MODELING, CIRCUIT DESIGN, AND MICROARCHITECTURAL OPTIMIZATION OF EMERGING RESISTIVE MEMORY

A Dissertation in
Computer Science and Engineering
by
Cong Xu

© 2014 Cong Xu

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Doctor of Philosophy

December 2014
The dissertation of Cong Xu was reviewed and approved* by the following:

Yuan Xie  
Professor of Computer Science and Engineering  
Dissertation Co-Advisor, Co-Chair of Committee

Vijaykrishnan Narayanan  
Professor of Computer Science and Engineering  
Dissertation Co-Advisor, Co-Chair of Committee

Mary Jane Irwin  
Professor of Computer Science and Engineering

Suman Datta  
Professor of Electrical Engineering

Raj Acharya  
Professor of Computer Science and Engineering  
Department Head

*Signatures are on file in the Graduate School.
Abstract

Conventional memories technologies such as SRAM, DRAM, and NAND flash are facing formidable device scaling challenges. Various new non-volatile memory (NVM) technologies have emerged recently, including spin-torque-transfer random access memory (STT-RAM), phase-change memory (PCM), and resistive random access memory memory (ReRAM). Among them, ReRAM stands out due to its simple structure, low programming voltage, fast switching speed, high on/off ratio, excellent scalability, good endurance and great compatibility with the silicon CMOS technology. Although the initial target of ReRAM is NAND flash replacement, ReRAM holds the potential to revolutionize the memory hierarchy from the last-level cache to mass storage system. Through years’ efforts from both academia and industry, Gb-scale prototype ReRAM prototypes have been demonstrated. However, most of ReRAM research has still been focused on device-level development. As the ultimate goal of ReRAM research is to advance ReRAM in current memory hierarchy, the key question is how to architect ReRAM in different levels of the memory hierarchy. The work in this dissertation aims to address these issues.

First, several array/macro ReRAM models with different simulation accuracy and speed requirement are built and validated. Second, circuit-/architecture-level techniques that mitigate the large overhead in straightforward implementation of ReRAM prototypes are proposed and evaluated. Third, architectural-level case studies of adopting ReRAM in main memory and storage system are conducted. Fourth, the impact of cell failures in ReRAM design is analyzed and efficient hard error detection unit is proposed.
# Table of Contents

List of Figures viii  
List of Tables xiii  
Acknowledgments xiv  

## Chapter 1  
### Introduction  
1.1 The Demand for New Memory Technologies 1  
1.1.1 comparison of emerging NVM technologies 2  
1.1.2 advantages of ReRAM over STT-RAM/PCM 3  
1.2 Unique Challenges of ReRAM Development 4  
1.3 Roles of ReRAM in Memory Hierarchies 6  
1.4 Organization of This Dissertation 8  

## Chapter 2  
### Technology Background  
2.1 Emerging Non-Volatile Memory Technology 9  
2.1.1 spin-torque-transfer RAM 9  
2.1.1.1 perpendicular MTJ 11  
2.1.2 phase change memory 12  
2.2 ReRAM Cell Basics 13  
2.3 ReRAM Array Structures 15  
2.3.1 1T1R structure 16  
2.3.2 cross-point structure 16  
2.3.3 3D ReRAM structure 17
Chapter 3
Modeling of ReRAM

3.1 A Simplified Model for Cross-Point ReRAM ....................... 20
3.2 An Improved Analytical Model for Detailed Array Analysis .... 21
  3.2.1 mathematical description of the model ........................ 22
3.3 Modeling of 3D vertical ReRAM ................................... 24
  3.3.1 modeling of a ReRAM element ............................... 24
  3.3.2 modeling of a 3D-VRAM array .............................. 25
  3.3.3 geometry parameters and design constraints .................. 26
    3.3.3.1 modeling 3D-VRAM with planar access transistor . 26
    3.3.3.2 modeling 3D-VRAM with vertical access transistor 28
3.4 NVSim: Integrating the Models .................................. 29
  3.4.1 generic timing and power estimation .......................... 30
  3.4.2 data sensing models ....................................... 31
  3.4.3 charge pump ............................................. 34
  3.4.4 validation result ....................................... 34
    3.4.4.1 PCRAM validation .................................. 34
    3.4.4.2 ReRAM validation .................................. 35

Chapter 4
Optimize ReRAM Design for Low Cost Memory Component ........... 36

4.1 Cost Optimization: Circuit Design Tricks ......................... 37
  4.1.1 area-efficient output buffers ............................... 38
  4.1.2 sense amplifier sharing .................................. 39
  4.1.3 design evaluation ....................................... 39
4.2 Cost Optimization: Leveraging MLC ................................ 41
  4.2.1 trade-offs in resistance allocation schemes .................. 41
  4.2.2 trade-offs in programming strategies ....................... 43
    4.2.2.1 impact of retention failure on MLC reliability ...... 43
    4.2.2.2 fast programming strategy ........................... 44
    4.2.2.3 reliable programming strategy ....................... 45
    4.2.2.4 monte carlo simulations ............................. 45
  4.2.3 trade-offs in circuit design ................................ 47
  4.2.4 trade-offs in material selection ............................ 49

Chapter 5
Architecting ReRAM for main memory replacement ................... 53

5.1 Data Dependent Write Latency .................................. 55
  5.1.1 voltage-time dilemma of ReRAM ........................... 56
  5.1.2 design constraints analysis ............................... 58
5.2 Architectural Enhancement

5.2.1 double-sided ground biasing

5.2.2 impact of data pattern on write latency

5.2.2.1 split RESET phase

5.2.2.2 compression-enabled dual-write-speed mode

5.2.3 latency aware write scheduling

5.3 Results and Discussion

5.3.1 simulation setup

5.3.2 memory read latency

5.3.3 system performance

5.3.4 energy

5.3.5 sensitivity to mat size

5.3.6 sensitivity to $n$

5.3.7 comparison with DRAM and PCM

5.4 Summary

Chapter 6

Architecting ReRAM for next-generation storage system

6.1 3D-VRAM Array Structure

6.2 3D-VRAM Array Design

6.2.1 write and read margin

6.2.2 design space exploration

6.2.2.1 nonlinearity

6.2.2.2 resistance

6.2.2.3 read voltage

6.2.2.4 read and write energy

6.2.2.5 multi-bit access

6.3 Circuit and Architecture Design

6.3.1 optimize write and read circuitry

6.3.1.1 multi-directional write driver

6.3.1.2 remote sensing scheme

6.3.2 macro-architecture design

6.3.3 macro-Level model

6.3.3.1 timing model

6.3.3.2 cost model

6.3.4 results and discussions

6.4 System-Level Evaluation

6.5 Summary
List of Figures

1.1 Device characteristics of conventional and emerging memory technologies. ............................................. 3

2.1 Demonstration of a STT-RAM cell: (a) Conventional connection scheme; (b) Reverse connection scheme. .............. 10
2.2 The schematic view of a PCM cell with a MOSFET selector transistor (BL=bitline, WL=wordline, SL=sourceline). .............. 12
2.3 The temperature-time relationship during SET and RESET operations. ..................................................... 13
2.4 Schematic view of a RRAM cell ................................................. 14
2.5 Multi-level switching in ReRAM: (a) H2L and (b) L2H programming. .................. 14
2.6 Write-and-verify in H2L programming ............................................. 15
2.7 Schematic view of the 1T1R Structure (WL=wordline, BL=bitline, SL=sourceline). ............................................. 16
2.8 Zoomed-in view of the 1T1R Structure. .................................................................................. 16
2.9 Schematic view of crosspoint structure (WL=wordline, BL=bitline). ......................................................... 17
2.10 Zoomed-in view of crosspoint structure. .................................................................................. 17
2.11 Schematic view of 3D Horizontal ReRAM. ................................................................................. 18
2.12 Schematic view of 3D Vertical ReRAM. ................................................................................. 18
2.13 Voltage biasing in 3D-VRAM for (a) write and (b) read operations 19

3.1 A detailed analytical model of the cross-point array. .......... 22
3.2 Circuit model: abstraction of a 2-layer 3D-VRAM array .......... 25
3.3 Analysis model for current sensing scheme. .................. 32
3.4 Analysis model for current-in voltage sensing scheme. .......... 32
3.5 Analysis model for voltage-divider sensing scheme. ........ 32
3.6 Delay model verification of three sensing schemes comparing to HSPICE simulations. .................. 33

4.1 Buffer design optimization: (a) latency-optimized; (b) balanced; (c) area-optimized. ................................. 38
5.14 Percentage of incompressible cachelines by the proposed compression approach across SPEC 2006 and STREAM benchmarks  71
5.15 Normalized IPC speedup for SPEC 2006 and STREAM benchmarks  71
5.16 Normalized IPC speedup for SPEC 2006 and STREAM benchmarks  71
5.17 Performance sensitivity to mat size.  73
5.18 Performance sensitivity to n.  73
5.19 Performance comparison between different memory technologies.  74

6.1 Schematic view of a 3D-VRAM array with vertical access transistor  79
6.2 Voltage regions of sate write (perfect switching probability), unsafe write (not perfect switching probability), and safe read (no disturbance).  80
6.3 A 16-layer 3D-VRAM with low resistance $R_{on} = 25k\Omega$: (a) Worst-case voltage drop on furthest cell with various nonlinearity $K_r$; (b) read sensing margin with various $K_r$; (c) voltage drop breakdown of each element in a $128 \times 128 \times 16$ array with $K_r = 200$.  81
6.4 A 16-layer 3D-VRAM with nonlinearity $K_r = 5$: (a) Worst-case voltage drop on furthest cell with various nonlinearity $R_{on}$; (b) read sensing margin with various read voltage $V_R$.  82
6.5 Shmoo plot with different $R_{on}$ and nonlinearity in a $128 \times 128 \times 16$ array for (a) $V_R = 0.5V$ and (b) $V_R = 1.5V$.  83
6.6 Access energy of a 16-layer 3D-VRAM array with various $R_{on}$ for (a) writing a single bit in an array; and (b) reading a single bit in an array.  83
6.7 Multi-bit write operation in a 128x128x16 3D-VRAM array: (a) worst-case voltage drop and write current versus number of bits to write in parallel; (b) array write energy and write energy per bit versus number of bits to write in parallel.  84
6.8 Circuit optimizations for 3D-VRAM: (a) conventional write driver (WD) design: WDs are connected to one edge of the array; (b) proposed design: WDs are connected to the North, South, East, and West edges of the array; (c) remote sensing scheme: shared SAs within a block (adapted from [1] with modifications.  86
6.9 Hierarchial architecture of 3D-VRAM macro: (a) An ReRAM die consists of 4 planes, and each plane has 32 sub-planes; (b) Each sub-plane is made up of 8 blocks, the SAs and WDs within a block are shared among the 3D-VRAM arrays; (c) a detailed view of a 3D-VRAM array.  86
6.10 Multi-directional write driver design versus conventional design in
(a) Write margin of the 3D-VRAM array and (b) Current require-
ment of individual WD ........................................ 87
6.11 Area of 64Gb (left y-axis) chip and Cost per GB(right y-axis) at
$F = 30\text{nm}$ for 2D SLC NAND, 2D 3b/c MLC NAND, 16-layer SLC
NAND, 16-layer 2b/c NAND, planar ReRAM, baseline 3D-VRAM,
3D-VRAM with multi-directional writer driver design .................. 89
6.12 (a) Average page read energy and (b) Average page program energy
at $F = 30\text{nm}$ for a page size of 8KB in 2D SLC NAND, 2D 3b/c
MLC NAND, 16-layer SLC NAND, 16-layer 2b/c NAND, 2D SLC
NAND, 2D 3b/c MLC NAND, 16-layer SLC NAND, 16-layer 2b/c
NAND, planar ReRAM with single-bit access(SBC), 16-layer 3D-
VRAM with SBC, 16-layer 3D-VRAM with multi-bit access(MBC) .... 90
6.13 IOPS comparisons between NAND flash and ReRAM across differ-
ent applications .................................................. 91
6.14 (a) IOPS/$ and (b) IOPS/$/J (both normalized to 2D SLC NAND
for each application) .................................................. 92

7.1 (a) LRS of an ReRAM cell, (b) RESET operation, (c) SET opera-
tion, (d) HRS of an ReRAM cell ........................................ 96
7.2 Types of endurance failure (hard errors) in an ReRAM cell: (a)
Type I, (b) Type II, and (c) Type III ..................................... 98
7.3 Read noise margin degradation in various array sizes for (a) type I
failure, (b) type II failure, (c) type III failure ........................... 99
7.4 Worst-case voltage drop over cycles in various array size for (a) type
I failure, (b) type II failure ............................................. 100
7.5 Worst-case write energy per array in various array size for (a) type
I failure, (b) type II failure ............................................. 101
7.6 Hard error detection for type I failure .................................. 104
7.7 Percentage of available capacity versus the number of writes to
ReRAM for type I failure, with various array sizes of (a) 128x128,
(b)256x256, (c)512x512 ............................................. 105
7.8 Sustainable number of writes to ReRAM for type I failure with and
without considering postcycling retention degradation, for various
array sizes of (a) 128x128, (b)256x256, (c)512x512 ..................... 106
7.9 Hard error detection for type II failure .................................. 106
7.10 Percentage of available capacity versus the number of writes to
ReRAM for type II failure, with various array sizes of (a) 128x128,
(b)256x256, (c)512x512 ............................................. 107
7.11 Sustainable number of writes to ReRAM for type II failure with and without considering postcycling retention degradation, for various array sizes of (a) 128x128, (b) 256x256, (c) 512x512 . . . . . . . . 108

7.12 Hard error detection for type III failure. . . . . . . . . . . . . . . . 108

7.13 Percentage of available capacity versus the number of writes to ReRAM for type III failure, with various array sizes of (a) 128x128, (b) 256x256, (c) 512x512 . . . . . . . . . . . . . . . . 109

7.14 Sustainable number of writes to ReRAM for type III failure with and without considering postcycling retention degradation, for various array sizes of (a) 128x128, (b) 256x256, (c) 512x512 . . . . . . . . 109
## List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Array voltage and current in the V/2 scheme</td>
<td>21</td>
</tr>
<tr>
<td>3.2</td>
<td>HSPICE Subcircuit of the ReRAM model</td>
<td>25</td>
</tr>
<tr>
<td>3.3</td>
<td>Geometry Parameters of the 3D-VRAM Array</td>
<td>26</td>
</tr>
<tr>
<td>3.4</td>
<td>NVSim’s PCRAM model validation with respect to a 0.12μm 64Mb MOS-accessed PCRAM prototype chip [2]</td>
<td>35</td>
</tr>
<tr>
<td>3.5</td>
<td>NVSim’s PCRAM model validation with respect to a 90nm 512Mb diode-selected PCRAM prototype chip [3]</td>
<td>35</td>
</tr>
<tr>
<td>3.6</td>
<td>NVSim’s ReRAM model validation with respect to a 0.18μm 4Mb MOSFET-selected ReRAM prototype chip [4]</td>
<td>35</td>
</tr>
<tr>
<td>4.1</td>
<td>ReRAM technology assumptions</td>
<td>39</td>
</tr>
<tr>
<td>4.2</td>
<td>Comparison between 1T1R and cross-point structures</td>
<td>40</td>
</tr>
<tr>
<td>4.3</td>
<td>Three strategies in allocating 8 resistance levels (unit: Ω) ranging from $R_{\text{min}} = 1kΩ$ to from $R_{\text{max}} = 10MΩ$</td>
<td>41</td>
</tr>
<tr>
<td>4.4</td>
<td>Trade-offs in three schemes in choosing resistance levels</td>
<td>41</td>
</tr>
<tr>
<td>4.5</td>
<td>Average programming latency (T), energy (E) from $C_2C_1C_0$ (left column) to $T_2T_1T_0$ (top row) and retention time (s) of $T_2T_1T_0$ in fast programming strategy (upper row) and reliable programming strategy(lower row)</td>
<td>46</td>
</tr>
<tr>
<td>4.6</td>
<td>Technology survey of MLC ReRAM metrics in different material systems</td>
<td>48</td>
</tr>
<tr>
<td>5.1</td>
<td>Power/Area overhead of hardware implementations</td>
<td>65</td>
</tr>
<tr>
<td>5.2</td>
<td>System Configuration</td>
<td>68</td>
</tr>
<tr>
<td>5.3</td>
<td>Memory Timing Specifications (unit:ns)</td>
<td>69</td>
</tr>
<tr>
<td>6.1</td>
<td>Timing parameters for NAND flash and ReRAM</td>
<td>91</td>
</tr>
<tr>
<td>7.1</td>
<td>Parameters of a Cross-Point Array</td>
<td>99</td>
</tr>
</tbody>
</table>
My sincerest gratitude is to my advisor, Prof. Yuan Xie, who brought me into the exciting field of non-volatile memory research. His patience, constant encouragement helped me overcome many obstacles. Besides direct support and guidance on my research, he also provided me a lot of opportunities to meet with other leading experts from academia and industry. I cherish the wealth of knowledge and perspective I gained from him.

I am indebted to my dissertation co-advisor Professor Vijaykrishnan Narayanan for his sharp questions and insightful comments. I would like to thank my committee members Professor Mary Jane Irwin, and Professor Suman Datta for their invaluable advice. I owe them my sincere appreciation.

I would also like to thank my manager Dr. Norm P. Jouppi for their great support while I was an intern at Hewlett-Packard Labs for three summers.

I sincerely thank the co-authors of my research papers, Dr. Yiran Chen from University of Pittsburgh, Dr. Xiaochun Zhu and Dr. Seung H. Kang from Qualcomm, Dr. Christopher Hughes and Dr. Yen-Kuang Chen from Intel, Dr. Zhe Wang and Dr. Daniel A. Jimenez from Texas A&M University, Dr. Tao Wang from Peking University, for the great collaboration experience and their generous help.

Many thanks are to past and present MDL members for providing a supportive and productive environment. Especially, I thank Guangyu Sun, Xiaoxia Wu, Xiangyu Dong, Yibo Chen, Jin Ouyang, Dimin Niu, Tao Zhang, Jishen Zhao, Jing Xie, Matt Poremba, Hsiang-Yun Cheng, Jue Wang, Qiaosha Zou, Ping Chi, Jia Zhan, Ivan Stalev, Yang Zheng, Kaisheng Ma, Fen Ge, Hang Zhang, Karthik Swaminathan, and Mi Sun Park for their great help.

I truly appreciate my friends who make my life in Pennsylvania and Intern life in California so colorful and wonderful, and I will treasure those beautiful moments forever, including but not limited to: Zhiwang Zhang, Yu-San Lin, Ye Zhang, Muhuan Huang, Mingyu Gao, Feng Liu, Ke Chen, Sheng Li, Rob Schreiber, Parthasarathy Ranganathan, Jichuan Chang, Chao Ni, Yutian Lu, Liangliang
Zhang, Jieying Luo, Weiruo Zhang, Yangsen Kang, Mengying Zhang, Yumin Li. Special thanks are to my long-time old friends: Shimeng Yu, Xiaomu Wu and Zhaoyi Kang, they have stood by my side throughout all the peaks and valleys in my past years. Last but not least, I thank Xiaochen Zeng for making my life much more vivid than I could ever imagine.

Most importantly, I would like to express my deepest love to my family members: my father and mother, my grandfather and grandmother, my uncles and aunts, my cousins. They have shown me unconditional and unbounded love, and they always encourage me to pursue my dream. Their support is the strongest driving force for me to go forward.
Chapter 1

Introduction

The resistive random access memory (ReRAM) technology is an emerging candidate for next-generation non-volatile memory (NVM) architecture due to its simple structure, low programming voltage (< 3V), fast switching speed (< 10ns), high on/off ratio (> 10^2), excellent scalability (< 10nm), good endurance (10^6 – 10^{12} cycles), and great compatibility with the silicon CMOS technology [5]. Although the initial target of ReRAM is NAND flash replacement, ReRAM holds the potential to revolutionize the memory hierarchy from the last-level cache to mass storage system. Recently, the monolithic 3D integration of ReRAM at back-end-of-line (BEOL) is an attractive approach for enabling an ultra-high density cross-point architecture featuring 4F^2/N cell size (N is the 3D layer number) [6, 7]. However, most of ReRAM research has still been focused on device-level development. As the ultimate goal of ReRAM research is to advance ReRAM in current memory hierarchy, the key question is how to architect ReRAM in different levels of the memory hierarchy. The work in this dissertation aims to address these issues.

1.1 The Demand for New Memory Technologies

In conventional computer architecture design, SRAM, DRAM, and NAND flash memories are the common memory embodiments at different levels in the memory hierarchy, as working-class memory or storage-class memory. As technology scales, increasing leakage power dissipation and significant degradation of the reliability of SRAM and DRAM are of increasing concern, and the performance/bandwidth of
NAND flash is still limited \((< 50\text{MB/s})\) due to the slow block-based erasing operations. In recent years, we have seen a lot of efforts to address the research and development of emerging non-volatile memory (NVM) technologies, e.g., *Phase-Change RAM (PCM)* [8], *Spin-Transfer-Torque Magnetoresistive RAM (STT-RAM)* [9], and *Resistive RAM (ReRAM)*\(^1\) [5]. By combining the speed of SRAM, the density of DRAM, and the non-volatility of flash memory, these emerging NVM technologies have a great potential to be the universal memories of the future.

It is anticipated that the emerging NVM technologies will break important ground and move closer to the market in the near future [11]. As such emerging memory technologies are maturing, it is important for computer architecture designers to understand their pros and cons for improving the performance, power, or reliability of future many-core systems.

### 1.1.1 comparison of emerging NVM technologies

The emerging NVM technologies (STT-RAM, PCM, ReRAM) share some common features: they are two-terminal devices, and their states are differentiated by the switching between a high resistance state (off-state) and a low resistance states (on-state). The transition between the two states can be triggered by electrical voltage/current pulses. However, the detailed switching physics is quite different: STT-RAM relies on difference in resistance between the parallel configuration (on-state) and anti-parallel configuration (off-state) of two ferromagnetic layers separated by a thin tunneling insulator layer, which is called *Magnetic Tunneling Junction (MTJ)*; PCM relies on the switch between the crystalline phase (on-state) and the amorphous phase (off-state) of the GeSbTe material, which is thus called *GST node*; and ReRAM relies on the formation (on-state) and the rupture (off-state) of *Conductive Filament* in thin film oxides between two electrodes. Due to the different underlying physics, the device characteristics of these memory technologies are also different, and consequently each is suitable for different target applications. Figure 1.1 shows a comparison of device characteristics of traditional and emerging memory technologies

\(^1\)ReRAM sometimes is also called memristor [10], we use ReRAM to avoid any ambiguity.
<table>
<thead>
<tr>
<th></th>
<th>Conventional Memories</th>
<th>Emerging Memories</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SRAM</td>
<td>DRAM</td>
</tr>
<tr>
<td>Cell area</td>
<td>~150F²</td>
<td>6F²</td>
</tr>
<tr>
<td>Multi-bit</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Voltage</td>
<td>&lt;1V</td>
<td>&lt;1V</td>
</tr>
<tr>
<td>Read time</td>
<td>&lt;1ns</td>
<td>~10ns</td>
</tr>
<tr>
<td>P/E time</td>
<td>&lt;1ns</td>
<td>~10ns</td>
</tr>
<tr>
<td>Retention</td>
<td>N/A</td>
<td>~64ms</td>
</tr>
<tr>
<td>Endurance</td>
<td>&gt;1E16</td>
<td>&gt;1E16</td>
</tr>
<tr>
<td>Write Energy (per bit)</td>
<td>&lt;fJ</td>
<td>~10fJ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

F: feature size of the lithography used for patterning the cell

Figure 1.1. Device characteristics of conventional and emerging memory technologies.

1.1.2 advantages of ReRAM over STT-RAM/PCM

Although there exist a lot of research activities on STT-RAM/PCM in the last several years [8, 12, 9], ReRAM has gained significant attentions recently, mainly due to its multiple advantages over STT-RAM and PCM: (1) The storage node of conductive filament in ReRAM has better scalability than the MTJ in STT-RAM or GST in PCM; (2) ReRAM has much better write endurance than PCM as well as much larger capacity than both STT-RAM and PCM; (3) Last but not the least, as monolithic 3D integration is the key method to lower the cost per bit and enable a large capacity (as seen in the 3D NAND flash’s development), ReRAM is the only one that has the potential for the monolithic 3D integration among all the emerging NVM technologies². STT-RAM has very small on/off ratio (< 2) that has to use the transistor as the selection element, but it is challenging to build transistors in monolithic 3D integration. PCM has too large programming power that is hard to deliver into the 3D architecture and also the generated heat is hard to dissipate. Furthermore, the atomic-layer deposition (ALD) technique

²It should be pointed out the 3D integration in this proposal refers to the monolithic 3D integration at back-end-of-line (BEOL) instead of the 3D integration by wafer/die stacking techniques such as silicon-through-via (TSV). Only the monolithic 3D integration can achieve the ultra-low cost per bit.
makes ReRAM material uniform coating in high aspect ratio trench feasible, which is vital for building the low-cost 3D architecture. However, STT-RAM and PCM have no mature ALD solution yet. As a result, in the past few years, significant progresses of ReRAM technology have been made. Prototype chips have been demonstrated, e.g. ITRI reported a 4Mb HfOx macro with 7.2ns read/write random access time [4], Panasonic reported a 8Mb TaOx macro with 433Mb/s write throughput [13], and SanDisk/Toshiba reported a 32Gb macro using 24 nm technology [1], etc. These early studies show the feasibility for ReRAM to become a commercially viable memory technology.

1.2 Unique Challenges of ReRAM Development

ReRAM share some common design challenges with other emerging NVM technologies (STT-RAM and PCM) such as relatively longer write latency, larger write energy, and limited write endurance (as compared with SRAM). Therefore, many of the mitigation techniques that were developed for STT-RAM and PCM such as hybrid with mainstream memories [14], adaptive scheduling policy [15, 16, 17], read-before-write [18, 19], data encoding [20, 21, 22], wear-leveling [23, 24], fine-grained power management [25, 26], and hard error tolerance [27, 28, 29, 30, 31, 32] can be applied to the ReRAM design. However, owing to the underlying physics of the ReRAM device, new challenges arise.

- **Modeling and Memory Array Design.** As mentioned above, ReRAM relies on the formation and the rupture of conductive filament in thin film oxides between two electrodes to represent different resistance, and ReRAM is the only one that has the potential for the monolithic 3D integration among all the emerging NVM technologies. Consequently, the modeling of ReRAM device and the memory array design (including peripheral circuitry design) is fundamentally different from that for STT-RAM and PCM.

- **Reliability.** The sophisticated reliability failure mechanism add new challenges the robustness design. Soft error in ReRAM is typically the data retention failure due to elevated chip temperature, however it is recoverable by the next-time programming. Hard error in ReRAM is the endurance
degradation - the on/off resistance ratio drops permanently with cycling. There are quite a few types of failure modes after cycling, i.e., stuck at the on-state, stuck at the off-state, stuck at the intermediate-state, etc. It is found that the programming condition (the amplitude and width of the voltage/current pulse) has impact on the failure modes [33]. Also there is a trade-off between the retention and endurance: better retention comes with poorer endurance [34].

• Data Pattern Dependency for Access Latency. The switching dynamics of ReRAM have a unique voltage-time relationship. The switching time exponentially increases as the applied voltage decreases linearly, which is almost universally observable in ReRAM devices [5]. The existence of voltage drop on the interconnect resistance is significant when large writing current is passing through the metal wires. Therefore, the actual voltage dropped on each cell in an array varies from one another, resulting in quite different switching time because of the exponential voltage-time relationship. This means that the worst-case switching time of an ReRAM array depends on the data pattern, array size, write current, metal resistance, and the number of bits being written in parallel.

• Variability. The variability of ReRAM in the switching parameters (i.e., the on and off resistances) is different and much worse than that in STT-RAM and PCM. The uniqueness of ReRAM’s variability is that it has two types: not only the spatial variation (from device to device) but also the temporal variation (from cycle to cycle). The spatial variation comes from the process variation that commonly occurs in today’s nanoscale devices. However, the temporal variation occurs in a single device due to stochastic nature of the conductive filament formation or rupture [35]. Due to the randomness of the oxygen vacancy generation and ions migration, the shape of the conductive filament varies from cycle to cycle even under the same programming condition. Such variation can be remarkable, e.g. the temporal variation in the off resistance may expand 3-4 orders of magnitude even in state-of-the-art ReRAM prototype [36]. This variability in resistance has two effects: first, if the resistance in the on-state is not sufficiently low or that in
the off-state is not sufficiently high, the sense margin shrinks; second, if the resistance in the on-state is too low or in the off-state is too high (so-called over-write [37]), write failure may happen in the next cycle because the same programming condition cannot switch it. It should be noted that such write failure is not a hard error, because if changing the programming condition (e.g. increasing the amplitude or width of the voltage pulse), the cell is able to operate normally. Therefore, we call such type of failure as pseudo-hard error, which is unique in ReRAM owing to the significant variability.

• **Read/Write Asymmetry.** ReRAM shows continuous improvement on the write latency. The best reported switching time of ReRAM is 300 ps [37], which is much smaller than the typical sensing time of a ReRAM cell (a few ns to 10’s ns, which is limited by the sense amplifier design for sensing low current). This causes possible slower read than write in ReRAM [4], which is opposite to other NVM technologies such as NAND flash and PCM.

Overall, these complexities in variability, the reliability, the switching dynamics add new challenges to the ReRAM design. In this dissertation, we tackle some of challenges at circuit/architecture-level.

### 1.3 Roles of ReRAM in Memory Hierarchies

The huge design space of ReRAM covers a broad spectrum from highly latency-optimized microprocessor caches to highly density-optimized secondary storage. For each replacement target, i.e. last-level cache, main memory, solid state storage, etc., the requirement on device metrics, array organizations, and architectural optimizations is quite different.

**Cache replacement:** The scaling of SRAM is increasingly constrained by technology limitations such as leakage power and cell density. Compared to SRAM (or embedded DRAM), emerging NVM technologies have advantages in higher density, lower standby power, better scalability and nonvolatility. Compared to STT-RAM, ReRAM has higher density (even in "1T1R" structure because of lower switching current) and better readability due to larger on/off ratio. However, similar to
PCM, ReRAM has an endurance issue, although its endurance is orders of magnitude higher than that of PCM. For example, recent ReRAM technologies has demonstrated endurance of $10^{11} \sim 10^{12}$ [38, 39]. Some wear-leveling techniques have been proposed for NVM-based main memories [23, 24]. Most of them can be adopted in ReRAM cache design based on the analogy of one set in cache and one row in main memory. One unique observation in cache design is that its intra-set write variations are also significant: the distributions among different ways in a given set are unbalanced. Prior work [40] has shown that such approach can extend the lifetime of an ReRAM cache by devising a variation-aware cache replacement policy.

Main memory replacement: The scalability of DRAM faces challenges such as refresh power consumption and the difficulty of building high aspect ratio capacitors. Compared to PCM based main memory [19], ReRAM has superior characteristics such as higher density, much lower write energy, and higher write bandwidth, making it a very attractive cost-efficient alternative for main memory. However, new challenges that are unique to ReRAM arise, e.g. the write latency of cross-point ReRAM is susceptible to the data pattern of a write operation. Special constraints come from the operating voltage of cells, their write latency, and row buffer size. All these combined with the exponential voltage-time relationship in ReRAM switching dynamics provide us opportunities to address them through architectural enhancements. In this dissertation, we architect 2D cross-point ReRAM for main memory replacement, and the design details will be discussed in Chapter 5.

Solid-state storage replacement: Compared to NAND flash, ReRAM has much better cell-level characteristics such as many orders of magnitudes lower write latency and higher endurance. The 3D vertical ReRAM proposed here makes ReRAM a competitive candidate for replacing NAND flash in terms of cost per bit. However, there are several issues that need to be addressed when targeting an ultra-high density memory with a large page size ($> KB$). First, the sensing resource is very scarce in highly density optimized design, and this limits the degree of parallelism of a read operation. In order to access a large trunk of data in 3D ReRAM, we might have to serialize the read operation, which in turn increase the macro-level access latency. Second, when activating a cross-point array, every cell in the cross-point array will consume energy due to the sneak paths,
resulting in a large activation energy of an array. This puts another constraint on the number of arrays that can be activated at the same time. Therefore, to make a fair comparison between the bandwidth of ReRAM and NAND flash, the advantage of ReRAM’s cell-level characteristics and the disadvantage of ReRAM’s limited parallelism should be both considered. In this dissertation, we architect 3D vertical ReRAM for storage replacement, and the design details will be discussed in Chapter 6.

1.4 Organization of This Dissertation

The fundamental goal of this Dissertation is to tackle some of the aforementioned challenges of ReRAM at the circuit/macro level, and also explore the ReRAM’s early design space at the architecture/system level. In this dissertation, we exploit a cross-layer hierarchical design approach from the device (modeling), circuit (design) to architecture (optimization) to advance state-of-the-art of the ReRAM technology in main memory or storage system.

First, several array/marco models with different accuracy and simulation speed are built and described in Chapter 3. Second, circuit-/architecture-level techniques that improve the density of ReRAM prototypes are proposed and evaluated in Chapter 4. Third, architectural-level case studies of adopting ReRAM are conducted. The case studies include DRAM replacement using 2D cross-point ReRAM in Chapter 5, NAND flash replacement in Chapter 6. Fourth, reliability of cross-point ReRAM design issues are addressed in Chapter 7.
Technology Background

2.1 Emerging Non-Volatile Memory Technology

In this section, the fundamentals of several emerging non-volatile memory technologies will be introduced.

2.1.1 spin-torque-transfer RAM

STT-RAM uses magnetic tunnel junction (MTJ) to store bit information by its two different resistance states. As shown in Figure 2.1, MTJ usually contains two ferromagnetic layers. One ferromagnetic layer is has fixed magnetization direction and it is called the reference layer, while the other layer has a free magnetization direction that can be changed by passing a write current and it is called the free layer. The relative magnetization direction of two ferromagnetic layers determines the resistance of MTJ. If two ferromagnetic layers have the parallel directions, the resistance of MTJ is low, indicating a “1” state; if two layers have anti-parallel directions, the resistance of MTJ is high, indicating a “0” state.

As shown in Figure 2.1, there are two possible schemes to stack MTJ atop access NMOS transistor. Conventionally, the free layer of MTJ is connected to bitline (BJ). In that scheme, when writing “1” state into STT-RAM cells, positive voltage difference is established between BL and SL and the switching current required is $I_c(\text{AP} \to P)$; when writing “0” state, negative voltage difference is established between BL and SL and the switching current required is $I_c(P \to$
Figure 2.1. Demonstration of a STT-RAM cell: (a) Conventional connection scheme; (b) Reverse connection scheme.

While the reverse connection scheme was proposed in [41] where the free layer of MTJ is connected to the drain of NMOS instead of BL. [41] argues that $I_c(P \rightarrow AP)$ is normally significantly larger than $I_c(AP \rightarrow P)$ [42] due to the inherent torque asymmetry of MTJ. But the SL-to-BL current is much smaller than the BL-to-SL current under the same wordline voltage and voltage difference between BL and SL because the body effect of access transistor degrades the SL-to-BL current remarkably. Thus reserving connection scheme can relax the sizing requirement on access transistor, which results in more compact STT-RAM cell size. However, device-level efforts have been put to improve the asymmetry of switching characteristic of MTJ and $I_c(AP \rightarrow P)$ slightly larger than $I_c(P \rightarrow AP)$ was even demonstrated in [43]. In our work, we always choose the MJT connection scheme that is responsible for relaxed sizing requirement on access transistor.

Another important metric for an MTJ is the tunnel magnetoresistance (TMR) which is defined as,

$$TMR = \frac{R_{AP} - R_{P}}{R_{P}}$$

(2.1)

where $R_{AP}$ is the electrical resistance in the anti-parallel state, whereas $R_{P}$ is the resistance in the parallel state. A large TMR means big gap between low resistance state (LRS) and high resistance state (HRS), which could essentially brings faster read sensing latency or relaxes the constraint for sense amplifier design. It’s critical to introduce an equivalent metric for a STT-RAM cell which contains both the
MTJ and the access transistor. Similarly the cell TMR (CTMR) is defined as,

\[
CTMR = \frac{R_{\text{cell,AP}} - R_{\text{cell,P}}}{R_{\text{cell,P}}}
\]  

(2.2)

where \( R_{\text{cell,AP}} \) is the total cell resistance when the MTJ is in the anti-parallel state, whereas \( R_{\text{cell,P}} \) is the total cell resistance when the MTJ is in the parallel state. CTMR can be expressed by another equation,

\[
CTMR = \frac{I_P - I_{AP}}{I_{AP}}
\]  

(2.3)

where \( I_{AP} \) and \( I_P \) are the currents for reading “0” state and “1” state. If we ignore the resistance difference of the access transistor for reading “0” state and “1” state. CTMR can be interpreted as,

\[
CTMR = \frac{R_{AP} - R_P}{R_P + R_{NMOS}} = \frac{R_{AP} - R_P}{R_P + C/W}
\]  

(2.4)

where \( R_{NMOS} \) is the equivalent resistance of access NMOS transistor and \( W \) is the transistor width, \( C \) is a constant related to the wordline voltage and threshold voltage of the transistor. From equation 2.4 we can conclude that sizing up the access transistor will make CTMR close to the inherent TMR of MTJ.

### 2.1.1.1 perpendicular MTJ

A key challenge for MTJ design is to reduce switching current while maintaining sufficiently high thermal stability in order not to affect data retention time and write/read errors. The conventional in-plane MTJ critical switching current \( I_{c0} \) divided by the thermal barrier \( \Delta \) can be expressed as in [44],

\[
\frac{I_{c0}}{\Delta} = \frac{\alpha}{\eta} \times (1 + \frac{H_d}{2H_k})
\]  

(2.5)

where \( \alpha \) is the damping constant, \( \eta \) is the STT efficiency, \( H_d \) is the out of plane demagnetization field, and \( H_k \) is the in-plane anisotropy field dominated by the shape anisotropy. The typical value of \( H_d/2H_k \) is about 20-150 [9]. From Equation 2.5 we can see that the magnetization has to overcome a very large out-of-plane demag-
Figure 2.2. The schematic view of a PCM cell with a MOSFET selector transistor (BL=bitline, WL=wordline, SL=sourceline).

netizing field before it can switch to the opposite direction. However, only \( H_k \) not \( H_d \) will contribute to thermal stability [44]. Perpendicular MTJ were investigated as a promising solution [44, 45] as the critical switching current of PMTJ can be described as,

\[
\frac{I_{c0}}{\Delta} = \frac{\alpha}{\eta}
\]  

(2.6)

Therefore, PMTJ can have much smaller switching current than in-plane devices if the same ratio of \( \alpha/\eta \) can be maintained. Indeed, very low switching current density of MTJ was demonstrated while maintaining high enough thermal stability factor [45]. There are some issues to be solved for PMTJ such as degraded compatibility with CMOS process, relative large damping constant, and potential lattice mismatch for high TMR ratio and STT efficiency.

2.1.2 phase change memory

PCM uses chalcogenide material (e.g. GST) to store information. The chalcogenide materials can be switched between a crystalline phase (SET state) and an amorphous phase (RESET state) with the application of heat. The crystalline phase shows low resistivity while the amorphous phase is characterized by high resistivity. Fig. 2.2 shows an example of a MOS-accessed PCM cell.

The SET operation crystallizes GST by heating it above its crystallization temperature, and the RESET operation melt-quenches GST to make the material amorphous as illustrated in Fig. 2.3. The temperature is controlled by passing a specific electrical current profile and generating the required Joule heat. High-power pulses for the RESET operation to heat the memory cell above the GST
melting temperature. In contrast, moderate power but longer duration pulses for the SET operation to heat the cell above the GST crystallization temperature but below the melting temperature. The large resistance contrast between the RESET and SET states (e.g. $10^2 - 10^3$) makes multi-level cell (MLC) PCM feasible. The degree of success of such an MLC write depends on the resistance distributions over a large ensemble of PCM cells. Unlike single-level cell (SLC) write, where the bit write quality can be ensured by over-SET or over-RESET, the intrinsic randomness associated with each write attempt and the inter-cell variability make it impractical to have a universal pulse shape for writing an intermediate state. In order to address this issue, resistance distribution tightening techniques have been developed based on the write-and-verify technique. In order to achieve non-overlapping resistance distributions of different bit levels, write-and-verify needs to iteratively apply partial set pulses and then verify that a specialized precision criterion is met, which leads to much longer write latency and hence the much larger programming energy.

2.2 ReRAM Cell Basics

A ReRAM cell has a very simple structure: a metal-oxide material is sandwiched between two metal layers of electrodes, named top electrode (TE) and bottom electrode (BE). As the size of the sandwiched layer is limited to the cross-sectional area between the vertical and the horizontal metal wires, it has the smallest possible cell size of $4F^2$. Similar to PCM, a low resistance state (LRS or ON-state) and
Figure 2.4. Schematic view of a RRAM cell

Figure 2.5. Multi-level switching in ReRAM: (a) H2L and (b) L2H programming.

high resistance state (HRS or OFF-state) are used to represent the logical ‘1’ and ‘0’ respectively. In order to switch a ReRAM cell between the LRS and HRS, an external voltage with specified polarity, magnitude, and duration is applied to the sandwiched layer. The switching of LRS-to-HRS is called a RESET operation and the switching of HRS-to-LRS is called a SET operation. But, unlike PCM, there is no significant latency or energy difference between a SET and a RESET operation.

When a voltage is applied across a ReRAM cell, depending upon the voltage polarity, one or more conductive filaments (CF) made out of oxygen vacancies is either formed or ruptured. Once CFs are formed inside the metal oxide to bridge the top and bottom electrodes, current can flow through CF, and the cell is in a
Figure 2.6. Write-and-verify in H2L programming

low resistance state (LRS). The larger the size of CFs, the lower the resistance. Figure 7.1(a) illustrates the formation of CFs in a bipolar ReRAM cell. Conversely, the rupture of CFs disconnects the top electrode from the bottom electrode, resulting in a high resistance state (HRS) of the cell. When a positive current passes through the cell, the oxygen atoms are knocked out of the lattice and become negatively-charged oxygen ions. Under a positive electric field, the oxygen ions will drift towards the anode, leaving corresponding oxygen vacancies in the metal oxide layer. Figure 7.1(a) also shows that the size of CFs is directly related to the value of the current, and by changing the strength of CFs, we can control the cell resistance. Thus we can program ReRAM to intermediate levels between the highest resistance state and the lowest resistance state by adjusting the programming current.

Similar to MLC PCM, programming to intermediate states of a ReRAM cell also uses write-and-verify, starting from either the highest resistance state (H2L programming) or the lowest resistance state (L2H programming). This multi-step process, illustrated in Figure 2.6, helps tolerate both temporal and spatial process variations.

2.3 ReRAM Array Structures

There are several typical array structures for ReRAM with trade-offs in density, latency, and reliability.
2.3.1 1T1R structure

Conventionally, memory cells are connected together to form an array and isolated by using MOS access devices as illustrated in Figure 2.7. In the MOS-accessed array structure, a.k.a 1T1R structure, the cell size is dominated by the large MOS device that is necessary to drive enough write current even though the storage element (MTJ, GST or MIM) itself is much smaller (shown in Figure 2.8. The driving current of NMOS, $I_{DS}$ can be first-order estimated as follows\(^1\),

$$I_{DS} = K \frac{W}{L} \left[ (V_{GS} - V_{TH}) V_{DS} - \frac{V_{DS}^2}{2} \right]$$  \hspace{1cm} (2.7)

if NMOS is working at the linear region; or calculated by

$$I_{DS} = \frac{K W}{2} \left( V_{GS} - V_{TH} \right)^2 (1 + \lambda V_{DS})$$  \hspace{1cm} (2.8)

if NMOS is working at the saturation region. Hence, no matter in which region NMOS is working, the current driving ability of NMOS is proportional to its width-to-length (W/L) ratio, which determines the NMOS size.

2.3.2 cross-point structure

Cross-point cell corresponds to the 1D1R (1-diode-1-resistor) [46, 1] or 0T1R (0-transistor-1-resistor) [47, 48] structures used by several high-density NVM chips.

---

\(^1\)Equations 2.7 and 2.8 are for long-channel drift/diffusion devices, the equations are subjected to change depending on the technology, though the proportional relationship between the current and W/L still hold for very advanced technologies.
Fig. 2.9 shows a cross-point array without diodes (i.e. 0T1R structure). For 1D1R structure, a diode is inserted between the word line and the storage element. Such cells either rely on the one-way connectivity of diode (i.e. 1D1R) or leverage materials’ non-linearity (i.e. 0T1R) to control the memory access path. As illustrated in Fig. 2.10, the widths of word lines and bit lines can be the minimal value of 1F and the spacing in each direction is also 1F, thus the cell size of each cross-point cell is,

\[
\text{Area}_{\text{cell,cross-point}} = 4(F^2)
\]  

(2.9)

Compared to 1T1R, cross-point cells have worse cell isolation but provide a way of building high-density memory chip because they have much smaller cell sizes. In some cases, the cross-point cell size is constrained by the diode due to limited current density.

### 2.3.3 3D ReRAM structure

To further improve the bit density of ReRAM, many 3D structures have been proposed and demonstrated [49, 6, 50, 51]. One straightforward approach is to stack planar cross-point structure layer by layer, namely 3D-HRAM, as shown in Figure 2.11. The adjacent layers share their wordlines and bitlines alternatively. Chen et al. [49] discussed the addressing scheme of 3D-HRAM. To maximize the density of 3D-HRAM, stacking more layers is desired. However, every additional layer introduces extra fabrication process steps, including lithography, etching and chemical-mechanical planarization (CMP).
As an alternative 3D ReRAM solution, 3D-VRAM was proposed to reduce the fabrication steps for high density ReRAM design. The schematize view of the 3D-VRAM architecture is illustrated in Figure 6.1. Each ReRAM array consists of $L$ wordline planes, $N_b$ bitlines and $N_s$ sourcelines. Two adjacent wordline plane electrodes are separated by a dielectric isolation layer. The cell is now located at every cross point of a vertical pillar electrode and a wordline plane. The key cost saver of 3D-VRAM is the elimination of the critical lithography and etching steps of the intermediate layers. The wordline planes and isolation layers are deposited consecutively. The process of defining the pillar electrodes and cells is involved only after the top most layer is deposited, and only two critical lithography and etching steps are required (one for patterning the pillar electrode, one for opening the contact for wordline planes). Chen et al. [50] have demonstrated the detailed fabrication process.

To address such an array, one access transistor is introduced at the bottom of each vertical pillar electrode. During a write operation, $V_g$ is applied on one
selected sourceline to turn on the $N_b$ access transistors alone the selected sourceline while all the other transistors remain off by grounding the unselected sourcelines. This operation basically activates a vertical plane, which is a de facto cross-point structure. Therefore normal voltage biasing schemes for writing and reading a cross-point structure can be applied on the activated plane.

During a read or write access, only one sourceline is selected to activate a vertical plane of cross-point structure. Within the cross-point structure, the voltage biasing for write and read operation are similar to that of a 2D cross-point structure, as shown in Figure 2.13.
Modeling of ReRAM

In this chapter, several models of ReRAM with different accuracy and speed will be introduced. We will also briefly talk about the modeling of 3D vertical ReRAM. Then a circuit-level modeling framework will be presented.

3.1 A Simplified Model for Cross-Point ReRAM

Write half-select problem is endemic in a cross-point array structure. When a memory cell is selected during a write operation, other memory cells connected to the selected row and columns are subject to the voltage bias and defined as half-select cells. In this work, we analyze the write half-select problem in $V/2$ biasing scheme [52], which is the common accessing method in a cross-point structure. Table 3.1 lists the details of $V/2$ biasing.

In the $V/2$ biasing scheme, the write operation should guarantee that the states of these half selected memory cells do not change over a specified number of write cycles or a specified time with $V/2$ applied. The half-select cells serve as current dividers in the selected row and columns, preventing the array size from growing unbounded since the available driving current is limited. The minimum current that a column write driver should provide is determined by the case when all the half-select cells are in their low resistance state (LRS),

$$I_{\text{driver}} = I_{\text{reset}} + (N_r - 1) \times I(V_{\text{reset}}/2)$$  \hspace{1cm} (3.1)

$N_r, N_c,$ and $N_{sc}$ are the numbers of total rows, total columns, and selected columns per row
Table 3.1. Array voltage and current in the V/2 scheme

<table>
<thead>
<tr>
<th>Item</th>
<th>Voltage</th>
<th>Current</th>
</tr>
</thead>
<tbody>
<tr>
<td>Selected row</td>
<td>V</td>
<td>$N_{sc} \times I(V) + (N_c - N_{sc}) \times I(V/2)$</td>
</tr>
<tr>
<td>Selected column</td>
<td>0</td>
<td>$-[I(V) + (N_r - 1) \times I(V/2)]$</td>
</tr>
<tr>
<td>Unselected row</td>
<td>V/2</td>
<td>$N_{sc} \times I(V/2)$</td>
</tr>
<tr>
<td>Unselected column</td>
<td>V/2</td>
<td>$-I(V/2)$</td>
</tr>
</tbody>
</table>

where $I_{reset}$ and $V_{reset}$ are the reset current and reset voltage.

Non-linearity of a memory cell is reflected by the fact that the current through the memory cell is not directly proportional to the voltage applied on it, which means non-constant resistance of the memory cell. We here define a non-linearity coefficient to quantify the current divider effect of the half selected memory cells as follows,

$$K_r(p; V) = p \times \frac{R(V/p)}{R(V)} \tag{3.2}$$

where $R(V/p)$ and $R(V)$ are equivalent static resistance of the memory cell biased at $V/p$ and $V$, respectively.

Then, we derive the upper limit in a cross-point memory array size when using the V/2 scheme by examining the worst case scenario, i.e., all the cells connected to the selected row/column are in LRS:

$$N_r = \left( \frac{I_{driver}}{I_{reset}} - 1 \right) \times K_r(2, V_{reset}) + 1 \tag{3.3}$$
$$N_c = \left( \frac{I_{driver}}{I_{reset}} - N_{sc} \right) \times K_r(2, V_{reset}) + N_{sc} \tag{3.4}$$

where $I_{driver}$ is the maximum driving current that the write driver attached to the selected row/column can provide. Thus, $N_r$ and $N_c$ are the maximum numbers of rows and columns in a cross-point array.

3.2 An Improved Analytical Model for Detailed Array Analysis

For the sake of simplicity, some prior work [53] also develops similar simplified models that use a linear resistor (either HRS or LRS) to represent the ReRAM
element in a cross-point structure. Such an approach results in an significant simulation error of the sneak current and voltage drop when the cell has a large nonlinearity. In this section, we present an improved version of a mathematical model of the cross-point array.

The basic circuit model of an $M$ by $N$ cross-point ReRAM array is shown in Figure 3.1. This model is built upon Kirchhoff’s Current Law (KCL) and its validity can be guaranteed by deductions from the basic circuit theory. The horizontal lines are wordlines and vertical lines represent bitlines. The ReRAM cells are located at each cross-point of wordline and bitlines.

A detailed cross-point is also shown in Figure 3.1(b). The resistance of the ReRAM cell at the cross-point of $i^{th}$ wordline and $j^{th}$ bitline is represented by $R_{i,j}$. We assume the resistance of the wire connecting two cross-points to be $R_{\text{line}}$. The input resistance of each wordline or bitline driver is $R_{\text{v}}$ and the resistance of sense amplifier is $R_{s}$. In order to set up the KCL equations, the voltage at each cross-point is indicated as $V_{i,j}$ for wordline layer and $V'_{i,j}$ for bitline layer. In addition, the input voltage for the $i^{th}$ wordline is $V_{Wi}$ and for the $i^{th}$ bitline is $V_{Bi}$. In the case that a wordline takes input from both the sides, the voltage at the other end of the $i^{th}$ wordline is represented as $V'_{Wi}$.

3.2.1 mathematical description of the model

Based on this model, the current equations for each cross-point can be set following KCL: $\Sigma_{k=1}^{k} I_k = 0$. All of the cross-points have similar structure with no more than
three current branches and therefore it is very easy to set up the KCL equations for each cross-point. However, we should treat the cross-points at the edges of the array specifically because KCL equations for these cross-points vary with different write/read schemes. For example, the unselected wordline for write operation can be either half biased or left floating. Thus, the edge conditions should be adjusted according to each write/read scheme. In particular, all of the cross-points in an array can be classified into three major categories: normal point, activated point and floating point.

The normal points are located inside the memory array. In other words, for all of the nodes with $1 < i < m$ and $1 < j < n$, the KCL equations take the form of

$$R_1^{-1}V_{i,j-1} - (2R_1^{-1} + R_{i,j}^{-1})V_{i,j} + R_1^{-1}V_{i,j+1} + R_{i,j}^{-1}V'_{i,j} = 0,$$  \hspace{1cm} (3.5)

for the node at wordline layer and

$$R_1^{-1}V'_{i-1,j} - (2R_1^{-1} + R_{i,j}^{-1})V'_{i,j} + R_1^{-1}V'_{i+1,j} + R_{i,j}^{-1}V_{i,j} = 0.$$  \hspace{1cm} (3.6)

for the node at bitline layer.

The activated point and floating point represent the nodes at the edge of cross-point array with different conditions: an edge point, which is directly connected to the voltage input or to the ground, can be considered as an activated point. Otherwise, it is a floating point. For example, consider the point located at the intersection of $i^{th}$ wordline and $1^{st}$ bitline. If the $i^{th}$ wordline is activated by an input voltage of $V_{Wi}$, this cross-point is an activated point, and the KCL equation for this point is:

$$-(R_v^{-1} + R_l^{-1} + R_{i,1}^{-1})V_{i,1} + R_l^{-1}V_{i,2} + R_{i,1}^{-1}V'_{i,1} = -R_v^{-1}V_{Wi}.$$  \hspace{1cm} (3.7)

Otherwise, it is floating and its KCL equation is

$$-(R_l^{-1} + R_{i,1}^{-1})V_{i,1} + R_l^{-1}V_{i,2} + R_{i,1}^{-1}V'_{i,1} = 0.$$  \hspace{1cm} (3.8)

For clarity, a $2mn \times 1$ vector $V$ is defined to represent all of the variables in
the KCL equations:

\[ V = [V_1^T, V_2^T, \ldots, V_m^T, V_1'^T, V_2'^T, \ldots, V_m'^T]^T, \]  \hspace{1cm} (3.9)

where,

\[ V_i = [V_{i,1}, V_{i,2}, \ldots, V_{i,n}]^T, \quad V_i' = [V_{i,1}', V_{i,2}', \ldots, V_{i,n}']^T, \]  \hspace{1cm} (3.10)

for \( i = 1, 2, \ldots, m \). Then all of the KCL equations can be considered as a system of linear equations, which has the form

\[ A \cdot V = C. \]  \hspace{1cm} (3.11)

\( A \) is a \( 2mn \times 2mn \) coefficient matrix, which is determined by Equations (3.5)-(3.8). \( C \) is a \( 2mn \times 1 \) vector, containing the constant terms of these equations. As shown, all of the KCL equations have simple structure and are similar to each other. Therefore, the linear equation system has a relatively fixed format and simple structure, making it easy to establish and adjust the coefficients and constants according to different design schemes. Besides, due to the simplicity of the KCL equation, \( A \) is populated primarily with zeros and can be saved as a sparse matrix, which will further reduce the storage cost during the computation.

3.3 Modeling of 3D vertical ReRAM

In this section, the modeling of 3D vertical ReRAM will be discussed.

3.3.1 modeling of a ReRAM element

Even the improved analytical model takes the nonlinearity into consideration, the resistance of half-selected cells is simply multiplied by a nonlinearity constant. As a result, the error can still vary depending on the shape of I-V curve of the cell. One alternative approach is to build a SPICE-compatible model [54] for ReRAM with full dynamics by incorporating the differential equations for the state variable of a ReRAM cell, which produces accurate results but the run time could go unbounded when simulating an array with \( > 10^5 \) cells. To maintain both good accuracy and
Table 3.2. HSPICE Subcircuit of the ReRAM model

*Using a behavior current source to model the nonlinear I-V curve
.param I0 = 1e-3, g0 = 2.5e-10, V0 = 0.25
.subckt reram top bot
Gram top bot CUR = 'I0*exp(-g/g0)*sinh (V(top,bot)/V0)'
*g is the tunneling gap distance
*I0, g0, V0 are fitting parameters
.ends reram

Figure 3.2. Circuit model: abstraction of a 2-layer 3D-VRAM array

Simulation speed, we implement the representative I-V relationship of a typical ReRAM cell, based on the experimental results in Yu et al’s work [55], as an HSPICE subcircuit. We take out the equations of the switching dynamics which is the most time-consuming part of the simulation, as the focus of this work is to perform DC analysis of 3D-VRAM. The description of the ReRAM element in HSPICE is shown in Table 3.2.

3.3.2 modeling of a 3D-VRAM array

We develop a circuit model of the 3D-VRAM array by approximating the ReRAM cells and plane resistance with segmented elements. As Figure 3.2 shows, each ReRAM cell is represented by four ReRAM elements, defined in Section 3.3.1. One advantage of 3D-VRAM over 3D-HRAM is that the wordline is a metal plane rather than multiple metal wires, making the effective resistance between adjacent cells smaller than the wire resistance in the 3D-HRAM counterpart. To model such effect, one virtual node is added to emulate the two-dimensional current flow through the wordline plane using discrete resistors. We also tried to add four
Table 3.3. Geometry Parameters of the 3D-VRAM Array

<table>
<thead>
<tr>
<th>Metric</th>
<th>Description</th>
<th>Explored Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>$H_m$</td>
<td>Height of a wordline plane</td>
<td>20, 30, 40nm</td>
</tr>
<tr>
<td>$H_i$</td>
<td>Height of an isolation layer</td>
<td>20nm</td>
</tr>
<tr>
<td>$H_s$</td>
<td>Height of a vertical stack</td>
<td>-</td>
</tr>
<tr>
<td>$T_{ox}$</td>
<td>Thickness of the switching layer</td>
<td>5nm</td>
</tr>
<tr>
<td>$D$</td>
<td>Diameter of a pillar electrode</td>
<td>-</td>
</tr>
<tr>
<td>$F$</td>
<td>Feature size of the design</td>
<td>22nm</td>
</tr>
<tr>
<td>$P$</td>
<td>Minimum distance from cell to cell</td>
<td>-</td>
</tr>
<tr>
<td>$W_{tran}$</td>
<td>Gate width of an access transistor</td>
<td>22, 44, 66nm</td>
</tr>
<tr>
<td>$L_{trans}$</td>
<td>Gate length of an access transistor</td>
<td>22nm</td>
</tr>
<tr>
<td>$W_{cell}$</td>
<td>Cell width along bitline direction</td>
<td>-</td>
</tr>
<tr>
<td>$L_{cell}$</td>
<td>Cell length along sourceline direction</td>
<td>-</td>
</tr>
<tr>
<td>$AR$</td>
<td>Etching aspect ratio</td>
<td>10, 20, 30</td>
</tr>
<tr>
<td>$N_s$</td>
<td>Number of sourcelines per array</td>
<td>16 ~ 256</td>
</tr>
<tr>
<td>$N_b$</td>
<td>Number of bitlines per array</td>
<td>16 ~ 256</td>
</tr>
<tr>
<td>$L_e$</td>
<td>Number of vertical stacks</td>
<td>2 ~ 64</td>
</tr>
</tbody>
</table>

and nine virtual nodes and our results will show that the error of adding one virtual node is already very small. Not shown in Figure 3.2, access transistors are implemented below the bottom layer using 22nm PTM model[56]. In order to minimize the voltage drop on the access transistor, we assume that the gate voltage is boosted when the NMOS is on.

### 3.3.3 geometry parameters and design constraints

To enable the exploration of the large design space, we parameterize some of the important geometries in the 3D-VRAM array and summarize them in Table 7.1. Under these definitions, the height of a vertical stack (one wordline layer plus one isolation layer) is $H_s = H_m + H_i$. The pitch is defined as the minimum distance from the center of a cell to the center of its neighboring cell,

$$P = D + 2T_{ox} + F$$ (3.12)

#### 3.3.3.1 modeling 3D-VRAM with planar access transistor

When modeling the cell size, the width of a cell is bounded by either the pitch or the width of the underlying access transistor assuming standalone memory design rule,

$$W_{cell} = \max(P, W_{tran} + F)$$ (3.13)
and the length of a cell is also bounded in the similar manner,

\[ L_{cell} = \max(P, L_{tran} + 2F) \]  

(3.14)

where \( L_{tran} \) is typically assumed to be a fixed value (\( F \) in this work).

The etching aspect ratio defines the maximum ratio of the total height of vertical stacks to the diameter of a pillar electrode,

\[ AR = \frac{H_s \times L}{D + 2T_{ox}} \]  

(3.15)

From equation (3.19) to equation (3.20) we get,

\[
A_{cell} = \begin{cases} 
  P^2 & \text{if Case 1} \\
  P \times (W_{tran} + F) & \text{if Case 2} \\
  3F \times (W_{tran} + F) & \text{if Case 3}
\end{cases}
\]  

(3.16)

and the conditions for the 3 cases are,

\[
Cond = \begin{cases} 
  Case1 : & \frac{H_s L}{AR} \geq \max(W_{tran}, 2F) \\
  Case2 : & (\frac{H_s L}{AR} - 2F)(\frac{H_s L}{AR} - W_{tran}) < 0 \\
  Case3 : & \frac{H_s L}{AR} \leq \min(W_{tran}, 2F)
\end{cases}
\]  

(3.17)

Equation (3.16) indicates that the cell size is bounded by the etching aspect ratio when building a 3D-VRAM array with many stacks and/or a thick metal layer (case 1). In case 1, we can afford to increase the width of the underlying transistor without increasing the cell size. The up-sizing in turn relaxes the design constraints from the perspective of sneak current and voltage drop due to the stronger driving capability of the wider access transistors, and potentially increases the maximum vertical stacks \( L_{max} \).

Another important metric is the bit density, defined as \( L/A_{cell} \). Combining
equation (3.16) with equation (3.19) the bit density can be calculated as following,

\[ D_{\text{bit}} = \begin{cases} 
\frac{H_s L}{AR} + \frac{1}{L} + \frac{2H_s F^2}{AR} & \text{if Case 1} \\
\frac{1}{(\frac{H_s}{AR} + \frac{L}{F}) \times (W_{\text{tran}} + F)} & \text{if Case 2} \\
\frac{1}{\frac{F}{L} \times (W_{\text{tran}} + F)} & \text{if Case 3}
\end{cases} \]  

(3.18)

As can be seen from Equation (3.18), in case 2 and 3 the bit density is improved when adding more stacks in the array (increasing \( L \)). One interesting observation for case 1 is that the bit density is actually a decreasing function of \( L \) given the boundary condition of case 1. That means adding more stacks will reduce the bit density when the cell size is bounded by the etching aspect ratio.

3.3.3.2 modeling 3D-VRAM with vertical access transistor

Planar access transistors (PATs) are implemented in some prior work [7]. The results show that the bit density of many design points in 3D-VRAM are bounded by the size of PATs. To overcome the problem, the PATs are replaced with vertical access transistors (VATs). As a result, the planar footprint of an 3D-VRAM cell can be as small as \( 4F^2 \) when it is not bounded by the etching aspect ratio, which is 33% less than the minimum planar cell size in 3D-VRAM with PATs. However, the maximum number of layers in 3D-VRAM is limited by the drivability of the VATs. Our model considers these effects.

The cell-to-cell pitch is the distance from one cell to another adjacent cell in the same planar electrode,

\[ P = D + 2T_{\text{ox}} + F \]  

(3.19)

The etching aspect ratio (AR) is defined as,

\[ AR = \frac{H_s \times L}{D + 2T_{\text{ox}}} \]  

(3.20)

with the constraint that \( D + 2T_{\text{ox}} \geq F \).

Then the cell area is calculated as,

\[ A_{\text{cell}} = \max(4F^2, (\frac{H_s \times L}{AR} + F)^2) \]  

(3.21)
The bit density can be derived from Equation 3.21,

\[
D_{\text{bit}} = \begin{cases} 
\frac{1}{4} L (b/F^2) & \text{if } AR \geq \frac{H_s}{F} \times \frac{1}{L} \\
\frac{1}{(\frac{H_s}{F} \times AR + 1)} L (b/F^2) & \text{if } AR < \frac{H_s}{F} \times \frac{1}{L} 
\end{cases}
\] (3.22)

As seen from Equation 3.22, when not bounded by the etching aspect ratio, that is, if \( AR \geq L \) (because \( H_s = F \) in our simulation settings), the theoretically maximum bit density \( 0.25L b/F^2 \) is achieved thanks to the introduction of VATs. It is worth mentioning that the maximum \( L \) with tolerable noise margin is limited by the saturation current of the VAT since the VAT should be able to sink the total current on a selected pillar electrode during the write operation.

### 3.4 NVSim: Integrating the Models

The NVM technologies are still not mature, and only a limited number of prototype chips have been demonstrated and just cover a small portion of the entire design space. In order to facilitate the architecture-level NVM research by estimating the NVM performance, energy, and area values under different design specifications before fabricating a real chip, in this work, we build NVSim, a circuit-level model for NVM performance, energy, and area estimations, which supports various NVM technologies including STT-RAM, PCRAM, ReRAM, and legacy NAND flash.

The main goals of developing NVSim tool are:

- Estimate the access time, access energy, and silicon area of NVM chips with a given organization and specific design options before the effort of actual fabrications;

- Explore the NVM chip design space to find the optimized chip organization and design options that achieve best performance, energy, or area;

- Find the optimal NVM chip organization and design options that is optimized for one design metric while keeping other metrics under constraints.

We build NVSim following the same modeling methodology as CACTI [57], but start from a new framework and add specific features for NVM technologies. Compared to CACTI, the framework of NVSim includes the following new features,
• It allows to move the sense amplifiers from the inner memory subarrays to the outer bank level and factor them out to achieve overall area efficiency of the memory module;

• It provides more flexible array organizations and data activation modes by considering any combinations of memory data allocation and address distribution;

• It models various types of data sensing schemes instead of voltage-sensing scheme only;

• It allows memory banks to be formed in a bus-like manner rather than the H-tree manner only;

• It provides multiple design options of buffers instead of latency-optimized option that uses logical effort;

• It models the cross-point memory cells rather than MOS-accessed memory cells only;

• It considers the subarray size limit by analyzing the current sneak path;

• It allows advanced target users to redefine memory cell properties by providing a customization interface.

3.4.1 generic timing and power estimation

In our circuit-level model, we consider the wire resistance and wire capacitance from interconnects, turn-on resistance, switching resistance, gate and drain capacitances from transistors, and equivalent resistance and capacitance from memory storage elements (e.g. MTJ in STT-RAM and GST in PCRAM). The methods of estimating wire and parasitic resistances and capacitances are modified from the previous versions of CACTI [57] by several enhancements. The enhancements include updating the transistor models by latest ITRS report [58], considering the thermal impact on wire resistance calculation, adding drain-to-channel capacitance in the drain capacitance calculation, and so on. We build a look-up table to model the equivalent resistance and capacitance of memory storage elements since they
are the properties of certain non-volatile memory technology. We only model the static behavior of the storage elements and record the equivalent resistances and capacitances of RESET and SET states (i.e. $R_{\text{RESET}}, R_{\text{SET}}, C_{\text{RESET}}, C_{\text{SET}}$).

After calculating the resistances and capacitances of nodes, the delay of each logic component is calculated by using a simplified version of Horowitz’s timing model \[59\] as follows,

$$\text{Delay} = \tau \sqrt{\left(\ln \frac{1}{2}\right)^2 + \alpha \beta}$$

(3.23)

where $\alpha$ is the slope of the input, $\beta = g_m R$ is the normalized input transconductance by the output resistance, and $\tau = RC$ is the RC time constant.

The dynamic energy and leakage power consumptions can be modeled as

$$\text{Energy}_{\text{dynamic}} = CV_D^2$$

(3.24)

$$\text{Power}_{\text{leakage}} = V_DIP_{\text{leak}}$$

(3.25)

where we model both gate leakage and sub-threshold leakage currents in $I_{\text{leak}}$.

The overall memory access latency and energy consumption are estimated by combining all the timing and power values of circuit components together.

### 3.4.2 data sensing models

Unlike other peripheral circuitries, the sense amplifier is an analog design instead of a logic design. Thus, we develop a separate timing model for the data sensing schemes. Different sensing schemes have their impacts on the trade-off among performance, energy, and area. We consider three types of sensing schemes: current sensing, current-in voltage sensing, and voltage-divider sensing.

In the current sensing scheme as shown in Figure 3.3, the state of memory cell (STT-RAM, PCRAM or ReRAM) is read out by measuring the resulting current through the selected memory cell when a read voltage is applied: the current on the bit-line is compared to the reference current generated by reference cells, the current difference is amplified by current-mode sense amplifiers, and they are eventually converted to voltage signals.

\[\text{One of the exceptions is that we do record the detailed I-V curves for cross-point ReRAM cells without diode because we need to leverage the non-linearity of the storage element.}\]
Figure 3.3. Analysis model for current sensing scheme.

Figure 3.4. Analysis model for current-in voltage sensing scheme.

Figure 3.5. Analysis model for voltage-divider sensing scheme.

Figure 3.4 demonstrates an alternative sensing method by applying a current source on the selected memory cell and sensing the voltage via the voltage-mode sense amplifier.

The voltage-divider sensing scheme is presented by introducing a resistor ($R_x$) in series with the memory cell as illustrated in Figure 3.5. The resistance value is selected to achieve the maximum read sensing margin, and it is calculated as follows,

$$R_x = \sqrt{R_{on} \times R_{off}}$$ (3.26)

where $R_{on}$ and $R_{off}$ are the equivalent resistance values of the memory cell in LRS and HRS, respectively.

We model the bit-line RC delay analytically for each sensing scheme. The most significant difference between the current-mode sensing and voltage-mode sensing is that the input resistance of ideal current-mode sensing is zero while that of ideal voltage-mode sensing is infinite. And, the most significant difference between current-in voltage sensing and voltage-divider sensing is that the internal resistance of an ideal current source is infinite while the resistor $R_x$ serving as a voltage divider can be treated as the internal resistance of a voltage source. The
Equation 3.27 and 3.28 show that voltage-divider sensing is faster than current-in voltage sensing with the extra cost of fabricating a large resistor. Comparing Equation 3.29 with Equation 3.27 and 3.28, we can see the current sensing is much faster than current-in voltage sensing and voltage-divider sensing since the former delay is less than the intrinsic line delay $R_T C_T / 2$ while the latter delays are larger than $R_T C_T / 2$. The bit-line delay analytical models are verified by comparing them with the HSPICE simulation results. As shown in Fig. 3.6, the RC delays derived by our analytical RC models are consistent with the HSPICE simulation results.
3.4.3 charge pump

The write operations of some NVM chips require voltage higher than the chip supply voltage. Therefore, a charge pump that uses capacitors as energy storage elements to create a higher voltage is necessary in some NVM designs. We neglect the silicon area occupied by charge pump since the charge pump area can vary a lot depending on its underlying circuit design techniques.

We use the model presented in [61] to calculate the area of the charge pump circuits,

\[ A_{\text{charge pump}} = k \cdot \frac{N^2}{(N + 1) \times V_{\text{DD}} - V_{\text{Out}}} \frac{I_L}{f}, \]  

(3.30)

where \( k \) is a technology-dependent constant, \( N \) is the number of stages in the charge pump, \( V_{\text{Out}} \) is the output voltage, \( I_L \) is the write current and \( f \) is the working frequency.

3.4.4 validation result

We validate NVSim against several PCRAM [2, 62] and ReRAM [4] prototypes in terms of area, latency, and energy. We first extract the information from real chip design specifications to set the input parameters required by NVSim, such as capacity, line size, technology node, and array organization. Then, we compare the performance, energy, and area estimation numbers generated from NVSim to the actual reported numbers in those chip designs. The validation results are listed in this section. Note that all the simulation results are for nominal cases since process variations are not supported in current version of NVSim.

3.4.4.1 PCRAM validation

We first validate the PCM model against a 0.12\( \mu \)m MOS-accessed prototype. The array organization is configured to have 2 banks, each has 8x8 mats. Every mat contains only one subarray. Table 3.4 lists the validation result, which shows a 10% underestimation of area and 6% underestimation of read latency. The projected write latency (SET latency as the worst case) is also consistent to the actual value.

Another PCRAM validation is made against a 90\( nm \) diode-accessed prototype [3].
Table 3.4. NVSim’s PCRAM model validation with respect to a 0.12\(\mu\)m 64Mb MOS-accessed PCRAM prototype chip [2]

<table>
<thead>
<tr>
<th>Metric</th>
<th>Actual</th>
<th>Projected</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Area</td>
<td>64mm(^2)</td>
<td>57.44mm(^2)</td>
<td>−10.25%</td>
</tr>
<tr>
<td>Read latency</td>
<td>70.0ns</td>
<td>65.93ns</td>
<td>−5.81%</td>
</tr>
<tr>
<td>Write latency</td>
<td>&gt;180.0ns</td>
<td>180.17ns</td>
<td>-</td>
</tr>
<tr>
<td>Write energy</td>
<td>N/A</td>
<td>6.31(nJ)</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 3.5. NVSim’s PCRAM model validation with respect to a 90\(nm\) 512Mb diode-selected PCRAM prototype chip [3]

<table>
<thead>
<tr>
<th>Metric</th>
<th>Actual</th>
<th>Projected</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Area</td>
<td>91.50mm(^2)</td>
<td>93.04mm(^2)</td>
<td>+1.68%</td>
</tr>
<tr>
<td>Read latency</td>
<td>78ns</td>
<td>59.76ns</td>
<td>−23.40%</td>
</tr>
<tr>
<td>Write latency</td>
<td>430ns</td>
<td>438.55ns</td>
<td>+1.99%</td>
</tr>
<tr>
<td>Write energy</td>
<td>54(nJ)</td>
<td>47.22(nJ)</td>
<td>−12.56%</td>
</tr>
</tbody>
</table>

Table 3.6. NVSim’s ReRAM model validation with respect to a 0.18\(\mu\)m 4Mb MOSFET-selected ReRAM prototype chip [4]

<table>
<thead>
<tr>
<th>Metric</th>
<th>Actual</th>
<th>Projected</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Area(^3)</td>
<td>187.69mm(^2)</td>
<td>33.42mm(^2)</td>
<td>-</td>
</tr>
<tr>
<td>Read latency</td>
<td>7.2ns</td>
<td>7.72ns</td>
<td>+7.22%</td>
</tr>
<tr>
<td>Write latency</td>
<td>0.3ns − 7.2ns</td>
<td>6.56ns</td>
<td>-</td>
</tr>
<tr>
<td>Write energy</td>
<td>N/A</td>
<td>0.46(nJ)</td>
<td>-</td>
</tr>
</tbody>
</table>

3.4.4.2 ReRAM validation

We validate the ReRAM model against a 180nm 4Mb HfO\(_2\)-based MOS-accessed ReRAM prototype [4]. According to the disclosed data, the subarray size is configured to 128Kb. We further model a bank with 4x8 mats and each mat contains a single subarray. The validation result is listed in Table 3.6. Note that the estimated chip area given by NVSim is much smaller than the actual value since the prototype chip has SLC/MLC dual modes but the current version of NVSim does not model the MLC-related circuitry.

\(^3\)A large portion of the chip area is contributed to the MLC control and test circuits, which are not modeled in NVSim.
Optimize ReRAM Design for Low Cost Memory Component

ReRAM is considered as the most promising universal memory technology since it has faster write latency compared to PCM and has smaller cell structure compared to STT-RAM. Projected as a low cost-per-bit memory technology, high density ReRAM can be improved in many ways: (a) Some ReRAM technologies exhibit non-linear relationship between voltage and resistance. We can leverage this property to build a cross-point structure with a cell size of $4F^2$ and avoid having a dedicated access device in each cell. On the flip side, the lack of access device results in sneak current, and voltage drop problems. Even with the diode implemented in the cross-point structure, the array size can still be limited by the voltage drop. (b) Another way to achieve high density is by building multiple layers stacked on top of each other, as described in Chapter 2 Note that this stacking refers to layers within a single die; 3D stacking of multiple dies is an orthogonal technique. While growing arrays in the vertical dimension increases density, the effective size of an array is still limited by the sneak current and the voltage drop problems. (c) Multi-level cells offer another opportunity to improve ReRAM density. Compared to the multi-layer approach, MLC relaxes the design from dealing with sneak current and voltage drop problems. But it requires more accurate tuning of the analog resistance value of each cell. Thus the resistance distribution and programming strategies in MLC can profoundly affect area, latency, and power.

In addition to the subarray size limitation mentioned in Section 3.1, the row
buffer size is another key factor that affects the area efficiency and performance. For DRAM, due to its destructive read operation, the sense amplifier array is not only responsible for comparing the voltage on the selected bit line to a reference voltage, but also for restoring the value back to the selected cell. The sense amplifier array is capable of temporarily holding the data from the row. Therefore, in DRAM design, the sensing amplifier array also acts as a row buffer. However, the sense amplifier design for the nonvolatile memory technologies is different from that for DRAM.

First, different from SRAM and DRAM, which use charge to represent the information stored in the cell, nonvolatile memories always use resistance to represent their value. Most of the nonvolatile memory prototypes use current-mode sense amplifiers [63, 64, 65, 13, 1]. Second, in nonvolatile memory, the read operation is no longer destructive. Therefore, the sense amplifier array is not necessary to hold the data for restoring. Therefore, the sense amplifier can be implemented separately from the row buffer. However, in our baseline design, the sense amplifiers in the same data line share the same row buffer latches. Third, although the current-mode sensing scheme shows a performance advantage, their area overhead can not be ignored. We surveyed a broad range of state-of-the-art nonvolatile memory prototypes to ascertain that the area overhead of the current-mode sense amplifier is much larger than the area of a voltage-mode sense amplifier, and is normally in the order of $10^5 F^2$. Therefore, the area efficiency of ReRAM is a critical issue when designed for low cost component in the memory hierarchy, i.e. main memory and storage system.

In this chapter, we will first present the circuit-level techniques to improve the area efficiency of ReRAM prototype. Then the design considerations in MLC ReRAM will be discussed.

4.1 Cost Optimization: Circuit Design Tricks

In this section we present several circuit-level techniques to improve the overall area efficiency of ReRAM cells.
4.1.1 area-efficient output buffers

The conventional way of designing an output buffer is to calculate the inverter chain stage and the size of each inverter using logical effort for a given load capacitance. This is always optimized from latency perspective. However, for the cross-point structure of ReRAM array, a word-line row driver with large current driving capability is required so that sufficient amounts of current can pass through the ReRAM cells during write operations. This gives a lower limit of driver size and nullifies the area advantage of ReRAM cross-point memory array.

Therefore, we offer three buffer design choices in our model: one optimizing latency, one optimizing area, while another balancing latency and area. An example is illustrated in Figure 4.1 demonstrating the different sizing methods when an output buffer with 4096 times the capacitance of a minimum-sized inverter is to be designed. In a latency-optimized buffer design, the number of stages and all of the inverter sizing in the inverter chain is calculated by logical effort to achieve minimum delay (30 units) while paying a huge area penalty (1365 units). In an area-optimized buffer design, there are only two stages of inverters, and the size of the last stage is determined by the minimum driving current requirement. This type of buffer has the minimum area (65 units), but is much slower than the latency-optimized buffer. The balanced option determines the size of last stage inverter by its driving current requirement and calculates the size of the other inverters by logical effort. This results in a balanced delay and area metric.
Table 4.1. ReRAM technology assumptions

<table>
<thead>
<tr>
<th></th>
<th>1T1R</th>
<th>Cross-point</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cell size</td>
<td>20$F^2$</td>
<td>4$F^2$</td>
</tr>
<tr>
<td>Maximum NMOS driver size</td>
<td>100$F$</td>
<td></td>
</tr>
<tr>
<td>RESET voltage and pulse duration</td>
<td>2.0V, 100ns</td>
<td></td>
</tr>
<tr>
<td>SET voltage and pulse duration</td>
<td>−2.0V, 100ns</td>
<td></td>
</tr>
<tr>
<td>READ input</td>
<td>0.4V voltage source, or 2μA current source</td>
<td></td>
</tr>
<tr>
<td>LRS resistance</td>
<td>10kΩ</td>
<td></td>
</tr>
<tr>
<td>HRS resistance</td>
<td>500kΩ</td>
<td></td>
</tr>
<tr>
<td>Half-select resistance</td>
<td>-</td>
<td>100kΩ</td>
</tr>
</tbody>
</table>

4.1.2 sense amplifier sharing

Desirable cross-point structures have relatively small cell sizes and also the design constraints of building large cell arrays. However, including sense amplifiers in each array can occupy a dominant portion of the total array area. As a result, in order to achieve a high-density ReRAM array, it is necessary to move the sense amplifiers out of the array and thus to use external sensing.

Figure 4.2 shows a common H-tree organization that connects all the sense amplifier-included arrays together. In contrast, a new external sensing organization is proposed in this work and is illustrated in Figure 4.3. In this external sensing scheme, all the sense amplifiers are located at the bank level and the output signals from each sense amplifier-free array are partial-swing. It is obvious that the external sensing scheme has much higher area efficiency compared to its internal sensing counterpart. However, as a penalty, sophisticated global interconnect technologies, such as repeater inserting, cannot be used in the external sensing scheme since all the global signals are partial-swing before passing through the sense amplifiers.

4.1.3 design evaluation

As Table 4.2 shows, 515 × 512 is the maximum array size for cross-point memory structure calculated by Equation 3.3 and 3.4 under the assumptions in Table 4.1. Therefore, a conventional peripheral circuitry design for latency optimization (using internal current-sensing, multiple-level output buffer, and H-tree organization)
Table 4.2. Comparison between 1T1R and cross-point structures

<table>
<thead>
<tr>
<th></th>
<th>1T1R</th>
<th>Cross-point (baseline)</th>
<th>Cross-point (area-optimized)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process Node</td>
<td>32nm</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Capacity</td>
<td>8MB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cell Size</td>
<td>$20F^2$</td>
<td>$4F^2$</td>
<td>$4F^2$</td>
</tr>
<tr>
<td>Array Size</td>
<td>$2048 \times 8192$</td>
<td>$512 \times 512$</td>
<td>$512 \times 512$</td>
</tr>
<tr>
<td>Total Area</td>
<td>$1.63mm^2$</td>
<td>$1.79mm^2$</td>
<td>$0.72mm^2$</td>
</tr>
<tr>
<td>Area Efficiency</td>
<td>84.3%</td>
<td>15.3%</td>
<td>38.4%</td>
</tr>
</tbody>
</table>

can lead to an unacceptable low area efficiency. Table 4.2 shows such a baseline cross-point design has an area efficiency of only 15.3%. As a result, the actual cross-point ReRAM chip area is larger than that of its 1T1R counterpart even if the 1T1R cell size is five times larger.

Figure 4.4 illustrates the ratio of each component that contributes to the total ReRAM chip area. As shown by the results, using a cross-point structure without building MOS access devices for each cell dramatically reduces the area ratio of ReRAM cells. As the peripheral circuitry starts to dominate the total chip area, using external voltage-sensing and area-aware buffer design become the key techniques for achieving an area-optimized ReRAM design.

To have a more insightful area evaluation, Figure 4.5 and Figure 4.6 show the effects of different design choices of buffers and sense amplifiers. Figure 4.5 shows that switching from latency-optimized buffer to balanced and area-optimized buffer increases area efficiency. This is because the load capacitance of row decoder buffer increases with the number of cells driven by it, which results in more stages in the inverter chain by logical effort but area-optimized design can avoid such area overhead.

Figure 4.6 shows that voltage-sensing always achieves higher area efficiency compared to the current-sensing. This is due to the fact that the chip I/Os are voltage signals and the current-sensing scheme needs to include an I-V converter for the final output stage, which occupies large chip area. However, current-sensing is necessary for latency-optimized design targets as later shown in the next discussion.
Table 4.3. Three strategies in allocating 8 resistance levels (unit: $\Omega$) ranging from $R_{\text{min}} = 1\, k\Omega$ to from $R_{\text{max}} = 10\, M\Omega$

<table>
<thead>
<tr>
<th>State</th>
<th>ISO-$\Delta R$</th>
<th>ISO-$\Delta I$</th>
<th>ISO-$\Delta \log(R)$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1k</td>
<td>1k</td>
<td>1k</td>
</tr>
<tr>
<td>111</td>
<td>1.43M</td>
<td>1.17k</td>
<td>3.73k</td>
</tr>
<tr>
<td>110</td>
<td>2.86M</td>
<td>1.40k</td>
<td>13.9k</td>
</tr>
<tr>
<td>101</td>
<td>4.29M</td>
<td>1.75k</td>
<td>51.8k</td>
</tr>
<tr>
<td>100</td>
<td>5.71M</td>
<td>2.33k</td>
<td>193k</td>
</tr>
<tr>
<td>011</td>
<td>7.14M</td>
<td>3.50k</td>
<td>719k</td>
</tr>
<tr>
<td>010</td>
<td>8.57M</td>
<td>7.00k</td>
<td>2.68M</td>
</tr>
<tr>
<td>001</td>
<td>10M</td>
<td>10M</td>
<td>10M</td>
</tr>
<tr>
<td>000</td>
<td>10M</td>
<td>10M</td>
<td>10M</td>
</tr>
</tbody>
</table>

Table 4.4. Trade-offs in three schemes in choosing resistance levels

<table>
<thead>
<tr>
<th>Metric</th>
<th>Min. $\Delta I$ ($\mu A$)</th>
<th>$t_{\text{sense}}$ (ns)</th>
<th>Avg. $P_{\text{write}}$ ($mW/b$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISO-$\Delta R$</td>
<td>0.016</td>
<td>513</td>
<td>0.05</td>
</tr>
<tr>
<td>ISO-$\Delta I$</td>
<td>143</td>
<td>13</td>
<td>0.31</td>
</tr>
<tr>
<td>ISO-$\Delta \log(R)$</td>
<td>0.27</td>
<td>29</td>
<td>0.13</td>
</tr>
</tbody>
</table>

4.2 Cost Optimization: Leveraging MLC

Multi-level cell (MLC) ReRAM, which can store multiple bits in a single ReRAM cell, can further improve density and reduce cost-per-bit, and therefore has recently been investigated extensively. However, the majority of the prior studies on MLC ReRAM are at the device level. The design implications for MLC ReRAM at the circuit and system levels remain to be explored. This section aim to provide the first comprehensive investigation of the design trade-offs involved in MLC ReRAM. Our study indicates that different resistance allocation schemes, programming strategies, peripheral designs, and material selections profoundly affect the area, latency, power, and reliability of MLC ReRAM.

4.2.1 trade-offs in resistance allocation schemes

In this section we study the impact of resistance allocation schemes on noise margin, sensing latency and programming power. Given the minimum resistance, maximum resistance and number of levels required, there are typically three schemes in determining the resistance values: ISO-$\Delta R$ where the resistance is linearly spaced, ISO-$\Delta I$ where the read current (inverse of resistance) is linearly spaced and ISO-$\Delta \log(R)$ where the resistance is geometrically spaced. Table 4.3 shows the distribution of resistances in a 8-level cell for various schemes. The eight resistance levels range from $R_{\text{min}} = 1 k\Omega$ to from $R_{\text{max}} = 10 M\Omega$. In the ISO-$\Delta R$ scheme, all
the states except "111" have resistance values above 1MΩ, limiting the programming current below 1µA for most of the time. Therefore this scheme is preferred from the energy standpoint. However, the difference between the reading current of state "000" and "001" is only 16nA, making it challenging to develop sensing circuitry to identify the state reliably and quickly. In the $ISO-\Delta I$ scheme, all the states except "000" have resistance values below 10kΩ, maintaining a constant read current difference ($> 100\mu A$) between adjacent states. This scheme enables fast read speed for all cases but requires a large average write power during programming. For a fixed power budget, the maximum write throughput is limited by the number of cells that be programmed at the same. Therefore, the $ISO-\Delta I$ scheme is preferred from the sensing point of view but incurs high energy overhead. One straightforward trade-off between sensing latency and programming power is to choose resistance values with equal interval in log space, denoted as the $ISO-\Delta \log(R)$ scheme. The resistance values are then shifted upwards compared to the $ISO-\Delta I$ scheme, lowering the average programming power; the read current difference between adjacent levels increases compared to the $ISO-\Delta R$ scheme, relaxing the sensing requirement. Moreover, this scheme also benefits from the observation that the resistance distribution in MLC spreads is wider as the resistance value increases [5].

We calculate the write power per bit for the three schemes. We also simulate the sensing latency in an advanced current-mode MLC sensing circuitry [4] using the 22nm PTM model [56] in HSPICE. As part of the sensing circuitry, we assume each bitline is connected with 2048 cells. The results are presented in table 4.4. We can see that the sensing latency $t_{\text{sense}}$ of the $ISO-\Delta R$ scheme is much longer than that of $ISO-\Delta I$ and $ISO-\Delta \log(R)$ due to its much smaller read current difference - 16nA. Among the three schemes, $ISO-\Delta I$ has the highest average programming power (0.31mW/b) because of the aforementioned reason. The $ISO-\Delta \log(R)$ scheme is clearly a reasonable trade-off between sensing latency and average programming power.
4.2.2 trade-offs in programming strategies

In this section we propose two programming strategies targeted either for performance and energy or reliability. The fast programming strategy (FPS) is optimized for write latency and energy with reduced reliability. It is devised for the memory subsystem whose performance is a very critical and can tolerate occasional soft errors. These scenarios include, but not limited to, the server DRAM main memory where ECC is originally implemented to correct particle-induced soft errors. The second option called reliable programming strategy (RPS) is optimized for extremely low error rate but it trades-off write latency. It is more suitable for storage system where data integrity is of most importance. For example, the data in solid-state disk (SSD) or USB driver may need to be stored for years. Moreover, even microsecond-level write latency can be hidden by a large block size in NAND flash as long as the bandwidth requirement is met. Consider a 3-bit MLC cell with bits $D_2 D_1 D_0$: $D_2$ is the most significant bit (MSB) and $D_0$ is the least significant bit (LSB). We use "111" to represent the state with lowest resistance and "000" to represent the state with the highest resistance.

4.2.2.1 impact of retention failure on MLC reliability

Most emerging NVM technologies such as ReRAM, PCRAM, and STTRAM are immune to particle strikes, but they are still susceptible to transient errors due to other reasons. For instance, STT-RAM suffers from thermal fluctuations and PCRAM has both short-term and long-term resistance drift problems. In ReRAM, unlike the gradual drift seen in PCRAM, sudden transitions occur in which the resistance changes abruptly to the highest resistance state (called LRS retention failure [66, 67]) or lowest value (called HRS retention failure [68]). However, once we reprogram the cell, it resumes normal operation.

This retention failure is essentially explained by the same principle as the normal switching behavior of ReRAM. The key difference between the HRS retention failure and H2L programming is that the thermal activation that causes the retention failure is a random process, which occurs very rarely and requires much longer time than a typical write operation. The results in prior work [68] indicates that for ReRAM with a large SET current (> 500µA), strong CFs exist in the cell
and thus only HRS retention failure is observed in these cells because the ruptured CFs are more likely to be reconstructed. In contrast, for ReRAM with a small SET current (< 100μA) [66], only LRS retention failure is observed, because weakly formed CFs are more likely to be ruptured due to the random degeneration of oxygen vacancies in the CFs. The experimental results in [67] confirm that there exists a reverse linear dependence between the resistance value of the LRS and the average retention failure time: \( t_{\text{failure}} \propto 1/R_{\text{LRS}} \). In this work, we consider a key reliability issue in MLC ReRAM design: after H2L programming higher LRS levels are associated with weak CFs and are vulnerable to LRS retention failure; while after L2H programming lower HRS levels are with strong CFs and are vulnerable to HRS retention failure.

### 4.2.2.2 fast programming strategy

The idea behind FPS is to reduce the average write latency by choosing either H2L or L2H programming to reach the target states in as fewer iterations as possible. Figure 4.7 shows the steps involved in FPS. Before the actual programming starts, the current state of a cell \( C_2C_1C_0 \) is first read and compared with the target state \( T_2T_1T_0 \) to be programmed. If they are equal, the programming phase was skipped. Otherwise it is required to identify whether the target state is faster to program from the highest resistance state or the lowest resistance state, which can be simply achieved by checking the MSB of the target state: \( T_2 = 0 \) means that the target state is greater than the median value and we can reach the final state in fewer iterations through H2L programming. In that case, we first apply a full RESET voltage across the cell and program it to the highest resistance state. After that, we simply employ H2L programming to reach \( T_2T_1T_0 \). If \( T_2 = 1 \) then we check if the current state is already in the lowest resistance state. If \( C_2C_1C_0 = 111 \), we can avoid a dedicated SET operation that set the cell to the lowest resistance state. Then, we proceed with L2H programming to complete the write operation. Note that this optimization is not adopted before H2L programming. We do not skip the RESET operation even if \( C_2C_1C_0 = 000 \), since the highest resistance has very wide distribution. It is critical to RESET it to the maximum resistance value.

The drawback of FPS is that some states may be vulnerable to retention failure. For example, if \( T_2T_1T_0 = 110 \), the final state is achieved by slightly rupturing the
strongly formed CFs through L2H programming. This process can lead to the LRS retention failure. In contrast, if $T_2T_1T_0 = 001$, weak CFs are reconstructed from the highest resistance state through H2L programming, and this can lead to the HRS retention failure.

4.2.2.3 reliable programming strategy

The key requirement of RPS is to meet the strict retention requirement (i.e. $>10$ years). This is done by associating LRS with strong CFs and HRS with weak CFs (ruptured). Figure 4.8. In RPS, a write operation is skipped only if the target state has low resistance and is also equal to the original cell state; otherwise a complete RESET operation is performed first. After that, $T_2$ is checked: if $T_2 = 1$ then H2L programming is employed to reach the target state (expect for the case that $T_2T_1T_0 = 111$ is directly SET by applying a full SET current pulse). If $T_2 = 0$, in theory, we can either do H2L programming or we can first perform a SET operation followed by L2H programming. But both of the approaches can impact retention. For example, H2L programming will form weak CFs and can lead to HRS retention failure. On the other hand, doing a SET followed by L2H programming results in ruptured strong CFs, which can lead to LRS retention failure. To overcome this, we first define a transient state which has one-level lower resistance than the target state, and use H2L programming to first reach the transient state. This state will have weak CFs formed since $T_2 = 0$. Then we do L2H programming to reach by $T_2T_1T_0$ rupturing the weak CFs. By controlling the strength of the CFs in the programmed state, stable state is created using RPS.

4.2.2.4 monte carlo simulations

In order to evaluate the trade-offs in the switching latency, energy and retention by the two different programming strategies, we conduct exhaustive Monte Carlo experiments using the SPICE compact model of ReRAM proposed in [69]. The duration of each $I_{SET}$ or $V_{RESET}$ pulse is set to 5ns. We assume $\Delta \log(R)$ for all experiments. For any given pair of states - switching from $C_2C_1C_0$ to $T_2T_1T_0$, we perform 1000 Monte Carlo simulations and record the programming iteration count in each simulation. Based on the results, we calculate the average programming
Table 4.5. Average programming latency (T), energy (E) from $C_2C_1C_0$ (left column) to $T_2T_1T_0$ (top row) and retention time (s) of $T_2T_1T_0$ in fast programming strategy (upper row) and reliable programming strategy (lower row)

<table>
<thead>
<tr>
<th>T (ns)</th>
<th>000</th>
<th>001</th>
<th>010</th>
<th>011</th>
<th>100</th>
<th>101</th>
<th>110</th>
<th>111</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>32.5</td>
<td>29.3</td>
<td>72.1</td>
<td>44.4</td>
<td>95.2</td>
<td>47.4</td>
<td>10.9</td>
<td>11.3</td>
</tr>
<tr>
<td>001</td>
<td>10.7</td>
<td>0</td>
<td>90.1</td>
<td>130</td>
<td>146</td>
<td>97.7</td>
<td>49.8</td>
<td>11.3</td>
</tr>
<tr>
<td>010</td>
<td>12.5</td>
<td>43.3</td>
<td>0</td>
<td>135</td>
<td>147</td>
<td>98.2</td>
<td>50.4</td>
<td>11.5</td>
</tr>
<tr>
<td>011</td>
<td>13.1</td>
<td>44.8</td>
<td>95.4</td>
<td>0</td>
<td>150</td>
<td>101</td>
<td>52.7</td>
<td>12.1</td>
</tr>
<tr>
<td>100</td>
<td>14.8</td>
<td>44.9</td>
<td>96.9</td>
<td>139</td>
<td>0</td>
<td>100</td>
<td>52.2</td>
<td>11.7</td>
</tr>
<tr>
<td>101</td>
<td>14.9</td>
<td>45.3</td>
<td>97.2</td>
<td>140</td>
<td>149</td>
<td>0</td>
<td>51.7</td>
<td>11.1</td>
</tr>
<tr>
<td>110</td>
<td>15.1</td>
<td>45.9</td>
<td>97.8</td>
<td>140</td>
<td>147</td>
<td>97.8</td>
<td>0</td>
<td>10.5</td>
</tr>
<tr>
<td>111</td>
<td>15.2</td>
<td>46.8</td>
<td>98.3</td>
<td>143</td>
<td>146</td>
<td>97.1</td>
<td>50.5</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>E (pJ)</th>
<th>000</th>
<th>001</th>
<th>010</th>
<th>011</th>
<th>100</th>
<th>101</th>
<th>110</th>
<th>111</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>5.08</td>
<td>16.2</td>
<td>31.4</td>
<td>34.9</td>
<td>19.1</td>
<td>7.74</td>
<td>1.48</td>
<td>3.32</td>
</tr>
<tr>
<td>001</td>
<td>1.68</td>
<td>18.9</td>
<td>33.8</td>
<td>35.3</td>
<td>19.4</td>
<td>8.03</td>
<td>1.52</td>
<td>3.35</td>
</tr>
<tr>
<td>010</td>
<td>1.88</td>
<td>6.77</td>
<td>0</td>
<td>34.9</td>
<td>35.5</td>
<td>19.5</td>
<td>8.59</td>
<td>1.54</td>
</tr>
<tr>
<td>011</td>
<td>1.94</td>
<td>6.92</td>
<td>19.7</td>
<td>0</td>
<td>36.6</td>
<td>20.4</td>
<td>8.87</td>
<td>1.58</td>
</tr>
<tr>
<td>100</td>
<td>2.12</td>
<td>6.93</td>
<td>20.0</td>
<td>35.6</td>
<td>0</td>
<td>20.3</td>
<td>8.81</td>
<td>1.55</td>
</tr>
<tr>
<td>101</td>
<td>2.13</td>
<td>6.98</td>
<td>20.0</td>
<td>36.4</td>
<td>35.9</td>
<td>0</td>
<td>8.75</td>
<td>1.49</td>
</tr>
<tr>
<td>110</td>
<td>2.16</td>
<td>7.04</td>
<td>20.1</td>
<td>36.4</td>
<td>35.5</td>
<td>19.4</td>
<td>0</td>
<td>1.44</td>
</tr>
<tr>
<td>111</td>
<td>2.17</td>
<td>7.14</td>
<td>20.2</td>
<td>37.0</td>
<td>35.3</td>
<td>19.3</td>
<td>8.61</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>R (s)</th>
<th>000</th>
<th>001</th>
<th>010</th>
<th>011</th>
<th>100</th>
<th>101</th>
<th>110</th>
<th>111</th>
</tr>
</thead>
<tbody>
<tr>
<td>XXX</td>
<td>10^6</td>
<td>10^7</td>
<td>10^8</td>
<td>10^9</td>
<td>10^9</td>
<td>10^9</td>
<td>10^9</td>
<td>10^9</td>
</tr>
</tbody>
</table>

To estimate the retention time of the resistance state after FPS and RPS, we simplified the flow presented in prior work [66] and did a first-order approximation of the retention under 150°Celsius. The results are summarized in Table 4.5. We can see that most the switching in the FPS completes in a few programming iterations. Actually the switching in the last two digits of a 3b/cell contributes to the major portion of the programming iterations in the FPS. Our simulation results indicate that the programming latency of FPS has an upper bound of 160ns, which is consistent with the reported value in the 2b/cell ReRAM design [4], given that our work and theirs use the same SET/RESET pulse.
duration. In RPS, if \( C_2C_1C_0 \neq T_2T_1T_0 \), writes are faster when \( T_2T_1T_0 = 000 \) or 111. In all the other cases, the latency will be >150ns. The worst-case programming latency in RPS observed in our simulations is about 300ns, almost twice the latency of the upper bound in FPS. Assuming equal switching probability between all the states, the average programming energy of FPS is 14pJ while that of RPS is 46pJ. Thus, FPS clearly has superior latency and energy, but it pays the penalty of reduced retention time in some programmed states. In our study, we found that with FPS, the intermediates states from "001" to "110" show limited retention time under 150°C.

### 4.2.3 trade-offs in circuit design

In order to better understand the trade-offs in the circuit design, we incorporate a circuit-level model into NVSim. Some circuitry modules in MLC NVMs have different requirements from those originally designed for SLC. For example, the existing sense amplifier model in NVSim is based on single-step single-reference sensing circuitry, while MLC sensing needs either multi-step (sequential) single-reference sensing scheme or single-step multi-reference (parallel) sensing circuitry. We model these modules in a compatible approach with the previous model. Moreover, MLC NVMs need specialized control circuits to properly handle their write iterations. We extend the sensing circuit to model these extra overhead, which models the flow chart in Figure 4.7 and 4.8.

The area and latency trade-offs between sequential and parallel sensing circuitry seem to be an interesting option to explore. In the sequential scheme, a single sense amplifier together with multiple latches are used, as shown in Figure 4.9(a). Take a 2b/cell read operation as an example, the enable signals of the three latches is activated at different time steps \( t_0, t_1 \) and \( t_2 \) to sample the output of the sense amplifier at the corresponding time point. After that, the output of the latches is passed to a 3bit-to-2bit encoder to produce the actual stored information of the selected cell. Figure 4.9(b) shows an alternative design - the parallel scheme, in which multiple sense amplifiers associated with different reference voltage levels are implemented. In the first stage, the read current of the selected cell is first converted to a voltage signal by passing through an I-V converter. In the second
Table 4.6. Technology survey of MLC ReRAM metrics in different material systems

<table>
<thead>
<tr>
<th>Material</th>
<th>TiO&lt;sub&gt;x&lt;/sub&gt;</th>
<th>TaO&lt;sub&gt;x&lt;/sub&gt;</th>
<th>HfO&lt;sub&gt;x&lt;/sub&gt;</th>
<th>WO&lt;sub&gt;x&lt;/sub&gt;</th>
<th>CuO&lt;sub&gt;x&lt;/sub&gt;</th>
<th>ZrO&lt;sub&gt;x&lt;/sub&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>MLC levels</td>
<td>5</td>
<td>4</td>
<td>5</td>
<td>8</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>R&lt;sub&gt;off&lt;/sub&gt;/R&lt;sub&gt;on&lt;/sub&gt;</td>
<td>1000</td>
<td>1000</td>
<td>&gt;1000</td>
<td>&gt;20</td>
<td>&gt;100</td>
<td>5000</td>
</tr>
<tr>
<td>Endurance</td>
<td>&gt;10&lt;sup&gt;9&lt;/sup&gt;</td>
<td>&gt;10&lt;sup&gt;8&lt;/sup&gt;</td>
<td>&gt;10&lt;sup&gt;4&lt;/sup&gt;</td>
<td>&gt;10&lt;sup&gt;4&lt;/sup&gt;</td>
<td>-</td>
<td>&gt;10&lt;sup&gt;2&lt;/sup&gt;</td>
</tr>
</tbody>
</table>

stage, the voltage signal is connected to the input of three voltage sense amplifiers and compared with the three different voltage levels simultaneously. The parallel sensing scheme reduces the sensing latency overhead but triples the number of sense amplifiers in for 2-bit cells.

We break down the area and latency of a single-level ReRAM (SLC), a 3b/cell ReRAM with the sequential sensing scheme (MLC-s) and a 3b/cell ReRAM with the parallel sensing scheme (MLC-p) for the same capacity. The results are normalized to the SLC ReRAM in shown Figure 4.10. As we can see, even though the area of memory cells in MLC ReRAM is reduced by 67% compared to SLC ReRAM, the overall area reduction is only about 40%. This is because the peripheral circuits, which do not scale well with the cell size, result in poorer area efficiency [4]. For example, the write drivers and control circuitry are more complex in MLC than in SLC, and their area overhead is higher in MLC. The area of the charge pumps is a function of the output voltage and current, and it stays almost constant from SLC to MLC. In MLC-s, although the number of sense amplifiers decreases as the number of MLC cells is one-third of the SLC cells, the area overhead in additional latches and timing control circuitry prevent the sensing circuitry from scaling linearly with the number of cells. In MLC-p, the additional sense amplifiers even increase the area of the sensing circuitry compared to SLC, causing about 6% die area overhead than MLC-s. Figure 4.10(b) illustrates that the wordline delay and routing delay in MLC is smaller than SLC due to the reduced array size and routing distance, but the read latency of MLC-s can be almost twice that of SLC. However, MLC-p has 30% smaller read latency than MLC-s because the parallel sensing reduces the sensing latency overhead.
4.2.4 trade-offs in material selection

Many ReRAM material systems, such as TiO$_x$ [70], HfO$_x$ [71], WO$_x$ [72], TaO$_x$ [73], CuO [74], and ZrO$_x$ [75] were reported to be capable of MLC operation. Table 4.6 summarizes the state-of-the-art MLC ReRAM metrics in different ReRAM technologies. There are many trade-offs in speed/power/endurance/uniformity of different ReRAM material systems. For example, compared to the TiO$_x$-based ReRAM, the TaO$_x$-based ReRAM has two orders of magnitude higher endurance but needs large RESET voltage (> 5V), and thus requires a more complex and higher overhead charge pump. The WO$_x$-based ReRAM has a tight resistance distribution but small $R_{off}/R_{on}$ ratio, favoring the ISO-$\Delta R$ scheme. the ZrO$_x$-based ReRAM has very large resistance ratio but poor endurance, preventing it from being adopted in frequently accessed random access memory. Based on this technology survey, we build a device library in our model and provide the flexibility of selecting different materials when exploring the MLC ReRAM design space.
Figure 4.2. An example of internal sensing using the H-tree routing organization.

Figure 4.3. An example of external sensing using the non-H-tree routing organization.

Figure 4.4. Area ratio comparison between 1T1R and cross-point designs.
Figure 4.5. ReRAM array with buffers of different optimization targets assuming current-in voltage sensing.

Figure 4.6. ReRAM array area efficiency with different sensing schemes assuming area-optimized buffer design.

Figure 4.7. Flowchart of fast programming strategy
Figure 4.8. Flowchart of reliable programming strategy

Figure 4.9. Schematic view of (a) Sequential and (b) Parallel sensing scheme for 2b/cell read operation

Figure 4.10. ReRAM area and latency breakdowns for SLC, MLC-s (3b/cell) with sequential sensing scheme, MLC-p (3b/cell) with parallel sensing scheme
Architecting ReRAM for main memory replacement

DRAM technology has been used in main memory for more than four decades. However, recent technological trends seriously challenge the continued dominance of DRAM, and open up new possibilities for future main memory systems.

In the past few years, many researchers have explored Phase Change Memory (PCM) technology for main memory – both as a stand-alone memory or in conjunction with DRAM [19, 14, 18]. Although PCM has many favorable characteristics, there are a few critical drawbacks that preclude PCM from becoming a clear winner. Writing a PCM cell is slow (\(\sim 300\,\text{ns}\)) and consumes high energy (\(\sim 30\,\text{pJ}\)) [19], resulting in low write bandwidth. A 20nm 8Gb PCM prototype only provides tens of MB/s write bandwidth [64]. Multi-level-cell (MLC) PCM further increases the write latency and energy by another order of magnitude. In addition, it suffers from long-term and short-term resistance drift [8]. In fact, a recent study has pointed out that it may not be practical to use a 2b/cell MLC PCM as main memory [76].

ReRAM shares several positive characteristics of PCM, and provides much lower write energy and higher density without resistance drift problem. Some recent prototypes, e.g., a 32 Gb part from SanDisk [1], are optimized for density and targeted as Flash replacements. On the other hand, an ISSCC 2014 paper from Micron [77], and HP’s Memristor crossbar project are the best examples of industry efforts to leverage latency-optimized ReRAM for main memory.
Most of the solutions to the limited endurance problem of PCM [20, 21, 21, 14, 18, 23, 24, 27, 21, 31, 28, 29, 30, 32] come with modest storage overhead. In contrast, ReRAM has demonstrated orders of magnitude higher endurance \( (>10^{10}) \) than PCM and such problem is much alleviated. Compared to STT-RAM, ReRAM has much smaller cell size, and thus significantly lower cost per bit.

In this chapter, we analyze design challenges for ReRAM-based memory architecture, and propose circuit-level and architecture-level optimization techniques to enable the adoption of this emerging technology for future memory architecture design. We discuss in detail the crossbar architecture of ReRAM along with its density advantage and shortcomings. We make several key contributions: first, we study the crossbar architecture and describe trade-offs involving voltage drop, write latency, and data pattern for both a conservative baseline design and an aggressive design with double-sided ground biasing. Second, we split the long-latency RESET operations to sub-phases (hRESET) to further reduce write latency. Third, we propose a simple compression based encoding scheme with negligible storage overhead to speed up most of the write operations by limiting the worst-case voltage drop across the selected cells. Finally, we present and evaluate a memory scheduling policy that considers the varying latency of ReRAM writes along with pending activity of a bank when flushing writes to the memory. In addition to improving performance, the proposed design can lower energy, and has no first-order effect on endurance.

While prior work has articulated PCM designs that can be used effectively as main memory, similar strategies do not apply to ReRAM cells. The key difference is that ReRAMs are best implemented with a dense crossbar architecture, while most architectural-level PCM work typically assumes that a PCM cell uses either an MOSFET or a bipolar as an access transistor \(^1\). Therefore, a different set of innovations is required and this chapter describes these crossbar-specific innovations.

\(^1\)While PCM can also use a crossbar, PCM being a unipolar device with different SET and ReSET latencies, its cell diode places very different constraints and has lower sneak current compared to ReRAM.
Old data stored: \( b_0 b_1 b_2 b_3 b_4 b_5 b_6 b_7 = 1 1 0 0 1 1 0 1 \) (n=8)

New data to write: \( 0 0 1 0 1 1 1 0 \) (m_1 = 3, m_0 = 2)

1st step (RESET phase): write “0 0 x x x x 0” (b_0 b_1 b_5)

2nd step (SET phase): write “x x 1 x x 1 x” (b_2 b_6)

Figure 5.1. Two phase multi-bit write operation in a crossbar array: (a) RESET phase and (b) SET phase.

5.1 Data Dependent Write Latency

Based on the characteristics of ReRAM cells and energy/delay/cost constraints of the main memory, ReRAM array is typically designed as a dense crossbar architecture.

In a crossbar design, it is possible to access a single cell in an array by applying a proper potential across the wordline and bitline to which the cell is connected. Hence, there is no over-fetch problem as in DRAM [78]. However, as selected cells are no longer isolated from unselected cells, activating a wordline and a bitline will result in current flow across all the cells in the selected row and column.

Ideally, when we activate a wordline and bitline(s), we want the entire current to flow through the full-selected cell(s) that lies at their intersection(s). For example, in order to SET specific cell(s) in the crossbar array, the selected wordline and bitline(s) are set to \( V \) (or \( V_W \)) and 0 respectively, as shown in Figure 5.1b. Therefore, the write voltage \( V \) is fully applied across the full-selected cell(s). However,
other cells in the selected row and column(s) also see partial voltage across them. These half-selected cells in the selected row and column leak current through them due to partial write voltage across them, which is commonly referred to as sneak current. In order to reduce the sneak current, all of the other wordlines and bitlines that are not selected are half biased at $V/2$. This limits the voltage drop on the half-selected cells to $V/2$ and voltage drop on the unselected cells to 0. Note that a dashed line in Figure 5.1 represents more than one unselected wordlines or bitlines. ReRAM cells exhibit a non-linear relationship between their applied voltage and their current, i.e., current decreases significantly with a small drop in voltage. This helps keep the sneak current through half-selected cells in check. Thus, a critical parameter in a crossbar architecture is the ratio of the amount of current flowing through a fully-selected cell ($I_{\text{fsel,RESET}}$) to a half-selected cell ($I_{\text{hsel}}$), referred to as non-linearity ($\kappa$). The higher the $\kappa$, the lower the sneak current, which will make it feasible to build bigger arrays.

To be cost competitive with DRAM, a memory with fairly large mat size is necessary to reduce peripheral circuit overhead, and for this, $\kappa$ of simple ReRAM alone is not sufficient [79]. Many recent ReRAM prototypes employ a dedicated selector or bi-polar diode in each cell to improve $\kappa$ [13, 1]. Since a selector can be built on top of the switching material, there is no extra area overhead required for the selector. A selector can be built with many different materials with different endurance, operating voltage, and current densities. In this work, we model a selector similar to the one showcased by Burr et al [80].

5.1.1 voltage-time dilemma of ReRAM

Although a crossbar architecture is best suited for building dense memories, most ReRAM memories, even with a dedicated selector in each cell, have only finite non-linearity. Hence, irrespective of how good the cells are, sneak current flowing through the cells poses a number of challenges, opening up new research possibilities for architects.

In a crossbar, the amount of sneak current ultimately determines the energy efficiency, access time, and area of an array. For example, ideally we want to build a big array to improve the density. However, as we size up an array, the number
of half-selected cells increases. Thus we need to provide sufficient voltage at the
driver to account for these sneak currents to avoid write failure [79]. However,
high voltage at the driver should not be significant enough to disturb cells closer
to the driver, which can lead to write disturbance [79]. Furthermore, since sneak
current can vary based on the data stored in the array, it is critical to architect
the sensing circuit and the array dimensions so that we have enough noise margin
to differentiate sneak current from the total read current.

A critical characteristic of an ReRAM cell is that its switching time is inversely
exponentially related to the voltage applied on the cell [36, 81]. The write latency
of the furthest selected cell is calculated based on the relationship between its
voltage drop $V_d$ and switching time $\tau$: $\tau \times e^{kV_d} = C$, where $k$ and $C$
are fitting constants extracted from experimental results [37]. For example, a HfOx-based
ReRAM has demonstrated that a 0.4V reduction in RESET voltage may increase
RESET latency by 10X [36].

Even though the switching time of some ReRAM cells can be small if it is
located near write driver and have almost full write voltage, many ReRAM cells
in the mat will see a different voltage drop across the cell due to the IR loss intro-
duced by Kirchoff’s Law and the sneak currents. The current passing through the
metal wires causes significant voltage loss on the metal wires and thus decreases the
voltage drop on the furthest cell in an ReRAM crossbar. Therefore, the worst-case
switching time of an ReRAM crossbar depends on the array size, write current,
metal resistance and number of bits being written in parallel in a row (wordline).
A straightforward solution is to increase the output voltage of the write driver so
that the worst-case voltage drop can be improved. However, it has several draw-
backs: (1) higher voltage results in higher write power; (2) larger output voltage
requirement increases charge pump stages and complexity, and thus corresponding
area and energy overhead; (3) a larger $V/2$ increases the probability of write dis-
trubance; (4) higher voltage introduces reliability issues, such as time-dependent
gate oxide breakdown of the write driver and worse drain-induced barrier lowering
effect. To deal with such issues, specialized transistor design is required, increasing
both cost and area overhead; (5) the excessive voltage may over-RESET the
nearest selected cells and cause stuck-at-0 faults, resulting in endurance degrada-
tion [37]. In this work, any optimization techniques we introduce aims at increasing
the worst-case (minimum) voltage drop on the selected cell without affecting the maximum voltage drop. Therefore, they should not further bring reliability issues and endurance degradation.

5.1.2 design constraints analysis

In this Section, we study the impact of optimizing cost and power for a 22nm 8Gb ReRAM chip. The goal of this analysis is to identify a few reasonable baseline design points for the rest of the chapter.

We assume an x8 ReRAM chip architecture with a DDR3-compatible interface and 64-bit internal prefetch width. These 64 bits are distributed to 64/n mats in one of the banks. Each activated mat reads (or writes) an n-bit group. Here n is also the minimum number of local sense amplifiers and write drivers in a mat. Increasing n means fetching the 64 bits from fewer mats and each activated mat accessing more bits. We evaluate the area and bank write power of the ReRAM chip with various values of n and mat sizes. We assume that the number of wordlines is equal to the number of bitlines in a mat (denoted as A in Figure 5.2). Later in this chapter, “A512n8” refers to the design point that has a mat size with 512 wordlines and bitlines, and each mat deals with an 8-bit group.

**Chip area** is the key indicator of chip cost and a primary optimization parameter for the memory industry. Figure 5.2a shows that the chip area is reduced as the mat size increases. Due to the large footprint of local sense amplifiers and write drivers because they are shared among two adjacent mats, as illustrated in Figure 6.9a

---

**Figure 5.2.** (a) Chip area and (b) Bank Write power of a 22nm 8Gb ReRAM chip with various mat sizes and number of bits to read/write per mat.
write drivers, the chip area increases significantly as $n$ increases. The increasing trend is more remarkable in smaller mat size. In our work, the area constraint of an ReRAM chip is defined to be the area of a DRAM chip with the same capacity, and a 22nm 8Gb DRAM chip area is estimated to be $45\text{mm}^2$ assuming an array efficiency of 55% [58].

**Write power** contributes to a significant portion of the total power consumption in ReRAM. It is a critical design concern for any memory when power budget is limited, and in this work we apply the same power budget of DRAM to our ReRAM design. The write power of DRAM is calculated from a DRAM power calculator by using the datasheet from Micron [82]. Figure 5.2b illustrates that the write power of ReRAM goes down quickly as $n$ increases. The reason is that the number of activated mats during write access is halved if $n$ doubles, which leads to much less energy consumption due to sneak current. In addition, larger mat size increases the mat power with the increasing of the sneak current. To summarize, a larger $n$ is preferred from the power perspective.

After applying the same chip area and power constraints of DRAM to ReRAM, a few ReRAM organizations meet the design criteria. The mat size of these organizations ranges from $256 \times 256$ to $1024 \times 1024$ while the values of $n$ lie between 4, 8 and 16.

### 5.2 Architectural Enhancement

One of the biggest challenges in crossbar architecture is to overcome the sneak current induced voltage loss. This problem is particularly challenging for ReRAM as the switching time of ReRAM varies exponentially with voltage drop across the cell. For a given mat size, the worst-case voltage drop across the *full-selected cell(s)* depends on two key parameters: 1) the biasing of unselected bitlines and wordlines; 2) the number of low resistance states in the selected row and column, i.e. the data pattern in the row and column. To understand how much biasing can help, we first discuss a microarchitectural enhancement called *double-sided ground biasing (DSGB)*. DSGB incurs an area penalty and might not be suitable in a cost conscious design. We simply use this as a second baseline to show the effectiveness of managing data patterns in a mat.
Figure 5.3. Proposed ReRAM architecture: (a) Schematic view of an ReRAM bank; (b) Mat organization; (c) Design of sense amplifier and write driver. (CSL: column select line; GWL: global wordline; GDL: global dataline; LBL: local bitline; LWL: local wordline; LDL: local dataline; SA: sense amplifier; WD: write driver.)

Figure 5.4. (a) The conventional biasing scheme during RESET phase, (b) the proposed double-sided ground biasing during RESET phase

5.2.1 double-sided ground biasing

The conceptual view of DSGB is demonstrated in Figure 5.4. Using “A512n8” as an example, the worst-case voltage drop occurs at cell $b_7$ which corresponds to the longest IR loss path (marked as red bold line) from the write driver to the ground. In the conventional biasing scheme, as seen in Figure 5.4a, the ground is located at one side of the selected wordline during the RESET phase. The key idea of DSGB is to apply another ground on the other side of the selected wordline. By
doing so, the length of the worst-case IR loss path has been reduced, as illustrated in Figure 5.4b. With DSGB, the worst-case voltage drop occurs at cell $b_3$, with significantly larger voltage than that was observed on cell $b_7$ in Figure 5.4a. The results show that the worst-case voltage drop in “A512n8” has been improved from 2.146V to 2.328V. As a result, the RESET latency has been reduced from 682ns to 240ns.

Although this looks fairly straightforward, the design overhead could be significant because it requires either (a) an additional set of row decoders/drivers on the other side of each array with reduced area efficiency, or (b) shortening the two ends of the array using another metal layer with extra cost overhead. To minimize the area or routing overhead, we borrow the decoder signal from the adjacent array, as shown in Figure 6.9a & 6.9b. In this design, the opposite side of the unselected wordlines are still left floating, as illustrated in Figure 5.4b, and thus the overhead is limited to one pass transistor per row.

The problem of DSGB is the reduced data parallelism: only half of the mats can be activated within an selected subarray due to resource contention (local row decoders). Consequently, only half of the write drivers and sense amplifiers are utilized during each access. To tackle the issue, we present a simple yet effective solution with small modifications of the ReRAM bank architecture. As in the conventional design, the write drivers and sense amplifiers are already shared between adjacent subarrays. It is possible to utilize these resources in alternating fashion. In the proposed design, as illustrated in Figure 6.9a, each time a pair of adjacent subarrays are selected in a bank. Subarray $i$ activates odd-numbered mats while subarray $i+1$ activates even-numbered mats. Therefore, the data parallelism has been maintained.

In summary, DSGB can improve access speed by increasing the voltage across the selected cell, but the downside is that it incurs area overhead. As we mentioned earlier, we describe DSGB not only to show the effectiveness of this technique, but also to understand the extent to which the data pattern impacts the write latency. As we show in the next sub-section, in spite of improving biasing with enhancements such as DSGB, there is still significant voltage loss due to the sneak current.
Figure 5.5. Impact of $m_{1i}$ on the RESET latency of A512n8 for (a) baseline1 without DSGB, (b) baseline2 with DSGB.

5.2.2 impact of data pattern on write latency

It is shown in Section 5.1 that the write latency for writing an n-bit group in a mat is a function of the number of "1" to "0" transitions ($m_{1i}$). The RESET phase can become a performance bottleneck for large $m_{1i}$, whereas the SET phase can always be completed in less than 10ns even if $m_{0i} = n$. To quantify the impact of $m_{1i}$, we evaluate the worst-case RESET latency of writing an 8-bit group in a 512 × 512 mat in a conservative baseline without DSGB (baseline1) and an aggressive baseline with DSGB (baseline2). From Figure 5.5, we can see that the worst-case RESET latency increases substantially as $m_{1i}$ goes up for both cases. In the following subsections, we discuss techniques that either try to avoid worst-case data patterns or mitigate their impact on write latency.

5.2.2.1 split RESET phase

Due to super-linear relationship between $m_{1i}$ and latency in a crossbar, it is more efficient to write one bit at a time sequentially to a mat rather than to perform a multi-bit write operation to reduce $t_{WR}$. However, this increases the bank decoding latency and the global bitline latency to send bits sequentially from memory IO pads to the write drivers of the subarray, which we refer to as $t_{CL}$. Alternatively, we can have logic to perform sequential writes near the subarray but this increases silicon area per subarray and hence cost. We found that when write width per crossbar is eight, writing four bits strikes a balance between the write recovery time $t_{WR}$ and the $t_{CL}$ overhead. For example, with $n = 8$ the idea of splitting
The illustration of split RESET phase under a worst case.

Figure 5.6. The illustration of split RESET phase under a worst case.

RESET phase is to have two half-RESET (hRESET) sub-phases. The first hRESET accounts for the “1” to “0”s in one half of the 8-bit group ($b_0, b_2, b_4, b_6$), while the second hRESET sub-phase accounts for the “1” to “0”s in the remaining half of the 8-bit group ($b_1, b_3, b_5, b_7$), as shown in Figure 5.6.

Figure 5.7 shows how the split RESET phase can reduce the write latency effectively. Again, taking “A512n8” as an example, the tWR decreases from 692ns to 200ns for the baseline1 without DSGB, and from 250ns to 138ns for the baseline2 with DSGB.

5.2.2.2 compression-enabled dual-write-speed mode

In spite of the optimizations discussed so far, the write latency of ReRAM is still many times that of DRAM. We can reduce the number of hRESET phases if the data word has few “0”s. This can be easily achieved with selective data inversion (see example in Figure 10). Similar approaches were also proposed for
PCM devices to reduce write energy [20] and to tolerate stuck-at faults [29]. We could do the same, but assuming a single inversion bit for \( n \) bits of data yields a storage overhead of \( 1/n \) since we need an additional flag bit to indicate whether the (n-1)-bit group is flipped or not, as shown in Figure 5.8. Given that \( n \) is typically \( \leq 16 \), the storage overhead will be at least 6.25\%, which reduces the effective cost-per-bit accordingly. In order to avoid a fixed overhead of 6.25\% for every cacheline in memory, we propose a compression based encoding technique. A compressed line has spare room to accommodate inversion flag bits. Therefore, if the line is compressible, we perform inversion and reduce the number of hRESETs.

Figure 5.9 is an overview of the compression/encoding flow in our framework. It works as follows:

- Every cacheline is assigned a bit to represent if it is compressed or not. This lowers the storage overhead to under 0.2\%. We assume B\( \Delta I \) as our compression algorithm [83] as it is a low-latency algorithm with high compressibility.
If a line cannot be compressed, then it does not use data inversion to reduce the number of hRESET phases.

If a line can be compressed, then at most n-1 bits of the compressed cacheline are placed in every n-bit group. At least one bit is available per n-bit group to indicate if the data has been inverted or not.

The compression bit per cacheline can be stored in virtual memory and cached in the LLC, similar to other approaches [84, 85].

On a write, the memory controller performs the compression to determine how data must be organized and how many hRESETs must be issued. The compression bit must be updated in the LLC or memory (off the critical path).

On a read, the memory controller must read the compression bit from the LLC or memory in addition to reading the cache line from memory. The compression bit is required before the read data can be interpreted and sent to the processor. The access of the compression bit can be performed in parallel and therefore hidden. But such compression metadata accesses can increase memory traffic and hence introduce small slowdowns.

As an example, Figure 5.10 demonstrates how B4Δ2 (4-byte base and 2-byte deltas) compression works. Here we assume a 16-byte long cacheline to save space, and the first 4 bytes in the cacheline is treated as the base. The compressed data are represented with a 4-byte base and three 2-byte deltas, and thus have 10 bytes in total after compression. We choose BΔI compression as our primary

Figure 5.10. A delta compression example: B4Δ2 compression

Table 5.1. Power/Area overhead of hardware implementations

<table>
<thead>
<tr>
<th>Metrics</th>
<th>Compression</th>
<th>Decompression</th>
<th>Encoders</th>
<th>Decoders</th>
</tr>
</thead>
<tbody>
<tr>
<td>Area (μm²)</td>
<td>9480</td>
<td>9460</td>
<td>2518</td>
<td>786</td>
</tr>
<tr>
<td>Power (mW)</td>
<td>4.41</td>
<td>2.43</td>
<td>4.77</td>
<td>3.58</td>
</tr>
</tbody>
</table>
compression units because it is a simple yet efficient hardware compression technique which features low decompression latency and high compression ratio. We adapt the compression in a different and much simpler way from prior work using this technique [83] as our goals are different. Instead of reducing the average compression ratio, ideally we would like to maximize the percentage of compressible cachelines as long as every compressed cacheline has a better compression ratio than $\frac{n-1}{n} \times 100\%$. For any given base size, we only need to choose one large delta size to maximize the chance of compressing a cacheline as long as the following inequality is obeyed,

$$B + \left(\frac{64}{B} - 1\right)\Delta \leq \frac{n-1}{n} \times 64 - 1 \quad (5.1)$$

For example, if $n=8$ and $B=4$, we can simply choose $\Delta = 3$ instead of exploring different delta values $(1,2,3)$ that targets minimum compression ratio [83].

Figure 5.9 is an overview of the decoding/decompression flow in our framework. The header bit of every cacheline from the read queue in the memory controller is checked: “0” indicates it is the original data; “1” indicates it requires decoding and decompression. The decompression unit is chosen based on the “CU” bits, and the corresponding effective data width will be determined.

Hardware implementations of compression/decompression units, encoders, and decoders are verified in behavioral Verilog by creating a testbench and simulat-
Algorithm 1: Write latency aware memory scheduling

\[
\begin{align*}
\text{if} \text{ number of writes in WQ} & > WQHT \text{ then} \\
& \text{//start WQ drain} \\
& nW = 0; \\
& \text{maxWi} = M \text{ for all banks; } \\
\text{while number of writes in WQ} & > WQLT \text{ do} \\
& \text{maxWi} = \text{maxWi} - \text{num of read requests in bank i; } \\
& \text{if maxWi} \leq 0 \text{ then} \\
& \quad \text{continue; } //\text{stop issuing write to bank i} \\
& \text{else} \\
& \quad \text{issue one write ReqW to bank i;} \\
& \quad nW = nW + 1; \\
& \quad \text{if ReqW is fast write then} \\
& \quad \quad \text{maxWi} = \text{maxWi} - 1; \\
& \quad \text{else} \\
& \quad \quad //\text{a slow write takes more than 1 slot} \\
& \quad \quad \text{maxWi} = \text{maxWi} - W; \\
& \text{end} \\
& \text{end} \\
\text{else} \\
& \quad \text{issue reads from RQ;} \\
\text{end}
\end{align*}
\]

ing using Mentor Graphics Modelsim [86]. We further investigate the overheads in terms of power, area, and critical path by synthesizing our Verilog code using Design Compiler [87]. We choose a 45nm technology library and scale the metrics to 22nm technology accordingly. The compression units take less than 100ps to compress the cacheline and this latency is negligible compared to the overall memory access latency. The latency of decompression, encoding, and decoding are also very small. The area and power overheads of the hardware implementations are summarized in Table 5.1. The storage overhead of our design is less than 0.2% (one bit per cacheline).

5.2.3 latency aware write scheduling

Once some cachelines are encoded based on their compressibility, write latency is no longer homogeneous across cachelines. The tWR of ReRAM in “A512n8” can vary from 200ns to 105ns for a conservative baseline and 138ns to 74ns for an
aggressive baseline. While writes are often in the non-critical path, long write latency can still hurt performance by blocking a bank and delaying subsequent reads to that bank. We propose latency aware write scheduling for ReRAM.

In a typical memory system employing bidirectional data bus, writes get queued in a write buffer at the memory controller. Once this queue gets filled beyond a specified higher threshold (WQHT), the controller turns around the bus and flushes the writes until number of pending writes is less than a specified lower threshold (WQLT). In the case of ReRAM, a compressed/encoded cacheline has at most half “0”s in any n-bit group and can be written at an improved latency. We refer to a write request with compressible cacheline as a fast write. In contrast, the write request with incompressible cacheline is referred as a slow write. To avoid slow writes from blocking subsequent reads, when flushing writes, we consider two additional parameters: first, the latency of writes, and second, the number of outstanding requests to the bank to which that write is being scheduled. We schedule writes such that slow writes are written to banks with lowest number of outstanding reads.

For example, any time when write queue drain starts, we assume 32 writes will be issued. Once these writes have been issued, the memory controller switches back to issuing reads. In the baseline design, it is possible that bank \(i\) finishes its writes at cycle 1000 and bank \(j\) finishes its writes at cycle 1320. If bank \(i\) has 10 pending reads and bank \(j\) has 20 pending reads, then this was not the best way to issue the writes as more reads to bank \(j\) will get stalled. Hence, when selecting 32 writes from the write queue, we pick writes such that more work is steered to banks that have shorter read queues than others. A detailed scheduling policy is described in Algorithm 1.
In this section the simulation results will be discussed.

5.3 Results and Discussion

In this section the simulation results will be discussed.

5.3.1 simulation setup

We use GEM5 [88] as our simulation platform and use it with NVMain [89], which is a cycle-accurate memory simulator for both DRAM and non-volatile memories. Table 5.2 shows the detailed baseline configurations of the processor and main memory in our experiments. A DDR3-1066 SDRAM main memory is modeled based on the parameters from Micron’s DDR3 technical specifications [90]. PCM is modeled similarly to Lee et al.’s work [19]. Table 5.3 lists the timing parameters (in nanoseconds) for DRAM, PCM and ReRAM. The selected SPEC2006 CPU benchmark with reference input size [91] and STREAM with all functions [92] are evaluated as multi-programmed testbench. We run all benchmarks for 500 million instructions for the cache warmup and then the following 100 million instructions for the statistics.

We evaluate our three optimization techniques on two baselines. The last digit “x” in the configurations indicates the baseline upon which these optimizations
are applied to: “1” means they are applied to baseline1 without DSGB, and “2” means they are applied to baseline2 with DSGB.

- **baselinex**: The baseline ReRAM architecture with long write latency.
- **spRSTx**: Models the split RESET based on baselinex.
- **DMx**: Models dual-write-speed mode after applying compression and encoding techniques on spRSTx.
- **DM-SSx**: Models smart scheduling on DMx to make the memory controller aware of write latency.
- **idealx**: Models the oracle scenario assuming each write can be serviced at a improved latency that matches the faster write in DMx.

### 5.3.2 memory read latency

The effective memory read latency is one of the key performance metrics for a memory system and it is dominated by two components: (1) the queuing delay of a request in the memory controller; and (2) the time taken to serve the memory access to/from the bank. Our techniques improve both components.

Figure 5.12 shows the average memory read latency of these different configurations, with results normalized to the baselinex. The spRST1 alone reduces the average memory read latency by 27% relative to baseline1 since there is a more than 70% reduction of tWR. Figure 5.13 shows that the spRST2 is able to reduce the average memory read latency by about 7%, as the reduction of tWR is modest. Even based on the aggressive spRSTx design at a much improved write latency,
our DMx and DM-SSx can further improve the read latency significantly, bringing it down close to that of the idealx. This works especially well for some write intensive workloads (e.g. *astar, libquantum, stream*, etc.). For example, the DM-SS1 reduces the average memory read latency by more than 50% for *libquantum* relative to baseline1.

To further examine the effectiveness of DMx, the percentage of incompressible cachelines across different benchmarks are plotted in Figure 5.14. Most applications have less than 5% incompressible cachelines, as our compression units are designed to optimize this metric. Some benchmarks have higher than 10% incom-
pressible cachelines such as *hmmer* and *lbm*, resulting in a significant gap from DMx to idealx, which in turn provides the opportunity for the DM-SSx.

### 5.3.3 system performance

Figure 5.15 & 5.16 show the performance improvement of different memory configurations over the baseline thanks to the effect of reduced memory read latency. As expected, the spRST1 and spRST2 improve the performance substantially by 32% and 8% over baseline1 and baseline2 respectively. The DM1 further improves the IPC significantly over spRST1 for some workloads with high write intensity (i.e. *gobmk*, *astar*, *mcf*, *libquantum*, *stream*, *hmmer*, *lbm*). The performance gap between DM1 and ideal1 for a few benchmarks (i.e. *libquantum*, *hmmer*, *lbm* etc.) indicates that there is room to overcome the issues from the memory scheduling perspective. The DM-SS1 is able to bring the performance within 1% of the ideal1.

### 5.3.4 energy

In the statistics of our exhaustive experiments, we observed that spRST1 and spRST2 can eliminate more than 30% of the hRESET sub-phases, and the average memory energy savings are estimated to be larger than 15% given that the write energy of ReRAM contributes to more than half of the total memory energy consumption. The DM-SS1 and DM-SS2 provide extra energy savings by limiting number of cell flips. Therefore, the techniques we proposed in the design has significantly brought down the energy consumption of ReRAM-based main memory.

### 5.3.5 sensitivity to mat size

Figure 5.17 shows the sensitivity of performance improvement to different mat sizes from $256 \times 256$ to $1024 \times 1024$. The trends for different mat sizes are similar. However, the improvement is higher for larger mat sizes. This is because larger mat size leads to a worse baseline write latency and our optimization techniques can work more effectively on these memory systems. Specifically, the DM-SS1 improves IPC by more than 50% over baseline1 on average.
5.3.6 sensitivity to $n$

Figure 5.17 shows the sensitivity of performance improvement to different values of $n$ (the data fetch width from a mat) from 4, 8 to 16. Again, the trends for different mat sizes are similar. It is observed that our optimizations work extremely well for a large $n$ since a large $n$ yields a higher baseline write latency. The average performance improvement of DM-SS1 over the baseline1 are 21%, 45%, and 73% for $n=4$, 8 and 16.

5.3.7 comparison with DRAM and PCM

We compare our optimized design DM-SS1 and DM-SS2 with DDR3 DRAM design to see how close the design comes in terms of performance. We also model a baseline PCM design (PCM-baseline) and a state-of-the-art PCM design (PCM-PreSET) [16] for comparison. As seen in Figure 5.19, PCM-baseline impacts the performance significantly, with a performance degradation of 32% compared to DRAM, which results from the overly large SET latency of PCM. Although PCM-
PreSET has significantly improved the performance of PCM, it is still much slower than DRAM and has an average performance degradation of 16% compared to DRAM. Both of our optimized ReRAM designs out-perform the PCM-baseline and the PCM-PreSET. The simulation results show that the average performance degradation for DM-SS1 and DM-SS2 are about 6% and 9% compared to DRAM, respectively. Thus, relative to DRAM, ReRAM is able to yield a significant benefit in terms of capacity, cost, and non-volatility, without a significant performance penalty.

5.4 Summary

DRAM is facing many challenges beyond 20nm technology node. As changes to the core memory technology are very rare and considered a big leap in computing systems, it is critical to study all upcoming technologies and scrutinize every characteristic of them before embracing a technology. In this work we explored the emerging ReRAM technology and its unique crossbar architecture. Although crossbar is critical to achieve best possible density, it poses serious challenges in terms of sneak current and voltage drop. To study their impact, we built a detailed modeling tool based on HSPICE and heavily modified CACTI. Based on our analysis using the tool, we showed that the data pattern in a crossbar architecture has a significant impact on the effective voltage across a cell, which in turn affects the overall write latency of ReRAM. To reduce sneak current, we proposed double-sided ground biasing and multi-phase write operations. The above two techniques together reduced the effective write latency from 692ns to 138ns, but the latency is still many times that of DRAM latency. To address this, we pro-
posed and evaluated a compression based encoding scheme to reduce sneak current and improve voltage drop. The benefit of this approach not only comes from the encoding scheme to reduce the number of “0”s and shorten the write latency, but also provides additional storage to track encoded cachelines. Together, this reduces the write latency to 74ns based on the compressibility of a cacheline. Finally, we presented and evaluated a memory scheduling policy that considers the varying latency of ReRAM writes of compressed data along with activity of a bank when flushing writes to the memory. Overall, our architecture improves the performance of a system using ReRAM-based main memory by about 44% over a conservative baseline and 14% over an aggressive baseline, and has less than 10% performance degradation compared to an ideal DRAM only system.
Architecting ReRAM for next-generation storage system

In the past decade, NAND flash based solid state drives (SSDs) have revolutionized the storage system landscape thanks to their smaller footprint, lower power, and orders of magnitude lower access latency than conventional hard disk drives (HDDs). The advent of several emerging non-volatile memory (NVM) technologies provides another opportunity to dramatically change the architecture of computer memory subsystems. Among them, spin-torque-transfer memory (STT-RAM), phase-change memory (PCM), and resistive random-access memory (ReRAM) are considered as promising candidates because all of them offer orders of magnitude lower access latency and higher endurance than flash. STT-RAM has demonstrated excellent endurance and superior switching speed [93], but it has larger cell size [93] and smaller capacity [63] than DRAM. Consequently, it is usually targeted as processor cache replacement. PCM that uses a bipolar or a diode as its access transistor can achieve similar density as DRAM chips [64, 19]. However, there is no evidence that the density of PCM could get close to that of NAND flash. ReRAM has shown better cell-level characteristics than PCM and flash [5], and a 32Gb prototype of 2-layer cross-point ReRAM has demonstrated its potential to build large-capacity memory chips [1]. The recent development in 3D vertical ReRAM (3D-VRAM) enables an ultra-high-density architecture as flash replacement [6, 94, 50]. There are two major reasons such 3D-VRAM can be a cost-competitive solution. First, its monolithic 3D multi-layer structure im-
proves effective bit density dramatically, just as the conventional 3D horizontal ReRAM (3D-HRAM) does [65, 49, 13, 1]. Second, the cost overheads associated with additional layers are eliminated by the removal of some intermediate fabrication process [50], saving significant fabrication cost compared to a 3D-HRAM counterpart [6, 7].

The scope of this chapter can be classified into three categories from the array-, circuit-/architecture- to system- level design and optimizations.

**Design space exploration:** Most prior work on 3D-VRAM has focused on device-level optimizations for 3D vertical ReRAM cells [94, 50, 51, 95]. These devices have shown a wide range of cell-level characteristics such as resistance, nonlinearity, and switching current. A few studies [96, 7] have analyzed the 3D-VRAM array design using their circuit models, focusing on the scaling trend, the impact of geometry parameters, and comparisons between 3D-HRAM and 3D-VRAM. Despite the analysis they conducted, there is little in literature about the trade-offs of array design by exploring the cell-level characteristics and read/write schemes for 3D-VRAM. Without a detailed design space exploration, it is difficult to get insights into some design choices such as: (1) Does the low resistance or nonlinearity of a cell play a more important role in the read/write margin? (2) Is single-bit or multi-bit access preferred in 3D-VRAM? (3) How to choose a proper read voltage to balance the sensing margin and disturbance probability? (4) What is the impact of the selection of access transistor (i.e. vertical or planar MOSFET) on the bit density? One may argue that the answers to these questions could be tracked down from the design implications of planar cross-point ReRAM since some of the issues that arise in 3D-VRAM design appear to mimic the problems tackled previously [97, 79]. However, we find that the conclusions could change slightly (i.e. Question 1) or significantly (i.e. Question 2) in the 3D scenario from the case of planar structure. The rational behind these difference could be the existence of many more sneak paths in a 3D-VRAM array or the limited driveability of the vertical access transistor. Moreover, some issues (i.e. Question 3) are rarely mentioned in prior work and others (i.e. Question 4) are unique in 3D-VRAM.

**Circuit/Architecture optimization:** The write and read circuitry needs careful design in ReRAM because its write drivers (WDs) and sense amplifiers (SAs) occupy a significant portion of footprint in an ReRAM chip. The area of
these circuits do not scale as well as the cells, especially when the area of cells are reduced by multi-layer structure rather than technology scaling. Multi-directional driver design is proposed to mitigate these overheads by leveraging the flexibility in connecting plane electrodes and the relaxed constraints in the layout of WDs. The results show that by doing so, we can almost halve the total area of WDs and, at the same time, quadruple the array size while maintaining the design margins. In addition, remote sensing scheme is motivated to tackle the limited on-die sensing resource problem [1]. This technique is also introduced into our design. A macro-architecture model is built to quantify the benefits of these circuit/architecture optimizations.

**System-level evaluation:** After applying the array and circuit/architecture optimizations, our optimized 3D-VRAM design is compared against conventional 2D NAND flash and emerging 3D NAND flash [98, 99]. The trace-based simulations are performed by customizing an disk simulator to characterize ReRAM timing and energy models. Leveraging the cost model, more generalized comparisons can be done beyond the performance-only metrics such as input/output operations per second (IOPS) that is the focus of most system-level research on storage memory [100, 101]. In particular, the mixed performance/cost/energy metrics are interesting to the memory industry about the adoption of new memory technology. The results show that our optimized 3D-VRAM design has better IOPS/\$ than other contenders for storage memory in most cases and has the best IOPS/\$/J in all tested cases.

### 6.1 3D-VRAM Array Structure

To improve the integration density in ReRAM, one simple solution is to stack the planar cross-point structures layer by layer (a.k.a 3D-HRAM), which improves the bit density [7] to $0.25L b/F^2$ where $L$ is the number of layers in 3D-HRAM. However, the fabrication cost of critical lithography, etching, chemical mechanical planarization, and other process associated every stacked layer increases linearly with $L$. These cost adders may eventually offset the benefits of larger density enabled by the multi-layer structure [6, 7].

From the industrial perspective, the cost per bit is a major driving force to pur-
Figure 6.1. Schematic view of a 3D-VRAM array with vertical access transistor
sue the 3D integration. This motivates the 3D-VRAM architecture, which tilts the
erizontal ReRAM by 90 degrees, as a much more cost-efficient solution by elimi-
nating the aforementioned fabrication cost overheads in 3D-HRAM. The schematic
view of a 3D-VRAM array is illustrated in Figure 6.1. The plane electrodes and
isolation layers are deposited consecutively. Only after the top most layer is de-
posited, the critical lithography and etching steps are involved for patterning the
pillar electrodes and opening the contacts for WLs. The vertical ReRAM cells
are sandwiched between the perpendicular pillar electrodes and multi-layer plane
electrodes. At the bottom of the pillar electrodes, there is a 2D arrays of access
transistors. Their sources are connected BLs and their gates are controlled by
the select-lines (SLs). With the appropriate bias schemes on WL (decoding in
z-direction), BL (decoding in x-direction) and SL (decoding in y-direction), each
memory cell in the 3D cross-point architecture can be individually accessed. Dur-
ing a read or write operation, one selected SL is biased to turn on the access
transistors connected to this SL while all the other access transistors remain off by
grounding the unselected SLs. This operation basically activates an x-z vertical
plane, which is a de facto cross-point structure. Within the vertical plane, the
same read/write biasing scheme as planar cross-point structure can be applied.

Planar access transistors (PATs) are implemented in some prior work [7]. Their
results show that the bit density of many design points in 3D-VRAM are bounded
by the size of PATs. To overcome the problem, the PATs are replaced with vertical
access transistors (VATs) in our design. As a result, the planar footprint of an 3D-
VRAM cell can be as small as $4F^2$ when it is not bounded by the etching aspect
Figure 6.2. Voltage regions of sate write (perfect switching probability), unsafe write (not perfect switching probability), and safe read (no disturbance).

ratio, which is 33% less than the minimum planar cell size in 3D-VRAM with PATs. However, the maximum number of layers in 3D-VRAM is limited by the drivability of the VATs. Our model considers these effects.

6.2 3D-VRAM Array Design

In this section, several critical issues in 3D-VRAM array design will be addressed.

6.2.1 write and read margin

To set up appropriate values for $V_W$ and the criterion of write margin, the switching voltage distribution within an array of cells should be considered as ReRAM are well-known for its switching parameter variability [5]. For ReRAM cells with an average switching voltage of 2$V$, a possible switching range of $1.7V \sim 2.3V$ is assumed, as shown in Figure 6.2. In this work, the write access threshold is set to 2.5$V$ to ensure a safe write operation, and $V_W$ is set to 3$V$ to obtain a 0.5$V$ toleration of voltage loss on the interconnect. Meanwhile $V_W/2 = 1.5V$ is lower than 1.7$V$ to avoid the disturbance of the half-selected cells. For the read margin, a minimum $\Delta I = 50nA$ is used as the criterion for a reliable sensing with reasonable latency [102]. The maximum $V_R$ is the same as $V_W/2$ to avoid the disturbance of the cells on selected WLs.

6.2.2 design space exploration

The design trade-offs involving cell-level characteristics, read/write schemes will be discussed.
Figure 6.3 shows the worst-case voltage drop on the furthest selected cell and read sensing margin in a 16-layer 3D-VRAM array with various nonlinearities versus the planar size of an array. Even with a large nonlinearity of 200, the worst-case voltage drop cannot meet the criterion. This is because the write current for \( R_{on} = 25k\Omega \) already exceeds the saturation current (100\( \mu \)A) of VAT, which causes significant voltage drop across the VAT, as illustrated in the breakdown of voltage drop in Figure 6.3c. On the other hand, the increase of nonlinearity dramatically decreases the read sense margin. Given \( R_{on} \), the sensing margin \( \Delta I \) decreases dramatically as the nonlinearity increases. These results suggest that current drivability of the vertical transistor put a hard constraint on the minimum \( R_{on} \). In other words, with a small \( R_{on} \), increasing nonlinearity alone is not able to meet the write margin criterion while it is detrimental to the read sensing margin.
Figure 6.4. A 16-layer 3D-VRAM with nonlinearity $K_r = 5$: (a) Worst-case voltage drop on furthest cell with various nonlinearity $R_{on}$; (b) read sensing margin with various read voltage $V_R$

6.2.2.2 resistance

Figure 6.4a illustrates that even with a small $K_r$ of 5 and $V_R$ of 0.5V, increasing $R_{on}$ (up to 500kΩ) is an effective way to improve the worst-case voltage drop while maintaining the read margin above the criterion. Along with the observations in Figure 6.3, it is concluded that $R_{on}$ plays a more important role in 3D-VRAM array than $K_r$ does due to the limited drivability of the VAT. Such conclusion does not apply on the planar cross-point array design [97, 79]. As a result, 3D-VRAM relaxes the design efforts from the perspective of device engineering because it is easier to increase $R_{on}$ by lowering the SET compliance current while it is more difficult to improve $K_r$ which typically requires metrial/structure innovations or additional selector devices.

6.2.2.3 read voltage

When tuning cell-level characteristics, there could exist a fundamental conflicting nature between the write and read margin. One design knob that can be tuned is the read voltage $V_R$, as long as it is less or equal than $V_W/2$ to avoid the disturbance of the cells in the selected plane electrode. Figure 6.4b demonstrates the sensing margin with different $V_R$ as a function of array planar sizes in a 3D-VRAM array. By increasing $V_R$ from 0.5V to 1.5V, the sensing margin are improved approximately by a factor of 4. With a larger $V_R$, a higher $R_{on}$ or a larger nonlinearity can be tolerated. However, the read access energy also increase by about 10X from
6.2.2.4 read and write energy

We develop an energy model to evaluate the read/write energy of a 3D-VRAM array. It is found that the static energy consumption due to the sneak path current, rather than the dynamic energy consumption due to the charging/discharging parasitic capacitances, dominates the access energy because a lot of sneak paths exist in an activated 3D-VRAM array. Therefore increasing $R_{on}$ or nonlinearity
Table 6.1. Worst-case voltage drop and write current versus number of bits to write in parallel.

<table>
<thead>
<tr>
<th># of bits to write per array</th>
<th>Current on selected wordline (mA)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.21</td>
</tr>
<tr>
<td>2</td>
<td>2.21</td>
</tr>
<tr>
<td>4</td>
<td>4.21</td>
</tr>
<tr>
<td>8</td>
<td>8.21</td>
</tr>
<tr>
<td>16</td>
<td>16.21</td>
</tr>
<tr>
<td>32</td>
<td>32.21</td>
</tr>
<tr>
<td>64</td>
<td>64.21</td>
</tr>
<tr>
<td>128</td>
<td>128.21</td>
</tr>
</tbody>
</table>

Figure 6.7. Multi-bit write operation in a 128x128x16 3D-VRAM array: (a) worst-case voltage drop and write current versus number of bits to write in parallel; (b) array write energy and write energy per bit versus number of bits to write in parallel.

should also reduce the access energy significantly. Figure 6.6 shows the read and write access energy when reading and writing a single cell in a 16-layer 3D-VRAM array. The duration of a write pulse is assumed to be 100ns [79], and the read sensing latency for $\Delta I \geq 100nA$ can be as small as 26ns [102].

### 6.2.2.5 multi-bit access

Theoretically, the entire row of the selected WL can be read or written in parallel. In practice, only a small number of bits are accessed at the same in a planar cross-point structure. The primary reason is that the total current on the selected WL increases dramatically as the number of full-selected cells increases. It degrades write margin and incurs high area overheads of WDs [79]. A planar cross-point array sized by $512 \times 512$ (=256k cells) with the same cell characteristics as our 3D-VRAM cells is simulated. The current requirement of an individual WD doubles when increasing the number of accessed bits in parallel from 1 bit to 128 bits.

*Does this conclusion still hold for 3D-VRAM?* Figure 6.7a plots the write margin and write current on the selected WL as a function of the number of bits $N_b$ that are written in parallel in a 3D-VRAM array with the same number of cells (256k) as the 2D case. The write margin degrades slightly as the $N_b$ increases. It can also be observed that the increasing of write current from 1-bit write to 128-bit write is only 7%, suggesting that multi-bit write operation is feasible in 3D-VRAM. The rational behind the difference between 2D and 3D ReRAM is that the number of half-selected cells on the selected WL in the 3D-VRAM ($=16284 - N_b$) is much
more than they are \((=128 - N_b)\) in the 2D cross-point array, and thus the current of these half-selected cells dominates the total current on a selected WL.

The array write energy and write energy per bit as a function of \(N_b\) is shown in Figure 6.7b. We can see that the write energy of the 3D-VRAM array increases by only 28% from 1-bit write to 128-bit write, and the write energy per bit is substantially reduced as \(N_b\) increases. We also examine the read case and find similar trends and conclusions (not shown due to the space limit). It is concluded that multi-bit access is much more favorable in 3D-VRAM than single-bit access with high energy efficiency and low area overheads.

6.3 Circuit and Architecture Design

In this section circuit techniques to relax peripheral overheads are introduced. Then we will explain macro-architecture design and use our developed macro model to evaluate some of the optimizations.

6.3.1 optimize write and read circuitry

Targeted as NAND flash replacement, the ReRAM design should be highly optimized for cost-per-bit, which is primarily determined by the die area of an ReRAM chip given the die capacity. Several factors have major impacts on the die area: (a) bit density determines the total area of cells, (b) array size determine the number of sets of peripheral circuits (i.e. decoders, multiplexers, write drivers etc.), (c) the style of peripheral circuity affects its area. The design space exploration in Section 6.2 tries to find optimal design points with high bit density and large array size. The techniques to be discussed in this section relaxes the peripheral overheads.

6.3.1.1 multi-directional write driver

In traditional memory structures including SRAM, DRAM, flash and 1T1R NVM, the WDs in the last-stage row decoders are sized up to balance the delay of charging/discharing the corresponding WL. Alignment of these WDs is challenging because the WDs have to layout in the space of WL-defined pitch. One solution is to
Figure 6.8. Circuit optimizations for 3D-VRAM: (a) conventional write driver (WD) design: WDs are connected to one edge of the array; (b) proposed design: WDs are connected to the North, South, East, and West edges of the array; (c) remote sensing scheme: shared SAs within a block (adapted from [1] with modifications.

Figure 6.9. Hierarchial architecture of 3D-VRAM macro: (a) An ReRAM die consists of 4 planes, and each plane has 32 sub-planes; (b) Each sub-plane is made up of 8 blocks, the SAs and WDs within a block are shared among the 3D-VRAM arrays; (c) a detailed view of a 3D-VRAM array.

layout the WDs with even-numbered WLs on one side of the WLs and the WDs with odd-numbered WLs on the other side of the WLs. For cross-point NVMs, the WDs are responsible for providing sufficient current of the selected WL to both the full-selected cells and the half-selected cells. This not only worsens the alignment problem but also increases the area of these WDs, reducing the array efficiency significantly. Our 3D-VRAM design tries to solve this problem. First, the alignment problem is much alleviated in 3D-VRAM because for an $N \times N \times L$ array there are only $L$ WDs to be aligned aside a planar size of $N \times N$, increasing the effective SL-defined pitch by $N/L$. For example, as shown in Figure 6.8a, the $L$ WDs are connected to one edge of the array through top metal layer in a conventional design.
Utilizing the flexibility in the placement of contacts for plane electrodes, we propose multi-directional WDs for 3D-VRAM. The design is demonstrated in Figure 6.8b. The WD for topmost plane electrode, marked as WD\(_{1N}\) in the conventional design, are distributed to the north (WD\(_{1N}\)), east (WD\(_{1E}\)), south(WD\(_{1S}\)), and west(WD\(_{1W}\)) of the array in the new design. By doing so, each single WD only need to provide one quarter of the current on that plane electrode, reducing the size of an individual WD and further relaxing the alignment constraints. Moreover, most WDs can be shared by two adjacent arrays along either the SL-direction or BL-direction, as illustrated in Figure 6.9b. For example, the WDs placed on the south of the array can be shared with the adjacent array to the south of it if either array is activated. Therefore, the array efficiency is improved as the total number of sets of WDs in a block is reduced.

Another side benefit of the proposed WD design is that the current path for the worst-case voltage drop on the plane electrode is almost halved, improving write margin significantly. As a result, larger array size may be allowed. Figure 6.10 shows the comparisons of write margin and write current between the multi-directional driver design and the conventional design. It is observed that the worst-case voltage drop of a \(2N \times 2N \times 16\) array in the new design is slightly better than that of a \(N \times N \times 16\) array in the baseline design. Moreover, the current requirement of each individual WD for \(2N \times 2N \times 16\) arrays in the new design is almost the same as it is for \(N \times N \times 16\) arrays in the baseline design. The sensing margin is well-maintained in the new design. Therefore it can be concluded that with the proposed write driver design we can quadruple the array size and reduce the total area of WDs at the same time.
6.3.1.2 remote sensing scheme

We find that the area overhead of current-mode SAs is significant after we surveyed a broad range of state-of-the-art nonvolatile memory prototypes [63, 64, 65, 13, 1]. Our calculation shows that the layout area of a current-mode SA is in the range of $10^4 \sim 10^5 F^2$. Given the footprint of one array in our 3D-VRAM design is in the order of $10^5 F^2$, the sensing resources in an ReRAM die is very limited as the array efficiency is an important design criterion. We use the concept of remote sensing scheme introduced in a recent 3D-HRAM prototype [1]. As shown in Figure 6.8c, for a block with $2M$ arrays with each array having its local SA, the global select buses GSELB_S are used to control the connections between local SAs and the central buses GSELB_T. The GSELB_T of the accessed array are multiplexed into one group of buses which connect to the selected BLs in the activated array. Only one array in a block can be activated at a time. The read operation within a block is pipelined to read out the required amount of data. The parasitic delay in the cross-block buses for read operation are calculated in our macro-level model.

6.3.2 macro-architecture design

The architecture of our 3D-VRAM macro is illustrated in Figure 6.9. Each ReRAM die is designed as a multi-plane architecture and multiple memory requests are served in parallel. Within each plane, there are 32 sub-planes and two sub-planes in the same row are activated at the same time. Each sub-plane is further divided into 8 blocks and 1 of them are activated during access. Assuming there are $2M$ arrays in a block, the $M$ arrays (marked in dark grey in Figure 6.9b) are activated for writing the first half of data in the block, then the remaining $M$ arrays (marked in white) are activated for writing the second half of data in the block. The switching between them is fast because we can simply disable the output of one direction and enable the output of the opposite direction in all the activated WDs in a block.

6.3.3 macro-Level model

We implement the architecture of our 3D-VRAM design in NVSim [103], which is an open source modeling framework for emerging NVMs. To evaluate the area and
energy savings of our proposed design, the modules of write drivers and sensing circuitry/structure are heavily modified in NVSim.

6.3.3.1 timing model

The physical access time for reading a page in 3D-VRAM can be expressed as,

$$t_{\text{page,read}} = S_r \times t_{\text{sense}} + t_{\text{peri}} + t_{\text{trans}}$$  \hspace{1cm} (6.1)$$

where $S_r$ the number of serial sensing steps within a block, $t_{\text{sense}}$ is the sensing delay including both the latency of the sense amplifiers and the RC delay of cross-block buses, $t_{\text{peri}}$ is the delay of other peripheral circuits such as decoders and multiplexers, and $t_{\text{trans}}$ is the data transfer latency from page buffers to I/O. Normally the internal data movement is transferred byte by byte, then the data transfer latency can be calculated by $t_{\text{trans}} = N_p / f_{\text{trans}}$ where $N_p$ the page size and $f$ is the data transfer frequency.

6.3.3.2 cost model

We reconstruct the cost models of ReRAM from previous work [7]. One modification made is to break down the details of fabrication process of VAT include its corresponding cost overhead in the IC Knowledge LLC [104].

6.3.4 results and discussions

The die area of 64Gb chips with different memory organizations are compared in Figure 6.11. MLC ReRAM in 1T1R structure is feasible, but it is not considered
Figure 6.12. (a) Average page read energy and (b) Average page program energy at $F = 30nm$ for a page size of 8KB in 2D SLC NAND, 2D 3b/c MLC NAND, 16-layer SLC NAND, 16-layer 2b/c NAND, 2D SLC NAND, 2D 3b/c MLC NAND, 16-layer SLC NAND, 16-layer 2b/c NAND, planar ReRAM with single-bit access(SBC), 16-layer 3D-VRAM with SBC, 16-layer 3D-VRAM with multi-bit access(MBC).

in this work because the actuate control of resistance values after programming is difficult to achieve in any form or cross-point structure. We can see that the invention of 3D vertical structure in both NAND flash and ReRAM can reduce the die area substantially. ReRAM has larger die size than its NAND flash counterpart with the same bit density because the WDs and SAs are much larger than they are in NAND flash. The multi-directional WD design reduces the overall die area of 3D-VRAM from $6.1 \times 10^{10}F^2$ to $4.2 \times 10^{10}F^2$.

The cost per GB of these memories are also plotted. To make a fair comparison, the calculations are based on the same feature size $F = 30nm$ for different memory structures and organizations. And all the simulations later assume $F = 30nm$ unless specified. As seen in Figure 6.11, the cost comparison almost follows the trend in the die area comparison, affirming that the process of 3D vertical structures do not introduce significant cost adders.

Figure 7.5 compares the page access energy among different memory organizations. The 2D ReRAM with multi-bit access are not shown because its area overhead is too large due to the aforementioned reason in Section 6.2.2.5. We can see that if single-bit access is implemented in 3D-VRAM, the read energy would be much larger than other memories because it is aggregated from a large number
Table 6.1. Timing parameters for NAND flash and ReRAM

<table>
<thead>
<tr>
<th>Item</th>
<th>SLC NAND</th>
<th>2b/c NAND</th>
<th>3b/c NAND</th>
<th>ReRAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read latency ($\mu$s)</td>
<td>35</td>
<td>50</td>
<td>90</td>
<td>6.4</td>
</tr>
<tr>
<td>Program latency ($\mu$s)</td>
<td>350</td>
<td>350−3000</td>
<td>350−5000</td>
<td>0.5</td>
</tr>
<tr>
<td>Erase latency (ms)</td>
<td>1.5</td>
<td>5.5</td>
<td>10</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Figure 6.13. IOPS comparisons between NAND flash and ReRAM across different applications of activated arrays which have a lot of sneak paths. With multi-bit access, the read energy could be reduced substantially. Similar conclusion applies to the write scenario. The multi-bit write operation reduces the write energy of the 3D-VRAM array from about 3X to only 12.5% that of the 3D MLC NAND.

6.4 System-Level Evaluation

After applying the optimizations on our 3D-VRAM design, trace-based simulations are performed by customizing an disk simulator with SSD extension [100] to characterize ReRAM models. Different workloads with various I/O request frequency and patterns are simulated, including the synthetic workload in the disk simulator [100], Iozone and Postmark, as well as Financial and Websearch [105].

In the macro model of storage memories, it is observed that the $t_{peri}$ term in Equation 6.1 contributes to less than 2% of the total physical page access time in most configurations. Therefore, the different peripheral delays between 2D and 3D memories in the system-level performance evaluation can be ignored. We use generalized timing parameters for SLC NAND, MLC NAND and ReRAM. Table 6.1 summarizes the page read/program latency of them and the block erase latency of NAND flash. These specifications are based on 64Gb NVM dies with 8KB page size, and the I/O date transfer rate is 166MBps.

The performance comparison between ReRAM and different NAND flash are il-
illustrated in Figure 6.13. It is observed that ReRAM (2D or 3D) as storage memory can improve the system throughput greatly. The increasing of IOPS are remarkable for the workloads with high (e.g. iozone) and modest (e.g. postmark) write intensity. Performance improvement over read-intensive workloads (e.g. financial and websearch) are also significant for ReRAM.

Performance-only metric is not sufficient for evaluating the potential of a new memory technologies to be adopted in industry. A major reason that SSDs took over the storage market from HDD is that it has lower price/performance ratio than HDD. Therefore we introduce the metric of IOPS/$ to compare the emerging 3D-VRAM with the existing technology (2D NAND flash) and other contenders (e.g. 3D NAND flash). Figure 6.14a shows the comparison results (the IOPS/$ of every configuration for a given workload is normalized to that of the 2D SLC NAND for the given workload). Our optimal 3D-VRAM design wins over most of other memories, including its 3D SLC NAND counterpart, in all the tested workloads. For iozone and postmark, the advantages of 3D-VRAM over others are
more than 45%. However, for read-intensive workloads, the IOPS/$ of 3D-VRAM can be 35% less than that of its 3D MLC NAND flash counterpart.

Another metric - IPOS/$/J - is also proposed, which combines the performance, cost, and energy aspects of a memory technology. As shown in Figure 6.14b, our optimal 3D-VRAM design is a clear winner over all the other memories for all tested workloads.

6.5 Summary

ReRAM is one of the most promising candidates for next-generation storage systems. Compared to NAND Flash, ReRAM has superior read/write access latency and many other advantages. 3D-VRAM has been demonstrated as a naturally low-cost architecture solution. As changes to the existing memory technology are challenging, it is critical to study every characteristics of the new technology that could affect the design choices. We explored the large design space of 3D-VRAM arrays and came to a couple of important conclusions that were different from, or not studied in, the conventional 2D cross-point design. We also proposed circuit/architecture optimizations to relax the peripheral overheads of 3D-VRAM and further reduces its cost-per-bit. The system-level evaluations showed that our optimized 3D-VRAM design has better IOPS/$ than other contenders for storage memory in most cases and has the best IOPS/$/J in all tested cases.
Reliability-Aware ReRAM Design

Soft and hard errors are vital concerns when designing a memory system. A soft error is a random, recoverable upsetting of the information stored in a memory cell, while a hard error is a permanent corruption of a memory cell resulting from physical defects. Although most emerging non-volatile memory technologies are not charge-based storage, they still suffer from soft and hard errors. The presence of hard errors normally results from the limited endurance compared to DRAM and SRAM technologies. The cause of soft error is distinctive for each NVM. For example, soft errors of PCM refer to the resistance drift behaviors, or the thermal disturbance from adjacent cells. As for STT-RAM, the stochastic properties imply that both write and read operation can bring in soft errors. For ReRAM, soft errors are caused by the retention failures of the cell, and hard errors are due to the limited endurance of the cell. In the presence of both soft and hard errors, the reliability of ReRAM array, especially for its unique cross-point structure, becomes a serious design challenge. Specifically, there is no isolation between cells in a cross-point array, and thus a single cell failure can affect the read/write noise margin when reading/writing a cell in the same row or column with one or more bad cells.

Most prior work on NVM reliability tackles either soft errors [106, 76] or hard errors [27, 28] assuming only a single type of error exists in the target NVM technology, which makes them less effective under some practical cases. For example, the solution for extending the lifetime of MLC PCM may not work well if the resistance drift speed of PCM is high enough to trigger the error correction code (ECC) [76] more frequently than the stuck-at-0 or stuck-at-1 hard errors. There-
fore, it is necessary to consider the co-existence of both soft errors and hard errors when designing an error resilient architecture. Conventionally, once an error is detected, a “rewrite-read-verify” (also called “write-verify”) is often involved to determine whether it is a hard error or soft error. However, this approach may bring in additional writes which further wear out the memory cells. Hence, it is critical to avoid such unnecessary writes.

The major contributions of this chapter are,

- We systematically studied the mechanisms of both soft and hard errors of ReRAM cells and proposed a unified model to characterize the behaviors of different types of failure.

- To the best of our knowledge, we are the first to study the impact of different types of failure on the reliability of a cross-point ReRAM array. We identify that some types of failure affect read noise margin most while others may affect worst-case write noise margin and write energy.

- We proposed an error resilient architecture to deal with both soft and hard errors for ReRAM design. A key innovation in our design is the hard error detection unit. We avoid the unnecessary writes by determining the error type based on the unique characteristics of retention failure (soft error) and each type of endurance failure (hard error).

- We are the first to analyze the impact of postcycling retention degradation on ReRAM lifetime under different error detection approaches.

### 7.1 Cell Failure in a Cross-Point Array

Reliability is a vital concern in the design of memory system. In the cross-point structure, the reliability issues come from two different sources: structural error and cell error. The structural error is determined by the special organization of the cross-point array. The impact of voltage drop, sneak current, write/read schemes, as well as data pattern on the array reliability are well studied in literatures [97, 79]. They show that the structural errors can be mitigated or eliminated with exhaustive worst-case design. On the other hand, because of the intrinsic characteristics
of ReRAM cells, the impact of cell errors is not avoidable. To implement a reliable ReRAM array, specialized detection circuitry are required. In this section, we first discuss the resistance switching behaviors of ReRAM cell. Based on the discussion, mechanisms and modeling of soft errors and hard errors of ReRAM cell are presented. Then, the impact of the cell errors at the array design is evaluated.

7.1.1 ReRAM switching mechanism

Several studies have been conducted to reveal the physical mechanisms of the resistance switching behaviors. The filamentary model is widely accepted to explain the resistance switching phenomenon in the ReRAM [5]: switchings between LRS and HRS are caused by the formation and rupture of the nanoscale conductive filaments (CFs) at the anode interface of the cell. A forming operation can be considered as a “preset” operation of the ReRAM cell. The schematic view of the switching mechanisms of ReRAM cell is illustrated in Figure 7.1.

During the forming step, a high voltage is applied across the cell. The dielectric soft breakdown in the materials generates a great amount of defects in the metal oxide layer, which form one or several CFs through the cell. At the same time, the anode becomes a reservoir of the oxygen ions. After the forming operation, the cell is in LRS, which is shown in Figure 7.1a. The RESET operation is shown in Figure 7.1b. In this step, the oxygen ions are forced back to the metal oxide layer and recombine with the oxygen vacancies (Vo). In this case, the CFs are “cut off”
and the cell becomes HRS, which is shown in Figure 7.1d. In contrast, the SET operation can be considered as a reversed process of the RESET operation. As shown in Figure 7.1c, the SET operation regenerates the CFs by a large electric field. In this case, the cell switches back to the LRS.

### 7.1.2 ReRAM soft errors and hard errors modeling

Soft errors of the ReRAM cell come from the retention failure. The retention failure is a recoverable upset of the resistance of the cell. The retention failure can either be a sudden resistance drop of the HRS cell (HRS failure) or an abrupt resistance increasing of the LRS cell (LRS failure). The retention failure behaviors result from the random generation of the Vo (HRS failure), and the recombination of Vo with oxygen ions (LRS failure). Both of them imply that the retention failure is a stochastic process. Theoretically, either the HRS failure or the LRS failure can happen, but in most practical cases the LRS failure dominates under low current operation [107, 108]. Given the operating range of write current in our design, the soft errors are dominated by the LRS failure (“1”-to-“0” flip).

In order to quantify the retention failure behavior, the cumulative failure probability is employed. A simplified model of the cumulative failure probability can be expressed as,

$$F(t) = 1 - (1 - p)^{at}$$  \hspace{1cm} (7.1)

where α is a constant value, t is the retention time, and p is the Vo loss probability.

A recent study [108] pointed out that the LRS retention are well maintained after $10^4$ write cycles but it can degrade by 10X after $10^6$ write cycles. We are the first to consider such effect in system-level ReRAM research.

Different from soft errors, the hard errors result from the limited endurance of the ReRAM cell compared to traditional DRAM/SRAM technologies. The endurance failure is caused by a gradual resistance change over the write cycles. According to different behaviors and physical mechanisms, the endurance failures are classified into three categories [109],

1. **Type I Failure**: This failure is caused by the generation of extra oxide layer at the anode during the SET operations. This layer prevents the movement of the oxygen ions and results in $R_{LRS}$ increment or $R_{HRS}$ decrement.
2. Type II Failure: The programming voltage generated extra Vo, which directly increases the diameter of the CFs. In this failure, both of the \( R_{LRS} \) and the \( R_{HRS} \) decrease gradually.

3. Type III Failure: This failure results from the undesired consumption of the oxygen ions at stored in the anode. In this case, the combination probability of Vo and oxygen ions will reduce. Thus the \( R_{HRS} \) decreases while the \( R_{LRS} \) keeps constant.

We proposed a unified model of different types of endurance failure, in which the resistance change can be expressed as,

\[
R = R_0(1 + \frac{sgn(c - c_0) + 1}{2}\beta(c - c_0)^\gamma)
\]  

(7.2)

where \( R_0 \) is the initial resistance of LRS or HRS, \( c_0 \) is the start cycle that the endurance degradation is observed, and \( \beta \) and \( \gamma \) represent the direction and rate of the resistance change. The results in Figure 7.2 show that our model with different parameters fit well with experimental data for each failure type [109].

### 7.1.3 impact of different types of failure on a cross-point array

A soft error is a recoverable error and is essentially a resistance state transition without applying external voltages. We conclude that soft errors can only affect the information stored in the cells where the endurance failures arise, and do

---

**Figure 7.2.** Types of endurance failure (hard errors) in an ReRAM cell: (a) Type I, (b) Type II, and (c) Type III.
Table 7.1. Parameters of a Cross-Point Array

<table>
<thead>
<tr>
<th>Metric</th>
<th>Description</th>
<th>Typical Value(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A_{cell}$</td>
<td>Cell Size</td>
<td>$4F^2$</td>
</tr>
<tr>
<td>$R_w$</td>
<td>Wire Resistance</td>
<td>0.65Ω</td>
</tr>
<tr>
<td>$V_{write}$</td>
<td>Voltage of selected wordline during Write</td>
<td>±2V</td>
</tr>
<tr>
<td>$V_{write}/2$</td>
<td>Voltage of half selected wordlines/bitlines</td>
<td>±1V</td>
</tr>
<tr>
<td>$V_{SB}$</td>
<td>Voltage of selected bitline</td>
<td>0</td>
</tr>
<tr>
<td>$V_{read}$</td>
<td>Read voltage</td>
<td>0.5V</td>
</tr>
<tr>
<td>$K_r$</td>
<td>Nonlinearity of ReRAM Cell</td>
<td>40</td>
</tr>
<tr>
<td>$N$</td>
<td>Number of wordlines or bitlines</td>
<td>128,256,512</td>
</tr>
</tbody>
</table>

Figure 7.3. Read noise margin degradation in various array sizes for (a) type I failure, (b) type II failure, (c) type III failure.

not affect the other cells in the cross-point array. To overcome the soft error problem, normally some form of the ECC is introduced. We will discuss such design overheads in Section 7.2.

Compared to the soft errors, the hard errors are more serious in a cross-point structure. In general, the reliability concerns about the hard errors are in three aspects: (1) the decreased ratio of $R_{HRS}/R_{LRS}$ may degrade the read noise margin and eventually results in a read failure. This problem appears in all the three types of failure; (2) the reduction of $R_{LRS}$ increases the amount of sneak current and thus reduces the worst-case voltage drop on the furthest cell in a cross-point array. This can cause a write failure of the selected cell [79]; (3) the reduction of $R_{LRS}$ also increases the total energy consumption of a cross-point array during the write operation. There are chances that all the activated arrays are under worst-case or near worst-case scenarios, and the total power consumption for a given chip may violate the peak power budget. Breaking power limits will result in unexpected IR drops or excessive current, and even make electro-migration worse etc. (2) and (3)
only exist in type II failure in which the $R_{LRS}$ decreases over write cycles.

Figure 7.3 shows the read noise margin over cycles with various array sizes for different types of failure. The baseline parameters of a cross-point array in summarized in Table 7.1. We also assume that there is no variation in the initial resistance and resistance degradation rate of the cells in a cross-point array. In other words, we fix the constants in Equation 7.2. As seen in Figure 7.3, for type I and III failure, the resistance noise margins degrade gradually because either the $R_{HRS}$ decreases or/and the $R_{LRS}$ increases. However, the trend is different for type II failure. As its $R_{LRS}$ starts to increase earlier than its $R_{HRS}$ starts to decrease, the resistance ratio is boosted and the sensing margin is improved. Even after its $R_{HRS}$ starts to decrease, its read noise margin may continue to go up a little (i.e. by 5%) over a few cycles until a high reduction ratio of $R_{HRS}$ is reached. In fact, the reduction of $R_{LRS}$ helps the cross-point array maintain a reasonable read noise margin in type II failure, compared with type I and III failure. The larger the array size is, the earlier its sensing margin goes below the sensing boundary.

To ensure successful write operations in a cross-point ReRAM design, the cross-point array is always designed for the worst case: (1) $V_{\text{write}}$ is large enough so that the furthest cell has enough voltage drop to switch its state given the worst-case data pattern stored in other cells in the cross-point array; (2) $V_{\text{write}}$ can not exceed twice the threshold switching voltage to ensure that the half-selected cell which has a voltage drop of $V_{\text{write}}/2$ is not disturbed; (3) the overall write energy does not break the power limits. It has been identified that $R_{LRS}$ is the key parameter for designing a cross-point array in terms of worst-case voltage drop and write energy.
Since the $R_{LRS}$ is not affected in type III failure, the worst-case write noise margin and write energy is well maintained over cycles for such type of failure. Figure 7.4 illustrates the worst-case voltage on the furthest cell in a cross-point array over cycles with various array sizes for type I and II failure. Not surprisingly, the voltage drop becomes better over time for type I failure as its $R_{LRS}$ continues to increase. However, the reduction of $R_{LRS}$ poses a significant reliability issue on type II failure. For example, the voltage drop of a 512 $\times$ 512 array can go below half of $V_{write}$ after $10^5$ write cycles, and will inevitably cause a write failure [79]. The problem is alleviated in smaller array sizes, but a 256 $\times$ 256 array cannot work reliably after $10^7$ write cycles even its read noise margin is still acceptable according to Figure 7.3b.

The write energy of a cross-point array is much higher than its 1T1R counterpart because all the cells and wire resistance in a cross-point array are consuming energy during the write operation. Given the peak power budget and number of activated arrays simultaneously, the write energy of a cross-point array should not exceed an upper bound. It is straightforward that the worst-case write energy occurs when all the cells in a cross-point are in LRS. Figure 7.5 illustrates such worst-case write energy of a cross-point array over cycles with various array sizes. For type I failure, the worst-case energy goes down as its $R_{LRS}$ increases over time. For type II failure, the worst-case energy can increase by several times with the reduction of $R_{LRS}$. For example, the write energy of a 512 $\times$ 512 array doubles after $10^5$ write cycles.
In summary, ReRAM with type I and type III failure suffers from small read noise margin problems and encounters occasional read failures as the resistance ratio of cells shrinks, but they are almost write failure free once the worst-case design is determined during manufacturing. For type II failure, the write failure is a more severe problem due to the reduction of its $R_{LRS}$. Even writing a good selected cell may fail if many half-selected cells have reduced $R_{LRS}$. For all types of failure, there is a clear trade-off between the lifetime and the array size of a cross-point array. The smaller the array size is, the longer the lifetime is. There is an important choice to make for balancing reliability and density at the design stage.

\section*{7.2 soft and hard error resilience design}

Most prior work on NVM reliability tackles either soft errors \cite{106, 76} or hard errors \cite{27, 28}. However, a reliable cross-point ReRAM design should be resilient to both soft and hard errors. In our design, we proposed an error resilient architecture to improve the reliability of the system. We classify each failure event into soft error or hard error based on the characteristics of each error type. The basic flow of our detection-handle mechanism is listed step by step,

- If the ECC, which can be as simple as a single-error correcting and double-error detecting (SEC-DED) code, detects a correctable error during a read operation, the data are sent to the read request after correction. At the same time, the hard error detection is triggered.
- The hard error detection unit will determine whether the failed cell is a retention failure (soft error) or an endurance degradation (hard error). It will take extra steps if necessary. The design of the hard error detection unit heavily depends on the failure type, and will be discussed later in this Section.
- If the failure event is identified as a hard error, the hard-error tolerating technique must be involved, such as ECP \cite{28} or DPM \cite{27}. In our design we adopt a light version of ECP. For type II failure, there is some extra work to do. This is because if we simply leave the bad cell as it is, this cell
can serve as a half-selected cell when writing a different block address next time. After accumulating a lot of bad cells, there are chances that some of the half-selected cells in a write operation are bad cells and they have lower $R_{LRS}$ than other normal LRS cells, resulting in an unintentional write failure even if the selected cell works perfectly. Therefore, we will apply a RESET pulse on any bad cell in type II failure once it is detected. This ensures the resistance of the bad cell is not smaller than the initial low resistance of a normal cell.

### 7.2.1 hard error detection unit

Most hard error detection works in a “rewrite-read-verify” (or “write-verify” for short) way. The approach is briefly explained as follows. After the error is specified, the correct data are written back, and immediately followed by a read operation. If the read succeed and ECC reports no error, then the previous error was identified as a soft error. If the ECC reports an error in the same location again, this cell will be marked as a bad cell.

The key drawback of this approach is that there is one additional write operation every time when the ECC is triggered. If the soft error rates are high and they trigger the ECC more frequently than the hard errors do, the cells will wear out even earlier.

Our solution to this problem is to identify the error type by leveraging the rational behind each error type in ReRAM. One key observation as discussed in Section 7.1.2 is that the soft error of ReRAM cells with low write current are dominated by LRS failure (“1”→“0” flip). If there is no write failure and the ECC detects that a cell is identified as “1” while it is supposed to be “0”, then it cannot be a soft error. In other words, an erroneous “1” in type I and III failure is determined as a hard error. However, the characteristics of each failure type in ReRAM endurance degradation make the design of the hard error detection unit different from each other.
Figure 7.6. Hard error detection for type I failure.

7.2.1.1 type I failure

Figure 7.6 demonstrates the hard error detection mechanism for type I failure. As mentioned, an erroneous “1” is determined as a hard error. While for an erroneous “0”, there are two possibilities: an increased $R_{LRS}$ due to cycling or an abrupt LRS-to-HRS jump due to retention failure. Given that the resistance changes gradually and the erroneous cell was not marked as a bad cell, the increased $\Delta R_{LRS}$ is expected to be much smaller than an abrupt LRS-to-HRS jump. Therefore, the erroneous cell is read again and its read current is compared with another reference current $I_{ref}$ which is smaller than the one used for normal read operation ($I_{ref0}$). If its read current $I_{read}$ is greater than $I_{ref}$, it indicates that the cell has a modest resistance value, indicating a hard error. Thus ECP will mark it as a bad cell. Otherwise the cell has a high resistance value, indicating a soft error.

The design is essentially based on a three-level output sense amplifier. Normally the reference current $I_{ref0}$ for read operation is generating by averaging the current from two complementary cells: one cell in LRS while the other in HRS, that is,

$$I_{ref0} = \frac{I_{LRS} + I_{HRS}}{2} \quad (7.3)$$

In our design, the reference current $I_{ref}$ is generated from a partially-RESET reference cell.

$$I_{ref} = m \times I_{HRS} \quad (m > 1) \quad (7.4)$$

where $m$ is the factor of multiplication. The area overhead of such sense amplifier design is estimated to be less than 5% of the total NVM chip area [110].

In order to evaluate the effectiveness of our hard error detection unit, we choose an ECP_6 scheme with 6 correction pointers that can mark up to 6 bad cells in a
512-bit memory block. We assign 10% variations for both $\beta$ and $\gamma$ in Equation 7.2. First we assume constant soft error rates without postcycling retention degradation. The soft error rate is calculated using Equation 7.1 and the LRS retention at $85^\circ$C are extrapolated from published data [107]. We do not assume wear-leveling techniques in our simulations. Figure 7.7 shows the fraction of memory blocks that survive given the number of block writes to ReRAM built in different array sizes. The baseline ECP without any hard error detection assumes every error reported by ECC is marked as a bad cell and occupies one correction pointer in ECP. Therefore, it has the worst lifetime though there is no associated hardware, performance and energy overhead for detecting hard errors. Compared to the baseline, the “write-verify” detection scheme improves the lifetime significantly because the soft errors are identified, avoiding unintentional usage of correction pointers in ECP. The detection mechanism we proposed further enhances the endurance curve because it does not involve unnecessary writes during the detection procedure. Our approach is more effective for ReRAM with larger cross-point array as they are more vulnerable to errors. Another advantage of our scheme over the conventional “write-verify” scheme is that the latency and energy overheads associated with the unnecessary writes are saved given that reads are much faster and more energy-efficient than writes in NVM.

Then the postcycling retention degradation is also considered, and the LRS retention after $10^6$ is assumed to be 10% that of the initial LRS retention time [108]. The sustainable number of writes to ReRAM $N_{WM}$ is defined such that 80% of

---

**Figure 7.7.** Percentage of available capacity versus the number of writes to ReRAM for type I failure, with various array sizes of (a) 128x128, (b) 256x256, (c) 512x512.
Figure 7.8. Sustainable number of writes to ReRAM for type I failure with and without considering postcycling retention degradation, for various array sizes of (a) 128x128, (b) 256x256, (c) 512x512.

Figure 7.9. Hard error detection for type II failure.

memory blocks survive in ReRAM after $N_{WM}$ block writes. Figure 7.8 demonstrates the reduction of $N_{WM}$ caused by postcycling retention degradation for different error detection approaches. The major reason is that as the soft error rate goes up after $10^6$ cycles, the ECC are triggered more frequently. For the baseline ECP without any hard error detection, a lot of retention-failure-induced errors will wear out the memory cells soon. As we can see from Figure 7.8 that the average $N_{WM}$ with postcycling retention is only about 71% of that without considering postcycling retention. For “write-verify” detection scheme, higher soft error rates mean the hardware detection unit will test (and write) the erroneous cell more frequently, and thus wear them out earlier. The average $N_{WM}$ with postcycling retention is about 89% of that without considering postcycling retention. With our proposed write-free hardware detection unit, the average $N_{WM}$ with postcycling retention is about 95% of that without considering postcycling retention.
7.2.1.2 type II failure

The unique characteristic of type II failure is its \( R_{LRS} \) can decrease over cycles, resulting in a write failure. In practice, more than one LRS cell in a memory block can be mapped to the same cross-point array, and they are fully selected during the write operation. The current of these fully biased LRS cells contributes the most to the total current of the selected wordline and thus causes a significant voltage loss along the wire. If the furthest selected cell fails to have enough voltage drop, the primary reason is that some of the fully selected LRS cells have degraded \( R_{LRS} \) values. The secondary reason is that there have accumulated a large number of degraded LRS cells among the half-selected cells. Our design also tries to avoid the latter case.

Figure 7.9 illustrates the hard error detection mechanism for type II failure. As the \( \Delta R_{LRS} \)-induced write failure is the primary concern in the reliability issue, each time the ECC detects an error, it will read all the “1”s in the memory block that mapped to the same cross-point array with the erroneous cell again. The read current of these cells (including the erroneous cell) is compared with a large reference current level \( I_{\text{ref}} \) to determine whether if there is notable \( R_{LRS} \) reduction in the cell. If the read current of any LRS cell is greater than the \( I_{\text{ref}} \), the cell is marked as a bad cell by ECP. Then we apply a RESET pulse on the bad cell. As long as the cell is RESET to a higher level than the initial \( R_{LRS} \), the cell will not be responsible for any write failure no matter whether it is fully selected or...
Figure 7.11. Sustainable number of writes to ReRAM for type II failure with and without considering postcycling retention degradation, for various array sizes of (a) 128x128, (b) 256x256, (c) 512x512

Figure 7.12. Hard error detection for type III failure.

half-selected during a future write operation. If no LRS cell in the array show significant $R_{LRS}$ degradation, then we determine an erroneous “1” is caused by decreased $R_{HRS}$ (hard error) and the cell is simply marked as a bad cell by ECP. No RESET operation is required for such cell since it is already in its HRS.

Figure 7.10 shows the percentage of surviving memory blocks for type II failure without postcycling retention. The improvement of our design over the “write-verify” detection scheme is more significant than it is for type I failure. This is because checking more than one cells after the ECC is triggered provides a wider error coverage range. For ReRAM with 512x512 arrays, our design extends the lifetime more than 12% compared to conventional “write-verify” detection scheme.

As Figure 7.11 illustrates, considering the postcycling retention degradation, our design shows more significant lifetime improvement over the baseline and the “write-verify” detection scheme.

7.2.1.3 type III failure

Detecting a hard error in type III failure is relatively easy since its $R_{LRS}$ almost keeps constant. In this case, an erroneous “1” indicated by the ECC is identified to be a hard error as a result of $R_{HRS}$. In contrast, an erroneous “0” is identified
Figure 7.13. Percentage of available capacity versus the number of writes to ReRAM for type III failure, with various array sizes of (a) 128x128, (b) 256x256, (c) 512x512.

Figure 7.14. Sustainable number of writes to ReRAM for type III failure with and without considering postcycling retention degradation, for various array sizes of (a) 128x128, (b) 256x256, (c) 512x512.
to be a soft error as a result of retention failure. Figure 7.13 shows the percentage of surviving memory blocks for type III failure. Given that our detection approach for this type of failure is almost free, the improvement over the conventional hard error detection schemes is significant. Again, our design only introduces less than 5% reduction of $N_{WM}$ when considering postcycling retention while the baseline and the “write-verify” detection scheme has more than 30% and 10% reduction in $N_{WM}$.

7.3 Summary

ReRAM is a promising candidate for next-generation non-volatile memory technology. The high density cross-point structure is the most attractive memory organization for low-cost ReRAM. However, due to the lack of isolation between cells in a cross-point array, the resistance degradation over write cycles observed in ReRAM cells will have a significant impact on the reliability of such structure. Our analysis shows that type I and III failure suffer from read noise noise margin degradations while type II failure has to deal with additional write issues including reduced voltage drop and increased write energy. Instead of a write-intensive hard error detection mechanism, we design effective hard error detection units for each failure type without involving write operations. For uncycled LRS retention, our design enables a soft error and hard error resilient architecture which extends the lifetime of ReRAM by up to 75% over the design without hard error detections and up to 12% over the design with “write-verify” detection mechanism. Considering postcycling retention degradation, our design show more significant lifetime improvement.
Chapter 8

Conclusion

Several non-volatile memory technologies are emerging these days. Among them, ReRAM seems to be one of the most promising candidate with attractive properties including high density, fast access, good scalability and comparability with CMOS technology. Therefore, it have drawn the attention of the computer architecture community and challenged the role of DRAM and NAND flash in mainstream memory hierarchy. Given the desirable properties of ReRAM, innovative research are required on designing the next-generation of high-performance and low-power computing systems using this technology.

This dissertation tackles this topic from several different but highly-related aspects of views.

First, several array/macro ReRAM models with different simulation accuracy and speed requirement were built and described in the first part of this dissertation. The motivation of building these models comes from the current situation that ReRAM is still in its proof-of-concept stage and most of the prototypes demonstrated covers a large range of device-level parameters. Without a detailed analysis tool, it is very challenging to evaluation the overall performance, energy, cost of ReRAM-based memory.

Second, since the straightforward implementation of ReRAM prototype is likely to introduce large area overhead, it is necessary to use circuit-level design enhancement to alleviate such drawbacks. Thus, the second part of this dissertation focuses on using circuit-level techniques to mitigate the aforementioned shortcomings of large footprint of sense amplifiers/write drivers. In addition, macro-level evalua-
tion is performed to show that the proposed techniques are all effective in reaching their goals.

Third, architectural-level case studies of adopting ReRAM are conducted in this dissertation. These case studies demonstrate how ReRAM can be architected in traditional memory hierarchy and greatly improve either the performance or the power efficiency. These case studies validate the ideas of replacing traditional main memory and storage technologies (i.e., DRAM, NAND flash, and HDD) with ReRAM. Such results are expected to be the driving force to push the maturity of these technologies.

Last but not least, reliability is a major concern in ReRAM. The fourth part of the dissertation shows that different types of cell failures have different impacts on cross-point ReRAM design, and it is much more efficient to design a error-tolerant system according the different characteristics of each type of failure. The proposed hard error detection unit design could help improve the lifetime of ReRAM-based memory system significantly.

We hope the work of this dissertation would be useful and have its impact on NVM-related research on future high-performance and low-power computing systems. The successful outcome of this research would have an enormous economic and social benefits. It will provide the design guidelines for enabling both large capacity storage system and high performance non-volatile main memory, which are beyond the present state-of-the-art. Consequently, it will enhance nearly every digital device available today from consumer electronics to enterprise electronics. It may also spawn new applications involving the computation on the exascale of data, e.g., data mining, machine learning, visual or auditory sensory data recognition, bio-informatics, etc.
Bibliography


URL http://www.mentor.com/products/fpga/model

URL http://www.synopsys.com/Tools


URL http://www.micron.com/products/dram/ddr3-sdram


URL http://www.cs.virginia.edu/stream


Vita

Cong Xu

Cong Xu was born in southern China. He received his B.S. degree in Microelectronics from Peking University. He joined the Ph.D. program of the Department of Computer Science and Engineering at Pennsylvania State University in 2009. His research interests include non-volatile memory technology, computer architecture, and emerging neuromorphic computing. He has authored and co-authored more than twenty conference papers, 1 journal paper, and 1 book chapter during his Ph.D. program. He has won two best paper awards. He is the student members of both IEEE and ACM.