed framework for the case-study system. The exact python code for the neural network described through the BRIAN package with the proposed modifications is detailed in Appendix A.

Table 6.2: BRIAN simulation settings and results

Parameter	**Excluding reliability soft-errors**	**Including reliability soft-errors**
Percentage of decrease in the synapses weights	0.01%	4.3%
Percentage of increase in the synapses weights	1%	3.5%
Weights limits	0 (maximum HRS) - 1 (minimum LRS)	No limits
System Accuracy	$A_{exp} = 91.6\%$	$A_{act} = 43\%$

As indicated in table 6.2, the accuracy of the system in recognizing the handwritten digits of MNIST patterns degrades from $A_{exp} = 91.6\%$ to $A_{act} = 43\%$ due to the RRAM reliability soft-errors. The large degradation in the output accuracy of the system in [142] is resulting from considering the case when all the RRAM devices in the network are simultaneously suffering from reliability soft-errors. This scenario is valid because:

- With the long duration of applying the input patterns (i.e., 350 ms [142]), the case when majority of the RRAM devices are suffering from soft-errors could easily happen.

- Through this study, we would like to assume the worst-case scenario, which is not overly pessimistic, and see if the system will be able to tolerate such effect.
- The randomness in applying the training and testing patterns enhances the effect of the reliability soft-errors resulting from unbalanced programming pulses on all the RRAM devices.

Fig. 6.6 shows the flow of simulation and the tools used to compute A_{act} based on the framework described in fig. 6.5. The simulation runs start by taking the maximum per-

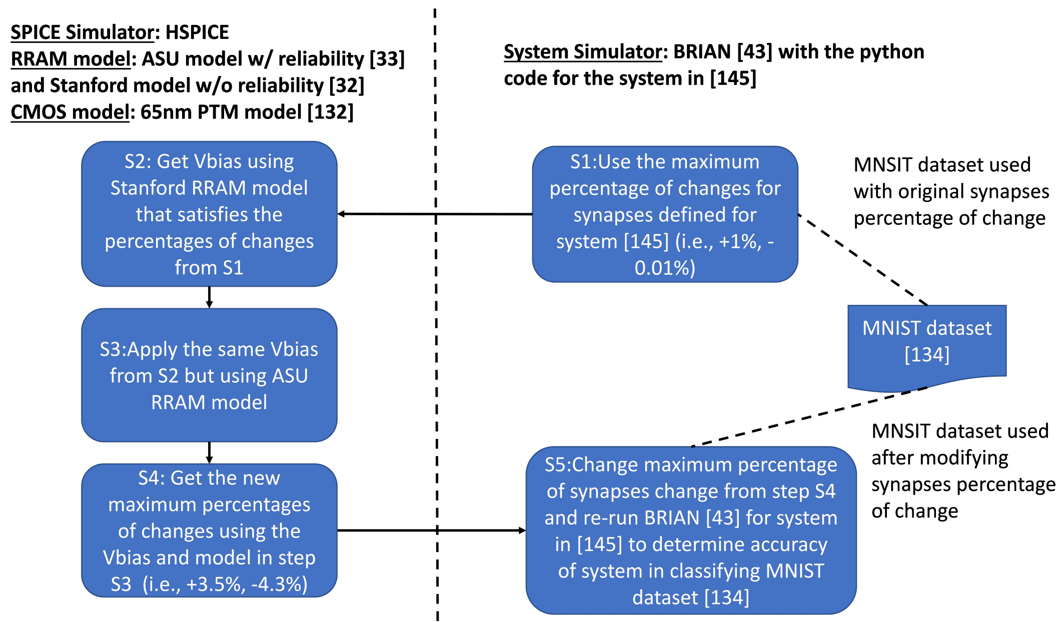

Figure 6.6: Block diagram for the simulation runs and the tools used to compute the degradation in RRAM-based neuromorphic system. BRIAN is chosen as the system level simulator since the code for the system in [142] is already written using this python package. In our case-study system, the classification of the handwritten digits of MNIST dataset [44] is used to estimate the degradation in system performance due to RRAM soft-errors.

centage required to be applied on the synapses with each input pattern as defined in the BRIAN code in [142] (i.e., step S1 in fig. 6.6). As explained in section 6.1, this maximum percentage depends on the neuromorphic system structure and the learning algorithm used to train the synapses. Then, using HSPICE simulator [124] and the Stanford RRAM model [32], which does not include the RRAM reliability soft-errors (i.e, mod = 0 in fig. 6.5),

we obtain V_{bias} to be applied on the RRAM device to satisfy the required maximum percentages of RRAM state change obtained from step S1 (i.e., step S2 in fig. 6.6). Using the model from ASU describing the RRAM reliability soft-errors [33] (i.e, mod = 1 in fig. 6.5), the V_{bias} from step S2 is reapplied to obtain the new maximum percentage of change in the synapses state (i.e., step S3 and S4 in fig. 6.6). Then, using the synapses equations defined in BRIAN python code, the value of maximum change is modified with the values obtained from step S4. After this, in step S5, the training and testing sets of MNIST dataset are rerun on the BRIAN python code in [142] to evaluate the accuracy of the system in classifying the hand-written digits after including the RRAM soft-errors effects.

6.3 Analysis of the Neuron Pulses

In this section, the various possible adjustments in the parameters of neuron pulses are analyzed to restore the degradation in the system performance caused by the RRAM reliability soft-errors. Similar to the biological neurons pulses shown in fig. 6.7 [159], the parameters of neuron pulses are:

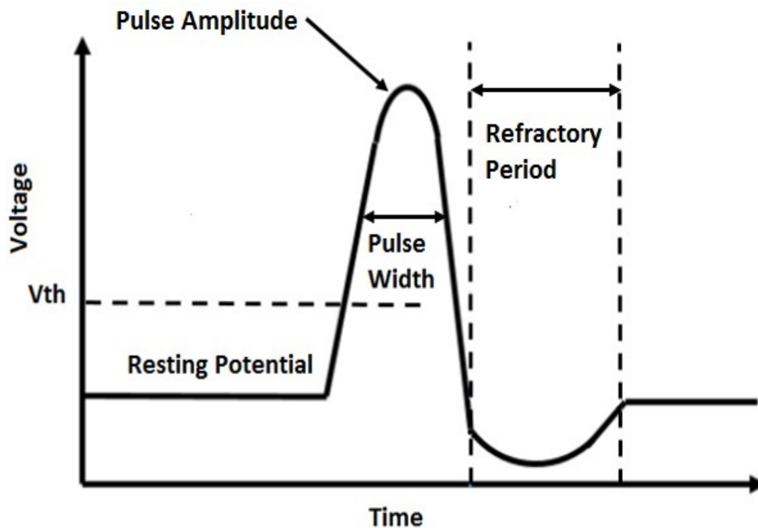

Figure 6.7: Properties of the action potential (neuron pulse) [159]. There are basically three main properties: pulse amplitude, pulse width, and pulse frequency which is defined through neuron threshold voltage, resting potential, and refractory time.

- **Pulse Width**: This parameter describes the duration of the neuron pulse which by consequence defines the period during which the RRAM state can change. Adjusting the pulse width has conflicting impact on the RRAM reliability soft-errors and the energy consumption of the system. In order to reduce the impact of the RRAM reliability soft-errors, smaller pulse widths are required as this leads to smaller changes in the RRAM state. However, decreasing the pulse widths increases the duration of the training cycle which, by consequence, increases the energy consumption of the system as explained later in section 6.4.3.
- **Pulse Amplitude**: This defines the maximum voltage that can be applied on the RRAM device. Similar to the effect of the pulse width, smaller pulse amplitudes reduces the impact of the RRAM reliability soft-errors. Yet, this increases the training cycle duration and, hence, the energy consumption of the neuromorphic system.
- **Pulse Frequency**: As illustrated in fig. 6.7, the pulse frequency can be controlled by three parameters:
  - **Threshold voltage (i.e., V_{th})**: This parameter defines the minimum voltage that needs to be applied on the neuron to start generating output pulses.
  - **Resting potential (i.e., V_{res})**: This parameter describes the voltage of the neuron when it is at rest (i.e., if there is no stimulus applied to it).
  - **Refractory period (i.e., T_{refr})**: This parameter determines the time, after a pulse is generated by a neuron, during which it can not produce any pulses even if the voltage applied to it is high enough to create a pulse (i.e., $> V_{th}$).

  Adjusting the pulse frequency impacts how many times the RRAM state gets modified. Hence, the lower the frequency, the fewer are the time frames available for the RRAM devices to change their state.

Fig. 6.8 summarizes the results of analyzing the various pulse parameters with the target of restoring the actual system accuracy A_{act} (i.e., 43% in our case-study system) to A_{exp} (i.e., 91.6% in our case-study system).

6.3.1 Effect of Changing the Pulse Frequency

Reducing the frequency of pulses (i.e., by adjusting V_{th}, V_{res}, T_{refr}) decreases the time overlap between the pulses from the pre- and post-neurons and, by consequence, reduces the rate of change in the RRAM state. To lower the pulse frequency, the following experiments are conducted:

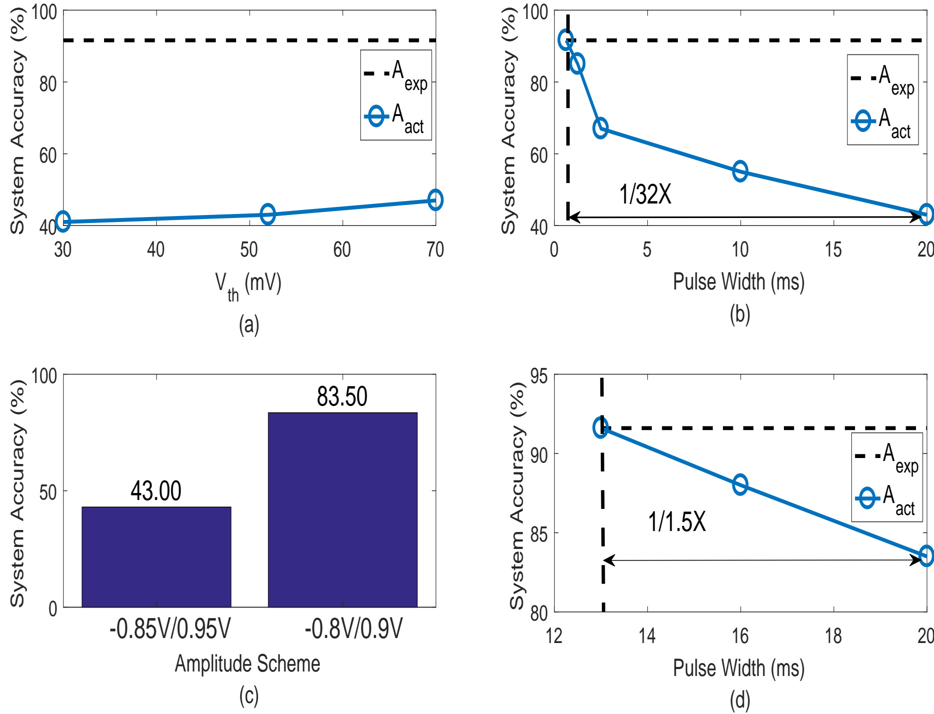

Figure 6.8: Impact of changing the various parameters of the neuron pulses on the actual system accuracy A_{act}. a) Changing the pulse frequency, b) Changing the pulse width, c) Effect of changing the pulse amplitude, and d) combining the changes in the pulse amplitude and frequency to restore the system accuracy with minimum impact on its energy consumption.

- Increase V_{th} of the pre- and post-neurons from its original value of 52 mV by 5 mV, 10 mV, and 20 mV. This limits the number of pulses generated by the neurons to the signals with amplitude higher than V_{th}.

- Lowering V_{res} of the pre- and post-neurons from 65 mV by 5 mV, 10 mV, and 20 mV. Similar to the case with increasing V_{th}, this prolongs the time required for the voltage of the neurons to reach V_{th}. Hence, the number of pulses, generated throughout the duration of applying the input patterns, decreases.

- Increase the V_{th} and lowering V_{res} together by 5 mV and 20 mV to maximize the effect of the frequency reduction.

- Increase T_{refr} from 5 ms by 2 ms, 5 ms, and 10 ms. This reduces the period of active time during which the neurons can generate output pulses. By doing so, the time available for the applied pattern to generate output pulses is reduced. Accordingly, the pulse frequency decreases.

After running the different experiments, it is found that the enhancement in the system accuracy is negligible. For example, changing V_{th} can only improve the system accuracy by $< 4\%$ as illustrated in fig. 6.8a. This is because the degradation in the RRAM state, caused by the reliability soft-errors, is significantly large (i.e., -4.3%/-0.01% = 430x as listed in table 6.1 in section 6.2). Hence, changing the pulse frequencies is simply unable to compensate this huge difference. Running the other frequency reduction experiments (i.e., T_{refr} and V_{res} adjustments) results in the same minor improvement in the system accuracy shown in fig. 6.8a (i.e, $< 4\%$ improvement). For the sake of clarity, only the enhancement due to V_{th} modification is illustrated in fig. 6.8a.

6.3.2 Effect of Changing the Pulse Width

Reducing the pulse width decreases the time overlap between the pulses from the pre- and post-neurons, which, by consequence, reduces the period during which the RRAM state can change. Referring to the results in table 6.2 for our case-study system, since the discrepancy in the maximum percentage for the decrease in RRAM resistive state is higher than that for its increase (i.e., -4.3%/-0.01% = 430x versus +3.5%/+1% = 3.5x), reducing the pulse width of the post-neurons reduces the impact of RRAM reliability soft-errors on the system performance. This is because, according to the STDP rule [155] mentioned in section 6.1, if the duration of overlap between the generated pulses from the post-neurons and those from the pre-neurons is decreased, the timing window during which the RRAM state can change towards its HRS is also reduced. Fig. 6.8b demonstrates that, to restore the system accuracy to A_{exp}, the pulse width is required to decrease by almost 1/32x from the original pulse width value (i.e, 20 ms in [142]). One way for reducing the pulse width is to divide the 784x400 excitatory neural network into 32 units (31 unit consists of 25x400 neurons and the last unit contains 9x400 neurons) as illustrated in fig. 6.9. This is because the pulse width depends on how fast the BL capacitance, which is directly proportional to the number of wordline connected to it, is discharged [160]. Although the capacitance of each of the units BL is decreased by 32x, the power consumption of the system is increased due to the extra circuitry needed for collecting the information from each unit (i.e., buffers, decoders, multiplexers). Using the CACTI C++ code [42], we have evaluated the effect of dividing the 784x400 memory array into 32 units by adjusting the value of the

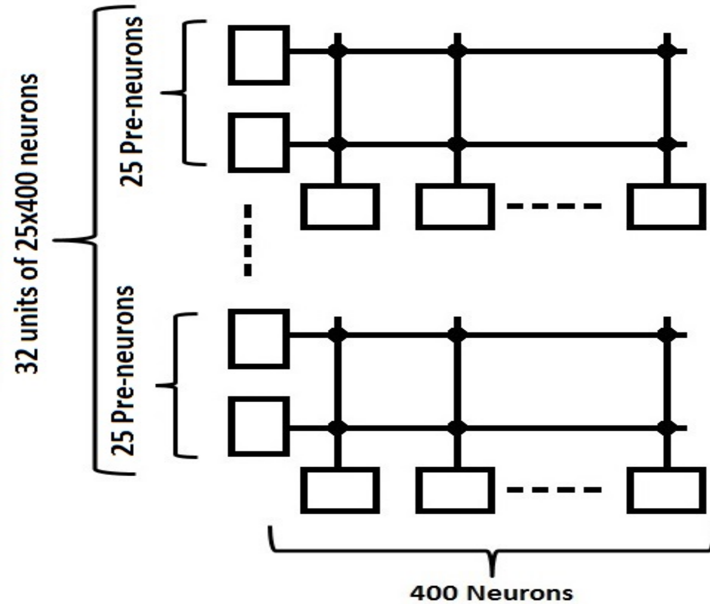

Figure 6.9: Decomposing the 784x400 network into 32 units (31 units consist of 25x400 neurons and the last unit contains 9x400 neurons). Using CACTI C++ files, the increase in power consumption due to the split of the neural network is about 11x.

"NdBL" parameter to 32. The CACTI results show that the power consumption of the system increases by 11x. Hence, reducing the pulse width is one of the approaches that can theoretically help in restoring the system accuracy, however it can not be used alone in practice due to the huge energy consumption overhead.

6.3.3 Effect of Changing the Pulse Amplitude

As explained in [161, 162, 163, 125], modifying the pulse amplitude has the highest impact on changing the way the RRAM resistive state is updated. Using the model in [33] with our case-study system, it is found that, reducing the pulse amplitude by 50 mV (i.e., +0.9 V/-0.8 V instead of +0.95 V/-0.85 V) lowers the maximum percentage of change in the RRAM resistive state from -4.3%/+3.5% to -0.2%/+0.9%. The reduction in the pulse amplitude is chosen based on an iterative process which decreases the pulse amplitude by 25 mV with each iteration. The 25mV decrease step agrees with the minimum 2% accuracy that can be achieved with the lower-power voltage reference circuits similar to those discussed in [164, 165]. Using the newly proposed biasing voltages, the system accuracy has increased

to 83.5% as illustrated in fig. 6.8c. Any further decrease in the pulse amplitude yields to a lower enhancement in the restored system accuracy due to the decrease in the number of pulses whose amplitude exceeds V_{th} of the neurons.

6.3.4 Combining the Effect of Changing the Pulse Amplitude and Width

To restore the system accuracy to A_{exp} (i.e., 91.6% in our case) with minimum impact on its performance, the effect of decreasing the post-neuron pulse width, in section 6.3.2, is combined with the suggested change in the biasing voltages in section 6.3.3. Fig. 6.8d demonstrates that decreasing the pulse width by only 1/1.5x, while simultaneously reducing the pulse amplitude by 50 mV, can restore the system accuracy to its expected value. Repeating the CACTI experiment described in section 6.3.2, partitioning the memory array into two units increases the power consumption of the system by only 20%.

6.4 Proposed Framework to Detect and Fix the RRAM Reliability Soft-Errors

In this section, a novel methodology is proposed to automatically detect and fix the degradation in system accuracy resulting from the RRAM reliability soft-errors. As detailed in section 6.2, the focus in this chapter is to detect and fix the soft-errors that can occur during the training cycle as it is the controllable phase of the neuromorphic system and the main contributor to its energy consumption. The main idea is to keep the system operating based on its original biasing conditions (i.e., for our case-study system, the pulse width is 20 ms and the maximum pulse amplitude is +0.95 V/-0.85 V). Once the degradation in the output accuracy is detected, a restore methodology is triggered to revert the negative effect of the RRAM reliability soft-errors.

Fig. 6.10 shows the flowchart for the suggested framework. The concept behind the framework in fig. 6.10 is based on the observation that the RRAM reliability soft-errors cause a significant drop in the number of generated pulses. This is because, in the low-power systems, when the reliability soft-errors occur, the RRAM resistive state moves towards its HRS as discussed in section 6.2. Accordingly, the connections between the pre- and post-neurons are weakened and hence, less pulses will be generated by the post-neurons. The algorithm in fig. 6.10 consists of two main steps:

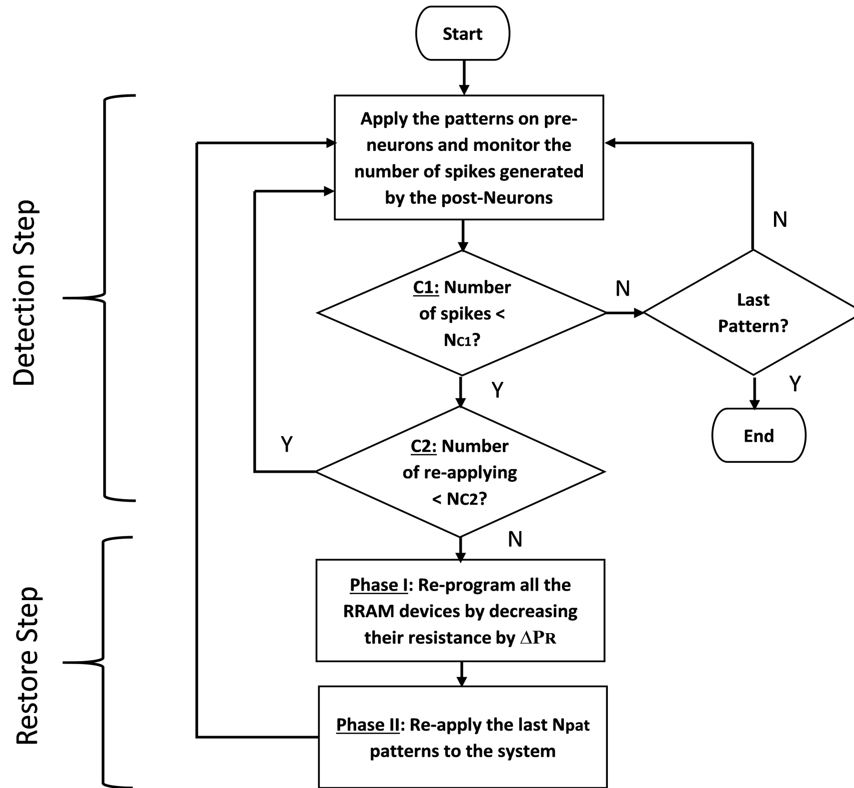

Figure 6.10: Flowchart of the suggested methodology for detecting and fixing the system performance drop caused by RRAM reliability soft-errors. The proposed framework consists of two main phases: "Detection Phase" which monitors the number of generated pulses with each input pattern to detect if the RRAM array is suffering from reliability soft-errors , and "Restore Phase" which is responsible of fixing the degradation in system performance by first decreasing the resistance of all RRAM devices by ΔP_R and then re-apply the last N_{pat} patterns to reprogram the RRAM states correctly.

- **Detection step**: The purpose of this stage is to detect when the RRAM reliability soft-errors occur by tracking the number of pulses generated by the post-neurons in response to the input patterns.
- **Restore step**: In this step, the degraded output accuracy is restored back to its A_{exp} through re-adjusting the RRAM resistive state and re-applying some of the input patterns as indicated in phase I and phase II in fig. 6.10.

6.4.1 Detection Step

As discussed in section 6.2, when the RRAM reliability soft-errors occur in low-power systems, the connectivity between the pre- and post-neurons is weakened, causing the decrease in the number of pulse generated in response to the input patterns. For our case-study example, the authors in [142] explained that if the number of generated pulses drops below a threshold value of 5 (i.e., N_{C1} in the C1 condition in fig. 6.10), the input pattern has to be re-applied while lowering the rate of input pulses. Consequently, the period, during which the synapse weights (i.e., RRAM resistive state) can change, is increased. To assign a limit on the number of times an input pattern can be re-applied, the worst case scenario is considered for the required change in the RRAM resistive state. This occurs when the RRAM resistance needs to change from its HRS all the way to its LRS to start generating pulses at the post-neurons. In our case-study system [142], using the average overlap period between the pre- and post-neurons (i.e., 10 ms) and the original biasing conditions of the system (i.e., pulse width is 20 ms and maximum pulse amplitude is +0.95 V/-0.85 V), it is found that the upper limit for re-applying an input pattern is 50 times (i.e., N_{C2} in the C2 condition in fig. 6.10). The average overlap period between the pre- and post-neurons are computed based on the BRIAN simulations for the system in [142], where the maximum and minimum overlap periods between the pre- and post-neurons are 20.0 ms and 10 μs, respectively.

After re-applying an input pattern N_{C2} times (i.e., 50 times for our case-study system), if the number of generated pulses at the post-neurons is still less than N_{C1} (i.e., 5 for our case-study system), the restore step is initiated. This indicates that the RRAM devices are suffering from reliability soft-errors, which prevent them from changing their HRS in response to the given input pattern.

6.4.2 Restore Step

As illustrated in fig. 6.10, the restore step consists of mainly two phases:

- **Phase I**: During this step, all the RRAM resistances are decreased by a minimum of ΔP_R, which is calculated based on the worst-case degradation in the RRAM resistive state computed using the modeling framework discussed in section 6.2. For our case-study system, as listed in table 6.1, the worst-case degradation for the RRAM state occurs when the percentage of change towards the HRS is increased from 0.01% to 4.3%. By applying a pulse of 0.95 V for 20 ms on the post-neurons while the pre-neurons are grounded, the resistance of the RRAM devices at HRS are reduced by 4.3%. However, for other RRAM devices at lower resistance than that of HRS, their states are changed by a higher percentage than 4.3% depending on their original state. Decreasing the resistance of the RRAM devices by ΔP_R restructures the conductive filaments in their oxide material by restoring the diffused oxygen vacancies.

- **Phase II**: In this step, the last N_{pat} number of patterns, applied on the pre-neurons before detecting the RRAM reliability soft-errors, are re-applied.

Assuming that the RRAM reliability soft-errors are detected during the 2nd tier of the training sequence (i.e., soft-errors are detected while applying patterns in the range from 20,000 to 40,000 of the 60,000 training sequence), fig. 6.11a shows the change in the restored output accuracy depending on N_{pat}. fig. 6.11a demonstrates that the optimal number of re-applied patterns (i.e., $N_{pat,opt}$) to revert the system accuracy back to its original A_{exp} = 91.6% is 1000. Increasing the number of re-applied patterns beyond $N_{pat,opt}$ (i.e., N_{pat} = 2000 in fig. 6.11a) leads to more RRAM devices to be in their LRS. Hence, more connections will be able to transfer pulses between the pre- and post-neurons. Accordingly, for any given pattern, more post-neurons will generate pulses which lowers the ability to differentiate between the various applied input patterns reducing the overall system accuracy. In addition to this, the delay and energy consumption of the system significantly increase due to extending the training cycle as discussed in section 6.4.3. Oppositely, decreasing the number of re-applied patterns below $N_{pat,opt}$ (i.e., N_{pat} = 500 in fig. 6.11a) decreases the number of RRAM devices which can transmit pulses between the pre- and post-neurons reducing also the restored system accuracy.

Fig. 6.11b demonstrates how the $N_{pat,opt}$ changes depending on when the RRAM reliability soft-errors are detected during the training cycle. If the reliability soft-errors are observed during the first tier of the training cycle, less $N_{pat,opt}$ patterns are required to be re-applied compared to the case if they are detected during the third tier. Owing to the reduced number of the remaining training patterns, higher $N_{pat,opt}$ is required to be re-applied if the RRAM reliability soft-errors are observed at the third-tier of the training cycle. Re-applying the correct $N_{pat,opt}$, based on when the reliability soft-errors are noticed

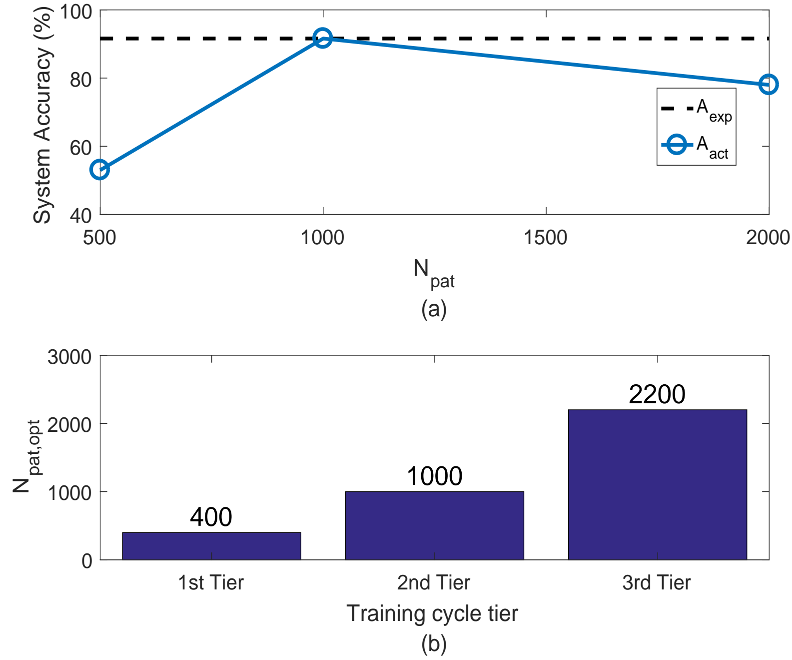

Figure 6.11: N_{pat} study. a) Impact of changing N_{pat} on restoring the system accuracy, b) the required $N_{pat,opt}$ patterns when the RRAM reliability soft-errors occur during the various tiers of the training sequence. 1st Tier, 2nd Tier, and 3rd Tier in fig. 6.11b refers to the cases when soft-errors occur in the first, second, and third 20,000 patterns of the MNIST training dataset, respectively.

during the training cycle, is essential to reduce the overhead on the energy consumption and delay of the system as explained in section 6.4.3.

6.4.3 Impact of the Proposed Framework on the System Performance

Fig. 6.12 illustrates the impact of using the proposed framework on the delay and energy of our case-study system.

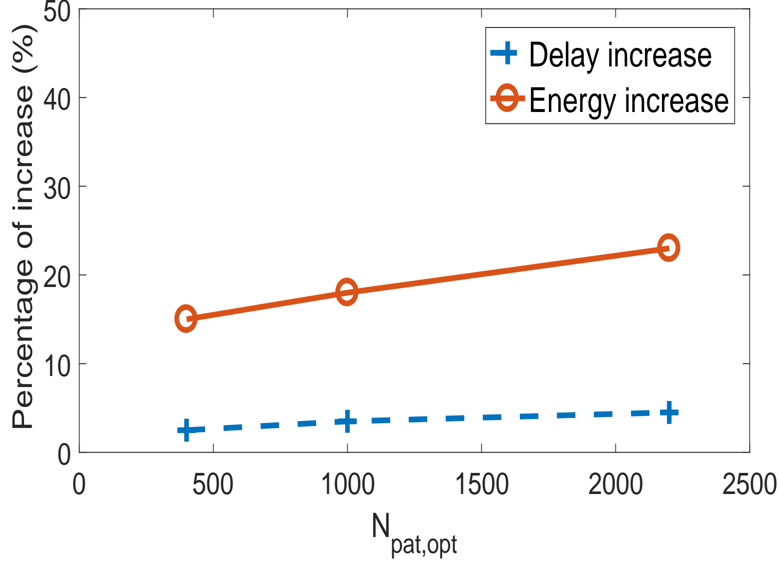

Figure 6.12: Effect of incorporating the suggested detection and fix algorithm on the delay and energy consumption of the system [142]. When the number of $N_{pat,opt}$ grows, the training cycle duration increases and by consequence, the energy consumption of the system. The "percentage of increase" in the figure represents the increase in the energy and delay of the training cycle in comparison to the case when the suggested methodology is not integrated to the system.

Fig. 6.12 demonstrates that, in the worst-case when the RRAM reliability soft-errors are detected during the third-tier of the training cycle, the delay and energy consumption can increase by 4% and 20%, respectively. This is because, in such scenario, the training cycle is extend by $N_{pat,opt} = 2200$ as shown in fig. 6.11b, which increases the delay and energy consumption of the system. The increase in the system energy is much higher than that of its delay due to the hyperbolic relation between the current passing through the RRAM device and the period allowed for its state to change as described in the SPICE models in [32, 33].

The percentages of increase in energy (i.e., ΔE) and delay (i.e., ΔT) in fig. 6.12 are calculated based on equations 6.5 and 6.6, respectively:

$$\Delta E = \frac{E_{avg,orig} + E_{avg,phaseI} + E_{avg,phaseII}}{E_{avg,orig}} * 100\% \tag{6.5}$$

$$\Delta T = \frac{T_{orig} + T_{PhaseI} + T_{PhaseII}}{T_{orig}} * 100\% \\ = \frac{350ms * (60,000 + num_{N_{pat,opt}}) + 20ms}{350ms * num_{patterns}} * 100\% \tag{6.6}$$

where:

- $E_{avg,orig}$: describes the average energy consumed by the RRAM array during the training cycle without applying the proposed framework.
- $E_{avg,phaseI}$, $E_{avg,phaseII}$: define the average energy consumed during phases I and II of the restore step explained in section 6.4.2.
- T_{orig}: is the period required to apply the 60,000 training patterns of MNIST dataset. As described in the system specs in [142], each pattern is applied for 350 ms.
- T_{PhaseI} , $T_{PhaseII}$: define the time required to complete phase I and II of the restore step. As discussed in section 6.4.2, a 0.95 V pulse is applied for 20 ms during the phase I of the restore step, while $N_{pat,opt}$ are re-applied for 350 ms to the system during phase II.

The average energy in eq. 6.5 (i.e., $E_{avg,orig}$, $E_{avg,phaseI}$, and $E_{avg,phaseII}$) is calculated using the following sequence:

1. Consider the case which causes maximum and minimum change in the RRAM state. As described in section 6.4.1, the maximum and minimum overlap between the pulses of pre- and post-neurons, which causes maximum and minimum change in the RRAM resistive state, are 20.0 ms and 10 μs, respectively.
2. Using mod = 1 of SPICE model, discussed in section 6.2, the average energy consumed to change the RRAM state is calculated for pulses with amplitudes +0.95 V/-0.85 V and duration of 20 ms and 10 μs pulses. The simulation runs are repeated for the cases when the RRAM state is at LRS and when it is at HRS.
3. Using BRIAN system level simulations, the number of pulses for all the patterns are monitored to detect the maximum and minimum number of pulses occurred during the training phase. The average number of pulses is multiplied by the average energy calculated in step 2.

6.5 Improving the Proposed Framework to Detect and Fix the RRAM Reliability Soft-Errors

Despite the simplicity of the proposed framework in section 6.4, it increases the energy consumption of the system by 20%. In order to reduce this impact, fig. 6.13 illustrates the proposed modifications to the original algorithm in fig. 6.10.

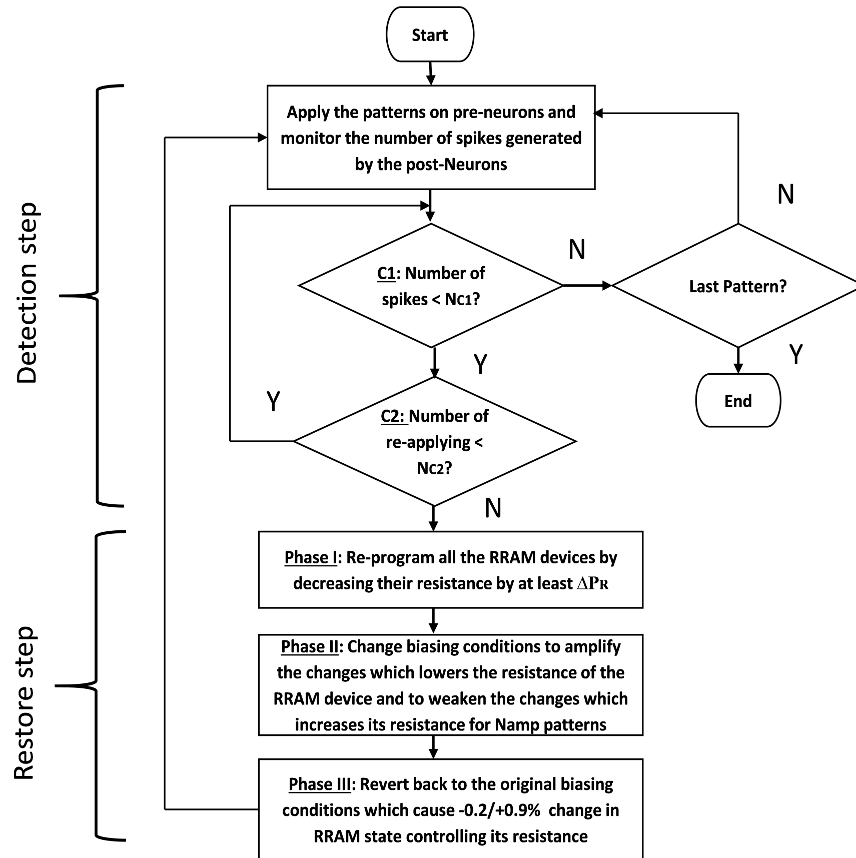

Figure 6.13: Flowchart of the modified methodology for detecting and fixing the RRAM reliability soft-errors. The algorithm is very similar to the one described in fig. 6.10. The main difference is eliminating the need to re-apply patterns by: 1) increasing the amplitude of input pattern causing a decrease in the RRAM resistance, and 2) decreasing the amplitude of input patterns otherwise. This change is only applied for next N_{amp} patterns of the training cycle.

The main difference between the methodology suggested in section 6.4 and the one described in fig. 6.13 is in phase II of the restore step. Instead of re-applying $N_{pat,opt}$ patterns, which increases the delay and energy consumption of the system, the proposed framework in fig. 6.13 adjusts the pulse amplitude for a number of the remaining patterns to be applied during training cycle (i.e., N_{amp}). This eliminates the need to re-apply patterns and hence, the training cycle duration does not change and the energy consumption of the system is only marginally increased. The pulse amplitude is changed during phase II of the restore step in fig. 6.13 such that:

- Whenever the applied pattern reduces the RRAM resistance towards its LRS, this change (i.e., ΔP_{LRS}) is amplified. For our case-study system, by increasing the pulse amplitude at the post-neurons by 30 mV, ΔP_{LRS} changes to 5% instead of 0.9%.

- Oppositely, if the applied input pattern increases the RRAM resistance, this modification (i.e., ΔP_{HRS}) is weakened. For our case-study system, by decreasing the pulse amplitude at the pre-neurons by 100 mV, ΔP_{HRS} is lowered to 0.02% instead of 0.2%.

Fig. 6.14 demonstrates the relation between ΔP_{LRS} and N_{amp} in our case-study system. Similar to the study in fig. 6.11a, fig. 6.14a shows that, with fixed $\Delta P_{LRS} = 5\%$, increasing/decreasing N_{amp} from its optimum value of 800 decreases the restored system accuracy. As discussed in section 6.4.2, this is caused by having extra/less RRAM devices close to their LRS, which affects the efficiency in transmitting the pulses between the pre- and post-neurons. Fig. 6.14b illustrates that the optimum number of patterns, for which the pulse amplitude is modified (i. e $N_{amp,opt}$), decreases with the increase in ΔP_{LRS}. This is because, when ΔP_{LRS} is high, the percentage of RRAM devices, which are at their LRS, also grows. Hence, to avoid the reduction in the restored system accuracy, resulting from having too many RRAM devices in their LRS, the $N_{amp,opt}$ decreases.
Fig. 6.15 illustrates the change in the energy consumption resulting from modifying ΔP_{LRS}. The increase in the energy consumption of the system is minimal ($\approx 0.1\%$) since reducing the pulse amplitude of the pre-neurons by 100 mV, as described in phase II in fig. 6.13, decreases their energy consumption by 16%. Hence, the overall percentage of increase in the energy consumption is lowered. Also, the percentage of $N_{amp,opt}$ to the total number of MNIST training patterns is only 2.5% (i.e., 1400 patterns compared to the 60,000 patterns of the training cycle of MNIST dataset). Although a higher voltage is required to increase ΔP_{LRS} to 8% instead of 3% (i.e., 1.0 V instead of 0.93 V), the increase in the energy consumption is lower when ΔP_{LRS}=8%. This is because, when ΔP_{LRS}=8%, the required $N_{amp,opt}$ patterns, during which the pulse amplitude is increased, is 2.3x less than that needed when $\Delta P_{LRS} = 3\%$.

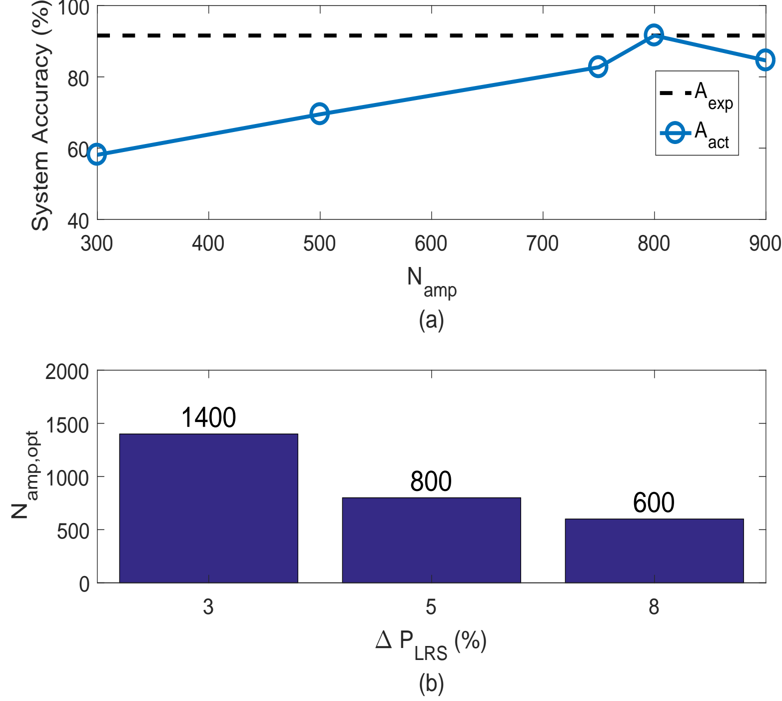

Figure 6.14: Effect of the restore step parameters (i.e., ΔP_{LRS} and N_{amp}) on recovering the degraded system performance: a) N_{amp} with fixed ΔP_{LRS}=5%, b) $N_{amp,opt}$ for various ΔP_{LRS} values.

The results summarized in the graphs in fig. 6.14 and fig. 6.15 are generated assuming the worst-case scenario for the RRAM reliability soft-errors when they occur by the end of the training cycle as explained in section 6.4. Hence, the remaining number of patterns in the training cycle is another parameter that define the ΔP_{LRS} to use. If the RRAM reliability soft-errors are detected, while there are still more than 1400 training patterns to be applied, ΔP_{LRS} can be as small as 3%. However, in the case when there are only 600 patterns remaining in the training cycle, ΔP_{LRS}=8% must be used. If less than 600 training patterns are left after the detection of the RRAM reliability soft-errors, either a higher ΔP_{LRS} than 8% can be used or the last 600 training patterns have to be re-applied as what is described in section 6.4 but with $\Delta P_{LRS} = 8\%$.

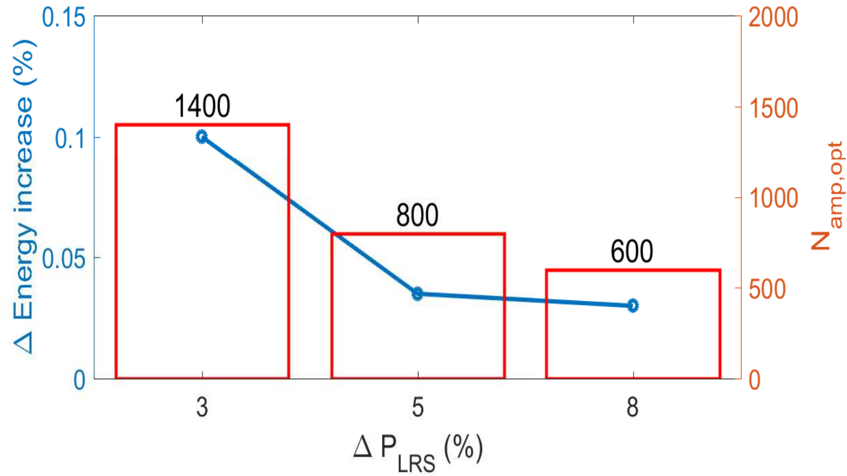

Figure 6.15: Effect of the restore step parameters on the system energy consumption increase. With smaller ΔP_{LRS}, the value of $N_{amp,opt}$ increases as well as the energy consumption of the system. However, the total increase is not high due to small value of $N_{amp,opt}$, for which amplitude is increased, in comparison to the total number of training patterns.

6.6 Comparative Analysis Between the Two Proposed Frameworks

Table 6.3 summarizes the comparison discussion between the basic and modified frameworks in fig. 6.10 and fig. 6.13, respectively. While the results in table 6.3 indicates

Table 6.3: Comparison between the basic and modified frameworks

Parameter	**Basic Framework**	**Modified Framework**
System Delay	$\leq 4\%$	0%
System Energy Consumption	$\leq 20\%$	$< 0.1\%$

that the modified framework is better than the basic methodology in our case-study system, this might not be true for other RRAM-based neuromorphic systems. In particular, although the detection phase in both algorithms is the same and can be implemented us-

ing the circuit described in [166], the restore step in the modified methodology dictates constraints on the DC-DC converter, which generates the various voltage levels required by the algorithm. Accordingly, a modification or an addition of an independent DC-DC converter may be required depending on the voltage levels needed to achieve the ΔP_{LRS} percentages for the RRAM device incorporated in the system [167, 168, 169]. This can, not only increase the energy consumption of the design significantly (since the modified and/or the new DC-DC converter has to be added to each neuron circuit), but also increase the chip area. In our case-study system which is using the HfO_x RRAM device, no modification to the DC-DC converter is required. This is because, in order to achieve the required ΔP_{LRS} (i.e., 3%/5%/8%), the amplitude of the pulses from the post-neurons needs to become 0.93/0.98/1.0 V, respectively. Using digitally controlled low drop-out DC-DC converter, similar to those discussed in [167, 168, 169], the required voltage levels can be generated for the various ΔP_{LRS} percentages in addition to those needed during normal operations. Also, for a neural network structure with many layers of hierarchy (i.e., Convolutional Neural Network (CNN) [170, 171]), adding a dedicated DC-DC converter for each layer of hierarchy might not be a practical solution in terms of design area and energy consumption. In conclusion, choosing the proper framework to use depends on the RRAM device incorporated in the system as well as the system architecture.

6.7 Modifications of the Read and Write Circuits

The modifications to the read and write circuits to implement the proposed framework in sections 6.4 and 6.5, are discussed separately.

6.7.1 Write Circuit Modifications

One of the key modules of the write circuit for RRAM-based neuromorphic system is a digitally controlled low dropout DC-DC converter similar to those discussed in [167, 168, 169]. Those designs are connected to Digital-to-Analog Converter (DAC) which translates a digital input word into a proper output voltage level lower than VDD (i.e., VDD = 1.2V in our case). Using the HfO_x RRAM device SPICE model in [33], the voltage levels of the pre- and post-neurons pulses required for the various percentage of changes in the RRAM state are in the range between [0.75-0.95]V (assuming ΔP_{LRS}=5%). The original voltage levels in our system example, described in [142, 172], are in the range between [0.8-0.9]V. Bounded by the minimum voltage level of NMOS/PMOS transistors used in the DAC, the 4-bit DACs of the DC-DC converter in [167, 168, 169] can provide 16 voltage levels

from the range of VDD (i.e., 1.2V) down to almost 0.4V with an accuracy as small as 50 mV. Thus, in our example, the DAC-controlled DC-DC converter is capable of providing all the required voltage levels without the necessity to modify the write circuit. In case if other voltage levels are required, the DAC design of the DC-DC converter may need to be modified.

6.7.2 Read Circuit Modifications

Fig. 6.16 illustrates the structure of the modified read circuit, which consists of the normal I&F circuit [23, 24, 160] and two newly added modules:

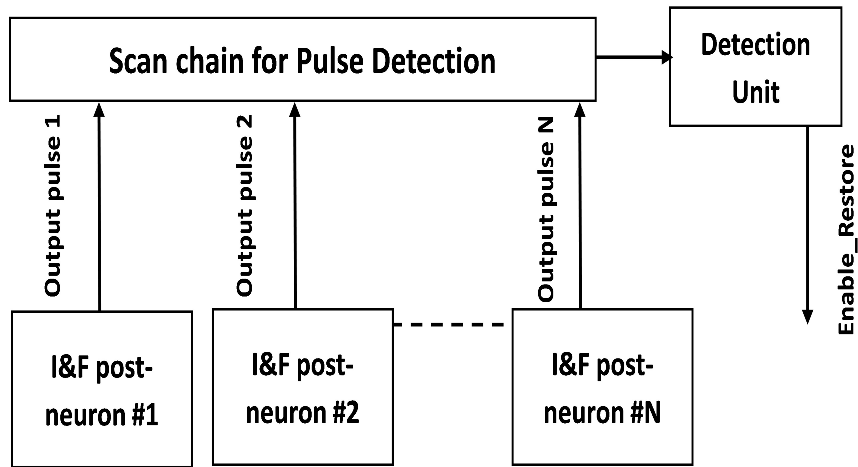

Figure 6.16: Schematic of the modified read circuit. The "scan chain for pulse detection" and "detection unit" are added to the normal I&F circuit of the neuron. "Scan chain" unit is added to collect the pulses generated from the various neurons at different time instances and send them to "detection unit". The "detection unit" counts the pulses sent from the "scan chain" in response to the applied input pattern to determine whether the RRAM array is suffering from soft-errors or not.

- **Scan chain for pulse detection**: This module is used to collect the pulses generated by the different neurons at various time instances and pass them to the detection unit to assess whether RRAM reliability soft-errors have occurred or not as described in sections 6.4 and 6.5.

- **Detection unit**: This module is responsible of counting the pulses coming from the "scan chain" in order to: a) detect if the number of pulses at the post-neurons is less than 5 (i.e., condition C1 of the detection step in fig. 6.12) which indicates RRAM reliability soft-errors have occurred), and b) find whether the restore step needs to be initiated or not based on the number of times the input pattern has been re-applied (i.e., condition C2 of the detection step in Fig. 6.12).

The design and operation of the "scan chain" and "detection unit" are discussed separately.

Scan Chain Circuit

Fig. 6.17 shows the schematic of the scan chain circuit.

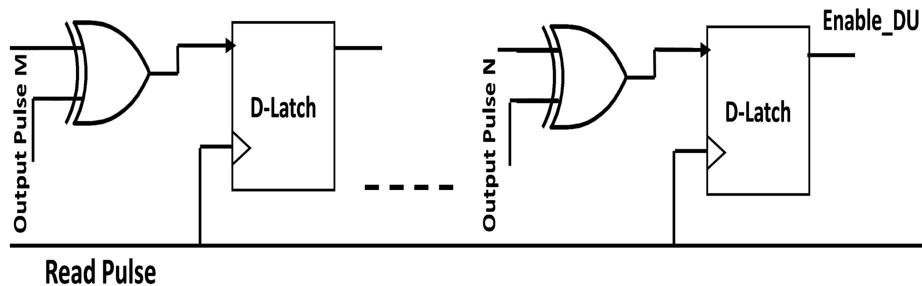

Figure 6.17: "Scan chain" structure. The XOR gates are used to insert delays between the stages of the chain.

Since pulses can occur from any neuron asynchronously, the D-Latch of each stage of the "scan chain" is enabled throughout the duration of read operation (i.e., 350 ms as per [142]). In our system example, given that the duration of each pulse is 20 ms, our BRIAN simulations show that the post-neurons can generate pulses either simultaneously or with a minimum time split of around 10 μs. Since the delay of the entire scan chain is around 400 ns (i.e., $((Delay_{D-latch} + Delay_{XOR})$*number of stages = 1 ns* 400)), any two pulses separated by $\geq$ 10 μs can travel through the scan chain without causing errors at the "detection unit" counters, which can occur due to the inability of counters in distinguishing overlapped pulses. To overcome the overlap between simultaneously generated pulses, the XOR gates at the input of each D-Latch in fig. 6.17. Fig. 6.18 illustrates the worst-case scenario, when two pulses from two adjacent stages (i.e., stages M-1 and M) are generated simultaneously.

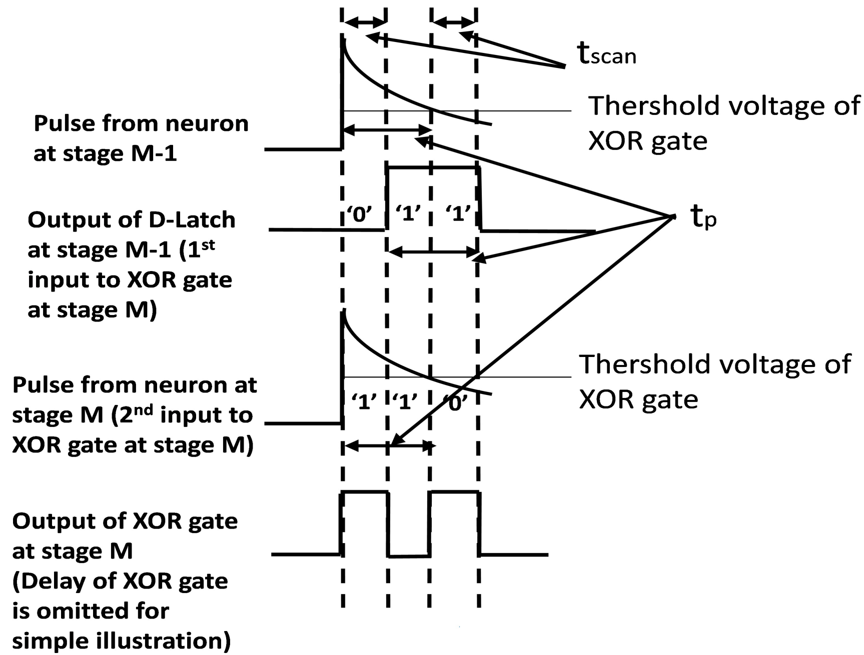

Figure 6.18: XOR gate output signals in case when the pulses observed at post-neurons occur simultaneously. Due to the delay of XOR gate, even if pulses from adjacent neurons are generated simultaneously, they are still going to be counted separately at "detection unit".

The parameter t_p in fig. 6.18 defines the period of write pulse, during which the pulse amplitude is higher than the threshold voltage of XOR gate (i.e., logic '1'). Although t_p is about 4 ms, fig. 6.18 is not drawn to scale in order to focus mainly on the effect of introducing the delay of XOR and D-latch (i.e., t_{scan} = 1ns) on insuring the correct operation of the framework in worst-case scenario. This large delay of t_{scan} is achieved by increasing the channel length of all the transistors (i.e., decreases W/L) used in "scan chain" design by a factor of 8. Hence, a delayed version of the same pulse is applied to the input of XOR gate at stage M, which guarantees the generation of two pulses at the output of XOR gate with a time split of 1 ns. As a result, the correct number of pulses (i.e., 2) are computed by the "detection unit" counters.

However, the introduction of "scan chain" increases the delay of read operation as it needs to account for the worst-case scenario when a pulse is generated at the end of read operation cycle at the "scan chain" first stage. In this case, the delay of the read operation is augmented by the delay of each stage of the "scan chain" multiplied by the number of

post-neuron stages. This increase is less than 0.1%.

Detection Unit Circuit

Fig. 6.19 shows the structure of the detection unit. This unit consists mainly of: a) "3-bit counter" which is used to detect whether the number of received pulses from the "scan chain" is less than 5 or not (i.e., condition C1 of the detection step in fig. 6.12), and b) "6-bit counter" which is used to recognize whether the input pattern, causing the generation of less than 5 pulses, has been already re-applied for more than 50 times or not (i.e., condition C2 of the detection step in fig. 6.12).

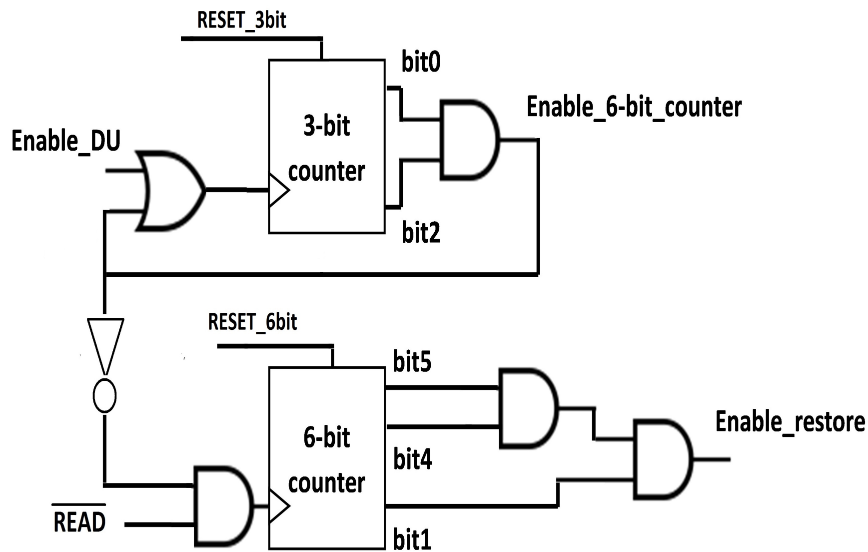

Figure 6.19: "Detection unit" structure. The unit consists basically of two counters: "3-bit counter", which checks whether more than 5 pulses have been generated for each input pattern, and "6-bit counter" which checks, in case if the same input pattern is re-applied (due to generating less than 5 pulses), the number of retrying the same pattern is higher than 50 to trigger the restore phase of the algorithms described in sections 6.4 and 6.5.

The OR-gate in fig. 6.19 disables the "3-bit counter" once 5 pulses are computed from the sequence of pulses passed by the "scan chain" (i.e., 'Enable_DU' signal in fig. 6.19), which could be as many as 200 pulses. This is needed to: a) save energy consumption, b) speed-up the operation of "detection unit", and c) prevent the value of the counter to be

reset when more than 8 pulses are passed by the scan chain. The output from the "3-bit counter" (i.e., 'Enable_6-bit_counter' signal in fig. 6.19) is then checked at the end of read operation by anding it with the control signal '$\bar{READ}$' signal, which is at logic '1' when the read operation is finished. This is needed to enable the "6-bit counter". If the value of "6-bit counter" reaches 50 (i.e., condition C2 in the detection step in fig. 6.12), the restore step of the proposed framework is enabled. Fig. 6.20 demonstrates the operation of "detection unit", in case if the number of pulses passed by the "scan chain" is 4 (i.e., $<$ 5).

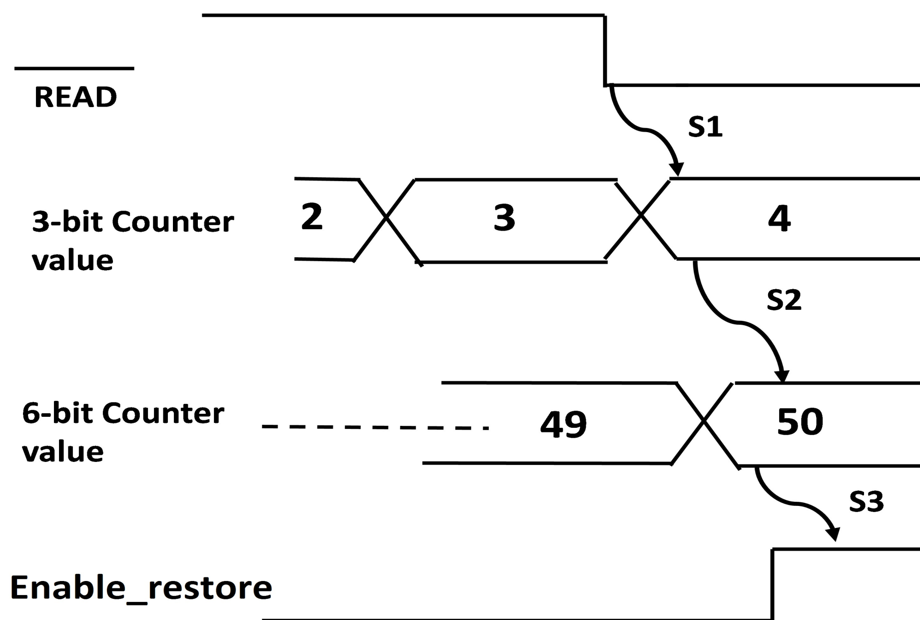

Figure 6.20: Waveforms for the operation of "detection unit" when the restore step of the framework is initiated. In this case, it is assumed that the input pattern was reapplied 49 times.

The steps S1-S3 in fig. 6.20 describes the sequence of "detection unit" operations. After the value of "3-bit counter" is checked at the end of read operation (S1), the "6-bit counter" is incremented by 1 (S2). Since the value of enquote6-bit counter has reached the limit value of 50, the restore step of the framework, shown in fig. 6.12, is initiated (S3). In this case, the 'RESET' signals of the "3-bit" and "6-bit" counters (i.e., 'RESET_3bit' and 'RESET_6bit' signals in fig. 6.20) are also raised to re-initialize the counters. If the restore step is not triggered (i.e., number of times an input pattern has been re-applied is

less than 50 or the number of pulses passed by the scan chain is higher than 5), only the "3-bit counter" will be reset at the end of "detection unit" operation.
It is worth mentioning that the counters used in the "detection unit" are actually asynchronous ripple counters to reduce the overhead energy consumption of the "detection unit" by removing the need to have an extra clock signal.

6.8 Simulation Results for the Modified Read Circuit

The simulation results for the read circuit modifications, required to support the various operations of newly proposed framework, is discussed in this section. The modified write circuit simulation results are already discussed in sections 6.5 and 6.6. As mentioned in section 6.7.2, the read operation delay is increased to account for: a) the delay introduced by the "scan chain" and b) the worst-case scenario when a pulse, resulting from the neuron attached to the first stage of the "scan chain", is generated at the end of read operation. The percentage of increase in the delay is calculated using equation 6.7:

$$T_{read,incr} = \frac{T_{read} + num_{st} * T_{scan} + T_{DU}}{T_{read}} - 1 \tag{6.7}$$

where:

- T_{read}: is the duration of read pulse, which is equal to the time required for applying an input pattern to the system in [142] (i.e, 350 ms).
- T_{scan}: is the delay of each stage of the "scan chain".
- num_{st}: represents the number of stages of the "scan chain", which is equal to the number of post-neurons.
- T_{DU}: is the delay of "detection unit".

Our SPICE simulation results show that, for a 65 nm technology, the T_{scan} and the T_{DU} are equal to 1 ns and 1.2 ns, respectively. Given that the number of post-neurons for the neuromorphic system in [142] is 400, the percentage of increase in the read operation delay (i.e., $T_{read,incr}$) is less than 0.1%. This is due to the fact that the read operation duration is 8 order of magnitude (i.e., 350 ms) larger than the delay introduced by the "scan chain" (i.e., 1 ns) and "detection unit" (i.e., 1.2 ns).

To compute the increase in read energy consumption, equation 6.8 is used

$$E_{read,incr} = \frac{num_{st} * (E_{read,avg} + E_{scan,avg}) + E_{DU,avg}}{num_{st} * E_{read,avg}} - 1 \tag{6.8}$$

where:

- $E_{read,avg}$: refers to the average energy consumption consumed by the I&F circuits.
- $E_{scan,avg}, E_{DU,avg}$: are the average energy consumed by the "scan chain" and "detection unit", respectively.
- num_{st}: is the same parameter as that used in eq. 6.7.

Using the SPICE simulation results and substituting them in eq. 6.8, it is found that the increase in energy consumption is around 1.1%. This is due to the fact that the energy drawn by each of the I&F circuit of post-neurons is much higher compared to that of the "scan chain" and "detection unit". The I&F circuit mainly consists of: a) Schmitt trigger, and b) shift register to count the number of pulses generated by each post-neuron to identify which neurons represent a certain digit of the MNIST dataset [160]. Those circuits draw large currents from the power supply leading to a large energy consumption compared to that of the "scan chain" and "detection unit" (i.e., $E_{read,avg} = 0.06$ mJ; $E_{scan,avg} = 0.0007$ mJ = 1.1% of $E_{read,avg}$; $E_{DU,avg} = 87.2\mu$J = 0.1% of $E_{read,avg}$).

6.9 Summary

In this chapter, a novel modeling technique, based on a combination of SPICE and BRIAN system level simulations, is proposed to estimate the degradation in RRAM-based neuromorphic system performance due to RRAM reliability soft-errors. Using the suggested modeling methodology, we show that the accuracy of a case-study RRAM-based neuromorphic system in recognizing the MNIST dataset can degrade by more than 48%. To restore the loss in the system performance, two frameworks are proposed in this chapter to automatically detect and fix the degradation in the system accuracy due to the RRAM reliability soft-errors. Using the case-study system, our BRIAN system level simulations demonstrate that the newly suggested frameworks can increase of the energy consumption of the system by as low 0.1% and with possibly no increase in its training cycle duration. Choosing between the two proposed mitigation techniques depends on the system structure and RRAM device incorporated in its design.

Finally, the required modifications to the read and write circuits to support these new frameworks are explained. In our example, the SPICE simulation results demonstrate that the increase in read delay and energy consumption is less than 0.1% and 1.1%, respectively. Similarly, for the write operation, the increase in energy consumption is less than 0.1%, while there is no change to its delay. Using a RRAM-based 1T1R arrays not incorporating the HfO_x RRAM device may require changes in the computed biasing conditions in this work. However, the main concepts and methodologies explained throughout the sections of this chapter are still applicable to any RRAM-based neuromorphic system.

Chapter 7

Conclusion and Future Work

The main focus of this thesis is related to providing initial solutions to reliably incorporate the RRAM devices in current and futuristic platforms used to run machine learning applications (i.e., GPU-based and neuromophic-based platforms). The proposed research offers solutions on how to address the RRAM soft-errors which is one of the main challenges hindering the usage of RRAM devices in production. Fig. 7.1 summarizes the work and contributions discussed in the different chapters of this thesis.
As illustrated in fig. 7.1, the main contributions of our work can be categorized into two main groups:

- **Reliably using RRAM devices in current-platforms running machine learning applications**: This is related to the work discussed in chapter 3, where we have discussed the proposed 8T1R NV-SRAM which can be incorporated in the GPU-based platforms. The contributions from this work are:
  - A new RRAM-based NV-SRAM design, which has minimal impact on the basic SRAM read and write operations, has been presented.
  - Compared to previously proposed RRAM-based NV-SRAM cells, the energy required to store/restore data on the RRAM device in the 8T1R cell has decreased by 60%/70%, respectively. This has led to higher resilience to reliability soft-errors due to the reduction in the heat generated in the conductive filaments of the RRAM device.
- **Reliably using RRAM device in futuristic platforms running machine learning applications**: This represents the work discussed in chapters 4, 5, and

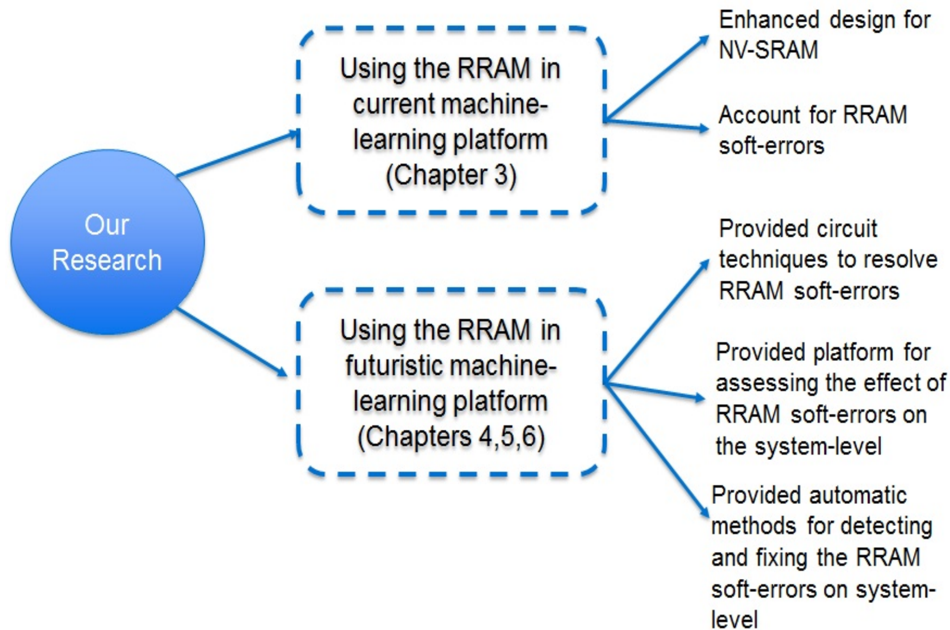

Figure 7.1: Contributions of the work presented in the thesis.

6, where we have provided circuit and system solutions to address RRAM reliability and radiation soft-errors. These soft-errors are major challenges hindering the adaptation of RRAM technology in production. The contributions from this work are:

- Circuit solutions are provided for addressing reliability and radiation soft-errors of the RRAM arrays with minimal impact on the main read/write operations of the memory array.
- A systematic framework is developed for assessing the impact of RRAM soft-errors on the performance of any RRAM-based neuromorphic systems.
- Two system level algorithms are proposed to detect and fix the degradation in the RRAM-based neuromorphic performance due to RRAM soft-errors. The circuits required for supporting the proposed algorithms have been also provided.

It is worth mentioning that some of the qualitative results are obtained from running simulations using specific system architectures (e.g., the multi-perceptron system architec-

ture in chapter 6). While the concepts behind the proposed methodologies for addressing the RRAM soft-errors should apply when the detailed implementation of the system changes, it is expected that the qualitative results may vary. In addition to this, even though we have run multiple SPICE and system level simulations using accredited models and test benches, fabricating the various proposed techniques in this work is essential in order to: a) verify their validity with silicon data, and b) enhance the suggested solutions to account for other RRAM device shortcomings including its sensitivity to temperature fluctuations and process variations, and the instability in its switching characteristics due to the stochastic nature of oxygen vacancies. This should be one of main focuses of the future work. In addition to this, future research on both the software and hardware side for more advanced Deep Neural Network (DNN) structures is going to be very active. Software companies (e.g., Google, Facebook, Amazon) will continue to provide more advanced and standardized techniques to design more complicated DNN systems through platforms such as TensorFlow [173] and Caffe [174]. ICs design and manufacturing companies (e.g., Qualcomm, Nvidia, Intel, TSMC, Samsung, GlobalFoundries) will need to provide the means to implement and fabricate chips supporting the complicated designs described through the software. Hence, the future work should focus on:

- Fabricating the proposed circuit and system solutions provided in this work is required to verify their validity and enhance them to account for other RRAM technology shortcomings.

- From software point of view, the current active research will need to continue in the field of optimizing the learning algorithms to offer more advanced DNN systems [175, 176]. This is in addition to the continuation of recent efforts from companies to standardize the libraries in platforms (e.g, TensorFlow [173], Caffe [174]) to open doors for more people to innovate and come up with even more elaborate network structures.

- From device point of view, research activities need to proceed to investigate the possibility to use other materials to build more reliable, less energy consuming, and smaller RRAM device. Proper read and write circuits need to be modified accordingly.

- From system design point of view, research work is required to study the possibility for integrating the RRAM devices in more advanced neuromorphic systems such as DNN. The work presented in this thesis could be used as an indication of how such study can be done. Yet, our proposed solutions and techniques might need to be modified depending on the structure of the neuromorphic system and the RRAM devices incorporated in its design.

References

[1] T.-C. Chen, “Where is CMOS going: trendy hype versus real technology,” in *Proceedings of the International Solid-State Circuits Conference (ISSCC)*, pp. 22 – 28, 2006.

[2] S. R. Nassif, “Modeling and analysis of manufacturing variations,” in *Proceedings of IEEE Custom Integrated Circuits conference*, pp. 223 – 228, 2001.

[3] B. Wong *et al.*, *Nano-CMOS Circuit and Physical Design.* Wiley-Interscience, 2004.

[4] R. O. Topaloglu, “Device and circuit implications of double-patterning A designer’s perspective,” in *Proceedings of International Symposium on Quality Electronic Design (ISQED)*, pp. 1–4, 2011.

[5] H. Yaegashi, “The important challenge to optimize the double patterning process toward 22nm node and beyond,” in *Proceedings of International Symposium on VLSI Technology, Systems and Applications (VLSI-TSA)*, pp. 1–3, 2011.

[6] V. Joshi *et al.*, “Design-patterning co-optimization of SRAM robustness for double patterning lithography,” in *Proceedings of Asia and South Pacific Design Automation Conference (ASP-DAC)*, pp. 713–718, 2012.

[7] J. Sun *et al.*, “Post-routing layer assignment for double patterning,” in *Proceedings of Asia and South Pacific Design Automation Conference (ASP-DAC)*, pp. 793 – 798, 2011.

[8] X. Qi *et al.*, “Efficient subthreshold leakage current optimization - Leakage current optimization and layout migration for 90- and 65- nm ASIC libraries,” *IEEE Circuits and Devices Magazine , Vol. 22, No. 5*, p. 3947, September 2006.

[9] T. D. Burd *et al.*, “A dynamic voltage scaled microprocessor system,” *IEEE Journal of Solid-State Circuits , Vol. 35, No. 11*, pp. 1571 – 1580, November 2000.

[10] K. J. Nowka *et al.*, "A 32-bit PowerPC system-on-a-chip with support for dynamic voltage scaling and dynamic frequency scaling," *IEEE Journal of Solid-State Circuits , Vol. 37, No. 11*, pp. 1441 – 1447, November 2002.

[11] M.-F. Chang *et al.*, "Wide VDD embedded asynchronous SRAM with dual-mode self-timed technique for dynamic voltage systems," *IEEE Trans.Circuits Syst. I, Reg.Papers, Vol. 56, No. 8*, pp. 1657 – 1667, August 2009.

[12] K. Zhang *et al.*, "SRAM design on 65-nm CMOS technology with dynamic sleep transistor for leakage reduction," *IEEE Journal of Solid-State Circuits, Vol. 40, No. 4,*, pp. 895 – 901, April 2005.

[13] C. H. Kim *et al.*, "PVT-Aware leakage reduction for on-die caches with improved read stability," *IEEE Journal of Solid-State Circuits, Vol. 41, No. 1,*, pp. 170 – 178, January 2006.

[14] J.-M. Choi *et al.*, "Zero-Sleep-Leakage Flip-Flop Circuit With Conditional-Storing Memristor Retention Latch," *IEEE Transactions on Nanotechnology, Vol. 11, No. 2*, pp. 360 – 366, March 2012.

[15] S. Onkaraiah *et al.*, "Bipolar ReRAM Based non-volatile flip-flops for low-power architectures," in *IEEE International Conference on New Circuits and Systems (NEWCAS)*, pp. 417 – 420, 2012.

[16] I. Kazi *et al.*, "A ReRAM-Based Non-Volatile Flip-Flop with Sub-VT Read and CMOS Voltage-Compatible Write," in *IEEE International Conference on New Circuits and Systems (NEWCAS)*, pp. 1 – 4, 2013.

[17] W. Wang *et al.*, "Nonvolatile SRAM cell," in *Proceedings of IEEE International Electron Devices Meeting (IEDM)*, pp. 27 – 30, 2006.

[18] S. Agawal *et al.*, "The energy scaling advantages of RRAM crossbars," in *Berkeley Symposium on Energy Efficient Electronic Systems (E3S)*, pp. 1–3, 2015.

[19] B. Li *et al.*, "MErging the Interface: Power, area and accuracy co-optimization for RRAM crossbar-based mixed-signal computing system," in *IEEE Design Automation Conference (DAC)*, pp. 1–6, June 2015.

[20] C. Hermes *et al.*, "Fast pulse analysis of TiO2 based RRAM nano-crossbar devices," in *Annual Non-Volatile Memory Technology Symposium (NVMTS)*, pp. 1–4, November 2011.

[21] C. Nauenheim *et al.*, "Nano-Crossbar Arrays for Nonvolatile Resistive RAM (RRAM) Applications," in *IEEE Conference on Nanotechnology*, pp. 464–467, August 2008.

[22] S. Yu *et al.*, "An Electronic Synapse Device Based on Metal Oxide Resistive Switching Memory for Neuromorphic Computation," *IEEE Transactions on Electron Devices, Vol. 58, No. 8*, pp. 2729–2737, August 2011.

[23] D. Kadetotad *et al.*, "Parallel Architecture With Resistive Crosspoint Array for Dictionary Learning Acceleration, Vol. 5, No. 2," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, pp. 194–204, June 2015.

[24] S. Park *et al.*, "RRAM-based synapse for neuromorphic system with pattern recognition function," in *IEEE International Electron Devices Meeting (IEDM)*, pp. 10.2.1–10.2.4, December 2012.

[25] Z. Chen *et al.*, "Optimized learning scheme for grayscale image recognition in a RRAM based analog neuromorphic system," in *IEEE International Electron Devices Meeting (IEDM)*, pp. 17.7.1–17.7.4, December 2015.

[26] S. Yu *et al.*, "A neuromorphic visual system using RRAM synaptic devices with Sub-pJ energy and tolerance to variability: Experimental characterization and large-scale modeling," in *IEEE International Electron Devices Meeting (IEDM)*, pp. 10.4.1–10.4.4, December 2012.

[27] Y.-W. Chin *et al.*, "Point Twin-bit RRAM in 3D Interweaved Cross-point Array by Cu BEOL Process," in *Proc. IEEE Int. Electron Devices Meeting (IEDM)*, pp. 6.4.1 – 6.4.4, December 2014.

[28] Liaw *et al.*, "Nonvolatile 3D-FPGA with monolithically stacked RRAM-based configuration memory," in *Proc. IEEE Int. Solid-State Circuits Conference (ISSCC)*, pp. 406 – 408, 2012.

[29] Wong *et al.*, "Metal oxide RRAM," *Proceedings of the IEEE, Vol.100, No. 6,*, pp. 1951 – 1970, June 2012.

[30] S. H. Jo, "Nanoscale memristive devices for memory and logic applications," *PhD thesis, University of Michigan*, 2010.

[31] J. P. Strachan *et al.*, "State dynamics and modeling of tantalum oxide memristors," *IEEE transactions on electron devices, Vol. 60, No.7*, pp. 2194 – 2202, July 2013.

[32] Z. Jiang *et al.*, "Verilog-A compact model for oxide-based resistive random access memory (RRAM)," in *Proc. of Int. Conference on Simulation of Semiconductor Processes and Devices (SISPAD)*, pp. 41 – 44, September 2014.

[33] P.-Y. Chen and S. Yu, "Compact modeling of RRAM devices and its applications in 1T1R and 1S1R array design," *IEEE Transactions on Electron Devices, Vol. 62, No. 12,*, pp. 4022 – 4028, December 2015.

[34] C. F. Liao *et al.*, "Zero static-power 4T SRAM with self-inhibit resistive switching load by pure CMOS logic process," in *2016 IEEE International Electron Devices Meeting (IEDM)*, pp. 16.5.1–16.5.4, December 2016.

[35] Y. Chen *et al.*, "Balancing SET/RESET Pulse for $> 10^{10}$ Endurance in HfO2/Hf 1T1R Bipolar RRAM," *IEEE Transactions on Electron devices , Vol. 59, No. 12*, pp. 3243 – 3249, December 2012.

[36] Y. Y. Chen *et al.*, "Understanding of the Endurance Failure in Scaled $HfO2$-based 1T1R RRAM through Vacancy Mobility Degradation," in *IEEE International Electron Devices Meeting (IEDM)*, pp. 20.3.1 – 20.3.4, December 2012.

[37] Y. Y. Chen *et al.*, "Improvement of data retention in HfO2 / Hf 1T1R RRAM cell under low operating current," in *IEEE International Electron Devices Meeting (IEDM)*, pp. 10.1.1 – 10.1.4, December 2013.

[38] Y. Y. Chen *et al.*, "Postcycling LRS Retention Analysis in HfO2/Hf RRAM 1T1R Device," *IEEE Electron Device Letters, Vol. 34, No. 5*, pp. 626 – 628, May 2013.

[39] S. Ambrogio *et al.*, "Data retention statistics and modeling in HfO2 resistive switching memories," in *IEEE International Reliability Physics Symposium (IRPS)*, pp. M.Y.1 – M.Y.6, April 2015.

[40] W. G. Bennett *et al.*, "Single- and Multiple-Event Induced Upsets in HfO2/Hf 1T1R RRAM," *IEEE Transactions on Nuclear Science, Vol. 61, No. 4,*, pp. 1717 – 1725, August 2014.

[41] R. Liu *et al.*, "Investigation of Single-Bit and Multiple-Bit Upsets in Oxide RRAM-Based 1T1R and Crossbar Memory Arrays," *IEEE Transactions on Nuclear Science, Vol. 62, No. 5,*, pp. 2294 – 2301, October 2015.

[42] S. Thoziyoor *et al.*, "CACTI 5.3 technical report," in *HP Labs, Palo Alto, CA, Tech. Rep. HPL-2008-20*, 2008.

[43] D. Goodman and R.Brette, "Brian: a simulator for spiking neural networks in Python," *Frontiers in Neuroinformatics, Vol. 2*, p. 5, 2008.

[44] Y. LeCun *et al.*, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE, Vol. 86, No. 11*, pp. 2278–2324, November 1998.

[45] L. O. Chua, "Memristor-the missing circuit element," in *IEEE Transactions on Circuit Theory, Vol. 18, No. 5*, pp. 507–519, September 1971.

[46] D. B. Strukov *et al.*, "The missing memristor found," in *Nature, Vol. 453, No. 7191*, pp. 80–83, 2008.

[47] J. Yang *et al.*, "Memristive devices for computing," *Nature Nanotechnology, Vol. 8*, p. 1324, January 2013.

[48] L. O. Chua and S. M. Kang, "Memristive Devices and systems," in *Proc. IEEE, Vol.64*, pp. 209 – 223, 1976.

[49] M. D. Pickett *et al.*, "Switching dynamics in titanium dioxide memristive devices," *Journal of Applied Physics, Vol. 106, No. 7*, pp. 074508 – 074508–6, October 2009.

[50] J. J. Yang *et al.*, "The mechanism of electroforming of metal oxide memristive switches," *Nanotechnology, Vol. 20, No. 21*, May 2009.

[51] D. Strukov *et al.*, "Exponential ionic drift: fast switching and low volatility of thin-film memristors," *Journal of Applied Physics, Vol. A94*, p. 515519, 2009.

[52] J. Strachan *et al.*, "The switching location of a bipolar memristor: chemical, thermal and structural mapping," *Nanotechnology, Vol. 22, No. 25*, p. 15, June 2011.

[53] S. H. Chang *et al.*, "Effects of heat dissipation on unipolar resistance switching in Pt/NiO/Pt capacitors.," *Applied Physics letters, Vol. 92*, p. 114, February 2008.

[54] D. Sturkov *et al.*, "Thermophoresis/diffusion as a plausible mechanism for unipolar resistive switching in metaloxidemetal memristors," *Applied Physics A, Vol. 106, No. 4*, pp. 509 – 520, March 2012.

[55] S. H. Chang *et al.*, "Occurrence of Both Unipolar Memory and Threshold Resistance Switching in a NiO Film," *Physical review letters, Vol. 102, No. 2*, January 2009.

[56] E. Kyriakides *et al.*, "Low-cost, CMOS compatible, Ta2O5-based hemi-memristor for neuromorphic circuits," *Electronics letters, Vol. 48, No.23*, pp. 1451 – 1452, November 2012.

[57] P. R. Mickel *et al.*, “A physical model of switching dynamics in tantalum oxide memristive devices,” *Applied Physics Letters, Vol. 102, No. 22*, June 2013.

[58] S. H. Jo *et al.*, “Si Memristive devices applied to memory and neuromorphic circuits,” in *Proceedings of International Symposium on Circuits and Systems (ISCAS)*, pp. 13 – 16, 2010.

[59] Y.-F. Chang *et al.*, “Study of SiOx-based Complementary Resistive Switching Memristor,” in *Proceedings of Device Research Conference (DRC)*, pp. 49 – 50, 2012.

[60] M. J. Marinella *et al.*, “Resistive Switching in Aluminum Nitride,” in *Proceedings of Device Research Conference (DRC)*, pp. 89 – 90, 2012.

[61] S. Zou *et al.*, “Resistive switching characteristics in printed cu/cuo/(ago)/ag memristors,” *Electronics letters, Vol. 49, No. 13*, pp. 829 – 830, June 2013.

[62] Z. Fan *et al.*, “Resistive switching in copper oxide nanowirebased memristor,” in *Proceedings of IEEE International Conference on Nanotechnology (IEEE-NANO)*, pp. 1 – 4, 2012.

[63] N. S. Kamarozaman *et al.*, “Effect of Annealing Duration on the Memristive Behavior of Pt/TiO2/ITO Memristive Device,” in *Proceedings of IEEE International Conference on Semiconductor Electronics (ICSE)*, pp. 703 – 706, 2012.

[64] Y. S. Chen *et al.*, “Challenges and Opportunities for HfOX Based Resistive Random Access Memory,” in *IEDM Tech. Dig.2011*, p. 717, 2011.

[65] S.-S. Sheu *et al.*, “A 4Mb embedded SLC resistive-RAM macro with 7.2ns read-write random-access time and 160ns MLC-access capability,” in *International Conference on Solid-State Circuit Conference Technology (ICSSC)*, pp. 200 – 202, 2011.

[66] A. Kumar *et al.*, “Fabrication and characterization of the zno-based memristor,” in *Proceedings of International Conference on Emerging Electronics (ICEE)*, pp. 1 – 3, 2012.

[67] W. Wang *et al.*, “Memristive Behavior of ZnO/Au Film Investigated by a TiN CAFM Tip and Its Model Based on the Experiments,” *IEEE transactions on Nanotechnology, Vol. 11, No. 6*, pp. 1135 – 1139, November 2012.

[68] T. V. Kundozerova *et al.*, “Anodic Nb2O5 Nonvolatile RRAM,” *IEEE transactions on electron devices, Vol. 59, No.4*, pp. 1144 – 1148, April 2012.

[69] A. S. Oblea *et al.*, "Silver Chalcogenide Based Memristor Devices," in *Proceedings of International Joint Conference on Neural Networks (IJCNN)*, pp. 1 – 3, 2010.

[70] G. Piccolboni *et al.*, "Investigation of the potentialities of Vertical Resistive RAM (VRRAM) for neuromorphic applications," in *2015 IEEE International Electron Devices Meeting (IEDM)*, pp. 17.2.1–17.2.4, December 2015.

[71] G. W. Burr *et al.*, "Large-scale (512kbit) integration of multilayer-ready access-devices based on mixed-ionic-electronic-conduction (miec) at 100pp. 41–42, June 2012.

[72] K. S. Li *et al.*, "Study of sub-5 nm rram, tunneling selector and selector less device," in *2015 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 385–388, May 2015.

[73] J. J. Huang *et al.*, "Bipolar nonlinear ni/tio2/ni selector for 1s1r crossbar array applications," *IEEE Electron Device Letters, Vol.32, No.10*, pp. 1427–1429, Oct 2011.

[74] S. H. Jo *et al.*, "3d-stackable crossbar resistive memory based on field assisted superlinear threshold (fast) selector," in *2014 IEEE International Electron Devices Meeting*, pp. 6.7.1–6.7.4, Dec 2014.

[75] H. Li and Y. Chen, *Nonvolatile Memory Design: Magnetic, Resistive, and Phase Change*. CRC Press, 2012.

[76] J.-M. Choi *et al.*, "PCRAM flip-flop circuits with sequential sleep-in control scheme and selective write latch," *Journal of Semiconductor Technology and Science, Vol. 13, No. 1*, pp. 58 – 64, February 2013.

[77] U. Russo *et al.*, "Modeling of programming and read performance in phase change memories - part II: Program disturb and mixed scaling approach," *IEEE Trans. Electron Devices., Vol. 55, No. 2*, pp. 515 – 522, February 2008.

[78] A. Pirovano *et al.*, "Scaling analysis of phase-change memory technology," in *IEDM Tech. Dig.*, pp. 29.6.1–29.6.4, 2003.

[79] A. Pirovano *et al.*, "Low-field amorphous state resistance and threshold voltage drift in chalcogenide materials," *IEEE Trans. Electron Devices., Vol. 51, No. 5*, pp. 714 – 719, May 2004.

[80] I. V. Karpov *et al.*, "Fundamental drift of parameters in chalcogenide phase change memory," *Journal of Applied Physics, Vol. 102, No. 12*, p. 124503, December 2007.

[81] W. Zhao *et al.*, "TAS-MRAM based Non-volatile FPGA logic circuit," in *International Conference on Field-Programmable Technology (ICFPT 2007)*, pp. 153–160, 2007.

[82] A. D. Smith *et al.*, "Non-volatile Spin-Transfer Torque RAM (STT-RAM): An analysis of chip data, thermal stability and scalability," in *IEEE International Memory Workshop (IMW)*, pp. 1–3, 2010.

[83] M. Hosomi *et al.*, "A Novel Nonvolatile Memory with Spin Torque Transfer Magnetization Switching: Spin-RAM," in *IEDM Tech. Dig.*, pp. 459 – 462, 2005.

[84] S. Shin *et al.*, "Analysis of passive memristive devices array: data-dependent statistical model and self-adaptable sense resistance for RRAMs," *Proceedings of the IEEE, Vol. 100, No. 6*, pp. 2021–2032, June 2012.

[85] C. Yakopcic *et al.*, "Analysis of a Memristor based 1T1M Crossbar Architecture," in *Proceedings of International Joint Conference on Neural Networks (IJCNN)*, pp. 3243 – 3247, 2011.

[86] Y. Cassuto *et al.*, "Sneak-Path Constraints in Memristor Crossbar Arrays," in *Proceedings of IEEE International Symposium on Information Theory (ISIT)*, pp. 156 – 160, 2013.

[87] M. J. Lee *et al.*, "2-stack 1D-1R cross-point structure with oxide diodes as switch elements for high density resistance RAM applications," in *Proceedings of IEEE International Electron Devices Meeting (IEDM)*, pp. 711 – 714, 2007.

[88] B. Mohammad *et al.*, "Hybrid Memristor-CMOS Memory Cell: Modeling and Design," in *Proceedings of International Conference on Microelectronics (ICM)*, pp. 1 – 6, 2011.

[89] P. O. Vontobel *et al.*, "Writing to and reading from a nano-scale crossbar memory based on memristors," *Nanotechnology, Vol. 20, No. 42*, October 2009.

[90] P. P. Sotiriadis, "Information capacity of nanowire crossbar switching networks," *IEEE Transactions on Information Theory, Vol. 52, No. 7*, p. 3019 3032, July 2006.

[91] C. Xu *et al.*, "Design implications of memristor-based RRAM cross-point structures," in *Proceedings of Design Automation and Test in Europe Conference and Exhibition (DATE)*, pp. 1 – 6, 2011.

[92] P.-F. Chiu *et al.*, "Low Store Energy, Low VDDmin, 8T2R Nonvolatile Latch and SRAM With Vertical-Stacked Resistive Memory (Memristor) Devices for Low Power Mobile Applications," *IEEE Journal of Solid-State Circuits, Vol. 47, No. 6,*, pp. 1483 – 1496, June 2012.

[93] S. Yamamoto *et al.*, "NonvolatileSRAM (NV-SRAM) using functional MOSFET merged with resistive switching devices," in *Proceedings of IEEE Custom Integrated Circuits Conference (CICC)*, pp. 531 – 534, 2009.

[94] M. F. Chang *et al.*, "Endurance-aware circuit designs of nonvolatile logic and nonvolatile SRAM using resistive memory (memristor) device," in *Proceedings of Asia and South Pacific Design Automation Conference (ASP-DAC)*, pp. 329 – 334, 2012.

[95] T. Shirai and K. Usami, "Hybrid design of dual Vth and power gating to reduce leakage power under Vth variations," in *Proceedings of the International System On Chip (SoC) Design Conference*, pp. I–310 – I–313, 2008.

[96] S. H. Jo *et al.*, "Nanoscale memristor device as synapse in Neuromorphic systems," *Nano letters, Vol. 10, No. 4*, pp. 1297 – 1301, March 2010.

[97] S. P. Adhikari *et al.*, "Memristor Bridge Synapse-Based Neural Network and its learning," *IEEE transactions on Neural Networks and Learning systems, Vol. 23, No. 9*, pp. 1426 – 1435, September 2012.

[98] W. Lu *et al.*, "Two-Terminal Resistive Switches (Memristors) for Memory and Logic Applications," in *Proceedings of Asia and South Pacific Design Automation Conference (ASP-DAC)*, pp. 217 – 223, 2011.

[99] D. Binder *et al.*, "Satellite anomalies from galactic cosmic arrays," *IEEE Transactions on Nuclear Science, Vol. 22*, pp. 2675 – 2680, December 1975.

[100] C. Slayman, "Soft errors - Past history and recent discoveries," in *IEEE International Integrated Reliability Workshop Final Report*, pp. 25–30, October 2010.

[101] F. Lei *et al.*, "Cosmic-ray heavy ions contributions to the atmospheric radiation field," in *European Conference on Radiation and Its Effects on Components and Systems*, pp. 375–376, September 2009.

[102] H. Li *et al.*, "Variation-aware, reliability-emphasized design and optimization of RRAM using SPICE model," in *Design, Automation Test in Europe Conference Exhibition (DATE)*, pp. 1425–1430, March 2015.

[103] Standford Nanoelectronics Lab [Online]. Available:. https://nano.stanford.edu/stanford-rram-model.

[104] ASU Emerging Devices and Circuits Group [Online]. Available:. http://faculty.engineering.asu.edu/shimengyu.

[105] P.-Y. Chen *et al.*, "Technology-design co-optimization of resistive cross-point array for accelerating learning algorithms on chip," in *Design, Automation Test in Europe Conference Exhibition (DATE)*, pp. 854–859, March 2015.

[106] L. Chang *et al.*, "Stable SRAM cell design for the 32 nm node and beyond," in *Proceedings of IEEE International Symposium on VLSI Technology*, pp. 128 – 129, 2005.

[107] K. Takeda *et al.*, "A read-static-noise-margin-free SRAM cell for low-vdd and high-speed applications," *IEEE Journal of Solid-State Circuits, Vol. 41, No. 1*, pp. 113 – 121, January 2006.

[108] I. Chang *et al.*, "A 32 kb 10T sub-threshold SRAM array with bit-interleaving and differential read scheme in 90 nm CMOS," *IEEE Journal of Solid-State Circuits, Vol. 44, No. 2,*, pp. 650 – 658, February 2009.

[109] Y. Kanno *et al.*, "Hierarchical Power Distribution With Power Tree in Dozens of Power Domains for 90-nm Low-Power Multi-CPU SoCs ," *IEEE Journal on Solid-State Circuits, Vol. 42, No. 1,*, pp. 528 – 535, January 2007.

[110] J. Ou *et al.*, "Performance investigation of sram cells based on gate-all-around (gaa) si nanowire transistor for ultra-low voltage applications," in *2012 IEEE International Conference on Solid-State and Integrated Circuit Technology*, pp. 1–3, Oct 2012.

[111] J. Y. Tsai and H. H. Hu, "Novel gate-all-around high-voltage thin-film transistor with t-shaped metal field plate design," *IEEE Transactions on Electron Devices, Vol.62, No.3*, pp. 882–887, March 2015.

[112] Mentor Graphics corp., "Eldo user Manual ver.11.2a," 2011.

[113] M. H. Abu-Rahma and M. Anis, *Nanometer Variation-Tolerant SRAM: Circuits and Statistical Design for Yield*. Springer, 2012.

[114] S.-S. Sheu *et al.*, "A ReRAM integrated 7T2R non-volatile SRAM for normally-off computing application," in *Proceedings of IEEE Asian Solid-State Circuits Conference (A-SSCC)*, pp. 245 – 248, November 2013.

[115] M. Khellah *et al.*, "Effect of power supply noise on SRAM dynamic stability," in *Proceedings of IEEE Symposium on VLSI Circuits*, pp. 76 – 77, 2007.

[116] S. Nalam *et al.*, "Dynamic write limited minimum operating voltage for nanoscale srams," in *Proceedings of Design Automation Test in Europe Conference Exhibition (DATE)*, pp. 1 – 6, 2011.

[117] F. Akopyan *et al.*, "TrueNorth: Design and Tool Flow of a 65 mW 1 Million Neuron Programmable Neurosynaptic Chip," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol. 34, No. 10*, pp. 1537–1557, October 2015.

[118] T. Li *et al.*, "Optimized deep belief networks on CUDA GPUs," in *International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, July 2015.

[119] M. Pasotti *et al.*, "An application specific embeddable flash memory system for non-volatile storage of code, data and bit-streams for embedded FPGA configurations," in *Symposium on VLSI circuits Digest of Technical Papers*, pp. 213 – 216, June 2003.

[120] J. Jex, "Flash memory BIOS for PC and notebook computers," in *IEEE Pacific Rim Conference on Communications, Computers and Signal Processing*, pp. 692 – 695, May 1991.

[121] M. A. A. Sanvido *et al.*, "NAND Flash Memory and Its Role in Storage Architectures," *Proceedings of IEEE , Vol. 96, No. 11*, pp. 1864 – 1874, December 2008.

[122] S. Schechter *et al.*, "Use ECP, not ECC, for Hard Failures in resistive memories," in *International Symposium on Computer architecture (ISCA)*, pp. 141 – 152, June 2010.

[123] D. Shinkel *et al.*, "A Double-Tail Latch-Type Voltage Sense Amplifier with 18ps Setup+Hold Time," in *IEEE International Solid-State Circuits Conference (ISSCC)*, pp. 314 – 315, February 2007.

[124] Synopsys corp., "HSPICE user Manual 2015.06," 2015.

[125] M. Mao *et al.*, "Programming strategies to improve energy efficiency and reliability of ReRAM Memory Systems," in *IEEE Workshop on Signal Processing Systems (SiPS)*, pp. 1 – 6, October 2015.

[126] A. Tosson *et al.*, "8T1R: A Novel Low-power High-speed RRAM-based Non-volatile SRAM Design," in *Great Lakes Symposium on VLSI (GLSVLSI)*, pp. 239 – 244, May 2016.

[127] I. Kazi *et al.*, "Energy/Reliability Trade-Offs in Low-Voltage ReRAM-Based Non-Volatile Flip-Flop Design," *IEEE Transactions on Circuits and Systems, Vol. 61, No. 11,*, pp. 3155 – 3164, November 2014.

[128] N. Wrachien *et al.*, "Investigation of Proton and X-Ray Irradiation Effects on Nanocrystal and Floating Gate Memory Cell Arrays," *IEEE Transactions on Nuclear Science, Vol. 55, No. 6*, pp. 3000–3008, December 2008.

[129] D. N. Nguyen *et al.*, "Radiation effects on advanced flash memories," *IEEE Transactions on Nuclear Science, Vol. 46, No. 6*, pp. 1744–1750, December 1999.

[130] S. Gerardin *et al.*, "Radiation Effects in Flash Memories," *IEEE Transactions on Nuclear Science, Vol. 60, No. 3*, pp. 1953–1969, June 2013.

[131] S. Gerardin and A. Paccagnella, "Present and Future Non-Volatile Memories for Space," *IEEE Transactions on Nuclear Science, Vol. 57, No. 6*, pp. 3016–3039, December 2010.

[132] NanGate FreePDK45 Open Cell Library. Accessed: 2017-03-17 [Online]. Available:. http://www.nangate.com/.

[133] NanGate 65nm Open Cell Library PDK ver1.3. Accessed: 2013-09-25 [Online]. Available:. http://www.nangate.com/.

[134] Predictive Technology Model (PTM). Accessed: 2017-03-17 [Online]. Available:. http://ptm.asu.edu.

[135] B. Wicht *et al.*, "Yield and speed optimization of a latch-type voltage sense amplifier," *IEEE Journal of Solid-State Circuits, Vol. 39, No. 7*, pp. 1148–1158, July 2004.

[136] A. Tosson *et al.*, "RRAM Refresh Circuit: A Proposed Solution To Resolve The Soft-Error Failures For HfO2/Hf 1T1R RRAM Memory Cell," in *Great Lakes Symposium on VLSI (GLSVLSI)*, pp. 227 – 232, May 2016.

[137] M. McCartney, *SRAM reliability improvement using ECC and circuit techniques*. PhD thesis, Carnegie Mellon University, Electrical and Computer Engineering, 2014.

[138] S. F. Liu *et al.*, "Efficient Majority Logic Fault Detection With Difference-Set Codes for Memory Applications, Vol. 20, No. 1," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, pp. 148–156, January 2012.

[139] H. B. Wang *et al.*, "Evaluation of SEU Performance of 28-nm FDSOI Flip-Flop Designs," *IEEE Transactions on Nuclear Science, Vol. 64, No. 1*, pp. 367–373, January 2017.

[140] W. Jing *et al.*, "Practice on layout-level radiation hardened technologies for I/O cells," in *International Conference on Information, Networking and Automation (ICINA)*, pp. V1–365–V1–369, October 2010.

[141] Y. Piccin *et al.*, "Radiation-Hardening Technique for Voltage Reference Circuit in a Standard 130 nm CMOS Technology," *IEEE Transactions on Nuclear Science, Vol. 61, No. 2*, pp. 967–974, April 2014.

[142] P. U. Diehl and M. Cook, "Unsupervised learning of digit recognition using spike-timing-dependent plasticity," *Frontiers of Computational Neuroscience, Vol. 9, No. 99*, August 2015.

[143] A. Toffoli *et al.*, "Phase Change Memory advanced electrical characterization for conventional and alternative applications," in *IEEE International Conference on Microelectronic Test Structures (ICMTS)*, pp. 114 – 118, March 2012.

[144] X. Dong *et al.*, "Circuit and Microarchitecture Evaluation of 3D Stacking Magnetic RAM (MRAM) as a Universal Memory Replacement," in *Design Automation Conference (DAC)*, pp. 554 – 559, June 2008.

[145] L. Zhao *et al.*, "Multi-level control of conductive nano-filament evolution in HfO2 ReRAM by pulse-train operations," *Nanoscale, Vol. 6, No. 11*, pp. 5698–5702, 2014.

[146] J. C. Liu *et al.*, "Categorization of Multilevel-Cell Storage-Class Memory: An RRAM Example," *IEEE Transactions on Electron Devices, Vol. 62, No. 8*, pp. 2510–2516, August 2015.

[147] S. R. Lee *et al.*, "Multi-level switching of triple-layered TaOx RRAM with excellent reliability for storage class memory," in *Symposium on VLSI Technology (VLSIT)*, pp. 71–72, June 2012.

[148] Z. Wei *et al.*, "Demonstration of high-density ReRAM ensuring 10-year retention at 85 C based on a newly developed reliability model," in *International Electron Devices Meeting*, pp. 31.4.1–31.4.4, December 2011.

[149] S. K. Pal and S. Mitra, "Multilayer perceptron, fuzzy sets, and classification," *IEEE Transactions on Neural Networks, Vol. 3, No. 5*, pp. 683–697, September 1992.

[150] K. L. Goh *et al.*, “Multilayer perceptron neural network classification for human vertical ground reaction forces,” in *IEEE Conference on Biomedical Engineering and Sciences (IECBES)*, pp. 536–540, December 2014.

[151] A. K. Jain *et al.*, “Statistical pattern recognition: a review,” *IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 22, No. 1*, pp. 4–37, January 2000.

[152] X. Zhai *et al.*, “MLP Neural Network Based Gas Classification System on Zynq SoC, Vol. 4,” *IEEE Access*, vol. 4, pp. 8138–8146, 2016.

[153] M. Stimberg *et al.*, “Equation-oriented specification of neural models for simulations,” *Frontiers in Neuroinformatics*, vol. 8, p. 6, 2014.

[154] D. Querlioz *et al.*, “Immunity to Device Variations in a Spiking Neural Network With Memristive Nanodevices,” *IEEE Transactions on Nanotechnology, Vol. 12, No. 3*, pp. 288–295, May 2013.

[155] G. Bi and M. Poo, “Synaptic Modifications in Cultured Hippocampal Neurons: Dependence on Spike Timing, Synaptic Strength, and Postsynaptic Cell Type,” *The Journal of Neuroscience, Vol. 18, No. 24*, pp. 10464 –10472, December 1998.

[156] J. P. Abrahamsen *et al.*, “A time domain winner-take-all network of integrate-and-fire neurons,” in *IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. V–361–V–364, May 2004.

[157] B. Chen *et al.*, “Multi-level resistive switching characteristics correlated with microscopic filament geometry in TMO-RRAM,” in *International Symposium on VLSI Technology, Systems, and Applications (VLSI-TSA)*, pp. 1–2, April 2013.

[158] D. Ielmini, “Modeling the Universal Set/Reset Characteristics of Bipolar RRAM by Field- and Temperature-Driven Filament Growth,” *IEEE Transactions on Electron Devices, Vol. 58, No. 12*, pp. 4309–4317, December 2011.

[159] A. L. Hogkin and A. F. Huxley, “Action potentials recorded from inside a nerve fibre,” *Nature, Vol. 144*, pp. 710–711, October 1939.

[160] C. Liu *et al.*, “A spiking neuromorphic design with resistive crossbar,” in *ACM/EDAC/IEEE Design Automation Conference (DAC)*, pp. 1–6, June 2015.

[161] F. Alibart *et al.*, “High precision tuning of state for memristive devices by adaptable variation tolerant algorithm,” *Nanotechnology, Vol. 23, No. 7*, January 2012.

[162] P.-Y. Chen *et al.*, "Programming Protocol Optimization for Analog Weight Tuning in Resistive Memories," *IEEE Electron Device Letters, Vol. 36, No. 11*, pp. 1157–1159, November 2015.

[163] S. Yu *et al.*, "Investigating the switching dynamics and multilevel capability of bipolar metal oxide resistive switching memory," *Applied Physics Letters, Vol. 98, No. 10*, p. 103514, September 2011.

[164] M. Al-Shyoukh and A. Kalnitsky, "A 500nA quiescent current, trim-free, +/- 1.75N-channel MOSFETs," in *Proceedings of the IEEE 2014 Custom Integrated Circuits Conference*, pp. 1–4, September 2014.

[165] H. Watanabe *et al.*, "CMOS voltage reference based on gate work function differences in poly-Si controlled by conductivity type and impurity concentration," *IEEE Journal of Solid-State Circuits, Vol. 38, No. 6*, pp. 987–994, June 2003.

[166] A. Tosson *et al.*, "Mitigating the Effect of the Reliability Soft-errors of the RRAM Devices on the Performance of the RRAM-based Neuromorphic Systems," in *International Great Lakes Symposium on VLSI (GLSVLSI)*, pp. 1–6, May 2017.

[167] G. Patounakis *et al.*, "A Fully Integrated OnChip DC-DC Conversion and Power Management System," *IEEE Journal of Solid-State Circuits, Vol. 39, No. 3*, pp. 987–994, March 2004.

[168] R. J. Milliken *et al.*, "Full On-Chip CMOS Low-Dropout Voltage Regulator," *IEEE Transactions on Circuits and systems -I: Regular papers, Vol. 54, No. 9*, pp. 1879–1891, September 2007.

[169] J. F. Bulzacchelli *et al.*, "Dual-loop System of Distributed Microregulators with High DC Accuracy, Load Response Time Below 500 ps, and 85mV Dropout voltage," *IEEE Journal of Solid-State Circuits, Vol. 47, No. 4*, pp. 863–874, April 2012.

[170] P. L. Callet *et al.*, "A Convolutional Neural Network Approach for Objective Video Quality Assessment," *IEEE Transactions on Neural Networks, Vol. 17, No. 5*, pp. 1316–1327, September 2006.

[171] X. W. Chen and X. Lin, "Big Data Deep Learning: Challenges and Perspectives," *IEEE Access, Vol. 2*, pp. 514–525, 2014.

[172] A. Tosson *et al.*, "Analysis of RRAM Reliability Soft-Errors on the Performance of RRAM-based Neuromorphic Systems," in *IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, pp. 1–6, July 2017.

[173] TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems [Online]. Available:. https://www.tensorflow.org/.

[174] Y. Jia *et al.*, "Caffe: Convolutional Architecture for Fast Feature Embedding," in *Proceedings of ACM International Conference on Multimedia*, pp. 675–678, 2014.

[175] D. T. Tran *et al.*, "Fast DNN training based on auxiliary function technique," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2160–2164, April 2015.

[176] J. T. Chien and P. W. Huang, "Variance reduction for optimization in speech recognition," in *IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, pp. 1–6, September 2016.

APPENDICES

Appendix A

Brian code for the RRAM-based neuromorphic system

```
import numpy as np
import matplotlib.cm as cmap
import time
import os.path
import scipy
import cPickle as pickle
import brian_no_units
import brian as b
from struct import unpack
from brian import *

# specify the location of the MNIST data
MNIST_data_path = ''

#----------------------------------
# functions
#----------------------------------
def get_labeled_data(picklename, bTrain = True):
"""Read input-vector (image) and
target class (label, 0-9) and return
it as list of tuples.
"""
```

```
if os.path.isfile('%s.pickles' % picklename):
data = pickle.load(open('%s.pickles' % picklename))
else:
# Open the images with gzip in read binary mode
if bTrain:
images = open(MNIST_data_path
+ 'train-images.idx3-ubyte','rb')
labels = open(MNIST_data_path
+ 'train-labels.idx1-ubyte','rb')
else:
images = open(MNIST_data_path
+ 't10k-images.idx3-ubyte','rb')
labels = open(MNIST_data_path
+ 't10k-labels.idx1-ubyte','rb')
# Get metadata for images
images.read(4)  # skip the magic_number
number_of_images = unpack('>I', images.read(4))[0]
rows = unpack('>I', images.read(4))[0]
cols = unpack('>I', images.read(4))[0]
# Get metadata for labels
labels.read(4)  # skip the magic_number
N = unpack('>I', labels.read(4))[0]

if number_of_images != N:
raise Exception('number of
labels did not match the number of images')
# Get the data
x = np.zeros((N, rows, cols),
dtype=np.uint8)  # Initialize numpy array
y = np.zeros((N, 1),
dtype=np.uint8)  # Initialize numpy array
for i in xrange(N):
if i % 1000 == 0:
print("i: %i" % i)
x[i] = [[unpack('>B', images.read(1))[0]
for unused_col in xrange(cols)]
for unused_row in xrange(rows) ]
y[i] = unpack('>B', labels.read(1))[0]
```

```
data = {'x': x, 'y': y, 'rows': rows, 'cols': cols}
pickle.dump(data, open("%s.pickle" % picklename, "wb"))
return data

def get_matrix_from_file(fileName):
offset = len(ending) + 4
if fileName[-4-offset] == 'X':
n_src = n_input
else:
if fileName[-3-offset]=='e':
n_src = n_e
else:
n_src = n_i
if fileName[-1-offset]=='e':
n_tgt = n_e
else:
n_tgt = n_i
readout = np.load(fileName)
print readout.shape, fileName
value_arr = np.zeros((n_src, n_tgt))
if not readout.shape == (0,):
value_arr[np.int32(readout[:,0]),
np.int32(readout[:,1])] = readout[:,2]
return value_arr


def save_connections(ending = ''):
print 'save connections'
for connName in save_conns:
connMatrix = connections[connName][:]
connListSparse = ([(i,j,connMatrix[i,j])
for i in xrange(connMatrix.shape[0])
for j in xrange(connMatrix.shape[1]) ])
np.save(data_path +
'weights_real_mod_10000_f/' + connName + ending, connListSparse)

def save_theta(ending = ''):
```

```
print 'save theta'
for pop_name in population_names:
np.save(data_path + 'weights_real_mod_10000_f/theta_'
+ pop_name + ending, neuron_groups[pop_name + 'e'].theta)

def normalize_weights():
for connName in connections:
if connName[1] == 'e' and connName[3] == 'e':
connection = connections[connName][:]
temp_conn = np.copy(connection)
colSums = np.sum(temp_conn, axis = 0)
colFactors = weight['ee_input']/colSums
for j in xrange(n_e):#
connection[:,j] *= colFactors[j]

def get_2d_input_weights():
name = 'XeAe'
weight_matrix = np.zeros((n_input, n_e))
n_e_sqrt = int(np.sqrt(n_e))
n_in_sqrt = int(np.sqrt(n_input))
num_values_col = n_e_sqrt*n_in_sqrt
num_values_row = num_values_col
rearranged_weights = np.zeros((num_values_col, num_values_row))
connMatrix = connections[name][:]
weight_matrix = np.copy(connMatrix)

for i in xrange(n_e_sqrt):
for j in xrange(n_e_sqrt):
rearranged_weights[
i*n_in_sqrt : (i+1)*n_in_sqrt, j*n_in_sqrt : (j+1)*n_in_sqrt] = \
weight_matrix[:, i + j*n_e_sqrt].reshape((n_in_sqrt, n_in_sqrt))
return rearranged_weights



#--------------------------------------
# save results
#--------------------------------------
```

```
print 'save results'
if not test_mode:
save_theta()
if not test_mode:
save_connections()
else:
np.save(data_path + 'activity/resultPopVecs'
+ str(num_examples), result_monitor)
np.save(data_path + 'activity/inputNumbers'
+ str(num_examples), input_numbers)
```

Appendix B

List of Publications

Here are the list of published papers from this work:

- A.Tosson and A.Neale and M.Anis and Lan Wei, "8T1R: A Novel Low-power High-speed RRAM-based Non-volatile SRAM Design," in Great Lakes Symposium on VLSI (GLSVLSI), pp. 239 - 244, May. 2016.
- A.Tosson and M.Anis and Lan Wei, "RRAM Refresh Circuit: A Proposed Solution To Resolve The Soft-Error Failures For HfO2/Hf 1T1R RRAM Memory Cell," in Great Lakes Symposium on VLSI (GLSVLSI), pp. 227 - 232, May. 2016.
- A.Tosson and S.Yu and M.Anis and Lan Wei, "Mitigating the effect of the reliability soft-errors of the RRAM devices on the performance of the RRAM-based neuromorphic systems," in Great Lakes Symposium on VLSI (GLSVLSI), pp. 1-6, May 2017
- A.Tosson and S.Yu and M.Anis and Lan Wei, "Analysis of RRAM Reliability Soft-Errors on the Performance of RRAM-based Neuromorphic Systems," in IEEE Computer Society Annual Symposium on VLSI (ISVLSI), pp.1-6, July 2017
- A.Tosson and S.Yu and M.Anis and Lan Wei, "1T2R: A Novel Memory Cell Design to Resolve Single-Event Upset in RRAM Arrays," in IEEE International Conference on ASIC (ASICON), pp. 1-4, Oct. 2017
- A.Tosson and S.Yu and M.Anis and Lan Wei, "A Study of the Effect of RRAM Reliability Soft Errors on the Performance of RRAM-Based Neuromorphic Systems,"

in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, pp. 1-13, Aug.2017

- A.Tosson and S.Yu and M.Anis and Lan Wei, ”Proposing a solution for Single-Event Upset In 1T1R RRAM Memory Arrays”, accepted for IEEE Transactions on Nuclear Science (TNS).