University of California
Santa Barbara

Towards Data Reliable, Low-Power, and Repairable Resistive Random Access Memories

A dissertation submitted in partial satisfaction
of the requirements for the degree

Doctor of Philosophy
in
Electrical and Computer Engineering

by

Amirali Ghofrani

Committee in charge:

Professor Kwang-Ting Cheng, Chair
Professor Dmitri Strukov
Professor Luke Theogarajan
Professor Malgorzata Marek-Sadowska

June 2016

The Dissertation of Amirali Ghofrani is approved.

Professor Dmitri Strukov

Professor Luke Theogarajan

Professor Malgorzata Marek-Sadowska

Professor Kwang-Ting Cheng, Committee Chair

March 2016

Towards Data Reliable, Low-Power, and Repairable

Resistive Random Access Memories

by

Amirali Ghofrani

To my loving parents, Saeed and Farideh, with their unconditional and never-ending love and support. And to my sister and brother, Mishka and Reza, that are the best siblings that one can ever ask for.

Acknowledgements

It was quite a journey for me to finish my PhD, and there are many people that I would like to acknowledge.

I am deeply indebted to my research advisor Prof. Kwang-Ting Cheng for his support, patience, and understanding throughout my PhD study. He showed what it takes to be a great advisor. Under his guidance, I became a better researcher and a better writer. He looks after his students like a real father, and puts lots of time and effort in making his students independent thinkers. Thank you Tim, you are awesome!

I would also like to thank the rest of my dissertation committee, Prof. Dmitri Strukov, Prof. Luke Theogarajan, and Prof. Malgorzata Marek-Sadowska for their insightful comments that helped me improve the quality of this work. Specially, Dmitri always encouraged me and his door was always open for discussions.

I should thank Prof. Wei Lu, Prof. Valeria Bertacco, Prof. Luca Benini, Prof. Rajesh Gupta, Dr. Siddharth Gaba, Dr. Bhaswar Chakrabarti, Dr. Farnood Merrikh-Bayat, Advait Madhavan, Justin rofeh, Dawen Xu, and Ritesh Parikh, for the fruitful collaborations that we had.

I thank my current and past fellow labmates in SoC Design and Test Lab at the University of California, Santa Barbara: Dr. Saeed Shamshiri, Dr. Peter Lisherness, Dr. Ming Gao, Dr. Hsiu-Ming (Sherman) Chang, Dr. Yan Zheng, Dr. Xin Yang, Dr. Yi-chu Wang, Miguel Lastras, Nicole Fern, Chun-Kai Hsu, Fan (Fred) Lin, Rui Wu, Yuyang Wang, and Lei-lai Shao. It was a pleasure working with you all.

And of course, research is not possible with empty pockets! I would like to thank AFOSR-MURI and GSRC for funding most of the research, the ECE department for giving me TA positions, and the OISS for their generous work-study awards.

I was truly blessed to have many wonderful people around me during the many years of my PhD studies. Without their presence, this work was either not possible, or would have

been a painful experience! I should thank: Miguel Lastras, for being my brother throughout all these years of PhD. Melika Payvand, for being a great friend, caring for me in good days and bad days alike. Zhinous, Seyyed, Samaneh, Ehsan, Omid, Behzad, Sayyad, and Mohammad, a.k.a "Khodemoon", for being a truly wonderful group of friends that I am blessed to have. Specially, Zhinous Marzi, for all the moral support and the brainstorming in the last months of this dissertation on top of her priceless friendship. Dr. Borzooyeh Naji, for being my best friend for $\approx$30 years. Dr. Abbas Rahimi, for his support with open arms when I needed it the most. I will never forget that. Hedieh Ekhlasi, for her invaluable care and support. Nahid, Kambiz, and Shireen Garrousi, for being my family in USA. Dr. Saeed Shamshiri for mentoring me when I first came here. And Dr. Shahab Karimi, Majid Namaki, Nima Tayyebi, Hossein Mazloumi, Dr. Elaheh Ahmadi, Dr. Ali Nabi, Parviz Alvandi, Avantika Sodhi, Kourosh and Soheila Mohsenzadegan, Saeed and Razieh Hashemi, Hadi and Elham Rasouli, and Fatemeh and Mehran Rajaeian, for touching my life in different ways, with their kindness and friendship. And thanks to my other friends in Iranian Graduate Student Association (IGSA), Persian Student Group (PSG), Heroes indoor soccer team, and the camping group!

And last but not least, I would like to express my sincerest appreciations to my parents, Farideh Khoshghalb and Saeed Ghofrani, and my siblings, Mishka and Reza. I am so happy to have you in my life.

Curriculum Vitæ

Amirali Ghofrani

Education

2016	Ph.D. in Computer Engineering, University of California, Santa Barbara.
2013	M.Sc. in Computer Engineering, University of California, Santa Barbara.
2010	M.Sc. in Computer Engineering, University of Tehran, Iran.
2007	B.Sc. in Computer Engineering, University of Tehran, Iran.

Publications

1. **A. Ghofrani**, M. A. Lastras-Montaño, Y. Wang, K.-T. Cheng, "In-place Repair for Resistive Memories Utilizing Complimentary Resistive Switches", submitted to International Symposium on Low Power Electronics and Design, San Francisco, USA, August 2016.
2. M. A. Lastras-Montaño, **A. Ghofrani**, K.-T. Cheng, "A Low-Power Hybrid Reconfigurable Architecture for Resistive Random-Access Memories", accepted in *22nd IEEE Symposium on High Performance Computer Architecture (HPCA)*, Barcelona, Spain, March 2016.
3. **A.Ghofrani**, A. Rahimi, M. A. Lastras-Montaño, L. Benini, R. K. Gupta, K.-T. Cheng, "Associative Memristive Memory for Approximate Computing in GPUs", accepted in *IEEE Journal on Emerging and Selected Topics in Circuits and Systems (JETCAS)*.
4. **A. Ghofrani**, M. A. Lastras-Montaño, S. Gaba, M. Payvand, W. Lu, L. Theogarajan, K.-T. Cheng, "A Low-Power Variation-Aware Adaptive Write Scheme for Access-Transistor-Free Memristive Memory", *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, Vol. 12, Issue 1, July 2015.
5. M. A. Lastras-Montaño, **A. Ghofrani**, K-T. Cheng, "Architecting Energy Efficient Crossbar-based Memristive Random Access Memories", *Intl. Symposium on Nanoscale Architectures (NANOARCH15)*, Boston, USA, July 2015.
6. J. Rofeh, A. Sodhi, M. Payvand, M. A. Lastras-Montaño, **A. Ghofrani**, A. Madhavan, S. Yemenicioglu, K.-T. Cheng, L. Theogarajan, "Vertical Integration of Memristors onto Foundry CMOS Dies using Wafer-Scale Integration", *IEEE Electronic Components and Technology Conference (ECTC15)*, San Diego, USA, May 2015.
7. M. Payvand, A. Madhavan, M. A. Lastras-Montaño, **A. Ghofrani**, J. Rofeh, K.-T. Cheng, D. Strukov, L. Theogarajan, "A Configurable CMOS Memory Platform for 3D Integrated Memristors", *IEEE International Symposium on Circuits and Systems (ISCAS15)*, Lisbon, Portugal, May 2015.
8. A. Rahimi, **A. Ghofrani**, K.-T. Cheng, L. Benini, R. Gupta, "Approximate Associative Memristive Memory for Energy-Efficient GPUs", *Design, Automation, and Test in Europe (DATE15)*, Grenoble, France, March 2015.

9. M. A. Lastras-Montaño, **A. Ghofrani**, K.-T. Cheng, "HReRAM, A Hybrid Reconfigurable Resistive Random Access Memory", *Design, Automation, and Test in Europe (DATE15)*, Grenoble, France, March 2015.
10. **A. Ghofrani**, M. A. Lastras-Montaño, K-T. Cheng, "Toward Large-Scale Access-Transistor-Free Memristive Crossbars", invited to *Asia South Pacific Design Automation Conference (ASPDAC15)*, Tokyo, Japan, Jan 2015.
11. A. Rahimi, **A. Ghofrani**, M. A. Lastras-Montaño, K.-T. Cheng, L. Benini, R. Gupta , "Energy-Efficient GPGPU Architectures via Collaborative Compilation and Memristive Memory-Based Computing", *51st Design Automation Conference (DAC14)*, San Francisco, USA, June 2014.
12. **A. Ghofrani**, M. A. Lastras-Montaño, K-T. Cheng, "Towards Data Reliable Crossbar-Based Memristive Memories," *IEEE International Test Conference (ITC13)*, Anaheim, USA, Sep 2013.
13. M. A. Lastras-Montaño, **A. Ghofrani**, K.-T. Cheng, "Architecting Low Power Crossbar-Based Memristive RAM", *Non-Volatile Memory Workshop (NVMW13)*, San Diego, USA, Mar 2013.
14. D. Xu, H. Li, **A. Ghofrani**, K.-T. Cheng, Y. Han, X. Li, "Test Quality Optimization for Variable n-Detection of Transition Faults", *IEEE Transaction on Very Large Scale Integration systems*, Vol. PP, Issue 99, Jul 2013.
15. **A. Ghofrani**, R. Parikh, S. Shamshiri, A. DeOrio, K-T. Cheng, V. Bertacco, "Comprehensive Online Defect Diagnosis in On-Chip Networks", *30th VLSI Test Symposium (VTS12)*, Maui, USA, Apr 2012.
16. Y. Zheng, P. Lisherness, S. Shamshiri, **A. Ghofrani**, S. Yang, K.-T. Cheng, "Post-Fabrication Reconfiguration for Power-Optimized Tuning of Optically Connected Multi-Core Systems", *17th Asia and South Pacific Design Automation Conference (ASPDAC12)*, Australia, Jan 2012.
17. S. Shamshiri, **A.Ghofrani**, K.T. Cheng, "End to End Error Correction and Online Diagnosis for On-Chip Networks", *IEEE International Test Conference (ITC11)*, Anaheim, USA, Sep 2011.

Awards

2015 UCSB Electrical and Computer Engineering Department Dissertation Fellowship (Spring)

2013 Gerald W. Gordon Award, IEEE Philadelphia Section, Test Technology Technical Council (TTTC), and International Test Conference (ITC)

2013 GSA Dixon-Levy Service Award, UCSB

2012-2013 Leslie Griffin Lawson Award for Outstanding Leadership, UCSB

2012-2013 Graduate Individual Co-Curricular Leadership and Activities Award, UCSB

2011 Best Student Paper Award, International Test Conference (ITC)

Abstract

Towards Data Reliable, Low-Power, and Repairable
Resistive Random Access Memories

by

Amirali Ghofrani

A series of breakthroughs in memristive devices have demonstrated the potential of memristor arrays to serve as next generation resistive random access memories (ReRAM), which are fast, low-power, ultra-dense, and non-volatile. However, memristors' unique device characteristics also make them prone to several sources of error. Owing to the stochastic filamentary nature of memristive devices, various recoverable errors can affect the data reliability of a ReRAM. Permanent device failures further limit the lifetime of a ReRAM. This dissertation developed low-power solutions for more reliable and longer-enduring ReRAM systems.

In this thesis, we first look into a data reliability issue known as write disturbance. Writing into a memristor in a crossbar could disturb the stored values in other memristors that are on the same memory line as the target cell. Such disturbance is accumulative over time which may lead to complete data corruption. To address this problem, we propose the use of two regular memristors on each word to keep track of the disturbance accumulation and trigger a refresh to restore the weakened data, once it becomes necessary.

We also investigate the considerable variation in the write-time characteristics of individual memristors. With such variation, conventional fixed-pulse write schemes not only waste significant energy, but also cannot guarantee reliable completion of the write operations. We address such variation by proposing an adaptive write scheme that adjusts the width of the write pulses for each memristor. Our scheme embeds an online monitor to detect the completion of a write operation and takes into account the parasitic effect of line-shared devices in access-

transistor-free memristive arrays. We further investigate the use of this method to shorten the test time of memory march algorithms by eliminating the need of a verifying read right after a write, which is commonly employed in the test sequences of march algorithms.

Finally, we propose a novel mechanism to extend the lifetime of a ReRAM by protecting it against hard errors through the exploitation of a unique feature of bipolar memristive devices. Our solution proposes an unorthodox use of *complementary resistive switches* (a particular implementation of memristive devices) to provide an "in-place spare" for each memory cell at negligible extra cost. The in-place spares are then utilized by a repair scheme to repair memristive devices that have failed at a stuck-at-ON state at a page-level granularity. Furthermore, we explore the use of in-place spares in lieu of other memory reliability and yield enhancement solutions, such as error correction codes (ECC) and spare rows. We demonstrate that with the in-place spares, we can yield the same lifetime as a baseline ReRAM with either significantly fewer spare rows or a lighter-weight ECC, both of which can save on energy consumption and area.

Contents

Chapter 1

Introduction

1.1 Motivation

CMOS-based memory technologies cannot keep up with the ever-increasing demand for denser and lower-power memories. As the memory cell size is mainly limited by the size of its access-transistor, CMOS technology scaling is reaching its limit due to the increased leakage current of the access-transistors and the yield drop induced by fabrication imprecision [1].

As an alternative, emerging resistive memory technologies such as phase change memories (PCM) [2], spin-transfer torque magneto-resistive memories (STT-MRAM) [3], and metal oxide valence change resistive random access memories (ReRAM) [4] have been investigated recently that offer ultra-small and low-power memory elements with fast switching speeds. Among them, metal oxide valence change ReRAMs, generally referred to as memristors [5], are especially promising as they exhibit unique electrical characteristics which enable the elimination of the access-transistor, while maintaining the same power/speed/endurance advantages [6].

A memristor is a two-terminal passive programmable resistor, which typically has a metal/insulator/metal (MIM) structure shown in Figure 1.1a. The resistance of a memristor is maintained in the absence of an electric field. 0/1 logic values can be represented by

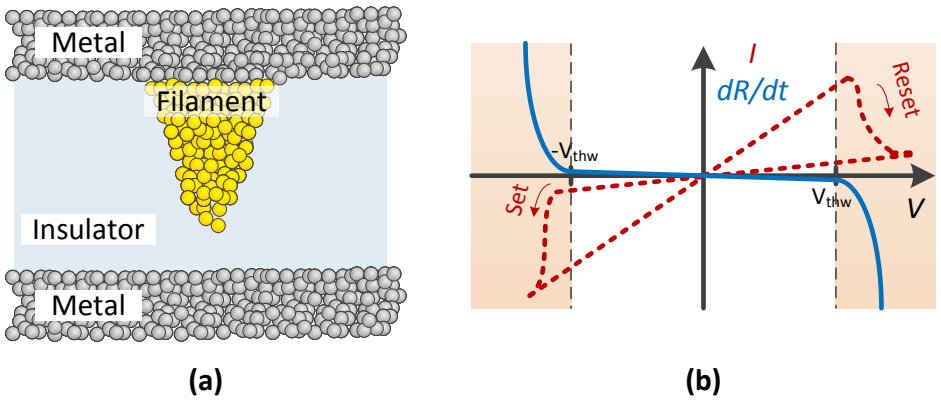


Figure 1.1: A memristor's exemplar realization, and its non-linear dR/dt-V characteristic. a) A typical metal/inuslator/metal structure. Formation of a conductive filament inside the insulator layer changes the resistance of the device. b) Electrical characteristics of memristive devices: the solid line shows the non-linearity in the rate of change for the resistance of the device based on the applied voltage. The dashed lines shows the linear I-V characteristic observed in a typical memristive device.

ranges of high/low resistances. The resistance of the device can be changed by applying adequate voltage (or current) pulses. The change in the resistance happens due to the non-volatile formation of a conductive filament inside the insulating oxide layer and has a strong non-linear dependency on the amplitude of the applied pulse [7], as shown in Figure 1.1b: while applying voltages above a write threshold, V_{thw}, effectively switches the internal state of the device, applying voltages below V_{thw} has negligible effect on the device's state. This non-linearity combined with proper voltage application schemes could effectively provide the functionality of an access-transistor, and thus, obviate the need for an access-transistor for each memory cell.

The elimination of the access-transistor and the simple structure of memristors facilitates the realization of ultra-high density access-transistor-free (ATF) memory arrays with sub 10-nm feature sizes [8]. Such arrays demonstrate lower power consumption than existing technologies, verified by analysis and preliminary experimental measurements, due to the ATF memory structure and the passiveness of the devices [9, 10]. Such characteristics make memristive memories attractive as an extremely dense and low-power non-volatile memory [6]. Several nanoscale memristive crossbars have been successfully demonstrated recently [11, 12, 13].

However, the elimination of the access transistor as well as the intrinsic characteristics of the memristive devices, also introduce several challenges. Such challenges should be addressed to enable the use of memristive devices as the next generation memory technology. Data reliability issues arise from the elimination of the access-transistors, where the logic value of non-selected devices might get affected during a write operation due to what is known as the *write disturbance* problem. Furthermore, successful write operations cannot be guaranteed for memristive devices, as memristors have a significant *write time variation*. Finally, memristive devices are prone to device failures, necessitating low-cost repair schemes to replace the failed devices in order to improve the lifetime of memristive memories. In this thesis, we focus on finding innovative and low-cost solutions to address these problems.

1.2 Addressing Write Disturbance

Write disturbance is an undesired coupling effect during write operation that affects several other memristors that share the same word- and/or bit-line [14, 11]. Due to the write disturbance, writing a logic value into one memristor may disturb the resistance of the line-shared memristors that store the opposite logic value. This effect is due to the access-transistor-free structure of the memory array, in which writing into a target cell also applies a notable partial voltage (*e.g.* , $V_w/2$) across line-shared devices. The resistance degradation due to the write disturbance could accumulate over several write cycles and may eventually result in corruption or complete inversion of the stored logic value.

Here we propose a solution [15] to address this problem. Our solution confines the write disturbance effect to word-line-shared devices by applying asymmetric voltages to the word-line and the bit-line (*e.g.* , $-V_w/3$ on target bit-line and $2V_w/3$ on target word-line): bit-line-shared devices will experience less partial bias (*e.g.* , $V_w/3$), which has a very negligible write-disturbance effect. Then, two regular devices on each word-line are assigned as *canary cells*,

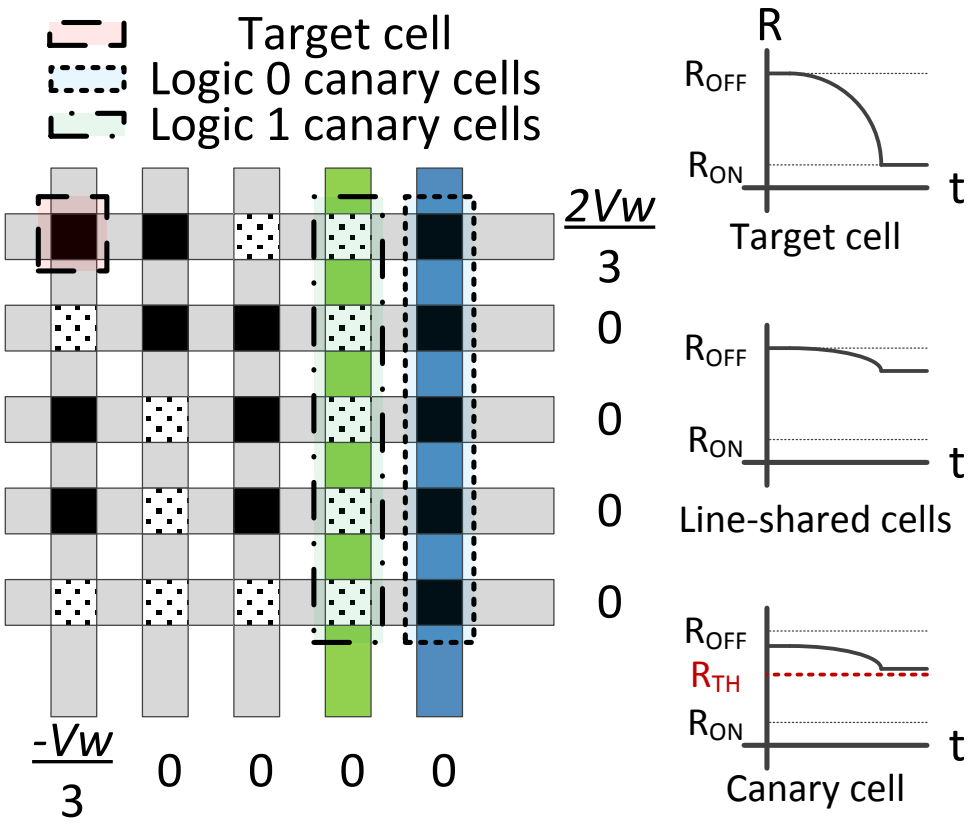


Figure 1.2: Write disturbance effect and solution. Applying V_w across the target cell effectively switches the target device to logic 1 (top-right), while degrading the logic 0 stored in other line-shared devices (middle-right). Two regular memristors per word-line, storing logic 1 and 0 respectively and placed on green and blue bit-lines, keep track of the worst-case logic 1 and 0 write disturbance effect on the line (bottom-right), and trigger a refresh operation when the degraded resistance reaches a close-to-corruption threshold, R_{TH}.

in which undisturbed logics 1 and 0 are stored initially. The canary cells cannot be accessed through the standard write interface and are meant to keep track of the worst-case, cumulative write disturbance effect for their corresponding logic on each word-line: while they are affected by the write disturbance effect similar to other memristors on the same word-line, they cannot be restored to the strong logic values via the standard write operations.

During a write operation on word-line W, the resistance values of W's two canary cells are monitored to avoid data corruption: As canary cells experience the worst disturbance accumulation of all devices on W (explained in the last paragraph), therefore, as long as the

resistance value of each canary cell is in its valid range, the validity of the data stored on other devices on W can be guaranteed. Whenever the resistance value of a canary cell reaches a known close-to-corruption threshold, R_{TH}, a refresh operation is triggered that refreshes all memristive devices on W. Fig. 1.2 illustrates write disturbance effect as well as the proposed solution. Chapter 2 discusses the write disturbance problem and the proposed solution in more details.

1.3 Addressing Write Time Variation

Another intrinsic issue with the memristive devices is the significant variation in their write time characteristics. The length of the required write pulse to switch the state of a device, varies from device to device, and even from cycle to cycle for the same device, and is not known beforehand. Hence, a method is required to find the proper length of a write pulse for each device/cycle to ensure a successful write operation.

Adaptive methods are commonly utilized for this purpose. In an adaptive write operation, the length of the write pulse is adjusted for individual devices by terminating the write pulse as soon as the target device completes the switching. This is typically done by monitoring (*i.e.*, sensing) the write-current through the target cell, I_{target}, to detect a sudden jump in the current that indicates the completion of switching.

However, the existence of partially-selected devices in an ATF crossbar renders the conventional adaptive schemes useless: During the write operation, line-shared devices are partially-biased. Such a partial voltage introduces a *leakage current* on the target bit-line, as shown in Fig. 1.3. The leakage current is also data-dependent: the larger the number of ON devices on the bit-line, the greater the leakage current. Hence, in ATF crossbars, I_{target} is mounted on top of a considerable data-dependent leakage current I_{leak}, due to the bit-line-shared devices. A typical sensing circuitry cannot detect the switching in such noisy conditions.

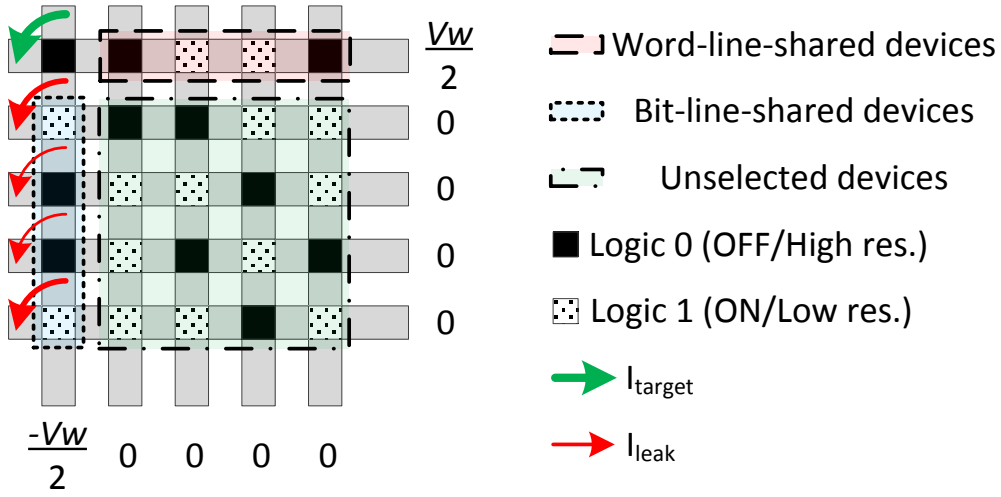


Figure 1.3: $V_w/2$ voltage application scheme for a write operation. Black (dotted white) cells represent stored logic 0 (1). Highlighted areas show the word- and bit-line-shared devices and the unselected devices respectively. A leakage current inversely proportional to the resistance of the bit-line-shared devices leaks into the bit-line during the write operation.

To address this problem, we propose a leakage-current-filtering mechanism. In this method, each adaptive write operation consists of two stages. In the first stage, the data-dependent leakage current of the bit-line-shared devices is latched. The latched I_{leak} is then subtracted from the total current observed on the bit-line in the second stage, to obtain the write-current contributed only by the target cell, *i.e.* , I_{target}. This filtered current is then sensed by a typical sensing circuit to detect the switching event. As there exists significant temporal variation in memristive devices for a complete switching of the write operation [16], this method yields a considerable energy saving in ATF memristive crossbars by enabling adaptive write operation in such crossbars. Chapter 3 provides more details on the write-time variation problem and the proposed solution.

1.4 Extending ReRAM Lifetime

Owing to their stochastic filamentary nature, memristive devices are prone to device failures [17]. Physical defects and endurance problems could lead to permanent "hard errors" that can adversely affect the lifetime of memristive memory modules. This necessitates low-cost repair schemes to replace the failed devices in order to improve the lifetime of memristive memories. Several solutions exist for conventional memory technologies to address hard errors, such as error correction codes (ECC) [18] and spare row/columns [19]. However, such solutions impose significant area and energy overheads on the memory module. When applied to memristive memories, such overhead becomes even more noticeable, considering the highly dense and ultra-low-power characteristics of memristive memories.

In chapter 4, we propose a zero-area-overhead in-place spare for each bit to repair the failed devices [20]. The proposed low-cost memory repair scheme is inspired by the possibility of stacking two memristive devices at the footprint of a single device at negligible extra cost, shown in *complementary resistive switches* [21]. The proposed method requires only minor modifications to the memory architecture. We further explore the possibility of using our in-place spares to enable lighter-weight ECC for the memory module while yielding a similar lifetime as a baseline ReRAM.

1.5 Permissions and Attributions

1. Chapter 1 contains material taken from "Toward Large-Scale Access-Transistor-Free Memristive Crossbars," by Amirali Ghofrani, Miguel-Angel Lastras Montaño, and Kwang-Ting Cheng, which appears in IEEE Asia South-Pacific Design Automation Conference (ASPDAC), 2015.

2. Chapter 2 contains material taken from "Towards Data Reliable Crossbar-Based Memris-

tive Memories," by Amirali Ghofrani, Miguel-Angel Lastras Montaño, and Kwang-Ting Cheng, which appears in IEEE International Test Conference (ITC), 2013.

3. Chapter 3 contains material taken from "A Low-Power Variation-Aware Adaptive Write Scheme for Access-Transistor-Free Memristive Memory," by Amirali Ghofrani, Miguel-Angel Lastras Montaño, Melika Payvand, Siddharth Gaba, Wei Lu, Luke Theogarajan, and Kwang-Ting Cheng, which appears in ACM Journal on Emerging Technologies in Computing Systems (JETC), Vol. 12, Issue 1, July 2015.

4. Chapter 4 contains material taken from "In-place Repair for Resistive Memories Utilizing Complementary Resistive Switches" by Amriali Ghofrani, Miguel-Angel Lastras Montaño, Yuyang Wang, and Kwang-Ting Cheng, which is submitted to International Symposium on Low Power Electronics and Design (ISLPED), 2016.

Chapter 2

Towards Data Reliable Crossbar-Based Memristive Memories

A series of breakthroughs in memristive devices have demonstrated the potential of using crossbar-based memristor arrays as ultra-high-density and low-power memory. However, their unique device characteristics could cause data disturbance for both read and write operations resulting in serious data reliability problems.

This chapter discusses such reliability issues in detail and proposes a comprehensive yet low area-/performance-/energy-overhead solution addressing these problems. The proposed solution applies asymmetric voltages for disturbance confinement, inserts redundancy for disturbance detection, and employs a refreshing mechanism to restore weakened data. The results of a case study show that the average overheads of area, performance and energy consumption for achieving data reliability, over a baseline unreliable memory system, are 3%, 4%, and 19% respectively.

2.1 Introduction

The evolutionary improvement of current memory technologies cannot keep up with the fast-growing demand for denser, lower-power, and higher-bandwidth memories. In traditional transistor-based memories, high leakage current is becoming a major concern, and imprecision in the fabrication process is reducing the yield to an alarming level as the technology feature size continues to shrink [22]. To address such problems, several new memory technologies have been proposed. Redox-Based Resistive Switching Memories [4], Phase Change Memories [2], and Spin-Transfer Torque Magneto-resistive Memories [3] are some of the emerging technologies that could possibly serve as the next-generation memories for various applications. Among these candidates, metal oxide valence change ReRAMs (more generally referred as memristor [5]) are especially promising due to excellent scaling prospects, high endurance and high speed which can also be combined with great retention [23, 24].

A memristor is a passive non-linear resistive device, the resistance of which depends on the time integral of current applied across its terminals. Hence, it maintains its resistance in the absence of electrical current, which makes it suitable as a non-volatile memory element. The theoretical foundation of memristors goes back to 1971, when L. Chua predicted the existence of such a device [25]. However, it took researchers decades to unify the theory with experimental observations [26].

There have been very active research efforts recently, in both industry and academia on various aspects of memristors [27, 28, 7, 29, 30]. Memristive devices of a feature size 15 nm has been fabricated in academia [31] to form a crossbar-based memory array [11]. A crossbar architecture has been used due to its high density and regularity. It is anticipated that feature sizes as small as 3 nm are feasible [32] due to simpler and imprecision-resistant fabrication process [33]. The fabrication process is CMOS-friendly [34], and efficient methods exist to stack layers of such memories [35] which facilitate the integration of 3D memristive

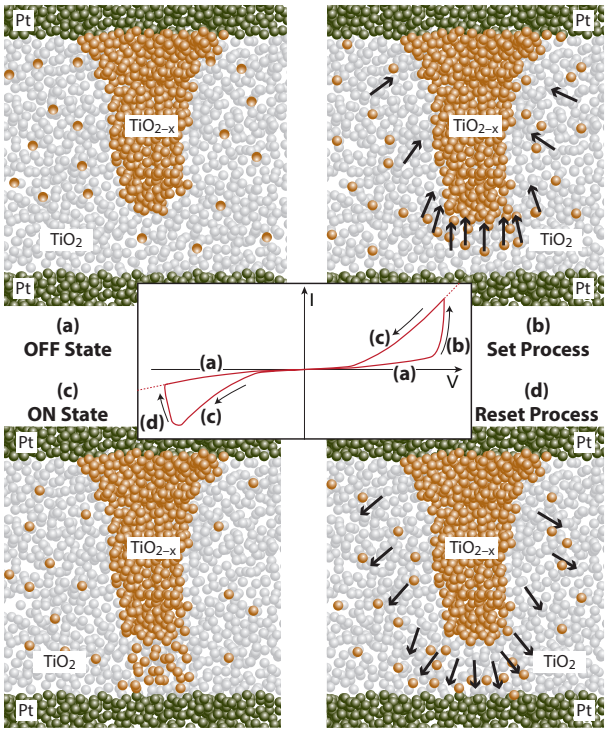


Figure 2.1: Memristor realization and typical hysteretic I-V behavior. (a) OFF state: An initial filament is formed during a one-time formation process. No conductive channel exists, thus the device is in high resistance state. (b) Set process: Applying a positive voltage drifts the dopants toward the filament, forming a channel, and decreasing the resistance. (c) ON state: a low-resistance channel is formed between the two electrodes. (d) Reset process: Applying a negative voltage repels the dopants and ruptures the channel, increasing the resistance.

memories with CMOS computing cores and decoding logic. Estimations as well as preliminary experimental measurements in their power consumption show considerable improvement over existing technologies [9, 23], as maintaining the data stored in memory does not incur any power consumption, and there is no active leakage current (as they are two-terminal passive elements). Reported experimental data show very fast write operations [36], while the speed of a read operation is limited by that of its CMOS sensing circuitry. All these characteristics make memristive memories ideal for integration with computing cores as an extremely dense and low-power on-chip non-volatile memory in the near future [6, 37].

However, there are some intrinsic characteristics of memristive memories that result in data reliability issues when memristors are used to form a crossbar-based memory. One issue

with such memories is an undesired coupling effect with which writing into one memristor may affect the data in several other memristors sharing the same word and/or bit lines. The effect is referred to as *write disturbance* [14, 11]. Moreover, as the resistance of a memristor is current-history-dependent, reading its resistance value by applying a read voltage across the memristor and measuring the resulting current can slightly change the strength of the stored data [38]. This effect is referred to as *read disturbance*. Both effects could be accumulative for a series of read/write operations which could result in data corruption and degrade the reliability of memory data. Thus these issues must be addressed before memristive memories can serve as system memories.

In this chapter, we first describe the data reliability issues of memristive memories in detail, and then present a comprehensive solution to address them. Our proposal is based on restraining the write disturbance effect, detecting data corruption by adding redundancy, and restoring/refreshing the disturbed data before corruption. We then evaluate the cost of the proposed solution in terms of the area, performance and energy overheads beyond the baseline crossbar structure.

The main contribution of this work is that it solves the data reliability problems of crossbar-based memristive memories. In addition, the proposed solution achieves the following goals:

- Incurring low area-, performance-, and energy-overheads.
- Using only standard memristive elements without adding any special elements, thus preserving the regularity and the scalability to achieve high-density memristor arrays.

The rest of the chapter is organized as follows: Section 2.2 provides the readers with the necessary background on memristors and memristive memory architectures. Section 2.3 describes the data reliability issues of memristive memories in detail. Section 2.4 presents our proposed solution followed by the experimental results in Section 2.5. We also elaborate on a comprehensive electrical model for the memristor crossbar array which is used to analyze the

performance and energy overheads of our solution. Section 2.6 concludes the chapter.

2.2 Background on Memristors

2.2.1 Device Physics

Figure 2.1 shows one possible realization of memristors. A simple memristor consists of three layers: two metallic electrodes, such as Pt, on top and bottom, and a doped thin film, such as TiO_2, in between.

In the initial state, a filament of conductive TiO_{2-x} is formed in the non-conductive TiO_2 film in an irreversible forming step [39]. However, the filament does not connect the two electrodes together, thus the device is in a High Resistance State (HRS). In order to turn ON the device, a sufficiently high positive voltage is applied across the electrodes of the device. This makes the filament connected to the top electrode attract positively charged vacancies in the oxide. This essentially grows the filament, as the vacancies start to drift in the applied electric field through the most favorable diffusion paths, and form a channel between the two electrodes [39]. Once such highly conductive channels are formed, the device is in Low Resistance State (LRS) and considered as ON.

To switch the device to the high resistance OFF state, a voltage with the opposite polarity should be applied on the electrodes. This repels away the vacancies that formed the conductive channel, thus shifting the device back to its high resistance state.

The state of the device and thus its resistance only changes when an electric current is passing through the device, and this change is continuous between two extremes: the R_{HRS} and the R_{LRS}. The change can be modeled according to the time integral of the current. R_{HRS} and R_{LRS} depend on the initial filament and are set in the forming step. However, forming free-devices are also under research [40].

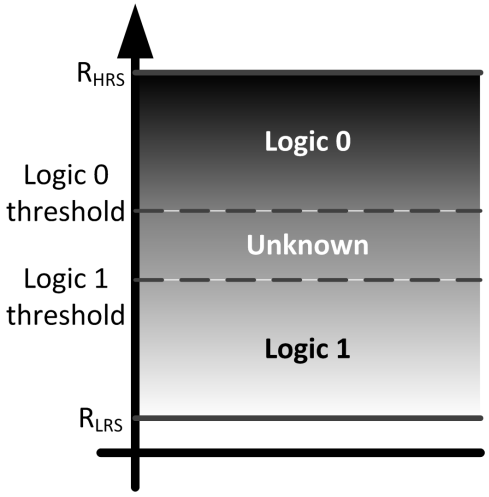


Figure 2.2: Different resistance regions.

2.2.2 Data Storage

In order to use memristors as memory elements to store binary data, the possible resistance range of the memristor is divided into three regions, as illustrated in Figure 2.2. Lower resistances are considered as logic 1 and higher resistances are considered as logic 0. Any resistance that falls in the marginal region in between is considered as unknown to ensure accurate distinction of logic 0 and 1.

Throughout this chapter, the term *value* is used to refer to a memristor's resistance value, while the actual binary data is referred to as *data*. Moreover, the term LRS (HRS) is generally used to refer to the range of resistance values representing logic 1 (logic 0).

2.2.3 Read and Write Operations

In order to write a binary data into a memristor, a proper write voltage (V_w) pulse of width t_{write} is applied across the device to set its resistance to the desired value. V_w and t_{write} are chosen so that the write pulse can completely shift the memristor's resistance to R_{HRS} or R_{LRS}, based on the polarity. That is, a negative pulse shifts the resistance toward R_{HRS} and a positive

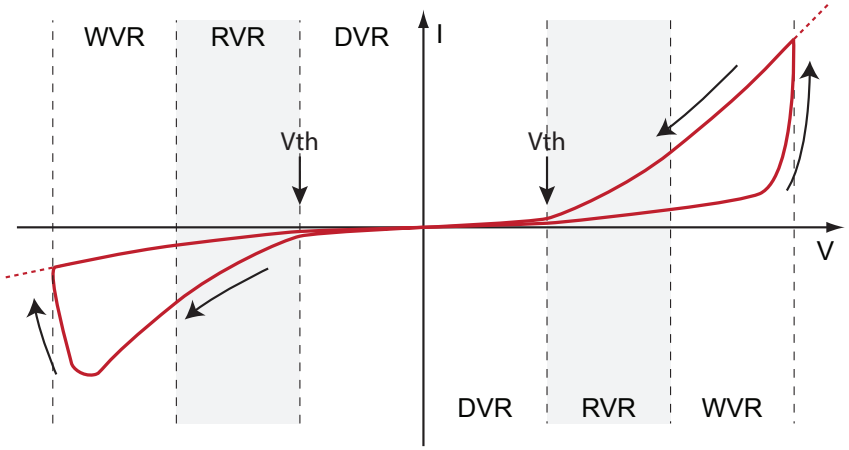


Figure 2.3: A typical I-V characteristic of a memristor. Three different voltage regions can be observed.

pulse shifts it toward R_{LRS}.

The read operation decides if the resistance is in LRS or HRS. To do so, a read voltage (V_r) is applied across the memristor terminals. This results in the injection of a current through the memristor, the magnitude of which depends on the memristor's resistance. The stored data can be read by measuring this current.

2.2.4 I-V Characteristics

Without loss of generality, Figure 2.3 can be used as a model of a memristor's I-V characteristics [41], based on which the following three regions can be defined:

- **Diode Voltage Region (DVR):** Applying a small voltage across the memristor terminals would not generate any noticeable current, and ideally would not change the device resistance. For example, this could be due to integrated Metal-Insulator-Metal (MIM) structure in series with the memristive layer. For a relatively small applied voltage, the bias would drop mostly across the MIM layer resulting in a negligible change of the resistance in the memristor.

  However, as the resulting current is negligible regardless of the memristor resistance,

an applied voltage in this region cannot determine the stored data. This kind of diode behavior has been further strengthened by the introduction of complementary resistive switches [42].

- **Read Voltage Region (RVR):** As the applied voltage rises above a certain threshold, the resulting current starts to increase considerably. This current is still small and just slightly changes the resistance of the device, but is large enough to differentiate between the current of a memristor in HRS or LRS and determine the stored data. Voltages in this range (both negative or positive) can be used for the read operations.

- **Write Voltage Region (WVR):** By further increasing the applied voltage, the resulting current increases even further, having exponentially higher altering effect on the resistance of the memristor. This is due to the highly nonlinear kinetics typically associated with truly non-volatile memristors [24]. Such voltages can effectively change the memristor's state from LRS to HRS (or vice versa, depending on the polarity of the applied voltage), and are used to write data into a memristor.

2.2.5 Memory Architecture

Different architectures have been proposed to utilize memristors to form a memory array. The most popular architecture is the crossbar organization, shown in Figure 2.4, which consists of two perpendicular layers of parallel nanowires forming a memristor at each cross section. The memory controller and address decoding circuitry are implemented in a peripheral CMOS subsystem [11].

However, the simple crossbar architecture encounters some scalability limitations: (1) voltage drop along long nanowires, can prevent effective application of read or write voltages on the desired cross-point, and (2) there is an upper limit on the maximum possible number of cross-points on each nanowire, imposed by noise margin requirements. To address such

limitations, in [43], authors proposed an innovative crossbar-based architecture called CMOL, that addresses the scalability issues by segmenting the crossbar nanowires, thus limiting their length and the number of cross-points per nanowire, while preserving the cross-point density.

In this chapter we use the simple crossbar architecture as the underlying memory architecture to illustrate the problem and our solution for the following reasons:

- It is easier, but without losing generality, to explain the concept using the simple crossbar architecture.
- Currently functional memristive memories are built in the form of a simple crossbar [11].
- The proposed solution can also be generalized to architectures such as [43], which are variants of the simple crossbar architecture, with minor modifications.

Here we are assuming an n-by-n crossbar memory, where M_{ij} refers to the memristor at the cross section of i^{th} row (also referred to as word-line) and j^{th} column (also referred to as bit-line).

2.2.6 Read and Write Operations in Crossbar

To write data into M_{ij}, a sufficiently wide pulse with an amplitude V_w in the WVR is applied across its terminals. Among several methods proposed to apply the V_w on the memristor [44, 30], the least intrusive one is applying $V_w/2$ on the i^{th} word-line, $-V_w/2$ on the j^{th} bit-line, and grounding all other word- and bit-lines.

For a read operation, a sufficiently wide pulse of amplitude V_r in the RVR is applied across the memristor in a similar way. Then a sensing and comparing (S&C) circuitry is used to read the resistance value. One possible implementation of such circuitry is shown in Figure 2.5. As the current passing through a transistor is a function of its gate-source voltage, the diode-connected transistor T in S&C circuitry makes the gate voltage (V_{out}) follow the transistor's

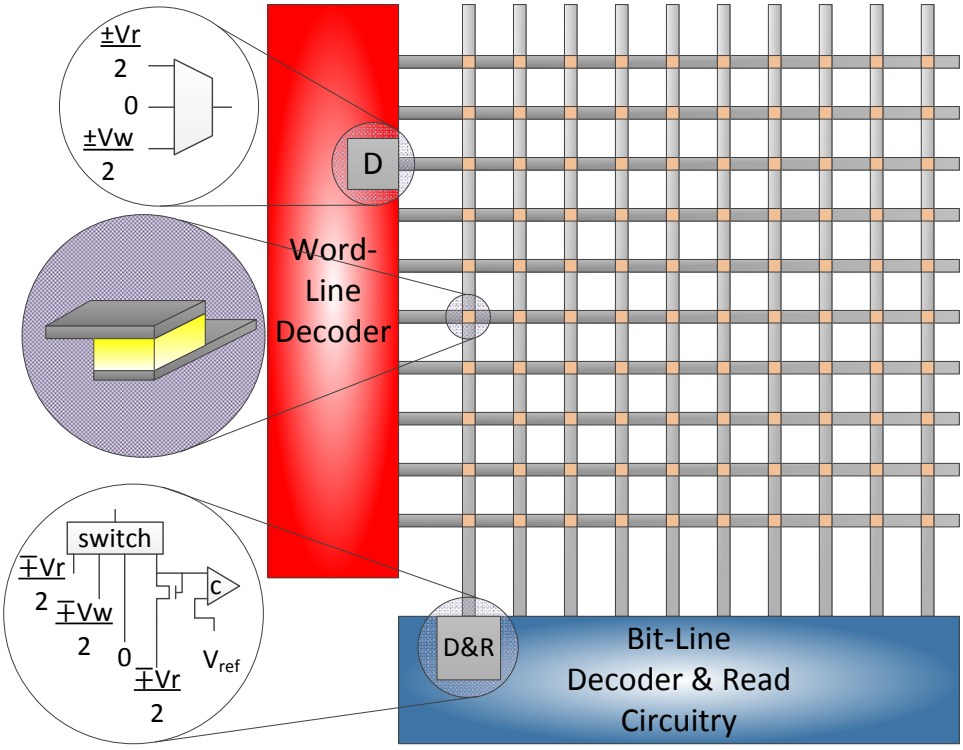


Figure 2.4: Crossbar memory architecture. To access each memristor, a bit-line and a word-line are selected and applied with appropriate voltages which depend on the read or write access. Other lines are grounded.

current, essentially converting the current to a voltage. Since the transistor's and the memristor's currents are identical, and depend on the memristor's resistance, V_{out} reflects the value of the memristor's resistance. This voltage is then compared with a reference voltage (V_{ref}) to determine the stored binary data. Ideally, V_{ref} is set between the V_{out} of a memristor in LRS and that of a memristor in HRS.

2.3 Data Reliability Issues in Memristive Memories

2.3.1 Read Disturbance

As current flows through the device during the read time, the device's resistance might slightly change. This *read disturbance* effect, mostly affects the target memristor, and not the

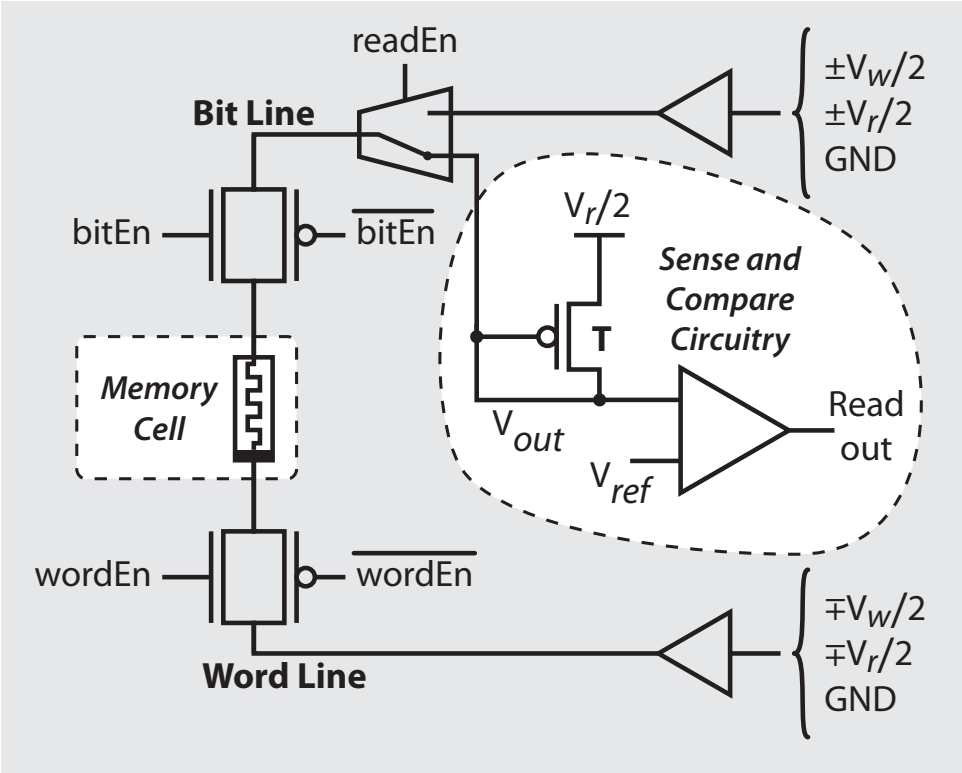


Figure 2.5: Sensing and Comparing circuitry. The comparator is an Op-Amp. The memristor's read current passes through transistor T. By connecting T's gate to its drain, the drain voltage V_{out} reflects the read current, as T's current depends on its V_{GS}.

other memristors in the crossbar array. The read voltage V_r is chosen to meet two criteria: (1) it is within RVR, and (2) $V_r/2$, the voltage applied to other memristors that share the same word- or bit-line with the target memristor, falls within the DVR. The second criterion ensures that the side-effect of the read operation on other line-shared memristors is negligible.

It should be noted that not every read operation is disturbing. For a given polarity of V_r, only one of the two logic values will be disturbed. For example, assume a positive V_r is applied for the read operation. If the stored data is logic 0 (i.e. its resistance value is within HRS), the resulting current will slightly shift its resistance toward R_{LRS}, making the stored data a weaker 0. However, if the stored data is logic 1 (i.e. LRS) the read current will have a healing effect, by shifting its resistance toward R_{LRS}, thus making the stored data a stronger 1.

Few articles in the literature addressed the read disturbance problem. In [38] the authors

propose the use of more complex read pulses consisting of alternating V_r and $-V_r$ pulses to compensate for the destructive effect of the read operation. The alternate pulse, appended to the original read pulse, heals the destructive effect of the original read. However, this method doubles the read time and energy unnecessarily, as not all read operations are disturbing. The incurred overhead is particularly expensive as the read operations are more time consuming than the write operations for memristive memories.

In Section 2.4, we propose a couple of different restoring schemes to address this issue. Our solution reduces energy overhead of a reliable read operation by triggering data restoration only for disturbing reads and expedites the data restoration by utilizing other existing voltages in the system.

2.3.2 Write Disturbance

Applying $V_w/2$ and $-V_w/2$ on the word- and bit-lines respectively to write data to a memristor has an undesired side effect: a $V_w/2$ voltage (which falls within RVR) is also applied to all memristors that share either the word- or the bit-line with the memristor under write, which can slightly change their resistances. This effect can be disturbing or healing based on the written logic and the logic stored in the line-shared memristors: If memristor M stores a logic 0, writing a logic 1 (logic 0) to one of its line-shared memristors shifts M's resistance toward logic 1 (logic 0), weakening (strengthening) the stored logic. Same thing happens to all other memristors on the same line as the memristor-under-write and storing a logic 0. Note that the *write disturbance* problem is harder to deal with than read disturbance due to its broad impact. Figure 2.6 illustrates the effect of write disturbance on other line-shared memristors.

One solution is to add a switch (i.e. transistor) for each memristor to enable the isolation of a memristor from the rest of the memory array [45] (referred to as the 1T-1M technique), thus avoiding the destructive effect on the line-shared memristors. However, this technique

encounters the same technology scaling limitations as other transistor-based memories, due to the integration of the transistors.

In Section 2.4, we propose a solution to this problem by having additional ordinary memristors with known data content in the memristor layer. These extra memristors are used as references for detecting possible data corruption. This solution, to the best of our knowledge, is the first solution to the write disturbance problem that preserves the scalability advantages of the memristor technology.

2.3.3 Disturbance Accumulation

The data reliability problem arises from the fact that the effects of read and write disturbances are accumulative and could eventually lead to data corruption. That is, a memristor's resistance can be shifted to the unknown region or even the opposite logic region. Figure 2.7 illustrates the disturbance accumulation after a sequence of write operations.

2.4 Addressing Memristor Data Reliability

Read and write disturbances are intrinsic features of memristors, which if not addressed, will result in frequent data errors, that cannot be handled only by system-level solutions such as Error Correction Codes (ECC). Here we try to prevent, detect, and resolve data errors caused by such disturbances by proposing a circuit- and architecture- level solution. However, ECC can always be used in conjunction with our method to provide additional protection.

2.4.1 Read-Restore solution for Read Disturbance

The read-restore mechanism in [38] can be optimized for energy efficiency by detecting destructive reads. Then, only in case of a destructive read, the read operation is extended by

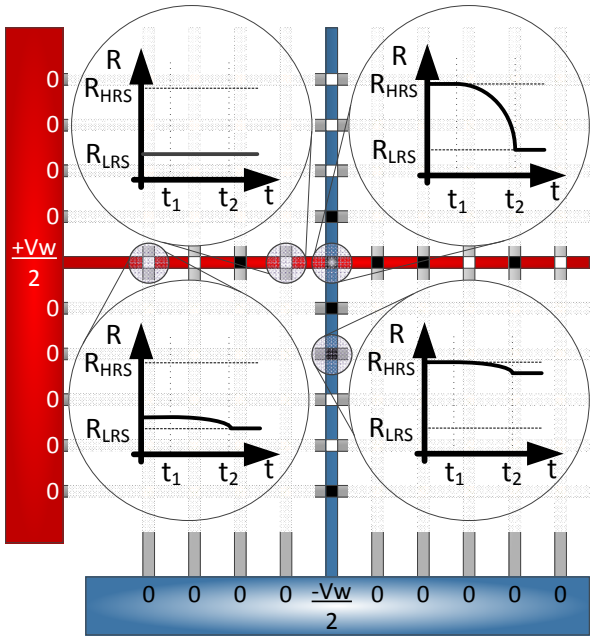


Figure 2.6: Write Disturbance. V_w is applied between times t_1 and t_2 to set the memristor-under-write to logic 1 (LRS). Black (white) memristors are in HRS (LRS). The data is written into the target memristor correctly (top-right). The white memristors sharing the same row or column are either not affected (top-left) or slightly healed (bottom-left), but the black ones are slightly disturbed (bottom-right).

applying a voltage of an opposite polarity to heal the destruction. That is, if the original read uses V_r ($-V_r$) which causes disturbance, then the value is restored by applying $-V_r$ (V_r). Note that after the original read operation, both the stored data and the polarity of V_r are known. Hence it is known if the read operation was destructive or not. The peripheral memory controller circuitry is extended to differentiate a disturbing read from a non-disturbing one. Moreover, during restoration, the power-hungry S&C circuitry is turned off which helps minimizing the energy overhead.

However, restoring by applying V_r roughly doubles the read time, because the restoring pulse with the opposite polarity needs to have the same pulse width as the original read pulse in order to recover the disturbing effect.

To accelerate the restoring process, we propose applying a larger voltage V_w instead. This can improve the performance of restoring operation by one order of magnitude since the higher

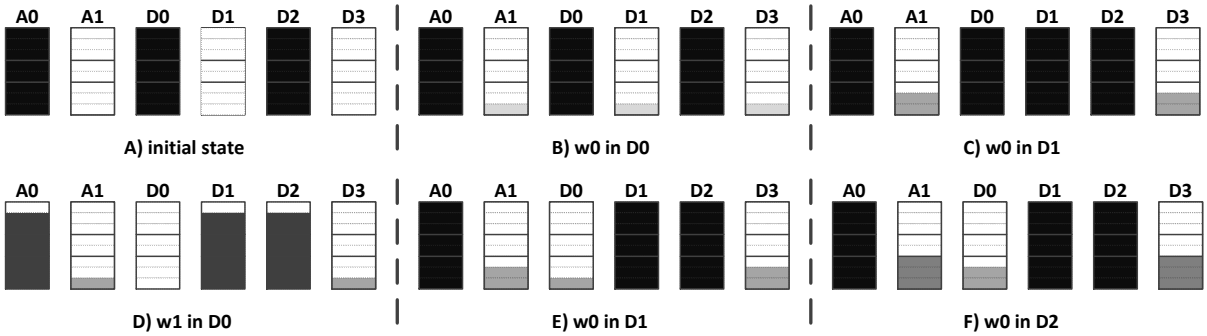


Figure 2.7: Adding redundancy for corruption detection. (A) The initial state of the memory. The gray-scale color coding is used to show the resistance (i.e. Black for logic 0, and white for logic 1). Discrete resistance levels are assumed for illustration. A1 contains logic 1 and A0 contains logic 0, while the rest of cells have random data. (B) Status after writing logic 0 (w0) to D0. As D0's resistance was already R_{HRS}, it is not changed. However, it shifts the resistance of the cells storing logic 1, referred to as 1-bits hereafter, (D1, D3 and A1) one level toward logic 0. (C) Status after w0 in D1. This changes the value of D1, and further weakens the data in 1-bits. (D) Status after writing 1 (w1) in D0. This changes the value of D0, weakens the data in memristors storing logic 0, called 0-bits hereafter, (D1, D2 and A0), but strengthens the data in 1-bits (A1 and D3). (E) Status after w0 in D1. This changes the value of D1, weakens the data in 1-bits (A1, D0, D3), and strengthens the data in 0-bits (A0 and D2). (F) Status after w0 in D2. At this point, the resistance value of A1 has reached the corruption level, which triggers the refreshing of 1-bits (A1, D0, D3). It can be observed that A1 has the worst-case disturbance among the 1-bits on the word-line, since other bits are either equally or less disturbed.

V_w can heal the degraded data faster and more efficiently. Moreover, since the voltage V_w is already available, as it is used for the write operation, no extra voltage resources are needed to implement this method.

The potential problem with the idea of restoring by V_w is the write disturbance effect on other line-shared cells. However, our write disturbance solution described in the next subsection, resolves this side-effect and makes it possible to use the V_w for data restoration. Note that the width of a restorative V_w pulse is shorter than that of a normal write V_w pulse, thus its negative effect is also less significant.

Applying a higher restorative voltage V_w incurs higher energy consumption as: (1) larger current passes through the target memristor, (2) other line-shared memristors will experience

higher partial voltage $V_w/2$, which is in RVR region, thus generates more current. The energy and performance trade-offs of this method will be illustrated in Section 2.5.

2.4.2 Redundancy-based Corruption Detection for Write Disturbance

Write disturbance affects all memristors sharing the same word- or bit-line with the target memristor. Our solution addresses the problem by employing the following principles: (1) limiting the disturbance to only those memristors sharing the same word-line with the target, (2) adding the capability of detecting the disturbance accretion, and (3) refreshing the disturbed data before it is corrupted.

The reason that both word- and bit-lines are affected by the write operation is the common assumption of applying symmetric voltages of $\pm V_w/2$ on word-lines (bit-lines). In order to confine the domain of disturbed memristors, we propose asymmetric application of V_w, i.e., applying a higher absolute voltage on the word-lines, and a lower voltage which falls within the DVR of memristors, on the bit-lines. This makes it easier to address the write disturbance effect, by protecting the memristors on the bit-line from write disturbance at the cost of having more destructive effect on the word-line-shared memristors. For asymmetric voltage application, we propose applying $2V_w/3$ on the word-line and $-V_w/3$ on the bit-line, where we assume:

$$\frac{V_w}{3} = \frac{V_r}{2} \Rightarrow \frac{2V_w}{3} = V_r \tag{2.1}$$

This offers several advantages: (1) the bit-line-shared memristors will not experience write disturbance as $V_w/3$ (i.e. $V_r/2$) is always within the DVR. (2) The voltage applied to the word-line-shared memristors is equal to V_r, making it possible to read other cells in the same word-line simultaneously as the target memristor is written, by just enabling their S&C circuitry (i.e. sensing and comparing). (3) The number of required voltage levels remains the same (i.e. $\{\pm 2V_w/3, \pm V_w/3, \text{GND}\}$ instead of $\{\pm V_w/2, \pm V_r/2, \text{GND}\}$). Note that while other asymmetric

voltage applications are also feasible (i.e. $\pm$3Vw/4 and $\mp$Vw/4, etc.), that will increase the number of required voltage levels.

In the next step, we add the capability of detecting data corruption before the resistance change accumulates to the level of moving the memristor to the unknown state. The key difficulty for such detection is that if the correct data stored in the memristor is unknown, it is not possible to distinguish a weakened but correct data from an already corrupted (inverted) data.

To address this challenge, we propose the addition of an always-1 (A1) and an always-0 (A0) memristors in each word-line, as shown in Figure 2.8, to facilitate the detection of data corruption. Such bits are ordinary memristors, initially set to their corresponding states (LRS for A1 and HRS for A0). The only difference is that the user does not have write access to these cells, which can be ensured by a proper decoder design. There are two nice features of having such bits on the word-line: (1) As their correct binary data is always known, detection of data corruption for them becomes feasible, and (2) A write operation disturbs them in the same way as it disturbs other memristors on the same word-line. This makes them experience the worst possible case of accumulated disturbances among all cells on the same word-line, as they are never written into through standard memory accesses. Unlike them, other cells may have been written into by write accesses, which offset the accumulated disturbance.

This means that the A0 (A1) cell always has the weakest 0 (1) on their word-line. Thus, as long as the resistance value of the A0/A1 cells stays within the correct range, which can be ensured by continuously monitoring them, the integrity of the data stored in other cells on the same word-line can be guaranteed. Figure 2.7 illustrates the idea.

According to the asymmetric voltage application for the write operation which applies $2V_w/3$, that is V_r, to the memristors on the word-line, the value of other cells on the same word-line, thus the A0 and A1 cells, can be read and monitored simultaneously in every write cycle, using the same Sense and Compare (S&C) circuitry as shown in Figure 2.5.

Note that A0/A1 bits intend to detect a potential corruption *before it actually happens* to trigger the refreshing mechanism. Hence, the reference voltages (V_{ref}) of the S&C circuitry on the A0 and A1 bit-lines, are chosen accordingly to ensure that the output of the comparator is asserted close to but *before* the corruption. When this happens, a refresh is required on the close-to-corruption logic. That is, if the output of the A0 (A1) bit-line is asserted, all the memristors on the same word-line storing a 0 (1) should be refreshed.

Note that here it is assumed that A0 and A1 are disturbed exactly in the same way as any other memristor on the word-line for clarification purposes. In general case, there might be small variations. The small resistance of the nanowires may result in a voltage drop along the line, which in turn causes the memristors to experience slightly different disturbing effect. This can be addressed by placing the A0/A1 bits closest to the word-line driver. Hence, they experience the worst case disturbance effect, as they are not affected by the voltage drop. Moreover, the disturbing effect might slightly differ among memristors due to the process variation. Conservative adjustment of A0/A1 corruption threshold (V_{ref}), can take this variation into account to consider the worst case.

The next step is refreshing the close-to-corruption data. That is, if A0 cell's 0 becomes too weak, all the 0s on the line are refreshed. Refreshing the memristors storing logic 0 (0-bits) consists of two steps: (1) Finding out which memristors are 0-bits, for which all bits on the word-line of interest are read simultaneously, by applying $V_r = 2V_w/3$ on the target word-line, grounding all bit-lines, and turning on the S&C circuitry, and (2) Refreshing the 0-bits simultaneously by applying a write voltage $-2V_w/3$ on the word-line, and $V_w/3$ on all the bit-lines whose corresponding memristors need to be refreshed, while grounding other bit-lines.

During the refresh procedure, A0 is also refreshed, thus it experiences the same refreshing imperfection, if any, that other 0-bits might encounter (e.g. not wide enough refreshing pulses, etc.). Similarly, during the refresh, the A1 bit on the same word-line will experience the same side-effects as other memristors storing logic 1 (e.g. disturbance of their value due to the

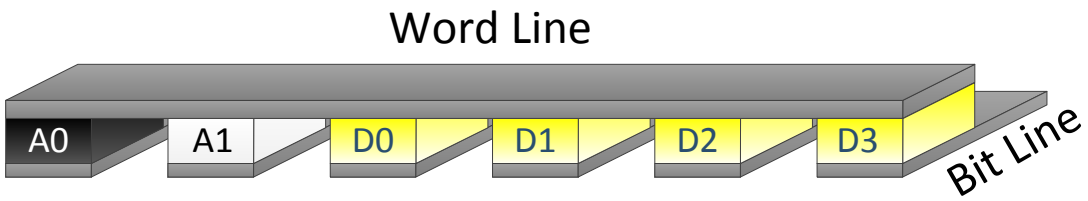


Figure 2.8: The configuration of the A0 and A1 on the word-line, assuming a word-line of 4 bits. All the word-lines would be similar.

refreshing of 0-bits). Hence the method is robust and will not be affected by such imperfections or side-effects.

The main parameter affecting the refresh rate is the *Write Disturbance Tolerance* (WDT), which is defined as the number of consecutive writes of only logic 1 (0) before corrupting the resistance of the line-shared memristors from a strong 0 (1) to the unknown state. The higher the WDT, the lower the number of refreshes needed. This number depends on two factors: (1) the applied write voltage V_w, as a lower write voltage has a smaller destructive effect, and (2) the non-linearity of the device's I-V curve, as higher non-linearity would help decrease the destructive side effects of write accesses. Hence, with technology advancement and introduction of devices with better non-linear kinetics, WDT would continue to improve. Measurements in [16] show that applying partial voltage of $2/3V_w$ (i.e. the partial voltage that causes write disturbance) takes $100\times$ more time (i.e. 100 write operations) to completely change the state of the device compared to when V_w is applied. Hence, assuming equal division of the possible resistance range into logic 0, logic 1, and unknown regions, it can be deduced that WDT for current memristive devices is ≈ 33 (i.e. $\frac{100}{3}$).

The average number of random write operations (logic 0 or 1) that necessitates a refresh is estimated based on WDT and is called $\psi_{(WDT)}$ hereafter. ψ is used for energy and performance overhead estimation and is calculated by Monte Carlo simulations. In that, we count the number of refreshes required during a run of 10^9 write operations for different WDTs. ψ is then obtained by dividing the total number of write operations by the number of refreshes. Figure 2.9 shows the number of refreshes and the resulting ψ versus WDT.

As the disturbance effect is confined to the word-lines, the number of bit-lines has no effect on ψ. Moreover, as it is assumed that write operations on any memristor on the word-line affect the A0/A1 bits similarly, the number of memristors on the word-line does not affect ψ either. It should be noted that the proposed method guarantees the data integrity regardless of $\psi_{(WDT)}$, which only changes the refresh rate.

The energy and performance overheads of the proposed solution, as well as the effect of WDT on those metrics will be discussed in Section 2.5.

2.5 Experimental Results

We derived an electrical model for crossbar-based memories using the Cadence Virtuoso tool, and designed the peripheral CMOS Comparing and Sensing, and decoding circuits. With these circuits and models, we simulated the electrical properties of the crossbar and evaluated the energy consumption and performance of the proposed solution.

In the following, we first elaborate on the electrical model, based on which we discuss the overhead figures of our solution.

2.5.1 Electrical Model and Experimental Setup

The crossbar structure is shown in Figure 2.10, which is represented as two perpendicular layers of parallel nanowires. The separation of parallel nanowires is $\alpha \times F_{\text{nano}}$, where F_{nano} is the width of the nanowire and α would be 2 for the highest density. $t \times F_{\text{nano}}$ in Figure 2.10 represents the thickness of nanowires.

In order to electrically model each nanowire, they are partitioned into nanowire segments of length αF_{nano} and a resistor and a capacitor are used to model each segment. The resistance per nanowire segment can be extracted using the cross-sectional area and the resistivity ρ of the

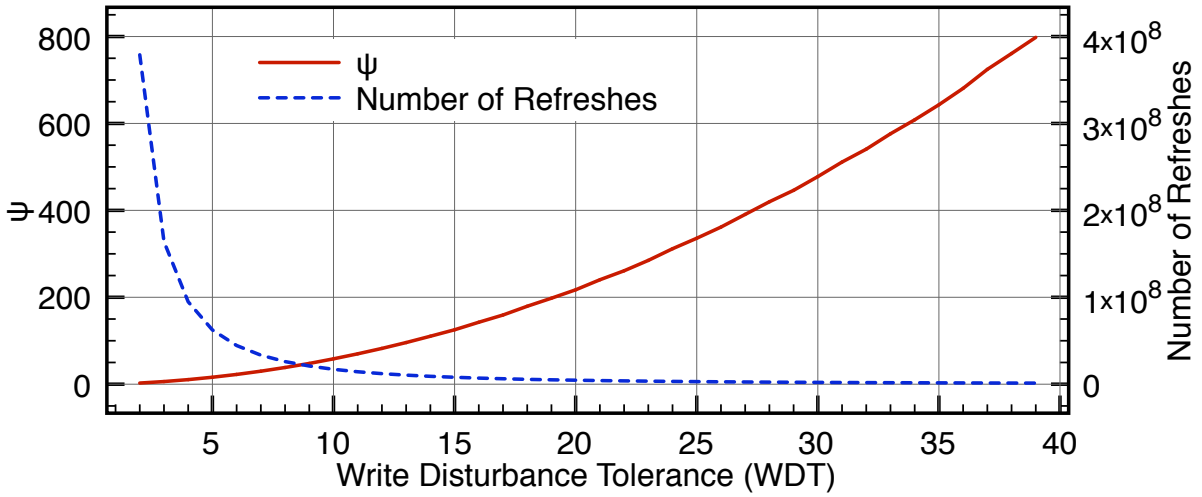


Figure 2.9: The number of required refreshes in a run of 10^9 write operations (dashed) and the average number of writes before a refresh is required, called ψ (solid), for different WDT numbers. As WDT increases, the number of refreshes drops significantly, which in turn increases ψ considerably.

material:

$$R_{\text{seg}} = \rho \frac{\alpha F_{\text{nano}}}{t F_{\text{nano}}^2} = \rho \frac{\alpha}{t F_{\text{nano}}} \tag{2.2}$$

It is a well-known effect that in nanometric scales, the electrical resistivity (ρ) of a material increases when the mean free path of the electrons in the bulk material becomes comparable to the dimensions of the structure. The expected increment in the resistivity [46] is considered and is plugged in Equation 2.2 to estimate the resistance of the segment.

As for the capacitive effect, we use the results obtained in [37] in which the capacitance per nanowire segment can be approximated as:

$$C_{\text{seg}} \approx 0.48 \times 10^{-10} \varepsilon \alpha F_{\text{nano}} \tag{2.3}$$

where ε is the relative dielectric constant of the insulating material. For SiO_2, $\varepsilon = 3.9$.

Hence, for a given feature size F_{nano}, pitch αF_{nano}, and relative wire thickness t, we can

extract the capacitive and resistive components for each nanowire segment and form an RC network that is driven by lateral circuitry, as shown in Figure 2.11.

The memristive devices, formed at every cross-point, are modeled based on the model proposed in [47] which considers the dynamic characteristics of the device.

The peripheral S&C circuitry (i.e. Sense and Compare) is implemented in 45 nm CMOS technology and uses a latch-based comparator based on [48] to produce the output. This comparator only latches the output at selected times and is effectively turned off at other times for energy saving. However, the energy consumption of read operation is mainly consumed in the S&C circuitry.

Crossbar memories of size 1Kb to 64Kb are modeled and simulated to estimate the performance and energy overheads. We do not show simulation results for larger memory due to the following reasons: (1) The use of spice-level simulation limits the memory size for simulation. (2) Simple crossbar does not scale well for larger memories. Instead, as stated earlier, other crossbar-based architectures such as CMOL [43], enhanced from the simple crossbar but with a similar number of cross-points per nanowire segment, has a much larger capacity and thus addresses the scalability issue. While the proposed method can be adapted to these architectures, in order to show results of a significantly larger memory size under these architectures, the results must accompany an in-depth explanation of these architectures which is prevented due to space limitation. Therefore, we illustrate the trends using the memory sizes in the range of 1Kb to 64Kb (i.e. 32 to 256 cross-points per nanowire) under simple crossbar architecture. Larger memories under those enhanced architectures should follow a similar trend.

Table 2.1 summarizes the estimated energy consumption and timing numbers of baseline (unreliable) read and write operations, based on our electrical model. Memories with higher number of cross-points per nanowire have considerably higher energy consumption due to the increase in the number of partially activated line-shared devices.

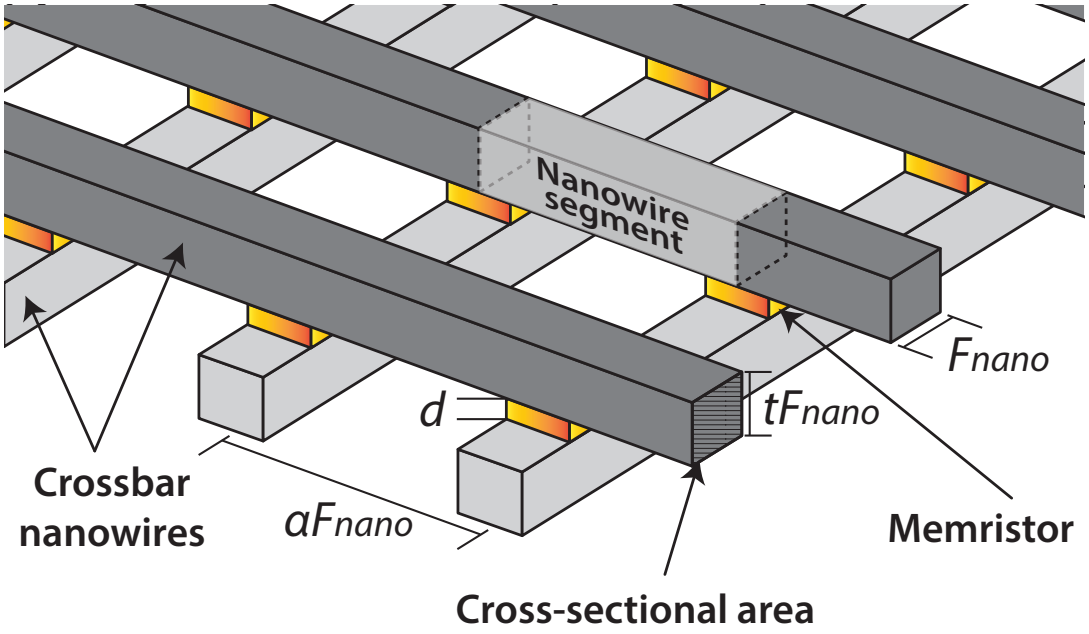


Figure 2.10: Physical characteristics of crossbars

2.5.2 Read Disturbance

Figure 2.12 illustrates the timing and energy consumption of different restoring methods in our simulations. The estimated energy consumption and timing numbers of those methods are demonstrated in Table 2.2 for different memory sizes. Performance and energy overheads are calculated over the baseline read operation, based on the numbers shown in Table 2.1.

It can be observed in the sixth column that restoring by V_r has a prohibitively large (≈80%)

Table 2.1: estimated energy consumption and timing numbers for baseline read and write operations

Size		Time (ns)			Energy (fJ)		
Xpoint per nanowire	Memory (Kb)	read	write	decoder	read	write	decoder
32	1	5.00	2.44	0.30	36.7	37.2	160
64	4	5.00	2.45	0.34	36.9	49.7	130
128	16	5.00	2.48	0.38	37.6	74.1	120
256	64	5.00	2.52	0.43	39.1	119.8	118

performance overhead, as the restorative pulse has approximately the same width as the original read pulse to heal the destructive effect. However, as shown in the eighth column, this method offers very low energy overhead ($<1\%$), because the power-hungry S&C circuitry for the typical read operation is turned off during restore operation. Moreover, the restorative pulse width is a bit shorter than the original pulse, as it is directly applied to the bit-line, rather than being applied through the sensing circuitry that delays the effective application of the voltage. This also contributes to the lower energy overhead.

By applying a restorative V_w pulse instead, the performance overhead can be improved significantly (to $\approx 8\%$), as shown in the seventh column. This is due to the high non-linearity of memristive devices: a higher voltage changes the device state significantly faster. However, as demonstrated in the ninth column, this method incurs higher energy consumption ($<4\%$), since the partial voltage on the line-shared memristors is not in the DVR range anymore, thus injecting more current through those partially selected memristors.

It can be observed that the energy consumption increases with more memristive devices on each nanowire, as more devices will be partially activated due to the partial restorative voltage applied on the line-shared memristors. However, the effect of memory size on performance is negligible, as the increase in restoring time is small.

There is an energy-performance trade-off between the proposed restoring methods. However, considering the fact that memristors offer a huge improvement in energy rather than

Table 2.2: estimated energy consumption, timing, and overheads of the reliable read operation

Size	Time (ns)		Energy (fJ)		Performance Overhead (%)		Energy Overhead (%)	
Xpoint per nanowire	restore by Vr	restore by Vw	restore by Vr	restore by Vw	restore by Vr	restore by Vw	restore by Vr	restore by Vw
32	4.32	0.435	0.73	2.00	81.4	8.20	0.185	0.51
64	4.23	0.436	0.86	3.38	79.1	8.16	0.257	1.01
128	4.20	0.441	1.18	6.24	77.9	8.19	0.374	1.98
256	4.22	0.462	1.95	12.32	77.8	8.52	0.623	3.93

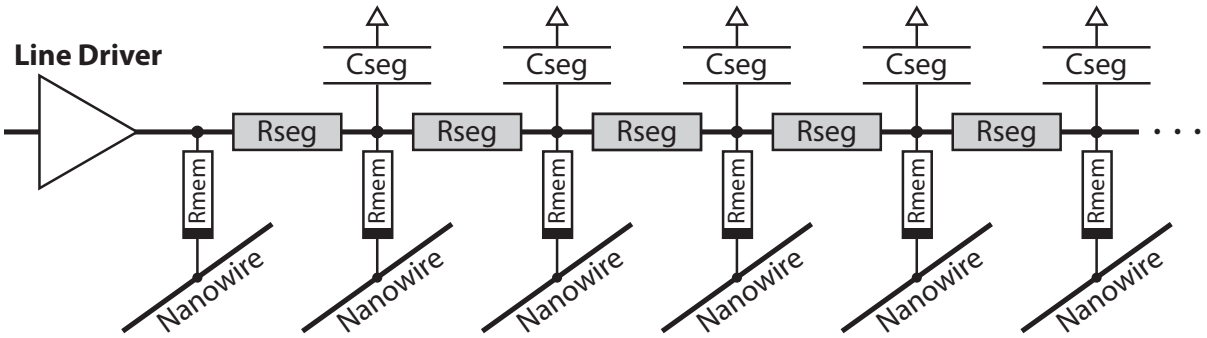


Figure 2.11: Electrical model of a nanowire in the crossbar

in performance, using restorative V_w pulses would be a better choice. Application of other restorative voltages is also feasible, but generating additional voltage levels would increase energy consumption and is not desirable.

2.5.3 Write Disturbance

The performance overhead of our write disturbance solution is due to the refresh operation. The refresh procedure takes $t_{read} + t_{write}$, which is due to (1) simultaneous read of all memristors on the word-line, and (2) concurrent refreshing of those memristors storing the close-to-corruption data.

The meantime between refreshes (MTBR) depends on: (1) $\psi_{(WDT)}$ (i.e. the average number of random writes before a refresh is required), and (2) the probability of having write accesses, as lower write probabilities (P_{write}) slows down the accumulation of the write disturbances, thus decreases the refresh frequency. That is, if $\alpha = P_{read}/P_{write}$, then for every $\psi_{(WDT)}$ write operations (that on average necessitate a refresh), $\alpha \times \psi_{(WDT)}$ read operations have been performed, which increases the MTBR, thus reducing the performance overhead. Moreover, since each read/write operation should be decoded as well, t_{dec} (i.e. decoding time) is also considered for every operation.

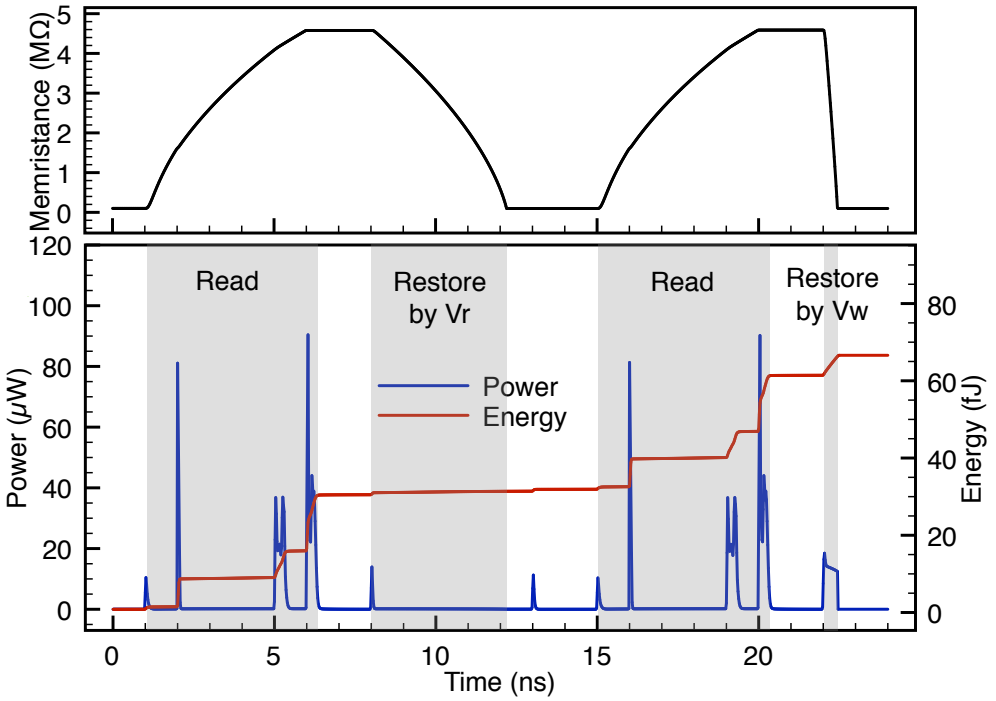


Figure 2.12: Different restoring schemes. The top part shows the target memristor's resistance that is disturbed during the read operation. Applying V_r takes longer to restore the value with negligible energy overhead, while a restoring V_w pulse quickly restores the value but with larger energy overhead.

Equation 2.4 estimates the performance overhead of the proposed method:

$$\begin{aligned} \text{Perf. Ovhd.} &= \frac{t_{refresh}}{\text{Meantime between refreshes}} \\ &= \frac{t_{read}+t_{write}}{\psi_{(\text{WDT})}\big((1+\alpha)t_{dec}+\alpha t_{read}+t_{write}\big)} \end{aligned} \tag{2.4}$$

Assuming equal read/write access probabilities, WDT equal to 33, and timing parameters extracted from simulation (Table 2.1), the performance overhead would be ≈0.15%, which is insignificant.

The energy overhead of the proposed method is due to the energy consumed for: (1) reading the A0/A1 bits, which is required for every write operation, and (2) performing the occasional refreshing process.

Refreshing energy overhead is caused by: (1) reading the value of all memristors on the

word-line, and (2) writing data back in those cells which should be refreshed. Hence, as multiple memristors should be read/refreshed, the refreshing energy also depends on the number of memristors on the word-line, *WS*, and the number of cells to be refreshed, *RC*. Since refresh procedure is triggered only when necessary, the refreshing energy should be divided among all the write operations performed between two refreshes (i.e. $\Psi_{(WDT)}$) to get the average energy overhead per write operation.

Equation 2.5 estimates the average energy overhead, where E_x shows the energy consumption of operation x:

$$\text{Energy Overhead} = \frac{2 \cdot E_{read} + \frac{\text{WS} \cdot E_{read} + \text{RC} \cdot E_{write}}{\Psi_{(\text{WDT})}}}{E_{write} + E_{decode}} \tag{2.5}$$

Table 2.3 summarizes the estimated energy and performance overheads of the reliable write operation, calculated based on Equations 2.4 and 2.5 and the timing and energy numbers presented in Table 2.1. Numbers are calculated for exemplar WDT value equal to 33.

Memories with fewer number of cross-points per nanowire have lower (≈40%) energy overheads, as fewer memristors are written (refreshed) and the energy consumption per write operation is small, while decoding is done in CMOS and consumes more energy. As the number of cross-points per nanowire increases, the energy consumption due to refreshing increases since:

Table 2.3: Energy and performance overhead of the reliable write operation over the baseline write operation for different memory sizes.

Xpoint per nanowire	Memory Size (Kb)	Performance Overhead (%)	Energy Overhead (%)
32	1	0.161	38.7
64	4	0.159	45.0
128	16	0.157	47.3
256	64	0.156	51.5

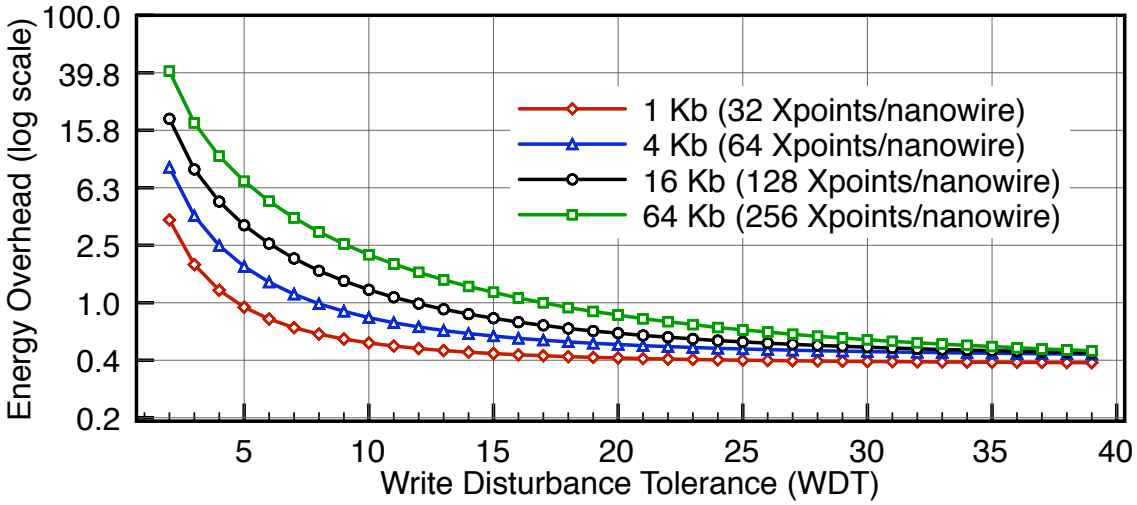


Figure 2.13: Write Disturbance Tolerance (WDT) vs. energy overhead of a reliable write over the baseline write.

(1) more memristors should be refreshed, and (2) the refreshing is more energy consuming due to the increase in the number of partially activated devices (due to the partial $2V_w/3$ on the word-line-shared memristors). However, note that in scalable crossbar-based architectures such as CMOL [43], the number of cross-points per nanowire segment does not increase as the memory scales. Thus, when applied to such structures, our proposed method will not suffer from this increment in energy overhead.

Figure 2.13 shows the effect of different WDT's on reliable write operation's energy overhead for different memory sizes in logarithmic scale. Smaller Write Disturbance Tolerances (WDT) necessitate frequent refresh operations, thus increasing the energy overhead of a reliable write operation. As WDT increases, the refresh rate and thus the energy overhead of a reliable write operation decreases to $\approx$40%. It is also shown that having higher number of cross-points per nanowire increases the energy overhead, as described before.

As for area overhead, the proposed method adds only two memristors (i.e. A0 and A1) on each word-line regardless of the word size. Hence, the area overhead depends on the word size and is equal to $\frac{2}{\text{Word Size}}$. For an exemplar word-line containing 64 memristors, this overhead is

3.12%.

2.5.4 Discussion on the Effect of the Device Variation

The degrading effect of the write disturbance could vary from device to device on the same word-line due to the variation in the switching characteristics of memristive devices. Hence, the corruption threshold for the A0/A1 cells, V_{ref}, should be adjusted conservatively to accommodate for devices that show the highest sensitivity to the write disturbance effect. Such a conservative corruption threshold translates into a smaller WDT, which in turn could increase the refresh rate and thus the energy overhead of the proposed method. However, according to Figure 2.13, the energy overhead of the proposed method tends to saturate for larger WDTs. Hence, with a large WDT (e.g. > 40), the reduction of the effective WDT due to a conservative threshold selection would have negligible effect on the energy overhead of the proposed method.

In order to have a reliable memristive memory module, one should consider that the sensitivity of memristive devices to write disturbance could also change over their lifetime. To accommodate for such over-time variations, a calibration scheme should be employed to adjust the corruption threshold of A0/A1 monitoring cells periodically.

2.6 Conclusion

In this chapter we addressed the data reliability issues of the emerging crossbar-based memristive memories.

The read disturbance problem is addressed by a read-restore mechanism. Utilizing the voltage levels already available in the memory system, two restoring methods are proposed and are evaluated for their energy-performance trade-offs.

The write disturbance issue, that affects the memristors on the same word-/bit-line as the memristor-under-write, is addressed by first limiting the disturbance domain only to the

memristors sharing the same word-line, through asymmetric distribution of the write voltage V_w. Furthermore, the possible corruption of data is detected by adding two extra memristors without write access on each word-line, which store logic 0 and 1 respectively and are used as references to check corruption trend and status. A refreshing scheme is also proposed to refresh the disturbed cells.

One main advantage of the proposed solution is that the design-for-reliability hardware uses only regular memristors in the memristor layer plus some CMOS circuitry that can be implemented outside the memristor arrays. Hence, unlike other methods which require integration of transistors to decouple memristors, our solution maintains array regularity and will not suffer from technology scaling issues.

Our case study shows that the performance overheads of the proposed reliable read and write operations, are 8% and 0.1% respectively and the energy overheads are 0.5% and 38% respectively in comparison with the baseline, unreliable implementation. This should be affordable due to the ultra-low-power characteristics of the memristive memories.

Chapter 3

A Low-Power Variation-Aware Adaptive Write Scheme for Access-Transistor-Free Memristive Memory

Recent advances in access-transistor-free memristive crossbars have demonstrated the potential of memristor arrays as high-density and ultra-low-power memory. However, with considerable variations in the write-time characteristics of individual memristors, conventional fixed-pulse write schemes cannot guarantee reliable completion of the write operations and waste significant amount of energy.

We propose an adaptive write scheme that adaptively adjusts the write pulses to address such variations in memristive arrays, resulting in $7\times$-$11\times$ average energy saving in our case studies. Our scheme embeds an online monitor to detect the completion of a write operation and takes into account the parasitic effect of line-shared devices in access-transistor-free crossbars. This feature also helps shorten the test time of memory march algorithms by eliminating the need of a verifying read right after a write, which is commonly employed in the test sequences of march algorithms.

3.1 Introduction

Traditional memory technologies cannot keep up with the ever-increasing demand for denser, faster, and lower-power memories. CMOS technology scaling is increasing the leakage current in transistors thus making the CMOS-based memory chips even more power hungry while the yield is dropping due to fabrication imprecision [1]. Various emerging resistive memory technologies (such as Phase Change Memories [2] and Spin-Transfer Torque Magneto-resistive Memories [3]) have been investigated to replace the conventional CMOS-based memories. However, the requirement of having an access-transistor per memory cell limits the overall power reduction and shrinkage of the memory cell size. To address these limitations, Metal oxide valence change ReRAMs [4], generally referred to as memristors [5], have been investigated extensively. They can be used to implement high-endurance non-volatile memories with a fast switching speed [24] without requiring an access-transistor for each memory cell [11].

A memristor is a two-terminal passive programmable resistor that maintains its resistance in the absence of an electric field. Hence, it is an excellent candidate as a low-power non-volatile memory. High/low resistances can be used to represent logic 0/1. The resistance of the device can be changed by applying adequate voltage/current pulses. The change to the resistance has a strong non-linear dependency on the amplitude and the duration of the applied pulse. This non-linearity opens up opportunities to obviate the need for an access-transistor for each memory cell. The elimination of the access-transistor and the simple structure of the memristors make it possible to shrink its feature size to a sub-10nm scale [8] for implementing ultra-high density memory arrays. Furthermore, both estimations and preliminary experimental measurements indicate considerable potential of such memristive memories consuming significantly lower power than existing technologies [9, 10, 49, 50] due to the passiveness of the memory cell and the access-transistor-free memory structure. These characteristics make memristive memories attractive as an extremely dense and low-power non-volatile memory [6]. Several nanoscale

access-transistor-free memristive crossbars have been demonstrated recently [11, 51, 13].

One major obstacle before the potential commercialization of these devices is the significant variation in their temporal characteristics for the write operation (i.e. write time) [52]. Such variations exist both between different devices and between different operation cycles of the same device. This can be a major challenge for memory applications in which a very low failure rate (e.g. lower than 10^{-12} [1]) is required. Hence, write pulses with a long duration (longer than necessary for the majority of the memory cells) are used in order to guarantee a sufficiently high completion rate for the write operations which results in considerable waste of power. Moreover, such long pulses still cannot guarantee a correct write operation due to significant temporal variations and a non-trivial probability for the existence of ultra-slow devices/write cycles. This necessitates a read-after-write operation to verify the correctness of the write operation which further grows the power figure and degrades the performance.

In this chapter, we propose a low-power adaptive write scheme to address write-time variation of memristive devices in access-transistor-free crossbars. The proposed scheme addresses the temporal variation of memristors by dynamically adjusting the duration of write pulses for individual devices. Our method monitors (i.e. reads) the resistance value of the target cell *during* the write operation while filtering the data-dependent parasitic effect of the neighboring cells, and terminates the write pulse, as soon as a desired resistance value is reached. This reduces the energy consumption of our method over the conventional, fixed-length write-pulse method. We evaluate its power saving based on SPICE-level circuit simulation. We use the measurements made on our experimental devices [16] to derive the parameters used in the simulation to ensure the accuracy of the results. The embedded read operation that monitors the state of the device also offers the advantage of verifying the successful completion of the write operation. This advantage helps reduce the test time of most march algorithms for testing memories as the self-verification capability of our adaptive write operation eliminates the need of a read operation in some march elements of the test algorithms.

The rest of the chapter is organized as follows: Section 3.2 provides the necessary background on memristors and Section 3.3 elaborates on the write-time variation issue in memristive devices. The proposed method is described in details in Section 3.4. Section 3.5 provides experimental and simulation results to evaluate the proposed scheme and validates the energy saving and test time improvement of the method. Section 3.6 concludes the chapter.

3.2 Background on Memristors

Memristors typically have a metal/insulator/metal (MIM) structure. The change in the resistance happens due to the non-volatile formation of a conductive filament inside the insulator layer. Such filament is formed by applying a voltage/current pulse across the device. The applied electric field mobilizes the conductive particles (e.g. oxygen vacancies, metallic ions, etc.) to form a filament by making them drift inside the insulator layer [53]. With the formation of such highly conductive channel, the device goes into a low resistance state (ON state). To program the device back to a high resistance state (OFF state), a pulse with an opposite voltage polarity is applied. This will disperse the conductive particles and rupture the filament. Figures 3.1a and c show one possible realization of memristors and the filament formation and rupture processes.

Typical memristive devices exhibit a non-linear behavior in their I-V characteristics and in the rate of the change in their resistance values based on the applied voltage (as shown in Figure 3.1b) [24, 7]. Utilizing these non-linearities, access-transistor-free crossbar architectures are proposed in [51, 13] to implement memristive memory arrays. Such crossbars consist of two perpendicular layers of parallel nanowires forming a memristor at each cross section (Figure 3.2a). It is worth noting that while individual memory cells do not need an access-transistor, a select-transistor is still needed per crossbar nanowire. Hence, this structure is referred to as *1TnR*, where *1T* refers to the line-select transistor of the nanowires, which is

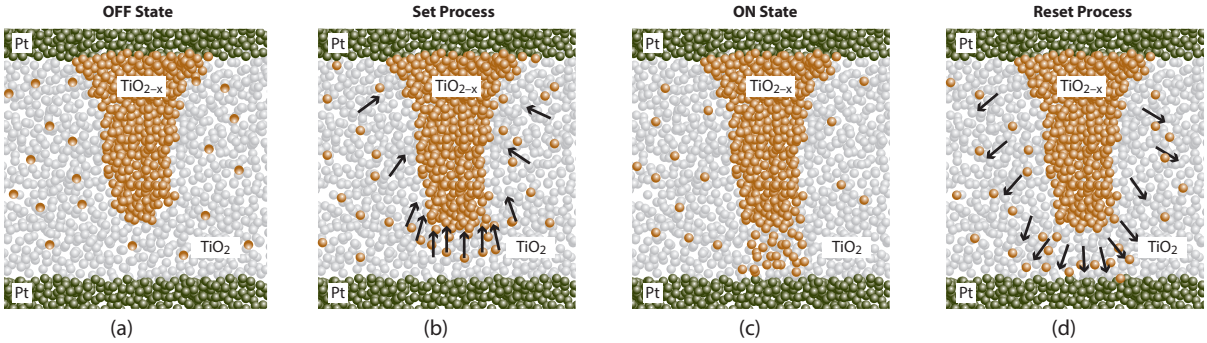


Figure 3.1: A memristor's sample realization and I-V characteristics. a & c) Applying a negative/positive write voltage to the device can destroy/form a low-resistance filament, by dispersing/attracting conductive particles, thus writing logic 0/1 to the cell. b) The non-linear behavior of a typical memristive device. The solid line shows the non-linearity in the I-V characteristics, while the dashed curve illustrates the non-linearity in the rate of change for the resistance of the device based on the applied voltage. Three regions can be observed. In the Diode Voltage Region (DVR), the device acts like a diode and the applied voltage pulse results in negligible currents and does not change the state of the device. This region is small in typical memristive devices. In the Read Voltage Region (RVR), the device acts like a fixed-value resistor: the resulting current depends on the state (resistance) of the device, and operating a device in this region has a negligible effect on the state/resistance. In the Write Voltage Region (WVR), higher currents result in exponentially higher altering effect on the resistance of the memristor, thus effectively switching the device from an ON state to an OFF state or vice versa based on the applied polarity.

shared among n access-transistor-free memristive devices on that nanowire (nR).

To read a single cell in a 1TnR structure, a read voltage pulse of amplitude V_r (in the Read Voltage Region shown in Figure 3.1b) is applied to the target cell, by applying $V_r/2$ on the target word-line (horizontal line), $-V_r/2$ on the target bit-line (vertical line), and grounding other lines as shown in Figure 3.2a. This method is known as the *all-holding* V/2 scheme [44], since all the lines are held at a known voltage value. The resulting current on the bit-line is then sensed by a sense circuit to identify the state of the target cell. In this scheme, the V_r pulse applied to the target cell generates distinct current values for the two possible states of the target cell, while other devices sharing the same bit- or word-line, only experience half of V_r which, due to the strong non-linearity of the memristors' I-V curves, results in very low leakage currents (i.e.

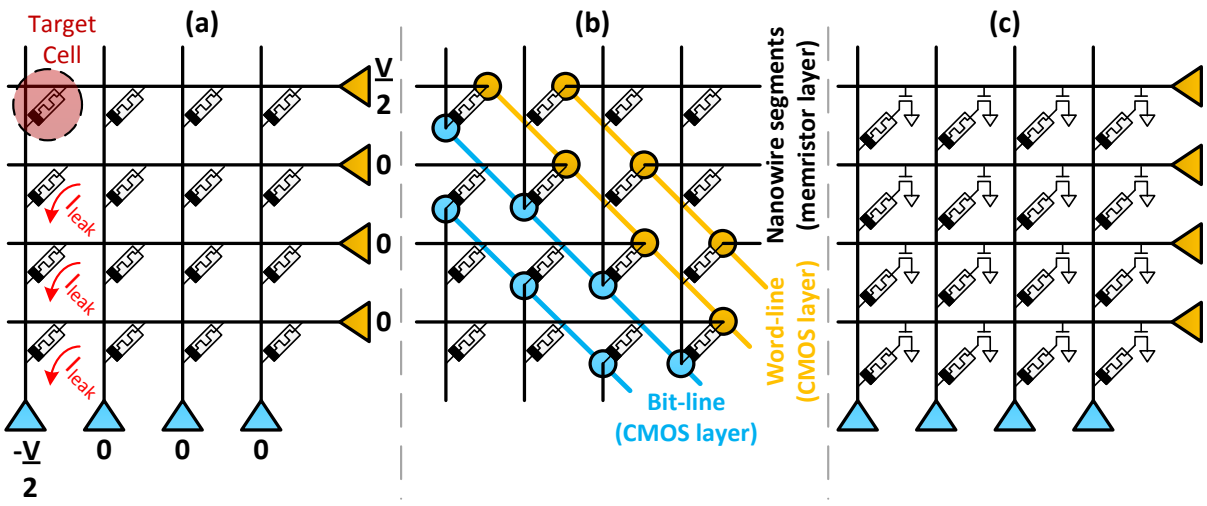


Figure 3.2: Different memory architectures. a) 1TnR: showing a V/2 voltage application scheme to access the target cell. I_{leak} is due to half-selected devices on the bit-line. The select-transistors of the bit- and word-lines are not shown. b) *CMOL*: nanowires in memristor layer are segmented and accessed via an area-distributed interface from the underneath CMOS chip, that are connected to bit- and word-lines in the CMOS layer. c) 1T1R: each memristive device requires an access-transistor.

I_{leak} in Figure 3.2a). Such low leakage currents generally would not cause errors in read if the sensing circuitry is designed properly.

Similarly for the write operation, a proper write voltage pulse of amplitude V_w (within the Write Voltage Region in Figure 3.1b) and width t_{write} is applied across the device following the same V/2 scheme. The write pulse effectively changes the target cell's state, while the states of the half-selected devices which experience only $V_w/2$ remain unchanged as this partial voltage is not strong enough to change a cell's state [24]. Note that due to the non-linearity in I-V characteristics of memristive devices, applying write voltages ($V_w > V_r$) may incur larger I_{leak} caused by the line-shared devices. However, this current does not interfere with the write operation as there is no current sensing involved.

It worth mentioning that an *all-floating* flavor of the V/2 accessing scheme exists in which V/2 and $-$V/2 pulses are applied on the target word- and bit-line respectively, while all other

lines are floated [54]. The all-floating method offers lower overall power consumption [44]. However, with an all-floating scheme, all memristive devices in the crossbar will experience a non-deterministic and data-dependent partial voltage bias [55]. During the write operation, this partial bias could become comparable to the write threshold voltage for several non-target devices, resulting in an undesired partial switching in such devices. Moreover, the all-floating scheme is more susceptible to the *sneak path problem* in access-transistor-free crossbars [44]. In this work we employ the all-holding V/2 scheme to avoid such issues.

In the case of large-scale simple 1TnR crossbars, in which each nanowire has many cross-points, the small leakage currents due to half-selected line-shared devices may collectively become too large for the sense circuitry to accurately determine if the target cell is in the ON or OFF state during the read operation. Architecture-level solutions have been proposed to address this issue [13, 43]. The general idea behind these solutions is to limit the number of cross-points per nanowire (i.e. n in 1TnR), while maintaining the high memory-density of a large-scale crossbar.

To this end, Likharev *et al.* proposed an innovative architecture designed for many-layer memristive memories, called "*CMOL*" [43]. In *CMOL*, long crossbar nanowires are segmented to limit the number of crosspoints on each nanowire segment, which effectively limits the level of the aggregated leakage current. Each segment is then accessed via an area-distributed interface from the beneath CMOS circuitry, as illustrated in Figure 3.2b [56]. A realization of *CMOL* was shown in [12], where the number of cross-sections per nanowire is limited to 16. Kawahara *et al.* proposed another architecture-level solution in which a *hierarchical bit-line* structure is employed [13] . In this structure, a long bit-line is segmented into several shorter segments, only one of which can be selected at any given time. Each short bit-line segment is restricted to have only 16 cross-points, which limits the level of leakage current. Note that both *CMOL* and *hierarchical bit-line* architectures can offer high memory bandwidth as they divide a large memristive crossbar into many electrically-decoupled *mini-crossbars* that can be accessed

simultaneously.

The ideas proposed in this chapter are applicable to any 1TnR crossbar architecture with an all-holding voltage application scheme. Nevertheless, due to the limited scalability of the general 1TnR, 1TnR architectures with a limited n, such as *CMOL* and *hierarchical bit-line*, are better and more practical platforms for implementation of the proposed ideas. However, describing the ideas on such architectures requires an in-depth explanation of their structure which is out of the scope of this thesis. Hence, without losing generality, we use the simple 1TnR crossbar architecture to explain the proposed method. In our explanation, the number of cross-points per nanowire is limited to mimic 1TnR architectures with a limited n, such as *CMOL* and *hierarchical bit-line*.

3.3 Write Time Variation in Memristors

One major challenge of memristive devices is the substantial spatiotemporal randomness experienced during the write operation [52]. Device to device (or spatial) variation is caused by issues like line edge roughness and film thickness irregularity, that can be ameliorated through improved fabrication processes. The more serious problem is the substantial temporal randomness experienced during the normal operation. This randomness is due to the filamentary nature of resistive switching. Filament formation involves physical processes like oxidation, ion transport and reduction, all of which are thermodynamically driven [57] and require overcoming specific activation energies. Typically one of the processes is rate-limiting so that switching is associated with thermal activation over a dominant energy barrier and is thus probabilistic. As a result, even for the same filament in the same device, the write time (aka wait time, that is the time delay between application of a write pulse and the toggling of the device state) has significant variations.

It is shown in [57] that if only one dominant energy barrier limits the switching process, the

write time is expected to follow an exponential distribution and the probability that a switching event occurs within Δt at a given time t is given by:

$$P_{Switch(t \to t+\Delta t)} = \frac{\Delta t}{\tau} \cdot e^{(-t/\tau)} \tag{3.1}$$

where τ is the average write time of the memristive device. This shows that while individual write times are random, their overall statistical distribution is not completely random and can be captured well mathematically. This insight provides an important tool at the architecture level to design variation-aware circuits.

Several methods have been proposed to address the variation in temporal switching characteristics of memristive devices. A common method is to apply a long write pulse to switch the slowest devices, ensuring a high "*success rate*" for the write operation. However, such long pulses are not necessary for fast devices and waste power. Moreover, even after applying a long pulse, it is still necessary to verify the success of the operation, due to a non-zero probability of having ultra-slow write cycles or devices. In an attempt to avoid over-application of write pulses, Alibart *et al.* proposed the application of multiple short write pulses, each followed by a read operation, for quicker detection of the completion of the switching [58]. However, this method suffers from the power and performance overheads incurred by the additional read operations. Another scheme was proposed by Yi *et al.* which measures the current during the write operation and compares it with a fixed reference to detect the completion of the switching [59]. However, this method is only applicable to a single device but not to memory arrays as it does not consider the considerable parasitic current (I_{leak}) during the write operation due to the line-shared devices in a 1TnR crossbar (Figure 3.2a). The I_{leak} value is not known beforehand and depends on the data pattern stored in the line-shared devices: The more devices at the low resistance ON state, the higher the aggregated leakage current. Hence, even for a moderate number of line-shared devices (e.g. 16), the parasitic I_{leak} can range from a negligible amount to as high as 8X the

Table 3.1: Variation-aware write schemes

Method	1TnR Crossbar Compatible	Fast	Power Efficient	Verified Write
Fixed-Length Pulse (FLP) [62]	✓	✗	✗	✗
High Precision Tuning [58]	✓	✗	✗	✓
Adaptive Single Cell [59]	✗	✓	✓	✓
Adaptive 1T1R [60]	✗	✓	✗	✓
Proposed Low-Power Variation-Aware (LPVA)	✓	✓	✓	✓

current drawn by the target cell (I_{target}). Typical sensing circuitry to monitor the state of a device has sensitive noise margins and does not work under such severe noisy conditions. Jo *et al.* [60] propose another adaptive scheme in which an access-transistor is added to each memory cell (aka 1T1R architecture as shown in Figure 3.2c) to filter the parasitic effect of line-shared memristors (I_{leak}). However, this method suffers from the intrinsic technology scaling issues of CMOS-based memories as the access-transistor limits the reduction of the memory cell's footprint and incurs static power consumption at all times.

Our proposed method resolves the shortcomings of the previous methods by utilizing (1) a monitor-while-write scheme that can detect the switching event to terminate the write pulse and verify the success of the write operation, and (2) a leakage current filtering scheme that is engineered for 1TnR crossbar arrays to take care of the parasitic effect of the line-shared devices. Moreover, the monitor-while-write scheme can directly benefit memory testing: The test time of most march algorithms [61] can be reduced as the read operation following a write for verification of successful write can be eliminated. To the best of our knowledge, our work proposes the first adaptive write scheme engineered for the I_{leak}-prone access-transistor-free crossbars. Table 3.1 summarizes the features of the existing schemes and the proposed method.

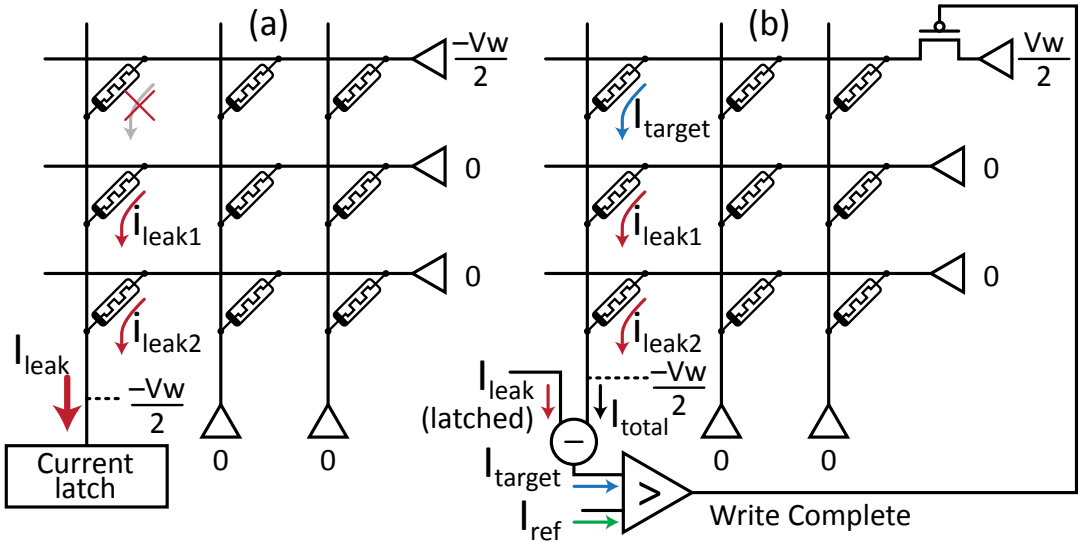


Figure 3.3: The proposed method. a) step 1: I_{leak} is latched b) step 2: I_{target} is measured by subtracting the latched I_{leak} from I_{total} and is then compared with I_{ref}.

3.4 Low-Power Variation-Aware Writing Scheme

Our *write-time-variation*-aware writing scheme for 1TnR architectures adapts the length of the write pulse for individual memristive devices. The major challenge of this solution is to filter the I_{leak} caused by the bit-line-shared devices during the write operation, so that only the current drawn by the target device is extracted to detect the completion of the switching. Our proposed method consists of two consecutive steps as illustrated in Figure 3.3:

Step 1: In the first step, the I_{leak} is latched. Both word- and bit-lines of the target cell are held at $-V_w/2$ while all other bit- and word-lines are grounded (Figure 3.3a). With this configuration, the current flowing through the bit-line will be only due to the bit-line shared devices and it will be identical to the I_{leak} incurred during the write operation. This is due to the fact that in both steps, the line-shared devices (1) experience the same voltage pattern, and (2) maintain the same data-pattern. This current is latched for later processing.

Step 2: Next, the write voltage V_w is applied to the target device using the V/2 scheme (Figure 3.3b). In this case, the current flowing through the bit-line (I_{total}) is equal to $I_{target} + I_{leak}$.

Since I_{leak} is previously recorded, it can now be subtracted from the total current to obtain I_{target}. I_{target} is then compared against a reference current (I_{ref}) to determine if the target memristor has reached the desired value, i.e. $I_{target} > I_{ref}$ for OFF→ON switching, and $I_{target} < I_{ref}$ for ON→OFF switching. Hence, the I_{ref}'s value is chosen in between the expected current levels of a target cell in the ON state (I_{ON}) and that in the OFF state (I_{OFF}). The same I_{ref} is applied to all memristors since the ranges of possible ON resistances for different devices is generally separated by orders of magnitude difference from that of OFF resistances [11]. The "*Write Complete*" output of the comparator is used to terminate the write pulse and indicate the success of the write operation.

3.4.1 Circuit Implementation Details

Figure 3.4 shows a simplified circuit-level implementation of the proposed write scheme. The op-amp and the N1 transistor are used to accurately maintain the voltage on the bit-line at a desired value (V_{bias}) to ensure that a right voltage level is applied to the device.

In the first step, the current flowing through the bit-line (I_{BL}) is I_{leak} which is latched for further use in step 2. To latch this current, first it is mirrored twice on Lines X and then Y, using NMOS (N2-N3) and PMOS (P1-P2) current mirrors, when the transmission gate T1 is open. In our implementation, current mirrors are realized using cascode current mirrors [63] for increased mirroring accuracy. "Current latching" works by converting the mirrored I_{leak} to a voltage (i.e. the common gate voltage of P1 and P2) and latch that voltage in a capacitor (C1) by closing T1 after a latching period t_L. Since current mirrors operate based on the application of the same gate voltages to similar transistors, by later applying the latched voltage in C1 to the gate of P2, the same I_{leak} current can be reproduced. The latching time t_L mainly depends on the C1's charge time and is often negligible. C1's capacitance variations are typically small [64], hence, conservative selection of t_L value could render the circuitry insensitive to such variations.

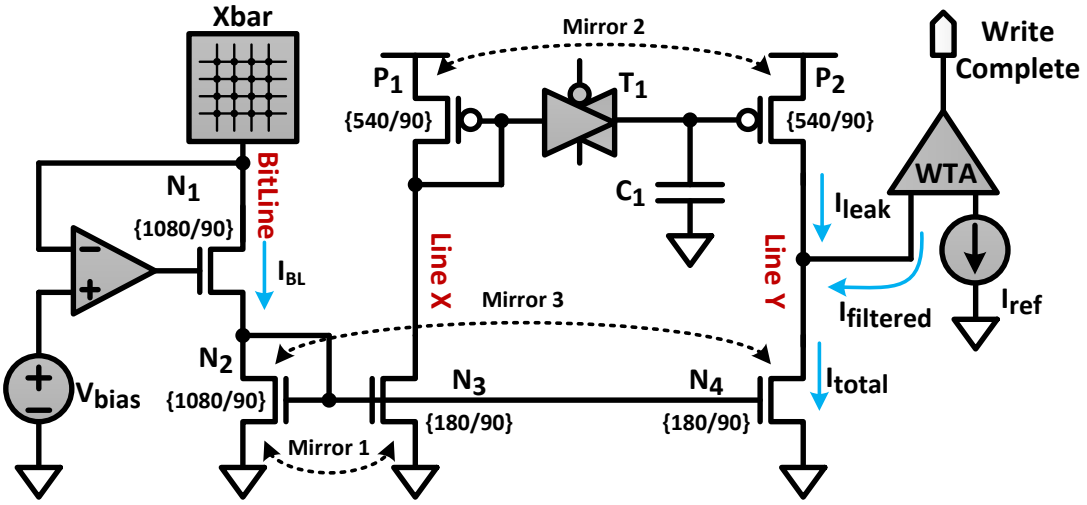
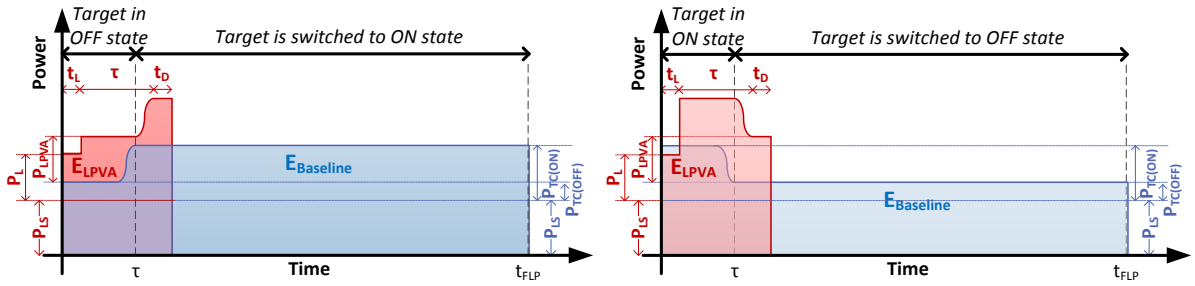


Figure 3.4: The circuit implementation. The size of each transistor is noted in nanometer ({Width/Length}). The currents shown are according to the second step of the method: the latched I_{leak} is subtracted from the total current on the bit-line. The resulting $I_{filtered}$, that is ideally equal to the current flowing through the target cell, is compared with I_{ref} to read the target cell.

In the second step, I_{total} (i.e. $I_{leak} + I_{target}$) flows through the bit-line. This current is mirrored on Line Y, using the current mirror 3 (N2-N4). Meanwhile, P2 is reproducing the I_{leak} on the same line (i.e. Line Y) by having the voltage latched in C1 applied on its gate. Thus, the current extracted from the comparator, $I_{filtered}$, is the difference of I_{leak} and I_{total}, which gives the desired I_{target} that flows through the target cell. A winner-take-all (WTA) design is then used to compare the $I_{filtered}$ (i.e. equal to I_{target}) with a reference current I_{ref} to read the target cell's value and thus detect the switching event after a detection time t_D. The reference current can be accurately generated by pinning the voltage across a DAC controlled resistor. The detection time depends on the response time of the WTA.

It should be noted that the $I_{filtered}$ observed by the WTA comparator might have some deviations from the actual I_{target}. While layout-variation-induced deviations can be ameliorated using an accurate common-centroid layout method [65], there are other reasons of deviation that should be considered: (1) non-ideal mirroring because of unequal drain voltages in current



(a) OFF→ON switching (b) ON→OFF switching

Figure 3.5: The energy trade-off during a) OFF→ON and b) ON→OFF switching. The blue and red regions illustrate the power-vs-time of the baseline FLP scheme and the proposed LPVA scheme respectively. OFF→ON switching offers better energy saving as the extra energy consumption of the FLP baseline, i.e. for $t > \tau$, is larger when the target memristor is in the ON state.

mirror's transistors, (2) switching noise on the voltage stored in C1 incurred by the closure of the transmission gate T1, and (3) C1 voltage drop during the write operation due to C1's leakage. While such deviations are small in general ($\approx$1% of total I_{leak}), they should be considered to ensure that they do not violate the necessary margins of the comparator. The SPICE-level simulation results in Section 3.5.2 illustrate the effect of such non-idealities and verify the correct functionality of the proposed circuit in the presence of the worst-case deviations.

The structure in Figure 3.4 is designed to detect ON values in the target cell (i.e. OFF→ON switching). Similar circuitry with a loser-take-all comparator can be used to detect the OFF values.

3.4.2 Energy Trade-offs

Figure 3.5 shows the energy trade-off of our low-power variation-aware (LPVA) method versus the baseline fixed-length-pulse (FLP) write method for both OFF→ON and ON→OFF transitions. The baseline FLP applies V_w pulse for a long period, $t_{FLP} = \alpha \times \tau$, where τ is the average write time, and α depends on the required success rate (e.g. $\alpha = 14$ to ensure that the

probability of an unsuccessful write at the end of the write period is less than 10^{-6}, assuming that the write times of devices follow an exponential distribution). In contrast, our method terminates the write pulse right after switching, which on average happens at $t = \tau$. In order to avoid excessively long write pulses in our method in case of very slow write cycles or devices, we set an upper bound on the duration of the write pulse to be equal to t_{FLP}, similar to the FLP method. Hence, as both methods have the same maximum write pulse duration, they both yield the same "*success rate*" for the write operations.

As a result of the earlier termination of the write pulse in our method, much less energy is consumed inside the memristive crossbar. Moreover, unlike FLP, our method reveals if a write operation is successful or not, indicated by the "Write Complete" signal, as shown in Figure 3.4. On the other hand, the extra circuitry required for the proposed method consumes some power (P_{LPVA}) which is not needed for the baseline FLP. Extra times for latching and detection (t_L and t_D) are also required for the proposed method, that contribute to the overall energy consumption, which are not needed for the FLP.

In Section 3.5.3, we will evaluate the merits of the proposed method in terms of energy saving by considering different data patterns on the line-shared devices and for both ON→OFF and OFF→ON transitions.

3.5 Results

To assess the amount of energy saving, we first verified, by taking measurements of our experimental devices, that if the write time of fabricated memristive devices indeed follows an exponential distribution. Next, SPICE-level simulation was performed based on the parameters extracted from measurements on fabricated devices, to validate the functionality and to evaluate the energy consumption of the proposed method. Finally, the impact of the proposed method on test time reduction of memory testing algorithms is discussed.

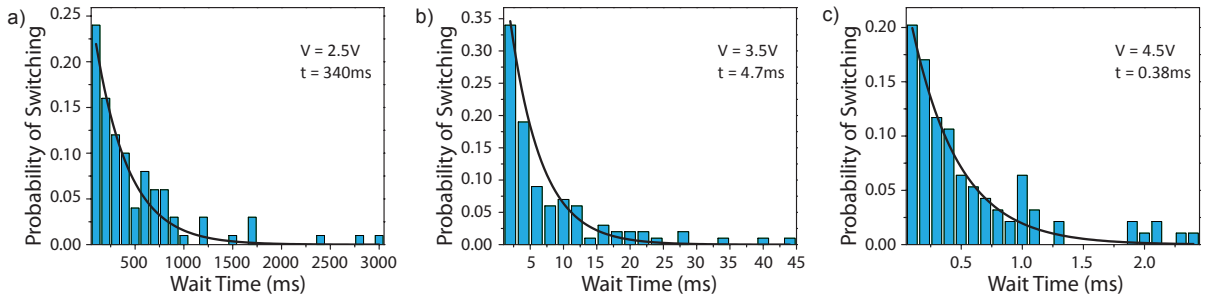


Figure 3.6: The write time distribution for Ag/a-Si/p-Si devices at different write voltages, that follows the predicted exponential distribution: a) 2.5V, b) 3.5V, c) 4.5V [16].

3.5.1 Write Time Variation Measurements

Our recent study on write time variation [16] shows that the switching probability of memristive devices has a good fit to Equation 3.1. In this study, to achieve accurate and robust measurement of the write times, single Ag/a-Si/p-Si devices are particularly engineered to have longer write times for this experiment. The write time distribution is obtained at different write voltages (2.5V, 3.5V, and 4.5V). It is observed that while the average write time (τ) decreases exponentially with higher write voltages, the distribution preserves the predicted exponential behavior at all write voltages, as shown in Figure 3.6.

Individual write times were obtained by resetting the device to the high-resistance OFF state and applying a constant DC bias of V_w to the device. Meanwhile, the current through the device is continuously monitored to detect the sharp increase in the current that indicates the device's successful transition to the ON state. Once the device was switched to ON, the device was reset to the same OFF state again and the measurement was repeated 100 times.

3.5.2 Electrical Model and SPICE-Level Simulation Results

Both baseline FLP and the LPVA write circuitry were implemented in Cadence Virtuoso tool using a 90nm IBM CMOS process. They are simulated along with an electrical model

Table 3.2: Electrical parameters of the 1TnR crossbar

Param.	**Value**	**Param.**	**Value**	**Param.**	**Value**
V_w	$2.2V$	R_{ON}	$200K\Omega$	R_{nw}	$0.02\Omega/nm$
τ	$50ns$	R_{OFF}	$100M-1G\Omega$	C_{nw}	$1.2aF/nm$

of the 1TnR crossbar-based memristive memory. The crossbar is modeled based on electrical characteristics of the cross-points (memristors) and the nanowires in the memristive layer [15]. We used Ag/a-Si/SiGe/W devices [11] to derive the electrical characteristics of the memristors (i.e. R_{ON}, R_{OFF}, average write time, etc.). These devices offer better electrical characteristics (such as higher Ron values, faster switching speed, lower write voltage, etc.) compared to the Ag/a-Si/p-Si devices and are thus more suitable for memory applications, while they are expected to preserve the exponential write-time distribution due to the same underlying "dominant-filament" mechanism. The crossbar nanowires are modeled based on their intrinsic resistance (R_{nw}) and capacitance (C_{nw}) that are extracted according to their physical characteristics. Table 3.2 summarizes the derived circuit parameters that are used to model the crossbar and memristive devices [15, 11].

A 1TnR crossbar architecture with 16 memristors per nanowire is modeled and simulated as a case study to evaluate the proposed method. A moderate number of cross-points is used to resemble limited-n 1TnR architectures, such as *CMOL* [43] and *hierarchical bit-line* [13] discussed in Section 3.2.

Figure 3.7 shows the SPICE simulation results for the $I_{filtered}$ input of the WTA comparator during an OFF→ON switching. The simulation is based on the circuit parameters listed in Table 3.2. The result is shown for two different cases in which all the bit-line-shared devices are in ON and OFF states, which produce the largest and the smallest I_{leak} respectively. Ideally the comparator's $I_{filtered}$ input should be exactly equal to the I_{target}, the current flowing through the target cell: During the latching phase (step 1), I_{target} is equal to zero, and in the writing phase

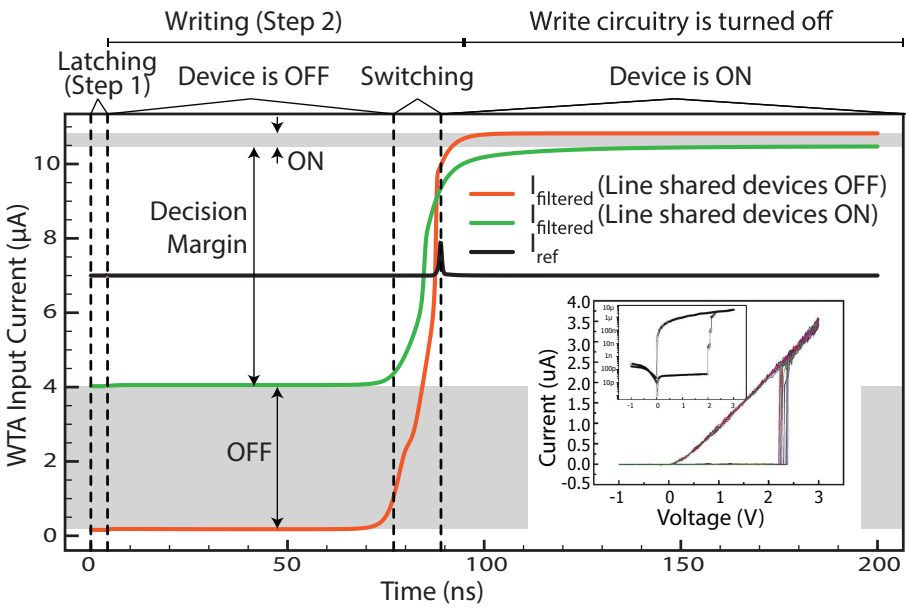


Figure 3.7: $I_{filtered}$ values in different steps of the method and for different data patterns on the line-shared devices. While the I_{ON} and I_{OFF} currents passing through the target cell are apart by more than three orders of magnitude, as shown in the inset, the I_{ON}/I_{OFF} ratio observed at the input of the comparator is much smaller due to the current latching and mirroring errors. Nevertheless, a wide decision margin exists to detect the switching of the device. The inset illustrates the typical I-V characteristics of an Ag/a-Si/SiGe/W device [11]. Note that higher I_{ON} values can be achieved by adjusting the current compliance.

(step 2), it is equal to I_{OFF}/I_{ON}, before/after switching. However, it is observed that the $I_{filtered}$ input of the comparator may deviate from the I_{target} values resulting in a "range" of currents, representing the target cell in the ON or the OFF state. This is anticipated and is due to limited accuracy in the mirroring operation and dependency on the stored data pattern as discussed in Section 3.4.1.

$I_{filtered}$ has greater deviation when the target cell is in the OFF state (i.e. $I_{filtered} < I_{ref}$): considering the schematic shown in Figure 3.4, the comparator tries to pull up the drain of N4, which results in higher drain-voltage mismatch between N2 and N4 transistors, and thus reduces the accuracy of the mirroring. After the switching, the comparator drops the drain voltage of N4, which reduces the drain-voltage mismatch between N2 and N4, and leads into a more accurate mirroring.

Despite the deviation of $I_{filtered}$ that the comparator experiences, there still exists a wide "decision margin" between the ranges of $I_{filtered}$ current resulted by a target cell in the OFF or ON state. This confirms that our leakage-current-filtering scheme can effectively filter I_{leak} and the WTA comparator can detect the switching by comparing the observed $I_{filtered}$ with a reference current I_{ref}, in spite of possible deviations of $I_{filtered}$.

In order to maximize energy saving, the currents involved in the latching and monitoring circuitry are scaled down by a factor β. This is done by adjusting the transistor channel width in the transistor pairs of the NMOS current mirrors, as shown in Figure 3.4. This scales down the $I_{filtered}$ input to the comparator and thus, the I_{ref} is adjusted accordingly. An optimized sizing factor $\beta = 6$ is found experimentally which provides maximum energy saving while providing a noise margin greater than $1\mu A$.

Table 3.3 lists the power and timing numbers acquired from SPICE simulation. P_L and P_{LPVA} indicate the power consumption of the extra write circuitry during the latching phase (i.e. step 1) and the writing phase (i.e. step 2) respectively. P_{LS} and P_{TC} represent the power consumption of the line-shared devices and the target cell respectively. Note that these power numbers depend on the data patterns stored in the line-shared devices: With more devices in the ON state, greater current flows through line-shared devices and the proposed write circuitry, resulting in higher power consumption. Hence, SPICE simulation is performed for all possible data configurations, resulting in a *range* of numbers for each power figure. Unlike the power figures, the latching and detection times (t_L and t_D) show negligible data-dependency.

Table 3.3: Energy/timing figures of the proposed LPVA method

Parameter	**Value**	**Parameter**	**Value**
t_L	1ns	t_D	3ns
P_L	$\{28.6\text{-}228\}\mu W$	P_{LPVA}	$\{42.8\text{-}243\}\mu W$
$P_{TC}^{OFF/ON}$	0.045 / $23.9\mu W$	P_{LS}	$\{0.36\text{-}182\}\mu W$

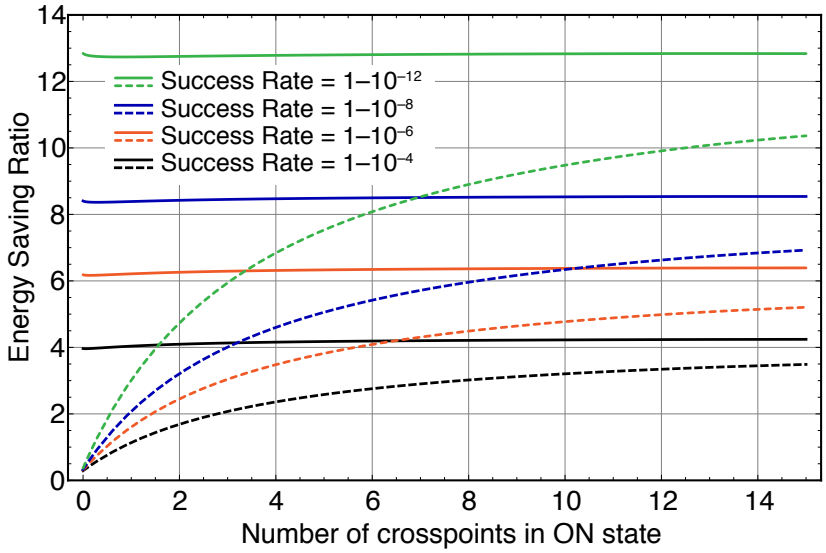


Figure 3.8: Energy saving vs. the number of low-resistance (ON) line-shared devices for various success rates of write operation. Solid and dashed lines are for OFF→ON and ON→OFF transitions in the target cell respectively. OFF→ON yields higher energy saving ratios.

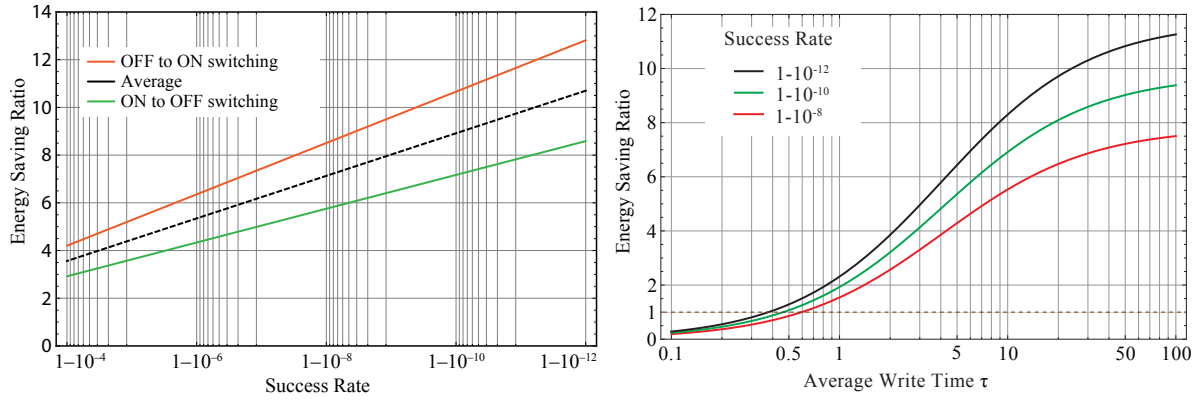
3.5.3 Energy Saving

Equation 3.2 calculates the average energy saving ratio that the proposed method can achieve over the FLP baseline method for an OFF→ON transition:

$$\frac{\tau \cdot P_{Xbar}^{OFF} + (t_{FLP} - \tau) \cdot P_{Xbar}^{ON}}{t_L \cdot (P_L + P_{Xbar}^{OFF}) + \tau \cdot (P_{Xbar}^{OFF} + P_{LPVA}) + t_D \cdot (P_{Xbar}^{ON} + P_{LPVA})} \tag{3.2}$$

where P_{Xbar} is the total power consumed in crossbar due to the line-shared devices (P_{LS}) and the target cell (P_{TC}). $P_{TC}^{ON/OFF}$ and $P_{Xbar}^{ON/OFF}$ show the power consumption of the target cell and the whole crossbar respectively, when the target cell is in the ON/OFF state. Other variables in the equation have been defined in Section 3.5.2. The equation is derived based on the energy trade-offs shown in Figure 3.5(a).

Figure 3.8 illustrates the dependency of the energy saving figures on the data pattern stored in the bit-line-shared devices for both ON→OFF and OFF→ON transitions. The OFF→ON



(a) Energy saving vs success rate

(b) Energy saving vs average write time

Figure 3.9: The energy saving offered by the method. a) Weighted average of energy saving vs. success rate of write operation. The plot illustrates the energy saving for both ON→OFF and OFF→ON transitions as well as the average case. b) Average energy saving vs. the average write time of the memristive device for different required success rates.

transition (solid lines) results in greater energy saving: the energy consumption of FLP is significantly higher in this case, as it applies a long write pulse to a target cell which is switched to and remains in the more power-consuming ON state (i.e. low-resistance state) for most of the write cycle.

For ON→OFF switching (dashed lines), only for the case where all the line-shared memristors are in the OFF state, FLP can do slightly better than our method since (1) line-shared devices are all OFF thus consuming little power, (2) similarly the target cell is in the OFF state during most of the t_{FLP}, while (3) the WTA comparator in the proposed method incurs some energy overhead which is greater than the energy consumption of FLP in that particular data configuration with ultra-low energy consumption. However, with more devices in the ON state, the increase in energy consumption of the FLP quickly surpasses the overhead of the proposed method.

Different line colors in Figure 3.8 show the effect of the required success rate of the write operation on the energy saving ratio: a higher required rate for successful write yields a higher energy saving ratio. This is because a higher desired success rate demands a longer worst-case

write pulse, and thus greater energy consumption, for the FLP method, while the average write time and thus the average energy consumption of the proposed method is constant regardless of the required success rate. Note that in our adaptive method the write pulse is terminated immediately after the completion of the switching, which on average occurs at τ. Given a desired success rate, the corresponding pulse length for the FLP method can be derived based on the cumulative switching probability of Equation 3.1:

$$t_{FLP} = -\tau \cdot \ln(1 - \textit{Success Rate}) \tag{3.3}$$

To characterize the average energy saving figure, we assume a uniform probability for all possible data patterns stored in the line-shared devices and then evaluate the weighted average of the energy saving. The weighted average considers the distinct probability of having different number of line-shared devices in the ON state:

$$\textit{Weighted Average} = 2^{-(Xpt-1)} \cdot \sum_{i=0}^{Xpt-1} \binom{Xpt-1}{i} \cdot ES_i \tag{3.4}$$

where ES_i represents the energy saving ratio while having i line-shared cross-points in the ON state, and Xpt stands for the total number of cross-points on the line. Figure 3.9a shows the weighted average of energy saving versus the desired success rate. Different lines show the trends for both ON→OFF and OFF→ON transitions, as well as the average case. It can be observed that with a reasonable requirement that the probability of an unsuccessful write operation at the end of the write period should be less than 10^{-8}, our method offers an average of $\approx 7\times$ saving in energy consumption. Note that the actual energy saving could be even more significant: Equation 3.2 does not consider the power consumption of CMOS interconnects, P_{IC}, which is highly dependent on the memory size and structure. By taking P_{IC} into account, the instantaneous power consumption of both FLP and LPVA methods are increased by the

same P_{IC} amount, decreasing the $\frac{P_{LPVA}}{P_{FLP}}$ ratio. In such a case, the LPVA's extra write circuitry introduces lower power overhead relative to the baseline FLP. Hence, the faster termination of the write pulse in the LPVA write-scheme leads to better energy saving figures.

3.5.4 Discussion on the Effect of Circuit and Device Parameters

While the proposed method is tailored for Ag/a-Si/SiGe/W devices, it can serve a wide range of memristive devices with different parameters. One requirement to adopt this method is a μA-scale difference between I_{ON} and I_{OFF}. Since the leakage-current filtering scheme is not perfect, in the case of having a memristive device with higher R_{ON} (>MΩ), the deviation in the filtering scheme might become larger than the actual difference between I_{ON} and I_{OFF} values. The R_{ON} is also required to be much larger than the resistance of the nanowire (i.e., R_{nw}) to ensure that the line-shared devices experience the same voltage pattern in both steps. This is true for practical memory applications because R_{ON} has to be larger than a few KΩ to avoid reaching mA-scale currents in the memory module.

In order to ensure very similar leakage current in both steps, the line-shared devices should maintain their resistance when a $V_w/2$ bias is applied across them. Such a partial bias could change the resistance of typical memristive devices, and thus the total leakage current on the bit-line. However, according to [24], the change is less than 0.01% during a write cycle and is thus negligible considering the fact that the decision margin of the proposed circuit is set to account for the presence of $\approx 1\%$ error due to the current latching and mirroring imperfection.

The average write times of memristive devices, τ, can also affect the amount of energy saving of our method. This effect is illustrated in Figure 3.9b for different success rates. As expected, the method offers better energy saving figures when the average write time is much larger than the detection time, i.e., $\tau >> t_D$. However, given a 3ns detection time, it can be observed that even in the case of having memristive devices with $\tau \approx 500ps$, our method still

Table 3.4: Test time improvement to march algorithms

Marching Algorithm	Original Test Len.	Effective Test Len.	Test Time Reduction
March MSL [66]	23n	21n	9%
March B [67]	17n	15n	12%
MATS++ [61]	6n	5n	17%
March-12n [68]	12n	10n	17%
March Y	8n	6n	25%
Marching 1/0	14n	10n	29%
March AB [69]	22n	14n	36%

offers improvements in the energy consumption. Note that faster memristive devices can be supported by decreasing the detection time, that can be achieved by employing smaller CMOS technology nodes, or decreasing the sizing factor β.

3.5.5 Self-Verifying Write Operation to Improve March Algorithms

Another advantage of the proposed method is that it verifies the correctness of each write operation. At the end of the write cycle, the "Write Complete" signal of the comparator shows if the target state has been reached correctly. Hence, the proposed write scheme effectively offers an atomic "write and read" operation, "*wr*". This feature provides a flag indicating erroneous write operations and helps keep track of degraded memory elements. Moreover, it also reduces the test time of most memory testing algorithms in which a read, *r*, is often performed right after a write, *w*, to verify that the correct value is indeed written into the memory cell. For example, in MATS++ [61], the following test sequences are applied to test the memory:

$$\Updownarrow (w0); \Uparrow (r0, w1); \Downarrow (r1, w0, r0) \tag{3.5}$$

With the proposed self-verifying write scheme, the last two operations (i.e. $\{$w0, r0$\}$) are

merged into one operation, effectively reducing the test length from $6n$ to $5n$:

$$\Updownarrow (w0); \Uparrow (r0, w1); \Downarrow (r1, wr0) \tag{3.6}$$

Table 3.4 lists some of the marching algorithms that benefit from the proposed write scheme. Note that the conventional march algorithms can be directly utilized for memory testing of memristive memories since similar fault models (e.g. stuck-at faults, various coupling faults, delay faults, address faults, etc.) are also applicable to such memories [70]. The availability of the atomic "write and read" operation, "*wr*", may also offer opportunities for developing new march algorithms, instead of just directly employing the existing algorithms, for further test quality improvement and test time reduction of such emerging memories.

3.5.6 Performance and Area Overheads

The proposed method offers energy saving advantages and monitoring capabilities without affecting the performance. While individual write operations are completed at different times, and on average they require much shorter write pulses, the write cycle time is still determined by the worst-case write time.

The custom-sized transistors of the extra write circuitry occupy $\approx 55\mu m^2$ of area in the 90 nm technology node, following the standard layout guidelines [71]. Note that one such circuit is required for each bit of data written simultaneously. Hence, given a word-size of 64 bits, our method incurs a total area overhead of $3502\mu m^2$. To evaluate the area overhead, we used NVSim [72] to estimate the area requirements of an exemplar 1Mb 1TnR memristive memory with the same word-size. We configured NVSim to use the same CMOS technology node, device characteristics, and architectural characteristics as those used in our SPICE simulation setup. The overall area reported by NVSim is $128146\mu m^2$, which indicates that our method incurs an area overhead of less than 3%.

Note that our method is addressing the write-time-variation in the access-transistor-free crossbars. Thus, compared to the variation-tolerant schemes that are proposed for "1T1R" arrays in [60], our method saves one access-transistor per memory element which is very significant.

3.6 Conclusion

In this chapter we address the energy consumption and data reliability problems caused by write time variation of memristive devices in emerging access-transistor-free crossbar-based memories. The proposed method applies a write pulse with a just-enough width to switch the target cell's state. The proposed write circuitry can correctly detect switching events in a noise-prone 1TnR memory system by a two-step noise filtering approach: the data-dependent leakage current is first latched, which is then removed from the total current captured in the second step, enabling accurate monitoring. The proposed write scheme automatically verifies the correctness of a write operation as it monitors the target device's state during the write operation. This self-verification feature can be utilized to reduce the memory test time as most march algorithms used for testing memories include test sequences which require a read immediately after a write to verify the correctness of the write.

Our case study demonstrates that the proposed method achieves significant reduction in energy consumption over the conventional fixed-pulse write scheme, in addition to the real-time verification capability. The exact energy saving ratio depends on the data pattern as well as the desired rate of correct write operations. For exemplar success rates at $1-10^{-8}$ and $1-10^{-12}$ combined with random data patterns, our method offers an average of $7\times$ and $11\times$ energy saving respectively.

Chapter 4

In-place Repair for Resistive Memories Utilizing Complementary Resistive Switches

Recent advances in resistive memory technologies have demonstrated their potential to serve as next generation random access memories (RAM) which are fast, low-power, ultra-dense, and non-volatile. However, owing to their stochastic filamentary nature, several sources of hard errors exist that could affect the lifetime of a resistive RAM (ReRAM).

In this chapter, we propose a novel mechanism to protect resistive memories against hard errors through the exploitation of a unique feature of bipolar resistive memory elements. Our solution proposes an unorthodox use of *complementary resistive switches* (a particular implementation of resistive memory elements) to provide an "in-place spare" for each memory cell at negligible extra cost. The in-place spares are then utilized by our repair scheme to extend the lifetime of a resistive memory. Our repair scheme detects data errors during regular memory accesses and triggers repair using the in-place spares at a page-level granularity. We show that in-place spares can be used along with other memory reliability and yield enhancement

solutions, such as error correction codes (ECC) and spare rows.

We develop a statistical model to evaluate our method's effectiveness on extending ReRAM's lifetime. Our analysis shows that the in-place spare scheme can roughly double the lifetime of a ReRAM system. Alternatively, our method can yield the same lifetime as a baseline ReRAM, with either significantly fewer spare rows or a lighter-weight ECC, both of which can save on energy consumption and area.

4.1 Introduction

CMOS-based memory technologies cannot keep up with the ever-increasing demand for denser and lower-power memories, as technology scaling results in increasing leakage and degraded reliability of memory elements. As an alternative, emerging metal-oxide valence-change resistive memory technology, generally referred to as memristors [5], have demonstrated great potential as the next generation random access memories.

A memristor is a two-terminal passive programmable resistor, that maintains its resistance in the absence of an electric field. High/low resistances are used to represent logic value 0/1. Memristors exhibit a set of desirable characteristics, including low-power operation [9], fast switching speed [73], possible elimination of the access-transistor per memory cell [7], and CMOS compatibility [74]. Ultra-high density memristive memory arrays can be realized as memristor's feature size can be shrunk to a sub-10nm scale [8]. Multiple layers of such arrays can be stacked on top of each other to further increase the density [13]. Minor modifications to the device stack, can further provide a double-memristor cell in place of a regular memristor, also known as complementary resistive switches (CRS) [75].

However, different sources of error could affect memristive devices, owing to their stochastic filamentary nature [17]. Physical defects and endurance problems could lead to "hard errors", which are permanent failures of memory cells [76]. This is in contrast to "soft errors", that

are random recoverable errors due to causes such as retention failures [77] or write time variation [52]. Hard errors are commonly addressed by adding redundancy, e.g., by remapping a row with defective bits to a healthy spare row. The existing redundancy-based repair schemes come with the area overhead of the spares, as well as the area and performance overhead of the remapping logic [19]. Alternatively, error correction codes (ECC) have been proposed to detect and correct few erroneous bits, hard or soft, by encoding the data and adding parity bits [18]. However, ECC incurs considerable overhead in terms of area and energy. The energy overhead becomes even more prominent when the ECC is applied to ultra-low-power ReRAMs.

In this chapter we propose a novel use of complementary resistive switches (CRS) to provide a zero-area-overhead *in-place spare* for each bit. A repair mechanism is also presented to activate the in-place spares. The proposed method extends the lifetime of memristive memory modules with minor modifications to the memory architecture. We derive a statistical model to evaluate the effectiveness of the proposed method for lifetime improvement of ReRAMs. Our model incorporates the impact of the ECC and the spare rows on ReRAM's lifetime. We also explore the possibility of using the in-place spares to yield a similar lifetime as a baseline ReRAM, for the objective of minimizing spare rows or using a lighter-weight ECC. We quantify the possible reduction in the area and energy consumption of a ReRAM if the in-place spare scheme is employed.

The rest of the chapter is organized as follows: Section 4.2 covers the necessary backgrounds on memristors. Section 4.3 discusses the origins of errors in memristive devices, as well as the solutions employed in conventional memory technologies. The use of in-place spares is detailed in Section 4.4. Section 4.5 presents a statistical model and evaluates the effectiveness of the proposed method. Section 4.6 concludes the chapter.

4.2 Background

4.2.1 Memristive Devices

A memristor, or a memristive device, is a passive programmable resistor, experimentally found by the HP Labs in 2008 [78]. Memristors are typically fabricated as a stack of thin layer(s) of non-conductive switching oxide, sandwiched between conductive metallic electrodes, as shown in Figure 4.1a.

Most devices need a *forming* step, in which a large *forming* voltage, V_{form}, is applied on the device[79]. The forming step breaks the oxide and migrate a large number of oxygen ions to the cathode, resulting in one or several filaments of oxygen vacancies inside the oxide layer [76]. These oxygen vacancies are conductive. Figure 4.1b shows a memristor after forming.

The resistance of the device can be reversibly switched between a high-resistance OFF state and a low-resistance ON state, by applying negative or positive voltage (current) pulses respectively. Applying a negative pulse mobilizes oxygen ions to recombine with the oxygen vacancies. The recombination partially ruptures the conductive filament and switches the device into an OFF state, as depicted in Figure 4.1c (i.e. a RESET operation). A positive pulse regenerates the oxygen vacancies and rebuilds the filaments (i.e. a SET operation). Memristors typically exhibit a very high OFF to ON resistance ratio [24].

Figure 4.1e shows a typical electrical characteristics of a memristor. Applying write voltages above a memristive write threshold, $\pm V_{thm}$, changes the resistance of the device, while applying smaller voltages has negligible effect on its state [24]. The resistance value of a memristor is read by applying a small read voltage and monitoring the resulting current.

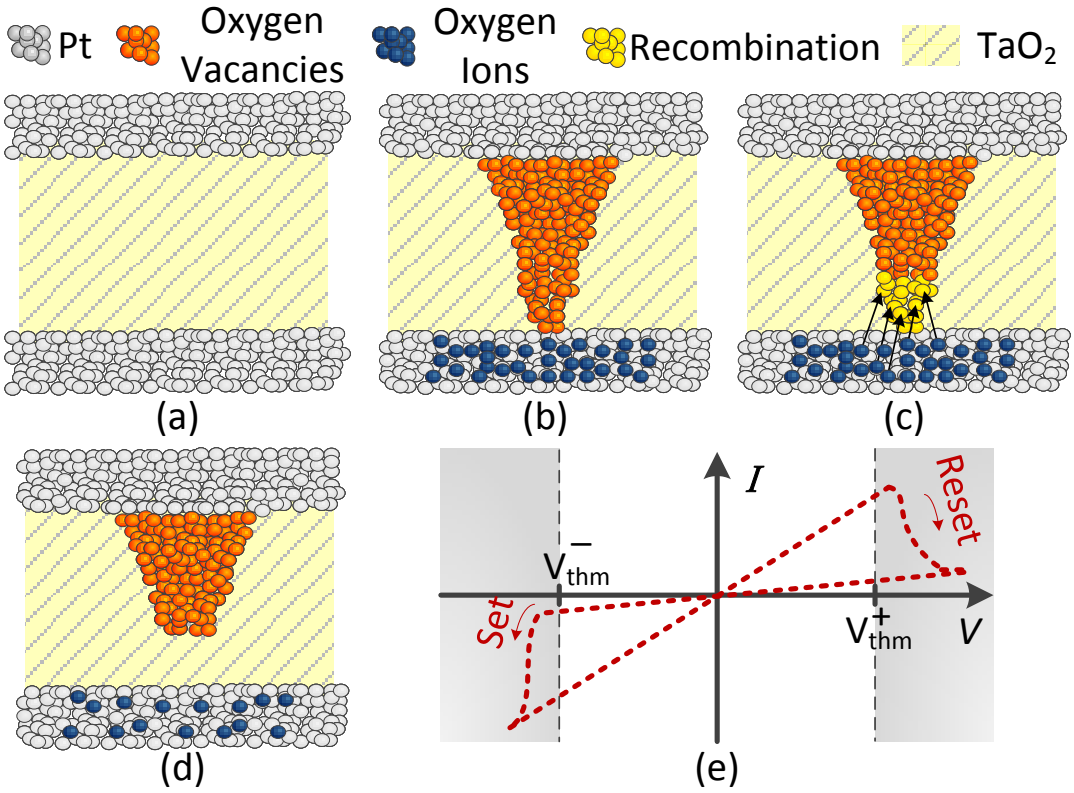


Figure 4.1: A possible realization of a memristor. a) A memristive device stack before forming, b) A formed memristive device in the ON state. c) RESET process d) a memristor in an OFF state e) a typical electrical characteristics of a memristor.

4.2.2 Complementary Resistive Switches

In 2010, Linn *et al.* [21] proposed the concept of complementary resistive switches (CRS) by anti-serially stacking two memristors sharing a common electrode, as shown in Figure 4.2a. Simpler CRS structures were proposed later by removing the common electrode and having two layers of the same oxide material but with different stoichiometries (e.g. Ta_2O_5 and TaO) [75], as illustrated in Figure 4.2b. The latter structure makes the cost and complexity of fabricating a CRS similar to that of a regular memristor.

The CRS was proposed to store data based on the combined state of the top and bottom memristors, M_t and M_b, rather than the overall device resistance. A CRS represents logic 0 (CRS-0) when the $\{M_t, M_b\}$ pair is in the $\{ON, OFF\}$ state. Similarly an $\{OFF, ON\}$ configuration

represents a logic 1 (CRS-1). With M_t and M_b being in series, both configurations exhibit a very high resistance which leads to lower current requirements and reduced power consumption [80].

Figure 4.2c illustrates a typical I-V characteristics of a CRS device, that exhibits two types of switching: CRS switching and memristive switching. Applying a voltage pulse above a CRS write threshold, V_{thc}^{+}, results in a *CRS switching* which forms a *strong* conductive filament in M_b while turning M_t OFF. Hence, a CRS is switched to an $\{OFF, ON\}$ configuration (i.e. transition ① in Fig 4.2c). With such a strong filament in M_b, applying voltages in the *memristive write region* (i.e. $[V_{thc}^{-}, V_{thm}^{-}]$ and $[V_{thm}^{+}, V_{thc}^{+}]$) provides a regular memristive write access to the top device M_t without affecting M_b: The top device exhibits a regular *memristive switching* behavior, and can switch between ON (i.e. transition ②) and OFF states (i.e. transition ③). Figure 4.2c shows memristive and CRS switching behaviors with blue and red lines, respectively.

Similarly, applying voltage pulses below V_{thc}^{-} switches a CRS into an $\{ON, OFF\}$ state and forms a strong filament in M_t (i.e. transition ④). With a strong filament in M_t, applying a voltage pulse in the memristive write region switches M_b between OFF and ON states, i.e. transitions ⑤ and ⑥ respectively. Note that M_t and M_b are anti-serially connected, thus they require opposite voltage polarities to switch.

In a nutshell, when one of the devices in a CRS stack is set to a strong ON state via a CRS switching, the other device can be written to exclusively, with regular memristive write accesses. Such voltage-range-controlled state transitions in CRS devices present a unique yet uninvestigated feature of such devices: to provide a dual-memristor memory cell with individual accesses to each of the constituent devices at the exact same footprint of a regular memristor. We explore this feature to provide a spare for each memory bit and extend the lifetime of ReRAMs.

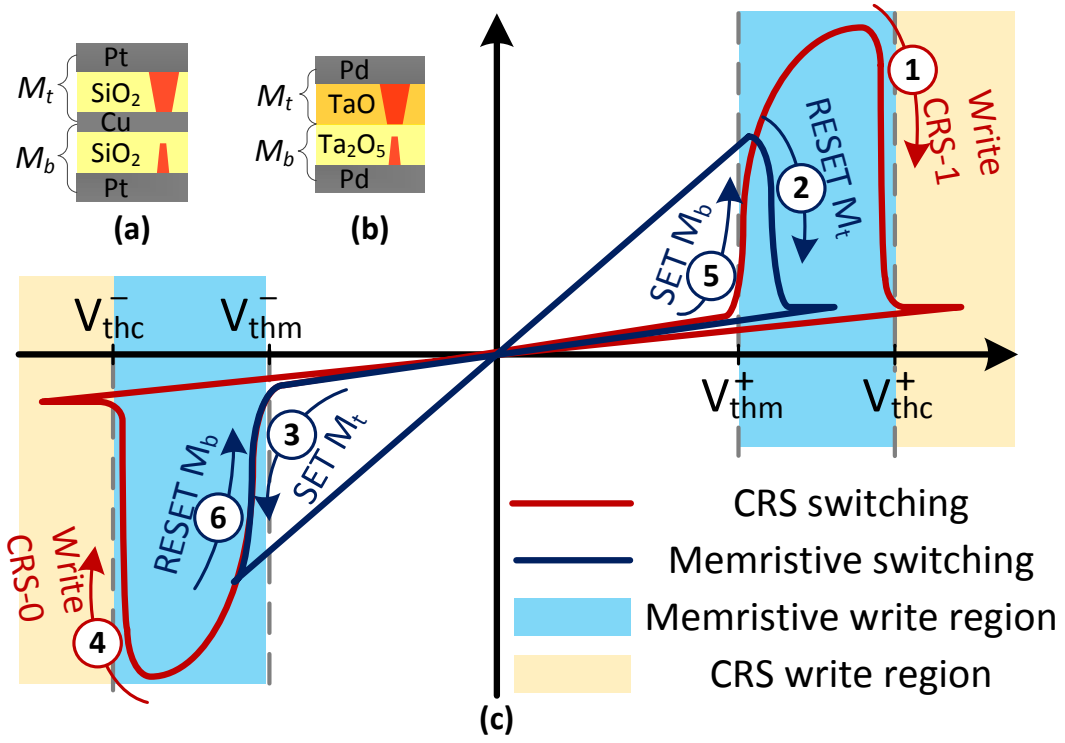


Figure 4.2: Complementary resistive switches. a) A CRS realized by two anti-serially memristors stacked on top of each other. b) Simple CRS structure with removed middle electrode. c) Typical I-V diagram and state transitions.

4.3 Failure Mechanisms and Solutions

Several mechanisms have been observed in memristors that result in recoverable data errors (i.e. soft error) or permanent device failures (i.e. hard errors). Such errors can be described by the stochastic and filamentary nature of the device.

4.3.1 Soft Errors

Soft errors in memristive devices are generally due to an "unintended" formation/rupture of the conductive filament inside a memristor. *Retention failure* [77] is an example, which occur when a weak conductive filament is ruptured due to the stochastic movement of the conductive particles, causing an ON→OFF flip.

The cycle-to-cycle variation in the write time of memristive devices, i.e. the time required to switch a device, is another source of soft errors in ReRAMs. Memristive devices could have ultra-slow write cycles for which the duration of the applied write pulse is not enough to switch the device [52]. An adaptive write mechanism can be used to address this issue, which monitors the state of the target cells during a write operation, and report any unsuccessful bit-write to trigger a re-write [81].

4.3.2 Hard Errors

Hard errors are due to permanent structural transformations inside a memristive cell. Several mechanisms lead to stuck-at-ON ($S@ON$) hard errors in memristors. *Extra vacancy attributed* failures result in an "irreversible" increase in the diameter of the conductive filament. The *depletion of the cathode from oxygen ions* is another reason behind $S@ON$ failures that reduces the recombination probability of oxygen vacancies and oxygen ions and decreases a memristor's OFF resistance [76].

As for stuck-at-OFF ($S@OFF$) failures, *over-reset* phenomena has been reported in which over-stressing the device with consecutive RESET operations could lead to a complete dissolution of the conductive filaments inside the device. An over-reset device cannot be recovered with regular SET operations [73].

While $S@OFF$ hard-errors might be fixed by applying high-voltage pulses (i.e. an extra forming step), $S@ON$ errors are harder to address. In a $S@ON$, a memristor is shorted and has a very low electrical resistance that is comparable to that of the voltage drivers' transistors. Hence, even in the case of applying a high-voltage RESET pulse, the effective voltage on the device would be small due to the large IR-drop on the line drivers' transistors, and thus cannot reverse the failure, .

To the best of our knowledge, there is no comprehensive study on the relative error rate

of $S@ON$ and $S@OFF$ errors in ReRAMs. However, the abundance of studies on $S@ON$ failures [76, 82, 83], as well as the reversibility of some $S@OFF$ errors (e.g. by another forming step)[73] suggests a higher error rate for $S@ON$ failures.

4.3.3 Potential Solutions

Soft errors are commonly addressed by the use of error correction codes (ECC). An ECC encodes the data and adds parity bits which enables the correction of T bits of errors during a read access. An ECC can also detect up D faulty bits, where D is greater than T. There is a myriad of ECCs in the literature, providing various levels of protection against errors. Among the most commonly used ECCs for memories are Hamming codes [84], that offer single-error-correction (i.e. $T = 1$), and Bose-Chaudhuri-Hocquenghen (BCH) codes [85], that are a family of ECC with multi-bit error correction capability.

Memory scrubbing is another method to address soft errors [86]. A scrubbing controller periodically reads data words, check them for errors through the use of ECC, and write the corrected data back in case of an error.

ECC can also be utilized to address hard errors. *Scrambling* methods are applied to distribute the faulty bits into different code-words such that the number of faulty-bits per code-word is less than the correction capability of the adopted ECC [87]. However, using ECC to correct hard errors reduces its effective correction capability against random soft errors. Moreover, ECC comes with noticeable energy overhead as it surcharges an encoding/decoding phase to each memory access. This is in addition to the area overhead of the parity bits and the ECC logic. The overhead increases with the ECC strength: stronger ECCs can correct more errors, but also incur more overhead.

Redundancy-based repair schemes are used to specifically address hard errors. Such repair schemes detect hard errors and use embedded “spare rows” to replace faulty rows [88, 18].

The row replacement is accomplished with the help of a remapping logic which relies on a content addressable memory (CAM). A CAM stores the addresses of the faulty rows along with the addresses of existing spares to replace them. The remapping logic uses the CAM to redirect future accesses to faulty rows to their corresponding spares [19]. To support the use of spare-rows, an additional access to the remapping CAM is necessary for each memory access, which adds a performance penalty to all memory accesses. Increasing the number of spares provides greater protection against hard errors, at the cost of increased area and performance penalty due to a larger CAM.

4.4 Motivation and Proposal

Existing solutions to extend the lifetime of memories and protect them against failures, such as spare rows and ECC, come with considerable energy and area overhead. This overhead becomes even more noticeable for emerging ReRAM technologies which are ultra-small and ultra low power. Hence, low-cost solutions that can help reduce such overheads will be attractive and valuable. To this end, we explore the use of complementary resistive switches, to provide "virtually-free" in-place spares per each memory element to extend the lifetime of a ReRAM.

4.4.1 CRS Devices as In-place Spares

A complementary resistive switch can be used to realize dual-memristor memory elements. It is shown that the unique electrical characteristics of a CRS provides exclusive write accesses to each of the two constituent devices by controlling the range of applied write voltages [80]. Furthermore, an exclusive read access to either of the devices in a CRS stack can be realized by keeping the other device in an ON state. Note that a CRS read operation reads the "total" resistance of the constituent devices that are in series. Hence, keeping either of the devices in an ON state makes it transparent to the read operation, in view of the orders of magnitude

difference between the ON and OFF resistance values of a memristive device [24].

Inspired by the possibility of such an exclusive read and write accesses, we propose the use of the extra memristor in a CRS cell as a spare. For clarity, we consider the top memristor in a CRS stack, M_t, as the *spare*, and the bottom one, M_b, as the *primary* device. The idea is to first utilize the primary device as the active memory element, and then use the spare, upon the failure of the primary device. The primary device is "activated" by applying a CRS write pulse below V_{thc}^{-}, as shown in Figure 4.3a. Such pulse initializes the $\{M_t, M_b\}$ pair to an $\{ON, OFF\}$ state and keeps the spare in an ON state that is transparent to read or write accesses. When activated, the primary device can be written to through regular write accesses without affecting the spare. That is, M_b can be switched between ON and OFF states (i.e. $\{ON, ON\}$ and $\{ON, OFF\}$ CRS configurations), as shown in Figure 4.3b, until it fails due to a hard error. If the primary device fails into a $S@ON$ state (which is more likely to happen than $S@OFF$, as discussed in Section 4.3.2), the memory element can be *repaired* by activating the spare device. To this end, a one-time CRS write pulse above V_{thc}^{+} sets the CRS to an $\{OFF, ON\}$ state (Figure 4.3c). From there on, the spare device becomes the active memory element that can switch between ON and OFF states, while the $S@ON$ primary device is transparent to the memory accesses.

A $S@OFF$ failure of the primary device could render the spare useless, as in that case, the whole CRS cell becomes $S@OFF$. In section 4.5 we examine the effect of $S@OFF$ failure rates and show that even under a pessimistic assumption that the $S@ON$ and $S@OFF$ error rates are equal, our method can still improve the memory lifetime considerably.

The use of such "in-place" spares provides two main advantages over the conventional redundancy-based repair schemes such as spare-rows: 1) No area-overhead is incurred, as the spare devices are fabricated on top of the primary devices and as part of the same device stack, and 2) in contrast to the spare-rows, such in-place spares exist at the exact same address as the failed memory element. Hence, there is no need for address remapping to activate the spares, and thus the overhead associated with the remapping logic can be avoided.

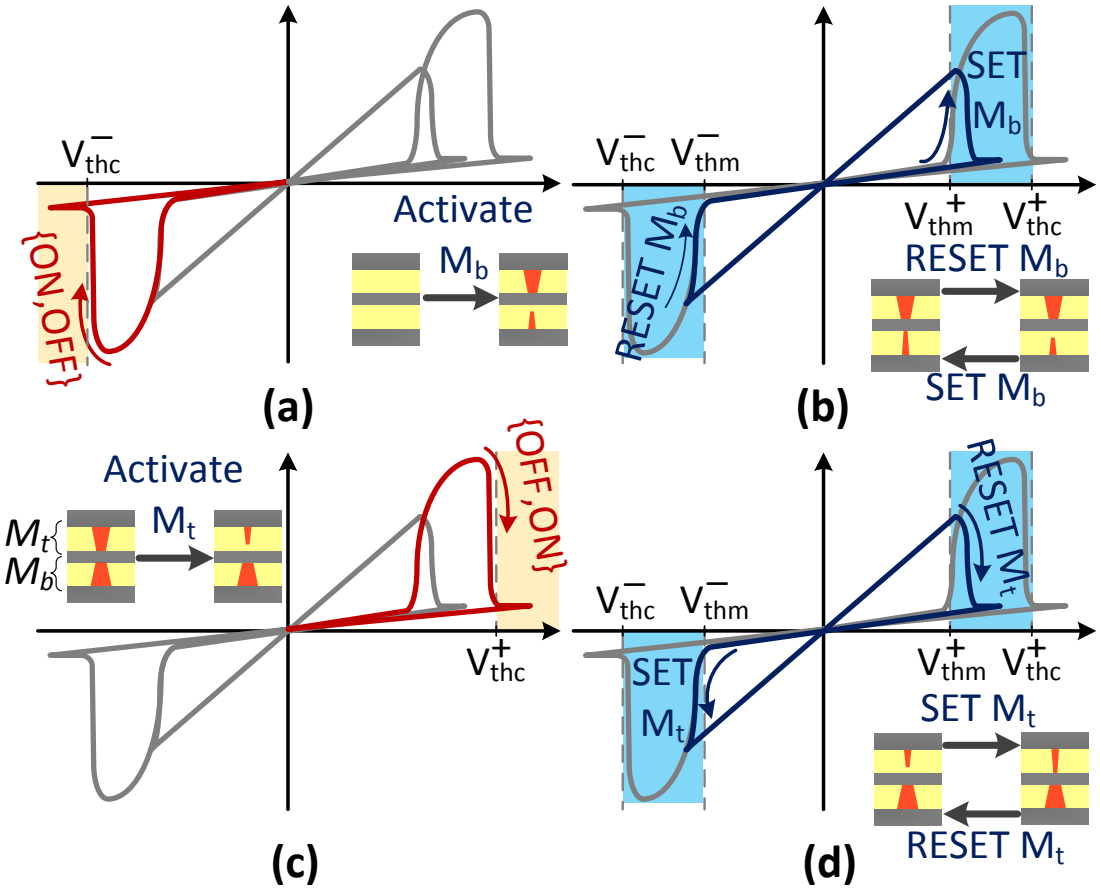


Figure 4.3: The evolution of a DMC as a memory element with an in-place spare: a) Activation of the primary device with a V_{thc}^{-} pulse. b) Use of the primary device as a regular memristor with ON $\leftrightarrow$ OFF switching. c) Spare activation with a V_{thc}^{+} pulse. d) The spare is used as the active memristor.

In order to differentiate the proposed use of a CRS as a dual-memristor-cell (DMC) with in-place spares, from the original CRS concept, hereafter we refer to a CRS stack as DMC.

4.4.2 Architectural Modifications

The anti-serially connected memristors in a DMC are accessed with opposite polarities: while applying a positive write pulse switches an active primary device into an OFF state, the same pulse would turn ON the spare device if it is activated. Hence, the memory management system needs to track which device in a DMC is active to ensure that for a write operation, the

right voltage polarity is applied to the active device.

We propose the use of a "polarity bit", *pbit*, to track the active device in a DMC. The *pbit* is accessed prior to each write operation to select the proper write voltage polarity. To minimize the overhead of bookkeeping, we use only one *pbit* per block: either all DMCs in a block use the primary device as their active device, or all of them use the spare.

To avoid the performance penalty of accessing the *pbit*, we take advantage of the *paging system* commonly implemented in the OS [86]. The OS maintains validity, permission, and address translation information for fixed-length contiguous blocks of memory that are called *pages*, in a *page table entry*. Page table entries are loaded into an extremely fast CAM called *translation look-aside buffer* (TLB), and are accessed as a part of each memory operation. Hence, by storing the DMC polarity data, i.e. *pbit*, at a page-level granularity, the *pbit* can be stored in the corresponding page table entry and accessed with no extra performance penalty during a write operation. Figure 4.4 highlights the minor modifications made to the datapath of a ReRAM to track the page polarity in green.

4.4.3 Repair Mechanism

The ECC can only correct up to T bits of errors per word-line. To extend the lifetime of a ReRAM, we propose a repair scheme that employs in-place spares to repair word-lines with more than T bits of errors. Our repair scheme reuses commonly adopted reliability improvement mechanisms, i.e. ECC and the adaptive write mechanism, to detect the number of errors during regular memory accesses: an adaptive write mechanism reports the exact number of bit-errors during a write operation, while ECC can detect up to D bits of errors during a read access, where D is larger than T.

Knowing the number of bit-failures, N_e, our repair scheme triggers the replacement of a word with a spare, as soon as N_e exceeds the correction capability of the ECC. Possible spare rows

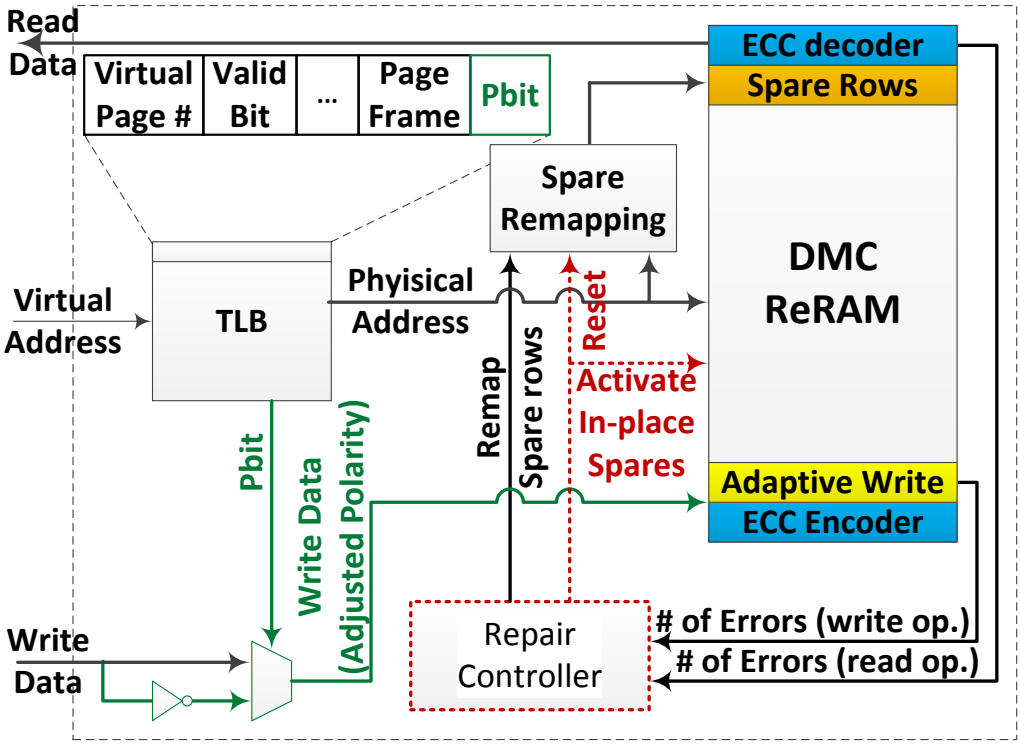


Figure 4.4: A DMC ReRAM. Green components keep track of the DMC page polarity: *pbit* is read from a page table entry to select the write voltage polarity. Dashed red components enable the repair: The existing repair controller is extended to activate in-place spares for the whole page once all spare rows are exhausted.

are utilized first to replace a defective word, W_f. The "address remapping" logic is configured to remap W_f to an available spare row. Once the spare rows are exhausted, the repair mechanism is triggered and the in-place spares are activated for the whole page.

The activation process of the in-place spares consists of three phases: For each word in a page, 1) the word is read and stored in a buffer, 2) spare devices are activated by applying a V_{thc}^{+} voltage pulse to the device stack, and 3) the buffered values are written back to the spare-activated word-line. The repair controller also updates the *pbit* and resets the remapping logic for the spare-activated page. Figure 4.4 shows a ReRAM equipped with in-place spares.

4.5 Analysis and Results

4.5.1 Viability Model

In order to evaluate the effect of the in-place spares on extending the lifetime of a ReRAM, we derive a statistical model to assess the *viability* of a ReRAM system in the presence of hard and soft errors. A viability function, $V(t)$, is defined as the probability that by time t, a ReRAM system has not yet experienced a failure, i.e. an error that cannot be corrected by ECC or repaired by spares. We use the Poisson distribution to model the probability of a $S@ON$ (or $S@OFF$) bit-failure at time t, $P_{S@1(or\ 0)}(t)$, with a fixed error rate, λ_1 (or λ_0) [88]:

$$P_{S@1(0)}(t) = 1 - e^{-\lambda_{1(0)}t} \tag{4.1}$$

Similarly, a fixed error rate λ_s is assumed for soft errors. We further consider a correction rate, μ, to model soft error mitigation mechanisms such as scrubbing. Equation 4.2 derives the probability of having a faulty bit due to a soft error, $P_{SE}(t)$:

$$P_{SE}(t) = \frac{\lambda_s}{\mu + \lambda_s}(1 - e^{-(\mu+\lambda_s)t}) \tag{4.2}$$

With the use of ECC, a B-bit word-line (that has B_D data bits and B_P parity bits) is viable as long as the total number of hard and soft bit-errors per word does not exceed T. Note that the number of parity bits depends on the ECC correction capability. $V_W(t,t_a)$ captures the viability of a word-line:

$$V_W(t,t_a) = \sum_{i=0}^{T} \sum_{j=0}^{T-i} \sum_{k=0}^{T-i-j} \binom{B}{i} P_{S@0}(t)^i (1-P_{S@0}(t))^{B-i} \cdot \binom{B-i}{j} P_{S@1}(t-t_a))^j (1-P_{S@1}(t-t_a))^{B-i-j} \cdot \binom{B-i-j}{k} P_{SE}(t)^k (1-P_{SE}(t))^{B-i-j-k} \quad (4.3)$$

where i, j, and k represent the number of $S@OFF$, $S@ON$, and soft errors respectively, and t_a denotes the activation time of the in-place spares. Note that activating the in-place spares *resets* $S@ON$ errors in a DMC ReRAM. Hence, for the calculation of the $S@ON$ bit-failure probability, the time origin is adjusted accordingly. The t_a equals 0 when measuring the viability of a word with regular memristors or a DMC word but before the activation of the in-place spares.

Considering S spare rows per page, a page with W words remains viable as long as the number of faulty words in the page does not exceed S. Equation 4.4 captures the page viability, $V_{page}(t,t_a)$, assuming hot spare rows:

$$V_{page}(t,t_a) = \sum_{i=0}^{S} \binom{W+S}{i} V_w(t,t_a)^{W+S-i} (1-V_w(t,t_a))^i \quad (4.4)$$

The viability of a regular ReRAM page is obtained by setting t_a equal to 0 in Equation 4.4. In case of a DMC memory, the page viability both before and after the activation of the in-place spares should be considered, as given in Equation 4.5:

Table 4.1: Simulation parameters

Parameter	Description	value
λ_s	Soft error rate	10^{-12}
λ_1	stuck at ON error rate	10^{-10}
λ_0	stuck at OFF error rate	$\rho\lambda_1$
ρ	$S@ON$ to $S@OFF$ error ratio	{1,10,100}
μ	Soft error correction rate	10^{-11}
T	ECC Correction capability	{0,1,2}
W	# of words per page	1024
S	# of spare words per page	[0-64]
B_D	# of data bits per word	{64,128,256}

$$V_{DMC}(t) = V_{page}(t,0) + \int_0^t -V'_{page}(t_a,0)V_{page}(t,t_a)dt_a \tag{4.5}$$

The first term in Equation 4.5 represents the viability of a page prior to the in-place spare activation. The activation of the in-place spares at time t_a provides an additional viability, $V_{page}(t,t_a)$. However, t_a is a random variable in the $[0,t]$ range. Hence, the viability component due to the spare activation is integrated over this range, with regard to the probability distribution function of t_a, that is $-V'_{page}(t,0)$.

The lifetime of a memristive page can be derived based on the viability function, according to Equation 4.6:

$$Lifetime = \int_0^\infty -tV'_{DMC}(t)dt \tag{4.6}$$

Note that while our calculations employ a Poisson distribution for hard and soft errors, other distributions can be applied by customizing Equations 4.1 and 4.2. Furthermore, a ReRAM with no spare rows and/or ECC, can be modeled simply by setting S and/or T to 0, respectively.

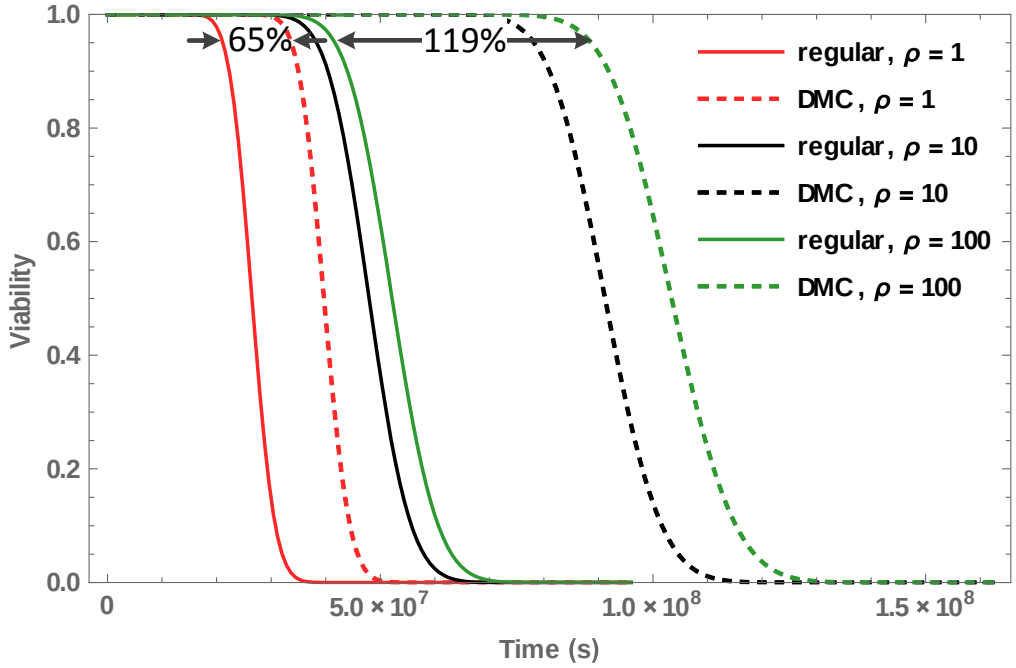


Figure 4.5: Viability of a DMC ReRAM versus a regular ReRAM for different $S@ON$ to $S@OFF$ error rate ratios.

Table 4.1 summarizes the employed parameters and their exemplar values.

4.5.2 Effect of In-place Spares on Lifetime

Figure 4.5 illustrates the viability of a DMC ReRAM page (solid lines) versus that of a baseline regular ReRAM (dashed lines). Results are shown for an exemplar case of $T = 2$, $B_D = 64$, $W = 1024$, $S = 8$, $\lambda_s = 10^{-12}$, $\lambda_1 = 10^{-10}$, $\mu = 10^{-11}$, and for different $S@ON$ to $S@OFF$ error rate ratios, ρ. To quantify the viability improvements, we consider the time at which a memory page shows 99% viability, $t_{99\%}$. For $\rho = 100$ (green lines), a DMC ReRAM extends $t_{99\%}$ by 119%. Even with equal $S@ON$ and $S@OFF$ error rates (red lines), $t_{99\%}$ is still improved by over 65%.

Figure 4.6 illustrates the effect of the in-place spares on the lifetime of a ReRAM page as a function of S and T, while ρ is set to 10. For example, with $S = 8$ and $T = 2$, use of the in-place spares can increase the lifetime by 91%.

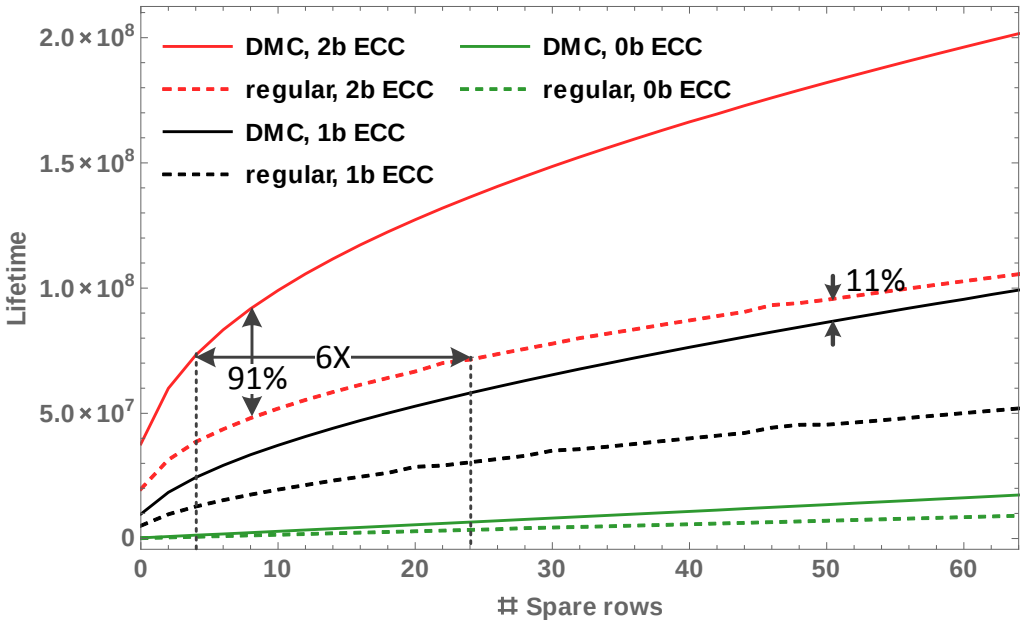


Figure 4.6: The effect of in-place spares, ECC correction capability, and the number of spare rows on the lifetime of a ReRAM page.

Figure 4.6 demonstrates the possibility of using in-place spares to reduce the number of spare rows. For example, with $T = 2$, a DMC ReRAM page with four spare rows provides the same lifetime as a regular memristive page with 24 spare rows.

The use of in-place spares also provides an opportunity to use lighter-weight ECCs in a ReRAM system to save on the area and energy requirements of the ECC, while maintaining a similar ReRAM lifetime. For an exemplar ReRAM page with 48 spare rows, i.e. 4% row redundancy, a DMC ReRAM page protected by a single-error-correcting (SEC) ECC exhibits a lifetime that is only 11% short of that of a regular page with a double-error-correcting (DEC) ECC.

4.5.3 Energy and Area Reduction

Figure 4.7 illustrates the possible reduction in area and power consumption by using the in-place spares to reduce the number of spare rows. This reduction is mainly attributed to the

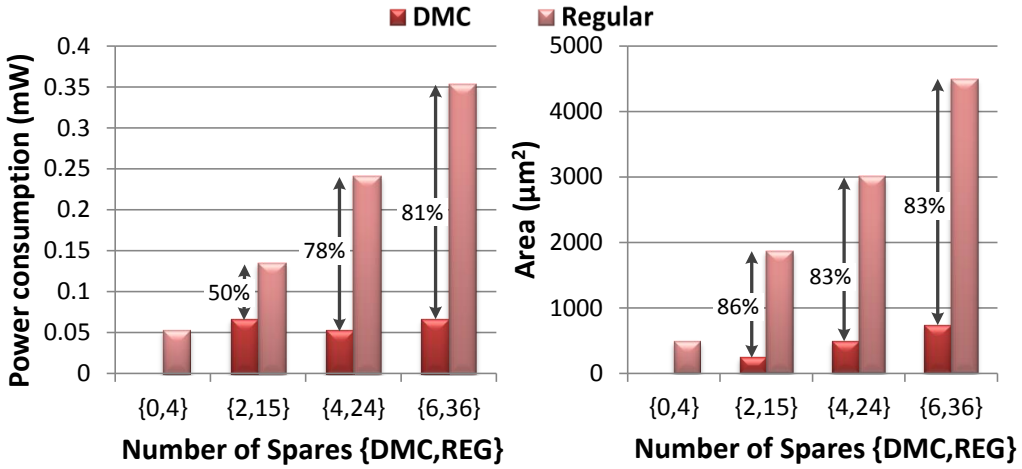


Figure 4.7: Power consumption and area requirements of the remapping logic CAM. A DMC ReRAM provides a similar lifetime with significantly lower number of spare rows, thus saves on the CAM's area and power consumption.

reduction in the size of the CAM module in the remapping logic. The horizontal axis shows the number of spares in pairs of $\{S_{DMC}, S_{Reg}\}$, where S_{Reg} is the necessary number of spare rows in a regular ReRAM, to provide a lifetime similar to that of a DMC ReRAM with S_{DMC} spare rows. The vertical axis shows the area and power overhead of a CAM to support S_{DMC} and S_{Reg} spare rows, respectively. For example, a DMC ReRAM page with six spare rows, provides the same lifetime as a regular ReRAM page with 36 spare rows. Hence, with smaller number of spare rows required, the power and area requirements of the remapping CAM can be reduced by 81% and 83%, respectively. Power and are numbers are obtained by synthesizing different CAM sizes with Synopsys design compiler at a 45nm CMOS technology node targeting a 200ps latency.

Figure 4.8 shows the potential of the in-place spares to reduce the area and energy consumption of a ReRAM system by enabling lighter-weight ECCs while maintaining a similar lifetime. SEC and DEC BCH encoder/decoders are synthesized using Synopsys design compiler, targeting a 400ps latency in a 45nm CMOS technology. For 128-bit data words, reducing the a BCH's correction capability from two to one reduces the area and power consumption of the

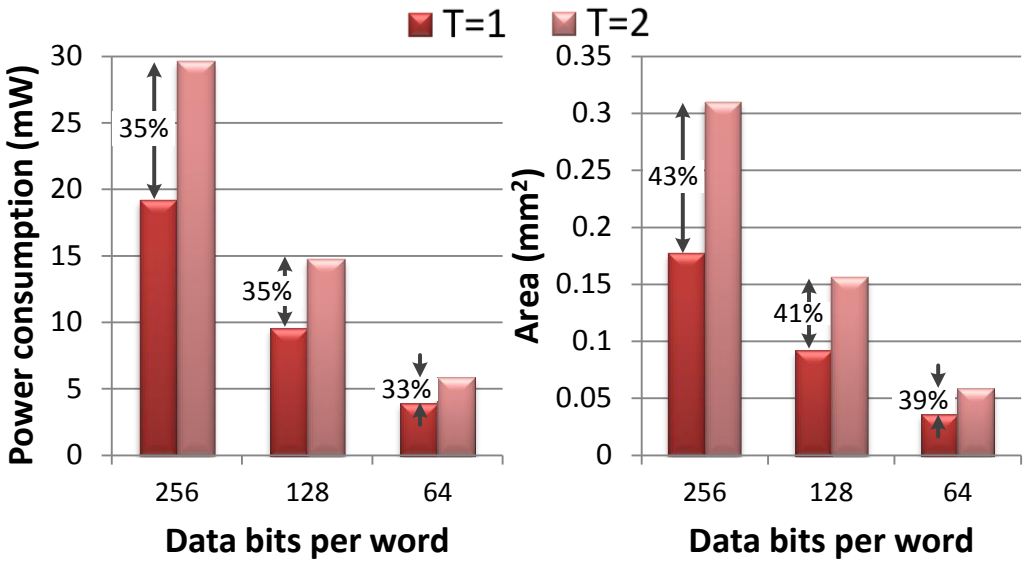


Figure 4.8: Power consumption and area requirements of a SEC and a DEC BCH ECC logic. The in-place spares enable employing a lighter-weight ECC, thus saving on area and power.

ECC logic by 41% and 35%, respectively. The savings improve further for words with more data bits. Note that reducing the ECC complexity results in greater savings compared to the savings resulting from reducing the number of spare rows.

4.6 Concluding Remarks

We propose a novel use of complementary resistive switches, to provide a dual-memristor-cell (DMC) with an in-place spare for ReRAM at negligible extra cost. The in-place spares can repair stuck-at-ON defects that are prevailing in a ReRAM system. Unlike conventional redundancy-based schemes, the proposed method incurs no area overhead due to the spares and does not require a remapping logic.

We present a statistical model to evaluate the effectiveness of the proposed method on extending the ReRAM's lifetime. The use of the in-place spares can roughly double the lifetime of ReRAMs. Alternatively, a DMC ReRAM exhibits a similar lifetime to a regular ReRAM, but with a lighter-weight ECC. The reduction in the complexity of the ECC can save an average of

39-43% on the area and 33-35% on the energy consumption of a BCH ECC module. Similarly, use of a DMC ReRAM can save on power and area by reducing the number of spare rows required to attain a given lifetime, which results in ≈6X reduction in the area overhead and power consumption of a CAM module inside the remapping logic.

Chapter 5

Conclusion

In this thesis we focus on addressing several sources of error and unreliability in the emerging memristive memory technology. One of the main advantages of this technology is the possibility of realizing access-transistor-free (ATF) memristive crossbars. The elimination of the access-transistor is enabled by the super exponential non-linearity in the switching characteristics of memristive devices versus the amplitude of the applied voltage pulse [7]. However, the elimination of the access-transistors comes with a cost: accessing a target memristor in an ATF crossbar applies a partial voltage across many other devices sharing the same word- or bit-line. We address several challenges caused by the existence of such partially-biased devices.

One of such issues is a data reliability problem known as the *write disturbance*. During a write operation, the write disturbance potentially degrades the resistance value of the memristive devices that are on the same word- or bit-line as the target memristor in an ATF crossbar. Such disturbance can accumulate over several write cycles and eventually corrupt the logic value stored in the affected devices. A major complexity before addressing this issue is the 2D domain of the affected devices that are both on the word- and bit-line. Hence, to address the write disturbance problem, we first confine the domain of the disturbance only to the word-line-shared memristors, via asymmetric distribution of the write voltage. The degradation of the data is

then monitored by having two regular memristors on each word-line. These memristors store logic 0 and 1 respectively and are used as references to check corruption trend and status. By monitoring these memristors, we can trigger a refresh operation to restore the disturbed data when necessary, and ensure data reliability.

The major advantage of the proposed method is the fact that the design-for-reliability hardware does not affect the regularity of the memristive crossbar, as it uses only regular memristors in the memristor layer, and requires no access-transistors. Hence, our method maintains all the benefits of an ATF crossbars while ensuring a reliable write operation. A case study shows that our method has less than 1% performance overhead and a moderate energy overhead of 41% to realize a reliable write operation in comparison with the baseline, unreliable implementation. This should be affordable due to the ultra-low-power characteristics of the memristive memories.

The other issue addressed in this thesis is the data reliability problem caused by significant variation in the write time characteristics of memristive devices. Such write time variation becomes particularly troublesome in ATF crossbar-based memories, where the leakage currents due to the partially-biased devices make it hard to detect the completion of a write operation. To address this problem, we propose an adaptive write circuitry that applies a write pulse with a just-enough width to switch the state of the target cell. The proposed write circuitry correctly detects switching events in a leakage-prone ATF memory system by employing a leakage-filtering approach. Our method first latches the data-dependent leakage current, so that the current only through the target cell can be then retrieved to enable accurate monitoring. The proposed method has an advantage of verifying the completion of the write operation as it keeps monitoring the state of the device during a write operation. This self-verification feature can be further utilized to reduce the test time of many march memory testing algorithms, as most of such algorithms have test sequences in which a read operation is required immediately after the write operation to verify the correctness of the write. With the proposed write scheme, the

verifying read operation can be omitted, resulting in up to 36% reduction in the test time of memory testing algorithms.

Our case study further shows the potential of the proposed method to achieve significant reduction in energy consumption over the conventional fixed-pulse write scheme. The exact energy saving ratio depends on the data pattern as well as the desired rate of correct write operations. For example, our method offers an average of 7X and 11X energy saving ratio, for random data patterns and exemplar success rates at $1-10^{-8}$ and $1-10^{-12}$, respectively.

Finally, we found a low-cost solution to repair stuck-at-ON device failures that are prevailing in ReRAM systems. The proposed solution makes a novel use of complementary resistive switches, to provide a dual-memristor-cell (DMC) with an in-place spare. Our method can be used both for conventional 1T-1R and the ATF memory architectures and incurs no area overhead to the memory. This is unlike traditional redundancy-based schemes which have the area overhead of the spares and a remapping logic. Presenting a statistical model to evaluate the effectiveness of the proposed method on extending the ReRAM's lifetime, we show that the use of the in-place spares can roughly double the lifetime of ReRAMs.

We further explore the possibility of using the in-place spares to replace the conventional spare-rows and/or reduce the complexity of the required error correction code (ECC) while maintaining a similar lifetime to a baseline ReRAM with spare-rows and ECC. We demonstrated that by using the proposed method to reduce the complexity of the required ECC, we can reduce an average of 39-43% on the area and 33-35% on the energy consumption of a BCH ECC module. Similarly, we showed that the use of a DMC ReRAM can reduce the number of spare rows required to attain a given lifetime by a factor of six.

The methods that are proposed throughout this thesis make future memristive memory modules more reliable and less prone to errors. Moreover, similar ideas can be used to address other open problems in ATF crossbars. One of such problems is the adverse effect of the leakage current of the bit-line-shared devices, I_{leak}, on a *current compliance* circuitry [11]. A current

compliance is used to ensure that the amount of current passing through the target memristor, I_{target}, does not exceed a given threshold. This is required for an accurate tuning of memristor's resistance which is needed to enable multi-bit storage in a single memristor. Current compliance is further utilized during electro-forming process [89]. However, in an ATF crossbar, we do not have a direct access to the I_{target}. Instead we can only observe and control the current at the end of the bit-line which has an extra I_{leak} current component. To address this issue, we can utilize the current-latching idea proposed in chapter 3. The leakage current can be latched and measured first, and the current compliance can be tuned to accommodate for the existing I_{leak}.

Bibliography

[1] *International Technology Roadmap for Semiconductors (ITRS)*, 2011.

[2] F. Bedeschi, R. Fackenthal, A. Resta, E. M. Donze, M. Jagasivamani, E. Buda, F. Pellizzer, D. Chow, A. Cabrini, G. Calvi, R. Faravelli, A. Fantini, G. Torelli, D. Mills, R. Gastaldi, and G. Casagrande, *A Multi-Level-Cell Bipolar-Selected Phase-Change Memory*, in *International Solid-State Circuits Conference*, p. 428, IEEE, February, 2008.

[3] A. Driskill-Smith, D. Apalkov, V. Nikitin, X. Tang, S. Watts, D. Lottis, K. Moon, A. Khvalkovskiy, R. Kawakami, X. Luo, , A. Ong, E. Chen, and M. Krounbi, *Latest Advances and Roadmap for in-Plane and Perpendicular STT-RAM*, in *3rd International Memory Workshop (IMW)*, pp. 1–3, IEEE, 2011.

[4] R. Waser and M. Aono, *Nanoionics-based resistive switching memories*, *Nature Materials* **6** (2007) 833–840.

[5] L. Chua, *Resistance switching memories are memristors*, *Applied Physics A: Materials Science & Processing* **102** (2011), no. 4 765–783.

[6] K.-T. Cheng and D. B. Strukov, *3D CMOS-Memristor Hybrid Circuits: Devices, Integration, Architecture, and Applications*, in *International Symposium on Physical Design (ISPD)*, IEEE, March, 2012.

[7] J. J. Yang, M.-X. Zhang, M. D. Pickett, M. Feng, J. P. Strachan, W.-D. Li, W. Yi, D. A. A. Ohlberg, B. J. Choi, W. Wu, J. H. Nickel, G. Medeiros-Ribeiro, and R. S. Williams, *Engineering Nonlinearity into Memristors for Passive Crossbar Applications*, *Applied Physics Letters* **100** (2012), no. 11 113501.

[8] C. Ho, C.-L. Hsu, C.-C. Chen, J.-T. Liu, C.-S. Wu, C.-C. Huang, C. Hu, and F.-L. Yang, *9nm Half-Pitch Functional Resistive Memory Cell with 1 uA Programming Current Using Thermally Oxidized sub-Stoichiometric WOx Film*, in *International Electron Devices Meeting (IEDM)*, pp. 19.1.1–19.1.4, IEEE, 2010.

[9] J. P. Strachan, A. C. Torrezan, M. F. Miao, M. D. Pickett, J. J. Yang, W. Yi, G. Medeiros-Ribeiro, and R. S. Williams, *Measuring the Switching Dynamics and Energy Efficiency of Tantalum Oxide Memristors*, *Nanotechnology* **22** (November, 2011) 505402.

[10] *International Technology Roadmap for Semiconductors (ITRS), Emerging Research Devices*, 2011.

[11] K.-H. Kim, S. Gaba, D. Wheeler, J. M. Cruz-Albrecht, T. Hussain, N. Srinivasa, and W. Lu, *A Functional Hybrid Memristor Crossbar-Array/CMOS System for Data Storage and Neuromorphic Applications*, *Nano Letters* **12** (2012), no. 1 389–395.

[12] Q. Xia, W. M. Tong, W. Wu, J. J. Yang, X. Li, W. Robinett, T. Cardinali, M. Cumbie, J. E. Ellenson, P. Kuekes, *et. al.*, *On the Integration of Memristors with CMOS Using Nanoimprint Lithography*, in *SPIE Advanced Lithography*, pp. 727106–727106, International Society for Optics and Photonics, 2009.

[13] A. Kawahara, R. Azuma, Y. Ikeda, K. Kawai, Y. Katoh, Y. Hayakawa, K. Tsuji, S. Yoneda, A. Himeno, K. Shimakawa, *et. al.*, *An 8 Mb Multi-Layered Cross-Point ReRAM Macro with 443 MB/s Write Throughput*, *Solid-State Circuits, IEEE Journal of* **48** (2013), no. 1 178–185.

[14] D. Niu *et. al.*, *Design trade-offs for high density cross-point resistive memory*, in *International Symposium on Low Power Electronics and Design*, pp. 209–214, ACM/IEEE, 2012.

[15] A. Ghofrani, M. A. Lastras-Montaño, and K.-T. Cheng, *Towards Data Reliable Crossbar-based Memristive Memories*, in *International Test Conference (ITC)*, pp. 1–10, IEEE, 2013.

[16] S. Gaba, P. Sheridan, J. Zhou, S. Choi, and W. Lu, *Stochastic Memristive Devices for Computing and Neuromorphic Applications*, *Nanoscale* **5** (2013) 5872–5878.

[17] C. Xu, D. Niu, Y. Zheng, S. Yu, and Y. Xie, *Impact of Cell Failure on Reliable Cross-Point Resistive Memory Design*, *ACM Transactions on Design Automation of Electronic Systems (TODAES)* **20** (2015), no. 4 63.

[18] T.-H. Wu, P.-Y. Chen, M. Lee, B.-Y. Lin, C.-W. Wu, C.-H. Tien, *et. al.*, *A Memory Yield Improvement Scheme Combining Built-in Self-repair and Error Correction Codes*, in *Test Conference (ITC), 2012 IEEE International*, pp. 1–9, IEEE, 2012.

[19] M. Lee, L.-M. Denq, and C.-W. Wu, *A Memory Built-in Self-repair Scheme based on Configurable Spares*, *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on* **30** (2011), no. 6 919–929.

[20] A. Ghofrani, M. A. Lastras-Montaño, Y. Wang, and K.-T. Cheng, *In-place Repair for Resistive Memories Utilizing Complementary Resistive Switches*, in *submitted to IEEE International Symposium on Low-Power Electronics and Design*, IEEE, 2016.

[21] E. Linn, R. Rosezin, C. Kgeler, and R. Waser, *Complementary Resistive Switches for Passive Nanocrossbar Memories*, *Nature Materials* **9** (2010) 403–406.

[22] *Grand Challenges*, *International Technology Roadmap for Semiconductors (ITRS)* (2011).

[23] *Emerging Research Devices*, *International Technology Roadmap for Semiconductors (ITRS)* (2011).

[24] J. J. Yang, D. B. Strukov, and D. R. Stewart, *Memristive Devices for Computing*, *Nature Nanotechnology* **8** (January, 2013) 13–24.

[25] L. Chua, *Memristor-the Missing Circuit Element*, *IEEE Transaction on Circuit Theory* **18** (1971), no. 5 507–519.

[26] R. S. Williams, *How We Found The Missing Memristor*, *IEEE Spectrum* **45** (2008), no. 12 28–35.

[27] *Molecular-junction-nanowire-crossbar-based neural network*, 4, 2008.

[28] A. C. Cabe, G. S. Rose, and M. R. Stan, *Reducing Stray Current in Molecular Memory Through Data Encoding*, in *7th IEEE conference on Nanotechnology*, pp. 70–75, IEEE, 2007.

[29] G. J. et al., *Circuit Fabrication at 17 nm Half-Pitch by Nanoimprint Lithography*, *Nano Letters* **6** (2006), no. 3 351–354.

[30] M. S. Qureshi *et. al.*, *CMOS Interface Circuits for Reading and Writing Memristor Crossbar Array*, in *International Symposium on Circuits and Systems (ISCAS)*, pp. 2954–2957, IEEE, 2011.

[31] J. E. Green *et. al.*, *A 160-Kilobit Molecular Electronic Memory Patterned at Bits per Square Centimeter*, *Nature* **445** (2006) 351–354.

[32] X. S. H. et al., *Design and Defect Tolerance Beyond CMOS*, in *6th International Conference on Hardware/Software Codesign and System Synthesis*, pp. 223–230, IEEE/ACM/IFIP, 2008.

[33] K. K. Likharev, *Electronics below 10 nm*, pp. 27–68. Elsevier, 2003.

[34] Q. Xia *et. al.*, *Memristor-CMOS Hybrid Integrated Circuits for Reconfigurable Logic*, *Nano Letters* **9** (2009) 3640–3645.

[35] D. TU *et. al.*, *Three-Dimensional CMOL: Three-Dimensional Integration of CMOS/Nanomaterial Hybrid Digital Circuits*, *Micro and Nano Letters* **2** (2007), no. 2 40–45.

[36] H. Y. Lee *et. al.*, *Evidence and Solution of Over-RESET Problem for HfOx Based Resistive Memories with Sub-ns Switching Speed and High Endurance*, in *International Electron Devices Meeting (IEDM)*, pp. 19.7.1–19.7.4, IEEE, 2010.

[37] D. B. Strukov and K. K. Likharev, *Prospects for terabit-scale nanoelectronic memories*, *Nanotechnology* **16** (2004), no. 1 137.

[38] Y. Ho *et. al.*, *Dynamical properties and design analysis for Nonvolatile Memristor Memories*, *IEEE Transaction on Circuits and Systems* **58** (April, 2011) 724–736.

[39] J. J. Y. et al., *Memristive switching mechanism for metal/oxide/metal nano devices*, *Nature Nanotechnology* **3** (2008) 429–433.

[40] E. Verrelli *et. al.*, *Forming-free resistive switching memories based on titanium-oxide nanoparticles fabricated at room temperature*, *Applied Physics Letters* **102** (2013), no. 2.

[41] D. B. Strukov and K. K. Likharev, *CMOL FPGA: A Cell-based, Reconfigurable Architecture for Hybrid Digital Circuits using Two-Terminal Nanodevices*, *Nanotechnology* **16** (2005) 888–900.

[42] E. Linn *et. al.*, *Complementary Resistive Switches for Passive Nanocrossbar Memories*, *Nature Materials* **9** (2010) 403–406.

[43] K. K. Likharev and D. B. Strukov, *CMOL: Devices, circuits, and architectures*, pp. 447–477. Springer, 2005.

[44] M. A. Zidan, H. A. H. Fahmy, M. M. Hussain, and K. N. Salama, *Memristor-based Memory: the Sneak Paths Problem and Solutions*, *Microelectronics Journal* **44** (2013), no. 2 176–183.

[45] S. Kim *et. al.*, *Flexible memristive memory array on plastic substrates*, *Nano Letters* **11** (2011), no. 12 5438–5442.

[46] *Interconnect*, *International Technology Roadmap for Semiconductors (ITRS)* (2011).

[47] S. Shin *et. al.*, *Compact Models for Memristors Based on Charge-Flux Constitutive Relationships*, *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on* **29** (April, 2010) 590 –598.

[48] H. J. Jeon and Y.-B. Kim, *A CMOS low-power low-offset and high-speed fully dynamic latched comparator*, in *IEEE International SOC Conference (SOCC)*, pp. 285–288, IEEE, September, 2010.

[49] A. Rahimi, A. Ghofrani, M. A. Lastras-Montaño, K.-T. Cheng, L. Benini, and R. K. Gupta, *Energy-Efficient GPGPU Architectures via Collaborative Compilation and Memristive Memory-Based Computing*, in *Design Automation Conference (DAC)*, June, 2014.

[50] A. Rahimi, A. Ghofrani, K.-T. Cheng, L. Benini, and R. K. Gupta, *Approximate Associative Memristive Memory for Energy-Efficient GPUs*, in *Design, Automation, and Test in Europe*, IEEE, 2015.

[51] F. Lee, Y. Y. Lin, M. H. Lee, W. C. Chien, H. L. Lung, K. Y. Hsieh, and C. Y. Lu, *A Novel Cross-Point One-Resistor (0T1R) Conductive Bridge Random Access Memory (CBRAM) with Ultra Low Set/Reset Operation Current*, in *Symposium on VLSI Technology (VLSIT)*, pp. 67–68, 2012.

[52] S. Yu, X. Guan, and H.-S. P. Wong, *On the Switching Parameter Variation of Metal Oxide RRAM; Part II: Model Corroboration and Device Design Strategy*, *IEEE Transactions on Electron Devices* **59** (2012), no. 4 1183–1188.

[53] J. J. Yang, M. D. Pickett, X. Li, D. A. A. Ohlberg, D. R. Stewart, and R. S. Williams, *Memristive Switching Mechanism for Metal/Oxide/Metal Nanodevices*, *Nature nanotechnology* **3** (2008), no. 7 429–433.

[54] H. Manem, G. S. Rose, X. He, and W. Wang, *Design Considerations for Variation Tolerant Multilevel CMOS/Nano Memristor Memory*, in *Proceedings of the 20th Symposium on Great Lakes Symposium on VLSI (GLSVLSI)*, pp. 287–292, ACM, 2010.

[55] A. Ghofrani, M. A. Lastras-Montaño, and K.-T. Cheng, *Toward Large-Scale Access-Transistor-Free Memristive Crossbars*, in *Asia South-Pacific Design Automation Conference (ASP-DAC)*, 2015.

[56] M. Payvand, A. Madhavan, M. A. Lastras-Montaño, A. Ghofrani, J. Roheh, K.-T. Cheng, D. B. Strukov, and L. Theogarajan, *A Configurable CMOS Memory Platform for 3D Integrated Memristors*, in *Circuits and Systems, 2015. ISCAS 2015. IEEE International Symposium on*, IEEE, 2015.

[57] S. H. Jo, K. H. Kim, and W. Lu, *Programmable Resistance Switching in Nanoscale Two-Terminal Devices*, *Nano letters* **9** (2008), no. 1 496–500.

[58] F. Alibart, L. Gao, B. D. Hoskins, and D. B. Strukov, *High Precision Tuning of State for Memristive Devices by Adaptable Variation-Tolerant Algorithm*, *Nanotechnology* **23** (2012), no. 7 075201.

[59] W. Yi, F. Perner, M. S. Qureshi, H. Abdalla, M. D. Pickett, J. J. Yang, M.-X. M. Zhang, G. Medeiros-Ribeiro, and R. S. Williams, *Feedback write scheme for memristive switching devices*, *Applied Physics A* **102** (2011), no. 4 973–982.

[60] K.-H. Jo, K.-H. Jo, C.-M. Jung, K.-S. Min, and S.-M. Kang, *Self-Adaptive Write Circuit for Low-Power and Variation-Tolerant Memristors*, *IEEE Transactions on Nanotechnology* **9** (2010), no. 6 675–678.

[61] A. J. Van de Goor, *Testing SemiconductorMemories*, pp. 27–68. John Wiley & Sons, 1991.

[62] S. Yu, Y. Wu, and H.-S. P. Wong, *Investigating the Switching Dynamics and Multilevel Capability of Bipolar Metal Oxide Resistive Switching Memory*, *Applied Physics Letters* **98** (2011), no. 10 103514.

[63] B. Razavi, *Fundamentals of Microelectronics*, vol. 1. Wiley, 2009.

[64] P.-Y. Chiu and M.-D. Ker, *Metal-Layer Capacitors in the 65nm CMOS Process and the Application for Low-Leakage Power-Rail ESD Clamp Circuit*, *Microelectronics Reliability* **54** (2014), no. 1 64–70.

[65] D. Long, X. Hong, and S. Dong, *Optimal two-dimension common centroid layout generation for mos transistors unit-circuit*, in *Circuits and Systems, 2005. ISCAS 2005. IEEE International Symposium on*, pp. 2999–3002, IEEE, 2005.

[66] G. Harutunyan, V. A. Vardanian, and Y. Zorian, *Minimal March Test Algorithm for Detection of Linked Static Faults in Random Access Memories*, in *Proceedings 24th VLSI Test Symposium*, IEEE, April, 2006.

[67] D. S. Suk and S. M. Reddy, *A March Test for Functional Faults in Semiconductor Random-Access Memories*, *IEEE Transactions on Computers* **C-30** (1981), no. 12 982–985.

[68] W. Z. Wan Hasan, M. Othman, and B. S. Suparjo, *A Realistic March-12N Test And Diagnosis Algorithm For SRAM Memories*, in *International Conference on Semiconductor Electronics*, pp. 919–923, IEEE, Oct, 2006.

[69] A. Benso, A. Bosio, S. D. Carlo, G. D. Natale, and P. Prinetto, *March AB, March AB1: New March Tests for Unlinked Dynamic Memory Faults*, in *International Test Conference (ITC)*, pp. 8–pp, IEEE, 2005.

[70] N. Z. Haron and S. Hamdioui, *On Defect Oriented Testing for Hybrid CMOS/Memristor Memory*, in *20th Asian Test Symposium (ATS)*, pp. 353–358, IEEE, 2011.

[71] N. H. E. Weste and D. Money, *CMOS VLSI Design: A Circuits and Systems Perspective*. Pearson Education India, 2005.

[72] X. Dong, C. Xu, Y. Xie, and N. P. Jouppi, *NVSim: A Circuit-Level Performance, Energy, and Area Model for Emerging Nonvolatile Memory*, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* **31** (2012), no. 7 994–1007.

[73] H. Y. Lee, Y. S. Chen, P. S. Chen, P. Y. Gu, Y. Y. Hsu, S. M. Wang, W. H. Liu, C. H. Tsai, *et. al.*, *Evidence and Solution of Over-RESET Problem for HfO_x-based Resistive Memory with sub-ns Switching Speed and High Endurance*, in *Electron Devices Meeting (IEDM), 2010 IEEE International*, pp. 19–7, IEEE, 2010.

[74] J. Rofeh, A. Sodhi, M. Payvand, M. A. Lastras-Montaño, A. Ghofrani, A. Madhavan, S. Yemenicioglu, K.-T. Cheng, and L. Theogarajan, *Vertical Integration of Memristors onto Foundry CMOS Dies using Wafer-Scale Integration*, in *Electronic Components and Technology Conference (ECTC) , 2015 IEEE 65th*, pp. 957–962, May, 2015.

[75] Y. Yang, P. Sheridan, and W. Lu, *Complementary Resistive Switching in Tantalum Oxide-based Resistive Memory Devices*, *Applied Physics Letters* **100** (2012), no. 20 203112.

[76] B. Chen, Y. Lu, B. Gao, Y. H. Fu, F. F. Zhang, P. Huang, Y. S. Chen, *et. al.*, *Physical Mechanisms of Endurance Degradation in TMO-RRAM*, in *2011 International Electron Devices Meeting*, 2011.

[77] B. Gao, H. Zhang, B. Chen, L. Liu, X. Liu, R. Han, J. Kang, Z. Fang, H. Yu, B. Yu, and D.-L. Kwong, *Modeling of Retention Failure Behavior in Bipolar Oxide-Based Resistive Switching Memory*, *Electron Device Letters, IEEE* **32** (March, 2011) 276–278.

[78] D. B. Strukov, G. S. Snider, D. R. Stewart, and R. S. Williams, *The Missing Memristor Found*, *Nature* **453** (2008), no. 7191 80–83.

[79] F. Miao, J. J. Yang, J. Borghetti, G. Medeiros-Ribeiro, and R. S. Williams, *Observation of Two Resistance Switching Modes in TiO_2 Memristive Devices Electroformed at Low Current*, *Nanotechnology* **22** (2011), no. 25 254007.

[80] M. A. Lastras-Montaño, A. Ghofrani, and K.-T. T. Cheng, *HReRAM: A Hybrid Reconfigurable Resistive Random-Access Memory*, *Proceedings Design, Automation, and Test in Europe (DATE), IEEE* (2015).

[81] A. Ghofrani, M.-A. lastras-montaño, S. Gaba, M. Payvand, W. Lu, L. Theogarajan, and K.-T. Cheng, *A Low-Power Variation-Aware Adaptive Write Scheme for Access-Transistor-Free Memristive Memory*, *ACM Journal on Emerging Technology in Computing Systems* **12** (Aug., 2015) 3:1–3:18.

[82] D. Strukov, *Endurance write speed tradeoffs in nonvolatile memories*, *arXiv preprint arXiv:1511.07109* (2015).

[83] B. Gao, H. Zhang, B. Chen, L. Liu, X. Liu, *et. al.*, *Modeling of Retention Failure Behavior in Bipolar Oxide-based Resistive Switching Memory*, *Electron Device Letters, IEEE* **32** (2011), no. 3 276–278.

[84] R. W. Hamming, *Error Detecting and Error Correcting Codes*, *Bell System technical journal* **29** (1950), no. 2 147–160.

[85] R. C. Bose and D. K. Ray-Chaudhuri, *On a Class of Error Correcting Binary Group Codes*, *Information and control* **3** (1960), no. 1 68–79.

[86] B. Jacob, S. Ng, and D. Wang, *Memory Systems: Cache, DRAM, Disk*. Morgan Kaufmann, 2010.

[87] S.-K. Lu, H.-C. Jheng, H.-W. Lin, M. Hashizume, and S. Kajihara, *Built-In Scrambling Analysis for Yield Enhancement of Embedded Memories*, in *Test Symposium (ATS), 2014 IEEE 23rd Asian*, pp. 137–142, IEEE, 2014.

[88] G. Mayuga, Y. Yamato, T. Yoneda, M. Inoue, and Y. Sato, *An ECC-based Memory Architecture with Online Self-repair Capabilities for Reliability Enhancement*, in *Test Symposium (ETS), 2015 20th IEEE European*, pp. 1–6, IEEE, 2015.

[89] Q. Xia, M. D. Pickett, J. J. Yang, X. Li, W. Wu, G. Medeiros-Ribeiro, and R. S. Williams, *Two- and three-terminal resistive switches: Nanometer-scale memristors and memistors*, *Advanced Functional Materials* **21** (2011), no. 14 2660–2665.