TU Delft

Delft University of Technology

Logic and arithmetic computation-in-memory accelerators Based on memristor devices

Singh, A.

DOI
10.4233/uuid:387a9451-e85c-475b-999d-b3a6cfb330d3

Publication date
2024

Document Version
Final published version

Citation (APA)
Singh, A. (2024). *Logic and arithmetic computation-in-memory accelerators: Based on memristor devices*. [Dissertation (TU Delft), Delft University of Technology]. https://doi.org/10.4233/uuid:387a9451-e85c-475b-999d-b3a6cfb330d3

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright
Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy
Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

This work is downloaded from Delft University of Technology.
For technical reasons the number of authors shown on this cover page is limited to a maximum of 10.

Logic and Arithmetic Computation-in-Memory Accelerators

BASED ON MEMRISTOR DEVICES

LOGIC AND ARITHMETIC COMPUTATION-IN-MEMORY ACCELERATORS

BASED ON MEMRISTOR DEVICES

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus Prof. dr. ir. T.H.J.J. van der Hagen,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen op woensdag 29 Mei 2024 om 10:00 uur

door

Abhairaj SINGH

Master of Science in Electrical Engineering,
Technische Universiteit Delft, the Netherlands
geboren te Chandigarh, India.

Dit proefschrift is goedgekeurd door de

promotor: Prof. dr. ir. S. Hamdioui
promotor: Prof. dr. R.V. Joshi
copromotor: Dr. R.K. Bishnoi

Samenstelling promotiecommissie:

Rector Magnificus,	voorzitter
Prof. dr. ir. S. Hamdioui,	Technische Universiteit Delft
Prof. dr. R.V. Joshi,	IBM Watson, USA / Technische Universiteit Delft
Dr. R.K. Bishnoi,	Technische Universiteit Delft
Onafhankelijke leden:	
Prof. dr. ir. G. Gaydadjiev	Technische Universiteit Delft
Prof. dr. H. Corporaal	Technische Universiteit Eindhoven, the Netherlands
Prof.dr. F. Catthoor	IMEC / KU Leuven, Belgium
Dr. A. Sebastian	IBM Research Zurich, Switzerland
Prof. dr. ir. L.C.N. de Vreede	Technische Universiteit Delft, reservelid

TU Delft Delft University of Technology

Keywords: Computation-in-Memory, Memristor, Circuit Design, Logic, Multiply and Accumulate, Matrix-Vector Multiplication, Analog-to-digital convertors, Edge-AI
Printed by: Ipskamp Printing, the Netherlands
Front & Back: by A. Singh & M. Kalsi

Copyright © 2024 by A. Singh

ISBN 978-94-6366-875-0

An electronic version of this dissertation is available at
http://repository.tudelft.nl/.

Dedicated to my parents and my lovely wife

CONTENTS

Summary xi

Samenvatting xiii

Acknowledgements xv

PART I: THE FOUNDATION 1

1 Introduction 3
1.1 Motivation . . . 4
1.1.1 Recent Trends in Computing . . . 4
1.1.2 Computing Architecture and CMOS Device Bottlenecks . . . 6
1.1.3 Memristor-based CIM and its Potential . . . 9
1.2 Research Topic . . . 11
1.2.1 Hierarchy of Memristor-based CIM . . . 11
1.2.2 Micro-architectural Challenges . . . 13
1.3 Thesis Contribution . . . 15
1.4 Thesis Outline . . . 17

2 Background 21
2.1 Classification of Memory-centric Computation . . . 22
2.1.1 Computational Units . . . 22
2.1.2 Memory Technology for CIM . . . 24
2.2 CIM Architecture for Logic & Arithmetic Operations . . . 28
2.3 Challenges & Opportunities in Memristor-based CIM . . . 30
2.3.1 Accuracy Challenges: Dealing with Non-idealities . . . 30
2.3.2 Efficiency Challenges: Achieving Stringent Requirements . . . 32
2.4 State-of-the-art & its Limitations . . . 35
2.4.1 Logic Accelerators . . . 35
2.4.2 Arithmetic Accelerators . . . 39

PART II: CIM-BASED LOGIC ACCELERATORS 49

3 Cascaded Logic using PCM 51
3.1 Introduction . . . 52
3.2 Database Query using Scouting Logic . . . 52
3.3 Experimental Demonstration . . . 54
3.3.1 Fabrication of Crossbar Arrays . . . 55
3.3.2 Basic Electrical Characterization . . . 57
3.3.3 Sources of Inaccuracy . . . 58

3.3.4 Optimal Reference Currents . . . 61
3.3.5 Implementation of Query Operation . . . 63
3.4 Simulations Studies . . . 64
3.4.1 A Circuit-level Device Model . . . 64
3.4.2 Large-scale Crossbar Simulations . . . 67
3.5 Database Query using Majority Logic . . . 68
3.6 Cascaded Database Query . . . 70
3.7 Conclusion . . . 73

4 Robust Logic Accelerator using STT-MRAM 75
4.1 Introduction . . . 76
4.2 Design Methodology . . . 77
4.2.1 Motivation and Approach . . . 77
4.2.2 Reference Generators . . . 77
4.2.3 Reference Arrangement . . . 78
4.3 Results . . . 79
4.3.1 Chip Prototype and Experimental Setup . . . 79
4.3.2 Circuit-level Simulation Results . . . 79
4.3.3 System-level Results . . . 82
4.4 Conclusion . . . 83

5 Referencing-in-Array Scheme 85
5.1 Introduction . . . 86
5.2 Proposed CIM Accelerator . . . 88
5.2.1 Overview . . . 88
5.2.2 Proposed CIM Architecture . . . 88
5.2.3 Design Optimizations . . . 92
5.3 Simulation Setup & Results . . . 93
5.3.1 Setup . . . 93
5.3.2 Circuit-level Results and Comparisons . . . 95
5.4 Conclusion . . . 96

6 Accelerating RRAM Testing with CIM 97
6.1 Introduction . . . 98
6.2 High-density RRAM . . . 99
6.2.1 RRAM Technology . . . 99
6.2.2 RRAM for Memory . . . 99
6.3 Targeted Faults . . . 100
6.4 DFT Schemes Proposed for ETD Faults . . . 101
6.4.1 Concept . . . 101
6.4.2 DFT Design and Implementation . . . 102
6.5 Reconfigurable DFT Schemes Proposed for HTD Faults . . . 106
6.5.1 Concept . . . 106
6.5.2 Design and Implementation . . . 107
6.6 DFT Validation and Test Development . . . 109
6.6.1 Setup, Fault Modeling and Analysis . . . 109

6.6.2 DFT Validation . . . 111
6.6.3 Test Procedure . . . 114
6.7 Prior Arts and Comparison Results . . . 115
6.7.1 Qualitative Comparison . . . 115
6.7.2 Quantitative Comparison . . . 116
6.8 Discussions and Future Directions . . . 116
6.9 Conclusion . . . 117

7 Multi-operand XOR using RRAM 119
7.1 Introduction . . . 120
7.2 Voltage-to-time based MOXOR-CIM . . . 121
7.2.1 Overview . . . 122
7.2.2 Proposed MOXOR-CIM Architecture . . . 122
7.2.3 Design and Implementation . . . 124
7.3 Results . . . 129
7.3.1 Simulation Setup . . . 129
7.3.2 Circuit-level Results and Comparison . . . 130
7.4 Conclusion . . . 131

8 Outlook and Discussion 133
8.1 Introduction . . . 134
8.2 Outlook . . . 134
8.3 Discussion on Design Explorations . . . 137
8.4 Conclusion . . . 139

PART III: CIM-BASED ARITHMETIC ACCELERATORS 141

9 Low-power IFC-based ADC for CIM 143
9.1 Introduction . . . 144
9.2 Proposed SRIF-ADC . . . 146
9.2.1 Design Methodology . . . 146
9.2.2 Design Components . . . 147
9.3 Results . . . 153
9.3.1 Simulation Setup . . . 153
9.3.2 Simulation Results . . . 155
9.3.3 Comparison with Prior Works . . . 157
9.3.4 Arithmetic Design Implementations using SRIF-ADC . . . 158
9.3.5 Discussion . . . 159
9.4 Conclusion . . . 161

10 Compact low-power CCO-based ADC for CIM 163
10.1 Introduction . . . 164
10.2 CIM Design . . . 165
10.2.1 ADC Design Methodology . . . 165
10.2.2 ADC Input-Output Characteristics . . . 168
10.3 CIM Implementation and Characterisation . . . 169
10.3.1 Chip Prototype and Experimental Setup . . . 169

10.3.2 ADC Characterisation Results . . . 169
10.4 System-level Results . . . 170
10.4.1 System-level Validation . . . 170
10.4.2 Comparison Results . . . 171
10.5 Conclusion . . . 172

11 Voltage-to-Time based ADC for 2T2R RRAM **173**
11.1 Introduction . . . 174
11.2 Proposed CIM Design . . . 176
11.2.1 Voltage-sensing based Computing . . . 176
11.2.2 Bitcell Configuration and Input Activation . . . 176
11.2.3 Dual-ramp based Voltage-to-Time Converter (VTC) . . . 177
11.2.4 Ramp Generator and Adaptive Low-power VCD . . . 179
11.3 Results . . . 181
11.3.1 Chip Prototype and Setup . . . 181
11.3.2 Circuit-level Results . . . 181
11.3.3 System-level Results and Comparison . . . 181
11.4 Conclusion . . . 182

12 Outlook and Discussion **183**
12.1 Introduction . . . 184
12.2 Outlook . . . 184
12.3 Discussion on Design Explorations . . . 186
12.4 Conclusion . . . 189

PART IV: THE CONCLUSION **191**

13 Summary and Future Directions **193**
13.1 Summary . . . 194
13.2 Future Directions . . . 197

Curriculum Vitæ **221**

List of Publications & Patents **223**

SUMMARY

Conventional computing systems involve physically separated storing and processing units. To perform the processing, data is shuttled from the storing unit to the processing unit followed by the actual processing, and the processed data is shuttled back into the storing unit. Unfortunately, this data shuffling contributes significantly to the overall latency and energy consumption of the system. Computation-in-memory (CIM) offers a promising alternative that can process the data within the storing unit, thereby, alleviating the need for data shuffling. This can potentially lead to high energy efficiency and high throughput computation. In addition, emerging non-volatile memristors provide an excellent storing element that can inherently perform the computation in CIM while also retaining the data value. Memristor-based CIM has been vastly explored to perform certain data-intensive computational tasks for numerous applications related to artificial intelligence (AI), Big Data, and data encryption while realizing high-performing, low-power micro-architectural solutions. This thesis first identifies the existing key challenges related to emerging memory technology and CIM memory array, and the circuit design of periphery logic to achieve low-power and high-performing CIM units. Thereafter, it presents several micro-architectural solutions to build CIM accelerators that can perform logic and arithmetic operations in an efficient manner.

Identifying the challenges of memristor-based CIM: The thesis briefly summarizes the key aspects of a memristor-based CIM architecture that can perform certain logic and arithmetic operations. First, the key components of a typical CIM architecture are presented, highlighting the design considerations to build these components. An account of computational accuracy and efficiency share of each of these components is presented supported by a comprehensive literature survey while highlighting the underlying concepts of each component. This is followed by identifying the key challenges in achieving stringent efficiency metrics required by the targeted applications while dealing with non-idealities. Finally, the limitations of the state-of-the-art solutions that target these challenges are highlighted, thus, providing the motivation to develop outperforming solutions that are presented in the thesis.

Developing micro-architectural solutions for efficient CIM-based logic accelerators: This part of the thesis presents several logic accelerators that can perform (N)OR, (N)AND, and XOR operations in a fast and energy-efficient manner. **FIVE** different solutions target different aspects of performing CIM-based logic, whereby, scaling the number of operands per cycle and the memory size, and expanding to different types of operations that can be supported while addressing the non-idealities. State-of-the-art solutions are comprehensively outperformed by our proposed solutions in terms of energy efficiency, performance, and the maximum number of operands that can be accurately performed in a single cycle.

Developing micro-architectural solutions for efficient CIM-based arithmetic accel-

erators: This part of the thesis presents several compact arithmetic accelerators that can perform multiply-and-accumulate (MAC) operations with low energy consumption. **THREE** different analog-to-digital converter (ADC) topologies are presented with two of them demonstrated with a chip prototype. Optimizing the memory structure is also explored to ensure accurate generation of analog output in the CIM memory array and to optimize the A/D conversion. A comparison of the proposed solutions with the state-of-the-art is presented, highlighting the promise of the developed micro-architectures.

SAMENVATTING

In conventionele computersystemen zijn het geheugen en de processor gescheiden. Voor een bewerking moet daarom data uit het geheugen naar de processor gestuurd worden en moet vervolgens het resultaat weer naar het geheugen gestuurd worden. Dit heen-en-weersturen van data draagt aanzienlijk bij aan hogere latentie en energiegebruik van het systeem. Computation-in-memory (CIM), waarbij data in het geheugen kan worden gemanipuleerd zonder het heen en weer te sturen, biedt een veelbelovend alternatief dat mogelijk kan leiden tot een systeem met hoge energie-efficiëntie en hoge bandbreedte. Bovendien kunnen opkomende niet-vluchtige memristors die inherent berekening kunnen uitvoeren in CIM en daarnaast hun opgeslagen waarde behouden zonder energie te verbruiken de basis vormen van een dergelijk systeem. CIM op basis van memristors is uitgebreid onderzocht voor data-intensieve toepassingen gerelateerd aan kunstmatige intelligentie (AI), massadata en cryptografie, waarbij microarchitecturen met goede prestaties en laag energieverbruik kunnen worden gerealiseerd. Dit proefschrift begint met het identificeren van de belangrijkste uitdagingen op het gebied van geheugentechnologie en CIM geheugenmatrices en het ontwerp van elektronische netwerken voor perifere logica om energiezuinige en goed presterende CIM systemen te realiseren. Vervolgens worden een aantal microarchitecturen voor CIM hardwareaccelerators die efficiënt logische en rekenkundige bewerkingen kunnen doen gepresenteerd. **De uitdagingen van CIM op basis van memristors**: In dit hoofdstuk worden kort de belangrijkste aspecten van een CIM-achitectuur die bepaalde logische en rekenkundige bewerkingen kan doen beschreven. Eerst worden de belangrijkste onderdelen van een typische CIM-architectuur gepresenteerd, met de nadruk op ontwerpoverwegingen voor het bouwen van deze onderdelen. Op basis van uitgebreid literatuuronderzoek worden de rekennauwkeurigheid en efficiëntie van deze onderdelen beschreven aan de hand van de onderliggende modellen voor hun werking. Dit wordt gevolgd door een beschrijving van de belangrijkste uitdagingen voor het halen van strenge efficiëntie-eisen en het rekening houden met niet-idealiteiten als voorgeschreven door de beoogde toepassingen. Ten slotte worden de beperkingen van de nieuwste systemen die deze problemen proberen op te lossen aangehaald, wat de motivatie vormt voor de ontwikkeling van de architecturen die in dit proefschrift worden gepresenteerd. **Ontwikkeling van microarchitecturen voor efficiënte logische accelerators**: In dit hoofdstuk worden enkele logische accelerators die (N)OR, (N)AND en XOR bewerkingen snel en energie-efficiënt kunnen uitvoeren beschreven. Vijf verschillende architecturen richten zich op verschillende aspecten van logica gebaseerd op CIM. Hierbij wordt gekeken naar het vergroten van het aantal bewerkingen per klokcyclus, het vergroten van het geheugen en het vergroten van het aantal soorten bewerkingen dat gedaan kan worden terwijl er rekening gehouden wordt met niet-idealiteiten. Onze voorgestelde architecturen hebben duidelijk hogere energie-efficiëntie, snelheid en maximum aantal bewerkingen per klokcyclus

dan de nieuwste architecturen van het moment. **Ontwikkeling van microarchitecturen voor efficiënte rekenkundige accelerators voor CIM**: In dit hoofdstuk worden enkele compacte rekenkundige accelerators die multiply-accumulate (MAC) bewerkingen met laag energieverbruik kunnen uitvoeren beschreven. Drie verschillende topologieën voor analoog-digitaalomzetters (ADC) worden gepresenteerd, waarvan twee door middel van een prototype chip worden gedemonstreerd. Het optimaliseren van de geheugenstructuur wordt ook onderzocht om accurate analoge signalen in de CIM geheugenmatrix te garanderen en om de analoog-naar-digitaal omzetting te optimaliseren. In een vergelijking met de nieuwste architecturen van het moment wordt de veelbelovende toekomst van de ontwikkelde microarchitecturen aangehaald.

ACKNOWLEDGEMENTS

My PhD journey is coming to an end with a bundle of mixed feelings of highs, lows, and existential crises packed into a neat little five-year package. On the one hand, I would never want to go through it again but on the other hand, I am extremely glad that I did it. I believe it is in human nature, at any given present time, that we tend to selectively recall and cherish the (most) pleasant memories of the past and constantly worry about unfortunate (but highly improbable) events that may never happen. Anyway, here I am, standing at the finish line, realizing that while I may not rush to repeat the experience, it has undeniably shaped me in profound ways. Ultimately, as the PhD journey nears its conclusion (or as the football season approaches its climax), the paramount goal is to attain the PhD and embark on the path toward becoming a successful scientist (or securing those crucial three points to clinch the championship).

First and foremost, I would like to thank my esteemed promotors, **Prof. dr. ir. Said Hamdioui** and **Prof. dr. Rajiv V Joshi**, for providing an excellent opportunity and platform to conduct my research. The right amount of push and desire to follow a flexible, honest, and exciting path towards my diploma can not be understated. Said's mentorship, evident from our collaboration during my Master's thesis, solidified my decision to pursue a PhD within his vibrant group at TU Delft. Apart from just working within my wonderful group at TU Delft, countless prospects in every step of my journey were filled with invaluable opportunities like working with some of the best European research universities and institutes; TU Eindhoven, RWTH Aachen, KU Leuven, ETH Zurich, IBM Research and IMEC. This has been made possible by my promotors who are constantly working hard (in the background) towards securing the funding for our research, driving and ensuring academic and industrial collaboration to conduct research. Thanks for your unwavering support and visionary leadership, Said and Rajiv. A special mention is owed to my co-promotor, **Dr. Rajendra Bishnoi**, whose guidance and technical expertise have been invaluable throughout my PhD. Our shared research interests and cultural background, and long walks in the evenings have fostered a unique bond, one that transcends mere academic collaboration. Your enduring presence throughout this journey serves as a testament to our shared commitment to academic excellence and personal growth. My sincere thanks to you, Rajendra.

Next, I am deeply grateful for the enriching collaborations I have had throughout my academic journey, particularly during my involvement with the MNEMOSENE project and my time at IBM Research. I extend my sincerest thanks to **Prof. Dr. Francky Catthoor, Dr. Stephan Menzel, Prof. Dirk Walters, Prof. Dr. ir. Henk Corporaal, and Dr. Amirreza Yousefzadeh** for their invaluable contributions and insightful discussions throughout our collaboration. I owe a tremendous debt of gratitude to my manager at IBM Research, **Dr. Abu Sebastian**, whose guidance and engaging discussions profoundly influenced my research trajectory. Abu, your mentorship has been invaluable, and I am

immensely grateful for it. Spending a significant portion of my PhD journey at IBM Research was truly transformative, and I am eagerly anticipating the opportunity to continue our collaboration post-PhD. Moreover, I feel incredibly fortunate to have had the privilege of working alongside exceptionally talented, experienced, dedicated, and humble colleagues at IBM. Special thanks to **Pier Andrea, Pritish, Judith, Manuel, Irem, Abbas, Geethan, Michael, Francesco,** special thanks to **Jesse** for the Dutch translation, and my lunch companions and my football gang, for fostering a supportive and stimulating work environment. I am also indebted to my dear friends and colleagues at TU Delft, including **Sumit, Yash, Mahdi, Moritz, Muath, Abdullah, Amin, Abdulghadar, and Anteneh,** for the enriching experiences we shared. Additionally, I extend heartfelt appreciation to the supportive administrative staff of the graduate school, **Ada van Gulik and Sanne Alblas,** for their unwavering assistance and reassurance during challenging times. Together, each of you has played an indispensable role in shaping my academic and professional journey, for which I am deeply thankful.

I am incredibly fortunate to have a support network that extends beyond the realm of academia. My time in the Netherlands has been enriched by the presence of amazing friends. To **Prabhav, Rishabh, Rupak, Yashwant, Pinakin, Manasa,** your friendship brings light and joy to my life. And to my dearest long-lasting friends from DCE, **Arjun** and **Aakash**, incredible 16 years and counting. Finally, to **Mr. Sudhanshu Jaitly** for igniting my passion for Chemistry 18 years ago and for your enduring wisdom and kindness today. Our connection over the years has meant the world to me.

I would also like to express my deepest gratitude to my family-in-laws for welcoming me with open arms, chole bhature, and single malt in your family. I thank my lovely **mother-in-law** for many honest and inspiring conversations and my **father-in-law** for always being welcoming and sharing his passion for food. Also, special thanks go to **Nani Ji** for her unconditional love, pampering, and positivity. Also, I want to thank my brother-in-law **Abheeuday** and his wife, and my lovely ex-roommate, **Taavishe**, who have stood by me as both cherished friends and family members, constantly (and painfully) reminding me of the beauty of life (especially) outside the realm of pursuing PhD. And for infinite Instagram reels and skiing lessons, canoeing, exotic cuisines, and especially that one-and-only time forced football session. Thank you and I love you both. You two hold a special place in my heart and life.

This dissertation is dedicated to my parents, who have never shied from showing their unwavering support, expressing their unconditional love, and lending a compassionate ear to my joys and struggles. I learned how to love from you, my dearest **Mummy**, and I loved to learn from you, my amazing **Pitaji**. Thank you for incessantly pushing me to achieve my dreams. Thank you for the long long talks over the phone, > three-hour long walks when we are together, and for being my constant support system. Also, my brother **Shivu** and **Bhabhiji**, for your unparalleled support even from far away in a cold distant continent, (a direct eight hours of flight that you could not make for my defence), for the frequent long calls where you admire how our parents are the most tech-savvy, always-open-to-learn attitude and how I am the best brother you could ever wish for. I missed you here but I still love you.

Finally, to my incredible soon-to-be-doctor wife, **Monique**, you have been my muse, tirelessly supporting and encouraging me every step of the way. Despite the challenges

of our respective PhD journeys, you have remained by my side, with unwavering love and commitment. Teaching me to be grateful, patient, humble, and fashionable. I do feel the infinite hate radiating when we are hiking up or playing football or tennis, but also the love and joy when we are hiking back, returning from football and tennis sessions, or watching countless space-time videos. Moving forward, I cannot wait to cherish more precious moments together. I love you more than words can express.

Abhairaj
Delft, May 2024

PART I: THE FOUNDATION

This part includes the following chapters.

- ***Chapter 1**: Introduction*
- ***Chapter 2**: Background*

This part is partially based on [1–3].

1

INTRODUCTION

Real-time data acquisition and processing have become an integral part of our daily activities. One of the key contributors is the amount of data that can be generated and processed to enable an incredible number of computing applications, thereby, revolutionizing the way our society functions. In the past few decades, computing systems have been addressing the ever-increasing demand for data-intensive applications with transistor downscaling and efficient parallelism. However, transistor downscaling is reaching its geometrical and economic limits and algorithmic parallelism is becoming stagnant in conventional computing paradigms. Emerging computation-in-memory (CIM) paradigm using memristor-based device technologies is promising as it can enhance the computation efficiency, solve the data transfer bottleneck, and at the same time deliver high energy efficiency using normally-off/instant-on attributes of their devices. In this chapter, we first describe the motivation to shift from the CMOS storage-based conventional computing paradigm to memristor-based CIM to perform certain fundamental data-intensive computational tasks. Thereafter, we briefly describe the underlying principle of memristor device technology and CIM and highlight the key challenges to realizing these tasks. This is followed by the contribution of this thesis summarizing the proposed micro-architectural solutions for efficient memristor-based CIM computing. Finally, we conclude with a brief outline of the thesis.

1

1.1. MOTIVATION

We live in a digital world. We can not simply function without the computing capability of technology. Healthcare, consumer electronics, sports, automotive, military, banking, and crypto-currency are just a few sectors where computing resides at the core and is responsible for not only technological but also socio-economical advancements. In this section, we highlight the recent trends in computing, point out the major bottlenecks of the conventional computing paradigm, and motivate the scope of our research for alternative ways of performing the computations while exploring emerging device technologies.

1.1.1. RECENT TRENDS IN COMPUTING

For the past several years, there has been an exponential growth in the amount of data, and processing of this data has been possible with crucial advancements in artificial intelligence (AI) and semiconductor technology. Avenues such as computing at the edge, and machine-learning algorithms pertaining to AI have found their way into personalized healthcare systems such as medically equipped wearable devices, smart homes such as ambient intelligence, perseverance, and ubiquitous computing, and consumer electronics such as smartphones to carry out day-to-day activities. With more data and processing it with smart algorithms, computing systems have become smart and can carry out much more sophisticated tasks to make our daily lives easy. Fig. 1.1a shows a general trend depicting an exponential growth in data [4] and Fig. 1.1b shows the increasing number of smart internet-of-things (IoT) devices [5, 6] that are equipped with AI-based algorithms over the years. With the current estimation of $12.4B, global invest-

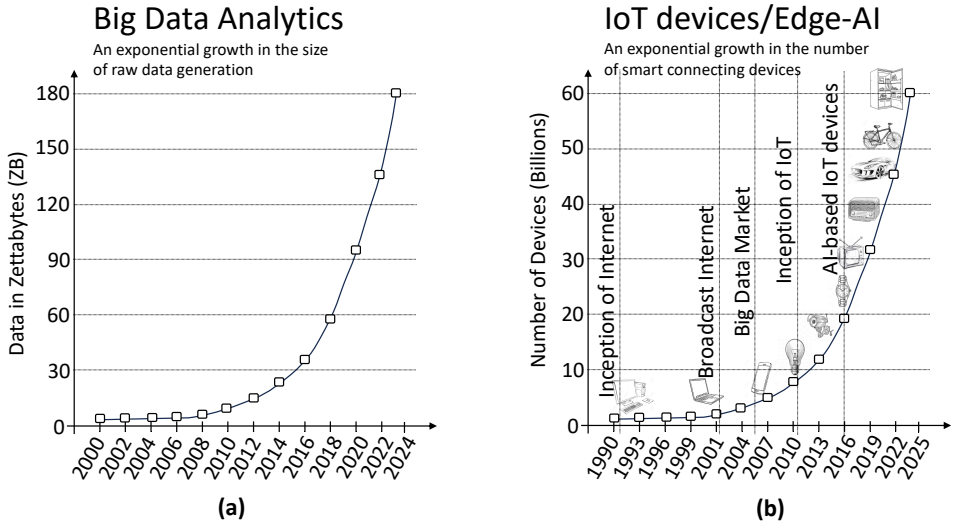


Figure 1.1: (a) Exponential growth in the amount of data produced (data from [4]) and (b) in the number of smart devices over the years(data from [5, 6]).

ments involving AI-related applications are expected to top $230B by 2025 [7], highlighting the impact of AI in the times to come.

To realize these applications involving data-intensive computational tasks, aggressive technology downscaling of CMOS has provided the platform to build cost-efficient fundamental blocks and massive parallelism has been exploited using dedicated processing units to build energy-efficient and high-performing computing architectures. CMOS downscaling has contributed to nearly 2× improvement every five years in terms of transistor density and performance. In other words, the number of transistors that can fit in a unit area nearly doubled every five-year period, thereby, reducing the manufacturing cost by nearly 2×. This also improves the performance as this geometrical shrinking reduces the effective RC loading leading to reduced propagation delays. In addition, the downscaling of the threshold voltage of the transistors aided voltage downscaling over the years which has been the key to effectively reducing the overall energy consumption. On the other hand, massive parallelism has been exploiting the different components of the computing architecture. Traditional von Neumann architecture involves physically separated memory and processing units, as shown in the leftmost figure in Fig. 1.2. The data to be processed resides in the memory units and is transferred to the processing unit to perform the computing, followed by transferring back the data into the memory units. One of the major efficiency bottlenecks is this data communication, which contributes to nearly two orders of magnitude latency and energy overhead compared to the actual computing process. To improve computing efficiency over the years, this communication has been traditionally mitigated by essentially bringing memory units closer to the processing units [8]. Moving from left to right in Fig. 1.2, we show the progression of bringing memory closer to the processing element (PE) over the years. By adding one or several levels of faster but smaller SRAM memory units, known as cache memories, closer to the PEs, the effective data movement is reduced while accommodating the growing size of the datasets. The rightmost diagram in Fig. 1.2 illustrates the extremities of this approach by placing the memory registers dedicated to each PE, thus, parallelizing several data-intensive tasks such as matrix-vector-multiplications (MVM) and bitwise logic operations to achieve high efficiency.

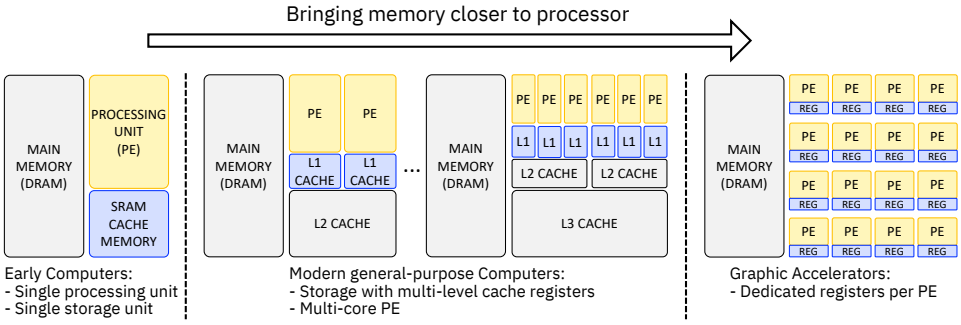


Figure 1.2: The evolution of the computing systems over the years to minimize memory-processor data movements. Working set data resides in blue colored memory units.

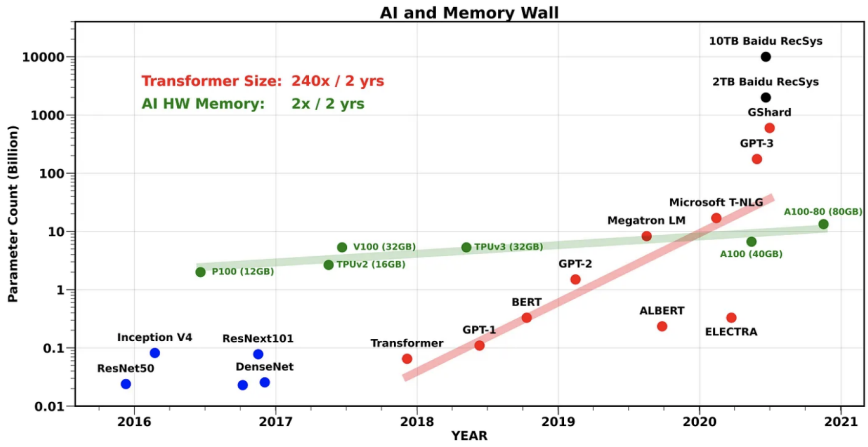


Figure 1.3: Exponential growth of AI model sizes and demanding compute, memory, and bandwidth requirements of AI workloads [9].

Unfortunately, most data-intensive applications exhibit a quite large working data set which cannot fit in the register-stage but also not in the L1 or L2 caches. Industrial players have introduced intermediate cache levels in the SoCs to mitigate some of these limitations and the main memory and LLC/SLC working memory space in 3D packaged SiP systems. Moreover, recent developments in chiplet technology has been showing huge promise, where different technology nodes combine together for better overall performance. However, the recent trend is that even the main memory working space is not sufficient because the requirement starts to exceed tera bytes (TB). For instance, the newest AMD MI300x (which has 2.4× memory of Nvidia's H100) still can not store the entire weight on a single chip, so larger on-chip memory would give the AI chip more flexibility when dividing the training workload and avoid communication [9]. The trend in Fig. 1.3 clearly shows that on-chip memory size was growing at a slower pace compared to the model size (directly contributes to weight size). This is also true for modern AI algorithms, especially for training but also more and more for the inference phase due to the introduction of activation data buffers in ResNET type topologies but also in applications such as transformers.

1.1.2. COMPUTING ARCHITECTURE AND CMOS DEVICE BOTTLENECKS

The data communication required between the storage and the processing unit in the traditional von Neumann architectures is becoming a memory-processor bottleneck, thereby, stagnating the advantages we once relied on in the past few decades. In addition, CMOS downscaling is reaching its geometrical and economic limitations and is struggling with excessive sub-threshold leakage and device-related reliability challenges. Below is a brief description of the architectural challenges faced by conventional von Neumann architectures.

ARCHITECTURAL CHALLENGES

Conventional computing systems implementing data centric applications still rely on frequent data shuffling between the physically separated storage and processing units. Several conventional architectures such as multi-level caches (L1, L2, L3 caches) [15, 16], processor-in-memory i.e processing in SRAMs [17, 18], memory accelerators [19, 20] attempt to bring memory closer to the processing unit. However, these architectural techniques only prolong the inevitable incompetency of the conventional approaches to address the following limitations.

- **Memory Wall** - This accounts for the large data transfer between storage and processing units. Fig. 1.4a The read latency of the SRAM/DRAM cells downscaled at a much slower rate, close to 50% slower per year, compared to the processing latency [10]. Therefore, the limited bandwidth of memories remains a key challenge.

- **Power Wall** - This is concerned with the huge amount of dynamic power consumed during the back-and-forth data transfer. The energy share of the data shuffling is becoming exceedingly higher than the energy share of performing actual computational tasks [14, 21].

- **Instruction Level Parallelism (ILP) Wall** - This is due to the saturation of operating frequency upscaling and exhaustion of parallel processing algorithms developed over the years [14]. Fig. 1.6 shows the operating frequency has stagnated around 2-3GHz for the past decade and exploiting parallelism through multi-core systems are exhausting close to a few hundreds. The figure also captures that this plateauing of operating frequency has also impacted the peak performance of a single-thread compute unit.

On the other hand, below is a brief description of the technological challenges faced by conventional CMOS-based memory unit cells.

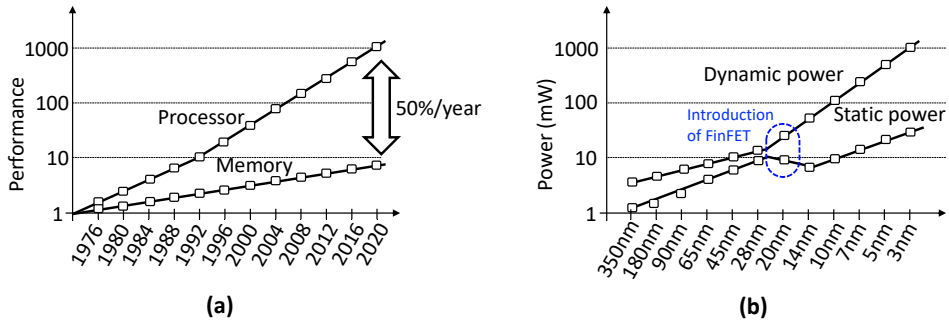


Figure 1.4: (a) The increasing gap in performance for memory and processing units over the years (data from [10]). (b) The share of static (leakage) power and dynamic power with technology scaling (data from [11, 12]).

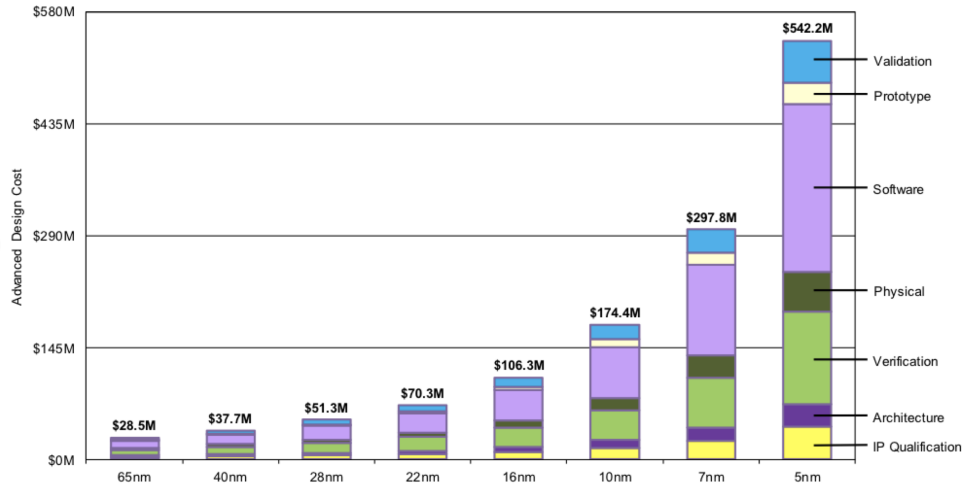


Figure 1.5: The cost of developing CMOS with technology scaling [13].

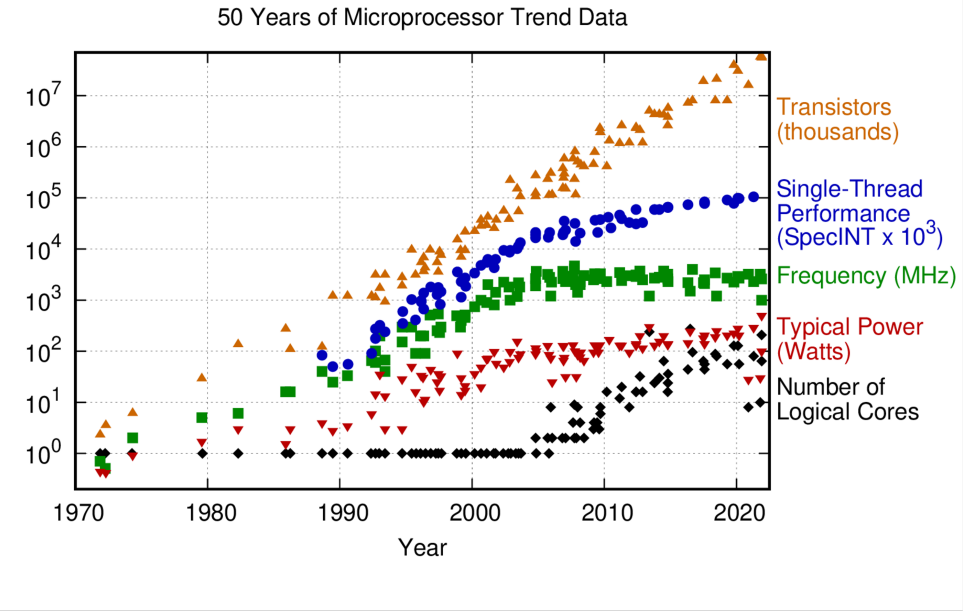


Figure 1.6: Microprocessor trend data over 50 years with technology scaling, instruction level parallelism, operating frequency and power consumption [14].

TECHNOLOGICAL CHALLENGES

While the increasing static power (leakage) component with respect to dynamic power component of the planar CMOS node was revived by the superior gate-controlled FinFET-based CMOS devices, significant improvement due to CMOS downscaling is reaching its limitations due to the following challenges or *walls*.

- **Static Power (Leakage) Wall** - One of the key aspects of reducing the dynamic or switching power of the CMOS transistor has been the downscaling of the operating voltage. However, over the years, the threshold voltage of the transistors has not reduced in the same manner, thereby, increasing the sub-threshold leakage of the transistors [11, 12]. Although the overall power per operation has been reducing over the period of time, an increased leakage power share, especially memory leakage, compared to the dynamic power limits the energy consumption downscaling, as shown in Fig. 1.4b.

- **Cost Wall** - With the geometrical downscaling already slowing down over the years due to physical limitations, it is also becoming more and more expensive in terms of manufacturing process complications and low yield [13]. Hence, the cost advantage of cramming more transistors in the same silicon area is hitting a wall. Fig. 1.5 shows the an exponential growth in cost of manufacturing advanced technology nodes.

- **Reliability Wall** - With the reduced physical size of the transistors, the device-to-device variations and souring manufacturing failure rates have drastically reduced the reliability of CMOS devices [22]. In addition, the lifecycle of the transistors is shrinking with the advanced nodes compared to the preceding nodes.

These severe limitations in the conventional architecture and the device technology call for a timely need for alternatives.

1.1.3. MEMRISTOR-BASED CIM AND ITS POTENTIAL

Computation-in-memory (CIM) involves performing the compute operation in the same physical location where the data resides. In other words, it shifts the location of the unit performing the compute operation f on data D to the memory units itself, as shown in Fig 1.7. Moreover, each column in the memory unit can potentially perform independent computing operations simultaneously, implying a massive throughput with $\mathcal{O}(1)$ time complexity. CIM performs analog-based computing where at least one of the inputs resides in the memory array. Typically, a multi-row READ or input activation is performed in CIM which accumulates elementary compute results into a common access line in the analog domain by exploiting Ohm's law and Kirchhoff's circuit law. In a typical multi-tile CIM, an input pre-processing block and an output post-processing block reside in the periphery logic to facilitate data conversions in CIM. The input pre-processing block converts the digital input into select signals or activation signals in the analog domain to enable analog-based computing inside the memory array. The memory array generates the outputs in the analog domain. In a multi-tile CIM, the output post-processing block converts these analog outputs back to the digital domain for subsequent data communication .

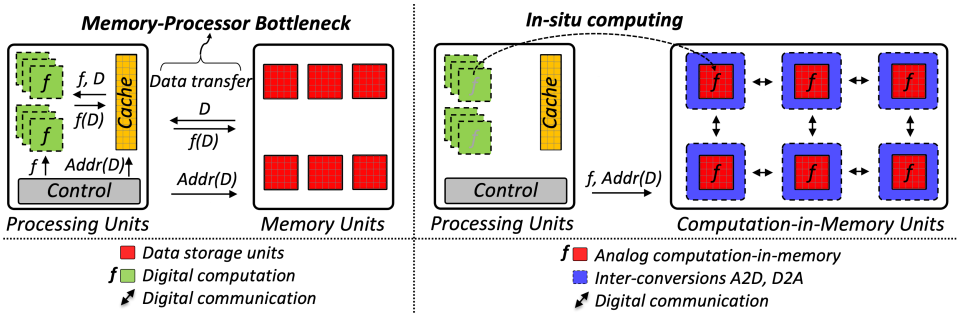


Figure 1.7: CIM (on the left) circumvents the massive memory-processor bottleneck encountered by conventional von Neumann architecture (on the right).

CIM can be realized using both conventional and emerging memory technologies. The memory technologies used for CIM can be broadly classified as charge-based memories and non-charge-based memories [23]. In conventional charge-based memories such as dynamic random access memory (DRAM), static random access memory (SRAM), and Flash, data is stored as and is represented by the amount of charge. Whereas, the emerging non-charge-based memories that were introduced nearly a decade ago include different types of storage elements that store data represented by resistance and these can be distinguished by their physical mechanism; these include resistive random access memories (RRAM) [24], spin torque transfer magnetic random access memories (STT-MRAM) [25], and phase change memories (PCM) [26]. These non-charge-based memories storing data as resistance values are commonly known as memristors. Memristor-based CIM flavors store the information in the form of resistance states, which can be in high resistance or low resistance states [23]. The resistance state can be changed using RESET or SET electrical pulses. In CIM, non-charge-based memory technologies have several key advantages over charge-based memory technologies such as zero leakage, non-volatility, density, and scalability. Due to these unique advantages, memristor-based CIM solutions have been massively explored and this fact is highlighted by the number of articles published over the past decade. Therefore, this thesis focuses on CIM using non-charge-based memories, or the so-called memristor-based CIM.

Memristor-based CIM has been explored to realize *certain fundamental* key data intensive kernels for a variety of applications [29, 30]. Most of the widely used kernels can be classified into two broad classes; 1) bulk-bitwise logic or simply logic operations: these include logic (N)OR, (N)AND, XOR that are involved in applications related to database query, encryption, DNA sampling, bio-informatics etc., and 2) arithmetic operations: these include matrix-vector multiplication (MVM) which is the most fundamental operation performed in any neural network implementation for applications related to deep learning, edge-AI etc. Note that not all limitations (or *walls*, as described earlier) are resolved using memristor-based CIM. There are significant prospects in accelerating certain fundamental kernels in an energy-efficient manner, however, there are several challenges, as will be described in the next chapter.

One of the critical feature of these applications that is typically exploited by these

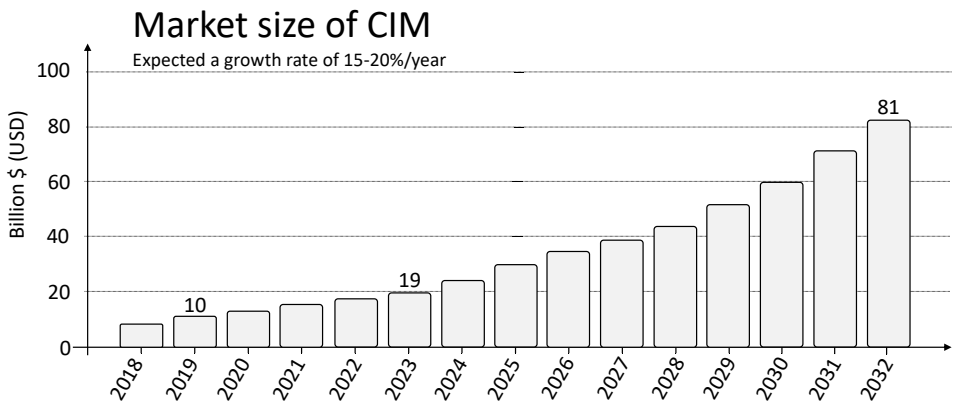


Figure 1.8: CIM warrants a massive market growth rate of 15-20% per year (data from [27, 28]).

memristor-based CIM architectures is the data/weight stationary nature of the operations, where the data is not frequently changed of a decent course of time. This mitigates the affect of exceedingly slow and energy inefficient write operations of the memristors on the overall inference efficiency. For instance, in database query applications, the ratio of the number of write operations and read operations is far less than 1 [31], and during inferences (post-training and fine-tuning of weights), the ratio of the number of write operations and read operations is low as well [23].

Over the past few years and years to come, CIM is projected to be deployed in a number of applications related to sales and marketing, predictive analysis, supply chain management, fraud detection, and processing in sectors including banking services and insurance, IT and telecom, retail and e-commerce, healthcare and life sciences, transportation and logistics, government and defence, media and entertainment, to name a few [27, 28]. Over the period of my doctoral studies, the global CIM market already rose from ~10 billion USD back in 2019 to ~19 billion USD in 2023. This is projected to rise at an incredible rate of 17%, reaching close to ~81 billion USD in 2032, as shown in Fig. 1.8. Notably, this has also been reflected in the rising number of scientific publications in top-tier conferences and journals in the past decade. Therefore, efforts in realizing CIM architecture based on emerging memristor devices are expected to have a significant impact on next-generation technological advancements.

1.2. RESEARCH TOPIC

This section describes the hierarchy of memristor-based CIM, highlights the challenges related to micro-architecture, and summarizes the research topic of this thesis.

1.2.1. HIERARCHY OF MEMRISTOR-BASED CIM

Here, we cover the hardware and software abstraction levels involved in memristor-based CIM while focusing on performing logic and arithmetic operations for data-intensive

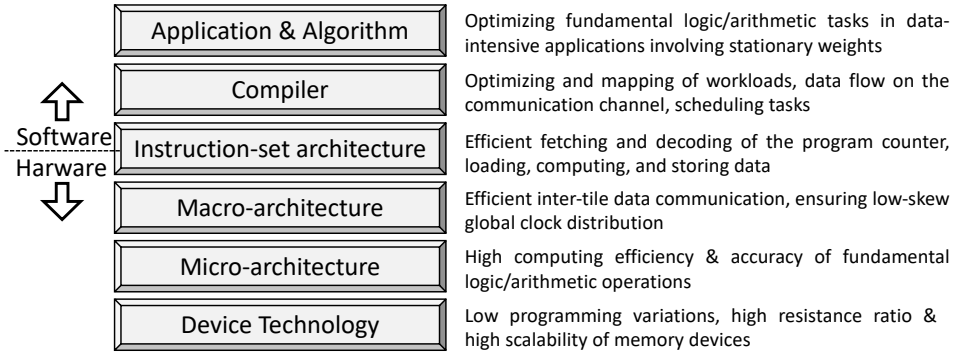


Figure 1.9: Hardware and software stacks of memristor-based CIM with related focus areas.

applications described above. Fig. 1.9 provides a brief overview of the different abstraction levels.

- **Device technology**: There are several memristor devices under research that are being explored for investigating the potential of CIM. Although CIM works using SRAMs[32–40], DRAM [41, 42], NAND/NOR Flash [43, 44] exists in literature, the primary focus of this thesis is investigating the three most popular memristor devices *i.e.,* RRAM[45–54], STT-MRAM[55, 56] and PCM[57–60]. Given the massive involvement of high-tech companies like Google, IBM, Facebook, and Microsoft in developing next-generation memristor devices, one can expect the characteristics associated with each device technology to improve as the research field matures. Memristors can typically be arranged in one-transistor-one-resistor (1T1R), and 2T2R configurations in CIM architectures. To perform CIM, the key efficiency metrics are read latency, read energy, programming variations, and resistance drift.

- **Micro-architecture**: Typically, complex instructions are executed on the arithmetic logic (e.g. 32- or 64-bit MAC) involving local single registers which typically enables the active wires to be shorter. On the one hand, CIM aggravates this and exhibits issues related to long wires within the memory array. Fortunately, it alleviates the movement of operands thanks to the *in-situ* charge accumulation based computing. CIM micro-architecture comprises the circuit design to perform typical memory operations (read and write) as well as computing operations. Typical tasks performed within the micro-architecture involve developing bitcell structure, and periphery logic for read, write and verify operations, and performing computational tasks such as logic and arithmetic operations utilizing these circuits. In a multi-tile architecture-based CIM, the inputs are in the digital domain, computation is performed in the analog domain, and output is converted back to the digital domain for data communication between different tile architectures.

- **Macro-architecture**: Large datasets related to AI and database query applications typically require several micro-architecture tiles to accommodate on a single chip and, unfortunately, may still require off-chip communication. Nevertheless, it is essential

to maximize on-chip storage capacity to minimize the disadvantages of frequent off-chip communication. The size of each micro-architecture is limited by wire parasitic and electromagnetic (EM) effects. For instance, multi-tile CIM accelerators are implemented to realize multi-layered neural networks for image classification applications. Designing a macro-architecture mainly involves ensuring efficient inter-tile data communication, ensuring low-skew global clock distribution to the CIM tiles, while potentially maximizing the advantages of ultra-high bandwidth of results produced by CIM.

- **Instruction-set architecture**: Instruction-set architecture (ISA) is the hardware/software interface block concerning the functional definition of operations, data locations, etc., and the precise description of how to invoke and access them. The most defining components of the von Neumann-based ISA follow repetitive fetching and decoding of the program counter, loading, computing, and storing data, significantly changing for CIM as at least one of the operands remains in the same physical location.
- **Compiler**: A compiler primarily includes optimization and mapping of workloads, data flow on the data communication channel, description of the tasks, and scheduling of these tasks for high efficiency. In image classification applications, for instance, CIM highly optimizes the data-intensive MVM operational tasks between the activation vectors and stationary neural network weight matrices. Therefore, compilers developed for CIM include the mapping of these stationary weights on the CIM tiles to respect data dependencies that can allow efficient scheduling of the different tasks. Depending on the target efficiency metrics of CIM performing these MVM operations, the scheduling of tasks including digital scaling, activation functions, etc. can be optimized.
- **Application and Algorithms**: Data-intensive applications where the majority of the computational tasks involve a significant share in data transfer activities of fixed data values can benefit from CIM. For instance, in database management systems, the data is occasionally updated with new information. The queries that are typically performed on these stationary datasets comprise primitive logic operations and data loading and storing can heavily dominate the efficiency metrics. In another instance, in an image classification application, large weight sets of the neural networks are primarily fixed after training. During inference, the efficiency of the MVM operations between the incoming activation inputs (images) and these weight sets is significantly dominated by the data loading and storing operations. To realize these applications in CIM, these large datasets can be mapped onto the memory arrays, thus alleviating the expensive data shuffling and algorithms can be developed around this concept.

The focus of this thesis is to investigate the micro-architectural aspects of CIM which revolves around one of the most essential *make-or-break* factors in realizing memristor-based CIM. Therefore, we provide a brief description of the challenges related to the micro-architecture of memristor-based CIM.

1.2.2. MICRO-ARCHITECTURAL CHALLENGES

Considerations related to the design and implementation of CIM micro-architecture often involve addressing many challenges related to memristor device technology and the

micro-architecture. These challenges are broadly divided into these two classes based on the ***source*** and can be summarized as follows (a detailed account of key challenges in presented in the next chapter *i.e.*, Chapter 2):

MEMRISTOR DEVICE TECHNOLOGY

Each of the different type of memristor devices offer a unique set of features that can be explored in CIM. However, each of them also suffers from various inherent limitations and challenges that need to be addressed. Below is a list of major challenges faced when memristor devices are used as storage and computing elements.

- **Resistance ratio**: In a CIM where data resides as binary storage, the difference in the resistance of the memristor device corresponding to the two binary states determines the sensing margin to distinguish the two states. A higher resistance ratio (RR) offers a higher sensing margin. Whereas, RRAM and PCM devices show a relatively higher RR, an STT-MRAM device has a theoretical upper limit of *two*, which is also known as the tunnel magnetoresistance (TMR). The highest reported effective RR for STT-MRAM is 1.76 [61]. This imposes a serious challenge to develop efficient and accurate read techniques.

- **Device variations**: The extent of device-to-device variations relates to the inherent mechanism of programming a certain memristor device [62]. For instance, an STT-MRAM device has a relatively much better-controlled switching mechanism compared to RRAM and PCM devices. This typically reduces the sensing margins that can lead to erroneous processing of operations. Resistance drift with time also adds to the device variations temporally.

- **Thick metal wires**: Memristors demand high switching voltage, typically ranging from 1.5V-2V for RRAM devices [63] and up to 4V for PCM devices [64]. This requires not only special thick-oxide transistors to supply such voltages but also low-resistance wires to avoid massive IR drop along the rows and columns. This leads to longer READ-/compute latency and higher READ/compute energy.

CIM MICRO-ARCHITECTURE

Optimizing the tile architecture can massively improve the efficiency and accuracy of the computing operations. However, realizing this can be quite demanding due to the following challenges.

- **Non-ideal wires**: Due to the way analog computing is performed, the signal reaching the sensing circuits residing in the periphery may suffer from varying parasitic effects. For instance, the RC delay, capacitive coupling, and IR drop significantly diminishes the available sensing margin to differentiate different states by the SA or ADC. These effects are also visible in activation signals from DACs or row decoders where the strength and timing of the signal reaching the destination bitcells can greatly vary and cause a major source of inaccuracies.

- **Variation**: This can be described as a circuit component exhibiting any change from the ideal electrical behavior. These variations can have several factors such as imperfect manufacturing processes, fluctuations in temperature, supply voltage(s), aging effects, etc. which can lead to device resistance and RC loading mismatch. This is particularly important for analog components such as SA, reference blocks, and ADC.
- **Pitch matching**: Realizing with potentially $\mathcal{O}(1)$ time complexity can be quite challenging in terms of matching the pitch of periphery logic with that of the bitcells. One can easily correlate it with SRAMs to realize the extent of this challenge, where circuits related to memory operations such as SA and write drivers are shared typically by four 6T SRAM bitcells. Implementing multi-bit ADCs, reference circuits, or dedicated SA to perform arithmetic or logic operations within a small pitch of memristor-based bitcells can be extremely challenging.
- **Energy consumption and latency**: One of the major challenges to develop low-power and low-latency computational units is the periphery logic involving analog-digital inter-conversions. Multi-bit ADCs and DACs typically consume more than 60-70% share of energy consumption in CIM-based arithmetic accelerators. Similarly, SA consume nearly 70-80% share in logic accelerators. With high demands of computational accuracy, this remains one of the key focus to realize low-power edge-AI and big data applications.

In summary, there are several showstoppers in realizing the promise of CIM architecture and addressing these can help us map realistic applications that can outperform state-of-the-art solutions. This leads to the research question that must be answered with the contribution of this thesis and it can be described as follows:

"Whether by addressing key challenges related to the design of memristor-based CIM micro-architectures and by optimizing these architectures to perform logic and arithmetic operations, we can outperform state-of-the-art solutions based on CMOS storing element and von Neumann architecture".

1.3. THESIS CONTRIBUTION

The thesis contributes towards improving the micro-architecture of memristor-based CIM to provide an accurate and efficient computing system. The contribution is divided into three parts; the first part of the thesis work highlights the key aspects and promise of CIM, classifying its various components and identifying the key challenges and existing limitations related to developing CIM tile architecture. The second part reports several memristor-based CIM works on accelerating the fundamental logic operations and energy-efficient computing. The last part reports several memristor-based CIM works on accelerating multiply-and-accumulate (MAC) operations by focusing on the design and development of efficient ADCs.

- **Perspective on memristor-based CIM**: This part is one of the most critical components that lay the foundation of this thesis work. This highlights the most critical components and related design considerations in developing efficient CIM architectures.

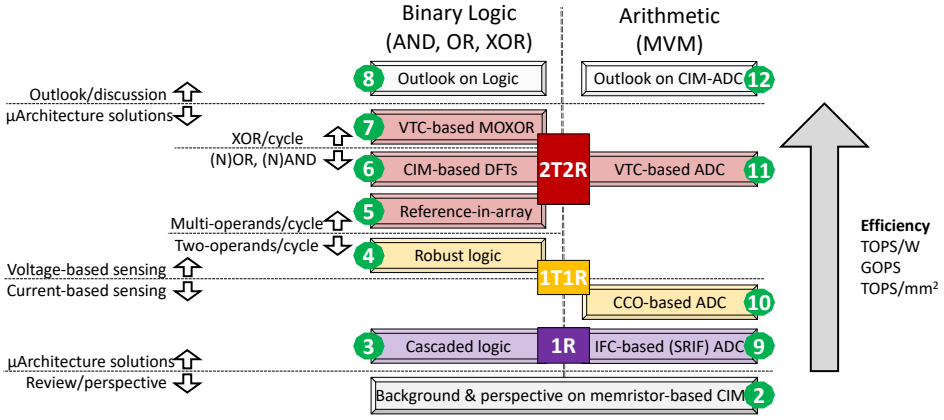


Figure 1.10: The outline of the chapters. The bitcell configuration groups different logic and arithmetic accelerators together. Type of sensing *i.e.*, voltage-based or current-based, type of logic operations and the number of operands supported per cycle *i.e.*, two- or multi-, classify different logic accelerators. The types of sensing also classifies different MAC accelerators.

There are several aspects that are covered while focusing on this in the following scientific articles:

- In [1], CIM architectures that have been explored to develop accelerators for low-power edge-AI applications are investigated. CIM is classified based on *where* data/operands reside, *where* computing is being performed, and *where* the outcome resides. It describes the key components of a typical CIM architecture; memory crossbar, input, and output processing units. Thereafter, it accounts for the most promising memristor technologies under investigation and highlights the challenges faced while implementing them in CIM. Finally, it dives deep into output processing units that are found to typically dominate the efficiency metrics. It accounts for different topologies of ADC designs that have been adapted for CIM. A detailed account of what one can expect from a certain topology is presented with respect to accuracy and important efficiency metrics.
- In [2], a summary of various potential operations (or kernels) that memristor-based CIM has been commonly investigated and related real-world applications is presented. The focus is on enlisting the non-idealities such as variations, wire parasitic, and non-zero low conductance states, and how certain solutions in literature address them. In [3], different aspects of CIM, including its classification, working principle, potential, and design-flow are discussed. The design-flow is illustrated through two case studies to demonstrate the potential of CIM in realizing orders of magnitude improvement in energy efficiency compared to conventional architectures. Finally, future challenges and research directions of CIM are covered.

- **CIM-based logic accelerators**: This part presents ***FIVE*** logic accelerators that target

real-world applications such as database query, testing, neuromorphic computing, and encryption [64–68]. The solutions address various challenges related to non-ideal memristor devices, non-ideal wires, and variations that typically lead to limited scalability in terms of array size and number of operands. With an aim to improve the throughput and energy efficiency of CIM-based logic accelerators, the solutions range from performing efficient two- and multi-operand (N)OR, (N)AND, and XOR operations to finally performing them within a single cycle. Results show the progression during the course of the thesis with each of the proposed solutions comprehensively outperforming our previously proposed solutions and at the same time, other state-of-the-art logic accelerators.

- **CIM-based arithmetic accelerators**: This part presents ***THREE*** arithmetic accelerators that target real-world applications such as neuromorphic computing and edge-AI [69–71]. The solutions investigate different topologies to develop efficient ADC designs while also optimizing the CIM tile to aid the conversion; current-controlled oscillator-based SRIF-ADC [69] and RO-ADC [70], and a voltage-to-time based ADC while exploring complementary data storage [71]. Results based on SPICE simulations and measurements conducted on chip prototypes show that each of the proposed solutions outperforms our previously proposed solutions and at the same time, other state-of-the-art MAC accelerators with favorable trade-offs in area efficiency, performance, and energy efficiency.

1.4. THESIS OUTLINE

The thesis is divided into four parts which are further divided into chapters. Fig. 1.10 shows how different chapters are connected to each other based on common bitcell configuration utilized and the type of sensing (whether it is voltage or current based) for the proposed accelerators. In addition, the proposed logic accelerators are also classified based on the type of operation and the number of operands supported per cycle. The details are as follows:

- **PART I: THE FOUNDATION**

  Chapter 2: Following the Introduction in Chapter 1, this chapter lays the foundation of the thesis. It begins by describing the fundamentals of conventional computing systems and how different flavors of CIM perform the computation tasks differently. It also touches briefly upon CIM based on volatile and non-volatile memory units and the inherent advantages of the two, while leaning towards emerging non-volatile memristors to develop next-generation computing. Thereafter, it presents the key components of a typical CIM architecture and their impact on overall computational accuracy and efficiency. A comprehensive account of the challenges of memristor-based CIM follows and the opportunities they provide to develop better computing systems. Finally, a study on state-of-the-art CIM-based logic and arithmetic accelerators is presented which are classified based on the challenges that they attempt to address. The major limitations of these solutions are highlighted which gives motivation to the work that is conducted as part of this thesis.

1

- **PART II: CIM-BASED LOGIC ACCELERATORS**

  Chapter 3: This covers the work presented in article [64] that targets database query applications. A cascaded logic design is presented that resides in the periphery logic of a selector-less PCM-based CIM which performs a series of logic operations of arbitrary length and operation type. This solution alleviates the need for expensive write-back schemes that have been proposed in prior-art solutions. The concept is also experimentally demonstrated using a 4x8 selector-less projected-PCM prototype with IBM.

  Chapter 4: This describes the work presented in article [65] that targets BNN-related applications. Given the inherent low sensing margins due to low TMR of STT devices, this work proposes an adaptive referencing mechanism to improve the sensing margin while performing logic operations in an STT-MRAM-based CIM. Reference signals are generated using multiple STT-MRAM devices and placed strategically into the array such that these signals can address the variations and trace the wire parasitics effectively. The concept is demonstrated using an STT-MRAM model, which is calibrated using 1Mb characterized array at IMEC and is validated by deploying it in a BNN.

  Chapter 5: This presents the work presented in article [66] that focuses on maximizing the throughput and energy efficiency while performing multi-operand (N)OR and (N)AND operations. This paper proposes a referencing-in-array scheme with a differential voltage-based sensing technique that enables accurate two and multi-operand logic operations for RRAM-based CIM architecture. The scheme makes use of a 2T2R cell configuration to create a complementary bitcell structure that inherently acts also as a reference during the operation execution resulting in a high sensing margin. Moreover, the variation-sensitive multi-operand (N)AND operation is implemented using complementary-input (N)OR operation to further improve its accuracy.

  Chapter 6: This describes the work presented in article [67] that targets to accelerate testing of RRAM devices using CIM-based logic operations. This paper presents low-cost CIM-based design-for-testability (DFT) solutions to expedite the detection and diagnosis of faults by developing logic designs involving multi-row activation. A novel addressing scheme is introduced to facilitate the diagnosis of faults. Reconfigurable logic designs are developed to detect unique RRAM faults that offer features such as programmable reference generations, period, and voltage of operation. DFT implementations are validated on a post-layout extracted platform and testing sequences are introduced by incorporating the proposed DFTs.

  Chapter 7: This presents the work presented in article [68] that targets to accelerate XOR-based encryption applications by performing multi-operand XOR in a single cycle. This paper proposes MOXOR-CIM, a circuit-level mitigation solution for CIM-based multi-operand XOR logic operations; the scheme uses a voltage-to-time converter (VTC) to perform a multi-phased XOR in a single clock cycle. Here, bitline capacitances are utilized for voltage-based sensing for computation which generates the output voltage value being linear to the operand values; the voltage is then converted into desired logic output using the VTC.

  Chapter 8: This provides an outlook on the proposed logic accelerators, presents the progression during the course of the thesis and discusses their comparative impact

on the state-of-the-art while also discusses briefly possible design explorations of this research.

- **PART III: CIM-BASED ARITHMETIC ACCELERATORS**

  Chapter 9: This covers the work presented in article [69] that targets neuromorphic and general-purpose arithmetic applications. A *scalable and reliable integrate and fire circuit* ADC (SRIF-ADC) design for CIM architectures is presented, suitable for stringent power and area constraints. Techniques to stabilize the node receiving analog inputs are implemented that allow more rows to be activated at the same time, thereby improving the scalability in terms of higher parallelism of operations. A self-timed variation-aware design approach is introduced along with design measures to drastically reduce the read disturb of memristor devices. In addition, a compact, built-in sample-and-hold circuit to replace the typically used large-sized capacitance is presented along with a built-in weighting technique to alleviate the need for post-processing when combining outputs of different bit significance.

  Chapter 10: This describes the work presented in article [70] that targets image classification applications. This work presents a memory-periphery co-design to perform accurate A/D conversions of analog matrix-vector-multiplication (MVM) outputs. A novel scheme is introduced where select-lines and bit-lines in the memory are virtually fixed to improve conversion accuracy and aid a ring-oscillator-based A/D conversion, equipped with component sharing and inter-matching of the reference blocks. In addition, we deploy a self-timed technique to further ensure high robustness addressing global design and cycle-to-cycle variations. The concept is demonstrated using a 4Kb CIM chip prototype using resistive bitcells on TSMC 40nm CMOS technology.

  Chapter 11: This describes the work presented in article [71] that targets compact and ultra-low power neuromorphic applications for edge-AI. The ADC design is greatly simplified by making use of complementary data storage whereby differential analog MAC outputs are generated by the crossbar. The differential output voltage generated by the memory crossbar array is converted into the digital domain using a compact dual-ramp VTC-based ADC technique. This work is demonstrated using a chip prototype of 256x8 CIM with resistive bitcells equipped with TSMC 40nm CMOS technology.

  Chapter 12: This provides an outlook on the proposed ADC designs for arithmetic accelerators, presents the progression during the course of the thesis, and discusses their comparative impact on the state-of-the-art while also briefly possible design explorations of this research.

- **PART IV: CONCLUSION**

  Chapter 13: This concludes the thesis by providing an executive summary of the concept, fundamentals, and challenges of memristor-based CIM followed by the proposed solutions for CIM-based logic and arithmetic accelerators. It also covers the key discussion points and the future directions of this research.

2

BACKGROUND

This chapter presents a brief background on memristor-based CIM. First, it presents the fundamentals of conventional computing and how near-memory computing and CIM perform certain computational tasks differently for memory-centric applications. Then, it covers different volatile and non-volatile memory technologies that are been investigated for implementing CIM. Thereafter, it presents the key components of a typical CIM architecture and their impact on overall computational accuracy and efficiency. A comprehensive account of the challenges of memristor-based CIM follows and the opportunities they provide for enabling better performance. Finally, a study on state-of-the-art CIM-based logic and arithmetic accelerators is presented which are classified based on the challenges they attempt to address. The major limitations of these solutions are highlighted which gives motivation to the work that is conducted as part of this thesis.

This chapter is partially based on [1–3].

2.1. CLASSIFICATION OF MEMORY-CENTRIC COMPUTATION

A computing task involves an interaction of data (or operands) where the computing unit defines the location of the data, the intended function, the location where the intended function is carried out, and the location of the final output. Different flavors of computing units can be classified based on *where* the final output is generated and are further classified based on *where* the modification of the components is required to execute the intended function for data-intensive tasks.

2.1.1. COMPUTATIONAL UNITS

- **Computation-outside memory (COM)** refers to units where the execution of an operation is performed *outside* the memory package (marked as circled 4). This typically refers to the traditional way of computing where the data (or operands) is read out of the memory core and the operation is performed using *shared* computational units *i.e.,* logic and arithmetic units. The data readout is generally stored temporarily in register files close to the computational units and following the operation performed by these units, the data is shuffled back into the memory core. Since the memory is always read out before the computing phase, the type of data on which the operation is performed and the final output are always in the digital domain. The data bandwidth is limited by the communication between the computational unit and the memory units. Examples of this category are GPUs and TPUs [72–74].

- **Computation-near memory (CNM)** refers to units where the execution of an operation is performed in the *vicinity* of the memory core, indicated by 'extra logic' (marked as circled 3). This typically refers to basic logic and arithmetic units that do not require any memory storage of their own and directly work on the readout data from

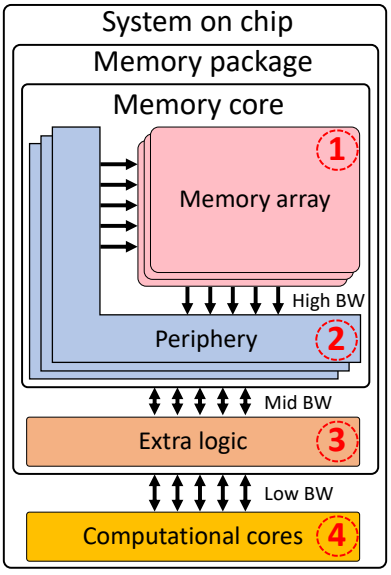


Figure 2.1: CIM core architecture and its classification [75].

the memory unit. These units are *dedicated* to each tile, however, the sequence of operation is similar to traditional computing; data is read sequentially, logic or arithmetic operation is performed and the output is written back into the memory unit or is transferred further to other tiles. Similar to COM, since the memory is always read out before the computing phase, the type of data on which the operation is performed and the final output are always in the digital domain. The data bandwidth is limited by the communication between this functional unit and the memory unit. These architectures, also referred to as digital computation-in-memory units (DCIM), and some of the very recent works are described in [76–78], where the readout operand from the memory interacts with the data available close to the arithmetic units, typically comprising adder tree and shift-and-add units.

Computation-in memory (CIM) refers to units where the computation (i.e., execution) of an operation is performed within the memory core. Depending on *where* the result is generated, CIM can be classified into two classes and are further classified based on *where* major modification of the components is required to execute the intended function i.e., in the memory array or the periphery or a combination of both.

- **Computation-in memory array (CIM-A)** refers to units where the results are generated *within* the memory array (marked as circled 1). This implies that the output resides as the state of a unit cell and is simply read out by a standard memory read operation that determines the binary state of that unit cell. The operands reside in the memory array as the states of the unit cells and the function is executed on them, leading to a modified state of one of the operand unit cells or a dedicated unit cell to store the final output. This can be further classified into two sub-classes; i) basic architectures requiring design changes *only* inside the memory array (CIM-Ab), and ii) hybrid where in addition to *major* changes in the memory array minimal to medium changes are required in the peripheral circuit (CIM-Ah). Examples of this architecture include [79–83]

- **Computation-in memory periphery (CIM-P)** refers to units where the results are generated *in* the periphery of the memory core (marked as circled 2), albeit outside the memory array. All the operands may or may not reside in the memory array, however, at least one of the operands resides in the memory. In the case when one operand is available outside the memory, this operand is provided using a digital-to-analog unit (DAC) to the memory array to interact with the data residing in the memory. Typically, the result is generated within the memory array in the analog domain and is converted to the digital domain in the periphery logic. Sense amplifiers (SA) are used for binary outputs and analog-to-digital converters (ADC) for multi-bit digital outputs. CIM-P can be further classified into two sub-classes; i) basic architectures requiring design changes *only* in the periphery (CIM-Pb), and ii) hybrid where in addition to *major* changes in the periphery minimal to medium changes are required in the memory array (CIM-Ph). Examples of this architecture include [49, 62, 84–86].

In this thesis, we focus on CIM architectures. Within CIM, we briefly cover CIM-A solutions where we clearly highlight the comprehensive advantages of CIM-Ph over the other CIM flavors. Therefore, we describe the rest of the background mostly focusing on CIM-Ph architectures, which are simply denoted as CIM architectures.

2.1.2. MEMORY TECHNOLOGY FOR CIM

VOLATILE AND NON-VOLATILE MEMORY

In a volatile memory, the unit cell or bitcell stores data as a charge value. This requires a constant voltage supply to hold this charge and is lost in the absence of the voltage supply. Typical volatile memories that have been investigated to perform CIM include mature technology nodes such as static random access memory (SRAM), dynamic random access memory (DRAM), and a hybrid of the two. Due to the highly dense DRAM with limited periphery area available, high latency READOUT operation, and the required refresh operation after the READOUT or compute operation, DRAM-based CIM solutions are limited. On the other hand, high-speed and lesser-dense SRAMs allow large components such as SA and ADC blocks and other supporting digital circuits within its periphery logic. In several explorations, SRAM bitcell configurations are either unchanged compared to their traditional memory-only configurations or are changed to improve the efficiency or/and accuracy when realizing CIM-based operations. Therefore, one of the most mature SRAM bitcell configurations such as single-port six-transistor (6T) [36, 39, 40, 87], dual-port eight-transistor (8T) [33, 35, 38], 10T [32] configurations have been explored along with several modifications of the two. On the other hand, one hybrid scheme of combining the elements of SRAM and DRAM bitcells is the use of capacitors within the SRAM bitcells to store the data as a capacitive charge which is capable of storing binary as well as analog values [88, 89].

In a non-volatile memory, the unit cell or bitcell stores data as a conductance value. These type of data storage elements do not require a voltage supply to hold the data and retains it even when the voltage supply is disconnected. Typical non-volatile memories that have been investigated to perform CIM include mature technologies such as NAND FLASH and NOR FLASH [43, 44], and several emerging device technologies such as resistive random access memory (RRAM) [45–54], spin torque transfer magnetic random access memory (STT-MRAM) [55, 56], phase change memory (PCM) [57–60], ferroelectric random access memory (FeRAM) [90–92]. 3D NAND and NOR FLASH memories are also explored in the context of CIM but are limited at present [93].

Following, we give a brief account of three of the most promising emerging memristor device technologies. Traditional memories are not explained in detail as they are well-known.

MEMRISTOR DEVICE TECHNOLOGY

Chua *et al.* first described memristive device [95] as the fourth *missing* element when combining the fundamental physical quantities in electromagnetism *i.e.* current (I), charge (Q), flux (ϕ), voltage (V) and time (t). Figure 2.2 shows the relation between each pair formed from these quantities. Out of the possible six combinations, three elements were already known prior to his work; *i.e.* capacitor, inductor, and resistor, and two are based on well-established Ampere's and Lens' law. The relation between flux (ϕ) and charge (Q) was identified as the missing element as pointed out in Figure 2.2. This implies that this missing element keeps track of not only the amount and direction but also the history of charge flown through that element, as described by Equation 2.1. This missing element was later coined as a memristive device and is interchangeably denoted

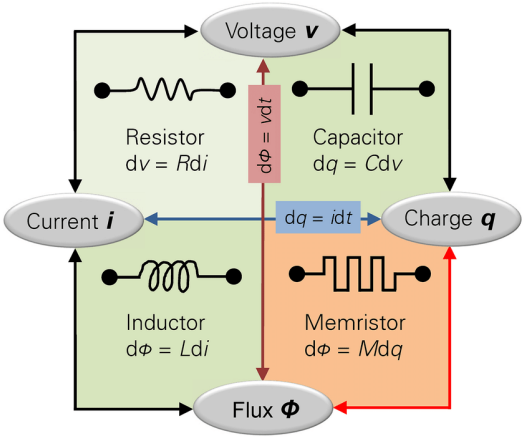


Figure 2.2: Memristor : The missing element [94].

by memristors, memory resistors, and mem-resistors in various scientific articles.

$$\frac{dS}{dt} = f(S, i) \tag{2.1a}$$

$$v(t) = M(S, i) \times i(t) \tag{2.1b}$$

In the above equations, S is the internal state variable, $i(t)$ is the memristive device instantaneous current, $v(t)$ is the memristive device instantaneous voltage, $M(S, i)$ is the memristance, and t is time. The most promising memristor technologies are PCM, STT-MRAM, and ReRAM which are discussed briefly.

- **STT-MRAM** - Spin-transfer-torque magnetic device is a memristive device that switches between stable resistive states based on whether the polarized pair formed by magnetic elements is parallel or anti-parallel [96]. The device is based on magnetic torque switching due to electron spin under the influence of an electric field. Recently, MgO-based spin-transfer (spintronic) device, also known as magnetic tunnel junction (MJT) device, has been proposed in [97] that shows high tunneling magneto-resistance (TMR) ratio *i.e.* easily distinguishable resistance states. The working principle of MgO-based MJT device is demonstrated in Figure 2.3a. MJT consists of 2 dielectric materials, one with fixed polarity and the other with free polarity of magnetic moment. The amplitude, duration, and direction of current applied to write an MJT-based memristor instigate a change in the angular momentum of the electrons, which tends to change the magnetization of the free layer. A change of magnetization in the same direction as the fixed layer allows high current *i.e.* low resistive state (LRS), whereas, a change in the opposite direction allows low current through it *i.e.* high resistive state (HRS).
- **RRAM** - RRAM is a memristive device that switches between stable resistive states based on the deficiency/excess of oxygen ions. TiO_2, HfO_2, MoS_2/MoO_2 [24] are most

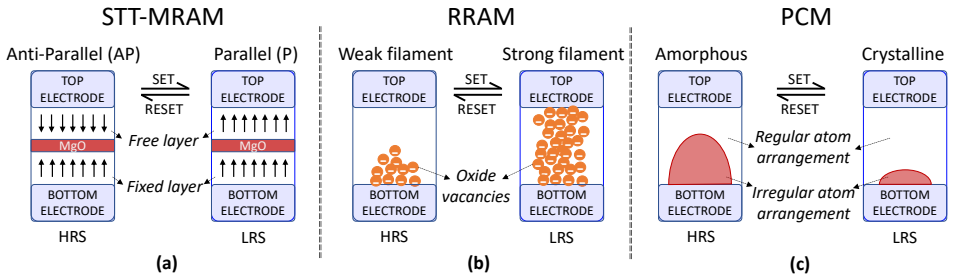


Figure 2.3: Typical mechanism of memristor-devices based on a) STT-MRAM, b) RRAM, and c) PCM.

widely used materials to develop oxide-based memristive devices. The structure of a TiO_2-based memristor is shown in Figure 2.3b. The structure basically comprises top and bottom metal electrodes with TiO_2 as a memristive material. The presence and absence of the valence O^{2-x} ions (which acts as a conductive filament) defines the LRS and HRS, respectively.

- **PCM** - PCM is a memristive device which on the application of joule heating, switches its physical form (or stable resistive state) between crystalline or amorphous [98]. Joule heating required for switching is applied in the form of an electric field or voltage. Some of the recently proposed phase change devices are chalcogenide glass-based GeSbTe, Sb_2Ti_3 AgInSbTe. A SbTe-based phase change device is shown in Figure 2.3c. SbTe switches to amorphous form by heating the filament above its melting point and then rapidly cooling to room temperature. It switches to crystallized form by heating the filament at a specific temperature *i.e.* between its critical temperature and melting point for a fixed period. Crystalline and amorphous form corresponds to LRS and HRS, respectively. It is due to the fact that the crystalline form exhibits a definite path for the flow of electric charge (electrons) whereas the amorphous form disrupts this path.

OVERVIEW OF MEMORY TECHNOLOGIES

This part summarizes the essential attributes that can be expected for a typical memory unit based on traditional as well as emerging device technologies described previously. The focus is primarily on the attributes key for performing computation within CIM and not particularly on the programming of unit cells. Table 2.1 presents the relative attributes. From the top of the table, we first define each attribute followed by the related key take-aways.

- **Feature size (in F^2)**: This is the typical bitcell size with respect to the feature size of the CMOS technology. The most common memristor-based bitcells contain one select transistor and the memristor is generally located between higher metal layers. For instance, the PCM device is located between M4 and M5 as described in [103]. Since

	Traditional Memories			Emerging Memories		
	SRAM	DRAM	Flash	RRAM	STT-MRAM	PCM
Size (F^2)	120-150	6-15	4-10	4-15	4-30	6-50
Non-volatility	No	No	Yes	Yes	Yes	Yes
Read latency	~1 ns	>5 ns	>50 ns	~5-10 ns	~3-5 ns	~5-10 ns
Read Energy	Low	Low	Medium	High	Medium	High
Endurance	~10^16	~10^16	~10^4 - 10^6	~10^7	~10^15	~10^12
Multi-bit	No	No	Yes	Yes	No	Yes
Prog. Variation	Low	Low	Low	high	Low	High
Res. Drift	No	No	No	Yes	No	Yes
Scalability	Medium	Medium	Medium	High	High	High

Table 2.1: Comparison of traditional and emerging memories, to the best of my knowledge and based on the latest technological reports focusing on these devices [99–102].

most of the area is consumed by transistors, emerging device-based bitcells having only one transistor form smaller, denser memory arrays.

- **Non-volatility**: This refers to whether the memory-storing element is volatile or not. For low-power applications, non-volatile devices provide a solid foundation to exhibit near-zero leakage energy during the inactivity of CIM.

- **Read latency**: This is the time spent (typically in n s) per read operation to determine the data stored in the bitcell. This may differ from performing a read operation for a logic operation or an arithmetic operation. This includes latency due to row selection, sufficient sensing margin development, sense amplifier and other supporting periphery logic. Due to the typically higher resistance of memristors, higher RC implies longer read latency.

- **Read energy**: This is the energy spent (typically in f J) per bitcell during a read operation. This includes row selection, row driver, sense amplifier, access line pre-charge, and other supporting periphery logic. Comparatively, higher energy is consumed for memristors due to increased read latency and some may additionally require reference signals for single-ended sensing.

- **Programming variation**: This accounts for the variation of the effective readout resistance of the bitcell following the write operation. This also leads to device-to-device variations. Due to the analog nature of the resistance-based storing mechanism, memristors tend to have high variation in the programmed resistance values. In comparison among memristors, an STT device has a more controlled mechanism due to its magneto-resistive properties where, unlike RRAM and PCM devices, it does not exhibit the creation and distraction of physical bonds.

- **Resistance drift**: This accounts for the resistance drift suffered by a memristor device from its programmed resistance value as a function of time, read voltage, and number and duration of read operations. Due to the analog nature of memristors, they tend to drift towards higher resistance states with time.

- **Endurance**: This refers to the maximum number of programming cycles that can be safely performed before the device is permanently damaged. Memristors have relatively lower endurance due to the nature of the switching mechanism i.e., creation and destruction of states, especially for RRAM devices.
- **Multi-bit**: This refers to whether the memory storing element can store multi-bit data in a single bitcell or not. Due to its analog nature, memristors can store presumably any value between a given range of resistance values. These can be quantized into multi-bits given sufficient resistance margin between each state is respected.
- **Scalability**: This qualitatively describes the extent of geometrical downscaling possible of the memory storing element. Recent developments suggest that these two-terminal memristors can be further downscaled due to their unique electrical properties.

Due to the inherent key advantages of memristors over conventional memory technologies, this thesis work focuses on investigating the potential of memristor-based CIM.

2.2. CIM ARCHITECTURE FOR LOGIC & ARITHMETIC OPERATIONS

This section describes a typical CIM architecture that can perform *in-situ* computation directly on the stored devices. This CIM architecture is illustrated using the most commonly targeted computational tasks for CIM that include bulk bit-wise logic operations such as (N)OR, (N)AND, XOR, and arithmetic operations such as matrix-vector multiplication (MVM). A component-wise description is provided to illustrate how CIM performs these compute operations in the analog domain.

ANALOG COMPUTING

The premise of analog computing lies in the fact that at least one of the inputs and the resulting output are analog in nature. We illustrate the underlining concept of analog

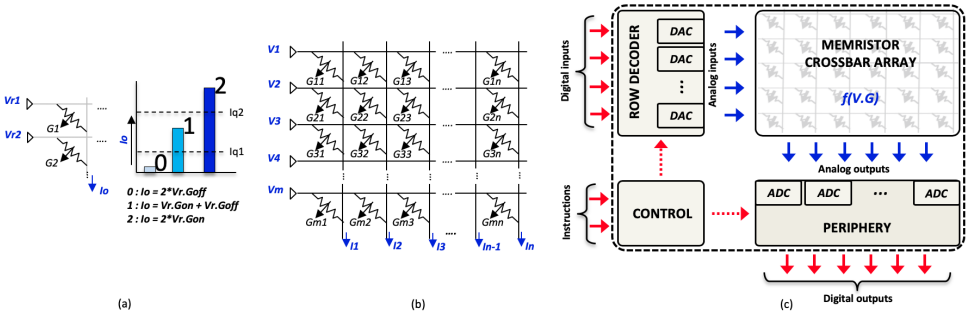


Figure 2.4: (a) Concept of analog-based computing abiding Kirchhoff's law. (b) Memristor-based crossbar array to perform MVM operations with $\mathcal{O}(1)$ time complexity. (c) Memristor-based CIM performing $f(V{\cdot}G)$ within the memory crossbar unit. Digital signals are in red and analog are in blue [69].

computing using Fig. 2.4a to perform a vector-vector multiplication (VVM) operation with each vector consisting of *1-bit* elements.

Considering resistors of conductance values G_1 and G_2 to be either in low (G_{off}) or high (G_{on}) conductance state and sharing a common node, we treat them as one operand vector G of size two. The other operand is the row voltage vector V of size two that is applied such that G_1 and G_2 be arranged in a parallel configuration and the total current flows through the common node. Voltage values of these row vector elements are kept either at V_r or $0\,V$. Total current follows Kirchhoff's law and is given by:

$$I_o = V_{r1} \cdot G_1 + V_{r2} \cdot G_2 \tag{2.2}$$

Permuting all possible combinations of the two operand vectors, we obtain the result of multiple-and-accumulate (MAC) operation with three possible total current values of $V_r(2{\cdot}G_{on})$, $V_r(G_{on}{+}G_{off})$ and $V_r(2{\cdot}G_{off})$. Since the output (current) is still in the analog domain, digitization is performed using ADC of at least *2-bit* or with three quantization levels. Thereby, as shown in Fig. 2.4a, ADC quantifies the current as 0, 1, or 2 as an output. Similarly, by activating two rows containing logic operands with V_r, the output current can be quantified by a 1-bit ADC i.e., SA to determine the logic output.

MEMRISTOR CROSSBAR ARRAY

This computational unit can now be extended to a larger operand vector size and/or more bits per memory element. This implies that ADC needs to support a larger number of quantization levels. Referring to Fig. 2.4b, programmable RRAM devices storing conductance values are arranged in a crossbar array of size $m \times n$ and vector of size m to be applied as row voltage to the crossbar. Here, the memory array is structured in a 1R configuration, implying that one bit-cell consists of one RRAM device as a storage element. However, it can be replaced by any bitcell configuration with the same effective conductance without loss of generality. With respect to the storage operand used in Fig. 2.4a, both the size and number of operands are increased to m and n, respectively. Consequently, operand $V = \{V_1, V_2, ...V_m\}$ is now applied to n different G operands. For simplicity, we illustrate the concept with every element of the operands still represented by *1-bit*. Following Kirchhoff's law to determine current I_j for any column j ($j \in \{1, n\}$), we have n MAC operation outputs at the same time. Therefore, a CIM architecture performs n MAC operations with a time complexity of $\mathcal{O}(1)$. The column current values can be determined using the following:

$$I_1 = V_1 \cdot G_{11} + V_2 \cdot G_{21} .. + V_m \cdot G_{m1}, \tag{2.3a}$$

$$I_2 = V_1 \cdot G_{12} + V_2 \cdot G_{22} .. + V_m \cdot G_{m2}, \tag{2.3b}$$

$$I_n = V_1 \cdot G_{1n} + V_2 \cdot G_{2n} .. + V_m \cdot G_{mn} \tag{2.3c}$$

Digitization of the output currents is done by ADC and other supporting circuits. Here, it is imperative that the actual computation is performed inside the memory crossbar and only analog-to-digital conversions occur in the periphery, hence illustrating the concept of the CIM paradigm. By replacing the vector V with activation signals to select the operands, and ADC with SA, logic operations can be performed using dedicated reference signals based on the type of operation.

CIM ARCHITECTURE

A CIM tile can be inherited from standard well-established memory tiles such as SRAMs and DRAMs, but with some major modifications to accommodate analog-based computing, as shown in Fig. 2.4c. Firstly, for memristor-based CIM, the CMOS-based bitcell comprising the memory unit is replaced by a memristor-based bitcell configured in a compact crossbar array. The circuit blocks comprising the periphery that supports the bitcell array are significantly modified depending on the operations CIM should accommodate. E.g., for MMM operations, the following is needed: 1) Row-decoder becomes complex as CIM involves enabling several rows in a single computation cycle. Also, *1-bit* row or word-line drivers are now replaced by digital-to-analog converters (DACs) that convert multi-bit VMM operands into an array of analog voltages. 2) Column periphery circuits performing read operations (i.e., *1-bit* sense-amplifiers) are now replaced by analog-to-digital converters (ADCs) to quantify currents as digital bit-streams. Post-processing circuits such as shift-and-add are required for MMM 3) The control block needs to deal with complex instructions such as handling intricacies of multi-operand VMM operations as opposed to a simple read or write memory operation.

2.3. CHALLENGES & OPPORTUNITIES IN MEMRISTOR-BASED CIM

This section highlights the key challenges related to dealing with non-idealities and realizing compact and low-power CIM performing logic and arithmetic operations for Big data and edge-AI applications. Fig. 2.5 summarizes these challenges.

2.3.1. ACCURACY CHALLENGES: DEALING WITH NON-IDEALITIES

This includes a list of challenges related to memristor devices and circuit non-idealities which significantly impact the overall accuracy of CIM. It is worth mentioning that these are challenges emerging from using memristor-based bitcell and analog-based computing. In conventional computing, the only analog component or operation is the SRAM read operation where the data is stored in true and complementary form in their stable binary states. This operation involves developing a small sensing margin to differentiate using a SA that senses and amplifies this margin to generate a binary output. Since

Accuracy challenges Dealing with non-idealities		**Efficiency challenges** Achiveing stringent requirements		
Memristor non-idealities	**Circuit non-idealities**	**Energy constraint**	**Latency constraint**	**Area constraint**
Programming error	Wire parasitics	Programming operation	Wire parasitics	Programming related
Resistance ratio	CMOS non-idealities	A/D conversion	Vector/matrix size	Vector/matrix size
Non-zero G_{min}	Design variations	D/A conversion	A/D resolution	A/D resolution
Resistance drift	Sneak path	Wire parasitics	Post-processing	Post-processing
Run-time variations	Random noise			

Figure 2.5: Challenges to develop accurate and efficient memristor-based CIM.

these differential signals are generated using the same bitcell i.e., from the same physical location while utilizing a self-referencing scheme, many of the challenges described below do not appear during conventional computing. In addition, once the SRAM readout binary data is available for computing, the computation is performed in the digital domain with high noise margins by making use of sequential and combinational CMOS logic and arithmetic circuits.

MEMRISTOR NON-IDEALITIES

- **Programming error**: This refers to the deviation of the actual resistance value programmed in the memristor device compared to the nominal value that is intended. Note that the mechanism and the electric properties of programming a memristor device impact the write circuitry and efficiency but we focus on how they impact the CIM tile architecture to perform computing. This directly impacts the readout analog value during the compute operation and thus reduces the sensing margin. For instance, in the case of logic operations, this may lead to an insufficient sensing margin for the SA, where it may fail to reliably generate the correct outcome.

- **Resistance ratio**: This refers to the effective resistance ratio in the readout analog value when reading the HRS and LRS of the memristor device. In a traditional 6T SRAM read, this ratio is the read currents of an OFF and an ON pull-down transistor corresponding to the sides of the bitcell storing 0 and 1, respectively. This ratio is close to 1000. When using memristor devices, this ratio is typically 10-20 for RRAM and PCM devices and less than 2 for STT-MRAM devices. In addition to other circuit non-idealities, this can lead to diminished sensing margins.

- **Non-zero G_{min}**: This refers to the finite conductance exhibited by the HRS of a memristor device. This is a concern while performing MVM operations where an element-wise multiplication of the two vectors produces a finite current when an ideal value of zero current is expected. This leads to a non-linear relation between the ideal MAC value and the actual analog value generated by the memory crossbar array.

- **Resistance drift**: This refers to the drift in the resistance value of the memristor devices with time. In addition, the voltage of operation during read or CIM operation and the number of these operations aggravate this phenomenon. In particular, RRAM and PCM devices involve physical bonds (ionic and covalent bonds in RRAM and PCM, respectively) in realizing a conductive path, these bonds are influenced by the computing operations or/and by time. This can lead to a shift in the analog readout values with time resulting in diminished sensing margins. On the other hand, STT-MRAM is immune to this effect due to its magnetization properties which do not change with time.

- **Run-time variations**: This refers to the variations that occur while performing CIM operations. For instance, temperature and voltage fluctuations affect the effective resistance of the memristor devices and can lead to reduced sensing margins in the worst-case corners.

CIRCUIT NON-IDEALITIES

- **Wire parasitics**: This refers to the finite resistance and capacitance of the access wires inside the memory array. While capacitance affects the transient behavior of the analog signals, resistance affects both the transient behavior and the magnitude of the analog signals due to IR drop. These effects can be seen in both input (select or activation) signals and the resulting output CIM output signals. This can lead to varying strength and activation time of pass transistors and varying degrees of degrading effects on analog signals produced by bitcell at different locations inside the memory array. This is an input dependent effect where depending on the input conditions of vectors (both activation and resistances mapped on the crossbar) this can have varying degrees of IR drop effects.
- **CMOS non-idealities**: This refers to traditional CMOS process variations that can affect the readout analog signal and the sensing circuits. While sensing circuits such as SA, reference signals, and ADC are traditionally well-known and can be accounted for with calibration schemes, additional varying effects coming from the memory array can lead to reduced sensing margins.
- **Design variations**: This refers to variations related to global variations such as temperature and voltage fluctuations that can lead to the varying strength of CMOS devices and temperature-dependent resistances of the access wires resulting in varying IR drop values. This also encompasses cycle-to-cycle variations in enable and clock signals where a change in the duration of the active compute cycle impacts the analog readout value.
- **Sneak path**: This refers to an unwanted leakage path formed via half-select memristors storing the LRS in a crossbar array configuration. Due to finite wire resistance, IR drop can cause an unwanted voltage difference across these devices which can then interfere with the currents flowing in the intended access line. This implies a HRS value can be incorrectly read out with enough LRS devices in the vicinity of the selected HRS device, leading to erroneous results.
- **Noise**: This refers to any random noise that can impact the analog readout value. While read and two-operand logic operations may have sufficient noise margin, noise can impact the small sensing margins involved when performing MVM operations.

To summarize, all the above non-idealities can lead to diminished sensing margins that impact the response of SA and ADC to determine the correct outcome.

2.3.2. EFFICIENCY CHALLENGES: ACHIEVING STRINGENT REQUIREMENTS

With the increasing complexity of algorithms, the amount of data to be processed, and the fact that most of them target compact, low-power hardware implementation for computing at the edge, achieving the required efficiency metrics for memristor-based CIM can be challenging. Below is a summary of the challenges in relation to achieving high energy efficiency, performance, and area efficiency while dealing with the non-idealities described above.

ENERGY CONSTRAINTS

- **Programming operation**: This refers to the programming of memristor devices during a CIM operation which typically involves high voltage/current values and high latency. This is mostly related to logic operations that involve the programming of memristors during the logic operations. It also affects the endurance of the memristor devices which is typically low compared to traditional memories.

- **Voltage of operation**: This refers to the trade-off between having high sensing margins or dynamic range while performing analog computing and the low energy consumption during the compute operation. A higher read voltage increases the dynamic range of voltage that allows larger sensing margins and a higher periphery voltage increases the headroom for transistors in sensing circuits leading to higher noise margins. However, as dynamic power is described by CV^2, an increase in voltage increases the energy quadratically.

- **A/D conversion**: This refers to the energy consumption of converting analog output generated by the memory array into digital domain during CIM operations. SAs and ADCs are traditionally one of the most expensive components in terms of energy, especially in the case of CIM, where most of the computation happens in the low-power analog domain.

- **D/A conversion**: This refers to the energy consumption of converting digital inputs or address selections into analog domain during CIM operations. While this consumption is low compared to consumption during A/D conversions, it can impact the overall consumption and more so when multi-bit inputs and large arrays need to be supported. In addition, an increased D/A bit resolution exponentially increases the energy consumption of typical DACs.

- **Wire parasitics**: This refers to energy spent when transiting enable signals through long parasitic wires across the bitcell array, providing stable access line conditions during CIM operations, and generating the output signals on highly capacitive wires. This leads to high current requirements for a longer period of time which are drawn by stabilizing components in ADCs such as operational amplifiers (Opamps) in current sensing schemes or by pre-charge circuits in voltage sensing schemes.

LATENCY CONSTRAINTS

- **High parasitics**: This refers to longer time spent on pre-charging or stabilizing the access lines, settling times of the activation, and enable signals due to large wire parasitics in CIM units.

- **Vector/Matrix size**: This refers to the dependency of latency of performing CIM operations with the size of the memory array. Naturally, with different signals traveling along a larger number of rows and columns, the RC delay increases the overall latency of the CIM operation.

- **A/D resolution**: This refers to the impact of A/D resolution requirements in MVM operations on the overall latency. Typical current and time-based ADC techniques suffer

from exponential latency increases with an increase in required bit resolution. While conversion latency in other ADC topologies such as SAR-ADC increases linearly, the Opamps/SA required are typically more complex.

- **Post-processing**: This refers to the latency of any post-processing required after the A/D conversion related to affine scaling and offset corrections, performing activation functions such as ReLu, tanh in neural network implementations, etc., accumulation of logic results in a multi-operand logic operation. Latency impact may be relatively low if dedicated post-processing units are utilized which may not be possible due to limited area.

AREA CONSTRAINTS

- **Programming related**: This refers to the area required due to the programming of memristor devices. Typical programming voltage and current required for switching memristor states is high, voltage is the order of 2V for RRAM devices and 3-4V for PCM devices, and currents close to 400-600 μA. These requirements not only need special large thick-oxide write driving transistors but also require a large-sized pass transistor within the bitcell. Where traditional memory units such as SRAM bitcells usually utilize minimum-sized transistors supported by the technology node, the pass transistors in memristor-based bitcell can be close to 4-5× the size of the minimum transistor size.

- **Vector/Matrix size**: This refers to the size of the overall area of the SoC consisting of several CIM tiles. Most of the data-intensive applications work on large datasets and there is a natural trade-off between increasing the capacity of one CIM tile and the number of CIM tiles. A larger CIM tile reduces the number of tiles and hence the overall area, however, it can be challenging to accurately perform analog computing within a large array.

- **A/D resolution**: This refers to the area required by the A/D conversion and related circuits such as reference blocks, calibration, and compensation circuits. One of the major challenges is not only the area of these analog components within the ADC perse, and the area of registers storing different possible configurations but also the fact that they all need to fit within a small column pitch (defined by bitcell pitch) in order to avail $\mathcal{O}(1)$ time complexity of performing CIM operations. This introduces additional challenges related to long and thin ADC designs where long parasitic wires can also impact conversion accuracy.

- **Post-processing**: This refers to the area occupied by post-processing units. These include registers and digital compute blocks such as adders and multipliers involved in performing the required post-processing described earlier, which can be challenging as they also need to fit within small column pitch(es).

To summarize the efficiency challenges, efficient design and implementation of A/D conversion can be challenging as it needs to deal with aforementioned non-idealities within constrained energy, latency, and area requirements.

2.4. STATE-OF-THE-ART & ITS LIMITATIONS

This section summarizes the state-of-the-art solutions to perform logic and arithmetic operations in memristor-based CIM and provides key limitations faced by them. First, the solutions are grouped based on the type of operation performed and further sub-grouped into challenges that are addressed and the related improvements they have compared to their respective prior arts. This is followed by an account of the limitations they pose to the overall efficiency and accuracy of these operations, therefore, motivating the work conducted as part of this thesis.

2.4.1. LOGIC ACCELERATORS

Several works on CIM-based logic operations have been reported. Such works can be classified into two classes: 1) stateful logic (CIM-A)- where the memristor state is altered in order to perform logic operations. 2) non-stateful or read-assisted logic (CIM-P)- where the memristor state is unaltered.

STATEFUL LOGIC

Memristive stateful logic gates have all input(s) and output(s) within the crossbar array and are represented as resistance values. Primitive logic designs discussed under this category are material implication logic (IMPLY), Snider, fast boolean logic (FBL), and memristive aided logic (MAGIC), and majority (MAJ).

- **IMPLY**: Borghetti *et al.* proposed material implication logic (IMPLY) [104] which performs p IMPLIES q or $p \rightarrow q$ and is defined as "if not p then q". The structure as shown in Figure 2.6a, consists of two memristors, namely P and Q, with applied supply voltages (stored resistance states) as $V_p(p)$ and $V_q(q)$, respectively. A resistor R_g connects the common node (joining their top electrodes) of the two memristors to the ground. IMPLY is evaluated by applying suitable voltages V_p=V_{cond} and V_q=V_{set} while connecting floating node x through a resistor R_g ($Ron << R_g << Roff$) to ground. Correct functionality requires $V_{cond} < |NVth|$ and $V_{set} = 2 \times V_{cond} > |NVth|$. The evaluated result is stored in Q; therefore, this gate configuration is destructive. The working principle is based on a voltage divider topology where the state of the input decides the amount of voltage it receives, and below or above the write threshold voltage implies that the memristor will be written or it holds its state. This is extended to a two-cycle operation to perform NAND/NOR operations. Here, a third memristor S

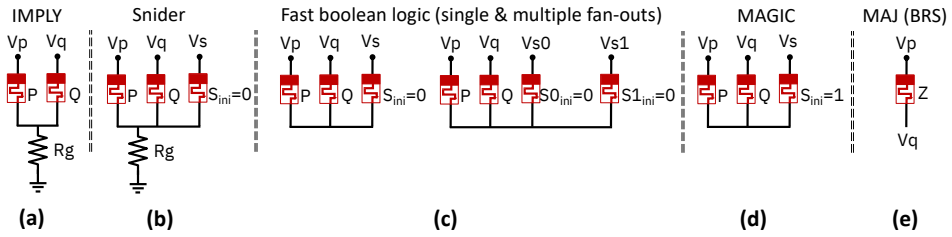


Figure 2.6: Stateful logic designs.

is used to store the final output which is initiated to the HRS state at the beginning of the cycle.

- **SNIDER**: Snider *et al.* proposed Snider logic designs [79] which can perform NOT, NAND and COPY (buffer) primitive functions. The structure as shown in Figure 2.6b and the working principle of the snider gates are similar to IMPLY gates, the difference is the way the voltages are applied to perform different operations. Note that Snider COPY does not use the resistor Rg. Unlike IMPLY, NAND/NOR logic can be performed in a single cycle by applying V_{cond} in the same cycle, thus achieving better performance.

- **Fast boolean logic**: Xie *et al.* proposed a family of FBL-based logic designs [82] that can perform AND logic and multiple fan-outs for COPY and NAND logics. In short, single fan-out snider COPY and NAND logic designs are extended to form double fan-out (DFO) FBL-based COPY and NAND gates by utilizing one/more additional memristor for providing identical second/more output. The initial conditions, working conditions are similar to that of Snider logic as shown in Figure 2.6c.

- **MAGIC**: Shahar *et al.* proposed a family of MAGIC logic logic designs [105] that can perform all the primitive logic functions, namely NOT, AND, OR, NOR, and NAND. However, MAGIC-based AND, OR, and NAND logic gate configurations does not support crossbar structure and hence are not discussed here. The structure as shown in Figure 2.6d and the working conditions are similar to that of Snider logic designs with major differences in the working principle; i) Rg is not used, ii) the given operations can be performed in a single cycle, but the voltage of operation is always V_W, and iii) the output memristor is initialized to the LRS instead of HRS. This work can also be extended to multi-operand NOR/NAND operations simply by having more memristors storing inputs and following the same principle.

- **MAJ**: Gaillardon *et al.* proposed resistive switching logic design based on two structures *i.e.* binary resistive switching (BRS) and complementary resistive switching (CRS) implement the majority (MAJ) logic function [83]. BRS logic design is prone to sneak path issues but can be solved by either using a rectifying memristor or using CRS logic design. BRS logic design that can perform MAJ logic while performing primitive logic functions such as NOT, AND, OR, NOR, NAND, IMPLY etc. in multiple cycles. As shown in Figure 2.6e, it consists of a single memristor Z, storing one of the inputs (z) as $R_{on}(1)$ or $R_{off}(0)$ resistive value, while the other inputs (p and q) determine the voltage applied as $V_{in}(1)$ or $0(0)$ across the top and bottom electrode (V_p and V_q, respectively). MAJ is a *3-input* (binary) logic function, also known as a median operator, whose output is the same as the value of input occurring majority of the times (here, ≥ 2). The output (resistance) is represented as ($z_{new} =$) $MAJ(p, \bar{q}, z) = p\bar{q} + \bar{q}z + zp$ in the memristor Z.

  The key challenges addressed by the stateful logic are (as highlighted in Fig. 2.7);

  - **Non-idealities**: Challenges related to memristor non-idealities are partially addressed. Non-zero G_{min} issue is alleviated as there is no multi-row read operation

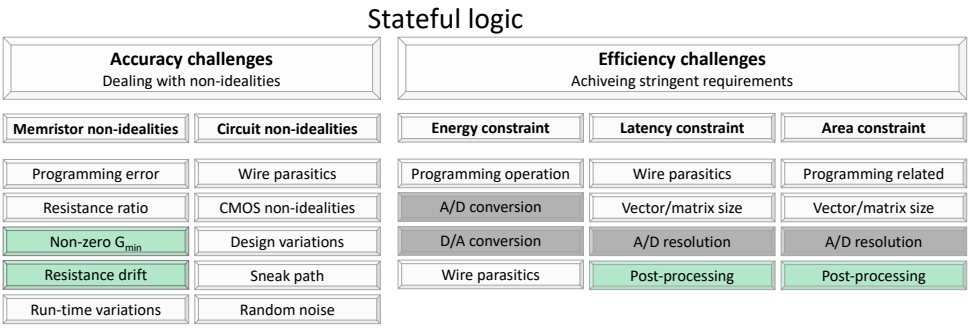


Figure 2.7: Challenges addressed and not addressed by stateful logic designs.

where G_{min} does not interact with G_{max} during the programming of memristors. Resistance drift is also not essential as the programming is part of the logic operation. Challenges related to circuit non-idealities are not addressed.

- **Efficiency constraints**: Challenges related to latency and area are partially addressed. The impact of post-processing is alleviated as there is no intermediate result in the periphery. This is because all the required sequences of operations are performed within the bitcell memory units.

However, the following challenges remain unaddressed by the state-of-the-art stateful-logic solutions (as highlighted in Fig. 2.7);

- The existing stateful logic units are in-efficient for a number of reasons: i) expensive programming of memristor devices is not only slow and power-in-efficient but also prone to noise, variations, and lower resistance ratio, ii) the coordination of configuring different high voltage values on the access lines, iii) the sequential way of operating further aggravates the aforementioned disadvantages, iv) challenges CMOS non-idealities, sneak path, and wire parasitic are not addressed, v) although the readout is a simple memory read operation with a single reference signal required via a SA, this additional SA energy further degrades the efficiency and vi) the limited endurance of the memristors is aggravated since multiple programming operations are involved.

NON-STATEFUL LOGIC

Memristive non-stateful or read-assisted logic designs have their inputs located within the crossbar array as resistive values while their output is located (or realized) in the periphery circuit such as a SA. Logic designs in this category are scouting, Pinatubo, and dual-SA.

- **Scouting logic**: Xie *et al.* [106] proposed scouting logic design [84] to perform primitive logic functions such as OR, AND and XOR but can be easily extended to NOR, NAND, NOT and XNOR with suitable modifications. Scouting can be illustrated in Fig 2.8(a). The underlying concepts for read-assist logic designs are: 1) The interaction of voltage

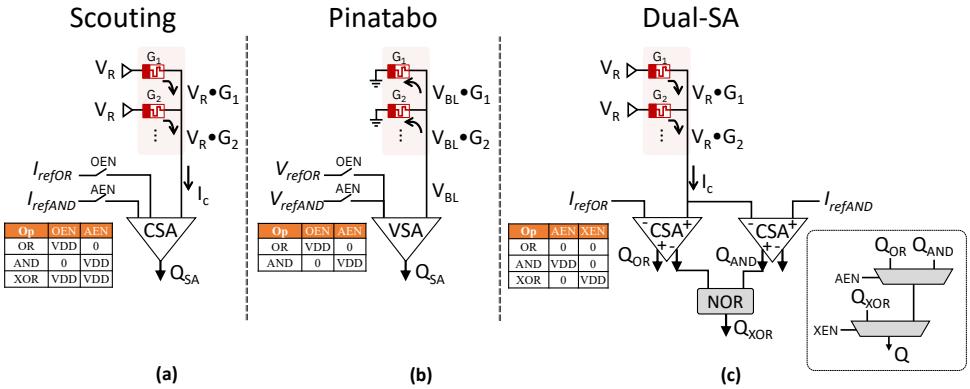


Figure 2.8: Non-stateful logic designs.

V_R and bitcell conductance G, in accordance with Ohm's law, resulting in the current $V_R \times G$ per bitcell; 2) Accumulation of these currents into a column current I_c, in accordance with Kirchhoff's current law; and 3) comparison of the current or voltage drop with an appropriate reference (selected by a MUX) using a customized SA. Therefore, logic operations can be performed simultaneously in all the activated columns and practically operate at O(1) time complexity, thereby achieving massive parallelism.

- **PINATABO logic**: Li *et al.* [107] proposed a similar approach to the scouting logic and uses a voltage-based SA. To perform XOR logic, where OR and AND are cascaded in two successive cycles. The reference is generated using memristors to address global memristor device-related variations.

- **Dual-SA logic**: Jain *et al.* [108] proposed a similar approach to the scouting logic, however, it includes current-based sensing and utilizes current-based SAs. In addition, it duplicates the SAs with dedicated references for OR and AND operations. Similar to PINATABO, the reference is generated using memristors to address global memristor device-related variations.

The key challenges addressed by the non-stateful logic are (as highlighted in Fig. 2.9);

- **Non-idealities**: Challenges related to memristor non-idealities are partially addressed. Run-time variations related to memristors can be captured in memristor-based reference generators, which can adapt to global variations such as temperature and voltage. Programming errors in memristors are also partially addressed as the reference signals are kept with a sufficient margin that also includes these errors. Challenges related to circuit non-idealities are not addressed.

- **Efficiency constraints**: Challenges related to energy and latency are partially addressed. Expensive programming operation is completely avoided. The fact that these logic operations do not involve the programming of memristor devices, the

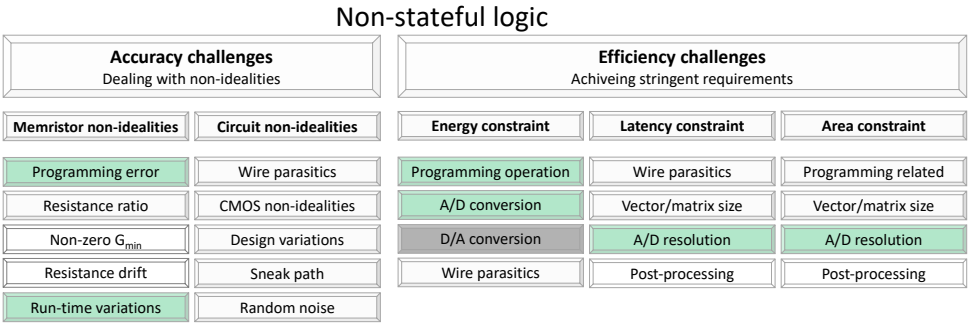


Figure 2.9: Challenges addressed and not addressed by non-stateful logic designs.

energy, and latency spent on multi-row read operations are typically low. In addition, the voltage of operation is typically much lower compared to the write voltage, thus reducing the energy consumption.

However, the following challenges remain unaddressed by the state-of-the-art non-stateful logic solutions (as highlighted in Fig. 2.9);

- The existing non-stateful logic units suffer from the presence of variations that affect the overall performance and computational accuracy; i) the conventional approach is based on a single-sided sensing mechanism, implying that array current/voltage output is compared against a reference that offers small sensing margins for computation [64, 106–110], ii) the inherent small margin associated with (N)AND operation as compared to (N)OR operation further limits the scalability in terms of operand size [64, 108], iii) the crossbar size scalability is limited due to the influence of wire parasitics that further degrades the performance and the accuracy of the operations, iv) a memristor device also suffers from the accumulated effect of a large number of read operations that can lead to significant conductance change (conductance drift) or unwanted bit-flip (read disturb) [111, 112], v) They include large and complex SAs and reference generators; however they degrade the overall efficiency of the computing [49, 64, 106–110, 113–115], and vi) supporting multi-operand operations further aggravates the above aforementioned challenges [49, 110]. Multi-operand can alternatively be performed by investing in aggregating multiple two-operand (N)AND or (N)OR operating cycles effectively in the periphery [64]. This, however, requires complex periphery circuits implying additional area and power consumption and reduced computational throughput.

2.4.2. ARITHMETIC ACCELERATORS

Several works on CIM-based arithmetic operations using memristor devices have been reported mostly focusing on performing MVM operations for deep neural network applications. The two key primary areas are bitcell configuration and the ADC design. Such works can be classified into two classes: 1) voltage-based conversion and 2) current-based conversion. These solutions have mostly utilized 1T1R bitcells and 2T2R bitcells

and the way these bitcell structures have been configured is described within each of the solutions.

VOLTAGE-BASED CONVERSION

Voltage-based conversion typically involves CIM generating an output voltage on the access lines *ideally* proportional to the MAC result or using a current-to-voltage converter following a current-based sensing scheme. This refers to the access lines starting from a discharged or pre-charged value and during the MVM operation of fixed length, the voltage on this access line changes in relation to the effective MAC value. Following are the state-of-the-art solutions that follow this approach.

VOLTAGE-BASED CONVERSION: SUCCESSIVE-APPROXIMATE REGISTER ADC (SAR-ADC)

SAR-ADC involves a successive comparison of analog values in a binary-search manner and an adaptive reference set of references generated based on the decision of the preceding comparison. It comprises four sub-blocks as shown in Fig. 2.10a; i) sample-and-hold (S/H): typically a switch that samples the input voltage value and a capacitor to hold the sampled value for further processing ii) comparator: typically a voltage-based SA that compares sampled input voltage value with configurable reference signals and results in a stream of binary outputs in a time-multiplex manner, iii) DAC: typically a capacitive DAC (CDAC) is involved which generates the required reference signal based on the previous binary output, and iv) register: typically a group of flip-flops to store the results of the intermediate binary output values. SAR-ADC utilized in CIM can be found in [51, 116–119].

The key challenges are addressed, as highlighted in Fig. 2.11;

- **Non-idealities**: Challenges related to non-idealities are partially addressed. Re-

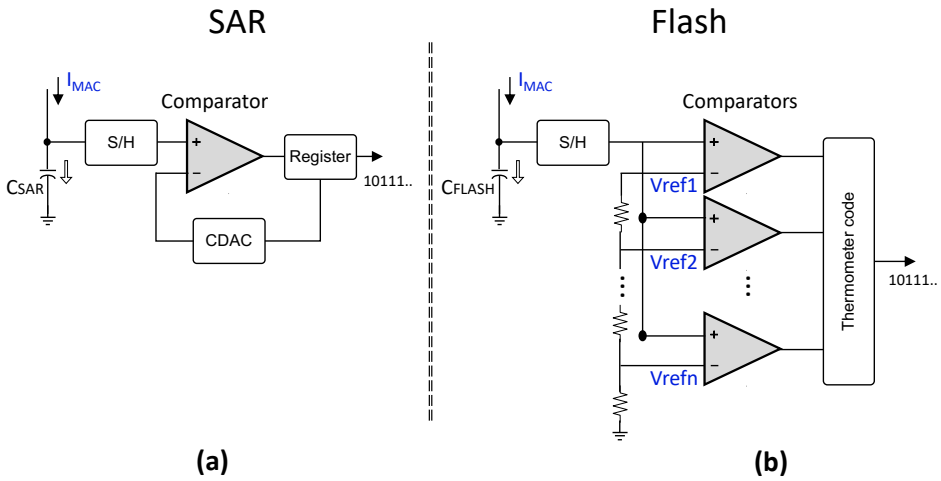


Figure 2.10: Voltage-based A/D conversion based on (a) SAR and (b) Flash.

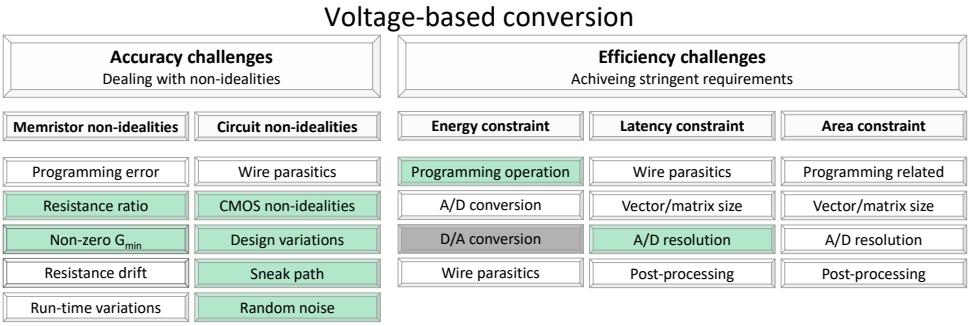


Figure 2.11: Challenges addressed and not addressed by voltage-based conversion schemes.

sistance ratio and non-zero G_{min} are partially addressed by using SAR-ADC with gain and offset calibrations, respectively, along with CMOS non-idealities, noise, and variations. Sneak-path is addressed using selector devices i.e., 1T1R bitcells.

- **Efficiency constraints**: Challenges related to energy and latency are partially addressed. Expensive programming operation is completely avoided. SAR-ADC typically scales well with increasing resolution in terms of latency, a linear increase with every bit of resolution. Post-processing is limited as the digital output is the final ADC output.

The following challenges remain unaddressed, as highlighted in Fig. 2.11;

- **Non-idealities**: Programming error, resistance drift, and run-time variations are not addressed since they lack memristor-aware solutions. Wire parasitic is also not addressed as they lack any global compensation scheme.

- **Efficiency constraints**: Voltage of operation is not really reduced to have a larger dynamic range for voltage-based sensing and the A/D conversion is typically expensive in terms of area and energy due to large CDACs and comparator.

VOLTAGE-BASED CONVERSION: FLASH ADC

Flash ADC involves multiple comparisons of an analog value with all possible set of references in a single cycle by utilizing several SAs as shown in Fig. 2.10b. The number of SAs is typically equal to the number of possible levels of the ADC output i.e., for an n-bit ADC resolution, 2^n-1 SAs are required. Thereafter, the thermometer decoder generates the final output. Flash ADC utilized in CIM can be found in [120–122].

The key challenges are addressed, as highlighted in Fig. 2.11;

- **Non-idealities**: Challenges related to non-idealities are partially addressed. Resistance ratio and non-zero G_{min} are partially addressed by using Flash-ADC with gain and offset calibrations, respectively, along with CMOS non-idealities, noise, and variations. Sneak-path is addressed using selector devices i.e., 1T1R bitcells.

- **Efficiency constraints**: Challenges related to energy and latency are partially addressed. Expensive programming operation is completely avoided. Flash ADC typically does not scale with increasing resolution in terms of latency, the number of comparisons per cycle is increased exponentially with increase with every bit of resolution.

The following challenges remain unaddressed, as highlighted in Fig. 2.11;

- **Non-idealities**: Programming error, resistance drift, and run-time variations are not addressed since they lack memristor-aware solutions. Wire parasitic is also not addressed as they lack any global compensation scheme.

- **Efficiency constraints**: Voltage of operation is not really reduced to have a larger dynamic range for voltage-based sensing and the A/D conversion is typically expensive in terms of area and energy due to a large number of SAs and reference blocks. Post-processing is needed as the output of Flash ADC is thermometer code which needs to be decoded to obtain the final ADC output.

VOLTAGE-BASED CONVERSION: MULTI-PHASED SA

Multi-phase SA can be considered as a hybrid solution utilizing parts of SAR-ADC and Flash ADC components as shown in Fig. 2.12. In this scheme, while some of the most significant bits (MSBs) are determined using the SAR-ADC approach, the rest of the least significant bits (LSBs) are determined using the Flash ADC approach. These types of ADC utilized in CIM can be found in [56, 60, 123, 124].

The key challenges are addressed, as highlighted in Fig. 2.11;

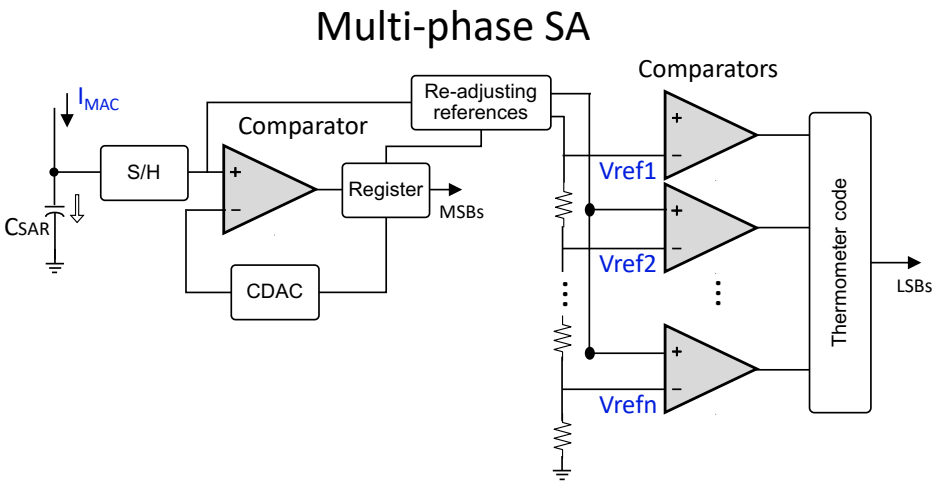


Figure 2.12: Voltage-based conversion with multi-phase SA: A/D conversion by combining SAR and Flash ADC topologies.

- **Non-idealities**: Challenges related to non-idealities are partially addressed. Resistance ratio and non-zero G_{min} are partially addressed by SAs with gain and offset calibrations, respectively, along with CMOS non-idealities, noise, and variations. Sneak-path is addressed using selector devices i.e., 1T1R bitcells. Programming error is accounted for with sufficient sensing margins, where the dynamic range when determining LSBs is reduced with the help of SAR-based MSB determination. Wire parasitic-related challenges are partially resolved using global compensation schemes.
- **Efficiency constraints**: Challenges related to energy and latency are partially addressed. Expensive programming operation is completely avoided. The hybrid approach scales semi-linearly in terms of latency, because of the advantages of using Flash ADC with increasing resolution. Challenges related to vector/matrix size are partially resolved by performing multi-phased SA comparisons.

The following challenges remain unaddressed, as highlighted in Fig. 2.11;

- **Non-idealities**: Memristor and circuit non-idealities are partially unaddressed. Wire parasitic-related challenges along the columns are not fully addressed as most of the solutions use a small number of rows, which may have scalability issues with the impact of IR drop.
- **Efficiency constraints**: Voltage of operation is not quite reduced to have a larger dynamic range for voltage-based sensing and the A/D conversion is typically expensive in terms of area and energy due to a large number of SAs and reference blocks. Post-processing is needed as the output of hybrid ADC is partially a thermometer code that needs to be decoded to obtain the final ADC output in combination with the SAR-based MSBs.

CURRENT-BASED CONVERSION

Current-based conversion typically involves CIM generating an output current on the access lines *ideally* proportional to the MAC result. This refers to the access lines maintaining a fixed voltage value across the access lines throughout the MVM operation of fixed length, following a stabilized current on this access line in relation to the effective MAC value. Following are the state-of-the-art solutions that follow this approach.

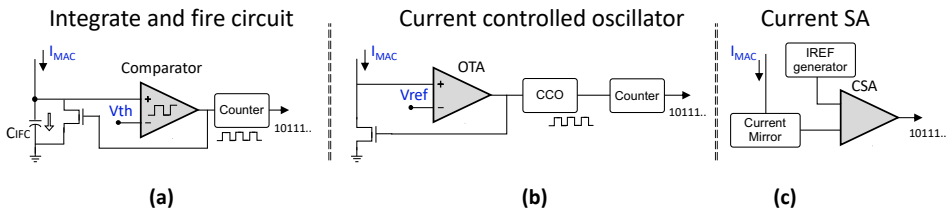


Figure 2.13: Current-based sensing and A/D conversion using (a) current SA, (b) integrate-and-fire circuit and (c) current-controlled oscillator.

CURRENT-BASED CONVERSION: INTEGRATE-AND-FIRE (IFC-ADC)

IFC-based ADC involves integrating the current to develop a voltage over a capacitor firing a spike when this voltage crosses a certain pre-defined threshold and resetting the capacitor at the same time. Here, the number of fired spikes is proportional to the current as the more the current is, the more frequent the spike generation. IFC-ADC comprises three blocks as shown in Fig. 2.13a; i) sampling unit: typically a capacitor (smaller than the sample-and-hold capacitor in SAR) that charges with the incoming current, ii) voltage crossing detector: typically a comparator where one input is fixed with the voltage threshold and other is connected to the sampling capacitor, and iii) spike generator and reset: typically an inverter/buffer and a simple combinational circuit that resets the capacitor to able to repeat the charging process. These types of ADC utilized in CIM can be found in [47, 125, 126].

The key challenges are addressed, as highlighted in Fig. 2.14;

- **Non-idealities**: Challenges related to non-idealities are partially addressed. With gain and offset calibrations of the comparator, respectively, CMOS non-idealities are partially addressed. Sneak-path is addressed using selector devices i.e., 1T1R bitcells.
- **Efficiency constraints**: Challenges related to energy and area are partially addressed. Expensive programming operation is completely avoided. IFC-ADC typically scales well with increasing resolution in terms of area, with a small increase with every bit of resolution. Energy is mostly consumed by the OTA to fix the voltage of the access lines which is low compared to the comparators in SAR and Flash ADC, and the comparator to compare the capacitor voltage with the reference. Post-processing is limited as the digital output is the final ADC output.

The following challenges remain unaddressed, as highlighted in Fig. 2.14;

- **Non-idealities**: Programming error, resistance drift, and run-time variations are not addressed since they lack memristor-aware solutions. Wire parasitic related challenges are also not addressed as they lack any global compensation scheme. Resistance ratio and non-zero G_{min} are not reduced with any dedicated scheme.

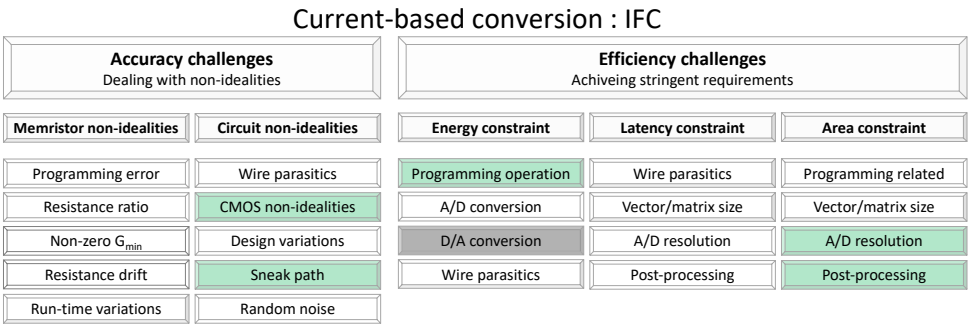


Figure 2.14: Challenges addressed and not addressed by current-based conversion schemes using IFC-ADCs.

- **Efficiency constraints**: Voltage of operation is not reduced to have a larger dynamic range on the capacitor charging value and the A/D conversion is typically expensive in terms of area and energy due to large hold capacitor, OTA, and comparator. Vector/matrix size is not quite as scalable as large high-performing OTAs are needed. On the other hand, OTA can be avoided altogether [125] while utilizing a voltage-based sensing scheme to reduce the area and energy consumption, however, it heavily impacts the linearity of the ADCs.

CURRENT-BASED CONVERSION: CURRENT-CONTROLLED OSCILLATORS (CCO-ADC)

CCO-based ADC involves the generation of oscillations on an oscillator with the oscillation frequency proportional to the current flowing into the oscillator. The MAC output current is proportionally mapped or mirrored onto the oscillator, which becomes the operating current for the oscillator and the number of pulses generated in a given fixed duration is the final digital output. CCO-ADC deployed in a CIM mainly comprises three components as shown in Fig. 2.13b; i) operational-transconductance amplifier (OTA): typically an amplifier that fixes the voltage on the access line(s) to establish a stable and proportional current value on these access lines and generates proportional output voltage ii) CCO: typically a ring of delay elements (inverters) that are biased by the mirrored voltage of the OTA output voltage to produce proportional oscillations, iii) counter: typically a ripple carry counter that counts the number of generated pulses. These types of ADC for CIM can be found in [62].

The key challenges are addressed, as highlighted in Fig. 2.15;

- **Non-idealities**: Challenges related to non-idealities are partially addressed. Resistance ratio and non-zero G_{min} are partially addressed by using OTA with gain and offset calibrations, respectively, along with CMOS non-idealities, noise, and variations. Sneak-path is addressed using selector devices i.e., 1T1R bitcells. Programming error is accounted for with sufficient sensing margins with longer integration times of the oscillator. Wire parasitic-related challenges are partially resolved using global compensation schemes. It also includes global adaptive drift compensation schemes to address the resistance drift.

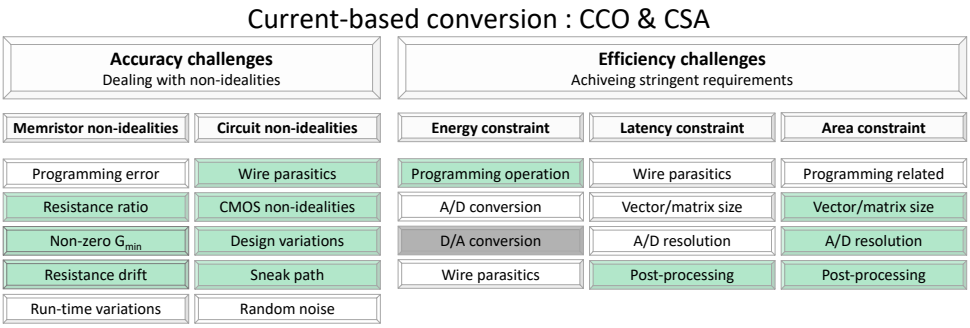


Figure 2.15: Challenges addressed and not addressed by current-based conversion schemes using CCO-ADCs and CSA-ADCs.

- **Efficiency constraints**: Challenges related to energy and area are partially addressed. Expensive programming operation is completely avoided. CCO-ADC typically scales well with increasing resolution in terms of area, with a small increase with every bit of resolution. Energy is mostly consumed by the OTA to fix the voltage of the access lines which is low compared to the comparators in SAR and Flash ADC. The voltage of operation is low across the memristors which reduces the energy spent on analog computing within the array. Vector/matrix size is quite scalable by investing in high-performing OTAs. Post-processing is limited as the digital output is the final ADC output.

The following challenges remain unaddressed, as highlighted in Fig. 2.15;

- **Non-idealities**: Memristor and circuit non-idealities are sufficiently addressed with programming noise mostly dominating the MVM error.
- **Efficiency constraints**: Although they are less expensive with respect to SAR and Flash ADCs, A/D conversion is typically expensive in terms of latency and energy as the number of counts exponentially increases with bit resolution. The required settling time and energy of the OTA to settle access lines increases with the size of the array and the latency can be reduced but with large sized OTAs.

CURRENT-BASED CONVERSION: CURRENT SA (CSA)

CSA-based ADC involves utilizing the current input from the crossbar and using several current references to quantize it into digital outputs. The I_{MAC} current can be directly used or mirrored version of this current can be used as the current input to the CSA, while the current references I_{REF} require a dedicated reference block. CSA-ADC comprises four blocks as shown in Fig. 2.13c; i) OTA: similar to CCO-ADC approach but can be optional, typically an amplifier that fixes the voltage on the access line(s) to establish a stable and proportional current value on these access lines and generates proportional output voltage, ii) optional current mirroring circuits to reduce the current biasing of the CSA, and iii) CSA: current-based SA to compare the input current values with the generated current reference signals to produce digital outputs, and iv) register to store the final outputs. These types of ADC utilized in CIM can be found in [48, 127, 128].

The key challenges are addressed, as highlighted in Fig. 2.15;

- **Non-idealities**: Challenges related to non-idealities are partially addressed. Resistance ratio and non-zero G_{min} are partially addressed by using OTA and dedicated reference signals with gain and offset calibrations, respectively, along with CMOS non-idealities, noise, and variations. Sneak-path is addressed using selector devices i.e., 1T1R bitcells. Programming error is accounted for with sufficient sensing margins by using adaptive reference signals. Wire parasitic-related challenges are partially resolved using global compensation schemes. It may also include global adaptive drift compensation schemes to address the resistance drift.
- **Efficiency constraints**: Challenges related to energy and area are partially addressed. Expensive programming operation is completely avoided. The reference

blocks required typically scales with increasing resolution in terms of area, with an exponential increase with every bit of resolution. Energy is mostly consumed by the OTA to fix the voltage of the access lines and the CSA, which when aggregated can be comparable to the comparators in SAR and Flash ADC. The voltage of operation is low across the memristors which reduces the energy spent on analog computing within the array. Vector/matrix size is quite scalable by investing in high-performing OTAs. Post-processing is limited as the digital output is the final ADC output.

The following challenges remain unaddressed, as highlighted in Fig. 2.15;

- **Non-idealities**: Memristor and circuit non-idealities are sufficiently addressed with programming noise mostly dominating the MVM error.
- **Efficiency constraints**: Although they are less expensive with respect to SAR and Flash ADCs, A/D conversion is typically expensive in terms of area and energy as the number of components increase exponentially increases with bit resolution. The required settling time and energy of the OTA to settle access lines increases with the size of the array and the latency can be reduced but with large sized OTAs.

In short, state-of-the-art logic accelerators are mostly small sensing margins, especially for (N)AND and XOR operations, and lack solutions related to wire parasitics. In addition, they are slow and energy-inefficient which is aggravated as they perform multi-operand logic in a sequential manner and also involve dedicated SAs with multi-reference blocks to operate. On the other hand, state-of-the-art arithmetic accelerators focusing on ADC design and implementation mostly suffer from small sensing margins as the vector/matrix size increases, and are prone to programming errors of the memristor device. ADC solutions are either high-speed with large area and energy overhead or low-power and compact with long exponential conversion time.

PART II: CIM-BASED LOGIC ACCELERATORS

This part includes the following chapters.

- ***Chapter 3**: Cascaded logic using PCM devices*
- ***Chapter 4**: Robust logic accelerator using STT-MRAM devices*
- ***Chapter 5**: Referencing-in-array scheme for 2T2R RRAM*
- ***Chapter 6**: Accelerating RRAM testing using CIM*
- ***Chapter 7**: MOXOR-CIM for RRAM*
- ***Chapter 8**: Outlook and discussion*

This part is partially based on [64–68].

3

CASCADED LOGIC USING PCM

In recent years, several in-memory logic primitives were proposed where bit-wise logical operations are performed in memory by exploiting the physical attributes of memristive devices organized in a crossbar array. However, a convincing real-world application for in-memory logic and its experimental validation are still lacking. Herein, the application of database query where a database is stored in an array of binary memristive devices is presented. The queries are formulated in terms of bulk bit-wise operations and are executed in memory by exploiting Kirchhoff's current summation law. The concept is experimentally demonstrated by executing error-free queries on a small 4×8 selector-less phase-change memory crossbar. The impact of crossbar size, the resistance of routing wires, and inter-device variability on the accuracy of the logical operations are studied through numerical and circuit-level simulations. Finally, a system for cascaded query is proposed that combines the in-memory logic with conventional digital logic and its functionality is verified on a healthcare-related database. It is estimated that an 11-step long query is executed in 36 ns, consuming 560 μW, thus achieving an energy efficiency of 166 TOPS/W.

This chapter is based on [64]. This work is reported in collaboration with IBM Research Zurich under an EU project. Note that i) device fabrication and characterization, and ii) demonstration on the 4x8 prototype were primarily performed by Iason Giannopoulos (IBM). My contributions include i) circuit model emulating PCM, ii) read and write biasing schemes, iii) simulation, and analyses of emulated PCM-based CIM with variations, wire parasitics, etc., and iii) design and implementation of cascaded logic for database query applications.

3.1. INTRODUCTION

It has recently been shown that computational primitives such as bulk bitwise logical operations [129–131] can be efficiently realized using memristive devices. Even though the matrix-vector multiplication primitive is gaining traction in a range of applications such as deep learning inference [132, 133], sparse coding [134], linear solvers [135] and hyperdimensional computing [136], there are only few experimental demonstrations of end-to-end applications involving in-memory logic operations.

A potential application involving a high percentage of logical operations is data analytics in database management systems. When performing queries on large databases, one of the main challenges is to retrieve the stored data and bring it to the processor that will execute the query. The latency and energy associated with moving data to the processor is a key performance bottleneck. Conventional acceleration methods range from software-oriented schemes that solve queries approximately [137] to hardware-oriented systems that parallelize operations using GPUs [138]. However, these approaches do not tackle the main bottleneck of data movement caused by the limited cache capacity of processing units and limited memory bandwidth. These constraints are further enhanced by the constantly increasing size of real-world databases. Near memory computing that incorporates CMOS-based computing units close to the main memory is a promising approach to reduce the cost of data movement [139]. However, it does not completely eliminate the memory/processor separation, involving additional challenges such as integrating the logic units into the main memory chip. Another tantalizing prospect is that of storing large databases in dense arrays of memristive devices and executing the queries directly in-memory by exploiting the analog properties of memory devices and specific circuit designs. In a previous work, a novel resistive memory-based architecture was proposed to support a variety of query functionalities (including bitwise logical operations) and perform them explicitly in-memory [140]. However, the proposed design, based on a content-addressable memory structure, does not support the efficient cascading of multiple in-memory query operations consisting of different logical operands. Moreover, a validation of the feasibility of realizing accurate query operations using existing resistive memory crossbar arrays has been missing.

In this chapter, we present the realization of database queries based on in-memory logic (IML). We experimentally demonstrate the concept on a small fabricated Proj-PCM crossbar array by successfully executing multiple queries in-memory involving AND and OR operations. We quantify the impact of inter-device variability and finite wire resistance on the accuracy of the system with respect to its array size, using circuit simulations. Subsequently, we introduce a design that executes cascaded queries of arbitrary length and complexity, by combining IML with near-memory logic circuits at the crossbar periphery. Using the Cleveland heart disease database as a case study, we demonstrate the functionality of the proposed system via HSPICE circuit simulations and evaluate its throughput and energy efficiency.

3.2. DATABASE QUERY USING SCOUTING LOGIC

Query operations can be performed on databases that are structured collections of attributes or features, associated with different subject or item entries. The objective of a

query is to retrieve the entries from the database that satisfy certain constraints related to the attributes. When databases are represented in a bitmap representation (vectors of logical "0" and "1"), it is possible to formulate the queries as bulk bitwise operations on the attribute vectors (see Figure 3.1a). The key idea of in-memory database query is to store the database entries in arrays of memristive devices using their conductance as the logic state variable. Memristive devices can be scaled down to nanoscale dimensions [141, 142] and their non-volatile storage capability ensures that no energy is spent to retain the stored information unlike in DRAM or SRAM. Subsequently, the bulk bitwise operations associated with the query are performed directly in the memristive arrays by employing IML.

For IML, we exploit the non-volatile binary storage capability of the memristive devices. For example, the 0s and 1s can be represented by the low and high conductance states of the memristive devices, respectively. Several logical operations can be enabled through the interaction between the voltage and conductance state variables [143, 144]. One particularly interesting characteristic of certain memristive logic families is statefulness, where both the operands and the result are stored as conductance state variables [145–147]. However, stateful logic operations involve the repeated programming of memristive devices [148, 149]. This is highly undesirable given the limited cycling endurance of these devices, as well as the large energy footprint associated with programming operations (typically in the order of picojoules per write event).

Hence, for the database query problem, we resort to non-stateful logic, where the logical operands are stored as conductance values and the result of the logical operation is obtained as a current signal. This method was first implemented in Pinatubo [150]

a

Attribute \ Entry	#1	#2	#3	#4	#5
A	0	1	0	1	0
B	1	0	0	1	0
C	0	1	0	0	1
D	0	0	1	1	0
E	0	1	1	0	1
F	1	1	0	0	0
C OR E	0	1	1	0	1
A AND D	0	0	0	1	0

b

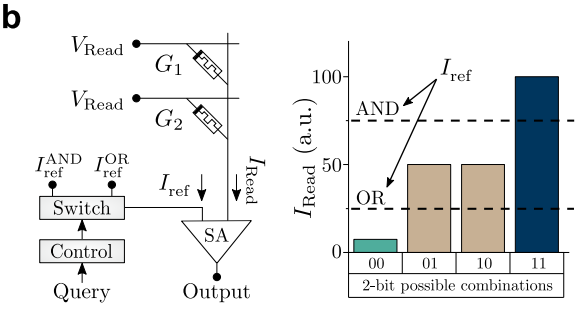


Figure 3.1: In-memory database query. (a) An example database consists of 5 subject or item entries each of them with 6 attributes expressed in a binary format (typically referred to as a bitmap representation). Queries consist of performing logical operations between the attributes and can be executed as bitwise logical operations. (b) A schematic illustration of scouting logic employed for bitwise logical operations. The operands are stored in terms of the conductance states of memristive devices organized in a crossbar configuration. To perform logical operations, multiple rows are biased simultaneously and the resulting current is sensed per column using variable reference sense amplifiers. By choosing the appropriate reference currents, AND and OR operations can be performed in place without having to move the operands into an external processing unit.

and has inspired the concept of scouting logic [151, 152]. The operands stay fixed in the memory array, meaning that device programming is not part of the execution routine. Figure 3.1b shows the realization of bitwise logical operations using scouting logic. When memristive devices are organized in a crossbar configuration, it is possible to implement logical operations such as AND and OR. Multiple rows are by simultaneously activated and sense amplifiers (SA) are configured with the appropriate reference currents (I_{ref}). This enables us to execute the database query operations. For example, a query that requests the input entries that satisfy attributes "A" AND "D" is performed by biasing simultaneously the crossbar rows that correspond to A and D with a specific read voltage (V_{Read}). The resulting read current (I_{Read}) at each column corresponds to the summed conductance of the memristive devices according to Kirchhoff's circuit laws. The logical output is obtained from comparison with the predefined I_{ref} in a SA at each column. Therefore, the logical result is calculated in place without having to move the contents to an external processing unit. Moreover, the output is a vector with equal length to the number of crossbar columns and contains the query response for all the entries, thus facilitating the operation with high parallelism. Note that although the concept is not limited to two attributes per operation, SAs are configured only for one kind of logic at once. For example, "A" AND "C" AND "D" is executable with appropriate adjustments to I_{ref} values, although it may require increased SA precision as a function of the number of operands.

Processors based on memristive devices with logic capabilities have been demonstrated in the past [153, 154] and even specifically using the scouting logic concept [155]. However, several aspects related to the functionality of the database application at the array level have been overlooked by previous studies, especially in the case of selector-less crossbars. In the following sections, we analyze the limitations arising as the crossbar size increases and we set boundaries for successful scouting logic as a function of size and routing wire resistance. Moreover, we develop and calibrate a dynamic circuit model that accurately simulates the read and write operations on PCM crossbars, in order to derive the maximum selector-less crossbar size for reliable database querying. Another fundamental shortcoming from previous scouting logic demonstrations is the optimal selection of reference currents, which is the most important parameter of scouting logic. It is typically chosen to be the geometrical mean of the expected I_{Read} values as shown in Figure 3.1b. However, this is not optimal in the presence of inter-device conductance variations and the robustness against such nonidealities is not extended to its full potential. In the following section, we analytically derive the optimal I_{ref} values, and based on this, we present an end-to-end experimental demonstration of the database query application on a fabricated selector-less PCM crossbar.

3.3. EXPERIMENTAL DEMONSTRATION

In this section, we present a detailed description of the experimental demonstration of in-memory database query. First, we describe the design and fabrication of selector-less crossbar arrays that enabled the experimental demonstration. It is possible to fabricate memristive crossbars with or without selector devices. When each memristive device is placed in series with a selector such as a diode or a transistor, it greatly simplifies the

operational challenges associated with the crossbar but at the expense of higher fabrication cost and larger footprint [156–158]. Even though the proposed in-memory database query approach can be implemented using both types of crossbars, here we study the more challenging case of selector-less crossbars. Experimental results are presented using fabricated crossbar arrays of Proj-PCM devices. Although for binary storage conventional PCM would have been an equivalent candidate, the performance benefits in conductance state stability and reduced readout noise are again exploited, following the previous sections of this thesis that advocate Proj-PCM as a general-purpose substitute for conventional PCM technology.

3.3.1. FABRICATION OF CROSSBAR ARRAYS

Proj-PCM devices were fabricated on a 100 nm SiO_2 isolation layer that was thermally grown on a 525μm n-type silicon wafer. The projection layer was made of Tantalum ni-

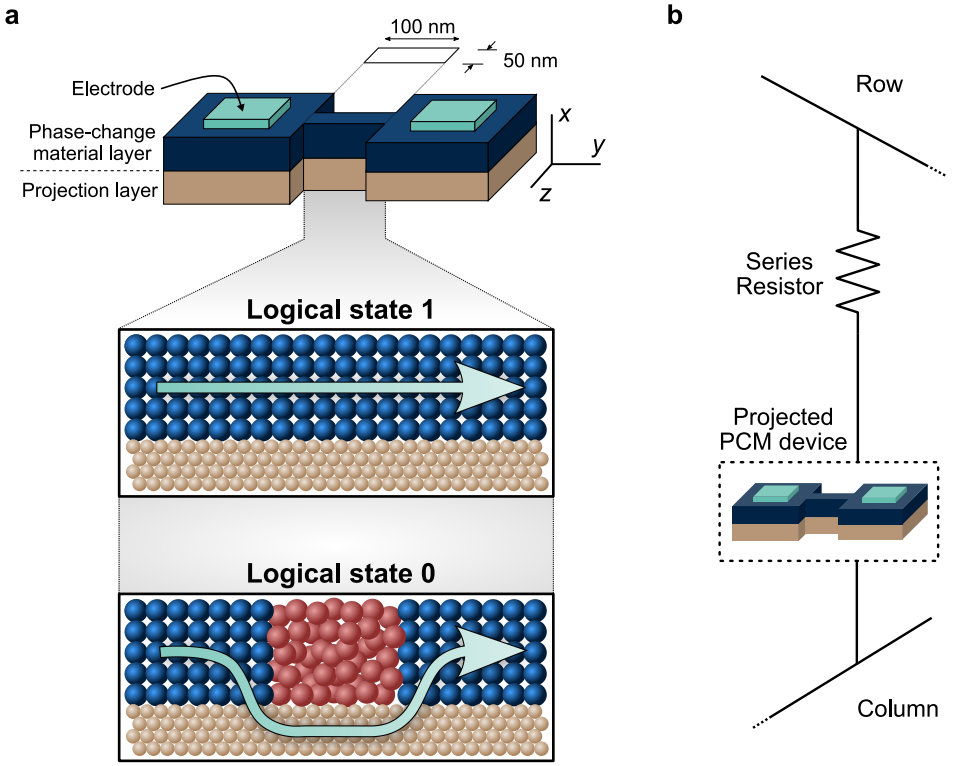


Figure 3.2: (a) Logical "1" is assigned to the highly conductive SET state of a Proj-PCM device, in which the phase-change material is in the crystalline phase. Logical "0" to the highly resistive RESET state where a portion of the phase-change material is amorphous. (b) Each row/column cross-point is populated with a Proj-PCM device and a fixed resistor in series to limit the programming current.

tride (TaN) formed by reactive sputtering on a Tantalum target in the presence of gaseous N_2. Note that in this step, the sheet resistance of this layer can be tuned by adjusting the N_2 flow rate in the sputtering chamber as well as the deposition temperature. Subsequently, an ultra thin film of phase-change material (Sb_2Te_3 10nm) was sputtered and protected by a 5nm SiO_2 capping layer. These three layers were deposited sequentially in the same sputtering chamber, in order to avoid exposure to atmospheric air that could oxidize the nitride interface and the PCM material. Several e-beam lithography routines were executed in order to: i) pattern the stacked layers via ion-milling to the design shown in Figure 3.2a and open contact areas through the capping layer, ii) define the contact electrodes, the series resistors and the 1^{st}-level metal wires by Reactive Ion Etching (RIE) of a sputtered Tungsten (W) layer, and finally iii) to form the 2^{nd}-level metal and robust top-level pads for the probe-card used in the electrical characterization setup.

A scanning electron micrograph of a fabricated 4×8 crossbar is shown in Figure 3.3a. Our Proj-PCM devices have a lateral design and hence the design differs from the conventional cross-point structure, in which the memristive devices are fabricated at the overlapping area of BL and WL wires. Although the area efficiency may be reduced, there is no functional difference between this crossbar architecture and the conventional cross-point one, because the left and right terminals of this lateral device are connected to the column and row wires, respectively (see Figure 3.3b). Each cross-point is populated with a metallic resistor in series with a Proj-PCM device as shown in Figure 3.2b.

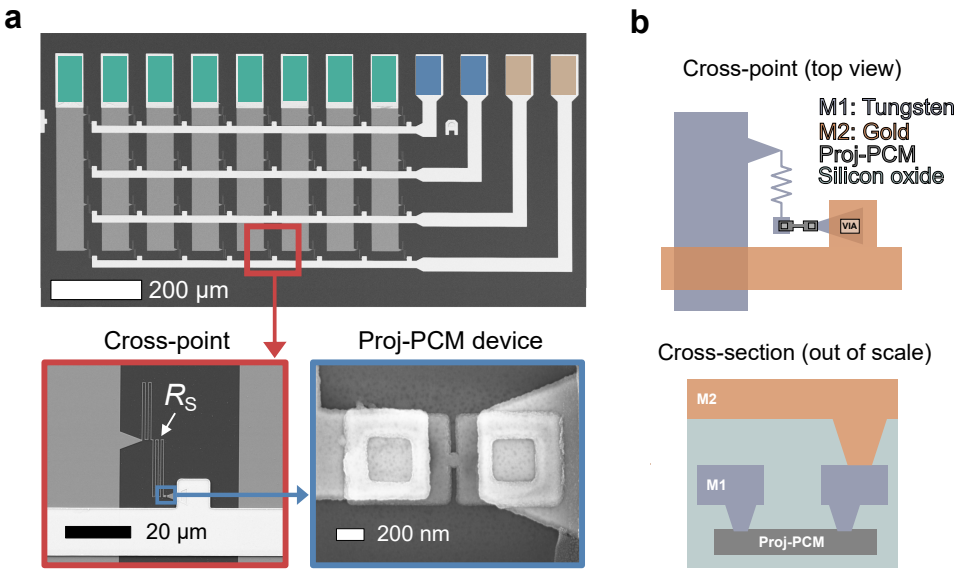


Figure 3.3: (a) SEM images of the fabricated 4×8 selector-less crossbar that consists of a Proj-PCM device (along with a fixed series resistor) at each cross-point. (b) Schematic illustrations of the corresponding SEM images. The left electrode is connected to M1 (column) and the right one to M2 (row). For better process control, the right electrode consists of a Tungsten layer that gets connected to M2 through a VIA in the dielectric.

A crossbar without selectors suffers inherently from current sneak-paths that depend on the states of the other devices in the array [159]. Special biasing schemes have to be used for each operation to limit this effect and make the selector-less crossbars functional (see Figure 3.4). To successfully write an individual device through the selected cross-point, pulses with half the programming voltage have to be applied simultaneously to all rows and columns, except for the row and column on which the selected device is located. The programming pulse is applied to the selected row, while the selected column is grounded. A fundamental prerequisite for this method is that V_{th} is higher than half of the programming pulse amplitude, otherwise, the state of the non-selected devices along the selected row and column would be disturbed. On the other hand, precise current measurement is required for both read and scouting logic operations, in which the non-selected rows are grounded so that the contribution of sneak paths to the selected column current is minimized. The basic goal is to suppress unwanted voltage differences across non-selected devices by giving them additional grounding paths other than the column readout node [160–162]. Although in a scouting logic operation, all column currents are read simultaneously, in cases where hardware limitations impose that certain columns are not read, then they have to be grounded as well.

3.3.2. BASIC ELECTRICAL CHARACTERIZATION

The fabricated Proj-PCM devices were electrically characterized individually. A RESET pulse in the 100ns timescale was applied to provide adequate power to melt the phase-change material (see Figure 3.5a). A nonlinear *I-V* characteristic of the amorphous-to-crystalline phase transition is shown in Figure 3.5b. Moreover, we highlight one of the main performance benefits of Proj-PCM devices, which is the notably weaker field dependence over a wider biasing range (see Figure 3.6) [163]. A resistor (R_S) in series with the device serves for controlling the current in a selector-less implementation. Despite

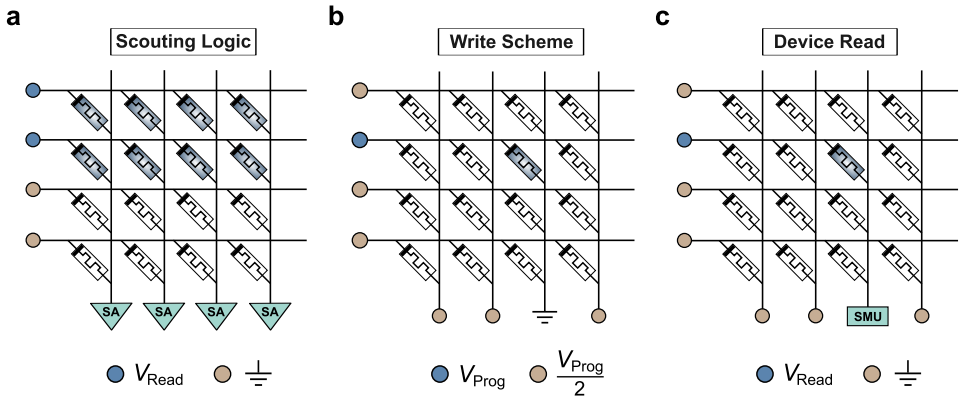


Figure 3.4: Biasing schemes employed to deal with sneak-paths during write, read and scouting logic operations. During write, a half-selection rule is employed to avoid disturbing the state of non-selected devices. During scouting logic, the non-selected rows are grounded to minimize the effect of sneak currents.

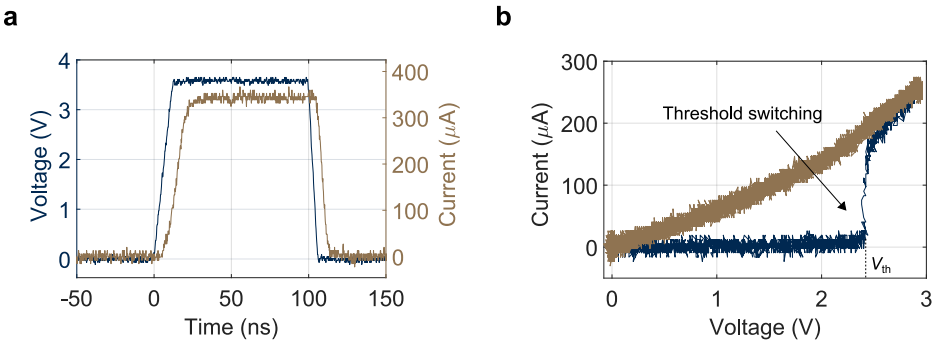


Figure 3.5: Projected phase change memory devices. (a) A RESET programming pulse. (b) Quasi-static *I-V* characteristics generated from a SET pulse. The threshold switching voltage (V_{th}) is critical for selector-less crossbar operation.

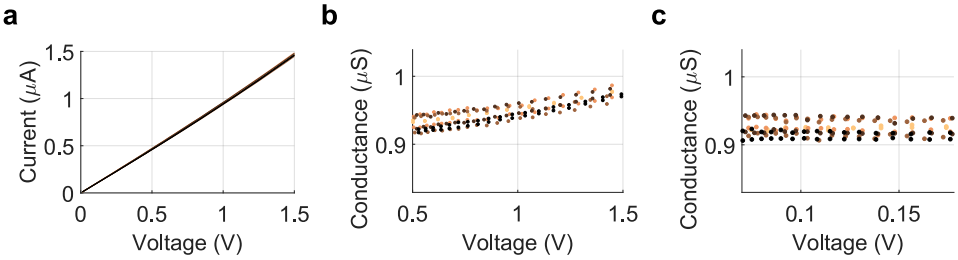


Figure 3.6: (a) *I-V* characteristics of 5 RESET (logical 0) states. (b) Conductance-voltage plot showing the nonlinear dependence at the high voltage range. **c** Conductance-voltage plot at the range of V_{Read}, in which the devices have Ohmic behavior.

the obvious areal cost and increased power consumption, current control during threshold switching is essential for the normal operation and endurance of the used devices. The threshold voltage (V_{th}) is a particularly critical parameter for the crossbar operation, therefore it will be studied thoroughly in Section 3.4.2. The variability associated with the conductance states upon repeated switching can be mitigated by an iterative programming scheme involving a sequence of program-and-verify routines (see Section 3.3.5). With this scheme, SET and RESET conductance values were roughly uniformly distributed over multiple cycles (see Figure 3.7b).

3.3.3. SOURCES OF INACCURACY

Prior to presenting the database query experiment, we investigated two primary sources of inaccuracies related to IML. Scouting logic operation is executed by a simultaneous read of two devices per column. Given that there are three possible combinations between the states of the involved devices, there are three distinct nominal values for the column currents (see Figure 3.1b).

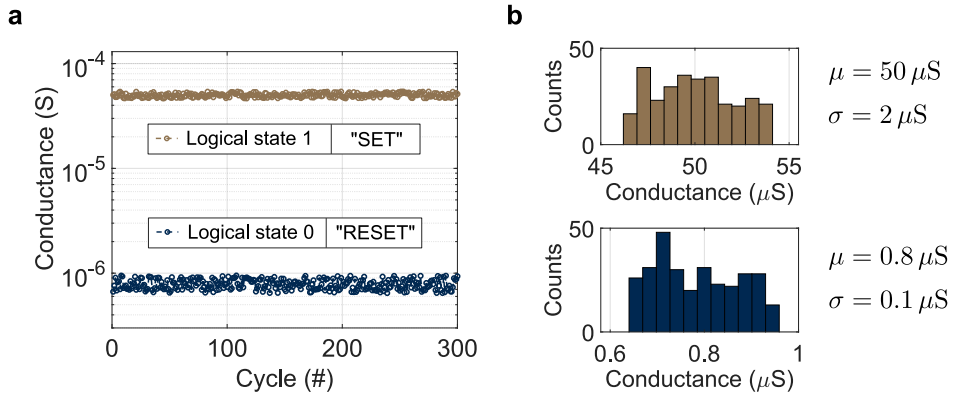


Figure 3.7: Projected phase change memory devices. (a) Successive SET/RESET switching behavior of a projected-PCM device. An iterative programming scheme was employed to reduce the SET and RESET variability. (b) Conductance distribution of SET (top) and RESET (bottom) states, along with their mean and standard deviation values.

The first source of inaccuracy arises from nonideal routing wires. The finite wire resistance causes voltage drops that lead to a reduced effective V_{Read} across the selected PCM devices. In particular, it becomes crucial for the devices that are positioned the farthest away from the voltage supply node and require the longest routing wires. The effect of this voltage drop is larger for the SET state because the routing wire resistance may become comparable with the SET state as the crossbar size increases. The finite resistance of the routing wires also results in sneak-path currents that tend to increase the effective I_{Read} from the RESET states by adding parasitic contributions from non-selected devices in the array. Although the row-grounding method is employed, sneak-paths cannot be entirely suppressed, because voltage gradients along the row-wires can bias devices that, otherwise, would have grounded terminals. Combined, these intertwined effects diminish the effective read-current ratio, to a point where the three possible column currents cannot be differentiated reliably by a sense amplifier.

The routing resistance path is a critical fabrication parameter that depends on both the physical distance between the devices and the sheet resistance of the routing metal. Ultimately, wire resistance per unit cell is the key factor that determines the maximum size of a functional crossbar. We used a circuit simulator to study the reduction of the read-current ratio as a function of wire resistance and crossbar size. We took the worst possible scenario in which: i) all devices are programmed to the SET state to maximize sneak-currents and ii) the selected device is at the outermost corner that has the longest routing path. The selected device is programmed sequentially to both logical states and their read-currents are calculated. We used these values to reconstruct and compare the ratios of column currents I_{00}, I_{01} and I_{11} that correspond to the logical combinations of $\{0,0\}, \{0,1\}$ and $\{1,1\}$. Resolution limits were set according to the rule that SA needs a signal that is at least 20% higher than its reference current, in order to work reliably against process-related variations and voltage imprecision. The failure points for AND and OR

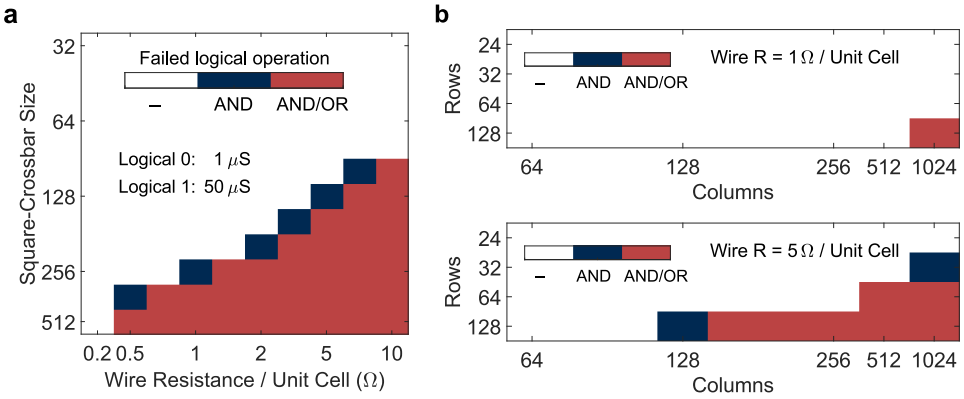


Figure 3.8: Inaccuracies associated with in-memory logic operations: (a) Circuit simulations showing which logical operation fails as the routing-wire resistance and the crossbar size increase. The study was based on the worst-case scenario where the read device lies at the longest routing point and the rest of the crossbar is at the SET state maximizing the sneak currents. Resistance/size combinations within the white area correspond to functional systems. AND requires higher precision and thus fails earlier than OR. (b) Circuit simulations of crossbars with non-unitary aspect ratio and wires of 1Ω and 5Ω per unit cell. A reduced number of rows (attributes) results in functional crossbars with a larger number of columns (entries).

operations are met when $I_{11}/I_{01} \leq 1.2$ and $I_{01}/I_{00} \leq 1.2$ respectively. AND requires higher precision, hence fails slightly earlier than OR (see Figure 3.8a). One approach towards increasing the size of functional crossbars is to employ an adaptive SA that adjusts the reference currents proportionally, in order to compensate for the changes in readout current along the routing path. Another approach is to use crossbars with a non-unitary aspect ratio (see Figure 3.8b). The majority of real-world databases have many more columns (entries) than rows (attributes), therefore the number of rows can be reduced to incorporate a larger number of columns.

The second key challenge for accurate IML is the inter-device conductance variations. During a logical operation, the column current is proportional to the sum of the two conductance values that represent the operands, for example, a low and a high one would be: $I_{\text{Read}} = V_{\text{Read}}(G_0 + G_1)$. Because of programming imprecision, G_0 and G_1 will be randomly spread around their respective target values (see Figure 3.7b). The probability distribution of the sum of two or more independent random variables is the convolution of their individual distributions. The distributions of the three possible column currents were analytically derived (see Section 3.3.4) and are plotted in Figure 3.9a. The distributions of G_1 and G_0 were assumed to be uniform with mean values of $50\mu S$ and $1\mu S$, respectively. We set $V_{\text{Read}} = 0.1V$ and calculated the expected distribution of the three column currents for three different values of standard deviation (σ), expressed as the percentage of their mean conductance (μ). Based on the observation that the SET state variability has a much higher impact on the column current, the *optimal reference cur-*

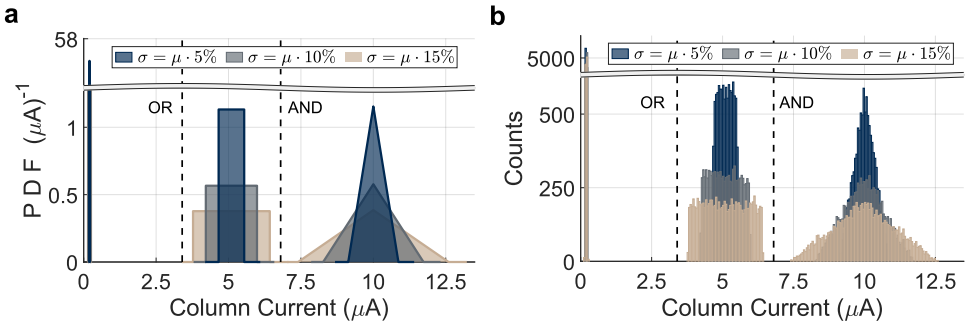


Figure 3.9: Inaccuracies associated with in-memory logic operations: (a) Probability density function (PDF) of the read current for the three possible state combinations. Each device conductance is assumed to be uniformly distributed with the standard deviation expressed as the percentage of their mean. Column current is calculated assuming $V_{\text{Read}} = 0.1V$ bias at both devices. The reference currents for AND and OR operations are set according to Equations 3.1. (b) Numerical simulations of the three possible column currents given by biasing two devices with $V_{\text{Read}} = 0.1V$. Each device is programmed to a conductance state that follows uniform distribution and the same relative standard deviations are used for a direct comparison with (a).

rents for AND and OR operations are given by Equation 3.1.

$$I_{\text{ref}}^{\text{AND}} = I_{00} + \frac{2}{3}(I_{11} - I_{00}) \tag{3.1a}$$

$$I_{\text{ref}}^{\text{OR}} = I_{00} + \frac{1}{3}(I_{11} - I_{00}) \tag{3.1b}$$

The analytically derived distributions are confirmed by numerical simulations of 20,000 devices (see Figure 3.9b). Every pair of devices is programmed to conductance states uniformly distributed around the means $\mu_{G_1} = 50]\mu S$ and $\mu_{G_0} = 1\mu S$. Histograms corresponding to I_{00}, I_{01} and I_{11} show a perfect match with the mathematical analysis of Section 3.3.4.

3.3.4. OPTIMAL REFERENCE CURRENTS

Here we present the theoretical calculation of the optimum $I_{\text{ref}}^{\text{AND}}$ and $I_{\text{ref}}^{\text{OR}}$ values prescribed by Equations (3.1). Their equations are derived with a statistical analysis of the sum of uniformly distributed conductance values.

When a variable x follows a uniform distribution within the range $[a, b]$, the equations for the mean and the standard deviation are $\mu = \frac{1}{2}(a+b)$ and $\sigma = \frac{b-a}{\sqrt{12}}$. The bounds a, b can be expressed in terms of μ, σ as:

$$a = \mu - \sigma\sqrt{3} \tag{3.2a}$$

$$b = \mu + \sigma\sqrt{3} \tag{3.2b}$$

The probability density function (PDF) of the continuous uniform distribution is:

$$f(x) = \begin{cases} \dfrac{1}{b-a} & \text{for } a \le x \le b, \\ 0 & \text{otherwise.} \end{cases} \tag{3.3}$$

The Irwin-Hall distribution [164] gives the probability density function of the sum of n independent variables uniformly distributed in the range $[0,1]$. Let two independent random variables follow continuous uniform distribution $X, Y \sim U(0,1)$. Their sum is $X+Y=Z$ and the probability density function is the special case of the Irwin-Hall distribution for $n=2$, in which Z follows a triangular distribution:

$$f_Z(z) = \begin{cases} z & \text{for } 0 < z \le 1, \\ 2-z & \text{for } 1 < z < 2, \\ 0 & \text{otherwise.} \end{cases} \tag{3.4}$$

However, we require the general solution of this problem when the two uniform distributions have different ranges $X \sim U(0,\chi)$ and $Y \sim U(0,\psi)$. Without loss of generality, we let $\chi < \psi$ and we can prove either with integration or with a geometric approach [165], that the resulting probability density function is:

$$f_Z(z) = \begin{cases} 0 & \text{for } z \le 0, \\ \dfrac{z}{\chi\,\psi} & \text{for } 0 < z < \chi, \\ \dfrac{1}{\psi} & \text{for } \chi \le z < \psi, \\ \dfrac{\chi+\psi-z}{\chi\,\psi} & \text{for } \psi \le z < \chi+\psi, \\ 0 & \text{for } z \ge \chi+\psi. \end{cases} \tag{3.5}$$

In this general solution, Z follows a trapezoidal distribution and the exact shape depends on the ratio χ/ψ. In the extreme case when $\chi \ll \psi$ the trapezoid tends to become a rectangle and when $\chi = \psi$ the plateau folds to a single point and the distribution becomes triangular. Moreover, the ranges could be translated along the horizontal axis by ζ, ξ so the new variables are $X \sim U(\zeta, \chi+\zeta)$ & $Y \sim U(\xi, \psi+\xi)$. This final case describes exactly our example, where $I_0 \sim U(a_0, b_0)$ & $I_1 \sim U(a_1, b_1)$ are directly proportional to conductance distributions according to Ohm's law $I_i = G_i V_{\text{Read}}$. Therefore the variables change as follows:

$$\zeta = a_0, \quad \xi = a_1, \quad \chi = b_0 - a_0, \quad \psi = b_1 - a_1 \tag{3.6}$$

Equations (3.5) and (3.6) are used to plot the analytic distributions in Figure 3.9a. The ranges are defined according to Equations (3.2) for various standard deviations.

The mean values of the three current distributions (I_{00}, I_{01}, I_{11}) correspond to the addition of the individual current means: $I_{00} = 2\mu_0$, $I_{01} = \mu_0 + \mu_1$, $I_{11} = 2\mu_1$. $I_{\text{ref}}^{\text{AND}}$ can be

calculated as the meeting point of I_{01} and I_{11} distributions. Equations (3.5) and (3.6) show that the right bound of I_{01} increases as $x = b_1 + b_0 - I_{01}$, which according to Equation (3.2b) is a function of the two standard deviations: $x = \sqrt{3}(\sigma_0 + b\sigma_1)$.

Similarly, the left bound of I_{01} increases as: $L - x = I_{11} - 2a_1 \xrightarrow{(3.2a)} L - x = 2\sqrt{3}\sigma_1$ where $L = \frac{1}{2}(I_{11} - I_{00})$ is the total distance between the two means. Solving for the point where the right bound of I_{01} is equal to the left bound of I_{11}, yields Equation (3.7). This expression reduces to Equation (3.1a) when $\sigma_0 \ll \sigma_1$.

$$I_{\text{ref}}^{\text{AND}} = \left(\frac{\sigma_1}{3\sigma_1 + \sigma_0}\right) I_{00} + \left(\frac{2\sigma_1 + \sigma_0}{3\sigma_1 + \sigma_0}\right) I_{11} \tag{3.7}$$

Equivalently for $I_{\text{ref}}^{\text{OR}}$, the right bound of I_{00} and the left of I_{01} increase as a function of the standard deviations $x = 2\sqrt{3}\sigma_0$ and $L - x = \sqrt{3}(\sigma_0 + \sigma_1)$, respectively. Their meeting point is given by Equation (3.8).

$$I_{\text{ref}}^{\text{OR}} = \left(\frac{\sigma_1 + 2\sigma_0}{\sigma_1 + 3\sigma_0}\right) I_{00} + \left(\frac{\sigma_0}{\sigma_1 + 3\sigma_0}\right) I_{11} \tag{3.8}$$

In our example, this value is very low. Although it is obvious that OR operation is notably robust to device variability, the functional limit for a system that solves arbitrary queries is set by the AND failure point. In addition, lower reference currents require larger transistors in the SA units, increasing the SA area and energy consumption. Therefore, in our example, we optimized the system by increasing $I_{\text{ref}}^{\text{OR}}$ to a value that is equally spaced with $I_{\text{ref}}^{\text{AND}}$ around the center of I_{01} distribution (Equation 3.1b). A similar study should be carried out, in case the conductance states follow Gaussian or any other distribution (see Section 3.3.4).

3.3.5. IMPLEMENTATION OF QUERY OPERATION

For the experimental demonstration, we employed a database with information about scientific awards presented to some of the worldwide IBM Research laboratories. We programmed the devices of a 4×8 crossbar according to the bitmap representation of the database.[1] SET states follow a uniform distribution around $\mu = 51\mu S$ with $\sigma = 3\mu S$ and RESET states have $\mu = 0.8\mu S$ and $\sigma = 0.1\mu S$ (see Figure 3.10a). An iterative program-and-verify scheme was responsible for the relatively low conductance variations. This method applies a programming pulse and checks whether the resulting conductance falls within a predefined target range given by $G_{\text{target}} \pm \delta G$. If the target state is not achieved, the voltage amplitude of the $i+1$ programming pulse is readjusted with respect to the i^{th} via a feedback equation:

$$V_{i+1} = V_i + \theta \cdot \left[log(G_{\text{i}}) - log(G_{\text{target}})\right] \quad \text{where } \theta = 0.2 \tag{3.9}$$

[1] **Electrical characterization setup:** This experiment features devices with multiple terminals (12 pads). The crossbar pads were contacted by a probe card (Celadon T40AF) with 25 probes arranged in-line with a 100 μm pitch. A 4-channel source-measure unit (Keithley 2606B) was used to DC bias the rows and measure the column currents. The nanosecond-long programming pulses were applied by a Pulse Generator (Agilent 81110A) and monitored by a digital oscilloscope (Agilent MSO6104A). The signals from and towards these instruments were distributed to the appropriate pads via a switching matrix (Keithley 707A).

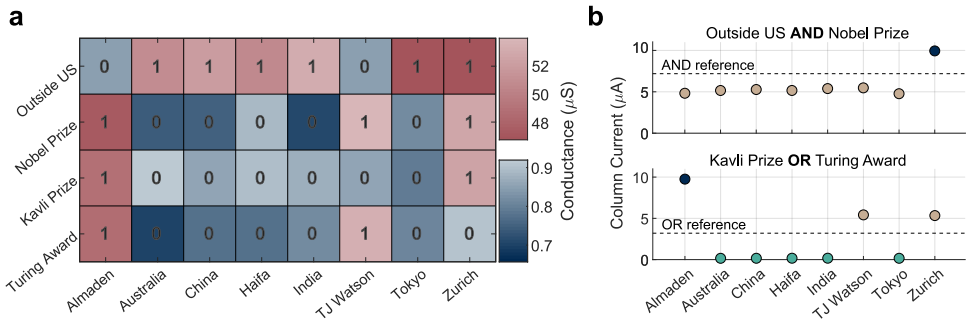


Figure 3.10: Database query experiment. (a) The bitmap representation of the IBM Research awards database is mapped onto a 4×8 crossbar. Depending on the 0 or 1, each device is programmed to the respective conductance for the RESET or SET states. (b) The measured column currents for two queries performed on the IBM Research awards database. The 2 selected rows that correspond to the attributes of each query are biased and the 8 column currents are plotted against the entries that are assigned to each column node.

Database queries are used to search for laboratories that meet certain criteria involving geographical location, awards received, etc. We constructed two simple queries and executed them using the crossbar. The appropriate reference currents are set according to the logical operation associated with the query (see Equations (3.1). The 8 column currents are measured and plotted against the names of the laboratories assigned to each column node (see Figure 3.10b). They are sufficiently separated for reliable detection by a SA.

3.4. SIMULATIONS STUDIES

In this section, we study the scalability of the proposed approach to real-world applications based on simulations. To facilitate this, first, we introduce a circuit-level device model.

3.4.1. A CIRCUIT-LEVEL DEVICE MODEL

The requirement to program the Proj-PCM devices according to the database entries, prior to executing the queries, introduces an additional limitation due to the size of functional crossbars. A consequence of voltage drop along the wires is that the programming voltage should be adequately increased in order to successfully program the outermost devices. In a selector-less architecture, this comes with the penalty of over-biasing the devices nearest to the voltage source, potentially, beyond V_{th} that inevitably would lead to state disturbance. According to the half-selection programming scheme, all the devices along the selected row and column receive a pulse with half the amplitude of the one applied to the selected device (see Figure 3.4b).

To study the influence of programming on the half-selected devices, we developed a circuit model that emulates the behavior of Proj-PCM devices based on conventional

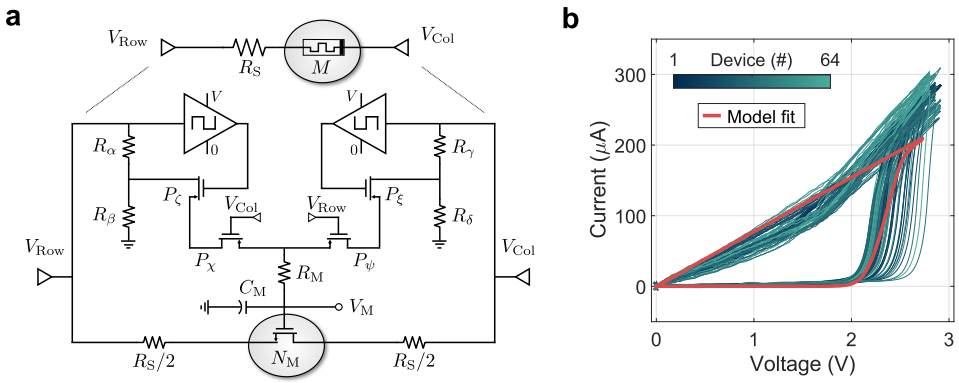


Figure 3.11: (a) A circuit model that emulates the behavior of the Proj-PCM device. The n-MOSFET N_M has a W/L ratio of 30 and the series resistance is $R_S = 5k\Omega$. The switches require additional components which are the resistors $R_\alpha, R_\gamma = 18k\Omega$, $R_\beta, R_\delta = 11k\Omega$ and the p-MOSFETs that have $W/L\{P_\zeta, P_\xi, P_\chi, P_\psi\} = 10$. The gate voltage V_M is built up by charging the capacitor $C_M = 30fF$ through a resistor $R_M = 300k\Omega$. (b) The experimentally obtained I-V measurements from the Proj-PCM devices are compared with that obtained from the emulator.

electronic components such as transistors, capacitors, and resistors drawn from TSMC 90nm library (see Figure 3.11a). It is a two-terminal component, designed to behave like a Proj-PCM that is initially in the amorphous phase and may (or may not) switch to the crystalline phase depending on the voltage level. The device M is modeled by a transistor N_M driven by a circuit of switches. The exact values of these components are finely tuned according to the experimental device characteristics. The emulator is configured to capture the mean experimental behavior of 64 I-V switching characteristics, that were measured on different Proj-PCM devices (see Figure 3.11b). Threshold switching is controlled by Schmitt triggers that are set at the experimentally obtained $V_{th} = 2.1V$. They are supplied with external DC power $V = 4V$, in order to keep the state unaffected from the input signals. Moreover, the circuit is designed in a fully symmetric manner, in order to capture the unipolar nature of PCM devices and be independent of the bias node (V_{Row} or V_{Col}). Non-volatility is ensured by the capacitor C_M that retains the gate voltage required for each conductance state during read or scouting logic operations. Under low electric field both phases have linear characteristics [166]. Because scouting logic is essentially a multitude of low-field read operations, the devices were represented by Ohmic resistors in the scouting logic simulations.

We implemented this model in a Synopsis HSPICE circuit simulator, in order to test the efficacy of the concept on a practical problem with a much larger dataset. The *Cleveland heart disease* database, which is available on the UCI machine learning repository, consists of medical metrics obtained from 303 patients with heart-related health problems[167]. After binarization, the database featured 41 attributes requiring a 41×303 crossbar array, that can be interconnected using a wide range of wire resistances according to Figure 3.9b. Problems arise when the device is at the high conductance state and

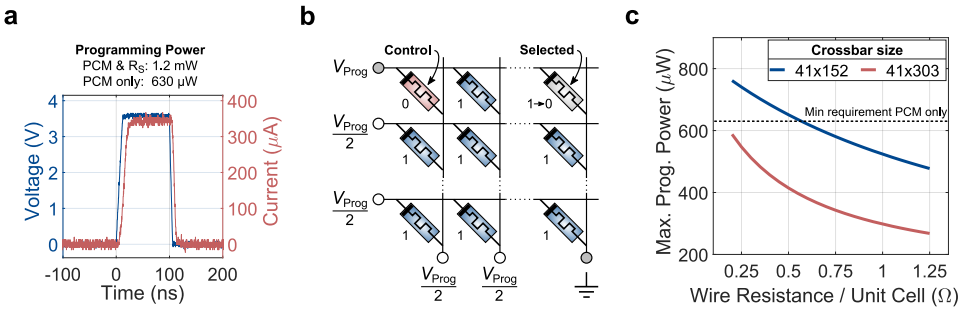


Figure 3.12: (a) The RESET programming pulse is 100ns long with a trailing edge of 10ns. Approximately half of the total programming power is consumed at the series resistance. (b) Schematic representation of a crossbar biased according to the "write scheme". This combination of stored values and the position of the selected device with respect to the voltage node is the most prone to failure. **c** The maximum programming power that can be provided to the farthest-positioned device without exceeding V_{th} at any non-selected device, as a function of wire resistance per unit cell.

there is a significant voltage drop across the series resistance, which implies higher programming voltage at the row node. A representative programming pulse obtained during the individual device characterization shows that approximately half of the programming power is dissipated in R_S (see Figure 3.12a). Given these power requirements, the *minimum* voltage for reliable programming is set at $V_{Prog} = 3.6V$. According to the write scheme (see Figure 3.4b), non-selected devices are biased with half the programming voltage, therefore, the condition for undisturbed write operations is $V_{Prog}/2 < V_{th} = 2.1V$. As a consequence, the voltage level that is applied at the node to compensate for the losses cannot exceed $V_{Prog}^{max} = 4.2V$.

We simulated the system at the worst possible scenario, in which the selected device is the outermost one. All the devices are in the high conductance state, apart from one control device that is at the nearest position to the voltage source (see Figure 3.12b). We measured the programming power delivered to the selected device using wires with resistance in the range of 0.2 to 1.25Ω per unit cell. V_{Prog}^{max} was applied at the selected node, ensuring that the voltage across the control device did not exceed V_{th}. Results indicate that for reliable programming under the worst conditions, the 41×303 crossbar should be split in half and the wire resistance cannot exceed 0.55Ω per unit cell (see Figure 3.12c). The latter implies that as the number of entries increases, the database should expand into more than one crossbar array. In the context of IML, this is not a particularly detrimental problem, because the crossbar can be divided into smaller segments along the rows, without influencing the scouting logic operation at the cost of a marginal increase in circuit-level complexity. Nevertheless, it is clear that a crossbar with selector devices would offer significantly better control and flexibility at the initial stage of write operations, in addition to the major power consumption benefits associated with not employing the "write scheme" at the non-selected devices.

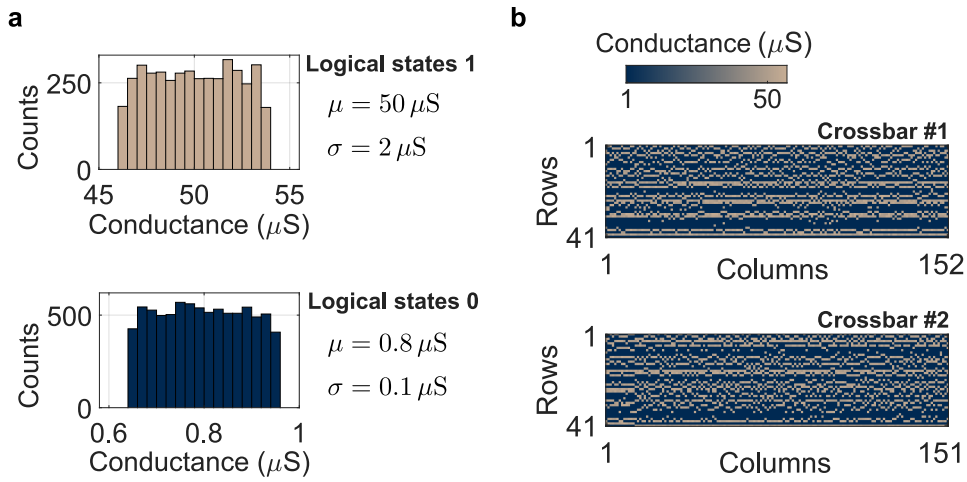


Figure 3.13: (a) The conductance states are uniformly distributed and have the same mean and standard deviation values as the experimental results of Figure 3.7b. (b) The simulated 41×303 crossbar was split in half and populated with the uniformly distributed conductance states of (a) that correspond to the mapped logical states of the heart-disease database.

3.4.2. LARGE-SCALE CROSSBAR SIMULATIONS

We generated SET and RESET conductance states according to the Cleveland heart disease database. The SET (RESET) conductance states were uniformly distributed with a mean and standard deviation of 50(0.8)μS and 2(0.1)μS for the logical 1 and 0 states, respectively (see Figure 3.13a). These distributions are mapped to the 41×152 and 41×151 sub-crossbars (see Figure 3.13b) and replicate the experimental results of Figure 3.7b in a larger scale. The resistance of the routing wires was set to 0.2Ω per unit cell. This value corresponds to the sheet resistance of the two lowest-level metals in the 65nm CMOS technology, assuming the most compact design ($4F^2$ footprint), in which the distance between devices is only twice the minimum wire width.

Queries were constructed for all possible permutations of the 41 attributes and applied to the crossbar. Statistical analysis of the simulated column currents shows errorless query responses for both OR and AND logic (see Figure 3.14). We repeated the simulation with zero-resistance routing wires and we saw that the wires reduced the column currents in all three output cases but more prominently in 10/01 and 11. The current distributions confirm the theoretical analysis. In the case of non-resistive wires, the shapes are triangular for the 00 and 11 cases and trapezoidal for the 10/01 ones. The routing wires changed the shapes of all three distributions towards Gaussian-like, because of the variable routing length depending on the position of each device. Finally, the limitations associated with reliable programming have bounded the individual crossbar size to a relatively low level. Therefore, any effects related to sneak-paths, that could not be suppressed by the "scouting logic" scheme, were practically obscured.

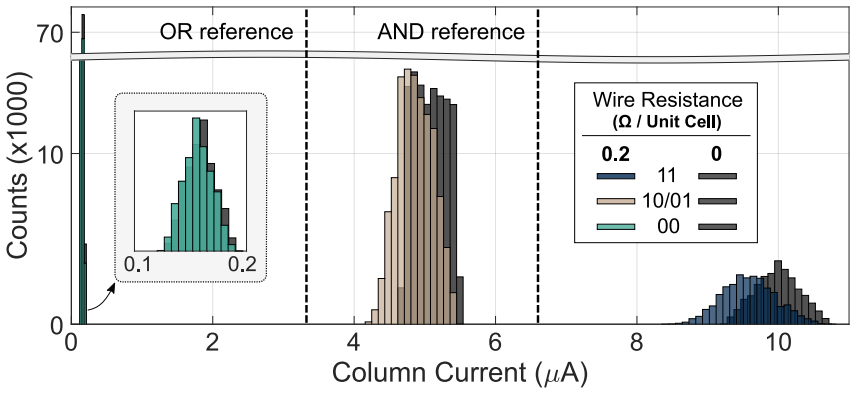


Figure 3.14: Distributions of the simulated column currents when biasing the rows at all possible permutations. Proj-PCM devices are interconnected using wire resistances of 0.2Ω per unit cell and 0Ω. Although non-zero resistance wires reduce the column currents, it is still possible to execute errorless queries.

3.5. DATABASE QUERY USING MAJORITY LOGIC

In-memory database query can be combined with a mechanism first introduced in a bulk bitwise in-DRAM logic demonstration, namely the *majority logic* [168]. In cases where a configurable SA cannot be implemented, the core logical operation for AND and OR can be executed with a single reference current. This is possible by reserving 2 crossbar rows and populating the 1^{st} (2^{nd}) with reference devices that are permanently programmed to the mean conductance values representing logical 1s (0s). One other modification with respect to the system of Figure 3.1b is that now the query controls a switch that applies V_{read} to the row hosting R_0 (R_1) devices to execute an AND (OR) operation. The column current corresponds to the 3 devices in parallel and is compared with the reference one (I_{ref}) that is independent of the query and the position of the switch (see Figure 3.15). Using the same parameters defined in Section 3.3.4, one can derive the optimal I_{ref} for majority logic (see Equation 3.10).

$$I_{\text{ref}} = \left(\frac{5\sigma_1 + \sigma_0}{6\sigma_1 + 2\sigma_0}\right) I_{00} + \left(\frac{2\sigma_1 + \sigma_0}{3\sigma_1 + \sigma_0}\right) I_{11} \tag{3.10}$$

Majority logic is based on a Boolean property. If A, B and R represent the logical states of the two stored data points and the reference device, the logical result is given by the left side of Equation 3.11. It is implied that at least two of the three operands should be 1 for the final state to be 1 and vice versa.

$$AB + RA + RB \;=\; R(A+B) + \bar{R}(AB) \quad \text{where: } R \equiv R_1 \text{ and } \bar{R} \equiv R_0 \tag{3.11}$$

Moreover, this expression can be rewritten to the equivalent format of the right side using R and its complement $\bar{R}$, that we attribute to R_1 and R_0, respectively (see Figure 3.15). Therefore, when the reference device is a logical 1, the final state at the SA is the bitwise

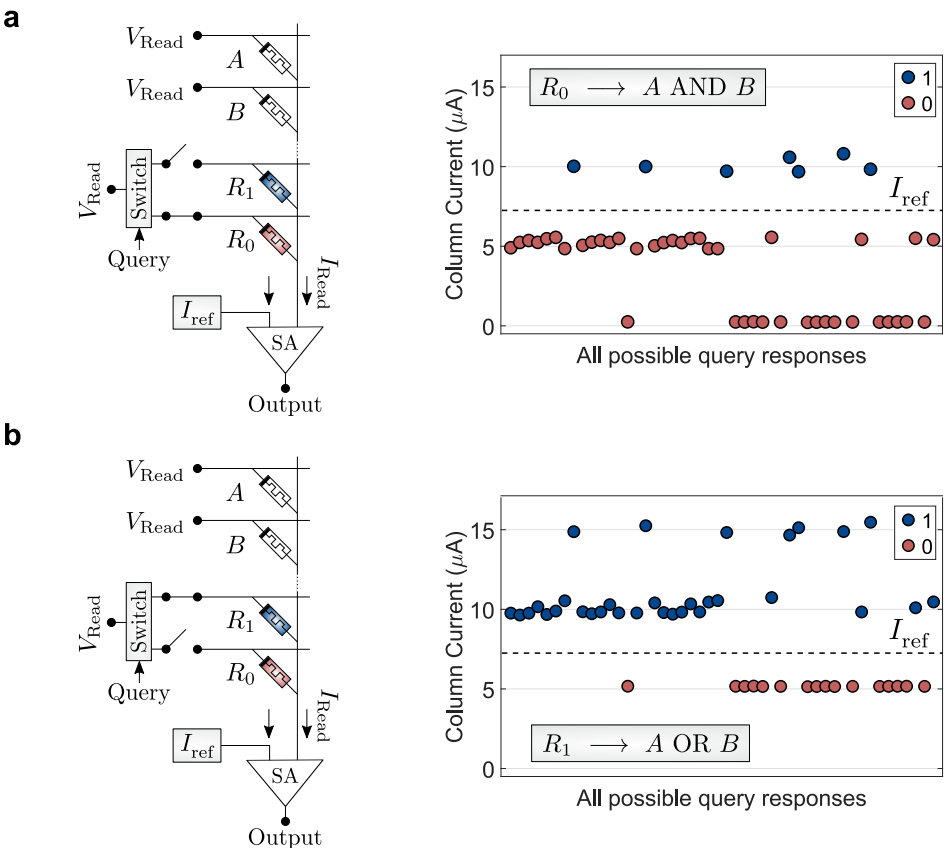


Figure 3.15: Database query using the concept of *majority logic*. System configuration at the columnar level and experimental demonstration for AND (a) and OR (b) logical operations based on the database of Figure 3.10a.

OR of A,B and when the 0 is selected, the result is the bitwise AND of A,B. The concept is experimentally demonstrated using the database of Figure 3.10 and I_{ref} defined by Equation 3.10. At each column, the three operands are biased with V_{Read} and the resulting I_{Read} is plotted for every possible query in both AND and OR configurations (see Figure 3.15).

The main operational difference with DRAM applications is that in DRAM, one controls the state of the devices in a single reference row. Adopting this idea would give rise to the drawbacks of stateful logic applications, due to the high programming power requirements and the relatively low endurance of PCM devices. On the other hand, the implementation that uses two reference rows and a query-controlled switch exploits two key advantages of PCM, namely the decade-long retention and ultra-low power readout with the marginal area cost of reserving one additional row.

3.6. CASCADED DATABASE QUERY

Real-world database queries consist of a multitude of sub-queries with associated logical operations rather than a single query. Solving such a query in the previously demonstrated fashion can be very inefficient. The main challenge is that it requires an additional memory unit for temporary storage of the intermediate logical results and subsequent fetching for further processing along with the next set of logical outputs. To address this, a configurable computing system is introduced that combines in- and near-memory computations. Any query can be expressed as the sum of products (SOP): $(a_1 \cdot b_1) + (a_2 \cdot b_2)$, or the product of sums (POS): $(a_1 + b_1) \cdot (a_2 + b_2)$, where sum and product operators correspond to OR and AND, respectively. However, their occurrence is instructed by the query and is not necessarily alternated OR and AND operations. Hence, an arbitrary query function $F(A)$ can be expressed as a combination of POS and SOP (see Equation 3.12).

$$F(A) = F_1(s) * F_2(s) * ... * F_p(s) \tag{3.12}$$

where $F_i(s) = a_i * b_i$, $*$ is an OR or an AND operator and p depends on the query length.

The key idea of the proposed *cascaded logic* computing system is to perform a logical operation both in-memory and near-memory, simultaneously. While an in-memory analog computation is executed at the memristive crossbar, a near-memory digital logic operation is carried out at the periphery using conventional CMOS gates (see Figure 3.16). But rather than independently computing in parallel, the system executes the decomposed query[2] in a cascaded manner. At a given clock cycle, the control unit selects the crossbar rows that correspond to the questioned attributes and configures SA by selecting the appropriate reference current (switch S1), as instructed by the query operator. Subsequently, the logical results obtained at the SA output nodes are stored temporarily in a buffer. This will serve as the first input to the digital gate that can be either an OR or an AND gate, depending on switch S2 that enables the corresponding flip-flop (FF) and multiplexer (MUX) channel. The second input to the digital gate is the accumulated logical result of all the previously executed logical operations. The output of the digital gate will serve as the new intermittent result that gets buffered at the delay circuit. In the next clock cycle, this buffered signal gets to the gate input along with the new crossbar output. In other words, at every cycle, the digital output gets updated with the subsequent result of the logical operation obtained from the crossbar, until the query function gets fully executed.

To demonstrate the concept, we created an example based on the Cleveland heart disease database and the simulator presented in Section 3.4.2. The query comprises the AND of two OR operations (see Figure 3.17a). At the first clock cycle, rows 3 & 41 are biased with V_{Read}, and each column current is measured by an SA that is configured to perform the OR scouting logic operation. The result of OR is input to the digital AND gate. For the first cycle, the second input to the digital AND gate would be initialized to logical level 1. At the subsequent cycle, rows 1 and 2 are activated and their partial logical result (OR) is input to the digital AND gate along with the buffered OR result from cycle #1. The final query response is the binary vector that consists of the 303 logical results, as obtained from the output nodes of the digital gates right after the end of cycle #2.

[2] Query decomposition is its reformatting into the expression given by Equation 3.12

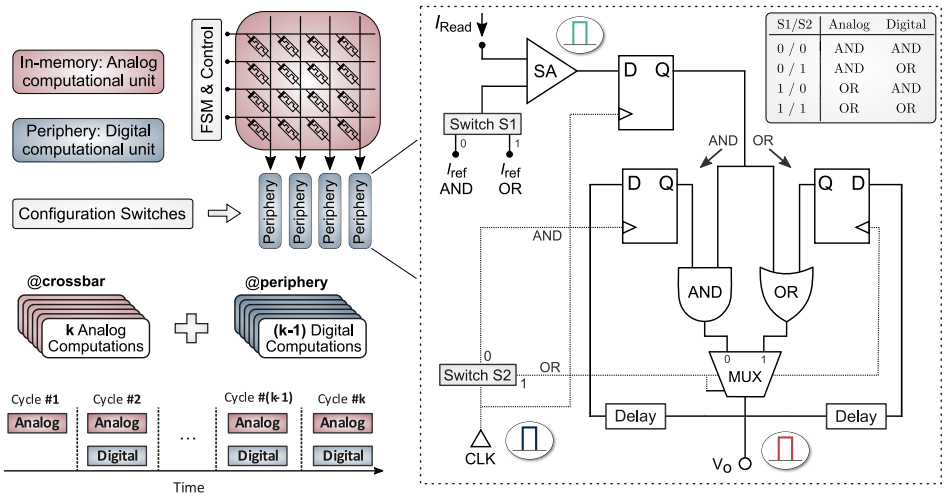
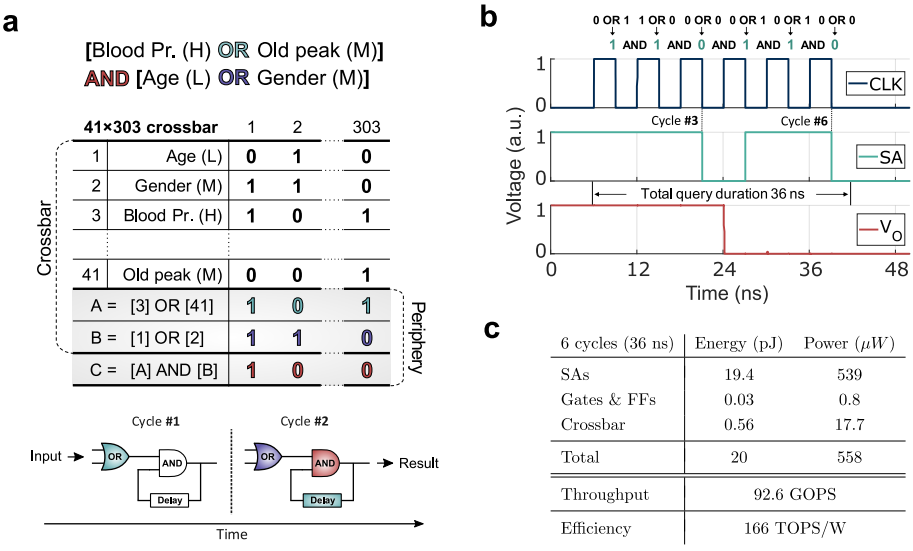


Figure 3.16: Schematic illustration of the cascaded system showing how the analog and digital computations are distributed to the crossbar and periphery, respectively. The digital circuitry design, located at each column node, is clock-governed (CLK) and consists of a sense amplifier (SA), CMOS logic gates, a multiplexer (MUX), and the switches that configure the circuit according to the query-instructed operation. SA converts the analog result from each crossbar column to digital and feeds it to the selected gate. There, it cascades with the preceding logical products until the final result is calculated at V^O, after as many cycles as the analog operations.

Cascaded logic systems are expected to execute queries with remarkable efficiency, owing to their massive parallelism combined with no need for high-power device programming. The 41×303 crossbar simulator provided computational metrics based on the experimentally measured power consumption, on which one can add the cascaded logic digital circuit assembled with 65nm CMOS components. We used a current-latched SA realized on 65nm SRAM technology [169]. Even though the design offers delay times lower than 3ns, this period was doubled to cover additional delays that may occur from charging the routing wires and other parasitic capacitors. The simulated system was clocked at 167MHz and configured to solve product-of-sums expressions. We chose an example query comprising 11 consecutive OR and AND operations, out of which 6 are performed in the analog domain. A maximum of 2 operations can be executed at each clock cycle (6ns), meaning that the total required time is 36ns (see Figure 3.17b).

Simulated performance metrics reveal that both the PCM-based crossbar and the digital-logic circuitry (gates, FFs, MUX) have a very low energy impact compared to the 303 SA units that dominate the time and power consumption. The total average power demand of the system is 558μW and the total required energy for the fully cascaded query is 20pJ (3.3pJ/cycle). These numbers refer explicitly to the core components, meaning that the control unit and any post-processing circuits are excluded from the simulation. The achieved throughput was 92.6GOPS and the energy efficiency reached 166TOPS/W. The performance metrics are summarized in Figure 3.17c for the total system. Note that



6 cycles (36 ns)	Energy (pJ)	Power (μW)
SAs	19.4	539
Gates & FFs	0.03	0.8
Crossbar	0.56	17.7
Total	20	558
Throughput	92.6 GOPS	
Efficiency	166 TOPS/W	

Figure 3.17: Cascaded logic operations. (a) An example query with 3 operations applied to the 41×303 crossbar, that is programmed according to the heart-disease database. The final operation C = A AND B uses the partial results A, and B that are OR operations. *Bottom:* Schematic illustration of the cascaded logic system in a "product-of-sums" configuration that outputs the result after the end of cycle #2. (b) Simulated waveforms of the digital circuitry solving an 11-step cascaded database query. The 3 nodes correspond to one column periphery and their positions are marked in Figure 3.16. The final result appears at the V^O node after the end of the clock cycle #6. **c** Simulated computational metrics for solving an 11-step example query on the heart-disease database.

power consumption in such crossbar-based systems is not deterministic, but depends on both the stored database and the query itself. In this example, the percentage of devices in the RESET state at the queried rows was below the average of the database, leading to a fairly increased power consumption due to, overall, higher I_{Read} values (17.7μW), although it is still 30× lower than the one of SAs.

Power losses associated with the "scouting logic" biasing scheme for selector-less crossbars are remarkably low. The strict conditions for successful programming implied the use of highly conductive wires and the division of the crossbar into two sub-arrays (see Section 3.4.1). Consequently, the uncontrollable voltage gradients along the wires are kept at a low level and only 0.1% of the crossbar power is wasted at the grounded rows. However, 18% of the crossbar power is dissipated to the series resistors (3.2μW) that act only as current controllers during write operations. Therefore, besides the limitations in write operations, crossbars with selector devices would have offered only marginal benefits in the presented scouting logic application.

3.7. CONCLUSION

In this chapter, the concept of in-memory database query was presented where the database is stored in a dense array of memristive devices, and the queries are performed in place by employing in-memory logic operations. A 4×8 selector-less crossbar populated with Proj-PCM devices was fabricated and characterized. Subsequently, it was used to demonstrate experimentally the errorless execution of database queries based on the concept of scouting logic. To simulate array sizes that are realistic for real-world applications, a circuit model was developed in order to emulate the nonlinear electrical behavior and the non-volatility of Proj-PCM devices. The emulator was incorporated into a circuit simulator that successfully solved a database query problem using a real healthcare-related database.

In addition, we investigated the main operational challenges of selector-less crossbars used for logic applications and provided design guidelines. More specifically, quantitative studies were carried out on the array size limitations as a function of the wiring resistance, the inter-device conductance variations, and the difficulty of programming the target device without affecting the stored data in the rest of the crossbar. The proposed solutions were verified using either numerical or circuit simulations and included design guidelines that overcome the size and programming limitations by using special biasing schemes that minimize sneak paths, changing the aspect ratio of the crossbar, and splitting it into smaller segments along the rows. The problem of inter-device conductance variations was approached theoretically, with a statistical analysis that provides the most variability-tolerant values for the reference currents.

Moreover, the flexibility of in-memory logic was highlighted by showing the compatibility with alternative operational mechanisms, such as the majority logic that differs from scouting logic by not requiring configurable sense amplifiers. Finally, we introduced the concept of *cascaded logic*, a system capable of executing queries of arbitrary size and complexity. It is based on a convenient query decomposition into a multitude of alternating AND & OR operations and it combines in-memory logic using resistive crossbars with peripheral CMOS logic. This system managed to process database queries with massive parallelism, high throughput (92.6 GOPS), and low energy consumption (166 TOPS/W).

4

Robust Logic Accelerator using STT-MRAM

This chapter presents an STT-MRAM-based CIM architecture that performs accurate logic operations while addressing challenges related to device variations and non-idealities that typically narrow down the sensing margin and severely impact computing accuracy. In this work, we propose an adaptive referencing mechanism to improve the sensing margin of a CIM architecture for logic operations. We generate reference signals using multiple STT-MRAM devices and place them strategically into the array such that these signals can address the variations and trace the wire parasitics effectively. We have demonstrated this behavior using an STT-MRAM model, which is calibrated using 1Mbit characterized array. Results show that our proposed architecture for binary neural networks (BNN) achieves up to 17.8 TOPS/W on the MNIST dataset and 130× performance improvement for the text encryption compared to the software implementation on the Intel Haswell processor.

This chapter is based on [65]. This work is reported in collaboration with IMEC under an EU project. STT-MRAM device characterization and fabricated chip measurement results are provided by IMEC, while my contribution is the design and implementation of logic operations.

4.1. INTRODUCTION

CIM architectures are well-known for their massive parallelism and high energy efficiency and have been vastly explored to perform logic operations for applications such as database query, binary neural networks (BNN), and encryption [1]. Emerging STT-MRAM technology stores values in terms of resistance states, and the feature of their state dynamics makes it naturally suitable for the CIM architecture. These devices are non-volatile, compact, scalable, and compatible with CMOS technologies [170]. In addition, STT devices exhibit high endurance and allow fast and low-power read/write operations. As illustrated in Fig. 4.1a, in a CIM architecture, the storing devices are arranged in a crossbar structure, where logic operations are accelerated by leveraging circuit laws, such as Ohm's law and Kirchhoff's current law. However, current STT-MRAM-based CIM suffers from challenges related to small sense margins, which are due to device variations, low tunnel magnetoresistance (TMR) value [61], and wire parasitics that can severely impact the accuracy of the system.

Several prior reference schemes have focused on building STT-MRAM based reference circuits and dedicated sense amplifiers, as opposed to using a constant reference signal [55], to adapt to global device variations [108, 171–173]. However, most of these solutions neither consider STT-MRAM and CMOS process, voltage and temperature (PVT) variations, nor the RC delay mismatch to validate the computing accuracy; hence, they could be highly optimistic about design closures. Existing efforts that do consider these effects mainly focus only on read operations and do not implement CIM-based logic operations [61, 86, 113]. Moreover, prior work on STT-MRAM based CIM lacks silicon-driven investigations to accurately determine the impact of these non-idealities on computing accuracy. In summary, to demonstrate the potential and robustness of a real STT-MRAM based CIM architecture, there is a need for comprehensive investigation using silicon parameters.

In this chapter, we develop and validate a design methodology that improves the sens-

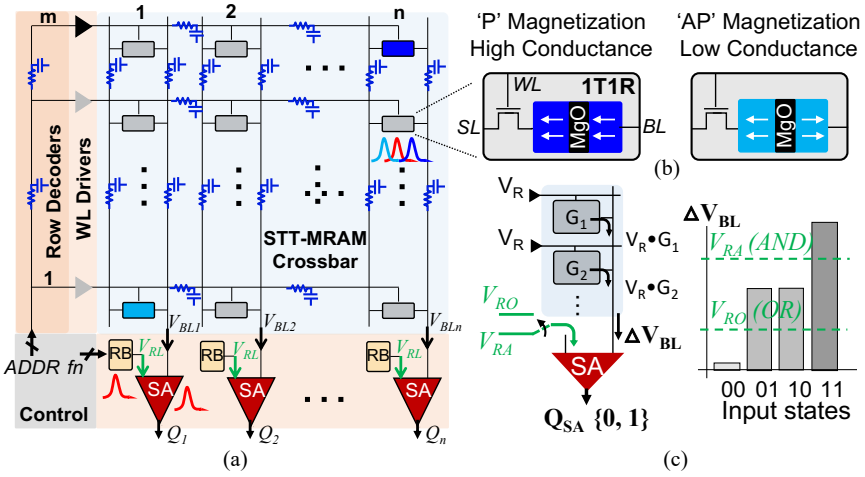


Figure 4.1: (a) CIM core and its non-idealities (b) STT-MRAM based 1T1R bitcells (c) Two-row read for CIM-based logic operations.

ing margins for robust CIM-based logic operations. The contributions of this chapter are:

- A novel referencing scheme to improve read margins where STT-MRAM devices are used in the reference cells that exhibit high tolerance against PVT variations.
- An approach to address RC delay mismatch where these reference cells are split into two sub-cells and placed strategically within the bitcell array that additionally, improves the performance of the logic operations.
- Integration and validation of our scheme on BNN and text-encryption, and analysis on the computing efficiency.

Our simulation results (based on parameters calibrated from an experimentally verified STT-MRAM 1Mbit characterized chip) show that we can achieve up to 17.8 TOPS/W on the MNIST dataset and 130× performance improvement compared to Intel Haswell processor-based CPU implementation.

4.2. DESIGN METHODOLOGY

Next, we present our proposed design methodology and implementation of STT-MRAM-based CIM for logic operations.

4.2.1. MOTIVATION AND APPROACH

Fig. 4.1a summarizes the non-idealities that adversely affect the sensing margins associated with the read and logic operations; i) Wire parasitics: RC delay mismatch along the critical path. ii) Systematic and random variations: in sense amplifiers (SA) and reference blocks (RB) generating reference signals. iii) PVT variations related to CMOS and STT-MRAM. The purpose is to develop a methodology that primarily focuses on generating reliable reference signals to provide and maintain high sensing margins in the presence of the aforementioned non-idealities. In our approach, we constrain the reference signal (reference line, with voltage signal V_{RL}) to experience similar non-idealities from which the input signal (bitline, with voltage V_{BL}) suffers, during a multi-row select read-assisted logic operation. To this end, we (a) build the reference circuits using a combination of STT-MRAM and CMOS devices to address PVT variations, and (b) integrate strategically the reference units within the bitcell array to tackle the increasing wire parasitics along the rows and columns.

4.2.2. REFERENCE GENERATORS

The required reference signal must differentiate the two critical resistance states (resulting in minimum margins) out of all the possible input states that determine the logic output. Moreover, the reference signal that falls in the *middle of these two critical states* indisputably offers maximum signal sensing margins. As shown in Fig. 4.2, this methodology has inspired many prior works to explore the generation of such reference signals for read operations using STT-MRAM devices [174].

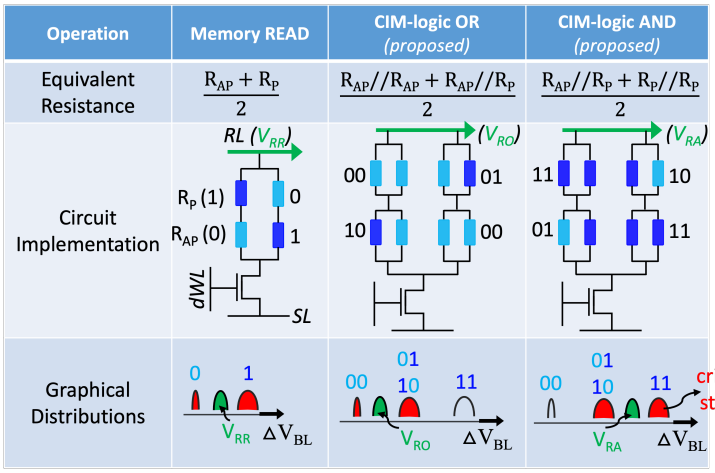


Figure 4.2: Reference for read, CIM-logic OR and CIM-logic AND operations.

To this end, we propose a scheme to arrange STT-MRAM devices in such a way that it produces an average equivalent resistance of the two critical resistance states for the desired operation. For instance, the OR reference falls in the middle of the two critical states 00 (i.e., both in 'AP' state with an equivalent resistance $\langle R_{AP}//R_{AP}\rangle$) and 01/10 (i.e., one in 'AP' and one in 'P' state with an equivalent resistance $\langle R_{AP}//R_P\rangle$). Hence the middle resistance is $\langle\frac{R_{AP}//R_{AP}+R_{AP}//R_P}{2}\rangle$. In a similar fashion, AND reference signal has an effective resistance of $\langle\frac{R_{AP}//R_P+R_P//R_P}{2}\rangle$. Fig. 4.2 shows the circuit implementations of the proposed reference generators and graphical distribution of input and reference signals V_{RO} and V_{RA} generated for the OR and AND operations, respectively.

4.2.3. REFERENCE ARRANGEMENT

In addition to PVT consideration, it is also important to account for the wire delay mismatch related to the position of the accessed bitcell in the crossbar array. For instance, as shown in Fig. 4.1a, the input signal V_{BL} while accessing the bitcell at position [1,1] (light blue) incurs by far less delay as compared with accessing the bitcell at [m,n] (dark blue). Note that the worst case takes place during a read operation when bitcell [1,1] is in the 'AP' state and bitcell [m,n] is in the 'P' state. Since the timing of the input and reference signals reaching the SA is critical, the reference signals must ensure enough signal margins including these worst-case scenarios. The multi-row read approach further complicates the delay mismatch. Additionally, in terms of different columns as shown in Fig. 4.1a, an accessed bitcell close to the wordline (WL) driver strongly enables pass transistor (NMOS, with high V_{GS}) compared to a cell far from the driver.

Therefore, we propose a scheme of placing the reference generators in such as way that the reference signals capture similar parasitic delays as those experienced by the aforementioned worst-case bitcell accesses. Hence, the reference cells (e.g., AND operation) as described earlier, are split into two sub-cells, each with a resistance equivalent to the sum of the critical resistance states. These sub-cells are placed as the top Ref_{AT} and the bottom Ref_{AB} cells in each column, as shown in Fig. 4.3. This solves three purposes:

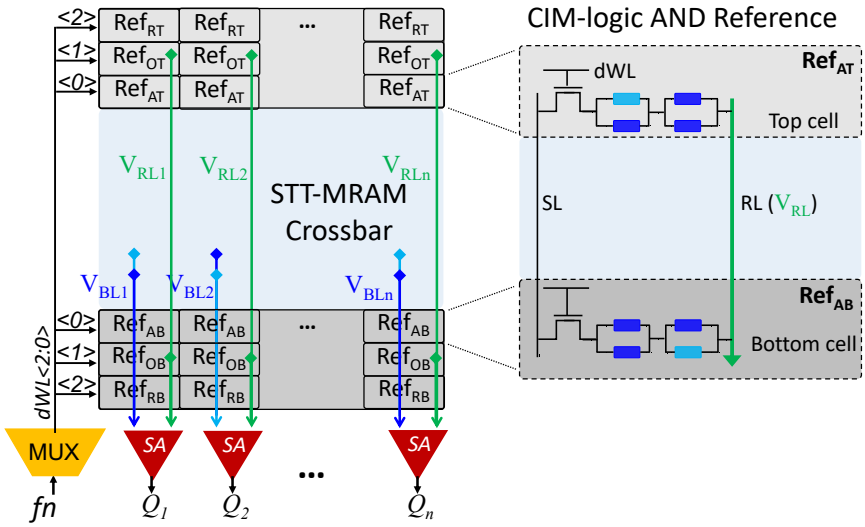


Figure 4.3: Proposed reference cells arrangement for read and CIM-based logic operations, and symmetrical reference sub-cell configurations.

1) The parallel combination of the two reference sub-cells effectively ensures an average resistance of the two critical resistance states of AND logic operation. 2) The placement at the top and bottom ensures row-wise tracking as the reference signal encounters an average delay of the worst-case bitcell access delays. (3) Placed in each column, these cells ensure column-wise tracking since they suffer from similar WL voltage degradation as experienced by the accessed bitcells. In a similar way, the read reference circuit is split to Ref_{RT} and Ref_{RB}, and CIM-logic OR to Ref_{OT} and Ref_{OB}.

4.3. RESULTS

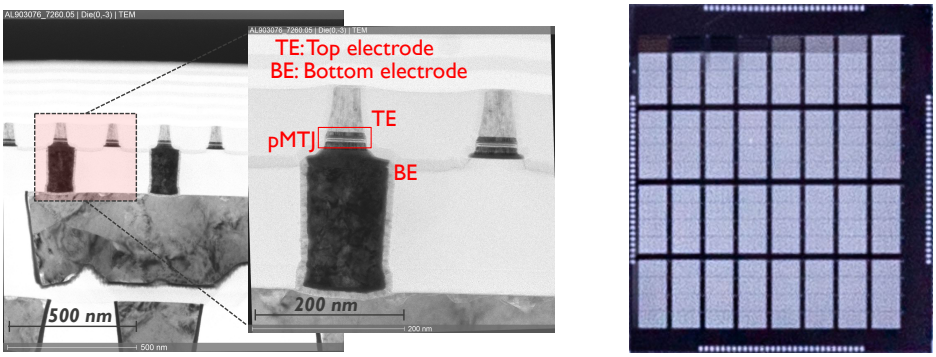
4.3.1. CHIP PROTOTYPE AND EXPERIMENTAL SETUP

Fig. 4.4a shows the microscopic view of CoFeB-based perpendicular magnetic tunnel junction (pMTJ) device [175]. Fig. 4.4b shows optical images of the fabricated characterization chip prototype which is experimentally verified for memory operations [175]. The STT-MRAM device model is calibrated using a characterization array of 4Gbit pMTJ devices, out of which 1Mbit pMTJ are electrically active. Detailed design specifications are summarized in Table 4.1.

4.3.2. CIRCUIT-LEVEL SIMULATION RESULTS

EVALUATION OF WIRE PARASITICS

Fig. 4.5 highlights the reduced read latency (associated with the required BL and RL discharge times) for four different CIM configurations; The reference block is placed at the 'top' (above the farthest row to the SA), 'bottom' (the nearest), at the 'centre' of the two equally split sub-arrays and the proposed 'split' configuration (two sub-cells at the top



(a) (b)

Figure 4.4: (a) Microscopic view of CoFeB based pMTJ. (b) Optical image of the fabricated chip depicting 4×8 (32) memory banks.

Parameters	Specifications
Memory, Banks, Array	1 *M*b, 32, 128 × 512
SA pitch, min. sensing margin	16 bitcells, 40mV
STT Device	*CoFeB*-based pMTJ [175]
Voltage Read/Write* (variations)	0.75 V/1.1 V, 0.9 V (±10 %)
CMOS (variations)	RVT, 28 *n*m TSMC (3σ)
Temperature	-40°C to 125°C

Table 4.1: Design parameters. *Separate core WL, and periphery voltages.

and bottom) as shown in Fig. 4.5a. Fig. 4.5b shows the spread of V_{BL} associated with Read1 and Read0 at the worst-case bitcell accesses (top-most, Row128) and (bottom-most, Row1) in the 'split' configuration. The worst-case signal margin arises between the cases Read1 at Row128 and Read0 at Row1. Fig. 4.5c shows that while each referencing scheme ensures that reference signals have an average resistance state of the critical states, the average is calculated using effective resistance, for instance, in the 'top' configuration, when the top bitcell is accessed. This adversely affects the time required to meet the minimum sensing margins for the bottom (the other worst case) bitcell access. A similar argument can be made for the 'bottom' configuration. In the 'split' configuration, the cells inherently achieve the desired average resistance while taking the top and the bottom worst cases into account. The 'centre' configuration exhibits similar considerations, however, since it affects the symmetry of the bitcell array, it is not considered in our methodology.

Fig. 4.6a shows the effect of degradation of the WL reaching the bitcell of the same row but with increasing column positions. A constant reference signal (V_{RO} or V_{RA}) which is generated independently and unaware of the column position fails to distinguish between the critical states. Our proposed scheme of placing the reference circuits along the row is adaptive to these undesired voltage drops and therefore, the reference signals ($^*V_{RO}$ or $^*V_{RA}$) effectively stay near the middle of the critical states (refer to Fig. 4.3). In

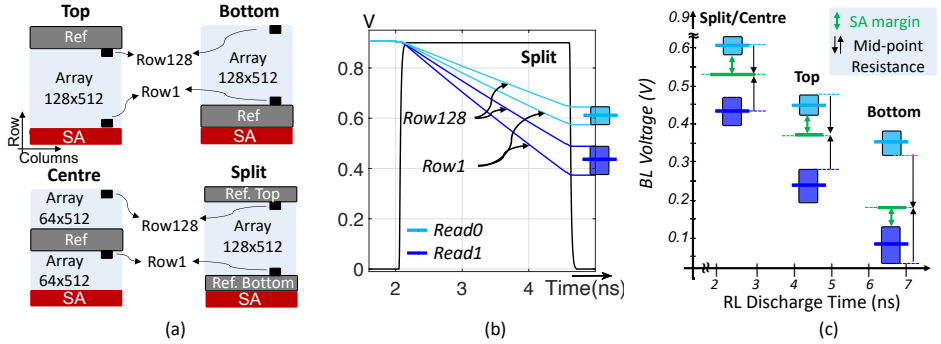


Figure 4.5: (a) Different reference arrangements. (b) Variation in V_{BL} for topmost and bottom-most bitcell accesses. (c) Latency comparison.

summary, the proposed 'split' configuration improves the performance and ensures high margins for high robustness and scalability.

EVALUATION OF PVT VARIATIONS

The validation and performance of the read operation for the global PVT (corner) cases, each with 3σ local variations, are presented in Fig. 4.6b. The corner cases are represented by [process, voltage, temperature] (including CMOS and STT-MRAM variations) along with the corresponding worst-case read latency. The spread of V_{BL} associated with the Read1, Read0, and spread of V_{RL} associated with the reference signal for each of these corner cases are shown in the figure. Slow corner understandably has a larger spread (more variations) and hence, the required minimum SA read sensing margin is delayed. The reference signals incur a smaller spread compared to the read signals since the reference signal is generated using four devices (two 'P' and two 'AP' states) where the individual spread of the STT-MRAM devices is averaged out. In short, in each of these corner cases, the reference signal is able to distinguish the input states, however, with different timing requirements.

In a similar way, Fig. 4.6c presents the global PVT (corner) cases, along with their 3σ

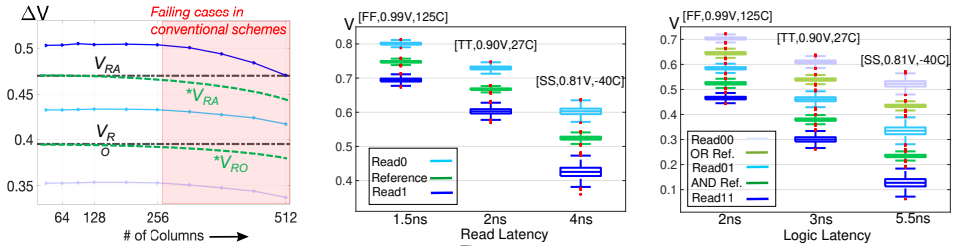


Figure 4.6: (a) ΔV $(VDD - V_{BL})$ and $(VDD - V_{RL})$ degradation due to WL degradation with column index. Our proposed scheme maintains high margins in the error-prone red area. Sensing margins and latency for (b) read and (c) logic operations with worst-case PVT analysis.

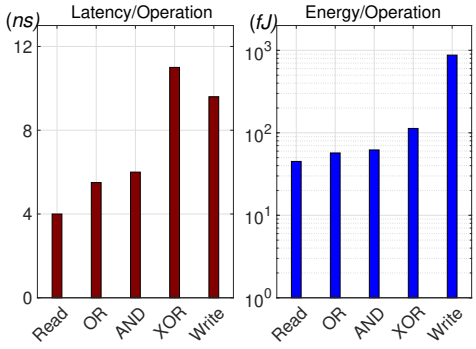


Figure 4.7: Latency, energy per operation.

local variations, for OR and AND operations. The sensing margins associated with the AND operation are smaller compared to that in the OR operation when identical time is invested in the two operations. Although the spread (variations) of STT-MRAM devices in the 'AP' state are larger compared to devices in the 'P' state, the inherently smaller ratio of the critical state resistances in the AND operation concedes to smaller sensing margins.

In summary, our methodology provides high robustness for read and logic operations in the presence of PVT variations.

DESIGN EFFICIENCY PER OPERATION

The worst-case latency and average energy per operation are presented in Fig. 4.7. For logic operations, XOR is the most time and energy-consuming one, because it involves logical combinations of simpler AND and OR operations. Write operation is ~10 *n*s, however, it consumes nearly 10× more energy (~900 *f*J) compared to logic operations (~70-110 *f*J) due to high compliance currents.

4.3.3. SYSTEM-LEVEL RESULTS

We evaluate the benefit of our STT-MRAM-based CIM design on two applications, namely binary neural network (BNN) and text encryption. Following is the brief summary:

BNN: BNN is an efficient way of implementing NN on low-power embedded platforms, as it converts float/integer values into binarized weights and neuron activation. BNN used for our evaluation has two hidden layers with 1024 neurons each and takes binarized 28 × 28 input image from the MNIST dataset. We program the input vector of each layer on the topmost row and the weights to the rest of the rows in the crossbar. We perform XNOR operations inside the memory by pairing each weight with the input vector and the required post-processing at the periphery. The overhead of communication between the crossbar arrays is ignored. Results show that our design classifies one input image in ~47 *n*s and consumes ~211 *n*J. It utilizes 36 CIM crossbars, each of size 128 × 512, and delivers 98% hardware accuracy. Compared to the CIM baseline with similar accuracy (without the proposed scheme), our implementation achieves 1.8× better performance and energy efficiency.

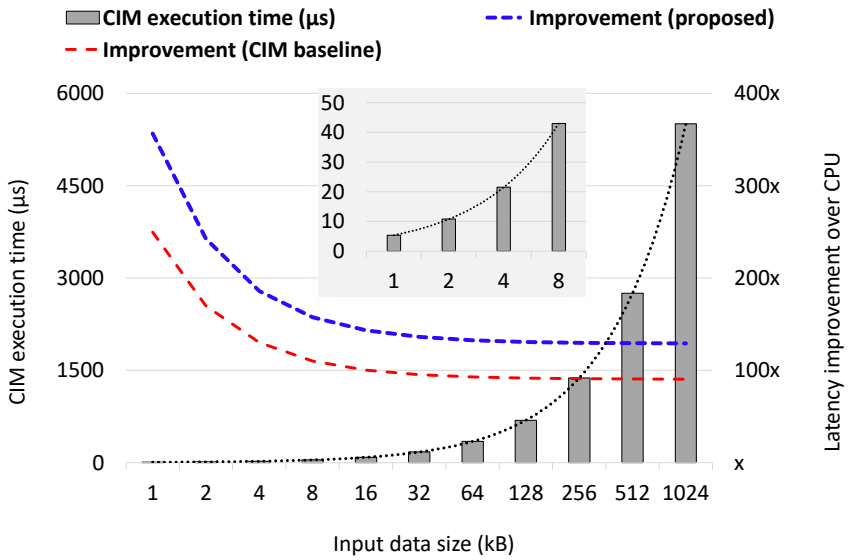


Figure 4.8: Execution time to encrypt different input text sizes and the improvement achieved compared to the software implementation

Processor	X86, out-of-order, 3.6 GHz
L1 cache	64 *k*B I-cache with 64 B D-cache
L2 cache	256 *k*B with 64 B cache line size
L3 cache	8 *M*B with 64 B cache line size
Main memory	DDR3 with 8 *G*B

Table 4.2: CPU simulation parameters

Text Encryption: An input text vector is encoded by performing a bitwise XOR operation with a predefined key vector. In our implementation, first, both the texts and the key are programmed to the crossbar. Second, the crossbar rows containing input text are activated sequentially while the row programmed with the key is active in all the steps. We assume each character has 8 bits implying that it has to be distributed over 8 cells. Fig. 4.8 shows that compared to the traditional software implementation on Intel Haswell processor using gem5 syscall emulation and CIM baseline (without the proposed scheme), we improve the performance by 130× and 1.4×, respectively. Table 4.2 lists the configuration of our CPU baseline architecture. In this simulation, we consider one crossbar which is reprogrammed with the incoming input text. It is clear that employing more crossbars would result in higher throughput improvements due to the parallelization.

4.4. CONCLUSION

This work presents a novel referencing scheme using STT-MRAM-based CIM to perform robust logic operations in the presence of design non-idealities. The chapter validates our proposed design using calibrated design parameters extracted from a silicon-verified characterization chip. Performance metrics are determined for logic operations which

are employed for a system-level framework and evaluated for BNN and text encryption applications. Results show that our implementation achieves up to 17.8 TOPS/W on the MNIST dataset and 130× performance *i.e.* execution time improvement is expected compared to software implementation on Intel Haswell processor.

5

REFERENCING-IN-ARRAY SCHEME

This chapter presents an RRAM-based CIM architecture that performs accurate multi-operand logic operations while addressing challenges related to device variations and non-idealities that typically narrow down the sensing margin and severely impact computing accuracy. In this regard, we propose a voltage-based differential referencing-in-array scheme that enables accurate two and multi-operand logic operations for RRAM-based CIM architecture. The scheme makes use of a 2T2R cell configuration to create a complementary bitcell structure that inherently acts also as a reference during the operation execution; this results in a high sensing margin. Moreover, the variation-sensitive multi-operand (N)AND operation is implemented using complementary-input (N)OR operation to further improve its accuracy. Simulation results for a post-layout extracted 512x512 (256Kb) RRAM-based CIM array show that up to 56 operands (N)OR/(N)AND operation can be accurately and reliably performed as opposed to a maximum of 4 operands supported by state-of-the-art solutions while offering up to 11.4X better energy-efficiency.

This chapter is based on [66].

5.1. INTRODUCTION

RRAM-based CIM architectures have been vastly explored to perform logic operations for applications related e.g., to artificial intelligence and big data in an energy-efficient manner [176]. However, RRAM devices suffer from non-idealities such as variations, resistance drift, and read disturb [1, 112]. These limitations, along with CMOS variations and wire parasitics, lead to inaccurate, unreliable, and energy-inefficient CIM-based logic operations. Therefore, energy-efficient circuit solutions that realize accurate and reliable CIM-based logic operations are required.

CIM-based logic operations are illustrated in Fig 5.1(a). The underlying concepts for read-assist logic designs are: 1) The interaction of voltage V_R and bitcell conductance G, in accordance with Ohm's law, resulting in the current $V_R \times G$ per bitcell; 2) Accumulation of these currents into a column current I_c, in accordance with Kirchhoff's current law; and 3) comparison of the current or voltage drop with an appropriate reference (selected by a MUX) using a customized SA. Therefore, logic operations can be performed simultaneously in all the activated columns and practically operate at O(1) time complexity, thereby achieving massive parallelism.

Several works on CIM-based logic operations have been reported, but they weakly address the aforementioned challenges. Such works can be classified into two classes: 1) non-stateful or read-assisted logic- where the RRAM state is unaltered. 2) stateful logic- where the RRAM state is altered in order to perform logic operations. Most of the non-stateful architectures are based on single-ended sensing mechanisms where multi-row (operands) read operations are performed using dedicated reference signals [106–110]; the signals enable the selection of the logic operations to be performed (e.g., (N)AND, (N)OR) and associated number of operands. However, this results in complex periphery circuitry with limitations. First, the sense amplifier (SA) employed incurs more area and consumes more energy as it needs to generate several reference schemes controlled using multi-MUXs [49, 64, 106–110, 113–115]. Second, due to the static nature of their refer-

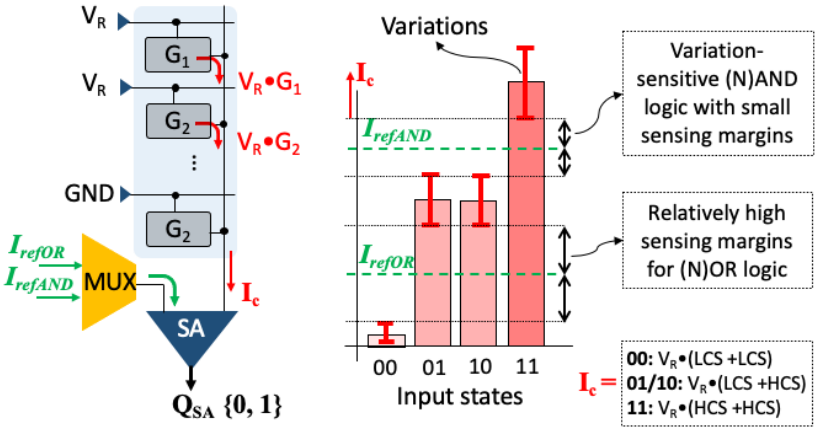


Figure 5.1: (a) Working of (N)AND, (N)OR operation; (b) Comparison of the relative sensing margins of (N)AND and (N)OR logic operation.

encing schemes, these architectures have a small read margin, especially in the presence of process variation and wire parasitics; this reduces the computing accuracy and limits the maximum number of operands. Third, as shown in Fig. 5.1(b), (N)OR and highly sensitive (N)AND operations are executed in a similar way, implying that the (N)AND operation becomes the bottleneck which significantly impacts the overall performance as well as limits the scalability of such CIM architectures [64, 177]. Fourth, supporting multi-operand operations further aggravates the above aforementioned challenges [49, 110]. Multi-operand can alternatively be performed by investing in aggregating multiple two-operand (N)AND or (N)OR operating cycles effectively in the periphery [64]. This, however, requires complex periphery circuits implying additional area and power consumption and reduced computational throughput. On top of that, an RRAM device also suffers from the accumulated effect of a large number of read operations that can lead to significant conductance change (conductance drift) or unwanted bit-flip (read disturb) [111, 112]. On the other hand, few non-stateful logic solutions have adopted differential sensing scheme [178–181]; however, the use of external references implies that they face the aforementioned issues as well. Other stateful logic solutions require programming of RRAM devices [79, 82, 83, 182, 183]; they result in a high energy consumption as well as reliability and endurance issues. In short, there is still a need for cost-effective circuit-level solutions that not only provide accurate and energy-efficient logic operations but also push the limit of the maximum allowed number of operands.

This chapter proposes a reference-in-array circuit-level scheme that accurately and reliably performs logic operations with up to 56 operands in a single cycle while realizing up to 11.4X energy efficiency compared to state-of-the-art CIM-based logic solutions. The key contributions of the chapter are:

- Introduces a differential sensing scheme by arranging cells in a complementary structure and a row of dummy bitcells in such a way that it inherently acts as a reference to enable a high sensing margin.
- Implements highly variation-sensitive (N)AND operation using complementary-input (N)OR operation to further enhance the accuracy and energy efficiency.
- Presents two design-for-reliability techniques to accurately perform up to 10^8 consecutive logic operations on the same operands or data bits.
- Improves the scalability of N(OR) operations in terms of crossbar size by implementing an inherently adaptive sensing scheme to track variations and wire parasitics.
- Reports simulation results and comprehensive comparison using a post-layout extracted 512x512 (256Kb) RRAM-based CIM array.

The rest of the chapter is organized as follows. Section II presents the proposed scheme, followed by results in Section III. Finally, Section IV concludes the chapter.

5.2. PROPOSED CIM ACCELERATOR

Here, we first present an overview of our proposed scheme, followed by its implementation details.

5.2.1. OVERVIEW

We use a bitcell design which has two transistors and two RRAM devices (2T2R), arranged in such a way that the two RRAM devices always have the opposite values as shown in Fig. 5.2(a); the complementary cell structure enables the use of differential sensing, resulting in a high sense margin (~2x compared to single-ended sensing). We also realize NAND operations using NOR logic design by exploiting De-Morgan's law[1]; this completely eliminates the sensing bottleneck of NAND operations. We introduce an extra dummy cell per column that acts as a reference to perform logic operations in a differential manner. The resulting scheme does not only deal with the process variation naturally, but it also simplifies the overall sensing mechanism as our scheme does not need any external reference signals or any other controlling circuits.

5.2.2. PROPOSED CIM ARCHITECTURE

Next, the different aspects of the proposed architectures are explained.

CELL STRUCTURE

The proposed architecture is based on 2T2R bitcell configuration as shown in Fig. 5.2(a). Data-bit '1' is represented by dark blue (HCS) and the complementary data-bit as light blue (LCS), and the pass transistors connect these RRAMs to bitline (BL) and negative bitline (NBL). The bitcell has a wordline (WL) and a common select line (CSL), connecting the top electrode of one RRAM to the bottom electrode of the other. Fig. 5.2(a) shows that programming such a bitcell is not different from programming a 1T1R bitcell, where CSL is connected to GND and both bitlines are supplied with the write voltages (V_W) for SET and vice-versa for RESET. Also, the figure shows the current-voltage (IV) characteristics of an RRAM device.

MEMORY SINGLE READ OPERATIONS

Fig 5.2(b) shows how a read operation is performed. The bitline pair is pre-charged to the voltage supply (VDD) and CSL is connected to the ground (GND) before the start of the cycle. WL activation enables the bitlines to discharge through complementary RRAM devices, creating a differential BL/NBL voltage ($\Delta V = V_{BL} - V_{NBL}$) which is sensed using a SA. We use cross-coupled SA that involves a positive feedback loop to amplify the input differential voltage.

TWO-OPERAND LOGIC OPERATIONS

Fig. 5.2(c) illustrates logic operations performed using a dummy row of 2T2R bitcells; the top part of the figure illustrates the NOR operation and the bottom part the NAND

[1] De-Morgan's law states that the NAND gate is equivalent to an OR gate with inverted inputs

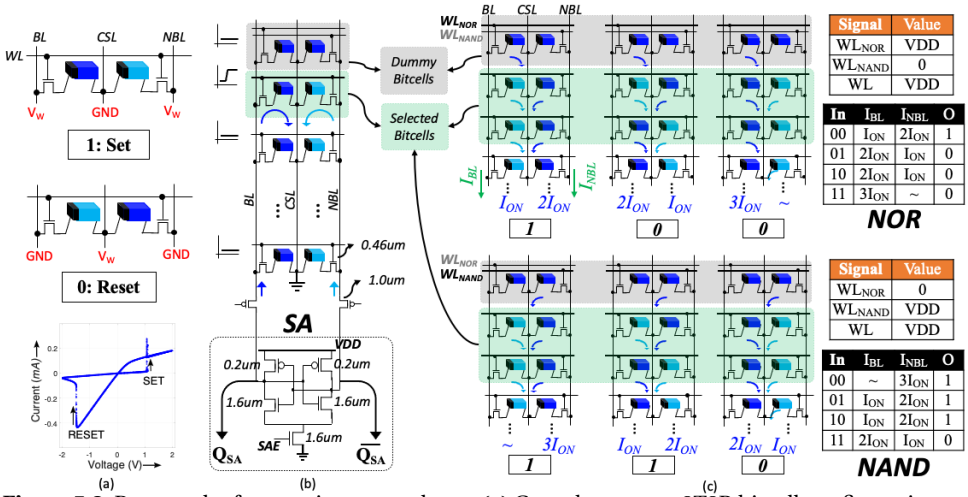


Figure 5.2: Proposed reference-in-array scheme (a) Complementary 2T2R bitcell configuration and associated programming mechanism (b) Read operation performed in a column within the CIM core. (c) Working of the proposed two-operand NOR and NAND logic designs.

operation. The idea is to bias the BL(NBL) side with an ON current while performing a NOR(NAND) operation. The dummy cells require two modifications; they need a) to have both RRAM devices of the cell in the HCS, and 2) to get two independent WLs, namely WL_{NOR} and WL_{NAND}, connected to the BL and NBL-sided pass transistors, respectively. The one-time configuration of dummy cells requires individual RRAM device programming to HCS by selecting the dedicated WLs one at a time. To perform a NOR function, two rows storing operands are activated along with WL_{NOR} in the dummy row. For simplicity, we assume ON current to be I_{ON}, and OFF current to be zero. In case both operands are in the RESET state (00); the BL/NBL discharge currents are $I_{ON}/2I_{ON}$, resulting in '1' as NOR result. In case the two operands are in different states (01/10); the BL/NBL discharge currents are $2I_{ON}/I_{ON}$, implying a value '0'. In case both operands are in the SET state (11); the BL/NBL discharge currents are $3I_{ON}/0$, leading to a value '0'. In a similar manner, the distribution of BL/NBL currents for NAND can be derived for each of the above cases; the results are shown in the table included in the bottom part of Fig. 5.2(c).

Fig. 5.3a shows the simulation results of our two-operand NAND and compares it with the conventional approach based on 1T1R and sensing mechanism with a fixed reference[106–108]. In the figure, WL_{conv} represents the array wordline timing used for the conventional case; while WL is the wordline behavior of one of the accessed cells (operands), and WL_{NAND} the wordline of the dummy cell of our scheme (see Fig. 5.2(b)). The operating cycle time depends on the time needed to develop the required ΔV on the bitline pair (assuming ΔV=40mV) to accurately differentiate the operand states. For the conventional scheme, the figure shows the BL discharge for cases 01 and 11, as well as the reference (BL_{ref}); note that a total of ΔV=80mV is generated between the 01 and 11 states. On the other hand, our proposal (making use of a differential self-referencing

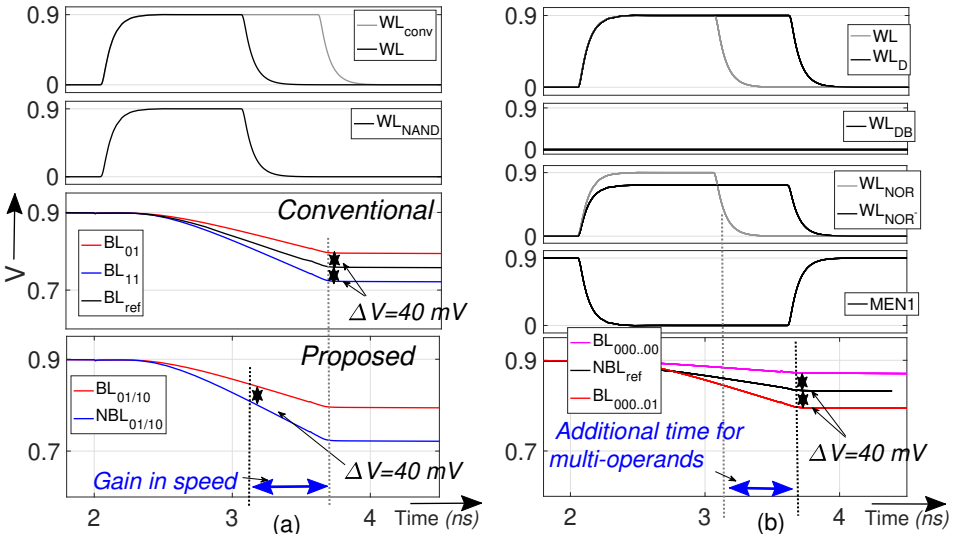


Figure 5.3: Timing diagrams (a) to illustrate performance gain for two-operand NAND operation (b) multi-operand NOR operation.

scheme) takes much less time to generate the required ΔV=40mV between BL and NBL; hence, enabling not only wide margins but also fast operations.

MULTI-OPERAND LOGIC OPERATIONS

Fig. 5.4(a) illustrates the multi-operand NOR and NAND operations. Each 2T2R array bitcell has now two independent WLs; one used for NOR operations and acts on data (WL_D), one used for NAND operations and acts on complementary data (WL_{DB}). During the NOR multi-operand operation, BL is discharged through the RRAM devices on the left side of the column (storing data), while during the NAND operation, NBL is discharged through the RRAM devices on the right side of the column (storing complement data). The dummy bitcell (per column) is used to enable the use of differential SA; it provides the reference current I_{ref}, and it has also two separate WLs; one selected during NOR operation and one during NAND operation. During the NOR operation, the BL/NBL = BL/I_{ref} discharge currents depend on the state of the selected operands. The value of I_{ref} has to guarantee the correct NOR operation for all states including the states with minimum difference in the BL discharge currents; these states are "all RESET (0)" (resulting in a BL discharge current of $n * I_{OFF}$) and "one SET (1) and $n-1$ RESET" (resulting in a BL discharge current of I_{ON}+$(n-1)$*I_{OFF}). Hence, I_{ref} has to be ideally at the middle of these two discharging currents; if we assume that $I_{ON}>>I_{OFF}$, then ideally I_{ref} should be about $I_{ON}/2$.

The dummy bitcell is used to generate I_{ref}; it is controlled with a dummy wordline WL_{NOR^-} driven by reduced voltage levels. Such a voltage can be generated using the PMOS-based bleeder circuit shown in Fig. 5.4(b). The PMOS devices enabled by MEN1 and MEN2 are used to provide the flexibility of generating different values of WL_{NOR^-} (hence of I_{ref}). Therefore, some calibration can be done if needed. The configuration of

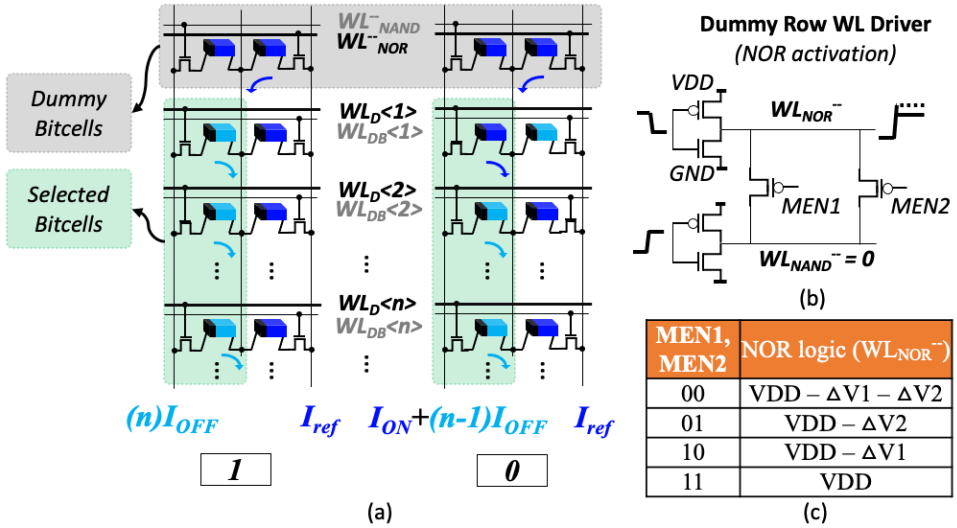


Figure 5.4: (a) Proposed multi-operand NOR operation. (b) Modified dummy WL driver to degrade WL_{NOR^-} or $WL_{NAND_}$ signal, and (c) Configurable MEN1, MEN2 switches and associated WL_{NOR^-}.

Fig. 5.4(b) allows the generation of three reduced voltage levels.

A similar analogy like the above can be used for multi-operand NAND. In this case, the WL_{NAND^-} should be driven with a reduced voltage ensuring the discharging current of NBL= I_{ref}, which has to be around $I_{ON}/2$ as well. During this operation, the difference in discharge currents BL/NBL= I_{ref}/NBL will ensure the correct operation through the SA.

Fig. 5.3b illustrates the working of 10-operand NOR operation and shows that additional time it required, as compared to the two-operand NOR operation of Fig. 5.3a. In the figure the same signal naming is used as that of Fig. 5.4(a). To show the comparison with two operands NOR operations, WL (representing one of the accessed bitcell wordlines) and WL_{NOR} (representing the dummy row wordline) for this two-operand operation are included. The results are shown for two cases resulting in minimum discharge of the BL: a) all the operands are set to 0 (00..00), b) all operands as set to 0 except one operand is set to 1 (00..01). To reach the required minimum ΔV=40mV between BL and NBL_{ref}, the duration of WL_D should be extended to give enough time for BL discharging though dummy cell selected by WL_{NOR^-}. Hence, increasing the number of operands to 10 while realizing robust operation comes at the cost of additional latency of 40%.

To ensure accurate execution of multi-operand operations in large-sized crossbars, both the *number* of operands and the *RC parasitic* of the wordlines have to be taken into consideration. The higher the number of operands, the smaller the sensing margin for a fixed wordline duration; the farther the bitcell accessed by the wordline signal, the larger the wordline delay and voltage degradation reaching that bitcell, and hence, the slower the corresponding bitline discharge. The cumulative effect is the degradation of signal

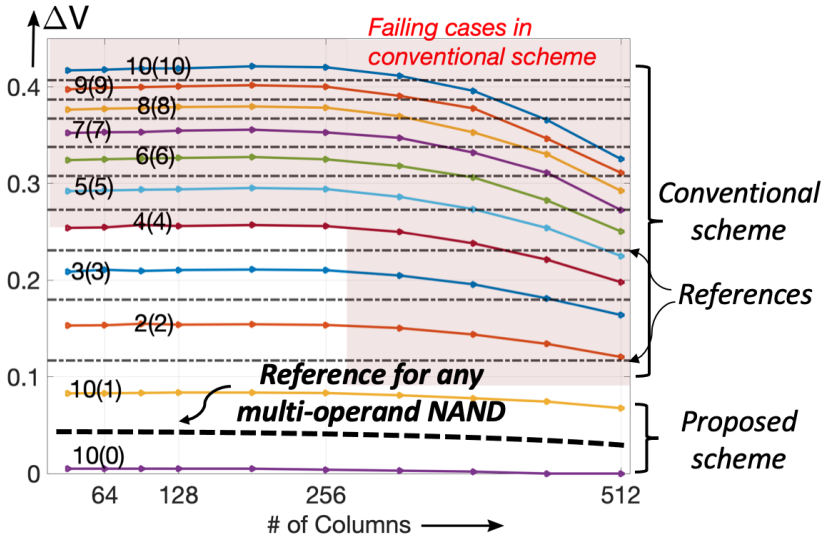


Figure 5.5: (a) Reduced signal margins due to operand size increase and RC parasitics. Proposed scheme to circumvent the erroneous region.

5

margins.

Fig. 5.5 shows the worst case sensing margins ΔV developed for multi-operand NAND operation (for the conventional scheme as well as for our proposed scheme) operating at a fixed wordline duration where the operations are performed for different columns (from 32 to 512), and by selecting the appropriate values for the operands resulting the minimum sensing margin. The notation $m(opr)$ denotes m-multi operand operation where opr operands are set to 1. E.g., 10(10) denotes the 10 multi-operand operations where all operands are set to 1. For the conventional approach, $3 \leq m \leq 10$, and the the interleaved horizontal lines are the corresponding reference signals for which changes depending on m. It can be seen that the reference signals in the conventional scheme fail to differentiate the operand states beyond a certain number of operands ($m > 4$) and/or beyond a certain column (which is m dependent). Hence, in these cases, the multi-operand operation fails (shown in light red color in the figure). On the other hand in our proposed scheme, the reference is fixed irrespective of the number of operands; it is set to the middle of discharging voltages of two cases: 10(0) and 10(1), meaning 10 multi-operand NAND operation where none of the operands is set to 1 (i.e., all in RESET) and only one operand is set to 1, respectively. Clearly, our proposal enables large m even beyond 10. Moreover, as the dummy row generating the references experiences similar degradation as the accessed operands, this can reliably provide undeterred sensing margins. In summary, significant operand scalability improvement for logic operations is achieved.

5.2.3. DESIGN OPTIMIZATIONS

RRAM devices suffer from device degradation, resistance drift, and eventual unwanted bit-flip. These are accelerated when the RRAMs are under voltage stress; the higher

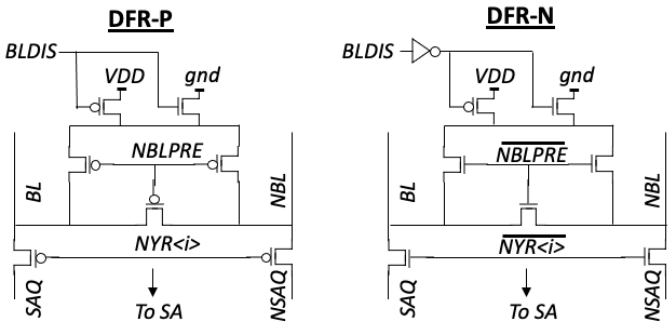


Figure 5.6: Design for reliability based on PMOS and NMOS device.

the voltage, the faster the degradation [112]. To suppress and/or to slow the degradation, one can reduce the voltage of the bit line precharge from VDD to a lower level, resulting in a small voltage ($V_{BL/NBL} - V_{CSL}$) across RRAM devices (see Fig. 3b). Fig. 5.6 presents two possible circuit solutions to generate such reduced precharge voltage using a VDD voltage supply: one based on a PMOS device (DFR-P) and one based on an NMOS device (DFR-N). For DFR-P, the bitline pair is precharged to VDD, then discharged through PMOS to VSS+VthP; while for DFR-N the bitline pair is pre-discharged to VSS, then charged through NMOS to VSS-VthN (VthP and VthN are PMOS and NMOS voltage thresholds, respectively). For both designs, BLDIS signal remains active while keeping NBLPRE=0 for a configurable amount of time to ensure equal voltages at the bitline pair before the start of the active cycle. NYR signal connects the bitline pair to the SA.

5.3. SIMULATION SETUP & RESULTS

5.3.1. SETUP

Table 5.1 summarizes the design specifications used for our circuit-level analysis. SA is designed to accurately sense a minimum ΔV of 40mV. Fig. 5.7 shows the layout view of the HfO_2/TiO_x RRAM-based conventional 1T1R bitcells and our 2T2R bitcell. Pass transistor (NMOS) is 540 nm/40 nm and it typically provides a resistance of 1.3 $K\Omega$. Vertical (Horizontal) wire resistance adds up to 0.4 Ω (0.8 Ω) and total capacitance adds up to 0.3 fF (0.6 fF) per unit bitcell.

Parameters	Specifications
CIM Array	512x512 (256Kb)
RRAM Device	HfO_2/TiO_x [112]
HCS/LCS	100 $K\Omega$ / 3 $K\Omega$
RRAM Variation	20 % parametric
Voltage supply	0.9 V with ±10 % variations
CMOS (variations)	SVT, 40 nm TSMC (3σ)
Temperature	-40°C to 125°C

Table 5.1: Design parameters.

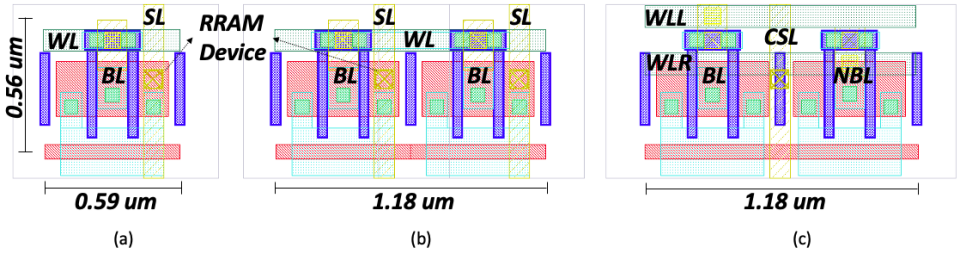


Figure 5.7: Layout footprint for (a) conventional 1T1R bitcell (b) two 1T1R bitcells and (c) 2T2R bitcell in our proposed scheme.

The simulation and comparisons with the conventional schemes are extracted using the same setup; e.g., technology node, SA design, RRAM bitcell, and wire parasitics. In this way, all penalties are considered in terms of latency (power is assumed to be nearly the same).

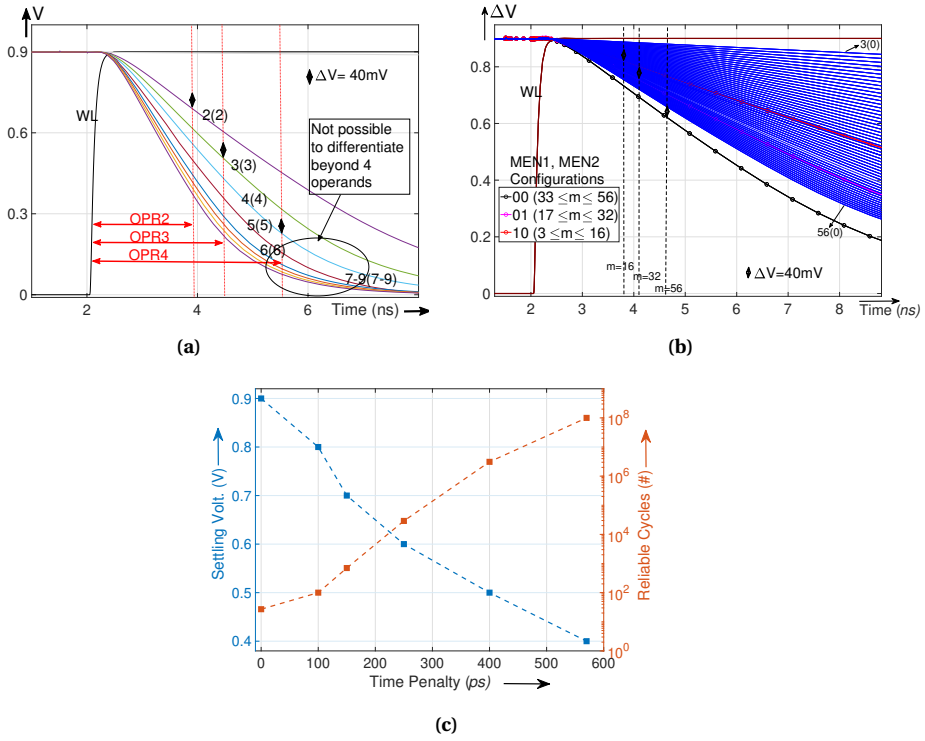


Figure 5.8: (a) BL discharge behavior for increasing number of NAND operands for conventional scheme (b) Multi-operand operations up to 56 operands can be performed by configuring MEN1 and MEN2 signals accordingly. (c) Reliable read cycles due to settling BL/NBL voltages associated with timing penalty using DFR-P design.

5.3.2. CIRCUIT-LEVEL RESULTS AND COMPARISONS

Fig. 5.8a shows the way the bitline is discharged for a m-operand NAND operation for $2 \leq m \leq 9$ where they all are in SET state (worst case); ΔV=40mV is the minimum sensing margin required by the SA (the reference lines are not included in the figure for clarity). Note that reaching the required sensing margin becomes impossible after 4 operands irrespective of additional discharge time. Hence, conventional schemes can support only a limited number of operands.

Fig. 5.8b illustrates how our proposed scheme can make use of signals MEN1 and MEN2 (see Fig. 5b) to better configure I_{ref} in order to maximize the number of operands m for NAND operation; the three configurable reference signals ensure $\Delta V = 40mV$ signal margin for the worst case scenarios; for our scheme, this is the case when all operands are 0 (RESET), which is the opposite of the conventional approach. The figure shows the discharge of the bitline for $3 \leq m \leq 52$; the wordline is activated for a long duration to capture different timing requirements to perform operations up to 56 operands. Note that the maximum achievable m is limited by the fact that a RESET state has a finite bitline current. Therefore, a higher HCS/LCS ratio can enable more number of operands.

Fig. 5.8c shows the significant impact of using DFR-P of Fig. 7 in reducing the conductance drift and the probability of eventual bit-flip during consecutive multi-operand logic operations; this comes at the cost of timing penalty which depends on the targeted BL precharge (settling) voltage. The more timing penalty we tolerate, the slower the RRAM device degradation and the more consecutive reliable operations.

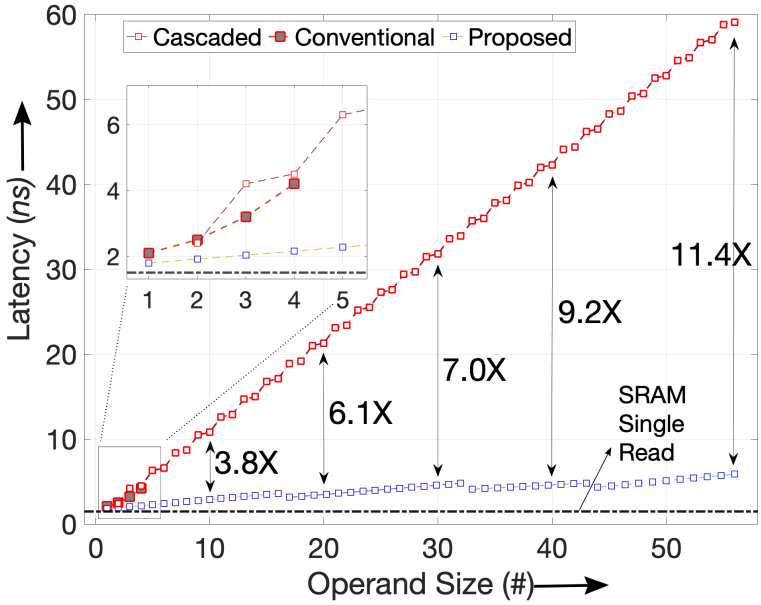


Figure 5.9: Comparison with an increasing number of operands (cascaded logic [64] and conventional solutions [106–110]).

Design Metrics	Conv. [106–110]	Cascade [64]	Proposed *(Standard)*	Proposed *(DFR-P)*
Max. Ops/cycle	4	2	56	17
Latency (ns)*	4.2/4.2	4.4/59.2	2.1/5.2	2.7/5.8
Energy (pJ)*	0.56/0.56	0.58/18.4	0.29/1.62	0.32/1.67
Energy/Op (fJ)*	140/140	145/380	72/29	80/29.8
Voltage supplies	2	2	1	1
Read Disturb Tol.	No	No	No	Yes
Variation Tol.	No	No	Yes	Yes

Table 5.2: Comparison of our proposed design with prior techniques. * indicates *four*/maximum operands supported. *Standard* and *DFR-P* are non-optimized and optimized designs for read disturb.

Fig. 5.9 presents the latency improvements compared to the cascaded logic design [64] with the number of operands given a maximum gain of 11.4X at an operand size of 56. Here, more the number of operands, higher is the gain.

Table 5.2 summarizes the results and shows the comparison with the state-of-the-art for different metrics. It is worth noting that although our 2T2R bitcell takes 2X area compared to the 1T1R bitcell, the complete CIM core of similar capacity is estimated to have 1.67X area owing to our smaller periphery.

5.4. CONCLUSION

This chapter has demonstrated how adding a dummy row in a computation-in-memory (CIM) crossbar based on RRAM can enable robust and reliable multi-operand bitwise logic operation. It does not only make the generation of references much simpler, but it also enables wider sensing margins. In addition, the chapter has shown how some basic circuits can be integrated with CIM to suppress/slow the degradation and boost the robustness of CIM logic operations while trading off some latency. Comparison results using 512x512 CIM core show that our proposed design offers an improvement of up to 11.4X in terms of energy efficiency while performing up to 56 operands in a single cycle.

6

ACCELERATING RRAM TESTING WITH CIM

This chapter explores CIM operations to accelerate the testing of RRAM-based high-density memory storage. The unique challenges related to the RRAM fabrication process render the traditional memory testing solutions inefficient and inadequate for high product quality. In this regard, this chapter presents low-cost design-for-testability (DFT) solutions that augment the testing process and improve fault coverage. A CIM-based DFT is realized to expedite the detection and diagnosis of faults by developing logic designs involving multi-row activation. A novel addressing scheme is introduced to facilitate the diagnosis of faults. Reconfigurable logic designs are developed to detect unique RRAM faults that offer features such as programmable reference generations, period, and voltage of operation. DFT implementations are validated on a post-layout extracted platform and testing sequences are introduced by incorporating the proposed DFTs. Results show that more than 2.3× speedup and better coverage are achieved with 6× area reduction when compared with state-of-the-art solutions.

This chapter is based on [67].

6.1. INTRODUCTION

RRAM is one of the most promising emerging device technologies that exhibit non-volatility, compatibility with CMOS, and high scalability, and can potentially replace conventional memories such as SRAM, DRAM, and flash [184]. In addition, these devices enable emerging energy-efficient CIM paradigms that allow computing within the memory units [49, 185]. However, manufacturing defects leading to unique faults have been identified in the RRAM production process when being integrated with CMOS [186]. Therefore, bringing RRAM to the market requires new dedicated test developments.

Traditional and well-established testing schemes for conventional memories are inadequate and cost-efficient for testing RRAMs [186–188]. Several existing efforts that have focused on RRAM testing can be typically classified into two broad classes: march test algorithms and design-for-testibility (DFT) solutions. Efforts offering modified march test algorithms that involve specific sequences of *sequential* memory (read, write) operations to optimize test time [189, 190] or improve coverage [191–194] fail in the detection of *unique* RRAM faults [187] and typically DFT schemes are introduced to further enable improved fault coverage (FC) and/or optimized test time. However, existing dedicated DFTs are expensive in terms of hardware [195], optimistic regarding variations and lack implementations [196], impractical due to large voltage requirements [188, 197, 198] and exhibit functional issues due to to the reliance on sneak-paths [189, 199]. DFTs that target high FC typically involve slow, probabilistic write operations [188, 200–204]. In summary, there is a conspicuous need for low-cost DFTs and new testing algorithms dedicated to RRAMs that can offer optimal high-quality test solutions.

In this chapter, we present low-cost DFT schemes by exploring reconfigurable CIM-based logic operations to improve the FC and accelerate the testing process of RRAMs. In our approach, multi-operand NOR logic facilitates the detection and diagnosis of unique RRAM faults with $\mathcal{O}(1)$ time complexity. The contributions of the chapter are:

- Proposes a DFT based on simultaneous access (read) of multiple bitcells in a multi-operand NOR logic of a varying number of operands to optimize RRAM test time. The scheme includes a customized sequence of address selection patterns that aid the diagnosis of faults.
- Realizes reconfigurable DFTs to improve the FC. We explore programmable reference signals, voltage, and duration of operations to develop high sensing margins that guarantee the detection of *unique* RRAM faults [187].
- Validates our compact DFT implementations with circuit-level simulations based on a post-layout netlist that facilitates the execution of high-quality test algorithms at a low cost.

A comprehensive simulation platform built on 40nm TSMC CMOS technology demonstrates the advantages of our proposed DFT schemes in terms of testing cost and FC. Comparison results with state-of-the-art solutions show that more than 2.3× speed and higher FC is achieved with 6× area reduction.

The organization of the chapter is as follows. Section 6.2 introduces RRAM technology and Section 6.3 covers the targeted faults. Section 6.4 presents our CIM-based DFT

and Section 6.5 presents our DFT schemes with additional reconfigurable features. Section 6.6 validates our DFT and develops testing algorithms. Section 6.7 provides a comparison with the prior arts. Section 6.8 reflects on the scalability of our schemes and future directions. Section 6.9 concludes the chapter.

6.2. HIGH-DENSITY RRAM

This section focuses on RRAM devices with binary storage capability (*bi-stable* element); however, this work can be easily extended to multi-level storage elements and other memristor technologies such as phase change memory.

6.2.1. RRAM TECHNOLOGY

An RRAM cell structure consists of a metallic oxide that is sandwiched between a Top (TE) and a Bottom Electrode (BE) as described in Fig. 6.1a [205]. The working principle of RRAM devices is based on the reversible formation of a conductive filament (CF) and the absence and presence of this CF delivers high (HRS) and low resistance states (LRS), respectively. This describes the analog nature of RRAM to realize different states. The switching from HRS to LRS is called 'SET', whereas that from LRS to HRS is called 'RESET'. Fig. 6.1b shows I-V characteristics during the SET and RESET operations with voltage applied in opposite polarities. The HRS and LRS represent the logic state *1* and *0*, respectively.

6.2.2. RRAM FOR MEMORY

A typical m×n RRAM architecture is shown in Fig. 6.2a. It consists of a bitcell array in a crossbar arrangement with periphery blocks such as WL drivers, address decoders, control blocks, sense amplifier (SA), reference block (RB), etc. It uses address *ADDR* and function *fn* to define the location and type of operation to be performed, respectively. The focus of this work is (but is not limited to) one-transistor-one-resistor (1T1R) bitcell configuration, since this is the most extensively explored configuration for high-speed memory [206] and for neural network realizations in a CIM architecture [207].

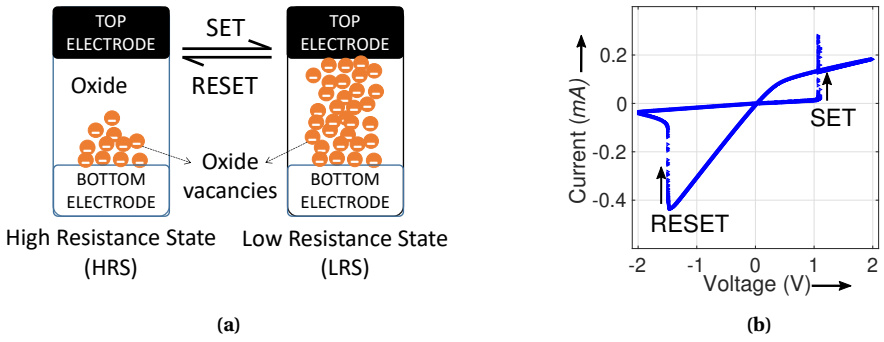


Figure 6.1: (a) Typical RRAM device and its (b) I-V characteristics.

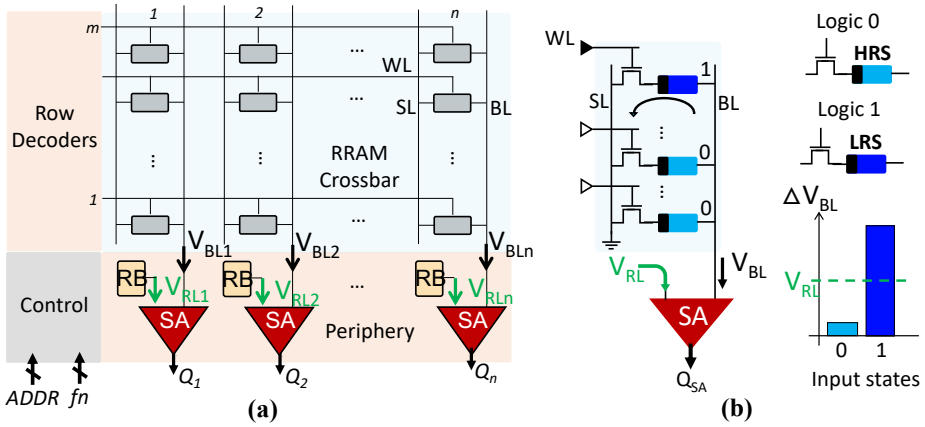


Figure 6.2: (a) RRAM-based memory. (b) Working principle of a typical read operation; dark (light) blue represents SET (RESET) RRAM state.

The 1T1R bitcell configuration is shown in Fig. 6.2b. The bitcell is accessed via the wordline (WL) that turns on the pass transistor (NMOS) while connecting the select line (SL) to the bitline (BL). In a write operation, BL is supplied with the write voltage and SL is connected to GND for the SET operation and vice-versa for the RESET operation. Fig. 6.2b shows how a read operation is performed. Before the start of the read cycle, BL is pre-charged to the voltage supply (VDD), and SL is connected to the ground (GND). WL activation selects a bitcell for reading that enables the BL to discharge through the RRAM device. The developed BL voltage V_{BL} is compared to a reference line voltage V_{RL}. The differential voltage $\Delta V = V_{BL} - V_{RL}$ is sensed using an SA to determine the state of the RRAM device.

6.3. TARGETED FAULTS

Manufacturing defects that lead to an erroneous behavior or a deviation from the intended behavior are modeled as faults. This section summarizes the observed faults in RRAMs [186–188, 190, 191, 208]. These faults can be classified into the following two classes: conventional faults and unique faults [187].

Conventional Faults are faults similar to those observed in traditional memories such as SRAM, and DRAMs; they are:

- Stuck-at-faults (SAF) [209]: An RRAM device is stuck in a state and cannot switch.
- Transition faults (TF) [209]: An RRAM device fails to switch properly.
- State coupling faults (CFst) [191, 202]: The state of an aggressor RRAM cell alters the state of the victim cell.
- Write disturbance faults (WDF) [191]: Unintentional alteration of the state of an RRAM device during a write operation.

- Incorrect Read faults (IRF) [200]: A read operation generates incorrect outputs whilst the state is correct.
- Read-disturb faults (RDF) [190]: A read operation switches the state of the RRAM device, while the read value is correct.

Unique Faults are faults emerging due to the nature of RRAM devices. Besides LRS and HRS, an RRAM device can also occupy an undefined (*U*) state [209]. A *U* can be defined as a state which, when read, can give random outputs because the ΔV developed is less than the minimum sensing margins ΔV_{min} required by the SA to define a deterministic state. Note that the detection of such faults *cannot* be guaranteed with a typical read operation. Other possible states are high (*H*) and low (*L*) states [189]; *i.e.,* deep states that lay beyond the resistance ranges of LRS and HRS. Resistance of *H* is more than the maximum HRS and that of *L* is less than the minimum LRS. Detecting faults due to such states *cannot* be guaranteed with existing march tests, since these tests only allow fixed, pre-determined patterns of *logic 1* or *logic 0* values corresponding to LRS and HRS, respectively. All unique RRAM faults are:

- Deep faults (DF) [189]: RRAM device falls into deep states.
- Undefined write faults (UWF) [209]: A write operation leads to *U* state.
- Unknown read faults (URF) [189]: A read operation switches the state of the RRAM device to *U* or/and produces random read outputs.
- Undefined coupling faults (CFud) [204]: The state of an aggressor RRAM cell alters the state of the victim cell to a *U* state.
- Intermittent undefined state faults (IUSF) [186]: RRAM state intermittently changes its switching mechanism from bipolar to complementary, which affects write operations and causes undefined state faults.

Depending on the efforts needed to detect the targeted faults, these can be classified into *easy-to-detect* (ETD) and *hard-to-detect* (HTD) faults [210]. Detection of ETD faults *can* be guaranteed using regular memory operations, whereas detection of HTD faults *cannot* be guaranteed using these operations. Therefore, special DFT schemes are required to *guarantee* the detection of HTD faults.

6.4. DFT SCHEMES PROPOSED FOR ETD FAULTS

This section presents DFT schemes to optimize the detection time of conventional ETD faults. These schemes are also the foundation of the DFT schemes to detect HTD faults.

6.4.1. CONCEPT

The DFT is based on performing *multi-operand* NOR logic operations within the memory unit that involves *multi-row* read operations. Such a memory unit that is modified to perform *in-situ* logic operations is referred to as a computation-in-memory (CIM)

unit [8, 66]. This allows the selection of multiple cells in parallel, and therefore, the execution of simultaneous read operations with N operands. The sensitization and detection of the faults by N sequential read operations can be accelerated by a factor of N by enabling simultaneous sensitization of the faults during an N-operand NOR operation. This logic operation can be represented by NOR^N and the proposed DFT can be referred to as DFT-NOR^N. Note that NOR^N can only replace *N read 0*, and not *N read 1* operations.

The working principle is illustrated using the detection of conventional SAF1 faults. Performing a correct fault-free NOR^N when all the bitcells are initialized to RESET (0) state returns *logic 1* as the output value; this is denoted as 0NOR^{N}1. However, if a faulty RRAM device is stuck at *logic 1*, NOR^N would result in an incorrect *logic 0*. To get a feeling of the potential speedup in RRAM testing, one can apply an appropriate march test algorithm while incorporating these logic operations as a detection sequence. Subsequently, N interleaving write and read 0 operations are replaced by N write operations and *one* NOR^N.

In addition to the detection of faulty behavior, faulty cell location can be identified by deploying a binary search algorithm. By performing NOR operations with varying numbers of operands while selecting certain patterns of rows in a binary search manner, the address selection based on logic outcomes converges to the faulty cell location in $\log_2 N$ (logic) + 1 (read) operations. In summary, the fault detection is accelerated by a factor of N for every read 0 operation and faulty-cell identification by a factor of $\frac{N}{log_2 N+1}$.

6.4.2. DFT DESIGN AND IMPLEMENTATION

The DFT scheme is based on using multiple references for single read and multi-operand NOR logic operations of varying numbers of operands. In addition, this scheme requires a modified row address decoder to enable simultaneous multiple-row activation. In this regard, we present a low-cost scalable multiple reference signals generator and row address decoder to facilitate the selection of multiple addresses.

REFERENCE SIGNALS GENERATOR

The multi-operand logic operation NOR^N creates N *parallel* and *independent* read paths. This implies that the total BL read current in such an operation is the aggregation of N bitcell currents. To perform a read or logic operation, the underlying concept is to generate a reference signal I_{ref} that guarantees the correct operation for all possible input states, including the *critical states* which are defined as the states with minimum $\Delta V = \Delta V_{min}$; ΔV_{min} is the minimum ΔV required by the SA. For a read operation, there are only two possible states; *i.e.*, RESET, and SET states. Hence, I_{ref} has to be ideally at the middle of these two discharging currents; if we assume SET state current to be I_{ON} and RESET current to be I_{OFF} and $I_{ON} >> I_{OFF}$, then ideally I_{ref} should be about $I_{ON}/2$. On the other hand, for a NOR^N operation, these critical states are: (1) *all* cells are *RESET (0)* denoted by N(0) that results in a BL discharge current of $N*I_{OFF}$, and (2) *one SET (1)* and *N-1 RESET* states denoted by N(1) that results in a BL discharge current of $I_{ON}+(N-1)*I_{OFF}$. Since these two states produce an accumulation of several I_{OFF} currents, the assumption of using $I_{ON}/2$ as a reference is not valid as N increases. Hence, an

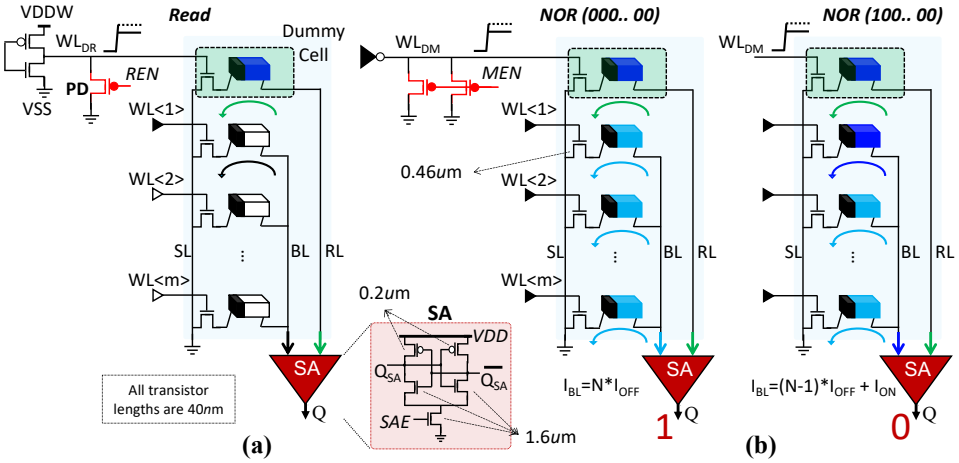


Figure 6.3: Proposed dummy wordline lowering technique in reference signal generator implementations for (a) read and (b) NOR logic operation.

appropriate I_{ref} needs to be regulated accordingly.

In our approach, we introduce a row of dummy RRAM bitcells to generate the required I_{ref} by regulating its wordline voltage WL_{DR}, as shown in Fig. 6.3. The following modifications are required: (1) one dummy row (*i.e.,* one dummy cell per column) to be programmed to SET (or LRS), (2) A bleeder circuit introduced in the wordline driver corresponding to the dummy row; this technique, typically used in a read-assist circuit in SRAMs [78], degrades or lowers the wordline voltage to a pre-defined value. This changes the overall conductance of the bitcell by changing V_{GS} of the pass transistor operating in the linear mode. Therefore, for any operation, we first determine the currents corresponding to critical *logic low* and *logic high* states that are responsible for discharging BL and configure the dummy wordline signal in such a way that it generates the average of these critical current values.

During a read operation, the pre-charged reference line RL is discharged through the dummy cell selected with the lowered WL_{DR}, as shown in Fig. 6.3a. The active low signal *REN* is switched to logic 0 (VSS) to enable the bleeder PMOS 'PD'. Note that the strength of the bleeder PMOS 'PD' is such that a voltage divider between the pull-up transistor of the WL_{DR} driver and 'PD' configures WL_{DR} to drive a current of $I_{ON}/2$ (being the ideal I_{ref}) for the dummy cell. A differential BL/RL voltage ($\Delta V = V_{BL} - V_{RL}$) is created which is sensed using a differential SA. We use a cross-coupled SA that involves a positive feedback loop to amplify the input differential voltage, shown in Fig. 6.3a.

This approach is scaled to perform multi-operand logic operations. Fig. 6.3b shows a possible design that uses two PMOS bleeder devices to configure the required reference current. However, a single dummy bitcell cannot be configured for a higher number of operands. Assuming an equivalent resistance ratio of the HRS and LRS of 100 (I_{ON}=100*I_{OFF}) [112], it can be seen that it is not possible to generate I_{ref} above 100*I_{OFF} using a single dummy cell in the SET state. For instance, I_{ref} in NOR^{256} requires the average current I_{ref} of the two currents generated by the two critical states *logic low* 256(0)

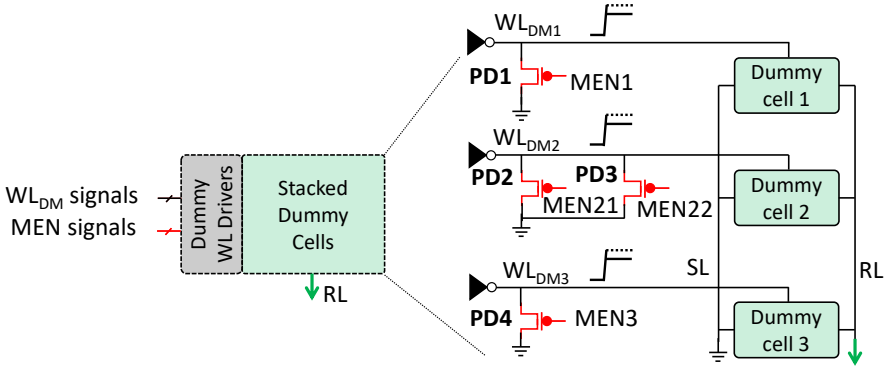


Figure 6.4: Proposed configurable reference generator to enable NOR logic of varying number of operands.

with I_{BL}=256*I_{OFF} and *logic high* 256(1) with I_{BL}=255*I_{OFF}+I_{ON}=355*I_{OFF}; in this case I_{ref} ≈300*I_{OFF} ≈3*I_{ON}, requiring at least three dummy cells. Moreover, additional reconfigurability is needed to generate references for NOR^N with varying N (N=2^z and 1<z<8) that can perform the binary search algorithm. In achieving this, we introduce a total of three dummy cells to perform up to NOR^{256}, as shown in Fig. 6.4. The size of the PMOS devices 'PD[1-4]' are such that a pre-determined combination of some or all dummy cell currents can generate the required I_{ref} currents. Table 6.1 shows the different configurations required to set up appropriate reference signal generations. These configurations ensure correct multi-operand NOR operations. Note that the bitlines are capable of allowing high multi-operand currents since these wires are conditioned to allow typically large programming currents. Therefore, peak power and current issues are not expected. The top part of the table shows the currents generated by the two critical states *logic low* and *logic high* (normalized to I_{OFF}) and the corresponding ideal I_{ref} currents for different values of operands. The second part of the table lists the different configurations of the three dummy cells leading to appropriate I_{ref} values. Here, column 3 enlists the possible pre-determined currents that different configurations of the corre-

Details	Parameters	Possible currents (in I_{OFF})	Number of Operands (N)					
			[1-8]	16	32	64	128	256
Critical currents (normalized to I_{OFF})	Logic low	-	8	16	32	64	128	256
	Logic high	-	107	115	131	163	227	355
	Ideal Ref.	-	53	66	82	114	178	305
Dummy cell 1	MEN1	50, 100	0	0	0	0	0	1
Dummy cell 2	MEN21/22	16, 32, 64, 100	x	00	01	10	10	11
Dummy cell 3	MEN3	64, 100	x	x	x	x	0	1
Generated Iref (in I_{OFF})		-	50	66	82	114	178	300

Signs	Meaning
-	Not applicable
x	Corresponding WLDM is not selected
MEN* = 0 (or 1)	Enables (or disables) corresponding PMOS bleeder device

Table 6.1: Different configurations to generate references for multi-operand NOR with varying number of operands; it is assumed that I_{ON}=100*I_{OFF}.

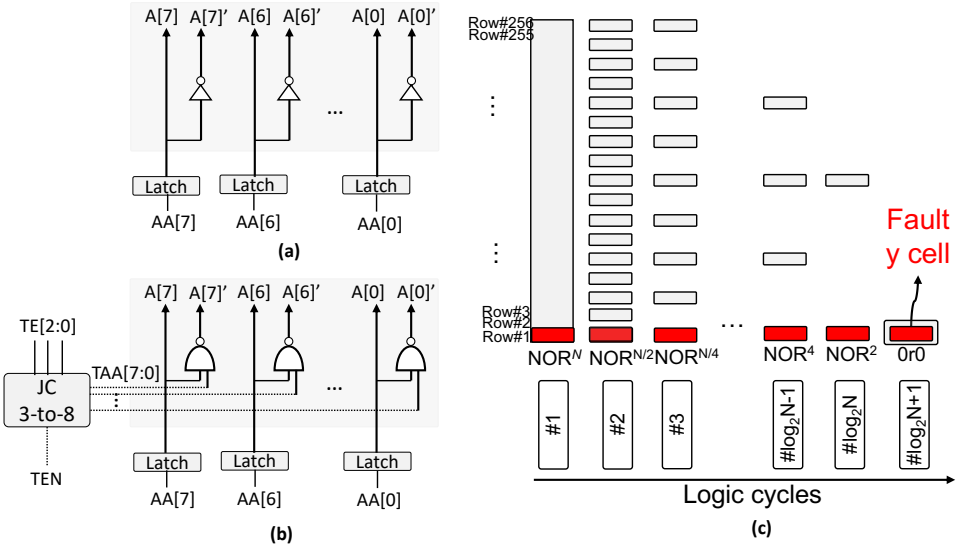


Figure 6.5: (a) Typical row address decoder for single row selection (b) Proposed circuit implementation of modified address decoder to realize binary search algorithm, and (c) the address selections for a given column.

sponding enable signal with prefix *MEN* can provide. As an illustration, let us consider N=64. The two critical states 64(0) and 64(1) result in currents 64*I_{OFF} and 163*I_{OFF}, respectively; the ideal I_{ref} in this case should be the average of the two; *i.e.*, 114*I_{OFF}. Referring to Fig. 6.4, with *MEN1*=0 (PMOS PD1 is ON), dummy cell 1 delivers 50*I_{OFF}; with *MEN21:MEN22*=10 (PMOS PD2 is ON, PD3 is OFF), dummy cell 2 delivers 64*I_{OFF}; '×' corresponding to *MEN3* implies WL_{DM3} is OFF and dummy cell 3 delivers no current; hence a total of 50*I_{OFF}+64*I_{OFF}=114*I_{OFF} as I_{ref} is generated. Similarly, it can be seen that all required I_{ref} currents can be generated to guarantee correct logic operations.

ROW ADDRESS DECODER

The row address decoder circuit in a 256-row RRAM memory decodes the input address vector AA[7:0] to select one row for memory operations. Typically, AA[7:0] bits are latched and the decoder generates the true and bit-wise complementary form of these inputs, *i.e.*, A[7:0] and A[7:0]' to perform the decoding, as shown in Fig. 6.5a. Similarly, in test mode, address AA[7:0] is decoded to select a row for testing. To select multiple rows, we propose to set all AA[7:0] bits to 1 and force some or all the complementary bits to 1, thereby, overriding the complementary circuit. This implies that all targeted WLs will be activated simultaneously. For instance, all (256) WLs are activated if all A[7:0] and A[7:0]' are forced to 1. In another instance, if LSB A_0=1 is allowed to have both true and complementary values, all the odd row-numbered WLs are activated.

The addressing scheme is described in Fig. 6.5b. Our scheme is implemented by replacing the inverters with NAND gates that are responsible for generating complementary bits A[7:0]' in the first stage of decoding. The inputs to these NAND gates are the latched AA[7:0] bits and the newly introduced TAA[7:0] control inputs generated by a 3-

to-8 Johnson counter (JC) [211]. During a normal memory operation, test enable signals TEN=0 and AA[7:0] propagate normally to perform single-row selection (TAA[7:0] are kept as 11111111 by TEN=0). In test mode, TEN=1 enables the JC to generate TAA[7:0] pattern that selects the required set of rows in each cycle (TEN=1 initially resets TAA[7:0] to 00000000). As explained earlier, a binary search engine requires the number of selected addresses in a logarithmic manner in each cycle, *i.e.*, the first cycle requires 256 WL activations, the second cycle requires 128, the third cycle requires 64, and so on. Fig. 6.5c depicts a possible set of address selections in each cycle. Note that replacing inverters with NAND gates in the decoder circuit has a negligible impact ($\sim 10\,ps$) on the critical path delay (i.e., operating cycle of the memory operations).

In summary, we realize the concept and methodology of our DFT approach with significantly low-cost design components; *i.e.*, a reference generator, and a row address decoder to accelerate the detection of *ETD faults*. However, a group of fixed reference signals cannot guarantee the detection of *HTD faults* by read or NOR logic operations. In this regard, we propose a reconfigurable DFT which is described next.

6.5. RECONFIGURABLE DFT SCHEMES PROPOSED FOR HTD FAULTS

This section presents low-cost reconfigurable logic design-based DFT schemes to target unique HTD RRAM faults.

6.5.1. CONCEPT

The idea is to make the DFT design capable of differentiating BL read voltage developed in the faulty case from the fault-free case with high certainty. Fig. 6.6 briefly illustrates the DFT concept. *Weak 1* and *weak 0* are defined as *faulty U states* that are closer to *logic 1* and *logic 0* states, respectively. These are imaginary states that allow us to discuss worst-case scenarios for detecting faulty *logic 1* and *logic 0* states. The following techniques change the condition of detecting a faulty state from a random read to a deterministic read output and are built on top of DFT-NOR^N. There are *two possible ways* to detect the faulty states, as shown in Fig. 6.6:

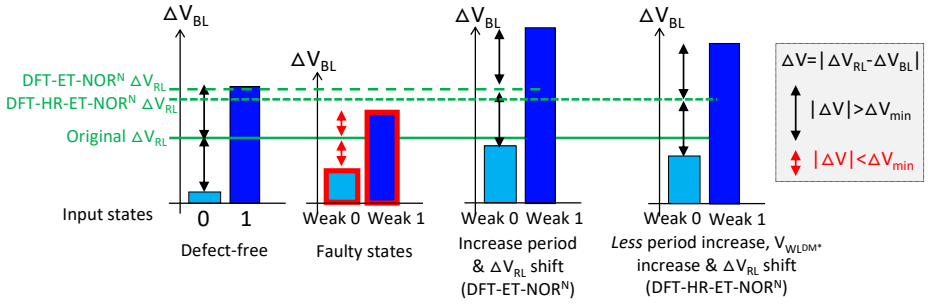


Figure 6.6: Concept of detecting faulty RRAM states.

- Enable *extended time (ET) of operation* to give sufficient time for the development of $|\Delta V| > \Delta V_{min}$, and shift the reference signal ΔV_{RL} to the middle of the two ΔV_{BL} developed by critical *logic 1* and *logic 0* states. This will be referred to as DFT-ET-NOR^N. This DFT requires a reconfigurable period of operation and a reconfigurable reference generator.
- In addition to the above technique, enable *high resistance ratio (HR)* of these two critical cases, e.g., one possible way is to decrease the resistance offered by the pass transistor by increasing the WL voltage $V_{WL_{DM^*}}$ (see Fig. 6.7); this also requires a shift in ΔV_{RL} but since this technique allows early development of the required margins compared to the case when DFT-ET-NOR^N is used, the magnitude of the shift is less. In other words, an increase in resistance ratio can allow fast detection of HTD faults compared to DFT-ET-NOR^N. This will be referred to as DFT-HR-ET-NOR^N. This DFT requires a reconfigurable period of operation, a reconfigurable reference generator, and a reconfigurable resistance ratio.

These solutions require three design components that are described in the next subsection. In summary, the duration of the BL discharge is increased to increase the developed ΔV, and the reference signal is tuned such that:

- If the cell is fault-free, the shift of reference signal is such that $|\Delta V| > \Delta V_{min}$, but the original polarity of ΔV is maintained to produce the correct read output. Otherwise, the test may declare a fault-free cell as faulty.
- If the cell is faulty (*i.e*, *U* state fault), the reference signal is shifted such that $|\Delta V| > \Delta V_{min}$, and polarity of ΔV is opposite to the one in the fault-free case; this enables the detection of the faulty cell.

6.5.2. DESIGN AND IMPLEMENTATION

To realize the programmable DFT scheme, three design techniques will be used; they are explained next.

RECONFIGURABLE PERIOD OF OPERATION

The (active) period of a read or logic operation, *i.e.*, period of WL activation, must be increased to allow sufficient time for fault detection. For example, the typical technique of using extra margin adjustment (EMA) that enables the reconfigurability of the active period of operation (i.e., duration of WL activation) in SRAMs [212] can be used. Here, the start of the active clock cycle is unchanged but the duration can be regulated by delaying the termination of the active clock with different permutations of the EMA signals. For instance, 3 EMA pins allow 8 different periods of operation.

RECONFIGURABLE REFERENCE GENERATOR

The DFT must be capable of shifting the reference signal with programmable magnitudes. To achieve this, we modify the reference generator of Fig. 6.4 by adding an additional row of dummy cells with some PMOS bleeder devices in the WL driver as shown in

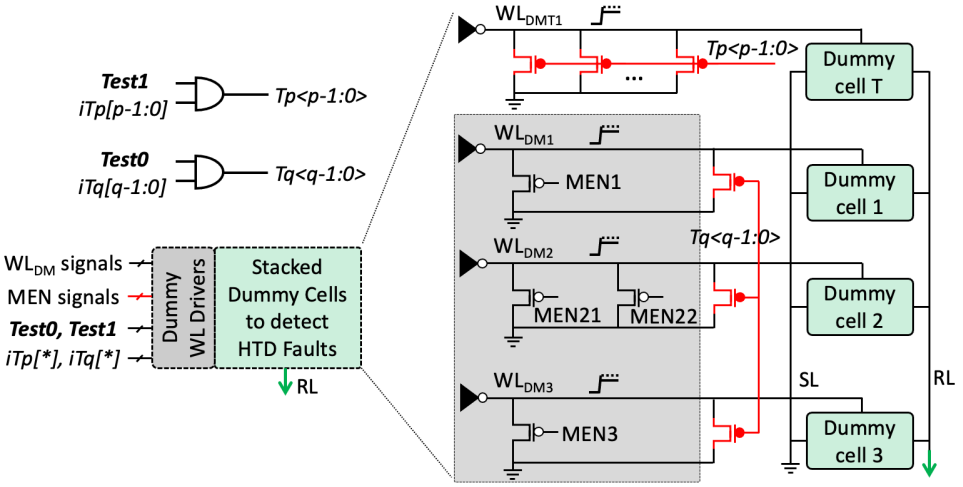


Figure 6.7: Proposed reconfigurable reference generator to detect HTD faults.

Fig. 6.7. This dummy row can supply additional I_{ref} to shift V_{RL} lower than the normal case. The number of bleeder PMOS devices say p, in the WL driver of this dummy row can be configured to provide up to 2^p different possible reference signal strengths. On the other hand, additional bleeder PMOS devices can be added to existing dummy WL drivers or drivers to reduce the strength of I_{ref} to shift V_{RL} higher than the normal case. In a similar way, additional q PMOS devices can possibly configure up to 2^q different reference signal strengths. The enable signals *Test0* and *Test1* activate the detection of *weak 0* and *weak 1* states, respectively. *Test1* also activates the newly added 'Dummy cell T'. Enable signals Tp[p-1:0] and Tq[q-1:0] regulates the magnitude of I_{ref}.

RECONFIGURABLE RESISTANCE RATIO

The effective resistance ratio of the faulty and fault-free states is increased to develop high read margins resulting in the detection of certain *unique* RRAM faults. To achieve this, we increase the global WL voltage VDDW of the WL driver in Fig. 6.3 (*i.e.*, increase V_{GS} of the NMOS pass transistor) to reduce its resistance and increase the overall resistance ratio. To illustrate this concept, consider the resistance offered by NMOS to be R_N, and resistances of RESET and SET states to be R_R and R_S, respectively ($R_R >> R_S >> R_N$). In the series combination of NMOS and RRAM device in a bitcell, change in R_N will have a much greater impact on the effective resistance of the SET state (R_N+R_S) compared to the RESET state (R_N+$R_R \approx R_R$). This implies that an increase in WL voltage does allow more discharge of V_{BL} in all cases, but this discharge is more severe if the bitcell is in LRS. This increases the chances of detecting a faulty state as well as allows faster detection of faulty cases. Below is a quantitative illustration to show the increase in resistance ratio (RR) of the faulty and the fault-free state. Let the resistance of the faulty RRAM state be R_F=100KΩ, R_S=8KΩ and R_N at 0.9V and 1.1V be 2KΩ and 1KΩ, respectively. The RRs in these scenarios are;

@0.9V: RR = $\frac{R_N+R_F}{R_N+R_S}=\frac{102}{10} \approx 10$, @1.1V: RR = $\frac{101}{9} \approx 11$

Here, we see that shifts in both RR (from 10 to 11) and LRS (from 10 to 9) are effectively about 10%, whereas the shift in faulty state resistance is only ~1%. This validates the underlying concept of our DFT scheme. Note that DFT-HR-ET-NOR^N allows faster detection of HTD faults at the expense of increased voltage of operation. Hence, a trade-off between the use of higher voltage and the optimization of test time.

In summary, we realize the concept of our DFT approach with low-cost design components to accelerate the detection of ETD and HTD faults and improve the FC.

6.6. DFT VALIDATION AND TEST DEVELOPMENT

This section presents validation of our DFT implementations and develops new testing algorithms. We show the potential to provide low-cost and efficient testing methodology.

6.6.1. SETUP, FAULT MODELING AND ANALYSIS

SIMULATION SETUP

The left part of Table 6.2 presents the details of our simulation platform. A 256x256 RRAM memory is built using industry-standard TSMC 40 nm CMOS device technology. The layout of the sense amplifier (SA) described in Fig. 6.3 is developed and extracted to establish minimum differential sensing margins of 40 mV. The rest of the digital components such as address decoders, drivers, control blocks, etc. are built using standard cell extracted netlists to develop a comprehensive platform for circuit simulations.

The right half of Table 6.2 covers the details of the RRAM device model. A physics-driven HfO_2/TiO_x-based RRAM device (Verilog-A) model is used for our simulations that is based on physical dimensions and the ions concentration of oxide [213]. The cylindrical RRAM device with the oxide concentration has a height l_{det} and radius r_{det}, and the minimum (n_{min}) and maximum (n_{max}) concentration of oxide ions corresponding to the LRS and HRS, respectively, can be set. The internal state parameter n_{real} represents instant oxide ions concentration, thus defining the state of an RRAM device at any given time. A higher (lower) value of n_{real} corresponds to the LRS (HRS).

Parameters	Specifications	Parameters	Specifications
Simulation Platform		**RRAM Bitcell**	
Simulator	Cadence Spectre	Configuration	1T1R
R/W Voltage	0.9 V/2.5 V ±10 %	RRAM Device	HfO_2/TiO_x [112]
R/W/NOR Time	0.7 ns/2 ns/1.2 ns	HRS/LRS	1 $M\Omega$ / 10 $K\Omega$
CMOS	40 nm TSMC, 3 σ	1T (NMOS: W/L)	460 nm/40 nm
Temperature	-40°C to 125°C	BL (WL) res.	0.2 Ω (0.4 Ω)
Memory Core		BL (WL) cap.	0.3 fF (0.6 fF)
Array	256x256MUX4	n_{min}, n_{max}	0.03, 30
SA	5T voltage-based	l_{ret}	0.05 nm
SA V_{min}	40 mV	r_{ret}	10 nm

Table 6.2: Design parameters.

To accurately capture the wire parasitics of the RRAM memory bitcell array and coupling effects of the neighboring bitcells, the layout of a 3x3 memory matrix is developed and extracted to derive the middle bitcell, as shown in Fig. 6.8a. Defects in the RRAM device, interconnects and transistors are modelled as linear resistors. All these defects are described as one of the three types of defects, as shown in Fig. 6.8b; short-circuit to power signals VDD or VSS (resistors are with prefix *RS*), open-circuit or broken connection (*RO*) and a bridge (*RB*) between any two nodes other than VDD or VSS. The resistance of these linear resistors ranges from 1Ω ($2^0\Omega$) to 134MΩ ($2^{27}\Omega$) swept with a geometric sequence of common ratio $2^3\Omega$ (10 possible values) to capture any possible defects. This also covers the minimum and maximum range of RRAM effective resistance *i.e.,* 10KΩ and 1MΩ, respectively.

FAULT MODELING AND ANALYSIS

To identify faulty behavior due to possible defects, simulations are conducted with a defect-free netlist and are then compared with those conducted based on defect injection in the netlist. The aforementioned defects are injected one at a time at the considered locations. The memory and DFT-related logic operations considered are:

- *0w0*: write 0 (RESET) to a cell initialized to 0.
- *0w1*: write 1 (SET) to a cell initialized to 0.
- *1w0*: write 0 (RESET) to a cell initialized to 1.
- *1w1*: write 1 (RESET) to a cell initialized to 1.
- *0r0*: read a cell initialized to 0 with the expected value 0.
- *1r1*: read a cell initialized to 1 with the expected value 1.
- $0NOR^N1$: N-operand NOR with all cells initialized to 0 with expected value 1.

Fig. 6.9 shows the simulation results for a read, write, and NOR operation for one defect RB5. Based on the strength of the defect, we can distinguish three cases: RB5=$2^{27}\Omega$

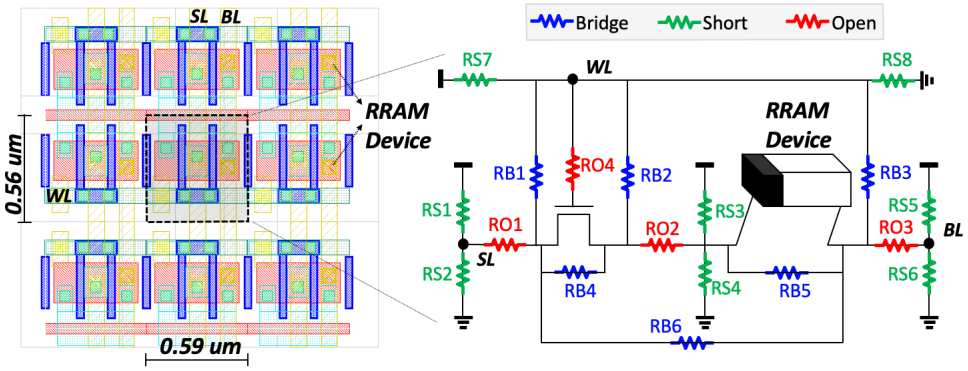


Figure 6.8: Layout of a 3x3 RRAM bitcells and derived defect model.

fault-free (first row), RB5=$2^9\Omega$ ETD fault (second row), $2^{15}\Omega$ HTD fault (third row). The bottom row shows the related WL timing. The voltage and timing specifications associated with these operations are described in Table 6.2. Next, we explain the observations for all three cases. Case fault-free: Fig. 6.9a shows that when reading a cell $\Delta V>40$ *mV*, and thus the cell is read without any faults. Fig. 6.9b shows that the cell can transition both from 1→0 and from 0→1. Finally, Fig. 6.9c shows that for the NOR operation $\Delta V>40$ *mV* as well, and thus that this operation also succeeds. Case ETD (second row): Fig. 6.9a shows that as the defect RB5=$2^9\Omega$ provides a low ohmic read path (SAF1), a higher discharge current flows than expected. This results in $\Delta V>40$ *mV* but with incorrect polarity. Hence, the operation is faulty as the read output is 1 when the expected value is 0. Fig. 6.9b shows that the low ohmic bridge results in a low voltage across the RRAM device. Hence, the two write operations are faulty (TF). Similar to the faulty read case, Fig. 6.9c shows a faulty NOR operation due to the low ohmic read path. Case HTD (third row): Fig. 6.9a shows that $\Delta V<40$ *mV* as the strength of the defect RB5=$2^{15}\Omega$ is in the middle between LRS and HRS resistance values. Hence, the read operation may produce a random read output as the effective RRAM resistance falls in the U state. Fig. 6.9b shows that the lack of sufficient voltage when writing across the RRAM device causes the RRAM to fall in the U state (UWF). Similar to the faulty read case, Fig. 6.9c shows a faulty NOR operation as the effective RRAM resistance falls in the U state.

6.6.2. DFT VALIDATION

TARGETING ETD FAULTS

ETD faults provide sufficient sensing margins (|ΔV|>40mV) with incorrect polarity. Fig. 6.10 shows the simulation results to illustrate the detection of ETD faults by deploying DFT-NOR^N (where N=256). We simulated the defects that introduce stuck-at-faults; RB5 is

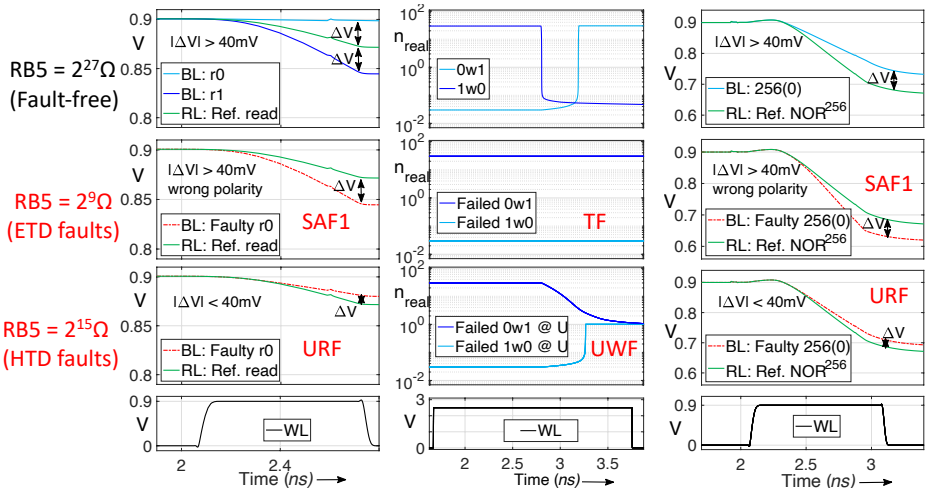


Figure 6.9: Timing diagram (from the top): Correct, ETD and HTD faulty behaviors of (a) read, (b) write, and (c) 0NOR^{256}1 operations. Note that these operations have different WL activation periods.

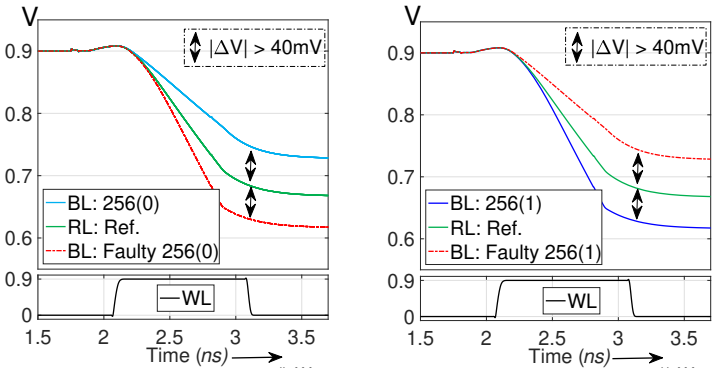


Figure 6.10: Accelerating the detection of ETD faulty behavior using NOR logic.

set to $2^9\Omega$ (refer to Fig. 6.8b) to simulate SAF1 and RO2 is set to $2^{18}\Omega$ to simulate SAF0. On the left side of the figure, defect-free and SAF1 cases corresponding to 256(0) state conditions are presented, where the faulty case allows a higher I_{BL} that discharges the BL more than expected. This fault can be detected by performing $0NOR^{256}1$ operation as the faulty case results in output 0 (the expected result is 1). Similarly, on the right side of the figure, defect-free and SAF0 cases corresponding to 256(1) state conditions are presented, where the faulty case allows a lower I_{BL} that discharges the BL less than expected. This fault can also be detected by performing $1NOR^{256}0$ operation as the faulty case results in output 1 (the expected result is 0).

To validate the binary search technique described in Section 6.4, we introduce an ETD fault at row address location 215 (binary: 11101011) and then determine its location. RB5 is set to $2^6\Omega$ in the bitcell present at this address location to simulate an SAF1. Table 6.3 illustrates how the identification of this faulty cell can be achieved after 8 ($\log_2 256$) consecutive NOR operations of varying operand size; *i.e.* by each NOR cycle from NOR^{256} to NOR^2, the selected addresses converge to the faulty cell address location. First, we initialize AA[7:0]=11111111 and enable signals TEN=1 (test mode ON) and TEN[2:0]=000 such that TAA[7:0]=00000000. This ensures all 256 address lines are selected in the first cycle. Following the sequence of NOR operations, with every *k*th NOR operation, the JC increments such that *k*th LSB of TAA[7:0] is toggled to 1 independent of the outcome. This allows true and complementary forms of first up to *k*th LSB of AA[7:0] to propagate in A[7:0] and A[7:0]' respectively. However, AA[7:0] follows a different sequence depend-

Cycle	Outcome	TE[2:0]	TAA[7:0]	AA[7:0]	A[7:0]	A[7:0]'
1 (NOR^{256})	wrong	000	00000000	11111111	11111111	11111111
2 (NOR^{128})	wrong	001	00000001	11111111	11111111	11111110
3 (NOR^{64})	correct	010	00000011	11111111	11111111	11111100
4 (NOR^{32})	wrong	011	00000111	11111011	11111011	11111100
5 (NOR^{16})	correct	100	00001111	11111011	11111011	11110100
6 (NOR^{8})	wrong	101	00011111	11101011	11101011	11110100
7 (NOR^{4})	wrong	110	00111111	11101011	11101011	11010100
8 (NOR^{2})	wrong	111	01111111	11101011	11101011	10010100
9 (Read0)	wrong	-	11111111	11101011	11101011	00010100

Table 6.3: Address selections to realize binary search algorithm.

ing on the outcome of the operation. Every incorrect *k*th NOR operation implies that the faulty cell is selected in that cycle. Therefore, AA[7:0] is not changed. However, if the *k*th NOR operation is correct, then it implies that the faulty cell is not selected; hence for the next cycle, *k*th LSB of AA[7:0] is reset to 0. The illustrative example presented in the table shows that operations of cycles 3 and 5 are correct, the 3rd LSB and 5th LSB of AA[7:0] are toggled to 0, while the others remain at 1. The final cycle is a read operation as ultimately one of the two addresses selected in NOR^2 operation is faulty. In summary, the faulty cell is identified in 9 ($\log_2 256+1$) cycles.

TARGETING HTD FAULTS

Reading a faulty cell suffering from an HTD fault produces random read outputs; this is because the developed ΔV is below ΔV_{min}. The DFT techniques that are proposed increase the difference in the developed say ΔV, between the faulty and fault-free state to differentiate one state from the other. We refer to Fig. 6.11 to validate our DFT schemes. The top sub-figure shows the simulation results without any DFT; the middle sub-figure shows the results when DFT-ET-NOR^N is deployed and the bottom sub-figure the results when DFT-HR-ET-NOR^N is deployed. Here, let us consider a logic NOR operation NOR^{256} where in one of the two critical cases, say 256(0), one device is in the U state, as shown in the top part of the figure. This shifts the input signal V_{BL} closer to V_{RL}, such that it violates the minimum sensing margin of ΔV_{min}=40 *m*V to a value of 20 *m*V. This faulty behavior can be detected by increasing the time of operation (here, by 4X)

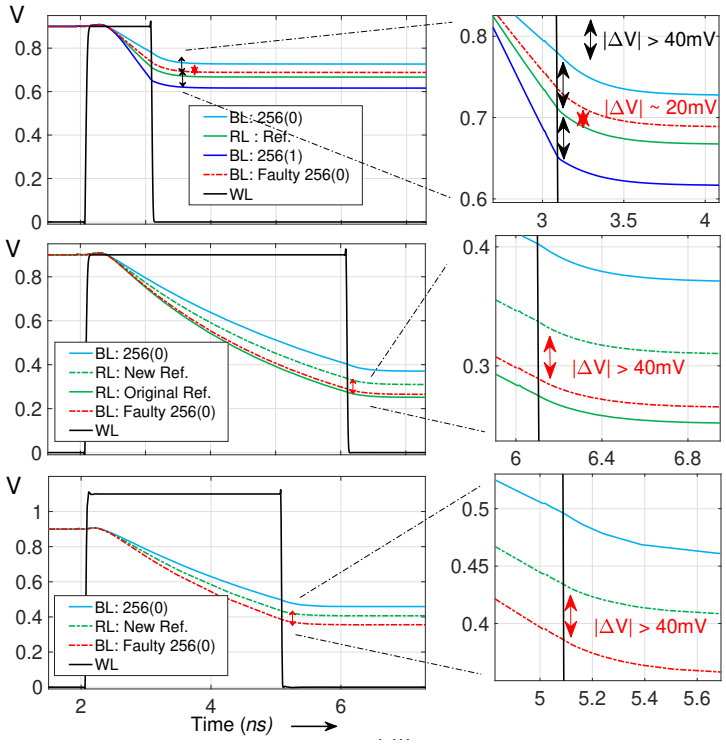


Figure 6.11: Increased time of operation and reference shift to detect HTD faults.

such that V_{BL} in the faulty case becomes at least 80 *m*V more than the fault-free case and then configuring V_{RL} >40 *m*V from both faulty and fault-free V_{BL} signals, as shown in the middle part of the figure.

The bottom part of Fig. 6.11 validates DFT-HR-ET-NOR^N, which can accelerate the detection of HTD faults by improving the read margins. The figure shows that an increase in V_{WL} pushes V_{BL} developed with *weak 0* into a range where, by increasing the time (here, by 3X) and shifting the reference signal, the *weak 0* cell is read as 1. In a similar way, *weak 1*, *L*, and *H* states can be detected with pre-determined reference signals associated with each of these states.

Similar to the ETD fault cell identification, 8+1 consecutive NOR+ read cycles converge the address decoder to determine the address with the faulty state bitcell.

6.6.3. TEST PROCEDURE

Any memory configuration built on any memory technology that involves read operation based on the difference in BL discharge currents (or developed BL voltages) can benefit from our proposed DFT schemes. Following this, we develop a test procedure for high-volume production that deploys our proposed DFT schemes in an efficient manner; i.e., targeting the maximum FC while minimizing test time. Fig. 6.12 summarizes the three proposed DFT schemes. We show that DFT-NOR^N can be used to speed up the execution of any test algorithm for ETD faults detected by a read operation. In addition, DFT-ET-NOR^N and DFT-HR-ET-NOR^N schemes can be used to increase the FC of any test targeting HTD faults as these DFTs enable deterministic read operations rather than random reads for some HTD faults. DFT-HR-ET-NOR^N provides similar FC with faster detection of the target faults compared to DFT-ET-NOR^N at the expense of increased WL voltage. In short, for the most optimized test time with maximum FC, DFT-HR-ET-NOR^N must be used. The idea to develop a testing sequence is to begin with the regular march test i.e., MATS+ algorithm [214], and then modify the algorithm incorporating our DFT schemes. More specifically, replace r0 with NOR^{N}1 for the detection of faults. As an illustration, we present our proposed march test algorithms to detect and determine the location of the URF (notation adapted from [214]). The URF is sensitized by 1w0 operation that can switch the faulty RRAM cell from 1→U state and the following NOR^{N}1 detects this faulty state. Similarly, the 0w1 operation can sensitize this fault by switching the faulty RRAM cell from 0→U state and the following r1 operation detects this state. Additionally, we duplicate every write operation to increase the possibility of capturing IUSF. Following are the march test algorithms to detect and determine the location of all possible faults with test lengths of $6N$+8 and $6N$+8log$_2 N$+8, respectively, where N is the RRAM memory size:

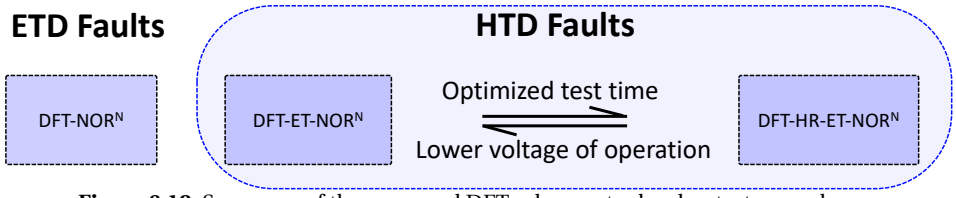


Figure 6.12: Summary of the proposed DFT schemes to develop test procedure.

Fault detection: $\{\Updownarrow_N(\text{w0w0}); \Uparrow_4(\text{NOR}_x^N 1); \Uparrow_N(\text{w1w1}); \Downarrow_N(\text{r}_x 1, \text{w0}); \Uparrow_4(\text{NOR}_x^N 1)\}$

Fault detection and location: $\{\Updownarrow_N(\text{w0w0}); \Uparrow_{4log_2 N}(\text{NOR}_x^* 1); \Uparrow_4(\text{r}_x 0); \Uparrow_N(\text{w1w1}); \Downarrow_N(\text{r}_x 1, \text{w0}); \ldots \Downarrow_{4log_2 N}(\text{NOR}_x^* 1); \Uparrow_4(\text{r}_x 0)\}$

Here, we define NOR, read 0, and read 1 operation as $\text{NOR}_x^N 1$, $\text{r}_x 0$ and $\text{r}_x 1$ when DFT-HR-ET-NOR^N is deployed. Here, x can be *H, weak 0, weak 1,* or *L* and they correspond to reference signals in the mentioned states), respectively. Note that $\text{NOR}_x^N 1$ and $\text{r}_x 0$ operations are performed four times ($\Updownarrow_4$) to capture these four faulty state conditions. $\text{NOR}_x^* 1$ implies that these NOR operations assume a varying number of operands required for the binary search algorithm.

6.7. PRIOR ARTS AND COMPARISON RESULTS

In this section, we present a qualitative and a quantitative comparison with the state-of-the-art solutions.

6.7.1. QUALITATIVE COMPARISON

Many test solutions for RRAMs have been presented in the literature. They are summarized in Table 6.4. The tests can be classified on their type, i.e., whether they are a march algorithm only, or whether they are a DFT (possibly in combination with a march algorithm). The table lists every fault if it can be detected by the test. It follows that march algorithms only solutions are unable to achieve a high FC. For e.g., March W-1T1R achieves only 64% FC. This is because RRAM unique faults are HTD, and thus require DFT to be reliably detected. The existing DFTs can be further divided into DFTs that speed up testing [189, 197, 202], and DFTs that increase the FC [195, 200, 204].

DFTS TO OPTIMIZE TEST TIME

DFTs that speed up the testing process either reduce the duration of the write operation during testing [202], or they read multiple cells at once and thus reduce the number of read operations [189, 197]. In general, these DFT schemes do not result in high FC. In [202], the authors propose to shorten the write *0* operation so that it just barely switches into the HRS range and check it afterward by using a dedicated reference signal. The assumption is that faulty cells will not switch in time and thus will be detected. However, our DFT schemes do not involve dedicated write operations and, moreover, replace a large number of sequential read operations with a single NOR operation to optimize the test time. In [197], a MAGIC CIM-based NOR operation is used to perform logic in parallel. However, the voltages required to do this are higher than 7 V, which limits the applicability. In contrast, our DFT schemes are based on read or NOR operations that typically operate at low voltages *i.e.,* 0.9 V. In [189], multiple cells are selected and the sneak paths are used to detect faults. However, selecting groups of multiple cells requires an expensive modification of the decoder circuitry and sneak paths may cause functional issues. In contrast, a small number of additional components consisting of a few PMOS transistors and bitcells per column are introduced to implement our DFTs.

DFTS FOR HIGHER FC

DFTs that increase FC either modify the read or write operation so that HTD faults are sensitized and subsequently detected. In [200], the write operation is weakened so that faulty cells will fail to switch properly. After such an operation, faulty cells will be in the wrong state, while fault-free ones will be in the correct state. The drawback of this DFT is that it needs to be calibrated precisely, to prevent the weak write operation may also fail on good cells. In our case, the DFTs consist of read operations that do not rely on the delicate programming of the RRAM device. In [204], two new references are introduced that can be used to detect the *U* state, i.e., one reference between *1* and *U*, and one between *U* and *0*. The drawback of this is that the sense margin of the SA may cause some faulty cells in *U* to be incorrectly read out as *1* or *0*, while some good cells in *1* or *0* will be read as *U*, resulting in yield loss and test escapes. In contrast, we ensure sufficient read margins by deploying DFT-ET-NOR^N and DFT-HR-ET-NOR^N schemes to have deterministic read and NOR operations. In [195], a sensor measures and compares the internal node of each cell with pre-determined references to detect the state of the cell. It is unclear how this sensor can be adapted to work for multiple cells in an economic way if every internal node needs to be measured. However, our DFTs do not involve any complicated structures and allow low-cost integration within the memory.

In short, no previous test solutions detect all RRAM faults in a reliable and cost-effective manner. Furthermore, none of the tests are able to guarantee the detection of the IUSF.

6.7.2. QUANTITATIVE COMPARISON

The testing cost in terms of area overhead and test length and the FC are compared with the state-of-the-art solutions in Table 6.4. In the table, we show that our proposed DFT detects all faults with partial detection of the IUSF. This is because of the probabilistic nature of the occurrence of this fault. However, as several NOR and read operations are performed on the faulty cell ($\log_2$N+1 cycles), we increase the chances of detecting this fault. With better FC than Enhanced March [204], an area-efficiency of 6× (normalized to number of transistors) and 2.3× speed (test length) is achieved, assuming number of rows N_r and columns N_c of the memory are 256 each.

6.8. DISCUSSIONS AND FUTURE DIRECTIONS

This section highlights the applicability and adaptability of our DFT schemes with different design technologies.

BITCELL CONFIGURATION

Memory units built on any bitcell configuration that involves read operation based on the difference in BL discharge currents (or developed BL voltages) can benefit from our proposed DFT schemes. BL voltage (or current) in any such configurations can be compared with RL developed by deploying our reference techniques.

Name	Type	Conventional						Unique					Fault Coverage	Test Length Write, Read	Cost # of transistors
		SAF	TF	WDF	IRF	RDF	CFst	UWF	URF	Deep	IUSF	CFud			
March-MOM [189]	March	Y	Y	P	N	N	P	N	Y	Y	N	N	36%	5N, 4N	-
March-1T1R [191]	March	Y	Y	Y	N	N	Y	N	N	P	N	N	36%	$5N^{+}$, 4N	-
March C* [190]	March	Y	Y	N	Y	Y	Y	N	N	N	N	N	45%	4N, 6N	-
March C*-1T1R [192]	March	Y	Y	N	Y	Y	Y	N	N	Y	N	N	55%	6N, 6N	-
March-CMOL [194]	March	Y	Y	N	Y	Y	Y	N	N	Y	N	N	55%	-	-
March W-1T1R [193]	March	Y	Y	Y	Y	Y	Y	N	N	Y	N	N	64%	9N, 8N	-
Parallel March [197]	DFT	Y	Y	N	Y	Y	Y	N	N	Y	N	N	55%	4(N+1), 5N+N_r	-
Sneak-path [189]	DFT	Y	Y	P	N	N	P	N	Y	Y	N	N	36%	7N, 5N/3	28+26N_r
Weak-write [200]	DFT	N	N	N	Y	N	N	Y	Y	Y	N	N	36%	No March	24+18N_r
Fast write [202]	DFT	Y	Y	N	N	N	N	Y	Y	Y	N	N	45%	(4T+1+x)N, 6N	50+18N_r
On-chip sensor [195]	DFT	Y	Y	Y	Y	Y	N	Y	Y	Y	N	N	73%	No March	20N
Enhanced March [204]	DFT	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	Y	91%	8N, 6N	13(N_c+N_r+4)
Proposed (DFT-HR-ET-NOR^N)	**DFT**	Y	Y	Y	Y	Y	Y	Y	Y	Y	P	Y	**91+%**	**6N, 8log$_2$N$_r$+8**	**2log$_2$N$_r$+96+4N$_c$**

Table 6.4: Comparison of RRAM test solutions. Yes (Y), No (N), Partial (P). N_r and N_c are number of rows and columns of the memory, respectively.

MULTI-BIT CAPACITY

Multi-bit capability of memristors has been greatly explored to increase the storage density. In terms of RR, we can assume a reduction in the difference of effective resistance states; this is because more quantized resistance states (>2) have originated from a similar range that is associated with only two states in a 1-bit RRAM. However, this reduction only reduces the minimum possible ΔV_{BL} of the two critical states which can be alleviated using a longer cycle of operation. In short, a reduced maximum number of operands in the NOR operation or an increase in cycle time can still offer fast testing and facilitate improved FC.

OTHER MEMRISTOR TECHNOLOGIES

The proposed DFT scheme requires high RR to accelerate the detection of faults. It stems from the fact that in a NOR^N operation, the difference in V_{BL} developed with critical states N(0) and N(1) increases as the RR increases and vice-versa (see Eq. 6.1). Therefore, memory technologies such as phase-change memory with possible high RR can be expected to have low-cost testing with high FC, whereas, spin-torque transfer magnetic random access memory (STT-MRAM) with inherently low RR can have improved FC.

ADVANCED CMOS TECHNOLOGY

With technology scaling, a higher RR can be expected [215]. In Eq. 6.1, a lower R_N implies higher RR, *i.e.* RR value approaches the ideal value of $\frac{R_{OFF}}{R_{ON}}$. Advanced technology nodes such as FinFET can allow lower variations which improves the overall accuracy of the DFT scheme [216]. In addition, with voltage down-scaling, altering the V_{WL} can have a greater impact on the conductance of the pass transistor (V_{GS} is closer to threshold voltage). SRAM designs can also benefit from our proposed scheme, potentially by comparing bitline pairs individually (alternatively, one BL or NBL per cycle) with the reference line. In short, this can facilitate a higher impact of our DFT schemes.

6.9. CONCLUSION

In this chapter, we present cost-efficient CIM-based DFT schemes that improve the cost of testing and fault coverage (FC). Our schemes deploy multi-operand NOR logic op-

erations that involve multi-row select read operations to accelerate the testing of ETD faults. In addition, the reconfigurability of the DFT implementation aids the detection of unique RRAM faults that are HTD. The design and implementation of a fast search algorithm facilitates the diagnosis of faults. Our proposed DFT implementations are validated on a post-layout extracted platform and testing sequences are introduced incorporating the proposed DFTs. Results show that more than 2.3× speedup, 6× area reduction, and better FC are achieved compared to the state-of-the-art.

7

MULTI-OPERAND XOR USING RRAM

This chapter presents an RRAM-based CIM architecture that performs accurate multi-operand XOR logic operations while addressing challenges related to device variations and non-idealities that typically narrow down the sensing margin and severely impact computing accuracy, efficiency, and scalability. In this regard, we propose MOXOR-CIM, a circuit-level mitigation solution for CIM-based multi-operand XOR logic operations; the scheme uses a voltage-to-time converter (VTC) to perform a multi-phased XOR in a single clock cycle. Here, we utilize the bitline capacitances for a voltage-based sensing for computation which generates the output voltage value being linear to the operand values; the voltage is then converted into desired logic output using the VTC. Moreover, low-power techniques are introduced in deploying sense amplifiers such as regulating power consumption during its operation and disabling it at the instant amplifier decision is made. Simulation results for a post-layout extracted 512x512 (256Kb) RRAM-based CIM array show that up to 16-operand XOR operation can be accurately and reliably performed as opposed to a maximum of 3 operands supported by state-of-the-art solutions while offering up to 5.5× better energy-efficiency and 12× better performance.

This chapter is based on [68].

7.1. INTRODUCTION

RRAM-based CIM architectures are extensively employed to accelerate the exclusive OR (XOR) logic operations that can be used for many applications such as database queries, testing, encryption, etc [49, 64, 67, 217, 218]. However, their performance is highly impacted due to their analog nature of storing and computing data, resistance drifts in RRAMs, device variation, etc. These inherent properties can impose serious limitations on the computing accuracy and the energy efficiency of the CIM architecture [2, 219].

The underlying principle of implementing an analog CIM-based logic operation is by converting aggregated information of the operands into analog form (voltage or current) as a function of equivalent compute output. Typically, to perform a logic operation, a multi-row READ is performed by selecting the desired operands, and based on the aggregated conductance of the selected bitcells, CIM generates an output in the analog domain. Depending on the different types and number of operations, and the number of operands supported, one or several reference signals aid the sense amplifier (SA) in determining the desired output. Fig. 7.1a illustrates the interaction of V_R with conductances of the selected bitcells (operands) i.e., $V_R \cdot G_1$ and $V_R \cdot G_2$, resulting in total accumulated column current I_c; the references corresponding to OR or NAND can be selected using a MUX to perform the desired operations via SA.

Several two and multi-operand CIM-based XOR implementations have been proposed where *all* operands reside in the memory. CIM-based n-operand XOR operation represented as XOR^n, can be performed using either a cascading of operations [64, 65, 106, 109, 217, 218, 220] or performing one-time multi-operand XOR [49, 110, 221]. The top and bottom parts of Fig. 7.1b illustrate 1) Cascading two-operand OR and NAND operations, and 2) cascading two-operand XOR operations, respectively. In these approaches, an operation is performed in each cycle and the partial result is stored in the periphery for further interaction with partial results in the subsequent cycles. However, this sequential approach of performing a series of OR·NAND or XOR operations significantly

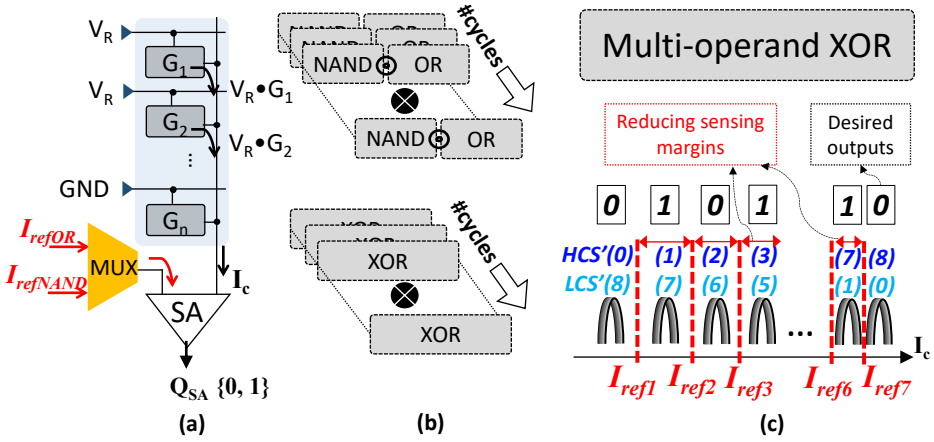


Figure 7.1: (a) Working of XOR, operation; Typical implementations via (b) sequential cascading, and (c) multi-operand XOR operations.

suffers from high latency and energy consumption, where each of these metrics increases *linearly* with operand size n. In addition, post-processing to accumulate the partial results in the periphery adds to large area overheads. On the other hand, XOR^n in a single cycle typically requires multi-reference signals I_{ref*} to differentiate each state. Fig. 7.1c illustrates this for XOR^8. Note that the signal margin, defined as the minimum sensing margin between two states, reduces as the operand size increases due to circuit non-idealities such as IR drop [49, 65, 66]. To address this, dedicated design units are introduced such as complex SA, and adaptive reference signals owing to an inefficient, large, and power-hungry system [49, 64, 106–108, 110, 221–223]. Unfortunately, this scheme is not scalable in terms of operand size as further reduction in sensing margin can not be addressed. In addition, unlike multi-operand (N)OR and (N)AND operations proposed in [66], where critical states deciding the logic output are either all 0's or all 1's, XOR functionality demands the output to change in an alternate manner with an incremental number of 1's among the operands, which is extremely challenging with the current methods. Although, few logic solutions have adopted differential sensing scheme [178, 179, 224]; however, the use of multiple references implies that either they face the aforementioned issues or are limited to multi-operand (N)OR and (N)AND operations [49, 66]. In short, there is still a need for cost-effective circuit-level solutions that not only provide accurate and energy-efficient XOR operations but also push the limit of maximizing the number of operands.

This chapter presents MOXOR-CIM, a multi-operand XOR-based CIM using compact voltage-to-time conversion (VTC) techniques for high-performance and energy-efficient computing. Our techniques accurately and reliably perform multi-operand XOR operations with up to 16 operands in a single cycle, while realizing up to 12× speedup and 5.5× energy efficiency compared to state-of-the-art CIM-based XOR solutions. The key contributions are:

- Develops two schemes: a compact one-reference, uni-polar VTC (UVTC)-based scheme and a high-speed reference-less bipolar VTC (BVTC)-based scheme with detailed trade-off analyses of the overall efficiency and the maximum number of operands supported.
- Designs them with high-computing accuracy by ensuring sufficient signal margins.
- Integrates them in CIM by utilizing bitline caps for voltage-based sensing and computing, whereby, CIM generates an output voltage signal linear to the operand values which is converted into desired logic output by the aforementioned VTC-based designs. A global counter counts to a number which is a function of the desired operand size.
- Reports simulation results and comprehensive comparison using a post-layout extracted 512x512 RRAM-based CIM.

The rest of the chapter is organized as follows. Section II presents the proposed MOXOR-CIM schemes, followed by results in Section III. Finally, Section IV concludes the chapter.

7.2. VOLTAGE-TO-TIME BASED MOXOR-CIM

This section presents the details of our proposed schemes.

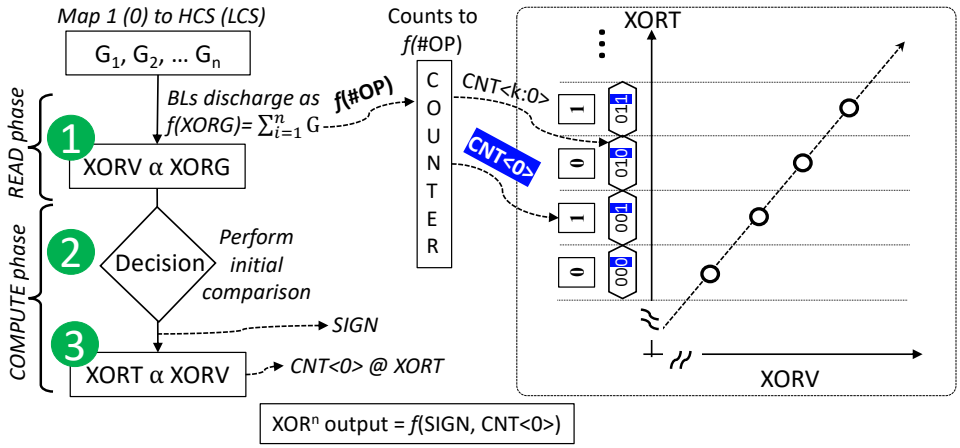


Figure 7.2: (a) Overview of our proposed schemes

7.2.1. OVERVIEW

The underlying principle is based on VTC to perform XOR^n of arbitrary operand size n, as described in Fig. 7.2. The entire operation is divided into two broad phases: 1) READ phase, where CIM generates an output voltage XORV being *linearly* proportional to the aggregated conductance value XORG (representing accumulated conductance of the selected data bits, XORG = $\sum_1^n G_i$); and COMPUTE phase, where the developed XORV is translated into *linearly* proportional time duration XORT. In the COMPUTE phase, first, an initial comparison of BL voltage with a reference voltage (for scheme UVTC) or with complementary BL voltage (for scheme BVTC) is performed to generate a SIGN-bit. Then, the combination of this SIGN-bit and the least significant bit (LSB) of a global counter (*i.e.*, CNT<0>) are used to determine XOR^n output based on whether XORT (corresponding to the selected operands) have an ODD or an EVEN parity.

We leverage a two-transistor-two-resistor (2T2R) bitcell to have the following key advantages over the typical 1T1R bitcell; 1) inherent ~2X READ sensing margin [66], 2) inherent addressing of circuit non-idealities due to a common-mode (complementary) structure, thereby, providing a *linear* XORG-XORV relation, and 3) aids the use of a reference-less scheme for a memory READ operation. Subsequently, we present two approaches to perform XOR^n; 1) UVTC: a compact scheme using a single reference, and 2) BVTC: a high-speed and energy-efficient reference-less scheme.

7.2.2. PROPOSED MOXOR-CIM ARCHITECTURE

Next, we explain different aspects of the proposed schemes.

BITCELL STRUCTURE AND MEMORY OPERATIONS

Fig. 7.3a shows a typical complementary 2T2R bitcell comprised of two select NMOS pass transistors sharing a common node CSL, and each is connected to bitlines BL and

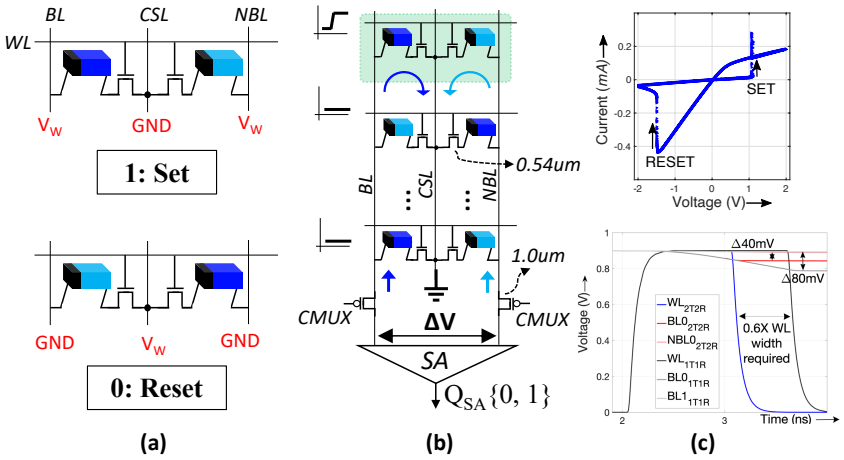


Figure 7.3: (a) Memory WRITE and (b) READ operation using 2T2R bitcells, and (c) IV characteristics and timing diagram during READ.

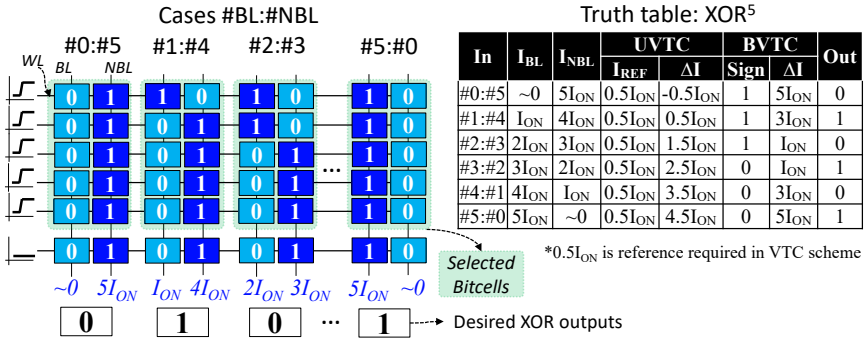
NBL through an RRAM device storing complementary data bits, respectively[1]. RRAM devices can be programmed to such states using the scheme shown in the figure, where a voltage magnitude of V_W is applied in opposite polarity to switch the RRAM device from LCS to HCS, and vice-versa. The switching characteristics of an RRAM device concerning the polarity and the magnitude of the voltage applied are shown in the top half of Fig. 7.3c.

Fig. 7.3b shows a memory READ scheme using differential voltage ΔV= $|V_{BL}$-$V_{NBL}|$ sensing on the respective bitlines BL and NBL. The select transistors are enabled via a row-select wordline signal (WL) and ΔV is developed due to the different conductance values of HCS and LCS. This ΔV is sensed and amplified using a SA, provided a minimum margin of ΔV_{SAmin} is developed between the two input signals of the SA. For further illustration in this chapter, we assume ΔV_{SAmin}=$40mV$ [65, 66]. Owing to the availability of differential BL and NBL signals with 2T2R bitcells, a reference-less READ scheme requires only ΔV=$40mV$, as opposed to a minimum of ΔV=$80mV$ required in single-ended sensing using a reference signal for 1T1R bitcells. As shown in the bottom half of Fig. 7.3c, 1T1R and 2T2R are suffixed to their corresponding WL, BL0, and NBL0 signals, where BL0, NBL0 correspond to a READ 0 and BL1 corresponds to READ 1 in 1T1R bitcells. The figure shows that close to 40% of READ latency can be reduced compared to single-ended sensing involved in reading 1T1R bitcells.

XOR FUNCTIONALITY

Our proposed techniques explore the output pattern of XOR^n as a function of the number of 1's in the selected operands. ODD and EVEN parity cases in the operands are defined as the cases when ODD and EVEN number of 1's are present in the selected operands, respectively. Here, zero number of 1's is considered EVEN. Note that perform-

[1] Note that this bitcell configuration is different from the 2T2R bitcell configuration described in Chapter 5. Here, the pass transistor is connected to the common select line to increase the speed of operations.



Truth table: XOR^5

In	I_{BL}	I_{NBL}	UVTC		BVTC		Out
			I_{REF}	ΔI	Sign	ΔI	
#0:#5	~0	$5I_{ON}$	$0.5I_{ON}$	$-0.5I_{ON}$	1	$5I_{ON}$	0
#1:#4	I_{ON}	$4I_{ON}$	$0.5I_{ON}$	$0.5I_{ON}$	1	$3I_{ON}$	1
#2:#3	$2I_{ON}$	$3I_{ON}$	$0.5I_{ON}$	$1.5I_{ON}$	1	I_{ON}	0
#3:#2	$3I_{ON}$	$2I_{ON}$	$0.5I_{ON}$	$2.5I_{ON}$	0	I_{ON}	1
#4:#1	$4I_{ON}$	I_{ON}	$0.5I_{ON}$	$3.5I_{ON}$	0	$3I_{ON}$	0
#5:#0	$5I_{ON}$	~0	$0.5I_{ON}$	$4.5I_{ON}$	0	$5I_{ON}$	1

*$0.5I_{ON}$ is reference required in VTC scheme

Figure 7.4: Generation of XORV during a XOR^5 operation.

ing an XOR^n operation on ODD and EVEN parity cases results in logic output 1 and 0, respectively.

Fig. 7.4 describes how the input pattern of XOR^n translates to XORV that later serves as an input to our VTC-based schemes, using n=5 as an illustration. The figure shows the dependency of XORV in terms of the discharging currents flowing in the BL and NBL. In the left half of Fig. 7.4, different input cases corresponding to XOR^5 are described at the top of each column, where #BL:#NBL is defined as the number of 1's in the respective bitline sides. In addition, at the bottom of each column, corresponding BL and NBL currents are shown along with the desired XOR^5 output. Naturally, in the case of XOR^5, the total of #BL and #NBL always equals five. With five WLs always activated, naturally the total current corresponding to both BL and NBL is $5I_{ON}$ as well.

Two implementations are proposed to differentiate different input cases from each other; 1) Uni-polar VTC (UVTC): This uses XORV solely related to BL and a reference signal XORREF, and 2) bipolar VTC (BVTC): This uses XORV from both BL and NBL without the use of any reference signal. In UVTC, the polarity and magnitude of difference in XORV and XORREF are utilized. XORREF is the voltage signal corresponding to $0.5I_{ON}$ discharge current of the reference line, to have an equal sensing margin between I_{OFF} ($<<I_{ON}$) and I_{ON} discharge currents in the BL corresponding to #0:#5 and #1:#4 cases, respectively. This implies a difference of $0.5I_{ON}$ discharge current translates to the desired difference for the above two cases, albeit with different polarities. The table on the right half of Fig. 7.4 describes the following; (from left to right) input cases described as #BL:#NBL, BL current I_{BL}, NBL current I_{NBL}, two columns corresponding to UVTC for I_{REF} and ΔI= I_{BL}-I_{REF}, two columns corresponding to BVTC for SIGN and ΔI= |I_{BL}-I_{NBL}|, and the desired output. Next, we show different implementations while utilizing different outcomes of SIGN and magnitude corresponding to different input cases, as described in the table.

7.2.3. DESIGN AND IMPLEMENTATION

MOXOR-CIM implementations follow the following steps; 1) CIM develops an output voltage XORV proportional to the aggregated conductance values XORG in each column of the crossbar. This is achieved in the READ phase, where XORV represents the amount of bitline discharge during the multi-row-select XOR operation, 2) In the decision subphase of the COMPUTE phase, XORV is processed to determine the original condition of

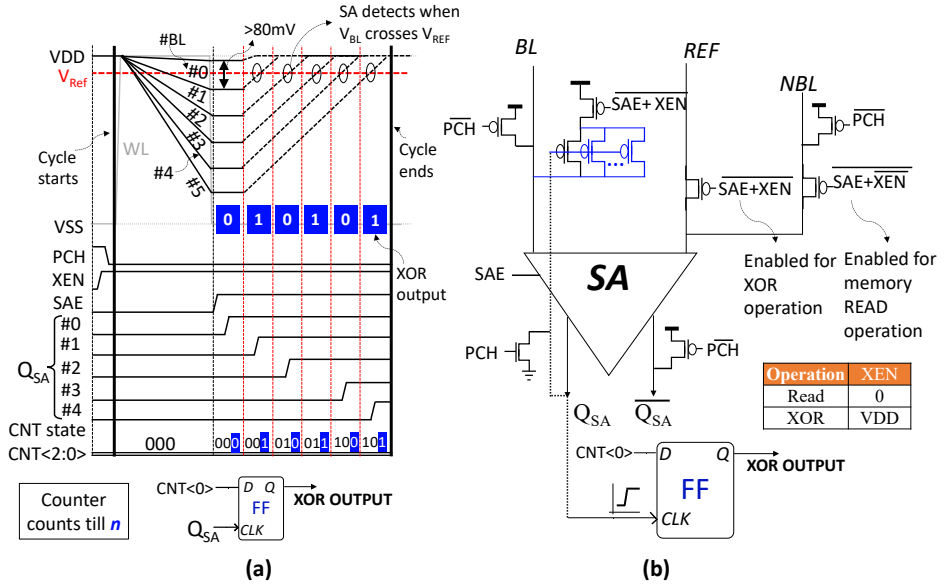


Figure 7.5: (a) Timing diagram and (b) design implementation for UVTC.

the bitlines, and 3) Based on this initial condition, a bitline or both bitlines are charged and/or discharged using pull-up and pull-down circuits, meanwhile, SA determines at what time instant XORT there is an alteration from the original condition, and translates XORT to either ODD or EVEN time sub-phase in the COMPUTE phase. To keep an account of the ODD and EVEN sub-phases of the cycle, a global ripple-carry-counter (GRCC) is used. GRCC is initialized to the RESET state and counts up to a number proportional to the number of operands n during the COMPUTE phase, and its LSB determines the ODD or EVEN sub-phase of the cycle.

UVTC

Implementation of UVTC is presented in Fig. 7.5 based on a single-ended sensing scheme to perform XOR^n, using XOR^5 as an illustration. Different cases in the figure are represented by #BL, where the number represents the number of 1's on the BL side. After the PCH signal pre-charges the BLs to VDD at the start of the cycle, XORV is generated in the READ phase. Given that XEN is enabled to perform an XOR operation, activating SAE enables the SA and the ramp-up operation. XORREF ensures that sufficient $\Delta V > \Delta V_{SAmin}$=40 mV [66] is developed for both #0 and #1 cases to have a deterministic SA functionality, implying at least 80 mV difference is required between the V_{BL}s of #0 and #1 cases. A SA with a complementary initialized output state is used i.e., Q_{SA}=0 and $\overline{Q_{SA}}$=1. The output of the SA Q_{SA} toggles to 1, when V_{BL} crosses XORREF. GRCC counts up from a RESET state and CNT<0> is latched at the time instant Q_{SA} toggles to 1 to generate the XOR^5 output. Fig. 7.5b presents the circuit design to implement this scheme. Note that a tunable cascaded pull-up circuit (not shown in the figure) is used to ensure a *linear* ramp-up of BL voltage.

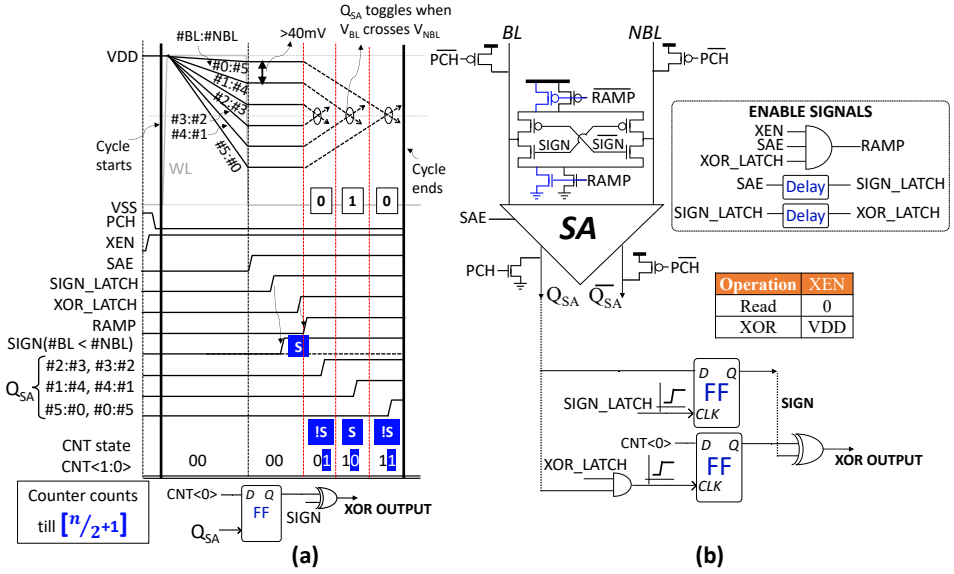


Figure 7.6: (a) Timing diagram and (b) design implementation for BVTC.

BVTC

By exploring a differential approach, a bipolar VTC can be used to design a reference-less implementation, as described in Fig. 7.6. This implies that a 40 mV difference is sufficient between the cases when the number of 1's increases by 1. This reduces the duration of the READ phase and since there is a fixed voltage dynamic range available, compared to UVTC, BVTC can support close to twice the number of maximum operands. As illustrated earlier, the READ phase is nearly reduced by 40% compared to single-ended UVTC. After the READ phase, the voltages developed on both BL and NBL are utilized in this approach. In the COMPUTE phase, first, SA determines the SIGN-bit depending on whether the initial V_{BL} is greater or smaller than V_{NBL}. Following this, depending on this SIGN-bit, one of the bitlines is charged and the other is discharged to arrange a crossing of each other. For instance, SIGN=0 implies initially V_{BL}<V_{NBL}, therefore, BL is charged and NBL is discharged towards each other. Similar to the UVTC approach, the magnitude of the difference in V_{BL} and V_{NBL} is determined when SA toggles again. GRCC counts up from a RESET state till [n/2+1] and CNT<0> is latched at the time instant Q_{SA} toggles again. XOR output is obtained by performing SIGN $\oplus$ CNT<0>.

Fig. 7.6 presents the circuit design to implement this scheme. In the COMPUTE phase, after a sufficient delay margin to allow SA to resolve the SIGN, the SIGN-bit is latched using a delayed version of the SAE signal i.e., SIGN_LATCH. SIGN-bit *selectively* enables the ramp circuit to ramp up or down the bitlines. Tuning the ramp speed can be controlled using calibration header and footer transistors in the ramp circuit, and a cascaded stack of transistors (not shown) is introduced to ensure linear ramp signals. After another sufficient delay of SIGN_LATCH, XOR_LATCH is enabled to activate the latching out of CNT<0> when Q_{SA} is toggled. Performing SIGN $\oplus$ CNT<0> generates the final XOR out-

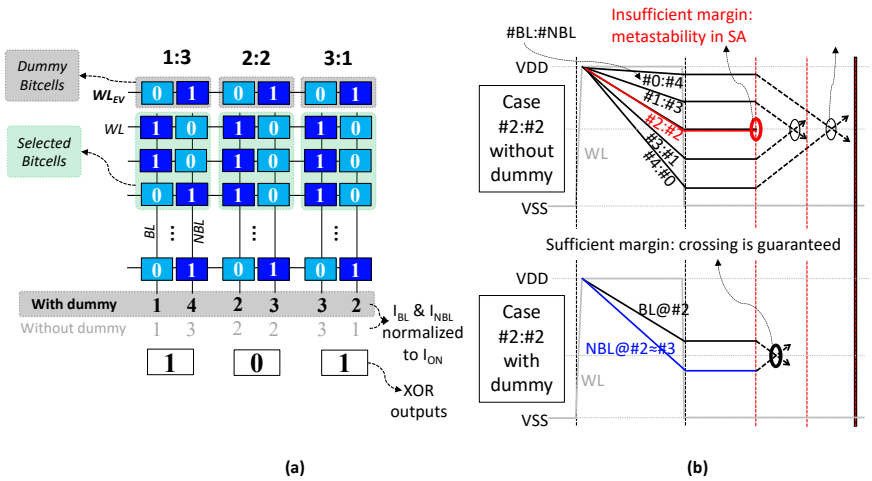


Figure 7.7: (a) Generation of XORV for a XOR^4 operation using a dummy bitcell and (b) the timing diagram.

put. An area overhead of ~30% is incurred in BVTC-based periphery logic compared to the case of UVTC.

However, in the case of EVEN number of operands, it can be seen that in the case when #BL and #NBL are equal, V_{BL} and V_{NBL} are close to each other, hence sufficient margin is not developed. This may lead SA to resolve into an erroneous output. Next, we present a scheme to address this issue.

7

BVTC FOR EVEN PARITY

The case when BL and NBL discharge equally in the READ phase creates $\Delta V << \Delta V_{SAmin}$, which implies that SA can not reliably determine the outcome, as shown in the top half of Fig. 7.7b. In the worst case, on top of giving an unreliable outcome, SA can go into meta-stability, implying it will burn power until the SAE disables the SA and resets its internal nodes. To address this, a dummy row is introduced with the same configuration as the array bitcells i.e., 2T2R. The idea is to bias one of the bitlines with an additional current (equal magnitude of one I_{ON}), which alleviates the case of having equal currents when the data and complementary bits have an equal number of 1's. In Fig. 7.7a, this technique is implemented using a dummy bitcell in each column storing 0 *i.e.,* LCS and HCS on BL and NBL sides, respectively. This implies that there is always a minimum signal margin for the SA to work reliably, as shown in the bottom half of Fig. 7.7b. Note that this dummy row is only activated in the EVEN number of operands, which is predefined information that can be easily configured before the operation. Also, note that this scheme works if 1 is stored in the dummy bitcell as well. Depending on the stored value chosen in the dummy cell, the post-processing required for obtaining the desired XOR output can be configured.

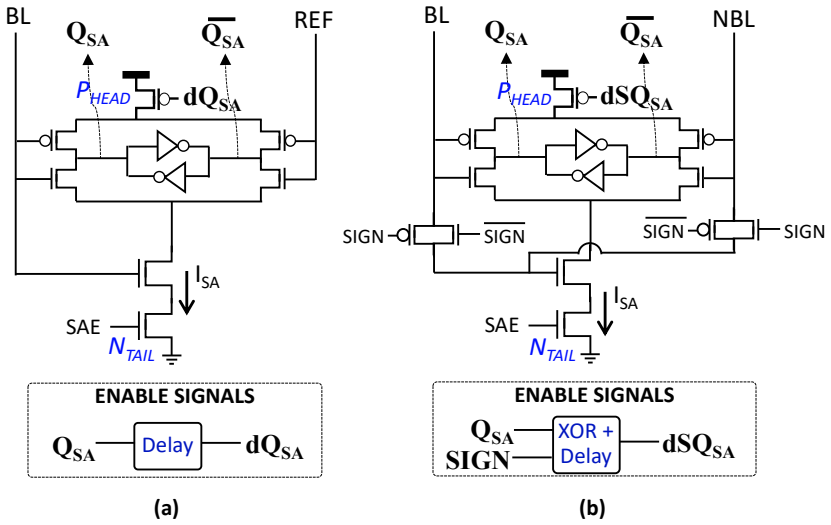


Figure 7.8: Low-power SA designs for (a) UVTC, and (b) BVTC.

LOW-POWER SA

SA takes up the largest power share in any READ scheme, especially in our scheme where SA is activated for a longer duration compared to the conventional schemes. In order to reduce the SA power consumption, two mitigation techniques are introduced; i) adaptive biasing current of the SA, and ii) disabling SA as soon as it resolves the outcome. Fig. 7.8a and Fig. 7.8b present the circuit implementation of the proposed SAs for the UVTC and BVTC XOR schemes, respectively. A two-stage amplification is introduced, where the inner regenerative cross-coupled inverters are minimum-sized transistors. Regarding the regularization of the biasing current I_{SA} of the SA, in the case of UVTC implementation, BL voltage biases the tail transistor N_{TAIL} of the SA. During the COMPUTE phase, BL is charged from a low voltage value towards VDD, and only when V_{BL} reaches close to V_{REF} (which is close to VDD) is when SA is tuned to have maximum amplification gain. On average, almost 70-80% of the power is expected to be saved compared to a scenario where SA is strongly biased for the entirety of the COMPUTE phase. In the case of BVTC implementation, SIGN-bit dependent biasing is introduced where a ramping-up signal (BL or NBL) is connected to N_{TAIL} with similar power savings. Fig. 7.9a shows instantaneous power consumed by SA in the BVTC case performing XOR[16]. This is illustrated using the operand cases #1, #5, and #9, where I_{SA} quadratically increases and only reaches the required biasing current near the crossing of the two bitlines. Regarding the disabling of SA, in the case of UVTC, a header P_{HEAD} is added to the cross-coupled inverting amplifier-based circuit. P_{HEAD} disconnects the power supply as SA makes a decision *i.e.,* Q_{SA} is set to 1 following the detection of V_{BL} crossing V_{REF}. Similarly, in the case of BVTC, a SIGN-dependent signal is generated to disable P_{HEAD}. On average, a further 50% of the power is saved compared to a scenario where SA is enabled for the entirety of the COMPUTE phase. This can be seen in Fig. 7.9a where the instantaneous I_{SA} drops to near zero right after Q_{SA} toggles to 1.

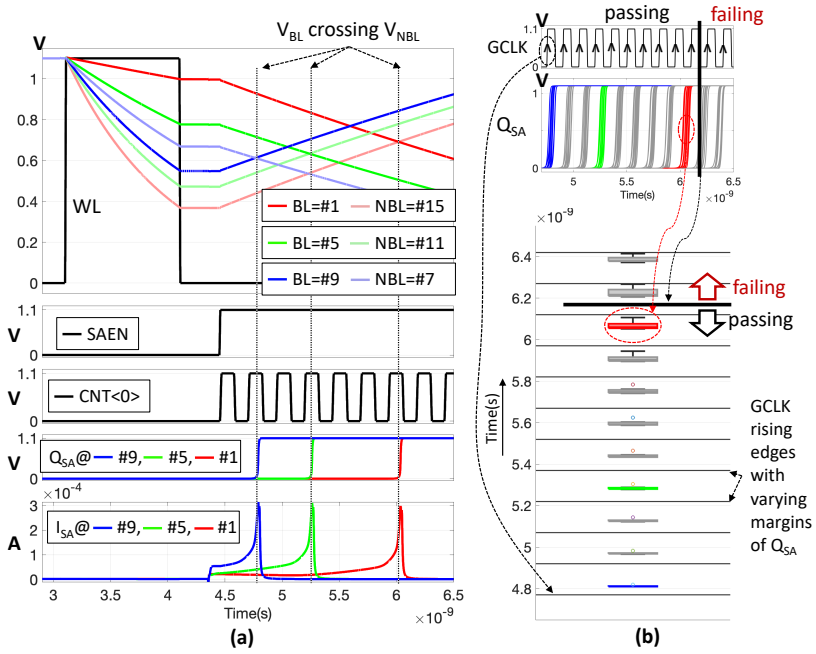


Figure 7.9: (a) Timing diagram for BVTC-based XOR[16] showing bitline pairs and low-power SA (biasing I_{SA}) with adaptive biasing and disabling SA upon decision. (b) All possible Q_{SA} distribution with worst-case corners failing after 8 cycles (16 operands).

7

7.3. RESULTS

7.3.1. SIMULATION SETUP

Table 7.1 summarizes the design specifications used for our circuit-level analysis. SA is designed to accurately sense a minimum ΔV of 40 mV. T_{SAmin} is defined as the time required by the SA to make a decision after the input voltage polarities are reversed. With 3σ variation analysis, a minimum of 126 ps is required to reliably determine the SA outcome. A pessimistic requirement of 150 ps cycle time of the GRCC is configured to allow reliable SA functionality. Fig. 7.10 shows the layout view of different HfO_2/TiO_x

Parameters	Specifications
CIM Array	512x512 (256Kb)
RRAM Device	HfO_2/TiO_x [112]
HCS/LCS	100 $K\Omega$ / 3 $K\Omega$ (±20 %)
Voltage supply	1.1 V (±10 %)
CMOS (variations)	SVT, 40 nm TSMC (3σ)
Temperature	-40°C to 125°C
ΔV_{SAmin}, T_{SAmin}	40 mV, 126 ps

Table 7.1: Design parameters.

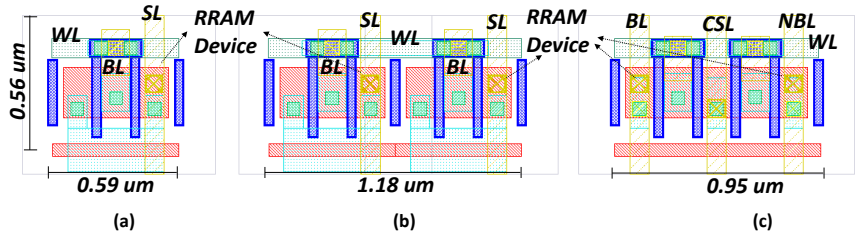


Figure 7.10: Layout footprint for (a) conventional 1T1R bitcell (b) two 1T1R bitcells and (c) 2T2R bitcell in our proposed scheme.

RRAM-based bitcells, where the proposed 2T2R bitcell consumes 61% more area than the conventional 1T1R bitcell but consumes 24% less area than two 1T1R bitcells. Pass transistor (NMOS) is 540 nm/40 nm and it typically provides a resistance of 1.1 $K\Omega$. Vertical (Horizontal) wire resistance adds up to 0.4 Ω (0.8 Ω) and total capacitance to 0.3 fF (0.6 fF) per unit bitcell.

The simulation and comparison with the conventional schemes are extracted using the same setup; e.g., technology node, LCS and HCS resistances of RRAM, and wire parasitics.

7.3.2. CIRCUIT-LEVEL RESULTS AND COMPARISON

Fig. 7.9b shows the variation analysis of all possible input combinations of XOR^{16} while highlighting three cases where operands are denoted by the number of ON devices on the BL side *i.e.,* #1, #5 and #9 (also highlighted in Fig. 7.9a). GCLK running at 6.67 GHz (period of 150 ps) as determined by T_{SAmin} implies that CNT<0> toggles every 150 ps. This requires that the toggling of Q_{SA} must occur within this 150 ps period to ensure correct functionality. Note that since there are an EVEN number of operands, the dummy bitcell is enabled and it injects I_{ON} on the BL side and I_{OFF} ~0 on the NBL side. The top of Fig. 7.9b shows the variation of the rising edge of Q_{SA} with respect to the rising edge of GCLK and these are also denoted by boxplot and horizontal lines, respectively, in the bottom half of the figure. It can be seen that 3σ variation fails beyond operating with 16 operands while having sufficient sensing margin up to 8 GCLK cycles. This ensures that up to 8 and 16 operands can be reliably performed using UVTC and BVTC schemes, respectively.

For comparison, we present data corresponding to four solutions to perform XOR^n; i) cascading OR^2 $\oplus$ NAND^2 [64, 107], ii) cascading XOR^2 or multi-operand XOR [49, 106, 108, 221], iii) proposed UVTC, and iv) BVTC. The combined latency and the combined energy related to the READ phase *i.e.,* including WL decoding and driver delay, bitline discharge, T_{SA}, setup/hold time of address and configuration bits, bitline pair pre-charge before the start of the next cycle, are defined as T_{READ} and E_{READ}, respectively. In addition, the combined latency and combined energy dedicated to the COMPUTE phase are defined as T_{COMP} and E_{COMP}, respectively. In short, total latency and energy to perform XOR^n is $T_{READ} + T_{COMP}$ and $E_{READ} + E_{COMP}$, respectively.

In terms of latency, T_{READ} is similar to that of conventional memory READ or two-operand logic since there is no additional circuitry in the critical path for the UVTC case and is 40% less for the BVTC case, as illustrated earlier. T_{COMP} includes the following;

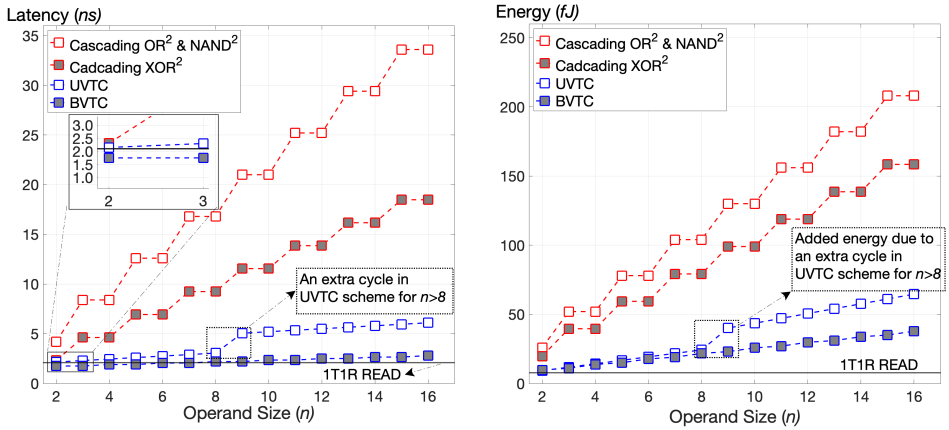


Figure 7.11: (a) Latency and (b) energy comparison. Cascading OR^2 & $NAND^2$ [64, 65, 109] and cascading XOR^2 [106–108, 110, 217, 218, 221–223].

i) the number of GRCC counts i.e., $n \times T_{SAmin}$ in the UVTC case and $\lfloor n/2+1 \rfloor \times T_{SAmin}$ in the BVTC case, and an additional ii) time required by the SA to reliably determine the SIGN a.k.a an additional T_{SAmin} and a small delay before the ramping sub-phase of the COMPUTE phase starts. Therefore, total XOR^n latency is $T_{READ}+n \times T_{SAmin}$ and $0.6 \times T_{READ}+\lfloor n/2+1 \rfloor \times T_{SAmin}$, respectively, for UVTC and BVTC implementations. Whereas, in the case of conventional approaches, cascading OR and NAND lead to $n \times (T_{READ}+T_{SA})$ and cascading XOR leads to $\lfloor \frac{n+1}{2} \rfloor \times (T_{READ}+T_{SA})$. Similarly, in terms of energy, there is no penalty in E_{READ} comparable to conventional memory READ or two-operand logic. The impact of enabling SA i.e., E_{COMP} for a longer time on overall energy is significantly reduced thanks to the low-power techniques.

The resulting latency and energy to perform XOR^{16} in a column are reported in Table 7.2 and are also reported in Fig 7.11a and Fig 7.11b, respectively. A performance improvement of 3.4-9.5× (2.1-5.6×) and an energy improvement of 4.4-13× (2.6-7.7×) is achieved by utilizing BVTC (UVTC) scheme compared to the state-of-the-art logic accelerators. In addition, up to an operand size of 16 (8) can be reliably supported using BVTC (UVTC) scheme, whereas prior-art is limited to an operand size of 3 [49, 221]. Between our two MOXOR-CIM implementations, a performance and energy efficiency improvement of 1.6× and 1.7×, respectively, can be achieved at an expense of ~30% additional area-overhead in the periphery logic (~6% in the CIM tile) in the BVTC case compared to UVTC.

7.4. CONCLUSION

This chapter presents MOXOR-CIM which realizes novel voltage-to-time conversion (VTC) based approaches to perform multi-operand XOR reliably in a single cycle for RRAM-based CIM architectures. In our schemes, CIM generates a voltage signal proportional to the operand data, and two VTC-based approaches are presented to determine if the

Design Metrics	[106]	[107]	[221]	[108]	[64]	[49]	**UVTC**	**BVTC**
Technology (nm)	90	65	45	45	90	65	**40**	**40**
Max. XOR Ops/cycle	2	2	4	2	2	3	**8**	**16**
Latency (ns)	20.8*	41*	16*	17*	48	24.5	**6.2**	**3.6**
Energy (fJ)	272	362	131	156*	176	-	**64**	**38**
Energy/Op (fJ)	17	22.7	8.2	9.7	11	-	**4**	**2.4**
Reference signals	2	2	3	2	1	3	**1**	**0**
Variation Tol.	No	No	Yes	Yes	No	Yes	**Yes**	**Yes**

Table 7.2: Comparison of our proposed design with prior techniques. * indicates normalized to RRAM-based READ of 2 ns and 15 fJ.

operand has an ODD or an EVEN parity. The two schemes proposed offer a desirable trade-off between computing efficiency and area overhead. Comparison results using 512x512 RRAM-based CIM show an improvement of up to 12× in performance and 5.5× energy efficiency, while also extending the maximum number of operands supported in a single cycle to 16.

8

Outlook and Discussion

This chapter puts into perspective the proposed CIM-based logic accelerators. It covers how the different challenges addressed, design methodology and optimization techniques have progressively improved the efficiency metrics as well as scalability in terms of array size and operand size over the course of the thesis. It also discusses possible design considerations required to incorporate various design features such as utilizing other memristor technologies or conventional memories, different bitcell configurations and data types, CMOS technology scaling, and scalability.

This chapter is partially based on [64–68].

8.1. INTRODUCTION

This chapter presents a brief outlook on the proposed CIM-based logic accelerators and how the progress can be interpreted during the course of the thesis work. Section 8.2 describes the critical challenges targeted by each accelerator that can be seen to evolve with each work, where the evolution can be defined by various factors; 1) minimum sensing margins involved, 2) enhanced periphery logic to address additional challenges, 3) complexity of the tasks involved, etc. It also presents a trend where this progress can also be seen in terms of the efficiency metrics achieved by each work. Thereafter, it provides a comparison with the prior-art solutions as well as with the preceding work. Section 8.3 presents a brief account of the discussion points on the feasibility of utilizing the proposed solutions and identifies several design explorations. Finally, Section 8.4 concludes the chapter.

8.2. OUTLOOK

This section highlights the key challenges addressed by each work and their key concepts while focusing on how each design has improved the state-of-the-art as well as the preceding works. The challenges addressed are summarized in Fig. 8.1 and the relative efficiency metrics are presented in Fig. 8.2.

Chapter 3 [64] describes the first work on this thesis that targets improving the efficiency of CIM-based logic operations using PCM devices. Prior to this work, a series of logic operations were performed either using two-/multi-operand stateful logic schemes or using two-operand non-stateful (via two-row activation read) logic. As described earlier, stateful logic schemes involve programming devices with long latency and high energy consumption which are highly unsuitable for memristor devices known to have low endurance. Whereas, in the case of using two-operand scouting logic, each of the partial results is required to be written back into the memory to perform further computing. In short, several programming cycles were involved in both cases. The proposed cascaded logic allows the capturing of partial results within the periphery logic, thus, alleviating the need to re-program the memristor devices and additionally, provide a fast and energy-efficient system. In this work, the limits of the array size are also investigated with highly dense selector-less bitcells (1R) and resistive wires in the presence of sneak path issues. Real-world application related to health care validates the concept.

Chapter 4 [65] presents the work that addresses challenges related to performing logic operations using STT-MRAM devices where the two binary states exhibit extremely low resistance ratios and are highly conductive. In addition, the wire RC delay mismatch due to the location of selected bitcells, process variations in the SA, and programming variations in STT devices drastically reduce minimum sensing margins between the critical states of OR and AND logic operations. In this work, an adaptive scheme involves reference blocks built using STT devices that are strategically placed within the bitcell that generate reference signals to improve the sensing margin. The scheme is validated where cascaded logic (Chapter 3 [64]) is used to perform XOR operations in realizing real-world BNN applications. In other words, the preceding work on cascading logic is orthogonal to this work and can be utilized for the proposed CIM-based logic operation using STT devices.

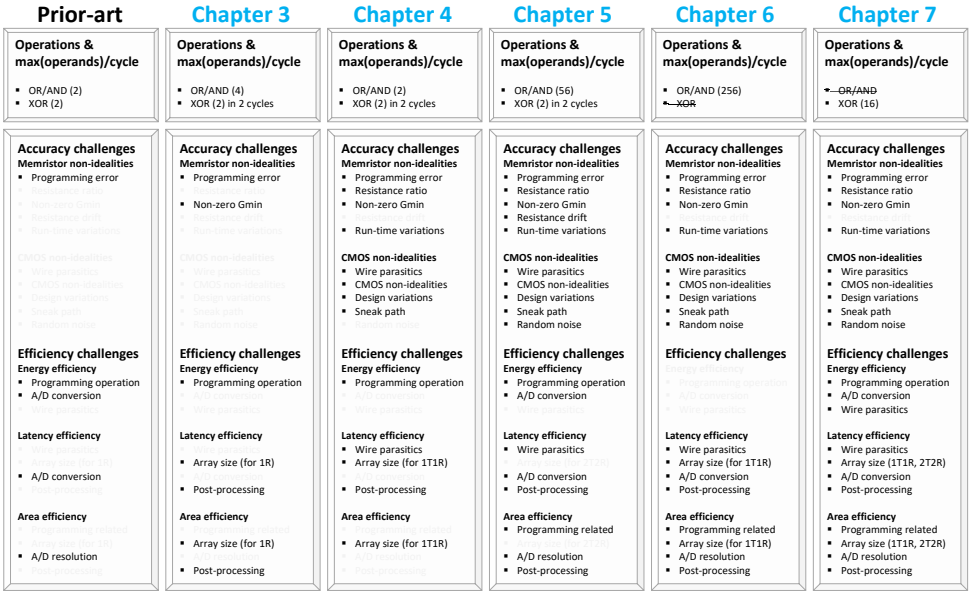


Figure 8.1: Summary of the challenges addressed by the prior-art and the logic-based CIM accelerators proposed in each chapter. Additionally, the operation types and number of operands that can be performed per cycle.

Chapter 5 [66] presents the work that addresses challenges related to performing two- and multi-operand (N)OR and (N)AND operations in a single cycle, where the resistance ratio is reasonably high (such as for RRAM and PCM devices). In the prior art as well as the preceding works of this thesis (Chapter 3 and Chapter 4 [64, 65]), (N)AND typically involves differentiating two conductance states with a theoretical ratio of "two" in the case of two-operand operations and reduces exponentially with increasing operand size (this ratio goes on as 2, 3/2, 4/3, 5/4... and converges to "one"). On top of that, dedicated reference blocks required to generate reference signals increase the complexity and limit the efficiency of the system. In addition, a large number of operations and high voltage of operations diminish the reliability of memristor devices. This work introduces a differential sensing scheme by arranging cells in a complementary structure and a row of dummy bitcells in such a way that it inherently acts as a reference to enable a high sensing margin. It implements highly variation-sensitive (N)AND operation using complementary-input (N)OR operation for better accuracy and efficiency with low-voltage read schemes and comprehensively reduces the number of operations for high reliability.

Chapter 6 [67] describes how multi-operand (N)OR operation can potentially accelerate real-world applications related to testing and improve the fault coverage of dense RRAM-based memory units. Several components of the design related to multi-operand (N)OR described in Chapter 5 [66] are utilized while also introducing novel components to facilitate the high-speed testing of RRAMs and better fault coverage. This chapter

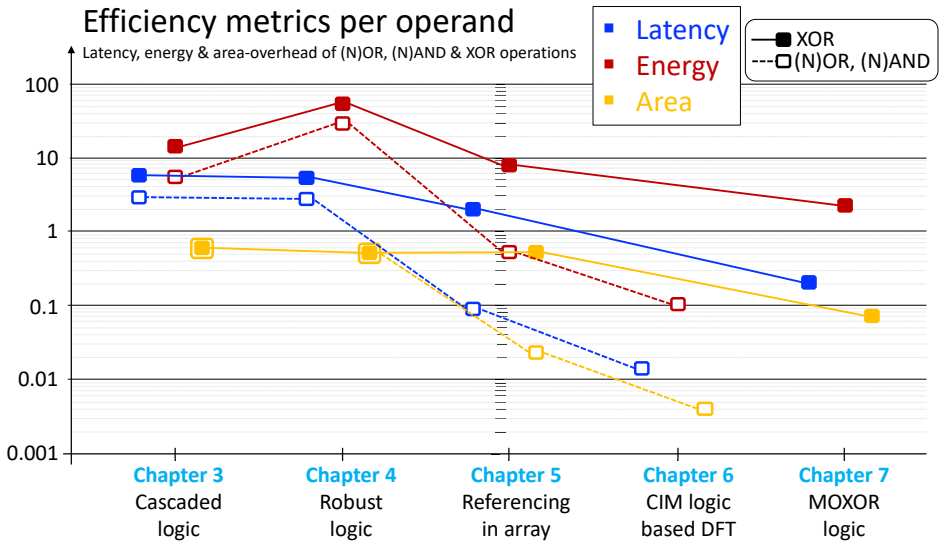


Figure 8.2: Summary of the relative efficiency metrics per operand of the logic-based CIM accelerators proposed in each chapter.

presents in detail these components such as a reconfigurable row decoder to generate the address selection patterns, reconfigurable reference signal generator, and tunable read voltage and duration of operations, to develop high sensing margins that guarantee the detection of *unique* RRAM faults. The CIM-logic-based DFT schemes are validated with circuit-level simulations based on a post-layout netlist that facilitates the execution of high-quality test algorithms at a low cost.

Chapter 7 [68] presents circuit solutions for XOR that are one of the most complex and widely used logic operations in applications related to error-correcting codes (ECC), encryption, and Big data. In addition to all the non-idealities and efficiency challenges addressed by two- and multi-operand (N)OR and (N)AND operations in the preceding works of this thesis, multi-operand XOR brings in several new challenges. The most critical challenge is that unlike multi-operand (N)OR and (N)AND operations proposed in Chapter 5 and Chapter 6 [66, 67], where critical states deciding the logic output are either all 0's or all 1's, XOR functionality demands the output to change in an alternate manner with an incremental number of 1's among the operands, which is extremely challenging with the current methods. Using and configuring multi-reference signals and complex SA to differentiate these states is not only inefficient, large, and power-hungry but also is not scalable in terms of operand size. This chapter reports two VTC-based schemes, a compact one-reference, uni-polar VTC (UVTC)-based scheme and a high-speed reference-less bipolar VTC (BVTC)-based scheme, with a configurable number of operands up to 16. The SA design is extremely power-efficient, thanks to low-power schemes described in the work.

8.3. DISCUSSION ON DESIGN EXPLORATIONS

This section discusses the applicability and adaptability of the proposed works with different design configurations.

MEMRISTOR TECHNOLOGIES

The design techniques involved in most of the works are applicable to other memristor technologies. Cascaded logic [64] is in the periphery and is not restricted to any memristor technology. Although robust logic design in [65] focuses specifically on STT devices that suffer from extremely low resistance ratios, it is applicable to other memristor technologies as well. Note that multi-operand designs in [66–68] are only applicable memristor technologies with sufficiently high resistance ratios such as in RRAMs and PCMs. Nevertheless, 1) CIM-based DFT techniques described in [67] can still benefit from having high fault coverage when using STT devices, and 2) two-operand (N)OR and (N)AND designs described in [66] can be extremely useful for STT-based CIM. Nevertheless, each memristor device offers some unique challenges or different resistance values/ratios that can be investigated while utilizing the proposed design solutions.

MEMRISTOR-BASED BITCELL CONFIGURATIONS

Most of the design techniques are easily applicable to other bitcell configurations. Designs proposed as cascaded logic [64] and robust logic in [65] are not restricted to any bitcell configuration. Whereas referencing in array design in [66] and BVTC design in [68] are presented using 2T2R bitcells, any bitcell configuration that can provide differential signals can benefit from these schemes. CIM-based DFT techniques in [67] are applicable to any bitcell configuration that involves read operation based on the difference in BL discharge currents (or developed BL voltages). Although the most common bitcells are 1T1R and 2T2R, other potential bitcell configurations with better prospects for certain tasks can be investigated.

MULTI-BIT CAPACITY

This is only applicable to the use of CIM-based DFT schemes for accelerating the testing of storage class RRAMs described in [67] since other techniques are related to binary logic operations. The multi-bit capability of memristors has been greatly explored to increase the storage density. In terms of resistance ratio, we can assume a reduction in the difference of effective resistance states; this is because more quantized resistance states (>2) have originated from a similar range that is associated with only two states in a 1-bit RRAM. However, this reduction only reduces the minimum possible ΔV_{BL} of the two critical states which can be alleviated using a longer cycle of operation. In short, a reduced maximum number of operands in the NOR operation or an increase in cycle time can still offer fast testing and facilitate improved fault coverage. These potential concepts can be investigated toward developing efficient testing schemes for multi-bit storing memories.

ADVANCED CMOS TECHNOLOGY

For advanced CMOS nodes, the wire issue becomes even more extreme and it dominates every aspect of the circuit and physical design flow. Fortunately, most of the design techniques will still benefit from advanced CMOS nodes. Cascaded logic [64] being a digital circuit and referencing techniques in robust logic [65] will simply scale with technology. Circuit techniques similar to techniques described in the robust logic chapter can become increasingly essential with wire delay mismatch issues. Efficiency improvements when utilizing multi-operand design techniques in [66–68] are expected to further increase. This stems from the fact that the latency and energy of the sensing component *i.e.* SA does not scale down as steeply as compared to digital logic. Therefore, operating SA once in the proposed schemes as opposed to multiple times in prior art solutions would scale up the reported gain factors in terms of efficiency. In addition, with voltage down-scaling, altering the V_{WL} can have a greater impact on the conductance of the pass transistor (V_{GS} is closer to threshold voltage) when utilizing DFT schemes presented in [67]. Many of the above hypotheses can be explored.

CONVENTIONAL MEMORY TECHNOLOGY

Most of the design techniques are applicable to conventional memory technologies such as SRAM, DRAM, and Flash. Design techniques involved in cascaded logic [64] and [65] are orthogonal to memory technology. Whereas referencing in array design in [66] and BVTC design in [68] are presented using 2T2R bitcells, SRAM bitcells inherently provide the required differential signals to utilize these techniques. DRAM and Flash can utilize techniques to perform multi-operand (N)OR operations described in [66] and multi-operand XOR operations described as UVTC in [68]. Multi-operand design techniques in [66] for high-speed testing and high fault coverage can easily improve the efficiency of testing conventional memories. However, this may need some additional techniques to handle, for instance, simultaneous charge/discharge by pull-up/pull-down transistors storing 1 and 0, respectively, on the same column. Fortunately, since most of our proposed schemes are voltage-based sensing, with voltage SA like typical SRAMs, and many of our work using the same SA as standard 5T SA. Therefore, this gives enough motivation to perform logic near or in conventional memories that can be inspired by our work. At least conceptually. For instance, an 8T single-ended read can be somewhat similar to a read scheme adapted to logic schemes. This, along with novel techniques to reduce the high dynamic power due to the path created in the above scenario, can be explored further.

SCALING OF ARRAY SIZE

The main challenge to reliably perform logic operations with increasing array size is dealing with the wire parasitics. Using dense selector-less bitcells in Chapter 3, the issue related to the sneak path is highlighted where finite parasitics of the wires limit the maximum size of the memory array that can reliably support logic operations. Therefore, the subsequent works utilize a selector device to avoid sneak path issues and several ways to generate adaptive reference signals to high sensing margins in the presence of finite wire parasitics. With the number of columns, the horizontal WL signal suffers

from IR drop which typically weakens the pass transistors, adaptive referencing schemes described in [65–68] essentially mitigates these effects as described in the respective chapters. Since the references in each of these design solutions are generated using dummy bitcells, the reference signals are inherently adaptive to the increased number of columns. With the number of rows, the proposed solutions to generate the required reference signals can be easily calibrated after fabrication. For instance, in multi-operand logic design schemes [66, 67], V_{WL} can be tuned via calibration to generate the required reference signals. Similarly, in multi-operand XOR schemes [68], the ramp speed can be tuned accordingly using calibration techniques. Nevertheless, the limit to which the wire parasitic mismatch and increasing IR drop with the number of rows and columns can be dealt with can be explored.

SCALING OF OPERAND SIZE

One of the main contributions of the proposed solutions in [66–68] is maximizing the number of operands that can be reliably performed in a single cycle. The solutions have explored high resistance ratios of RRAM devices (or PCM devices) which when increased can scale the operand size further. However, the resistance ratio, dynamic voltage range, and finite minimum sensing margins required by the SA to reliably determine the outcome limit the maximum operand size for a given memory size and memory technology.

8.4. CONCLUSION

This chapter puts into perspective how the proposed schemes are related and how they push state-of-the-art solutions and their preceding works in this thesis to achieve better efficiency and scalability. The key challenges addressed, the targeted logic operations, and the associated applications are summarized for each of the proposed logic accelerators. The targeted logic operations include two- and multi-operand (N)OR, (N)AND, and XOR operations which can be performed in one or several cycles in arbitrary or fixed sequences. A brief discussion on how the proposed design concepts can be applicable and how they can be explored while investigating various design choices *i.e.,* memory technology, bitcell configurations, scalability in terms of array size, and operand size.

PART III: CIM-BASED ARITHMETIC ACCELERATORS

This part includes the following chapters.

- ***Chapter 9**: Low-power VCO-based ADC for CIM*
- ***Chapter 10**: Compact low-power CCO-ADC for CIM*
- ***Chapter 11**: Voltage-to-time based ADC for 2T2R RRAM*
- ***Chapter 12**: Outlook and discussion*

This part is based on [1, 69–71].

9

LOW-POWER IFC-BASED ADC FOR CIM

CIM-based ADC designs dedicated to converting analog MAC output into digital domain are typically bulky, power-hungry circuits that are prone to design variations and therefore, play an important role in determining the computing efficiency and accuracy of CIM architectures. In this chapter, we present a scalable and reliable integrate and fire circuit ADC (SRIF-ADC) design for CIM architectures, suitable for stringent power and area constraints. We devise a technique to stabilize the node receiving analog inputs that allows more rows to be activated at the same time, thereby increasing the operand size of input vectors. This allows better scalability in terms of higher parallelism of operations. We employ a self-timed variation-aware design approach and design measures to drastically reduce read disturb of memristor devices that address reliability issues related to the ADC design. In addition, we present a compact, built-in sample-and-hold circuit to replace the large-sized capacitance and built-in weighting technique to alleviate the need for post-processing. For MAC operation, our simulation results show that we can improve the computational parallelism by 3X, and ADC conversion speed and energy efficiency are improved by 2X and 11.6X, respectively, compared to the state-of-the-art design.

This chapter is based on [69].

9.1. INTRODUCTION

The overall computing efficiency of CIM systems depends not only on proficiency in analog computation but also on the efficacy of inter-conversions of digital and analog data streams [225]. Conversions performed by ADCs are very critical and challenging due to 1) Analog signals have low noise margin and hence, can lead to erroneous output [226]; 2) Analog computation heavily relies on memristors and CMOS selectors strength (e.g., for 1T1R), therefore these variations induce variation in output current [227, 228]; 3) Quantization error in ADCs increases as the number of activation levels increases for higher resolution/accuracy [229]; In addition, the area/power increases drastically at higher accuracy, while speed reduces[229]. Fig. 9.1 shows that the ADC alone typically dominates CIM die area (>90%) and power consumption (>65%); this highlights the importance of ADC implementation for CIM architectures that could target e.g., machine learning algorithms such as Conventional Neural Networks (CNN) or Deep Neural Networks (DNNs).

Several ADC designs have been proposed to digitize the analog output of CIM crossbar arrays such as MAC, that form an integral part of implementing neural networks. However, these prior solutions force an extreme degree of trade-offs among speed, operand size scalability, accuracy, and power consumption which limits the efficiency and thus, the prospect of CIM architecture. For instance, *successive approximation register* based ADC (SAR-ADC) proposed in [231], can support large MAC operand size and offers pipelining of operations. However, it employs sample-and-hold ($S+H$) and several digital-to-analog converters (DACs) that require large-sized device capacitances, registers, and analog comparators, resulting in a large area and high power design. On the other hand, the *integrate and fire circuit* (IFC) based ADC proposed in [125] has its own set of limitations such as slower conversion speed, no pipeline features, and support only a limited number of operand sizes *i.e.* low scalability to larger sized operands. In addition to the above inefficiencies, other prior ADC designs based on the IFC working principle are bulky and power-inefficient as they use multiple sense amplifiers per column [121, 232] Operational-amplifier-based current ADC (Op-ADC) proposed in [126], improves this scalability. However, it suffers from higher latency, and power consumption and also does not support the pipeline feature. Some prior-art CIM macros do not include a con-

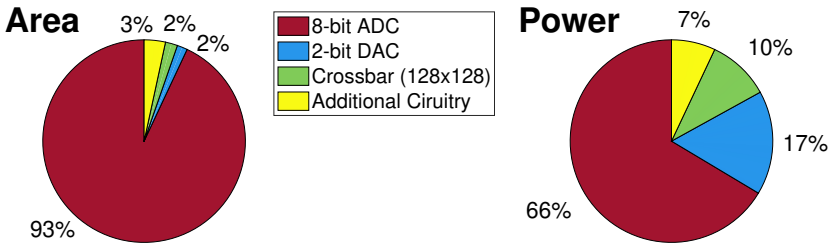


Figure 9.1: Area and Power share of CIM design blocks [230].

ventional ADC approach, where the column current is only converted to voltage and is fed to the next stages [226, 233]. While the approach is power-efficient, this approach is noise-prone and can only be used for approximate computing with high tolerance to signal sensitivity. Several other ADC designs such as [35, 121] involve flash ADC-like approaches, which are fast but are significantly expensive in terms of area and power consumption. A common limitation of all prior ADC designs is that they do not consider the impact of process, voltage, and temperature (PVT) variations. Furthermore, these ADC designs have overlooked the possibility of read disturbance of the storage cell. Therefore, there is a decisive requirement for a reliable and efficient ADC design that is suitable for CIM architecture, and useful for data-intensive applications.

In this chapter, we propose new design strategies for IFC-based ADC to make it reliable, scalable, and energy-efficient. The contributions related to the proposed ADC design are:

- Traditional ADC designs use post-processing to aggregate multiply-and-accumulate (MAC) results obtained from different columns. This aggregation requires shift-and-add ($S+A$) while taking into account different weights to be associated with different column MAC results. In this chapter, we propose a *built-in weighing technique* that aggregates weighted MAC results obtained from different columns. Therefore, our approach does not require post-processing.

- Prior ADCs use charge-RESET-charge scheme [47, 125, 234] involving current evaluated by a capacitor to perform integrate-and-fire functionality. We propose a discharge-SET-discharge scheme that stabilizes the current which improves the accuracy and speed of the conversion.

- Prior CIM-based ADC designs do not consider the impact of global variations at a memristor crossbar level that significantly degrades the accuracy of MAC operation. We propose *variation-aware self-timing path (STP)* technique that uses a dummy column to capture variation impacts in each cycle. In contrast to the traditional STP approach[235], we also include the output of the ADC and predetermined states of the RRAM device in the dummy column to offer robustness against global variations.

- High gain differential amplifier (HgDA) makes the first stage of conversion linear by virtually fixing the bitline voltage. This improves the accuracy of the conversion. While some of the prior ADCs use DA to stabilize bitline voltage [47, 226, 233], the novelty of our design lies in combining this technique with other blocks to provide different features such as built-in sample-and-hold, built-in weighing, variation-adaptability and ensuring a small voltage across memristor devices.

In addition to the above contributions, we further optimized our ADC in terms of power and area by using compact *built-in* $S+H$ (current mirror and a simple switch) that enables pipelining of operations instead of traditional ADCs using a large-sized capacitor. Our simulation results show that for a MAC operation on a 64X4 matrix array size, we can scale up to 3X operand size as well as improve the latency and energy by 2X

and 11.6X, respectively, compared to the state-of-the-art IFC design. In terms of reliability, we significantly enhance the resilience against process variations by performing extensive 3σ Monte Carlo simulations as well as considering voltage and temperature variations and 30 % variation in memristor device variations. In addition, our proposed design can improve the read disturb rate from 100 to $> 10^8$ cycles for read operations.

The organization of the chapter is as follows. Section 9.2 illustrates our novel design methodology and key features/components. Section 9.3 presents results of the proposed ADC design, and comparison with prior ADC designs. This section also presents frequently used arithmetic/functional units based on our proposed ADC design methodology. Section 9.4 concludes the chapter.

9.2. PROPOSED SRIF-ADC

In this section, we propose novel design solutions based on the IFC technique. We describe an ADC design methodology and consequently, present circuit design solutions which aim to solve the inefficiencies suffered by prior ADC designs.

9.2.1. DESIGN METHODOLOGY

A typical ADC comprises three stages. The breakdown of the process into three discrete design components is shown in Fig. 9.2. The input to our ADC is received by the sensing stage, which intermediately quantifies the input analog value and thus requires the following features and attributes; 1) The circuit does not affect the input value during sensing *i.e.* circuit component receiving current I_j with low input resistance will least affect the current flowing through it. Additionally, this implies that the circuit allows an expected change in current values with a change in operand values. Here, if we consider

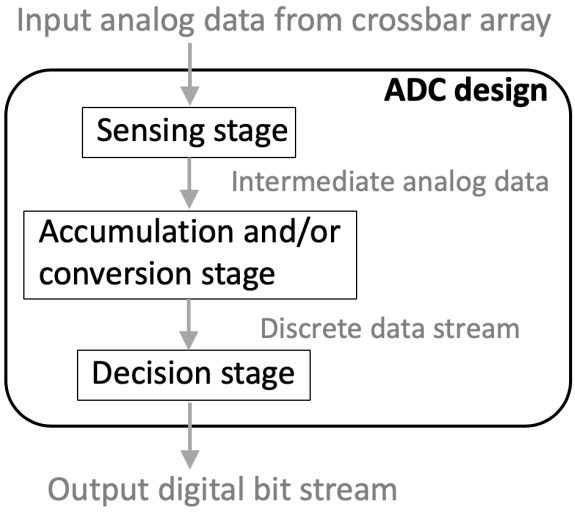


Figure 9.2: Three stages of a typical ADC design.

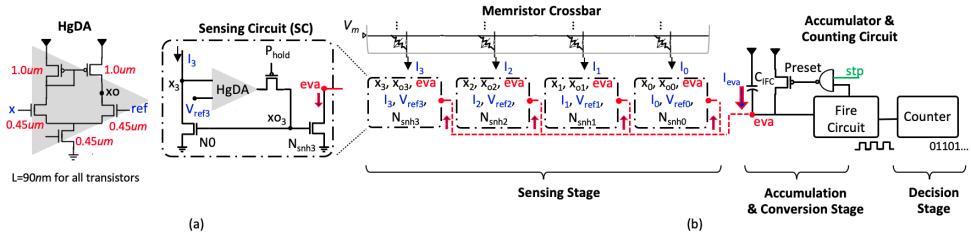


Figure 9.3: Circuit design implementation of the proposed scalable and reliable integrate and fire ADC (SRIF-ADC). (a) Sensing circuit: Receives current column I_j ($j = 3$), HgDA (left) stabilizes node x_3 to V_{ref3}, P_{hold} provides sample-and-hold. (b) SRIF-ADC design for evaluating four columns simultaneously. Sensing circuit (SC) units in different columns are equipped with different sets of $\{V_{refi}, N_{snhi}\}$ to provide column-wise weighting. The cumulative current received from four columns is accumulated and integrated, followed by a fire circuit and finally a counter.

R_{off}/R_{on} to be sufficiently high, the current should (pseudo) linearly increase with an increase in operand values (number of R_{on}). 2) The circuit disconnects the crossbar from the ADC unit if and when the input value is sensed and information is transferred to the next stage. This saves redundant time and energy spent during computation.

Next to the sensing stage is the accumulation and conversion stage, which is responsible for the actual conversion of quantified analog value to discrete digital samples. This stage requires the following attributes; 1) Throughout the conversion, it does not affect the input analog value in order to have proportional input-output characteristics. 2) If multiple input analog values are converted simultaneously, the accumulation must be seamless in terms of the expected aggregated result. The last stage *i.e.* decision stage receives the discrete digital samples and converts them into digital output bits to complete the process. This stage is generally a digital counter and it remains intact in our methodology. It is worth mentioning here that since CIM involves analog components, variation in the strength of CMOS devices utilized in ADC, drivers and access transistors along with memristor devices needs to be taken into account in the design methodology.

In this chapter, we improved the overall efficiency of the ADC by modifying the sensing stage and the accumulation & conversion stage.

9.2.2. DESIGN COMPONENTS

Fig. 9.3 presents the circuit implementation of the ADC design stages described above. We evaluate four consecutive crossbar columns (as a group) to demonstrate the operation of our circuit design.

SENSING STAGE

Built-in Sample-and-Hold *(S+H)*: This feature is introduced in IFC design to allow pipelining of operations. However, instead of using area inefficient large capacitance to implement *S+H* functionality, a novel circuit is proposed. Figure 9.3a shows the circuit implementation of *S+H* feature. The proposed circuit is a current mirror with input current

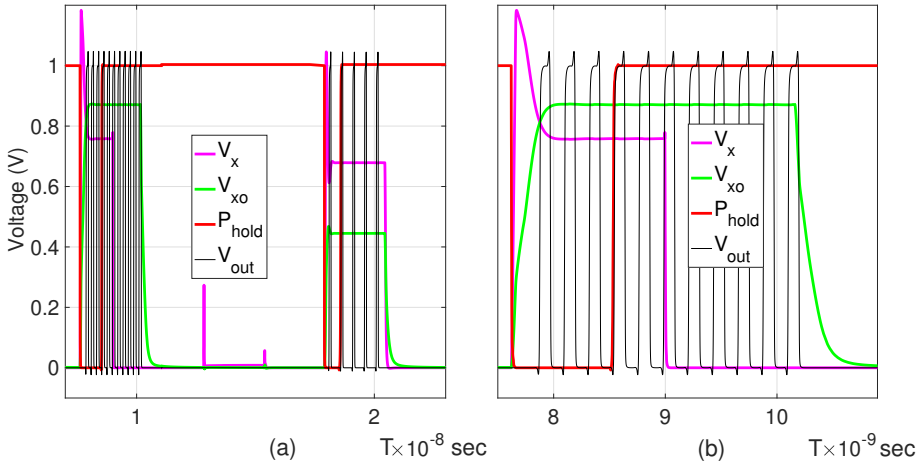


Figure 9.4: (a) The working of built-in *S+H* technique with two consecutive conversion cycles. (b) Magnified view of the first conversion cycle. The width of the P_{hold} signal *i.e.* the settling time T_{set} is sufficient enough to ensure appropriate settling of $x3$ and $xo3$ voltage nodes.

the same as the current I_j *(j=3)* received by the original NMOS $N0$. The current flowing through $N0$ is now mirrored at NMOS N_{snh3} and the integrate circuit is now connected to N_{snh3}. This implies that N_{snh3} is now responsible for charging the capacitor C_{IFC}. Evidently, the gate voltage V_{xo3} (at node $xo3$) of N_{snh3} is not influenced by the charging and discharging of node eva throughout the evaluation phase, as shown in the Fig. 9.4. Given the fact that voltage V_{xo3} quantifying the current I_j does not vary, the memristor crossbar can be disconnected from the peripheral evaluating circuit once a voltage is established at node $xo3$. Hence, *S+H* feature is achieved by disconnecting node $xo3$ from $x3$. A strong ($\sim 2-3\,\mu$m) PMOS P_{hold} is introduced to disconnect these nodes after a pre-deterministic settling time (T_{set}) $\sim 500\,ps$. T_{set} is determined by the time V_{xo3} takes to reach the settling voltage. This time depends on the memristor bitcell currents, resistance, and capacitance of the bitlines.

Input-Output Linearizer: One of the most important contributions of SRIF-ADC is fixing node x that improves the linearity of input-output characteristics. The premise of the proposed technique is that the two inputs of a *high-gain differential amplifier* (HgDA) connected through a negative feedback loop tend to converge at equal voltages. Referring to Fig. 9.3a, we provide a fixed V_{ref3} as a negative terminal input of HgDA such that V_{x3} (node x for $3th$ column is denoted by $x3$) tends to reach ($\sim V_{ref3}$). This tunes V_{xo3} appropriately to allow $\sim I_j$ current through $N0$. To make sure that V_{x3} reaches $\sim V_{ref3}$ at all possible scenarios, a HgDA is implemented using a higher length node to provide a large range of V_{xo3} for a very small differential input *i.e.* $(V_{x3} - V_{ref3})$. With appropriate sizing and biasing voltage, a mere $\pm 10\,mV$ shift in V_{x3} is achieved covering almost the entire range of possible cases. The two extreme cases considered are all memristors with mem-conductance G_{off} and G_{on} in a column, respectively.

Fig. 9.5a shows a comparison between node x of IFC and our proposed design. For

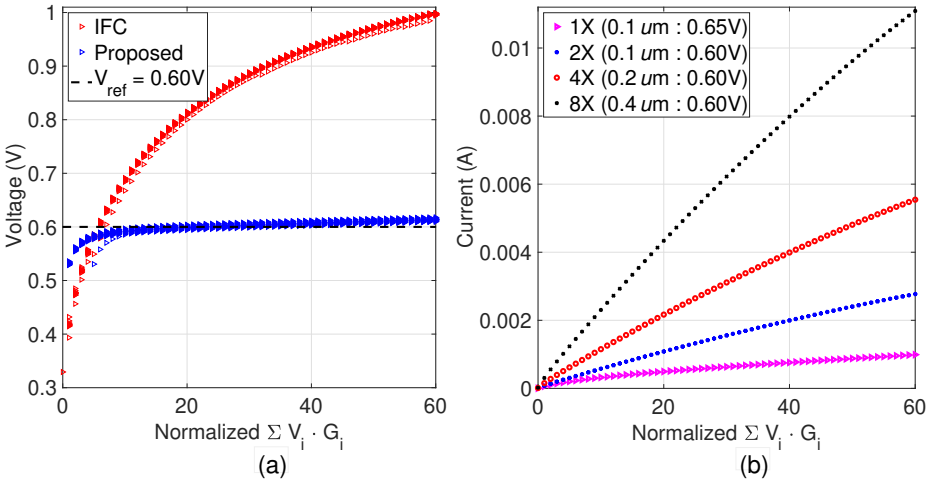


Figure 9.5: (a) Voltage at node x held nearly constant by HgDA design v/s prior IFC design. (b) An accurate ratio of 8:4:2:1 current ratio is maintained using a built-in weighting technique.

this analysis, V_{ref3} for HgDA is kept at 0.60 V, while read voltage V_r is kept at 0.7 V. Understandably, other than the first few cases where all memristors in a column are mostly in R_{off} state, node x in our design (in blue) is within 1 % of the intended voltage. This node x in IFC (in red) however experiences a drastic change as the number of R_{on} increases, which in turn disturbs the functionality after a few rows.

Yan *et al.* proposed to use a similar approach in [47], where it uses an operational trans-conductance amplifier (OTA) to stabilize the node x. However, the intention of this ADC is not to have a linear output *i.e.* many-to-one mapping of the number of pulses with mem-conductance. The focus of our work is to have a linear one-to-one mapping of input-output characteristics. Similarly, the use of DA & feedback approach by *et al.* M. F. Chang in [236, 237] refers to read and smart write-verify schemes i.e. it only concerns operations related to single device select per column. In contrast, we implement a multi-operand MAC unit that fixes the bitline voltage to have a linear increase in bitline current with effective mem-conductance in a column.

The above techniques are the key to enabling the implementation and working of the subsequent features to further improve the efficiency of our proposed design.

Built-in Column Weighting: The aggregation of outputs (# of spikes counted by an ADC) received from all the columns is usually performed using shift-and-add *(S+A)* circuitry. Since such a sequential addition is a slow process and involves large adders and supporting data registers, two novel techniques of weighting based on bit position are proposed to allow parallel processing of different columns. Thus, making the use of *S+A* circuitry redundant. Fig. 9.3 illustrates column-wise weighting techniques using weight-proportional current mirror sizing ratio and voltage reference of HgDA. These built-in weighting techniques utilize the premise of the IFC design *i.e.* digital output is linearly proportional to the current flowing through a memristor crossbar column. Therefore, the current flowing through consecutive columns is manipulated using two techniques.

Weighting is achieved by sizing N_{snh} accordingly, to get weight-proportional currents at the output. Weighting is also achieved by having different reference voltages (V_{ref}) connected to the negative input of the HgDA.

The sensing circuit is shown in Fig. 9.3a is connected per column with different N_{snh} size. Utilizing the weight-proportional sizing technique, four current mirrors are used for four consecutive columns with sizes of $N_{snh0} : N_{snh1} : N_{snh2} : N_{snh3}$ in the ratio 1:2:4:8 while keeping the same input NMOS $N0$ size. The drain terminals of N_{snh0}, N_{snh1}, N_{snh2} and N_{snh3} are connected together (as node eva), and therefore the summation of these weighted currents is used to charge C_{IFC} (DA receiving input voltage V_{eva} inside the fire circuit).

Using weight proportional voltage reference technique, different reference voltages can be maintained in consecutive columns to obtain the required ratio of voltage across the memristors. For example, keeping $V_{read} = 0.7\,V$, V_{ref0}, V_{ref1}, V_{ref2}, V_{ref3} can be kept at $0.65\,V$, $0.6\,V$, $0.5\,V$ and $0.3\,V$. The fact that HgDA virtually fixes V_x at V_{ref}, the voltage across the memristors in these columns are virtually maintained at $50\,mV$, $100\,mV$, $200\,mV$, $400\,mV$, respectively. This allows a current multiplication corresponding to the weights represented by each column position, similar to the outcome achieved by sizing N_{snh} previously illustrated.

Since the two weighting techniques can be exclusively implemented, they can be integrated together to combine more columns as well or a smart choice can be made based on the CMOS technology at hand. An efficient combination based on our analysis for integrating these techniques is illustrated with $\{S_{N0} : V_{read}\} = \{10\,um : 0.7\,V\}$ for aggregating four columns. We paired different $\{S_{Nsnhi} : V_{refi}\}$ as $\{0.1\,um : 0.65\,V\}$, $\{0.1\,um : 0.60\,V\}$, $\{0.2\,um : 0.60\,V\}$ and $\{0.4\,um : 0.60\,V\}$ for $0th$, $1st$, $2nd$ and $3rd$ bit positions, respectively.

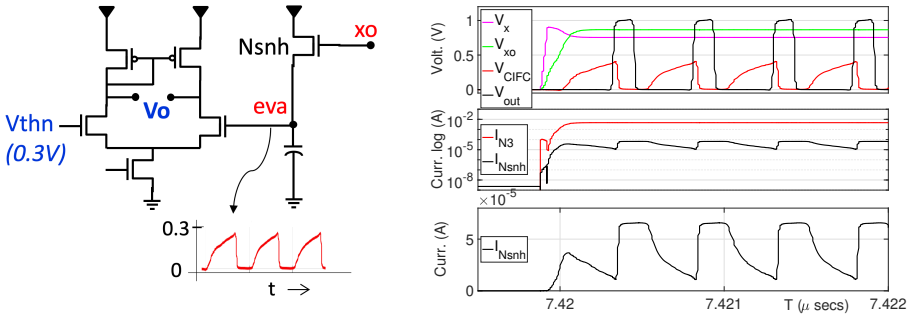
A similar technique of using weighted current mirrors can be found in [35]. However, we combine different voltage references and sizes which improves the scalability of this technique. Other circuit implementations include [238], but the design is rather expensive since it would require 15 sense amplifiers and 15 corresponding reference circuits.

Fig. 9.5b shows the scaling of currents with weights of the column position included. An accurate binary ratio of X, 2X, 4X, and 8X of current flowing through columns $0th$, $1st$, $2nd$, and $3rd$ is obtained, respectively as the number of R_{on} increases in a column.

ACCUMULATION AND CONVERSION STAGE

Fire Circuit Stabilizer: DA used in the IFC design [192] is NMOS-based *i.e.* input voltages are applied at the gate terminal of an NMOS device. To further improve the linearity of the proposed ADC, PMOS-based DA is used instead. Fig. 9.6 and Fig. 9.7illustrate the working of the previous and the proposed redesign.

The modification proposed involves discharging capacitor C_{IFC} from, say $VDD \rightarrow VDD - V_{thp}$ instead of charging it from, say $0 \rightarrow V_{thn}$, to trigger an output spike. Here, VDD is the supply voltage of IFC. V_{thn} ($VDD - V_{thp}$) is the threshold or trigger voltage for NMOS (PMOS)-based DA. Due to this modification, resetting the node eva involves pre-charging it to VDD instead of pre-discharging it to GND. Not to mention, the reset signal is inverted for this altered arrangement.

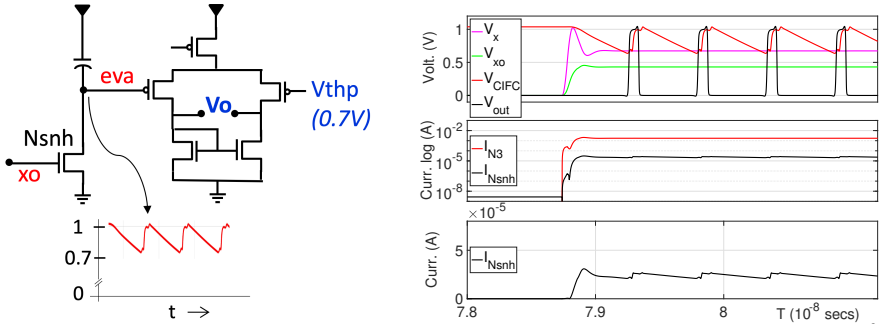


(a) (b)

Figure 9.6: (a) NMOS-based DA as a fire circuit forces N_{snh} to operate in partial weak saturation and triode region. (b) Waveforms related to NMOS-based DA. (c) Instead of using NMOS-based DA, PMOS-based DA as a fire circuit enables N_{snh} to operate in strong saturation mode. and (d) Waveforms related to PMOS-based DA.

The intention is to keep the current charging or discharging C_{IFC} to be nearly constant throughout the evaluation phase. While charging C_{IFC} (placed between N_{snh} and GND as shown in Fig. 9.6a) through N_{snh}, change occurs in both V_{DS} and V_{GS} of N_{snh}. This implies that current provided by N_{snh} fluctuates considerably as its V_{GS} switches between V_{xo} and $V_{xo} - V_{thn}$. However, if C_{IFC} is discharged (placed between VDD supply and N_{snh} as shown in Fig. 9.7a) through N_{snh}, change during the evaluation only occurs in V_{DS} (V_{GS} remains constant at V_{xo}). Since N_{snh} is always in strong saturation mode, a V_{DS} change of VDD to $VDD - V_{thp}$ does not change the discharging current considerably. Important voltage/current signals, the voltage received by NMOS, and a PMOS-based DA are shown in Fig. 9.6b and Fig. 9.7b, respectively.

Subplot 1 shows important voltage values responsible for the conversion. Subplot 2 shows current ratios flowing through $N3$ and mirrored N_{snh3} which needs to be roughly maintained at 100. Subplot 2 in the log scale (and subplot 3 in the linear scale) shows



(a) (b)

Figure 9.7: (a) NMOS-based DA as a fire circuit forces N_{snh} to operate in partial weak saturation and triode region. (b) Waveforms related to NMOS-based DA. (c) Instead of using NMOS-based DA, PMOS-based DA as a fire circuit enables N_{snh} to operate in strong saturation mode. and (d) Waveforms related to PMOS-based DA.

the difference in current flowing through a current mirror that is responsible for charging/discharging C_{IFC}. Variation in current flowing through C_{IFC} is significantly less in the PMOS-based DA in comparison to NMOS-based DA. This ensures a more linear increase in the number of output spikes with an increase in $\sum_{i=1}^{m} V_i \cdot G_{ij}$ for a given column j with m number of rows. Another point worth mentioning here is that keeping $V_{ref} = VDD - V_{thp}$ ensures a more accurate and faster current mirror, as now N_{snh} always works in saturation mode. Yan *et al.* proposed PMOS-based DA as well [47], however, it suffers from the same issue. This is because the scheme related to the switching of capacitor voltage is similar to the one in IFC, where the capacitor C_{IFC} charges to V_{thn} and reset to GND after every generated pulse.

Self-Timing Path (STP): Analog computation using IFC critically depends on the strength of CMOS devices and memristor devices in the crossbar used. Digital output (# of spikes) depends on the current flowing through a memristor crossbar column. However, analog components such as current mirrors receiving the current and DAs are highly sensitive to CMOS device process, voltage, and temperature variations. Consequently, the number of spikes varies significantly due to such variations. Total time (T_{total}) is key to quantifying the digital output *i.e.* the total time available to generate spikes. In this regard, in order to improve the accuracy of our ADC, a novel self-timing technique is proposed that adaptively varies the evaluation time (T_{total}).

The premise of STP is to calculate the time a column with known resistance states takes to produce a known output and then use that time interval as T_{total} for the required operation. As shown in Fig. 9.8, a dummy column is used (we take all mem-resistance values as R_{on} in this dummy column) to implement this technique in the same crossbar array. As soon as the output number of spikes for this dummy column reaches the

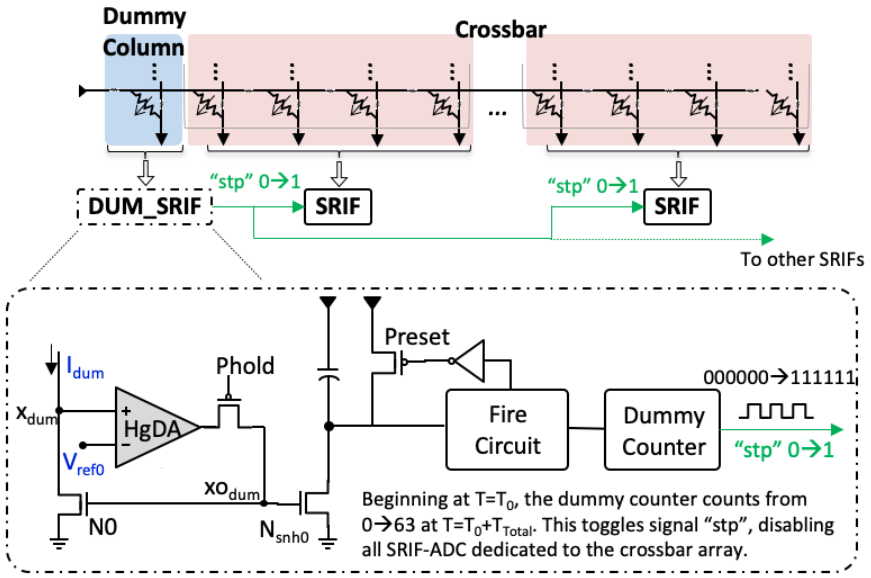


Figure 9.8: The working of the self-timing path (STP) technique to address variations in our proposed ADC design.

applied number of non-zero row voltages (here, all row voltages are kept at V_{read}), it toggles the signal *stp* at T=T_{total}, which disables the other columns to stop counting. This dummy column thus provides a variation-aware T_{total}. The acquired T_{total} is adaptive to global variations of CMOS and memristor devices, cycle-to-cycle variations in temperature, and peripheral/core voltage supplies. It is worth mentioning that device-to-device variations are not captured by this technique.

DECISION STAGE

In our design, the decision is made by counting the number of generated spikes which can be performed by a digital counter. We utilized a ripple carry counter common to all the columns that are combined using the built-in weighting technique. The counter needs to be fast enough to count the maximum of pulses possible.

9.3. RESULTS

This section presents a quantitative comparison of SRIF-ADC as a MAC operating unit with state-of-the-art ADCs described in Section 9.1 while taking into account control and other peripheral circuitry utilized by these designs. It includes simulation setup, and variation considerations, followed by result comparison. Apart from the MAC unit, we propose the implementation of adder and comparator designs to showcase the versatility of our design methodology.

9.3.1. SIMULATION SETUP

Table 9.1 lists the design specifications used for our analysis. Simulations are performed using HfO_2/TiO_x based RRAM device [239] that is assembled in a 1T crossbar memory structure. For our analysis, we choose an RRAM device capable of storing 1-bit data as two stable high (200 $K\Omega$) and low resistive states (2 $K\Omega$), with R_{off}/R_{on} of 100. The simulation setup is equipped with 90 *n*m TSMC CMOS device models to conduct accurate comparison.

Parameters	Specifications
RRAM Device	HfO_2/TiO_x [112]
R_{off} / R_{on}	200 $K\Omega$ / 2 $K\Omega$
RRAM Variation	30 % parametric (without noise)
Read Voltage	0.7 V with ±10 % variations
CMOS Technology	90 *n*m TSMC
Process Corner	Typical
Temperature	27°C
CMOS Variation	3σ (without noise)
ADC Voltage	1.0 V with ±10 % variations

Table 9.1: Design parameters.

The operation performed to evaluate ADCs is a MAC operation of two operands, each of length 64 elements but each element of width *1-bit* and *4-bits*, respectively *i.e.* $(64_1 \cdot 64_4)$. Implementing $64_1 \cdot 64_4$ in Fig. 9.9a implies that one operand of dimension 64×4 is stored in the memristor crossbar $(G_{3m},..G_{0m})$ where $m \in \{1,64\}$ corresponds to row index and other operand of dimension 64×1 is applied as voltage vector $(V_1,...V_{64})$ to this crossbar.

All prior ADCs under consideration are used in a time-multiplex manner *i.e.* four columns require four operating cycles, meaning an ADC is shared with four columns to have a (pseudo) iso-area comparison with SRIF-ADC. However, since SRIF-ADC is equipped with a built-in weighting feature to combine four columns, it requires only one operating cycle.

To investigate the accuracy against CMOS device technology variations, 1000 Monte Carlo samples equivalent to 3σ variation is considered at 27°C using a Typical CMOS process corner. Simulations also include ±10 % variation in nominal supply voltage to take into account voltage variations. Additionally, a 30 % memristor device variation is accounted for as well. This 30 % device variation mentioned in Table II implies percentage variation from the nominal value in physical parameters used in the Verilog-A model [239] that governs the static and dynamic behavior of the RRAM device. Some of the primary physical parameters described in the model are the minimum and maximum concentration of oxide ions, radius, and length of the RRAM device. Noteworthily, noise and distortion-related sources are not considered in our setup due to a lack of concrete RRAM-based experimental data related to these sources.

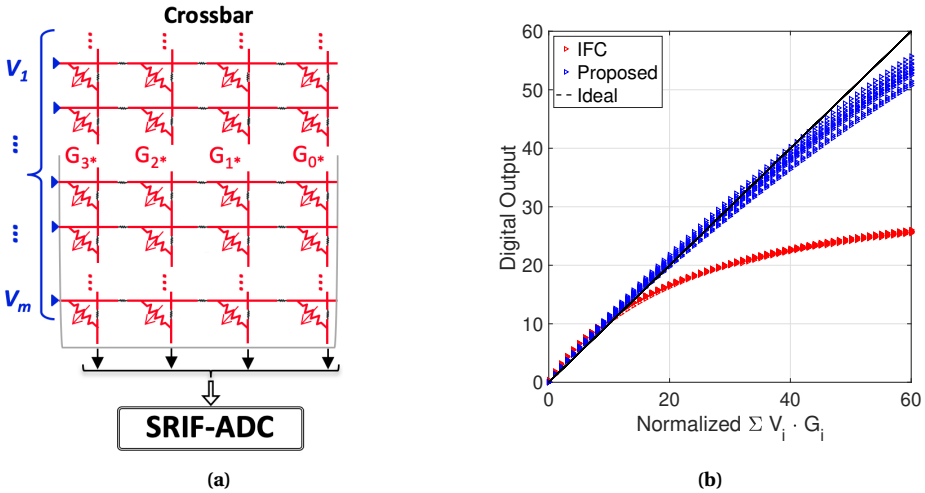


Figure 9.9: (a) Crossbar configuration to perform MAC operation by applying a voltage vector $(V_1,...V_m)$ to the crossbar array. (b) MAC operation is performed on a single column. The input-output characteristics are significantly improved compared to state-of-the-art IFC. In terms of functional crossbar size *i.e.* scalability, our design can support up to 3X the size supported by IFC design.

9.3.2. SIMULATION RESULTS

MAC operation is performed by applying an array of row voltages (each with possible value 0 or V_r) to the memristor crossbar, as shown in Fig. 9.9a. Possible combinations of outputs by changing stored mem-resistance values in the crossbar array and row voltages are simulated, and expected versus obtained outputs are shown in Fig. 9.9b. Also in this figure, the output of state-of-the-art IFC is shown to compare its trend with SRIF-ADC. Subsequently, we simulate $64_1 \cdot 64_4$ and evaluate the overall efficiency of our proposed design.

ENERGY

Built-in *S+H* feature reduces the time duration of large current I_j that flows through the crossbar columns as it disconnects the memory and evaluation unit (ADC) after a small pre-determined settling time ($T_{set} \sim 4ns$). Also, the current flowing through ADC is considerably reduced as we utilize current mirrors with low output to input current ratios (nearly 0.01X, $\sim \frac{size(N0)}{size(N_{snh})} X$). This technique is extremely advantageous for saving energy in cases where the memristor device used has a high R_{on} conductance state. Moreover, since a small virtually fixed voltage ($> 0.1\,V$) is maintained across the memristors, an extremely low current flows through all the columns of a memristor crossbar. Hence, we save energy consumed in both crossbar and periphery units. There is a small overhead $\sim \frac{1}{n} X$ of using a dummy column to implement the STP feature, as one column is used for an array of n columns. When compared to IFC, SRIF-ADC is 11.6X more energy efficient.

LATENCY

Apart from the pipelining offered by the in-built $S+H$ circuit, the built-in weighting technique allows parallel aggregation of digital outputs obtained from different columns. Therefore, time spent on multiplexing an ADC to evaluate a number of columns in a sequential manner, and subsequently post-processing to aggregate these column results using shift-and-add operation is eliminated. PMOS-based DA enables the discharging process of C_{IFC} much faster than the charging process of NMOS-based DA. As mentioned earlier, this is due to the fact that N_{snh} operates in a high saturation region throughout the evaluation phase and a weak saturation/triode region while utilizing PMOS and NMOS-based DA, respectively. With these techniques, we achieve a latency of 200 ns, and as compared to IFC, we improve the conversion speed by 2X.

AREA

We save significant area occupied by $S+A$ circuitry and registers utilized for shifting and adding per-column results. In addition, a large capacitor size is generally required by prior arts such as SAR-ADC, and IFC design ($\sim 150\,f$F). Given that the voltage supply of 1.0 V is applied across the memristors, the active column current is relatively high (in our design, we have a voltage drop of 0.1-0.2 V). Now, $S+H$ capacitor in these ADCs allows this large flow of active current throughout the evaluation phase. This necessitates a large $S+H$ capacitor as a small capacitor will have an extremely small time constant

(RC) for such a large current. Lest, this $S+H$ capacitor will *charge and reset* at an incredibly fast speed which can surpass by far the frequency of the counter counting the generated pulses. Moreover in SAR-ADC, several capacitors are utilized in implementing DACs to generate all combinations of $LSB*2^{(n-1)}$, $LSB*2^{(n-2)}$ ... $LSB*2^{0}$ (for an n-bit ADC) equivalent reference voltages. Original large-sized NMOS $N0$ ($\sim 10\,u$m) receiving the crossbar current is kept as such while added NMOS N_{snh} is kept extremely small ($\sim 0.1\,u$m). Hence, the additional mirror circuit has negligible area impact. Additional area of these techniques include 2 MOSFETs per reference voltage, therefore n different reference voltages will require $2n$ MOSFETs. Hence, a negligible area overhead to implement these techniques. Overall, we save 2.7X area utilization with respect to IFC.

THROUGHPUT

Due to the compact size and high energy efficiency, our proposed design offers a high throughput solution with one SRIF-ADC per four memristor columns, while computing these four columns simultaneously. This implies that with no sharing of ADC among multiple columns, no time-multiplexing is required. In terms of giga operations per sec ($GOPS$), our proposed design offers 2.56 $GOPS$ and in terms of picojoules of energy spent by the ADC per operation (pJ/Op), it offers 0.92 pJ/Op as compared to 1.28 $GOPS$ and 39.1 pJ/Op, respectively offered by IFC.

RELIABILITY

Read disturb: Since a small fixed voltage ($\sim 0.1\,V$) is maintained across RRAM devices as compared to a large voltage ($\sim 1\,V$) in other ADC designs, the read disturb margin improves significantly. In addition, a read voltage is applied for a reduced time as the built-in S+H feature allows a short (T_{set}) of RRAM devices before it can be isolated from

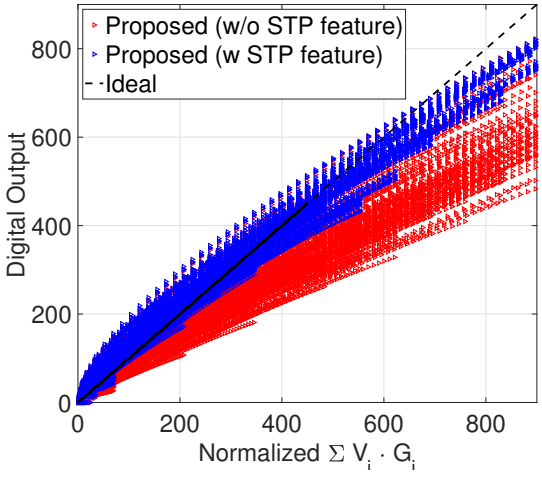


Figure 9.10: MAC operation $64_1 \cdot 64_4$ results are plotted with (blue) and without (red) variation-aware design using the STP feature. The deviation from the ideal value is significantly reduced.

the ADC unit and eventually removed. This results in an increase from several 100 cycles to $> 10^8$ cycles for read operations [240].

Design variation: High accuracy against process, temperature, and voltage (PVT) variations is achieved using the STP feature, as the dummy column in the crossbar tracks these global variations. The drastic improvement in accuracy is shown in Fig. 9.10. Understandably, RRAM device variations with global phenomena, such as resistance drift, are also captured using the dummy column. Our ADC works seamlessly against 3σ Monte Carlo simulations using Typical CMOS process corner at 27°C, 10 % variation in voltage, temperature range of 0-85°C and 30 % memristor device related variations.

RESOLUTION

Referring to Fig. 9.3 and 9.5, once V_j is virtually fixed at V_{ref}, I_j follows linearly with $\sum_{i=1}^{n} V_i \cdot G_{ij}$. This results in a linearly proportional charging time of the capacitance C_{IFC} by I_{eva} with memristor column resistance. Hence, SRIF-ADC produces a linearized number of spikes. PMOS-based DA used to accumulate and integrate current further linearizes the transfer function. As a result, in terms of size scalability, SRIF can reliably support 3X operand size that performs MAC operations with respect to IFC. Note that our proposed design can distinguish up to 64 distinct current (analog) levels, and thus can support a crossbar array consisting of 1-bit RRAM devices with 64 rows. However, in order to keep the same number of distinguishable levels, the maximum supportable operand size reduces if multi-bit RRAM devices are used. The above gain of 3X in operand size stands even if we use a multi-bit storage RRAM device. This is due to the fact that the maximum supportable operand size for SRIF-ADC reduces by the same factor as IFC. This is also valid for any other ADC design.

9.3.3. COMPARISON WITH PRIOR WORKS

Table 9.2 shows the comparison of IFC, SAR, Op, and SRIF against previously discussed design metrics. The results substantiate the improvement claims made qualitatively above. For an apple-to-apple comparison, we referred to prior designs equipped with 90 nm CMOS technology.

As compared to IFC, SRIF offers 2X speed, > 3X range, 2.7X area-efficiency, 11.6X energy-efficiency, an improved read disturb margin, and 42X reduction in pJ/Op. As compared to SAR, SRIF offers 144X area efficiency, 56X energy efficiency, and 14X reduction in pJ/Op with a reduced speed of nearly 0.4X but with a similar range and accuracy. As compared to Op, SRIF offers 2X speed, 4.5X area efficiency, 23.6X energy efficiency, 13X reduction in pJ/Op and an improved read disturb margin with similar range and accuracy. Besides these improvements, SRIF also offers pipelining of operations that IFC and Op do not offer.

Our proposed ADC provides an efficient solution that implements MAC operation in a group of memristor-based crossbar columns. In this manner, we have included a subset of a fully-fledged general-purpose vector-matrix multiplication (VMM), which is the most fundamental computational unit in today's hardware systems implementing machine learning algorithms. In other words, we presented a group of four columns as a simultaneously computed block in a crossbar of an arbitrary number of columns. This

implies that this sub-block of four columns can be repeated as many times, and similar efficiency can be easily scaled for larger crossbars.

9.3.4. ARITHMETIC DESIGN IMPLEMENTATIONS USING SRIF-ADC

Primitive arithmetic functions in any general computing system are addition, subtraction, multiplication, and division. All arithmetic computing hardware units can be essentially broken down into these primitive circuit designs as their building blocks. However, if the algorithms of the latter three functions are observed, their fundamental functional unit is addition. Subtraction (while working with the most widely used binary/radix-2 input representation), say $A-B$, is a special case of addition *i.e.* adding A with 2's complement of B and a carry-in at LSB. Multiplication, say $A \times B$, involves multiple shift-and-add operations on partial products. Division, say $A \div B$, is performed by algorithms such as reciprocation and successive multiplication ($A \times \frac{1}{B}$), restoring/non-restoring division (shift-and-subtract) or division by convergence ($q = \frac{z}{d}$, converging $z \rightarrow q$ such that $d \rightarrow 1$), with addition and multiplication (which in turn is based on addition) as fundamental functional units. Hence, it can be substantiated that addition is the prime arithmetic function for any computing system. Consequently, building memristor-based efficient adder circuit designs would strengthen the promise of memristor device technology to support CIM architectures.

In this section, we present *4-bit* adder, 4 operand *4-bit* adder circuit designs. To show the versatility of SRIF-ADC, we also illustrate the working of a *4-bit* comparator.

ADDER

Addition of two operands is performed by selecting two rows in a crossbar at the same time. Where accessing/activating a single row performs a read operation, accessing two rows adds equivalent mem-conductance of two rows per column. As shown in the Fig. 9.11a, we add two operands A_4 ($a_3a_2a_1a_0$) and B_4 ($b_3b_2b_1b_0$) of size *4-bits* stored as resistance values placed in a 64×4 crossbar array. Here, we perform addition by applying

Design Metrics	SAR [241]	IFC [125]	Op [126]	SRIF *(Proposed)*
Speed (ns)	80	400	200	200
Area (um^2)	5500	103	171	38*
Energy (pJ)	30.2	6.2	12.6	0.5
Resolution (*# levels*)	>64	~ 20	>64	>64
Efficiency (pJ/Op)	12.8	39.1	12.3	0.9
Variation Aware	No	No	No	Yes
ΔV_{RRAM} (V)	~ 1.0	~ 1.0	~ 1.0	~ 0.1

Table 9.2: Comparison of our proposed SRIF design with SAR [241], IFC [125] and Op [126]. One multiply-and-accumulate (MAC) operation corresponds to two arithmetic operations *i.e.* add and multiply. ΔV_{RRAM} is the average voltage across RRAM devices during MAC operation. * is the area per column. The total area is 4× per ADC since it is common to four columns.

a read voltage (V_r) on two rows simultaneously, while the rest of the rows are grounded. Fig. 9.11b shows the result covering all possible input (programmable) scenarios in the crossbar, which gives all possible outputs *i.e.* summation result from $0 \rightarrow 30$.

MULTI-OPERAND ADDER

Addition of four operands *4-bits*, namely A_4 ($a_3a_2a_1a_0$), B_4 ($b_3b_2b_1b_0$), C_4 ($c_3c_2c_1c_0$) and D_4 ($d_3d_2d_1d_0$), in the same crossbar is performed by selecting four rows simultaneously. V_r is now applied to four rows while the rest are grounded as shown in Fig. 9.12a and the result of this addition is shown in Fig. 9.12b. Similar to two operand addition operations, all possible scenarios are covered, which gives all possible outputs *i.e.* summation result from $0 \rightarrow 60$.

COMPARATOR

Circuit design of a *4-bit* comparator is an extension of the *4-bit* adder design. The outcome of a *2-input* comparator is to determine which input is greater of the two or whether both are equal. The working principle of the proposed comparator design is to compare the voltage equivalent of the accumulated current corresponding to individual operands. Fig. 9.13 shows the proposed circuit design.

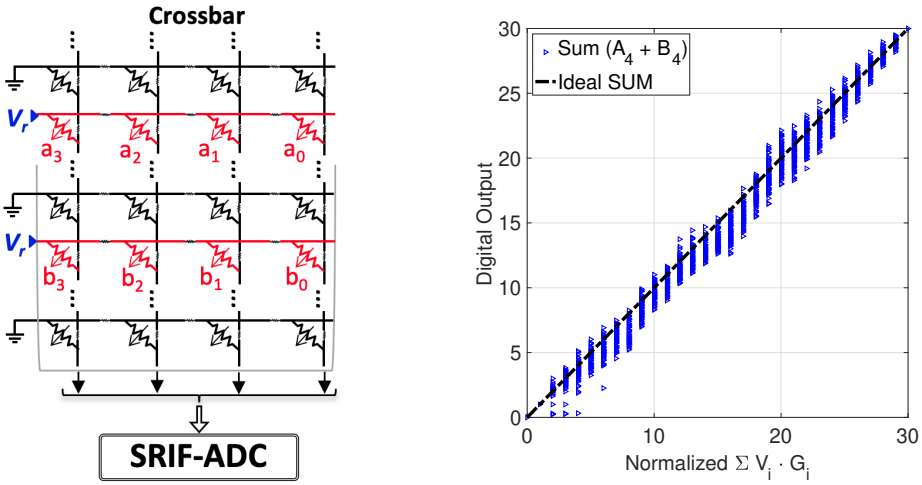
The expected output of the comparator is given by

$$p_1p_0 = \begin{cases} 00, & X_4 = Y_4 \\ 01, & X_4 < Y_4 \\ 10, & X_4 > Y_4 \\ 11, & \text{Invalid} \end{cases}$$

where $x_3x_2x_1x_0(X_4)$ and $y_3y_2y_1y_0(Y_4)$ are two *4-bit* inputs, while *2-bit* output is given by $p_1p_0(P_2)$. Array X_4 and Y_4 are stored in the memristor crossbar, as shown in Fig. 9.13. $X_4(Y_4)$ is evaluated, which results in a voltage equivalent at $evax(evay)$ as $V_{evax}(V_{evay})$. These voltages are fed into two DAs with opposite polarities. The combined output of the two DAs is given by P_2 *i.e.* p_1 and p_0, respectively. Note that since we have only binary outputs, integrate, and fire circuits, counters and registers are not required.

9.3.5. DISCUSSION

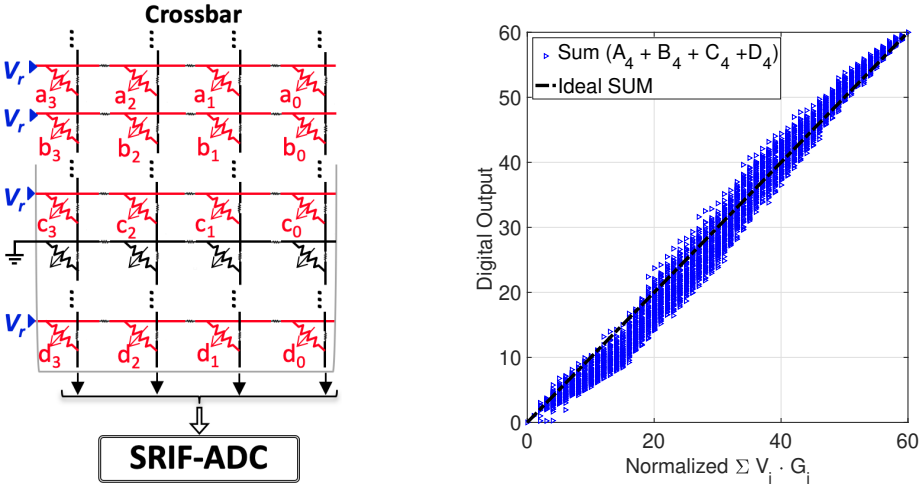
An interesting feature introduced in our proposed ADC is the built-in weighting scheme which allows multiple columns to be evaluated simultaneously and eliminates the post *S+A* operation. However, there are several key aspects to keep in mind as we increase the number of columns that can be evaluated by a single SRIF-ADC in a cycle. According to this technique, we have to increase the size of the current mirror S_{Nsnhi} and/or increase the number of reference voltages V_{refi}, which increases the area overhead. In principle, we should gain in speed as multiple columns are evaluated at the same time, but an exponential increase in the number of spikes has a detrimental effect on speed. Moreover, the increase in spikes degrades the power efficiency and accuracy of the conversion process. In our design, the average number of spikes that SRIF-ADC generates is 450, whereas IFC generates only 128. This does limit our speed and energy efficiency as



(a) (b)

Figure 9.11: (a) Crossbar configuration to perform two operand adder by applying read voltage (V_r) to two arbitrary rows storing operands that require addition. (b) Summation results of two *4-bit* operand addition.

we tend to increase the number of columns combined together or increase the number of rows per column. Therefore, a detailed study is required based on memristor device technology's R_{off}/R_{on}, the number of rows to be supported, bit-levels supported by a



(a) (b)

Figure 9.12: (a) Crossbar configuration to perform four operand adder by applying read voltage (V_r) to four arbitrary rows storing operands that require addition. (b) Summation results of four *4-bit* operand addition.

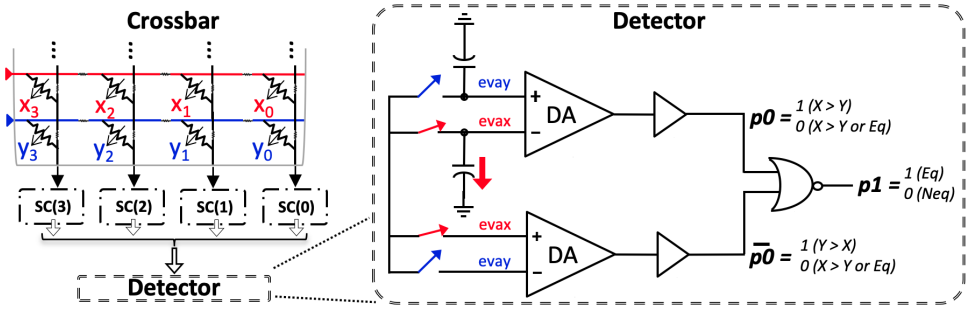


Figure 9.13: Implementation of a *4-bit* comparator using our design methodology.

bit-cell, and other design parameters to determine the number of columns that ensures a healthy improvement in overall efficiency. One possible solution is to have blocks (or groups) of a few columns, and then combine the outputs of these blocks using external *S+A* circuits.

9.4. CONCLUSION

Data-intensive applications such as big data and artificial intelligence have exposed the memory-processor bottleneck issues in CMOS-based von Neumann computing architectures. Emerging analog-based CIM has the potential to overcome these bottlenecks by performing computation within the storage units. However, the efficiency of CIM architectures heavily relies on periphery circuits, and in particular, analog-to-digital converters are identified as the primary limiting block. In this chapter, a novel ADC design methodology is presented that allows key design features and components to provide an efficient solution. The proposed SRIF-ADC is a scalable, variation-aware, compact solution that offers significant improvements in energy consumption and read-disturb margin. We compared our proposed ADC with various state-of-the-art ADC designs while performing MAC operations as a case study. Compared to compact and energy-efficient IFC, we obtained a gain of 2X speed, > 3X range, 2.7X area, and 11.6X energy efficiency. Compared to fast and accurate SAR-ADC, SRIF-ADC is 144X area and 56X energy-efficient with a reduced speed of nearly 0.4X but with similar range and accuracy. Our design also offers pipelining of operations and is a first-of-its-kind that provides RRAM and CMOS variation-aware ADC. Circuit design solutions of two, four operands *4-bit* adders and *4-bit* digital comparator were also presented to showcase the versatility of the proposed circuit designs.

10

COMPACT LOW-POWER CCO-BASED ADC FOR CIM

Data converters involved in analog CIM architectures typically achieve the required computing accuracy at the expense of high area and energy footprint which can potentially determine CIM candidacy for low-power and compact edge-AI devices. In this chapter, we present a memory-periphery co-design to perform accurate A/D conversions of analog matrix-vector-multiplication (MVM) outputs. Here, we introduce a scheme where select-lines and bit-lines in the memory are virtually fixed to improve conversion accuracy and aid a ring-oscillator-based A/D conversion, equipped with component sharing and inter-matching of the reference blocks. In addition, we deploy a self-timed technique to further ensure high robustness addressing global design and cycle-to-cycle variations. Based on measurement results of a 4Kb CIM chip prototype equipped with TSMC 40nm, a relative accuracy of up to 99.71% is achieved with an energy efficiency of 115.1 TOPS/W and computational density of 12.1 TOPS/mm^2 for the MNIST dataset. Thus, an improvement of up to 11.3X and 7.5X compared to the state-of-the-art, respectively.

This chapter is based on [70].

10.1. INTRODUCTION

Analog CIM has the potential to accelerate deep neural network applications by performing in-situ MVM operations on input data (IN) vectors and neuron weights (W) matrices [23, 231] with $\mathcal{O}(1)$ time complexity. CIM produces column current I_{BL} proportional to the aggregated product of IN and W represented by word-line (WL) voltages (V) and bitcell conductance (G) states, respectively [123]. In Fig. 10.1a, we show a crossbar that can be programmed to store the W matrix as G states and CIM-based in-situ MVM operation details. However, the computational accuracy and efficiency greatly depend on the analog periphery, in particular, the analog-to-digital converter (ADC) that converts an I_{BL} current into digital output for data communications among different CIM cores. The key challenges pertaining to CIM-based ADC design are the physical dimension and the energy efficiency while maintaining high conversion accuracy [1].

Recent works on ADC designs can be classified into three classes based on their intermediate physical quantity used for conversion: 1) voltage (V-ADC), 2) current (C-ADC), and 3) time-based ADCs (T-ADC). V-ADCs and T-ADCs are expensive as they typically consist of large components such as large hold capacitors [42, 120, 123, 125, 126, 242] or/and a series of sense amplifiers [42, 120, 123, 126] and large time-digital converters (TDCs) [242, 243], respectively, along with digital-to-analog converters (DACs) [51, 116, 120, 242, 243] for providing reference signals to compute intermediate output calculations. In addition, these ADC classes require several power-hungry comparison cycles for their final digital output conversions. Furthermore, they require an additional conversion from a current (I_{BL}) to a voltage or time domain, thereby, introducing an additional source of inaccuracy. On the other hand, C-ADCs alleviate the need for an additional conversion and typically occupy less area, however, encounter the following

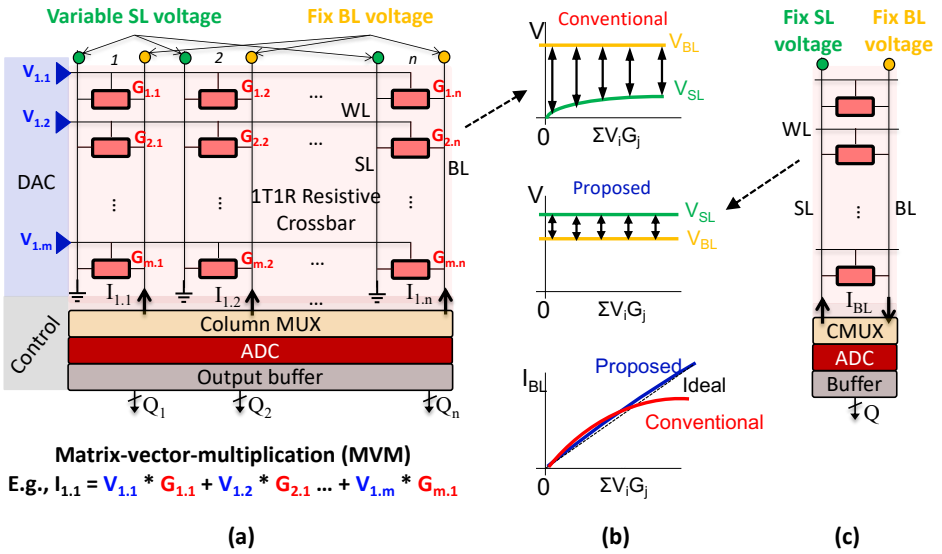


Figure 10.1: (a) Conventional CIM for MVM, (b) impact on transfer characteristics, and (c) overview of the proposed scheme.

serious challenges, as shown in Fig. 10.1b: (a) large I_{BL} range causes a variable differential voltage ($\Delta V=V_{SL}-V_{BL}$) leading to non-linear input-output (INxW - I_{BL}) characteristics and in addition, increases the energy consumption [69, 123, 127], (b) inaccuracies due to non-idealities such as process variations and wire parasitic delay mismatch [69, 244] necessitates the need to keep larger quantization margins, and thereby leading to a reduced dynamic range of the ADC. Therefore, to ensure the required computational accuracy ADCs contribute to major area overhead and energy consumption, severely degrading the overall efficiency of CIM.

To address these challenges, this chapter presents an optimized co-design of the CIM array and current-based ADC to build an ultra-low power and compact CIM-based MVM engine, as summarized in Fig. 10.1b and Fig. 10.1c. The contributions of this chapter are:

- A novel biasing scheme to obtain a linear activation function by virtually fixing the difference of select-line (SL) and bit-line (BL) to a constant voltage with precise inter-matching design techniques for accurate A/D conversion.
- An approach to improve the area/energy efficiency of the A/D conversion by introducing a voltage-controlled ring-oscillator (RO) and an asynchronous ripple carry counter (RCC) arranged in such a way that it captures high-speed RO pulse generations.
- A novel self-timed technique to address the impact of global design variations, cycle-to-cycle mismatch, and CIM array wire delay mismatch on computing accuracy by regulating the duration of the A/D conversion in each cycle.

Measurement results based on our 4Kb CIM chip prototype equipped with TSMC 40nm CMOS technology show that relative accuracy up to 99.71% and 94.74% realizing image classifications can be achieved for the MNIST and E-MNIST datasets, respectively, with an energy-efficiency of 115.1 TOPS/W and computational density of 12.1 TOPS/mm^2.

The rest of the chapter is organized as follows. Section II presents the proposed CIM design, followed by measurement results and chip prototype details in Section III. Finally, Section IV concludes the chapter.

10.2. CIM DESIGN

Next, we present our proposed design and A/D conversion characteristics for CIM-based MVM operations.

10.2.1. ADC DESIGN METHODOLOGY

Fig. 10.2 provides an overview of our ADC methodology and working principle. Our ADC design comprises three stages; (1) Sensing stage (SS): stabilizes nodes SL and BL and converts MVM output I_{BL} to proportional voltages V_{AP} and V_{AN}, (2) Conversion stage (CS): converts analog V_{AP} and V_{AN} values to a discrete number of pulses, and (3) Decision stage (DS): converts the discrete pulses to digital bit-streams. SS is realized using a combination of high-gain differential amplifiers DA_P and DA_N, SL driver PMOS P0, and BL driver NMOS N0. The combinations of DA_P/P0 and DA_N/N0 enable negative feedback to virtually fix SL and BL, respectively, such that $\Delta V=V_{SL}-V_{BL}$=50mV for all

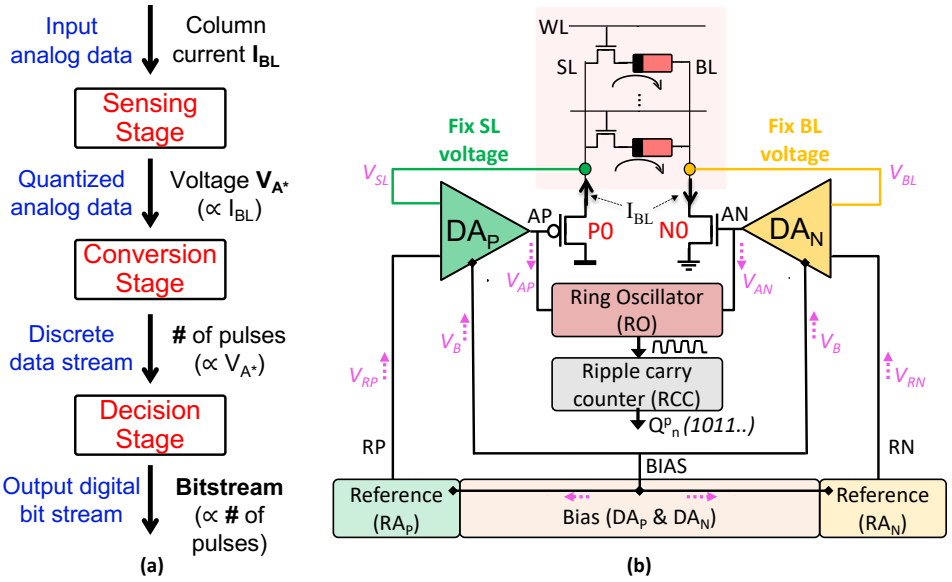


Figure 10.2: (a) Proposed ADC methodology and (b) ADC design concept.

possible INxW combinations. In CS, the amplifier outputs V_{AP} and V_{AN} bias the header and footer, respectively, of the ring-oscillator RO that generates pulses proportional to I_{BL}. DS is implemented using an asynchronous ripple carry counter (RCC) that converts these pulses into bit-streams.

SELF-TIMED ADC

To adapt to global variations and RC degradation, we introduce a dummy row and column that normalizes the duration of the conversion in each cycle. This duration determines the time ADCs are allowed to generate pulses, hence implying a variation-prone duration parameter that requires a normalization step. To achieve this in each operating cycle, we introduce a self-timed mechanism in which a dummy ADC (DmADC) is allowed to capture a pre-determined MVM output of a dummy column programmed with known G states. This column is programmed to have all ON devices while all pass transistors are enabled, independent of IN. Fig. 10.3a shows the implementation of the self-timed scheme and Fig. 10.3b presents the timing diagram to illustrate its working. As the amplifiers establish ΔV=50mV, INxW is performed to generate I_{BL} in the selected columns and in the dummy column during the WL activation time T_{WL}. After an adequate settling time, signal DAEN enables the RO in the DmADC and triggers AEN signals to enable all ROs in the array ADCs to generate pulses. The normalization period T_{EN} i.e., the regulation of the activation time of the RO units in each ADC is achieved by disabling them when DmADC reaches a pre-determined count at its output. For instance, for a 6-bit resolution presented in Fig. 10.3b, DmADC resets DAEN to disable array ADCs at the time instant when DQ<7:0>=64 i.e., DQ<6> toggles to 1.

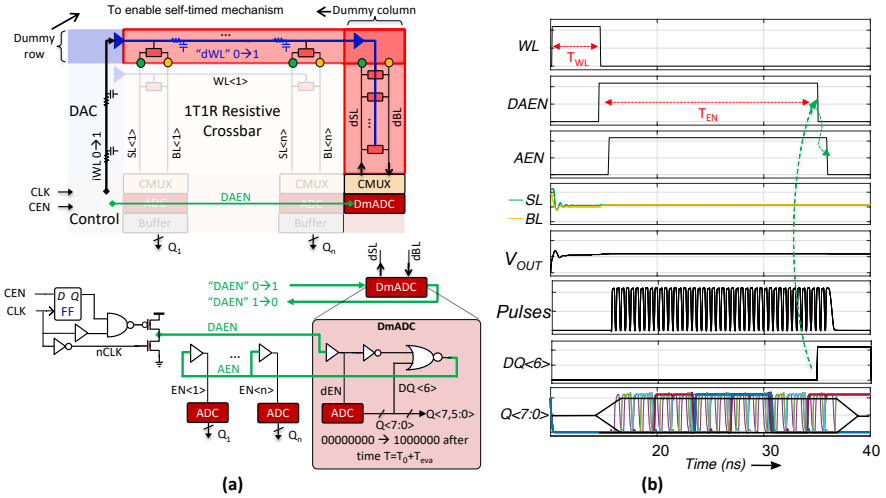


Figure 10.3: (a) Proposed self-timed scheme and (b) timing-diagram of the MVM operation.

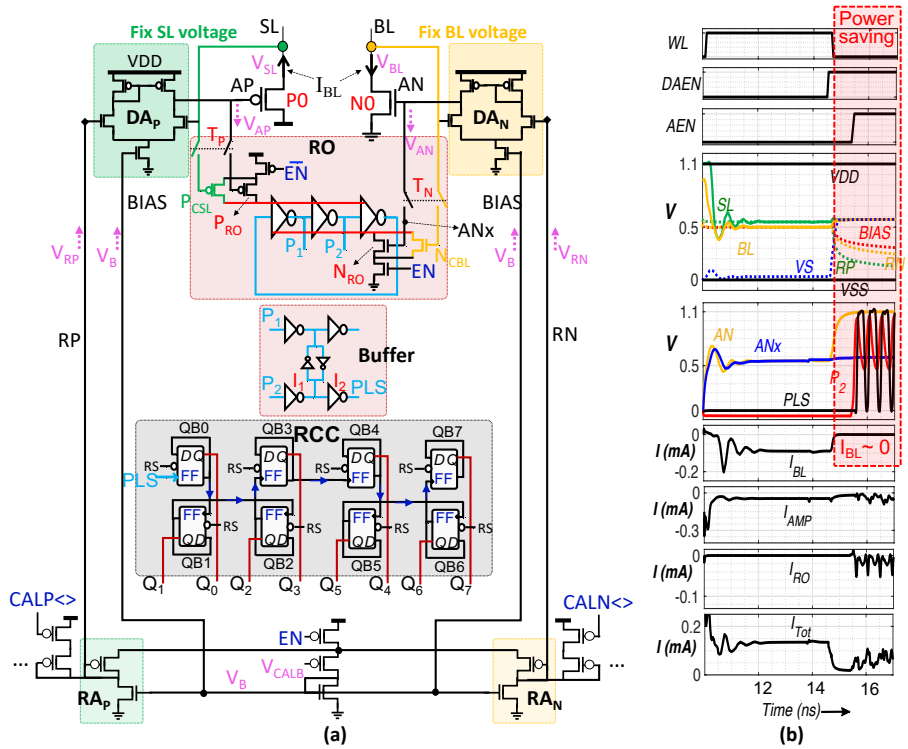


Figure 10.4: (a) Detailed circuit design of the proposed ADC and (b) timing diagram of the critical signals.

ADC DESIGN COMPONENTS

Fig. 10.4a shows the detailed implementation of our ADC design. The aim is to ensure linear input-output characteristics at each of the three ADC stages. In SS, with WL activation, DA_P and DA_N fixes ΔV=50mV to generate $I_{BL} \propto$ INxW. Drivers P0/N0 always operate in a linear mode to produce linear $|V_{GS}|$ (i.e., V_{AP} and V_{AN})$\propto$ INxW. Both DA_P and DA_N are designed using common-centroid matching techniques to accurately match load impedance and tolerate variation. Corresponding reference voltages V_{RP} (for DA_P) and V_{RN} (for DA_N), bias voltage V_B share transistors and have a common bias signal to minimize systematic and random offsets. An inevitable variation in ΔV occurs due to the finite gain of DA_P and DA_N; a higher ΔV value at low currents and vice-versa, incurring a degradation in the linearity of I_{BL} (and V_{AP} and V_{AN}). This is compensated by current sources P_{CSL} and N_{CBL}, which introduce additional current in the RO proportional to the settled V_{SL} and V_{BL}, respectively. Calibration of the bias signal is performed using external bias voltage V_{CALB} and of reference voltages, V_{RP} and V_{RN}, using a series of diode-connected PMOS drivers driven by signals CALP<> and CALN<>, respectively.

Fig. 10.4b shows the timing diagram capturing the behavior of various critical signals to illustrate the working of the ADC. During WL activation, V_{SL} and V_{BL} are settled to produce proportional V_{AP} and V_{AN}, respectively. V_{AP} and V_{AN} are captured at nodes APx and ANx, respectively. Thereafter, enable signal AEN (and DAEN for DmADC) disconnects the AP and AN nodes (also, SL and BL nodes) from the RO through T_P and T_N transmission gates, respectively, and simultaneously activates the RO. The RO produces pulses $P_2 \propto V_{AP}$ and V_{AN} as these voltages bias the header P_{RO} and footer N_{RO}, respectively, of the RO in a current-mirroring configuration while always operating in the linear mode. Post-buffering improves the dynamic range of P_2, thereby generating a PLS signal. To count these high-frequency PLS pulses, adjacent flip-flops FF of RCC are arranged in such a way that it allows minimum path delays. To save power, WL disconnects the array after 5ns, a sufficient period T_{WL} determined by the settling time of SL/BL and AP/AN nodes as required by the amplifiers. This significantly reduces the duration of the flow of I_{BL} in the array, thereby, reducing massive dynamic power at the cost of additional hold caps at APx and ANx. Fig. 10.4b highlights the impact of this power saving *i.e.*, reduction of I_{BL}~0 after WL deactivation. Note that WLs are deactivated after the EN signal disconnects the AP and AN nodes from the RO.

10.2.2. ADC INPUT-OUTPUT CHARACTERISTICS

Fig. 10.5a shows the input-output characteristics for the accumulation of 64 rows with 1-bit IN and 1-bit W elements. Virtually fixing ΔV=50mV allows $I_{BL} \propto$ INxW (or $\Sigma V_i \cdot G_j$) and V_{AP} and V_{AN} follow I_{BL} to generate proportional digital outputs. The compensation scheme described previously linearizes the number of pulses generated by the RO, albeit with a slight variation in ΔV, as depicted in the final digital output. The total energy consumption increases with $\Sigma V_i \cdot G_j$ corresponding to the increased currents in the array (I_{BL}), RO, and counter, while DA_P and DA_N consume nearly constant energy. Fig. 10.5b presents 3σ variation analysis of the MVM output. The spread (σ) of the settled ΔV is 6.2mV and the combined offset is 0.2mV.

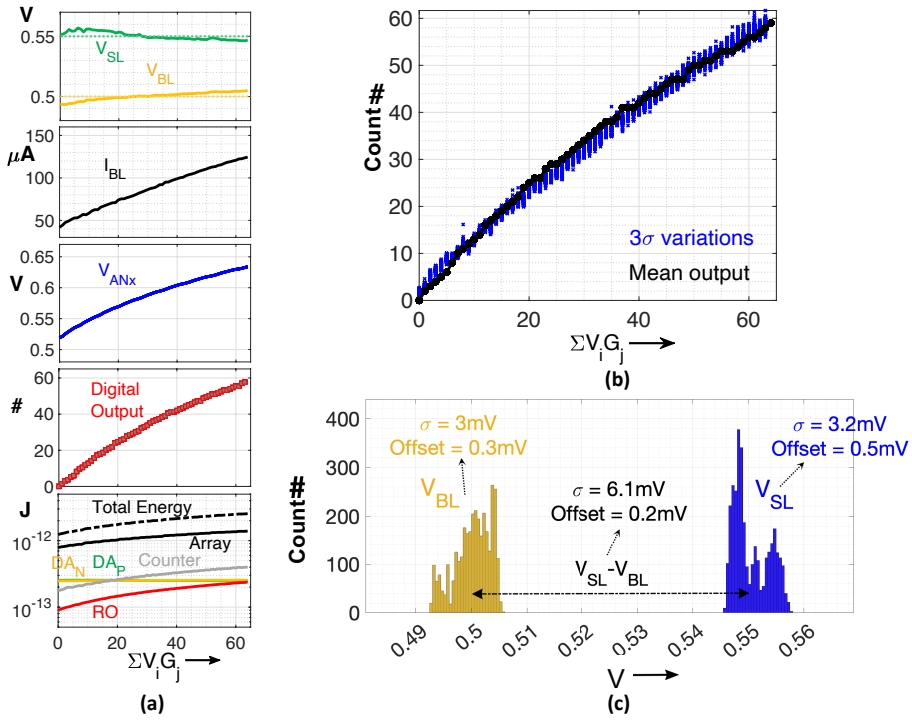


Figure 10.5: (a) Input-output characteristics of the ADC, (b) variation analysis of MVM, and (c) of settled voltages V_{SL} and V_{BL}.

10.3. CIM IMPLEMENTATION AND CHARACTERISATION

10.3.1. CHIP PROTOTYPE AND EXPERIMENTAL SETUP

Fig. 10.6a presents the microscopic view of the CIM implementation equipped with TSMC 40nm CMOS technology and Fig. 10.6b shows the experimental setup. The chip prototype comprises a 64x64 crossbar array built using a 1-transistor-1-resistor (1T1R) bitcell configuration. Here, to conduct the characterization of our ADC, the conductance of the bitcell (in 1R) is pre-programmed using fixed NMOS-based resistors to mimic the resistive properties of a 1-bit storage device i.e., a low (LCS) and a high conductance state (HCS) corresponding to logic 0 and 1, respectively. The programming sequence is such that all possible 64 combinations of W vector i.e., from the minimum to the maximum number of ON devices in a column are pre-programmed sequentially to the 64 columns in the crossbar array. This allows a complete range of conductance values possible for the MVM operation.

10.3.2. ADC CHARACTERISATION RESULTS

The chip prototype is characterized to determine the functional voltage boundaries and measure energy consumption and latency per conversion cycle of the ADC. Fig. 10.7a presents the shmoo plot along with the conversion speed for a 6-bit resolution. As ex-

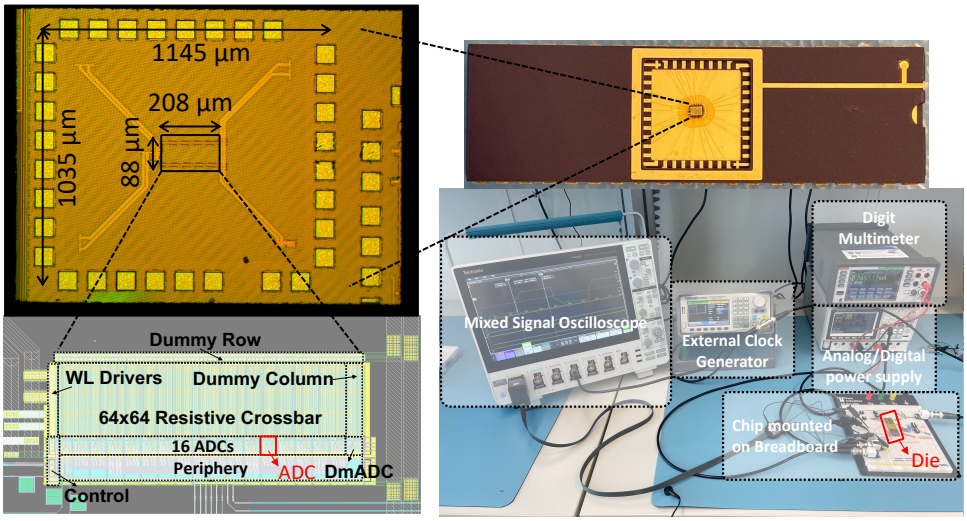


Figure 10.6: (a) Microscopic view of the fabricated chip with CIM layout and (b) experimental setup with the prototype die.

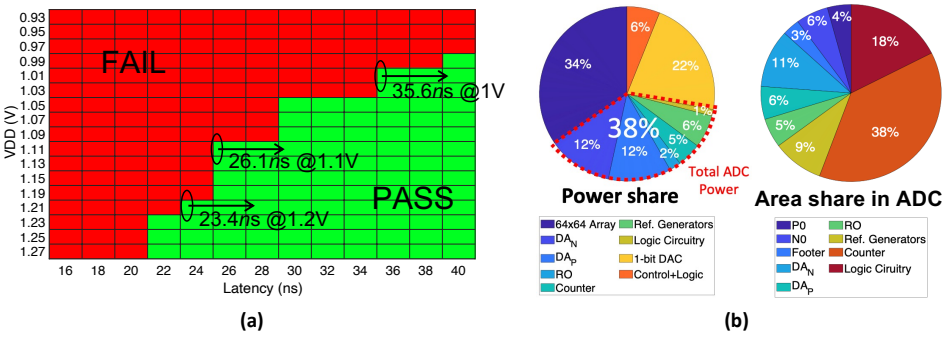


Figure 10.7: (a) Shmoo plot and (b) component-wise energy and area.

pected, the conversion speed increases with the voltage of operation with a conversion time of 26.1 *n*s at 1.1V. Fig. 10.7b presents the component-wise energy breakdown of the MVM operation per conversion cycle, where ADC consumes an average energy of 3.3 *p*J which is roughly 38% of the total energy. In addition, the figure shows component-wise area utilization in our ADC.

10

10.4. SYSTEM-LEVEL RESULTS

10.4.1. SYSTEM-LEVEL VALIDATION

We evaluate the benefits of our CIM design on image classification applications, using CNN-based Lenet-5 for MNIST and E-MNIST datasets.

Fig. 10.8a presents the validation of our design for different ADC bit-resolutions. The

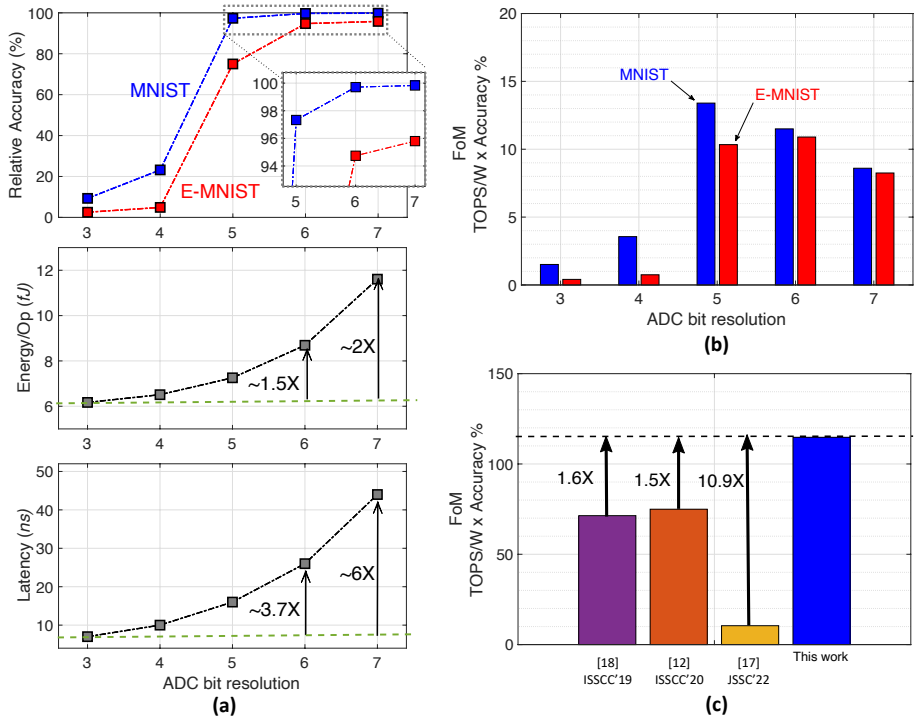


Figure 10.8: (a) Accuracy and efficiency, and (b) FoM with different ADC resolutions. (c) FoM comparison with MNIST dataset.

resolution is adjusted by varying the maximum number of allowed pulse generation that corresponds to the dummy column. For instance, in a 4-bit ADC, toggling DQ<4> to 1 disables DAEN and subsequently, AEN signals. It can be seen that a low-resolution ADC allows better energy efficiency at the expense of accuracy and latency. We introduce a figure of merit (FoM) which describes the energy efficiency of a system for accurate computations and shows the comparative merits of different ADC bit-resolutions. Fig. 10.8b shows that a mid-range resolution offers a good trade-off between accuracy and energy efficiency for image classification applications. Based on simulation results equipped with measured latency and energy consumption during an MVM operation, this work can provide as high as 10.9X FoM improvement compared to [62] when evaluated for the MNIST dataset, as shown in Fig. 10.8c.

10

10.4.2. COMPARISON RESULTS

A detailed comparison is presented in Table 10.1. We show that for our 6-bit resolution ADC, an improvement of 11.3X and 7.5X can be achieved in terms of TOPS/W and TOPS/mm^2, respectively, compared to [62] with comparable accuracy on the MNIST database. The conversion time is longer as compared to [33], [123] and [60]. However, the energy efficiency is improved by 1.6X, 2.4X, and 2X, respectively, owing to the re-

Parameters	[33] ISSCC-'19	[51] ISSCC-'20	[123] ISSCC-'21	[62] JSSC-'22	[60] ISSCC-'22	**This work (6b ADC)**
Technology	55nm	130nm	22nm	14nm	40nm	40nm
. Voltage	1V	4.2V	0.8V	0.8V	0.9V	1.1V
Storage device	SRAM	RRAM	RRAM	PCM	PCM	Resistive
. Storage	2b	analog	1b	analog	analog	1b
Bitcell	Twin 8T	2T2R	1T1R	8T4R	1T1R	1T1R
Capacity	4Kb	158Kb	4Mb	65.6Kb	2Mb	4Kb
Accumulation	9	-	1024	256	256	64
. DOUT	3b	8b	10b	8b	11b	6b
ADC resolution	1b (SA)	1b-8b	1b (VSA)	8b	4b	3b-7b
Latency	3.2	51.1	10.3	130	8.6	26.1
TOPS/W	72.1	78.4	47.3	10.5	57.6	**115.1**
TOPS/mm^2	-	-	-	1.6	-	**12.1**
Accuracy						
. MNIST	99.02%	95.6%	-	99.7%	-	99.7%
E-MNIST	-	-	-	-	-	94.7%

Table 10.1: Comparison table. - implies data is not reported.

duced number of components used in our proposed memory array-periphery co-design scheme.

10.5. CONCLUSION

This work presents a novel memory-periphery co-design to perform accurate A/D conversions of analog MVM outputs with high energy efficiency and computational density. The chapter introduces a scheme where array access lines can be virtually fixed to improve the accuracy of the MVM operation and pave the path for a compact and ultra-low power ADC design. In addition, the measurement results of the chip prototype validate the ADC design and derive its input-output characteristics. A relative accuracy of up to 99.71% is achieved with an energy efficiency of 115.1 TOPS/W and computational density of 12.1 TOPS/mm^2 for the MNIST dataset, thus, making it a suitable implementation for executing MVM operations in low-power edge AI devices.

11

VOLTAGE-TO-TIME BASED ADC FOR 2T2R RRAM

This chapter presents a memory-periphery co-design to perform accurate A/D conversions of analog MVM outputs. Here, the ADC design is greatly simplified by making use of complementary data storage and input activation, whereby the crossbar generates differential analog MVM outputs which are digitized using a compact dual-ramp voltage-to-time (VTC)-based ADC involving a start-stop ring-oscillator (RO). Moreover, low-power techniques are introduced in deploying voltage crossing detectors (VCD). These include regulating power consumption during its operation and disabling it the instant comparator decision is made. Based on measurement results of a 256x8 CIM chip prototype equipped with TSMC 40nm, a relative accuracy of up to 99.4% is achieved with an energy efficiency of 206 TOPS/W for the MNIST dataset.

This chapter is based on [71].

11.1. INTRODUCTION

Analog CIM has the potential to accelerate deep neural network applications by performing in-situ MVM operations on input data (IN) vectors and neuron weights (W) matrices [23, 231] with $\mathcal{O}(1)$ time complexity. Data converters involved in analog CIM architectures typically achieve the required computing accuracy at the expense of high area and energy footprint which can potentially determine CIM candidacy for low-power and compact edge-AI devices. CIM produces column current I_{BL} proportional to the aggregated product of IN and W represented by word-line (WL) voltages (V) and bitcell conductance (G) states, respectively [123]. In a typical CIM architecture, a memory unit arranged in a crossbar structure can be programmed to store the W matrix as G states, and CIM-based in-situ MVM operation is performed by the interaction of V and G while exploiting Kirchhoff's and Ohm's laws. However, the computational accuracy and efficiency greatly depend on the analog periphery, in particular, the analog-to-digital converter (ADC) that converts an I_{BL} current into digital output for data communications among different CIM cores. The key challenges pertaining to CIM-based ADC design are the physical dimension and the energy efficiency while maintaining high conversion accuracy [1].

Non-linearity due to wire parasitics and process variations occurs because of the changing settling voltages (biasing conditions) on the access lines with different INxW conditions, which leads to non-linear input-output characteristics (ideal MVM - I_{BL}), as

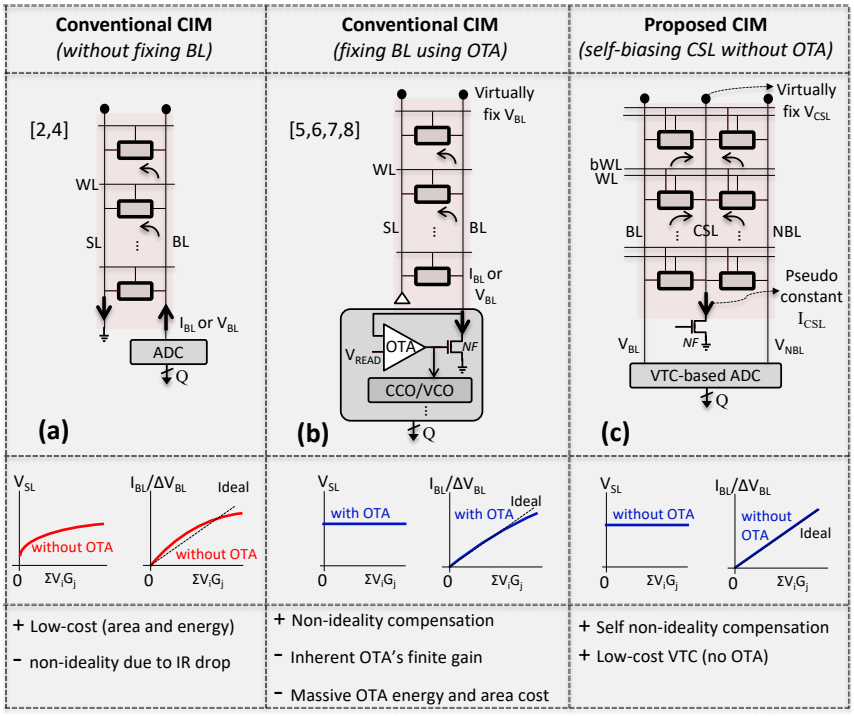


Figure 11.1: Overview of conventional and proposed CIM accelerators.

shown in Fig. 11.1a [60, 123]. State-of-the-art CIM-based ADCs primarily ensure ideal biasing conditions on the access lines i.e., bit lines (BL) by virtually fixing them using an operational-transconductance amplifier (OTA) via strong negative feedback loop, as highlighted in Fig. 11.1b [51, 62, 70, 245]. Unfortunately, with the scaling of CIM array size leading to large BL wire parasitics, these large-sized OTAs are typically biased with large biasing currents to reduce the BL settling time. This leads to significant constant static power during the entire MVM operation and overall area overhead, resulting in poor computational efficiency. In addition, OTA suffers from inherent finite gain leading to BL voltage fluctuations for different IN×W conditions, resulting in inaccuracy. Moreover, ADC schemes utilizing controlled oscillators, residing next to the OTA stage, typically suffer from inherent non-linearity (input voltage or current v/s oscillation frequency) and require expensive compensation schemes to address this [62, 70, 246]. On the other hand, comparator-based ADCs such as SAR/Flash ADCs typically include large CDACs or a large number of comparators and reference ladders, and therefore, are not amenable to the stringent area constraints of the CIM periphery [51]. Alternatively, a recently proposed IN vector breakdown and massive rearrangement approach enabling an equal number of row activations ensures a pseudo-constant biasing condition, however, at the cost of significant IN pre-processing and still utilizes an OTA [245]. In short, near-constant biasing conditions to achieve the required analog MVM accuracy come at the cost of significant efficiency limitations.

In this chapter, we propose a memory-periphery co-design approach equipped with voltage-to-time (VTC) ADC that utilizes true and complementary binary forms of both INs and Ws to perform MVM operations to inherently compensate for the non-linearity issues related to IR drop and circuit non-idealities, as described in Fig. 11.1c. The key contributions are:

- Introduces a differential MVM read scheme where true and complementary INs are applied to true and complementary stored Ws such that the array biasing condition remains the same, independent of INs and Ws values. Thus, alleviating the need for expensive OTAs.

- Develops a voltage-sensing scheme where BL caps sample the differential MVM voltage outputs via charge-sharing that are digitized using a compact dual ramp-based VTC.

- Presents design and implementation of ring-oscillator (RO) and voltage crossing detector (VCD) with novel low-power techniques to realize an efficient VTC design.

Measurement results based on our chip prototype equipped with TSMC 40nm CMOS technology show that relative accuracy up to 99.4% realizing image classifications can be achieved for the MNIST dataset. The energy efficiency of 206 TOPS/W, and MVM latency of 15 ns is achieved with our signed 8-bit ADC realization.

The rest of the chapter is organized as follows. Section II presents the proposed CIM design, followed by simulation results and chip prototype details in Section III. Finally, Section IV concludes the chapter.

11.2. PROPOSED CIM DESIGN

Next, we present details of our proposed CIM design.

11.2.1. VOLTAGE-SENSING BASED COMPUTING

Voltage-based sensing typically involves CIM generating an output voltage on the access lines *ideally* proportional to the MVM result. This refers to the access lines starting from a discharged or pre-charged value and during the MVM operation of fixed duration, the voltage on this access line changes proportional to the effective MVM value. We utilize this voltage-based sensing approach to perform analog MVM on the proposed resistive CIM architecture.

Fig. 11.2a describes our proposed scheme where we provide true and complementary INs to true and complementary Ws that ensure the total current flowing through the CSL remains constant, independent of the operand values. This allows the generation of fully differential linear voltage output (ΔV) on the access lines that facilitates low-power and compact A/D conversion for signed binary MVM operations.

11.2.2. BITCELL CONFIGURATION AND INPUT ACTIVATION

Fig. 11.2b presents the bitcell configuration storing signed 1-bit data and the signed 1-bit IN activation to the crossbar array. The data is stored in two 2T2R bitcells, namely data bitcell DCELL and data-bar bitcell DBCELL. In DCELL, the left and right resistive devices store true and complementary data values, respectively. In contrast in DBCELL, the left and right resistive devices store complementary and true data values, respectively. For instance, W=+1 implies DCELL holds high conducting state HCS and low conducting state LCS on left and right resistive devices, respectively, denoted by '10' and vice-versa for DBCELL, denoted by '01'. In the case of W=-1, DCELL stores '01' and DBCELL stores '10'. These bitcell configurations are described in Table 11.1.

Similarly, IN=+1 implies INP=1 and INN=0, and vice-versa for IN=-1. INP and INN are applied as shown in Fig. 11.2b *i.e.,* left resistive elements receive INP and right resistive elements receive INN to perform the typical four-quadrant MVM operation (four combinations of positive and negative multiplications). To ensure constant total current flowing into the CSL of the crossbar, we ensure a constant current contribution from each element-wise IN×W. This constant element-wise current generation is the accumulation of four currents; i) selected LCS device contributing I_{LCS}, ii) selected HCS device contributing I_{HCS}, iii) unselected LCS and HCS devices contributing to leakage currents (I_{leak}) i.e., all four combinations corresponding to binary data bits. This is described in Table 11.1 which highlights that all possible IN and W conditions result in constant I_{CSL}. Therefore, this leads to a linear relation between the ideal MVM output and the developed voltage at the bitlines. Fig. 11.2c illustrates the step-wise computation for the signed INxW using our scheme. The analog voltage outputs of the twin bitcells generate all four combinations of positive and negative INs and Ws; positive IN and positive W as ΔV_{PP}, positive IN and negative W as ΔV_{PN}, negative IN and positive W as ΔV_{NP} and negative IN and negative W as ΔV_{NN}. The positive analog MVM output is obtained by combining ΔV_{PP} and ΔV_{NN}. This is done via charge sharing of BL_{PP} (storing ΔV_{PP}) and BL_{NN} (storing ΔV_{NN}) bitlines by activating the enable signal EN_CS, finally gener-

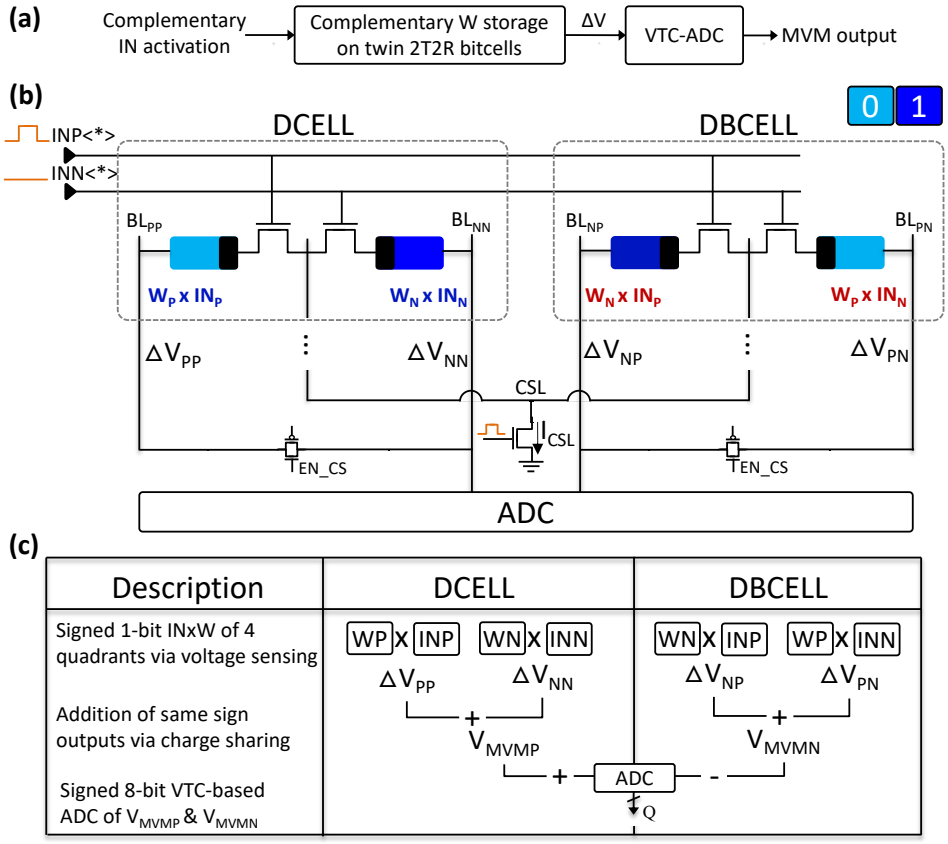


Figure 11.2: (a) Overview of the proposed CIM. (b) and (c) illustrates the IN and W mapping on twin-2T2R bitcells and detailed descriptions, respectively.

ating V_{MVMP}. Similarly, the negative analog MVM output is obtained by charge sharing of BL_{NP} (storing ΔV_{NP}) and BL_{PN} (storing ΔV_{PN}) bitlines, finally generating V_{MVMN}. We convert the difference of V_{MVMN} and V_{MVMN} into a digital domain using VTC-based ADC.

11.2.3. DUAL-RAMP BASED VOLTAGE-TO-TIME CONVERTER (VTC)

Fig. 11.3a describes our ramp-based scheme where we first translate the input voltage into a time domain and then convert this time duration into a proportional number of discrete pulses. In our proposed CIM, two linear current sources of equal strength (RAMPGENs) are used to ramp up V_{MVMP} and V_{MVMN} after the sign determination, and the time difference T_{MVM} it takes for these ramps to cross a reference voltage V_{REF} (detected by VCDP and VCDN, respectively) is $\propto \Delta V_{MVM}$=V_{MVMP}-V_{MVMN}. T_{MVM} is quantized using an RO running at a fixed frequency f_{RO}, whereby, the number of pulses generated within T_{MVM} determined by a ripple carry counter (COUNT) generates the

INxW		Bitcell configuration		IN activation		Intermediate MVMs				Result	I_{CSL} (per bitcell)
IN	W	DCELL	DBCELL	INP	INN	PP	NN	NP	PN		
+1	+1	10	01	1	0	1	0	0	0	+1	I_{LCS} + I_{HCS} + 2xI_{leak}
-1		10	01	0	1	0	0	1	0	-1	
+1	-1	01	10	1	0	0	0	0	1	-1	
-1		01	10	0	1	0	1	0	0	+1	

Table 11.1: Constant I_{CSL} contribution per bitcell with different INs & Ws.

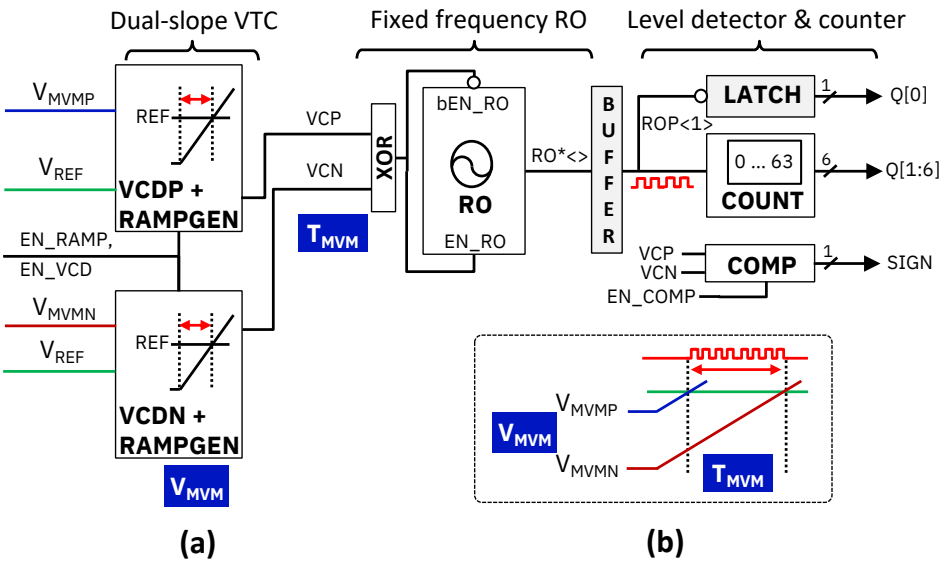


Figure 11.3: (a) Proposed VTC-based ADC and (b) its working principle.

digital output. The sign bit is obtained by comparing V_{MVMP} and V_{MVMN} using a SIGN comparator COMP (enable signal EN_COMP) before the ramp phase and an additional LSB bit is obtained using LATCH by determining the voltage level (phase) of the RO output signal. Fig. 11.3b illustrates the expected VTC-based A/D conversion.

Integral non-linearity (INL) of the A/D conversion is improved by tuning the ramp speed corresponding to f_{RO}, where f_{RO} tends to fluctuate due to variations. Therefore, the required calibration scheme includes full-scale integer counts generated by the RO aligned with the two differential voltage inputs having maximum voltage difference for each CIM column. Differential non-linearity (DNL) is inherently taken care of as the RO oscillates with a fixed f_{RO}, whereas, typically VCO or CCO-based ADCs suffer from inherent non-linearity and therefore, need dedicated compensation schemes.

Fig. 11.5a presents the details of the RO circuit design. The RO consists of five delay elements and the internal nodes are initialized in an alternate logic 1 and 0 fashion. Each delay element comprises a cascoded inverter and the delay chain which when enabled oscillates at f_{RO}=1/10×t_d (t_d delay of one delay element. We utilize a start-stop scheme, where the RO is enabled as one of the crossings (for either V_{MVMP} or V_{MVMN}) has occurred and disabled as the other crossing has occurred. The start-stop signal EN_RO is

generated using a simple XOR logic on two corresponding VCP and VCN voltage crossing signals, respectively. The number of completed oscillations translates to the MSBs of the digital output Q<1:6>. A level 1 (of ROP<1>) latched at the end implies that the RO is in the first half of its oscillation and a level 0 latched implies that the RO is in its second half, which translates to the LSB of the digital output Q<0>. The timing diagram of various signals is described in Fig. 11.4, illustrating the sampling of four intermediate quadrant MVM outputs onto respective BL, charge-sharing for same-sign result addition, determination of sign bit via SIGN comparator and conversion of V_{MVM} into T_{MVM} via VTC-based ADC.

11.2.4. RAMP GENERATOR AND ADAPTIVE LOW-POWER VCD

Fig. 11.5b shows a highly cascoded RAMPGEN (eight PMOS transistors in series, enabled by EN_RAMP signal) that ensures a linear ramp including a mirror configuration for biasing the ramp circuits. To reduce the power consumption of the two VCDs, two mitigation techniques are introduced; i) adaptive biasing current I_{VCD} of the VCDs, and ii) disabling VCDs as soon as they resolve the outcome. A two-stage amplification is introduced, where the inner regenerative cross-coupled inverters are minimum-sized transistors. Regarding the regularization of I_{VCD}, BL voltage biases its tail transistor N_{TAIL}. During the A/D conversion phase, BL is charged from a low voltage value towards VDD, and only when V_{BL*} reaches close to V_{REF} is when VCDs are tuned to have maximum amplification gain. Therefore, the instantaneous power consumed by the VCD quadrat-

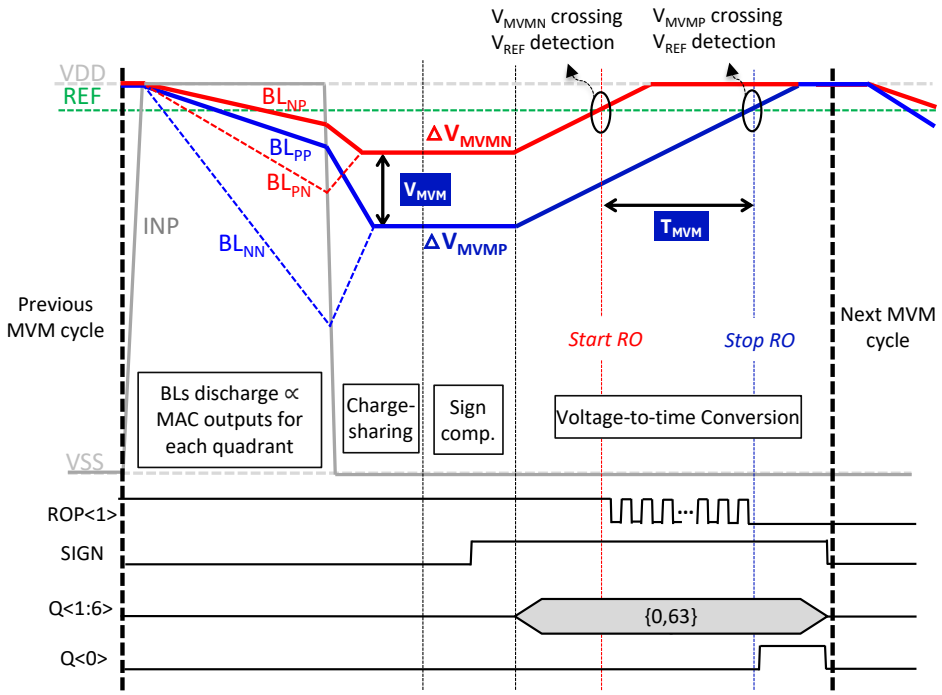


Figure 11.4: Timing diagram of the proposed analog MVM computation.

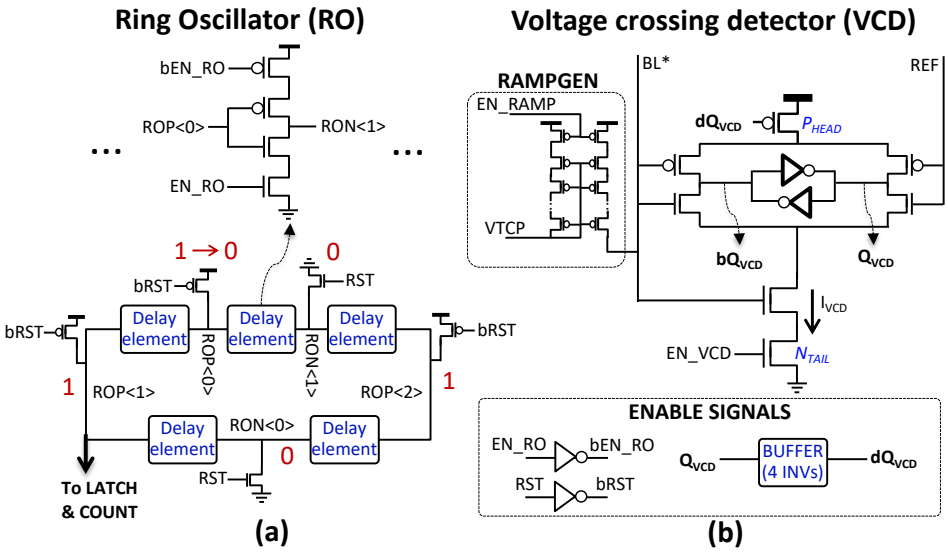


Figure 11.5: Circuit details of (a) ring oscillator and (b) voltage crossing detector.

ically increases and only reaches the required I_{VCD} near the voltage crossing.

Regarding the disabling of the VCDs, a header P_{HEAD} is added to the cross-coupled inverting amplifier-based circuit. P_{HEAD} disconnects the power supply as the VCD makes a decision *i.e.,* when Q_{VCD} toggles to 1 following the detection of V_{BL} crossing V_{REF}. Note-worthily, the propagation delay caused by the VCD is constant for any given MVM input conditions as the voltage of crossing is always the same i.e., V_{REF}. Thus, our scheme does not contribute to any input voltage-dependent time towards T_{MVM}, thereby, alleviating the need for any expensive compensation schemes required for VCO/CCO based schemes [62, 70].

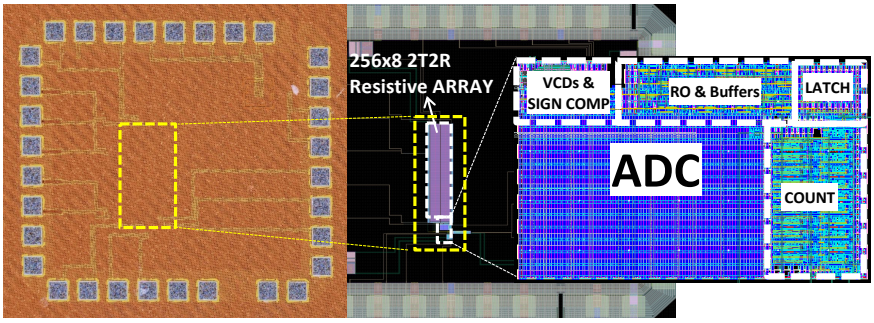


Figure 11.6: Microscopic view of the fabricated chip with CIM layout.

11.3. RESULTS

11.3.1. CHIP PROTOTYPE AND SETUP

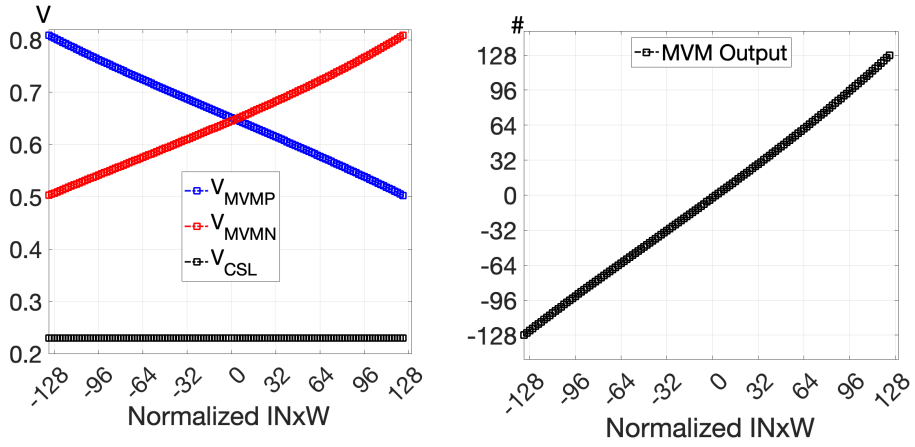
Fig. 11.6 presents the microscopic view of CIM equipped with TSMC 40nm CMOS technology. The chip prototype comprises a 256x8 crossbar array built using a twin-2T2R bitcell configuration. Here, to conduct the characterization of our ADC, the conductances of the resistive elements in the bitcell are pre-programmed using fixed N-poly resistors to mimic the resistive properties of a 1-bit storage device in the true and complementary fashion, where LCS and HCS correspond to logic 0 and 1, respectively. The IN activation sequence is such that all possible combinations of IN×W are tested. The setup details are presented in Table 11.2.

11.3.2. CIRCUIT-LEVEL RESULTS

Fig. 11.7a and Fig. 11.7b present analog and digital input-output characteristics for the accumulation of 256 rows with signed binary IN and W elements, respectively. Virtually fixing CSL voltage through the current equalizing technique allows $\Delta V_{MVM} \propto$ INxW and ΔT_{MVM} follows ΔV_{MVM} to generate proportional digital outputs. The calibration schemes described previously ensure full-scale A/D conversion i.e., the maximum digital output at the two cases of maximum $|\Delta V_{MVM}|$. The total energy consumption increases with IN×W corresponding to the increased oscillation cycles of the RO (and the associated toggling in the counter). The two VCDs combined, the RAMPGEN circuits and SIGN comparator consume nearly constant energy for any given input value. ADC consumes 2.5 *pJ* on average which is roughly 30% of the total energy.

11.3.3. SYSTEM-LEVEL RESULTS AND COMPARISON

A detailed comparison is presented in Table 11.2. Based on measurement results of MVM characterisation involving CIM-based column and our signed 8-bit resolution ADC, an



(a) **(b)**

Figure 11.7: (a) Simulated and (b) measured I/O characteristics of the analog and digital MVM output.

Parameters	[245] ISSCC-'23	[51] ISSCC-'20	[123] ISSCC-'21	[62] JSSC-'22	[60] ISSCC-'22	**This work**
Technology	22nm	130nm	22nm	14nm	40nm	40nm
. Voltage	0.7-0.8V	4.2V	0.8V	0.8V	0.9V	1.1V
Storage device	RRAM	RRAM	RRAM	PCM	PCM	Resistive
. Storage	1b/2b	analog	1b	analog	analog	1b
Bitcell	1T1R	2T2R	1T1R	8T4R	1T1R	Twin 2T2R
Accumulation	128	-	1024	256	256	256
. DOUT	8b	8b	10b	8b	11b	8b
ADC resolution	5b (VSA+CSA)	1b-8b	1b (VSA)	8b	4b	8b
Latency	-	51.1	10.3	130	8.6	15.0
TOPS/W	51.4	78.4	47.3	10.5	57.6	**206**
Accuracy						
. MNIST	99.0%	95.6%	-	99.7%	-	99.4%
CIFAR-10	90.4%	-	-	96.7%	91.9%	90.8%

Table 11.2: Comparison table. '-' implies data is not reported.

improvement of up to 19.6X can be achieved in terms of TOPS/W compared to [62] with comparable simulated accuracy on the MNIST and CIFAR-10 databases for image classification applications. The conversion time is longer as compared to [245], [123] and [60]. However, the energy efficiency is improved by 4X, 4.3X, and 3.5X, respectively, owing to the reduced number of components used in our scheme.

11.4. CONCLUSION

This work presents a novel memory-periphery co-design to perform accurate A/D conversions of analog MVM outputs with high energy efficiency. We introduce a scheme where the ADC design is greatly simplified by making use of complementary data storage and complementary input activation, whereby the crossbar generates differential analog MAC voltage outputs. These are converted into a digital domain using a compact dual-ramp VTC-based ADC with novel low-power techniques. A relative accuracy of up to 90.8% is achieved with an energy efficiency of 206 TOPS/W for the CIFAR-10 dataset, thus, making it a suitable implementation for executing MVM operations in low-power edge AI devices.

12

Outlook and Discussion

This chapter puts into perspective the proposed ADC designs that accelerate CIM-based arithmetic units. It covers how the different challenges addressed, design methodology and optimization techniques have progressively improved the performance metrics as well as scalability in terms of array size and bit resolution of the ADC over the course of the thesis. It also discusses possible design considerations required to incorporate various design features such as utilizing other memristor technologies or conventional memories, different bitcell configurations and data types, CMOS technology scaling, and scalability.

This chapter is based on [1, 69–71].

12.1. INTRODUCTION

This chapter, similar to Chapter 8 that focuses on the proposed logic accelerators, presents a brief outlook on the proposed ADCs for CIM-based arithmetic accelerators and how the progress can be interpreted during the course of the thesis work. Section 12.2 describes the critical challenges targeted by each ADC design that can be seen to evolve with each work, where the evolution can be defined by various factors; 1) minimum sensing margins involved due to increasing bit resolution, 2) maximum array size, etc. It also presents a trend where this progress can also be seen in terms of the performance metrics achieved by each work. Thereafter, it provides a comparison with the prior-art solutions as well as with the preceding work. Section 12.3 presents a brief account of the discussion points on the feasibility of utilizing the proposed solutions and identifies several design explorations. Finally, Section 12.4 concludes the chapter.

12.2. OUTLOOK

This section highlights the key challenges addressed by each work and their key concepts while focusing on how each design has improved the state-of-the-art and the preceding works. The challenges addressed are summarized in Fig. 12.1 and the relative efficiency metrics are presented in Fig. 12.2.

Chapter 9 [69] describes the first work on this thesis that targets improving the efficiency of ADC designs for CIM-based arithmetic accelerators. Prior to this work, most of the ADC-related efforts in literature were investigating current-based approaches because they could inherently provide compact, low-power solutions. As described in Chapter 9, SAR-ADCs are arguably quite expensive in terms of area utilization and power consumption. However, the state-of-the-art current-based solutions at the time [47, 125, 226, 233, 234] suffered from limited scalability in terms of operand size, digital post-processing of ADC outputs, area utilization, and high power consumption while also being susceptible to read disturbance. The proposed SRIF-ADC design introduces i) *built-in weighing* technique that aggregates weighted MAC results in the analog domain, thus alleviating the need for digital post-processing ii) discharge-SET-discharge approach when utilizing an integrate-and-fire (IFC) technique which improves linearity compared to prior approach of charge-RESET-charge, iii) *variation-aware self-timing path (STP)* to offer robustness against global variations, and iv) virtually fixing the BL voltage using a high-gain differential amplifier (HgDA) and combining this technique with other blocks to provide different features such as built-in sample-and-hold, built-in weighing, variation-adaptability and ensuring a small voltage across memristor devices that drastically reduces the probability of read disturbance. In this work, the scalability of this approach is also investigated for other functional and arithmetic operations such as addition, multiplication, and comparison using multi-bit operands. SRIF-ADC performs the A/D conversion of MAC operation of operands size of 64 and is shared among four columns storing 4-bit data. Equipped with TSMC 90nm CMOS technology, the ADC consumes roughly ~2pJ and 152μm^2 with a latency of 200ns.

Chapter 10 [70] describes the work that further improves the linearity of the oscillator-based ADC characteristics while improving the performance and energy efficiency of CIM-based MVM operations. When compared with the prior art and SRIF-ADC design

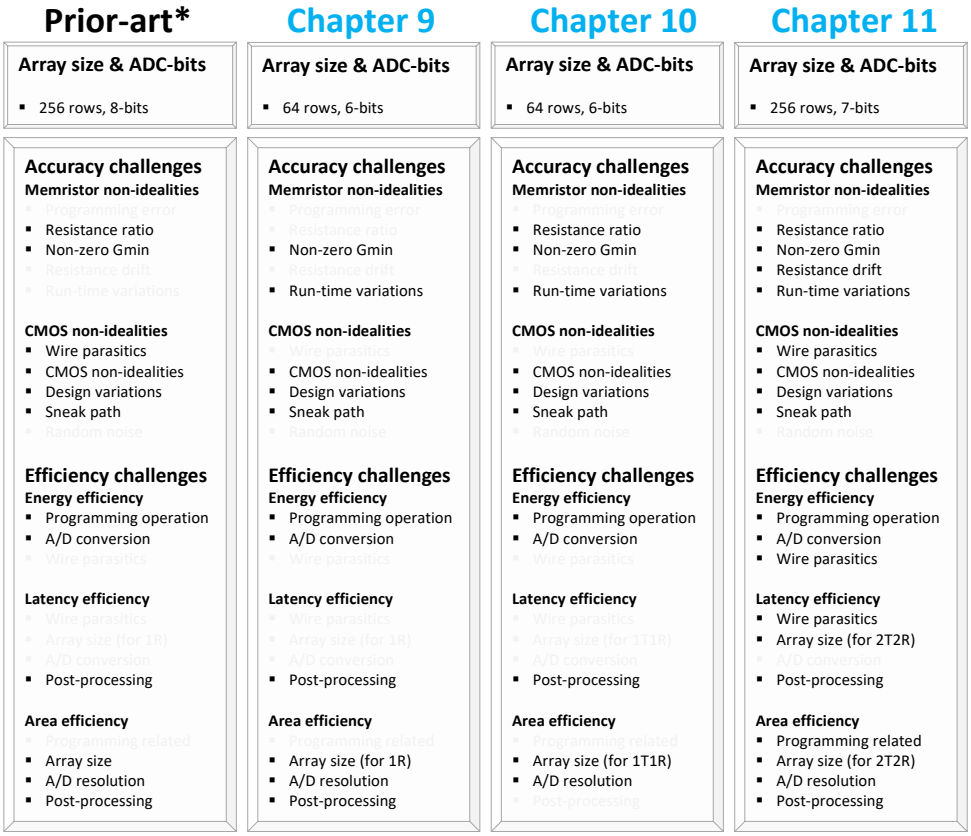
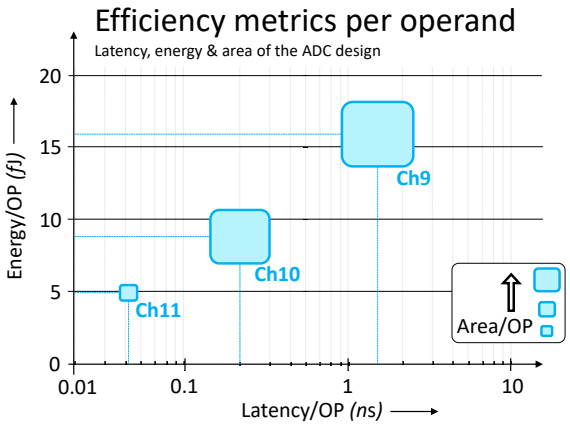


Figure 12.1: Summary of the challenges addressed by the prior-art and the ADCs for CIM accelerators proposed in each chapter. Additionally, the maximum array size and ADC resolution bits supported by these solutions. * including prior works till date for both voltage- and current-based ADC schemes for memristor-based CIM.

(described in Chapter 9 [69]), the key advancements include i) virtually fixing not only the BL voltage that connects into the ADC but instead fixing the difference of select-line (SL) and BL to a constant voltage with precise inter-matching design techniques, biasing schemes for the differential amplifiers (DA) for accurate A/D conversion, ii) the use of compact current-controlled ring-oscillator (RO) instead of using bulky IFC-based components including a comparator and integrating capacitors, iii) improving on the STP approach where wire delay mismatch, global variations and cycle to cycle variations of the CIM array are taken into account via a dummy row and a dummy column, and iv) additional power saving mode where the array can be disconnected and disabled after the amplifier output voltage is sampled. In addition, the design topology allows variable bit resolutions by regulating the maximum number of allowed pulse generations that correspond to the dummy column. Compensation techniques are also introduced



Specs/Column	Ch-9	Ch-10	Ch-11
Rows (#)	64	64	256
OPs (#)	127	127	511
Energy (*pJ*)	2.0	1.1	2.5
Energy/OP (*fJ*)	15.7	8.7	4.9
Latency (*ns*)	200	26	15
Latency/OP (*ns*)	1.6	0.2	0.03
Area (μm^2)	152	120	110
Area/OP (μm^2)	1.2	0.95	0.21
Bit resolution (#)	6	3-7	8
ENOB	3.7	4.3	6.1

Figure 12.2: Summary of the relative efficiency metrics of the proposed ADC designs for CIM.

to mitigate the inherent non-linearity of the oscillator-based conversion. Measurement results based on our 4Kb CIM chip prototype equipped with TSMC 40nm CMOS technology show that relative accuracy up to 99.71% and 94.74% realizing image classifications can be achieved for the MNIST and E-MNIST datasets, respectively. Energy efficiency of 115.1 TOPS/W, computational density of 12.1 TOPS/mm^2, and MVM latency of 26ns is achieved with 6-bit ADC realization.

Chapter 11 [71] describes the work that introduces a memory-periphery co-optimized approach to reduce the energy and area footprint of the ADC. When compared with the prior art and the preceding ADC designs of this thesis (Chapter 9 and Chapter 10 [69, 70]), the key advancements include i) applying true and complementary bitwise INPUTs to every pair of columns such that the total current flowing into the common ground of the pair is always constant. This alleviates the use of OTAs facilitating faster operations at low-power and low area utilization, ii) introducing a digital-intensive ADC design where a ramp-based voltage-to-time conversion (VTC) is realized for the A/D conversion. In addition, on-chip calibration and compensation schemes are introduced to address the mismatch due to process variations and IR drop, respectively. These key improvements lead to better scalability in terms of operand size where it is increased from 64 in the previous designs to 256. Measurement results based on our resistive CIM chip prototype equipped with TSMC 40nm CMOS technology show that relative accuracy up to 90.5% realizing image classifications can be achieved for the CIFAR-10 dataset. Energy efficiency of 206 TOPS/W, computational density of 16.1 TOPS/mm^2, and MVM latency of 15ns is achieved with signed 8-bit ADC realization.

12.3. DISCUSSION ON DESIGN EXPLORATIONS

This section discusses the applicability and adaptability of the proposed works with different design configurations.

MEMRISTOR TECHNOLOGIES

The proposed accelerators to perform analog-based MVM operations and the ADC design techniques that convert these analog MVM outputs into digital bits can be applicable to other memristor technologies. Note that the unique fabrication challenges related to a memristor technology, programming challenges related to switching threshold voltage and/or current challenges, and inherent characteristics such as endurance, resistance drift, and variation in programming states of the device need special design considerations. However, to some extent, these can be independent of the proposed A/D conversion techniques. Nevertheless, the type of memristor device does influence the absolute resistance values and the resistance ratio of the binary states, leading to modifying the circuit components when performing the analog CIM computing and the subsequent A/D conversion. The design techniques involved in the design SRIF-ADC [69] presented using RRAMs are expected to be applicable to other device types such as PCM. STT-MRAM, however, exhibits a low resistance ratio and cannot utilize these schemes to perform MVM. Similarly, other proposed design methodologies presented in RO-ADC [70] and VTC-ADC [71] can be expected to adapt easily to other device technologies. An exploration of dedicated compensation techniques required to address unique memristor-related non-idealities can be performed to develop these techniques for other emerging memristor devices.

MEMRISTOR-BASED BITCELL CONFIGURATIONS

The working of the proposed SRIF-ADC [69] and RO-ADC [70] are presented using 1R and 1T1R bitcells, respectively. Modifications related to stabilizing BL and/or SL may be required due to different bitcell configurations, however, certain techniques such as built-in weighing, built-in sample-and-hold, the STP mechanism, and the negative-feedback based DA arrangement to fix the access lines can be invariably utilized. In the case of VTC-ADC [71], the requirement of having true and complementary data bit storage in the memory crossbar array lies at the core of the underlying concept to function accordingly. It is worth mentioning that although 2T2R bitcell is utilized to demonstrate the concept, any bitcell that stores the data in both true and complementary form can ideally benefit from the techniques described in the related chapter. Although 1T1R and 2T2R bitcell configurations are the most widely explored currently, additional challenges related to emerging bitcell configurations would be required to be addressed, thus creating a new design space worth exploring.

MULTI-BIT CAPACITY & SCALING A/D RESOLUTION

All three propositions presented in the thesis to perform MVM operations are not restricted to data stored in binary or multi-bit form. Since all the given techniques use the generation of a proportional number of pulses with a given fixed period of time [69, 70] or proportional period of time to enable the generation of pulses at a fixed frequency [71], this period of time can be relatively expanded to extend the number of resolution bits while performing the A/D conversion. Note that the ideal number of possible quantization levels of the analog MVM output may increase by increasing the number of bits represented by each bitcell, the proposed techniques are expected to perform with the

same resolution. This is because, for a given fixed range of produced analog MVM values by the memory unit of CIM, ADC remains capable of resolving this to the same quantization levels (bit-resolutions). Nevertheless, it is worth exploring memristor technologies that exhibit multi-bit capability and can be extended to utilizing analog storage.

ADVANCED CMOS TECHNOLOGY

Technology scaling is amenable to ADC designs that involve digital-intensive components. This is because digital components such as counters, ring-oscillators, delay chains, etc., scale well with technology and voltage scaling in terms of energy consumption and performance compared to analog components such as amplifiers, comparators, etc. Some of the factors include i) technology (length) scaling leads to degradation in the inherent gain of the amplifiers, ii) voltage scaling leads to smaller voltage headroom for the transistor to operate in ideal regions of operation that leads to a reduction in the amplification factor, and iii) analog components are sensitive to increasing process variation with technology and voltage scaling as they suffer from increasing mismatch (with respect to loading, biasing, etc.) among the ideally matched circuit components. Nevertheless, switching KT/C noise, noise accumulation over the period of time ADC is enabled, and non-idealities related to memristors, CMOS, and wire parasitics, would arguably limit the applicability and scalability of analog components compared to digital components. Therefore, it can be seen that the proposed solutions tend to have more digital components compared to the preceding solutions and are primarily responsible for the improvement in performance metrics and scalability. Design explorations include further reducing the impact of analog components in the ADC design and focusing on how they can be potentially replaced by more digital components for better accuracy and efficiency.

SCALING OF ARRAY SIZE

Scaling the array size contributes to both accuracy and efficiency challenges. In relation to MVM accuracy, the inevitable IR drop due to finite wire parasitics and the dynamic range of MVM currents introduces a major limitation. Although the amplifiers can virtually fix the voltage of the access lines, the presence of IR drop causes variable voltage settling at different locations along the access lines. This also varies with the dynamic range of the currents as the extreme value of the current exhibits the maximum difference in the settled voltages. Additionally, regarding efficiency, amplifiers are required to virtually fix the access line voltages having finite bandwidth and therefore, require additional time (and energy) to settle the access lines of increasing length (with increasing RC loading). Therefore, as a design exploration, the A/D conversion techniques described in [69, 70] would require an improvement in the performance of the amplifiers typically at the cost of additional power and/or time. On the other hand, an increase in the number of footer devices in VTC-ADC scheme [71] with novel placement strategies would be required in order to mitigate the challenges related to IR drop. In addition, compensation schemes can be explored to compensate for the increased IR drop along the access lines.

12.4. CONCLUSION

This chapter puts into perspective how the proposed schemes are related and how they push state-of-the-art solutions and their preceding works in this thesis to achieve better efficiency and scalability. The key challenges addressed, the targeted bit resolution, and the array size are summarized for each of the proposed logic accelerators. A brief discussion on how the proposed design concepts can be applicable and how they can be explored while investigating various design choices *i.e.,* memory technology, bitcell configurations, scalability in terms of array size, and bit resolution.

PART IV: THE CONCLUSION

This part includes the following chapter.

- ***Chapter 13**: Summary and future directions*

13

SUMMARY AND FUTURE DIRECTIONS

This chapter concludes the thesis. The preceding chapters are summarized followed by suggestive directions for future design explorations to demonstrate end-to-end realization of real-world applications.

13.1. SUMMARY

This section provides an executive summary of the thesis, highlighting the key concept of the proposed accelerator and the performance metrics compared to the state-of-the-art solutions.

PART I: THE FOUNDATION

- **Chapter 1** described the importance of computing in our daily lives, highlighting the key challenges of conventional computing and the promise of memristor-based CIM to find low-cost and energy-efficient solutions. We covered the primary limitations of conventional computing based on traditional CMOS device technology and introduced non-volatile emerging memory technology-based CIM to accelerate certain logic and arithmetic operations in an energy-efficient manner. We described the hierarchy of CIM computing including the software and hardware stacks while focusing on micro-architectural opportunities and challenges to implement CIM based on memristors. Thereafter, we provide a brief summary of the logic and arithmetic accelerators proposed as part of the thesis, describing the challenges addressed, and the performance metrics achieved compared to state-of-the-art solutions. The chapter concludes with the outline of the thesis.

- **Chapter 2** presented the background on memristor-based CIM. We presented the fundamentals of conventional computing and how near-memory computing and CIM perform certain computational tasks differently for memory-centric applications. It also covered the different volatile and non-volatile memory technologies that are been investigated for implementing CIM. The key components of a typical CIM architecture and their impact on overall computational accuracy and efficiency were highlighted. A comprehensive account of the challenges of memristor-based CIM follows and the opportunities they provide for enabling better performance were discussed. Finally, a study on state-of-the-art CIM-based logic and arithmetic accelerators was presented which are classified based on the challenges they attempt to address. The major limitations of these solutions were highlighted which gives motivation to the work that is conducted as part of this thesis.

PART II: CIM-BASED LOGIC ACCELERATORS

- **Chapter 3** presented the concept of in-memory database query where the database is stored in a dense array of memristive devices, and the queries are performed in place by employing in-memory logic operations. A 4×8 selector-less crossbar populated with Proj-PCM devices was fabricated and characterized to demonstrate experimentally the errorless execution of database queries based on the concept of scouting logic. To simulate array sizes that are realistic for real-world applications, a circuit model was developed in order to emulate the nonlinear electrical behavior and the non-volatility of Proj-PCM devices. Quantitative studies were carried out on the array size limitations as a function of the wiring resistance, the inter-device conductance variations, and the difficulty of programming the target device without affecting the stored data in the rest of the crossbar. We introduced the concept of *cascaded logic*, a system capable of executing queries of arbitrary size and complexity. It is based on a conve-

nient query decomposition into a multitude of alternating AND & OR operations and it combines in-memory logic using resistive crossbars with peripheral CMOS logic. This system managed to process database queries with massive parallelism, high throughput (92.6 GOPS), and low energy consumption (166 TOPS/W) solving a database query problem using a real healthcare-related database.

- **Chapter 4** presented a novel referencing scheme using STT-MRAM-based CIM to perform robust logic operations in the presence of design non-idealities. The technique involves an adaptive referencing mechanism to improve the sensing margin of a CIM architecture for logic operations. We generate reference signals using multiple STT-MRAM devices and place them strategically into the array such that these signals can address the variations and trace the wire parasitics effectively. The chapter validates our proposed design using calibrated design parameters extracted from a silicon-verified characterization chip. Performance metrics are determined for logic operations which are employed for a system-level framework and evaluated for BNN and text encryption applications. Results show that our implementation achieves up to 17.8 TOPS/W on the MNIST dataset and 130× performance *i.e.* execution time improvement is expected compared to software implementation on Intel Haswell processor.

- **Chapter 5** demonstrated how adding a dummy row in a CIM crossbar based on RRAM can enable robust and reliable multi-operand bitwise logic operation. A voltage-based differential referencing-in-array scheme was presented that enables accurate two and multi-operand logic operations for RRAM-based CIM architecture. The scheme makes use of a 2T2R cell configuration to create a complementary bitcell structure that inherently acts also as a reference during the operation execution; this results in a high sensing margin. Moreover, the variation-sensitive multi-operand (N)AND operation is implemented using complementary-input (N)OR operation to further improve its accuracy. This not only makes the generation of references much simpler, but it also enables wider sensing margins. In addition, the chapter has shown how some basic circuits can be integrated with CIM to suppress/slow the degradation and boost the robustness of CIM logic operations while trading off some latency. Comparison results using 512x512 CIM core show that our proposed design offers an improvement of up to 11.4× in terms of energy efficiency while performing up to 56 operands in a single cycle.

- **Chapter 6** presented cost-efficient CIM-based DFT schemes that improve the cost of testing and fault coverage (FC). Our schemes deploy multi-operand NOR logic operations that involve multi-row select read operations to accelerate the testing of ETD faults. In addition, reconfigurable logic designs are developed to detect unique RRAM faults (HTD faults) that offer features such as programmable reference generations, period, and voltage of operation. A novel addressing scheme is introduced to facilitate the diagnosis of faults. The design and implementation of a fast search algorithm facilitates the diagnosis of faults. Our proposed DFT implementations are validated on a post-layout extracted platform and testing sequences are introduced incorporating the proposed DFTs. Results show that more than 2.3× speedup, 6× area reduction, and better FC are achieved compared to the state-of-the-art.

- **Chapter 7** presented MOXOR-CIM which realizes novel voltage-to-time conversion (VTC) based approaches to perform multi-operand XOR reliably in a single cycle for RRAM-based CIM architectures. In our schemes, CIM generates a voltage signal proportional to the operand data, and two VTC-based approaches are presented to determine if the operand has an ODD or an EVEN parity. We utilize the bitline capacitances for a voltage-based sensing for computation which generates the output voltage value being linear to the operand values; the voltage is then converted into desired logic output using the VTC. Additionally, low-power techniques are introduced in deploying sense amplifiers such as regulating power consumption during its operation and disabling it at the instant amplifier decision is made. The two schemes proposed offer a desirable trade-off between computing efficiency and area overhead. Comparison results using 512x512 RRAM-based CIM show an improvement of up to 12× in performance and 5.5× energy efficiency, while also extending the maximum number of operands supported in a single cycle to 16.

- **Chapter 8** presented the progression of the proposed logic accelerator solutions over the period of the thesis work and highlighted the key challenges that are addressed and how the progression can be described in terms of performance metrics. The comparison was made with the existing state-of-the-art solutions at the time and the preceding solutions presented in the thesis. We also discussed how the applicability and scalability of the proposed solutions with different design choices can be explored.

PART III: CIM-BASED ARITHMETIC ACCELERATORS

- **Chapter 9** presented a novel ADC design methodology that allows key design features and components to provide an efficient solution. We proposed a technique to stabilize the node receiving analog inputs that allows better scalability in terms of higher parallelism of operations. We employed a self-timed variation-aware design approach and design measures to drastically reduce read disturb of memristor devices that address reliability issues related to the ADC design. In addition, we presented a compact, built-in sample-and-hold circuit to replace the large-sized capacitance and built-in weighting technique to alleviate the need for post-processing. Compared to compact and energy-efficient IFC, we obtained a gain of 2X speed, > 3X range, 2.7X area, and 11.6X energy efficiency. Compared to fast and accurate SAR-ADC, SRIF-ADC is 144X area and 56X energy-efficient with a reduced speed of nearly 0.4X but with similar range and accuracy. Circuit design solutions of two, four operands *4-bit* adders and *4-bit* digital comparator were also presented to showcase the versatility of the proposed circuit designs.

- **Chapter 10** presented a novel memory-periphery co-design to perform accurate A/D conversions of analog MVM outputs with high energy efficiency and computational density. The chapter introduced a scheme where array access lines can be virtually fixed to improve the accuracy of the MVM operation. This facilitates a path for a compact and ultra-low power ADC design using a ring-oscillator-based A/D conversion, equipped with component sharing and inter-matching of the reference blocks. In addition, we deploy a self-timed technique to further ensure high robustness addressing global design and cycle-to-cycle variations. The measurement results of a 4Kb

CIM chip prototype equipped with TSMC 40nm validate the ADC design and derive its input-output characteristics. A relative accuracy of up to 99.71% is achieved with an energy efficiency of 115.1 TOPS/W and computational density of 12.1 TOPS/mm^2 for the MNIST dataset.

- **Chapter 11** explored the true and complementary nature of data storage to provide differential analog MVM output signals that aid the A/D conversion for high energy efficiency and computational density. The chapter described how providing complementary inputs to adjacent columns can aid the settling of the common select line at a fixed value by virtue, which alleviated the use of expensive OTA. In addition, the architecture generates a global drift compensation that addresses the inevitable drift of resistance values of the memristor device. Complementary voltage-based sensing is utilized followed by a compact, energy-efficient voltage-to-time (VTC) based ADC for A/D conversion. The measurement results of a 256x8 CIM chip prototype equipped with TSMC 40nm validate the ADC design and derive its input-output characteristics. A relative accuracy of up to 95.1% is achieved with an energy efficiency of 206 TOPS/W for the CIFAR dataset, thus, making it a suitable implementation for executing MVM operations in low-power edge AI devices.

- **Chapter 12** presented the progression of the proposed ADC designs as arithmetic accelerator solutions over the period of the thesis work and highlighted the key challenges that are addressed and how the progression can be described in terms of performance metrics. The comparison was made with the existing state-of-the-art solutions at the time and the preceding solutions presented in the thesis. We also discussed how the applicability and scalability of the proposed solutions with different design choices can be explored.

13.2. FUTURE DIRECTIONS

This section discusses possible avenues where the proposed solutions can be investigated in the future to demonstrate real-world applications.

END-TO-END APPLICATIONS

The ideal way to demonstrate the efficiency of the proposed memristor-based CIM logic and arithmetic accelerators is to deploy them in hardware units capable of running end-to-end applications. While the proposed concepts work on a comprehensive SPICE simulator and small-scale prototypes, the overall improvement of the system can be eventually determined while taking into account several factors. For instance, as highlighted in subsequent points, i) the data sets can be extended to represent multi-bit data bits, ii) the size of datasets can be expanded to fit on-chip weights/data for deep neural network applications or database query applications, iii) the trade-off of having larger tile size and having more number of tiles, and iv) determine peak and average efficiency achieved when mapping these real-world applications due to sparsity of datasets mapped on the memory tiles.

ROW-DECODING SCHEMES FOR LOGIC

In most of the presented works on logic accelerators, the focus was primarily on performing the logic operations while assuming that the operand locations are selected. While the addressing sequence and the associated circuit design to accelerate the testing RRAMs were presented, the rest of the solutions excluded the addressing or row-decoding schemes. While two-operand logic accelerators can have a duplicated row-decoding scheme (duplicating the row-decoder of conventional SRAM-based decoding circuitry), performing multi-operand operations may not be easily scalable in the same manner. The target application would provide the expected pattern of the selected address locations and may not be completely random. For instance, in database query applications with a series of logic operations to be performed, the preceding logic outputs typically determine the succeeding address locations.

SIGNED MULTI-BIT MVM

In most of the presented works on ADC for MVM accelerators, both the input vector and matrix residing in the memory array are in binary form to demonstrate the novel ADC concepts. SRIF-ADC does explore the 4-bit elements in the memory array, albeit on different bitcells, the rest of the solutions are demonstrated using a binary format of input operands. Realizing neural networks for image classification applications typically involves multi-bit activation inputs and weights, where different decoding schemes can be used to map these on CIM. For instance, converting signed multi-bit inputs into analog form includes voltage modulation, time modulation, or a hybrid of the two schemes. The design considerations for applying the inputs whether at the select lines or at word lines can be explored as well. On the other hand, mapping multi-bit weights onto the memristor devices can utilize one or multiple memristors/bitcells in digital or analog form. The aggregation of several intermediate outputs with varying bit significance with novel techniques can be explored.

HYBRID ADC TOPOLOGIES

We explored ADC topologies based on integrate-and-fire components, current-controlled oscillators, and voltage-to-time converters that push the state-of-the-art toward better MVM accuracy and energy efficiency. However, several articles focusing on ADC design using hybrid topologies for high-performing conversion that do not target CIM architecture can be explored for CIM. Nevertheless, the design considerations concerning CIM architecture need to be taken into account. For instance, a hybrid approach of SAR and flash has gathered a lot of attention for multi-Giga samples per second which is extremely over-designed in terms of performance for CIM architecture. Arguably a slower, compact version of such a scheme can be potentially explored for CIM with high energy efficiency at the cost of performance. Other topologies include using multi-phased SA, and ramp-based VTC which can also be combined with the aforementioned schemes to develop efficient conversion.

MULTI-TILE ARCHITECTURE AND TILE SIZE

To realize end-to-end applications, large datasets that must be mapped on-chip are in the order of MB and even GB with signed multi-bit elements. These datasets typically require multi-tile architectures with each tile capable of storing segments of data. For instance, in multi-layered DNN applications, each layer of the neural network can be mapped to one or several tiles and the mapping must be ideally such that the communication of these tiles is optimized. In these explorations, the choice of having more tiles or maximizing the size of each tile can present an interesting trade-off in terms of overall throughput, area footprint, and energy consumption. These would mostly include the wire parasitics limiting the performance of the CIM tile and the inter-tile data communication that could potentially occur among several tiles.

CIM BASED ON CONVENTIONAL MEMORIES

Future explorations could potentially involve testing the limits to which CIM based on traditional volatile SRAMs and DRAMs, and non-volatile NAND/NOR flash memories can provide in terms of accuracy and efficiency. Possible avenues also include non-stationary datasets which require frequent re-programming that may not be ideal for emerging memristor devices. There are concerns regarding CMOS technology in terms of scalability, reliability, and cost involved, nevertheless, since there are well-established manufacturing facilities to fabricate look-ahead prototypes, CIM architecture based on these is worth exploring. 3D NAND/ NOR flash memories are capable of storing massive signed multi-bit datasets in digital or analog form as these datasets are typically required for realizing fully on-chip DNN applications. With one or several additional tiers of periphery logic equipped with pre-processing of inputs (D/A conversion) and post-processing of analog MVM outputs (A/D conversion), this avenue is also worth exploring.

BIBLIOGRAPHY

[1] A. Singh *et al.* "Low-power Memristor-based Computing for Edge-AI Applications". In: *ISCAS*. 2021.

[2] A. Gebregiorgis, A. Singh, S. Diware, R. Bishnoi, and S. Hamdioui. "Dealing with non-idealities in memristor based computation-in-memory designs". In: *VLSI-SoC*. 2022.

[3] A. Gebregiorgis, A. Singh, A. Yousefzadeh, D. Wouters, R. Bishnoi, F. Catthoor, and S. Hamdioui. "Tutorial on memristor-based computing for smart edge applications". In: *Memories-Materials, Devices, Circuits and Systems* 4 (2023), p. 100025.

[4] *Big data predictions analytics trends in 2020*. URL: https://www.techrepublic.com/article/big-data-predictions-8-analytics-trends-in-2020/.

[5] *Top IoT trends for mobile platform 2019*. URL: https://www.solutionanalysts.com/blog/top-iot-app-development-trends-for-mobile-platform-2019.

[6] *Edge Computing Market Will Grow at a Steady Rate of 41.0% CAGR from 2018 to 2025: Grand View Research, Inc.* URL: https://www.grandviewresearch.com/industry-analysis/edge-computing-market.

[7] *Global AI investment to top 150 billion by 2025*. URL: https://outsideinsight.com/insights/global-ai-investment-150-billion-2025/.

[8] S. Hamdioui, L. Xie, H. A. D. Nguyen, M. Taouil, K. Bertels, H. Corporaal, H. Jiao, F. Catthoor, D. Wouters, L. Eike, *et al.* "Memristor based computation-in-memory architecture for data-intensive applications". In: *Proceedings of the 2015 design, automation & test in Europe conference & exhibition*. EDA Consortium. 2015, pp. 1718–1725.

[9] *LLM Explained: The LLM Training Landscape*. URL: https://liu-gendary.medium.com/llm-explained-the-llm-training-landscape-82c803495caa.

[10] *Increasing gap in performance for memory and processing units*. URL: https://www.cs.ucr.edu/~elaheh/papers/TECHCON19_presentation.pdf.

[11] *Static and dynamic power with technology scaling*. URL: https://www.semiconductors.org/wp-content/uploads/2018/08/2011SysDrivers.pdf.

[12] *Static and dynamic power of memory and logic units with technology scaling*. URL: http://www.%20itrs.net/Links/2012ITRS/2012Chapters/2012Overview.pdf.

[13] *Cost of manufacturing advanced CMOS technologies*. URL: https://faculty-web.msoe.edu/johnsontimoj/EE4980/files4980/cmos_cost.pdf.

[14] *Microprocessor trend data in the last 50 years*. URL: https://github.com/karlrupp/microprocessor-trend-data/blob/master/50yrs/50-years-processor-trend.pdf.

[15] J. Chhugani, A. D. Nguyen, V. W. Lee, W. Macy, M. Hagog, Y.-K. Chen, A. Baransi, S. Kumar, and P. Dubey. "Efficient implementation of sorting on multi-core SIMD CPU architecture". In: *Proceedings of the VLDB Endowment* 1.2 (2008), pp. 1313–1324.

[16] M. J. Mayfield, F. P. O'connell, and D. S. Ray. *Cache prefetching of L2 and L3*. US Patent 6,446,167. 2002.

[17] D. H. Lawrie. "Access and alignment of data in an array processor". In: *IEEE Transactions on Computers* 100.12 (1975), pp. 1145–1155.

[18] R. Lin and M. Margala. *Multiplier-based processor-in-memory architectures for image and graphics processing*. US Patent 7,167,890. 2007.

[19] X. Liu, M. Mao, B. Liu, H. Li, Y. Chen, B. Li, Y. Wang, H. Jiang, M. Barnell, Q. Wu, *et al.* "RENO: A high-efficient reconfigurable neuromorphic computing accelerator design". In: *2015 52nd ACM/EDAC/IEEE Design Automation Conference (DAC)*. IEEE. 2015, pp. 1–6.

[20] T. Luo, S. Liu, L. Li, Y. Wang, S. Zhang, T. Chen, Z. Xu, O. Temam, and Y. Chen. "DaDianNao: A neural network supercomputer". In: *IEEE Transactions on Computers* 66.1 (2017), pp. 73–88.

[21] M. Horowitz. "1.1 computing's energy problem (and what we can do about it)". In: *2014 IEEE international solid-state circuits conference digest of technical papers (ISSCC)*. IEEE. 2014, pp. 10–14.

[22] *Increasing failure rates with technology scaling*. URL: https://nepp.nasa.gov/files/16365/08_102_4_%20JPL_White.pdf.

[23] A. Sebastian, M. Le Gallo, R. Khaddam-Aljameh, and E. Eleftheriou. "Memory devices and applications for in-memory computing". In: *Nature Nanotechnology* (2020), pp. 1–16.

[24] R. Waser, R. Dittmann, G. Staikov, and K. Szot. "Redox-based resistive switching memories–nanoionic mechanisms, prospects, and challenges". In: *Advanced materials* 21.25-26 (2009), pp. 2632–2663.

[25] T. Kawahara, K. Ito, R. Takemura, and H. Ohno. "Spin-transfer torque RAM technology: Review and prospect". In: *Microelectronics Reliability* 52.4 (2012), pp. 613–627.

[26] G. W. Burr, M. J. Breitwisch, M. Franceschini, D. Garetto, K. Gopalakrishnan, B. Jackson, B. Kurdi, C. Lam, L. A. Lastras, A. Padilla, *et al.* "Phase change memory technology". In: *Journal of Vacuum Science & Technology B* 28.2 (2010), pp. 223–262.

[27] *In-memory computing market*. URL: https://www.marketsandmarkets.com/Market-Reports/in-memory-computing-820.html.

[28] *Growth of in-memory computing in the coming years, with market share, global industry trends*. URL: https://www.imarcgroup.com/in-memory-computing-market.

[29] M. A. Lebdeh, U. Reinsalu, H. A. Du Nguyen, S. Wong, and S. Hamdioui. "Memristive device based circuits for computation-in-memory architectures". In: *2019 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE. 2019, pp. 1–5.

[30] M. Zahedi, T. Shahroodi, G. Custers, A. Singh, S. Wong, and S. Hamdioui. "System Design for Computation-in-Memory: From Primitive to Complex Functions". In: *2022 IFIP/IEEE 30th International Conference on Very Large Scale Integration (VLSI-SoC)*. 2022, pp. 1–6. DOI: 10.1109/VLSI-SoC54400.2022.9939571.

[31] S. Harizopoulos, V. Liang, D. J. Abadi, and S. Madden. "Performance tradeoffs in read-optimized databases". In: *Proceedings of the 32nd international conference on Very large data bases*. Citeseer. 2006, pp. 487–498.

[32] A. Biswas and A. P. Chandrakasan. "CONV-SRAM: An energy-efficient SRAM with in-memory dot-product computation for low-power convolutional neural networks". In: *IEEE Journal of Solid-State Circuits* 54.1 (2018), pp. 217–230.

[33] X. Si, J.-J. Chen, Y.-N. Tu, W.-H. Huang, J.-H. Wang, Y.-C. Chiu, W.-C. Wei, S.-Y. Wu, X. Sun, R. Liu, *et al.* "24.5 A twin-8T SRAM computation-in-memory macro for multiple-bit CNN-based machine learning". In: *International Solid-State Circuits Conference-(ISSCC)*. 2019, pp. 396–398.

[34] S. Yin, Z. Jiang, J.-S. Seo, and M. Seok. "XNOR-SRAM: In-memory computing SRAM macro for binary/ternary deep neural networks". In: *IEEE Journal of Solid-State Circuits* 55.6 (2020), pp. 1733–1743.

[35] Q. Dong *et al.* "15.3 A 351TOPS/W and 372.4 GOPS Compute-in-Memory SRAM Macro in 7nm FinFET CMOS for Machine-Learning Applications". In: *2020 IEEE International Solid-State Circuits Conference-(ISSCC)*. IEEE. 2020, pp. 242–244.

[36] M. Ali, A. Jaiswal, S. Kodge, A. Agrawal, I. Chakraborty, and K. Roy. "IMAC: In-memory multi-bit multiplication and accumulation in 6T SRAM array". In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 67.8 (2020), pp. 2521–2531.

[37] B. Yan, J.-L. Hsu, P.-C. Yu, C.-C. Lee, Y. Zhang, W. Yue, G. Mei, Y. Yang, Y. Yang, H. Li, *et al.* "A 1.041-Mb/MM 2 27.38-TOPS/W signed-INT8 dynamic-logic-based ADC-less SRAM compute-in-memory macro in 28nm with reconfigurable bitwise operation for AI and embedded applications". In: *2022 IEEE International Solid-State Circuits Conference (ISSCC)*. Vol. 65. IEEE. 2022, pp. 188–190.

[38] C. Yu, T. Yoo, K. T. C. Chai, T. T.-H. Kim, and B. Kim. "A 65-nm 8T SRAM compute-in-memory macro with column ADCs for processing neural networks". In: *IEEE Journal of Solid-State Circuits* 57.11 (2022), pp. 3466–3476.

[39] P.-C. Wu, J.-W. Su, Y.-L. Chung, L.-Y. Hong, J.-S. Ren, F.-C. Chang, Y. Wu, H.-Y. Chen, C.-H. Lin, H.-M. Hsiao, *et al.* "A 28nm 1Mb time-domain computing-in-memory 6T-SRAM macro with a 6.6 ns latency, 1241GOPS and 37.01 TOPS/W for 8b-MAC operations for edge-AI devices". In: *2022 IEEE International Solid-State Circuits Conference (ISSCC)*. Vol. 65. IEEE. 2022, pp. 1–3.

[40] H. Wang, R. Liu, R. Dorrance, D. Dasalukunte, D. Lake, and B. Carlton. "A Charge Domain SRAM Compute-in-Memory Macro With C-2C Ladder-Based 8-Bit MAC Unit in 22-nm FinFET Process for Edge Inference". In: *IEEE Journal of Solid-State Circuits* 58.4 (2023), pp. 1037–1050.

[41] C. Yu, T. Yoo, H. Kim, T. T.-H. Kim, K. C. T. Chuan, and B. Kim. "A logic-compatible eDRAM compute-in-memory with embedded ADCs for processing neural networks". In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 68.2 (2020), pp. 667–679.

[42] S. Xie, C. Ni, A. Sayal, P. Jain, F. Hamzaoglu, and J. P. Kulkarni. "16.2 eDRAM-CIM: Compute-in-memory design with reconfigurable embedded-dynamic-memory array realizing adaptive data converters and charge-domain computing". In: *International Solid-State Circuits Conference-(ISSCC)*. Vol. 64. 2021, pp. 248–250.

[43] L. Zhao, C. Yan, F. Yang, S. Gao, G. Rosca, D. Manea, Z. Lu, and Y. Zhao. "A compute-in-memory architecture compatible with 3D NAND flash that parallelly activates multi-layers". In: *2021 58th ACM/IEEE Design Automation Conference (DAC)*. IEEE. 2021, pp. 193–198.

[44] H.-W. Hu, W.-C. Wang, C.-K. Chen, Y.-C. Lee, B.-R. Lin, H.-M. Wang, Y.-P. Lin, Y.-C. Lin, C.-C. Hsieh, C.-M. Hu, *et al.* "A 512Gb in-memory-computing 3D-NAND flash supporting similar-vector-matching operations on edge-AI devices". In: *2022 IEEE International Solid-State Circuits Conference (ISSCC)*. Vol. 65. IEEE. 2022, pp. 138–140.

[45] M. Prezioso, F. Merrikh-Bayat, B. D. Hoskins, G. C. Adam, K. K. Likharev, and D. B. Strukov. "Training and operation of an integrated neuromorphic network based on metal-oxide memristors". In: *Nature* 521.7550 (2015), pp. 61–64.

[46] R. Mochida, K. Kouno, Y. Hayata, M. Nakayama, T. Ono, H. Suwa, R. Yasuhara, K. Katayama, T. Mikawa, and Y. Gohou. "A 4M synapses integrated analog ReRAM based 66.5 TOPS/W neural-network processor with cell current controlled writing and flexible network architecture". In: *2018 IEEE Symposium on VLSI Technology*. IEEE. 2018, pp. 175–176.

[47] B. Yan *et al.* "RRAM-based Spiking Nonvolatile Computing-In-Memory Processing Engine with Precision-Configurable In Situ Nonlinear Activation". In: *Symposium on VLSI Technology*. IEEE. 2019, T86–T87.

[48] C.-X. Xue, W.-H. Chen, J.-S. Liu, J.-F. Li, W.-Y. Lin, W.-E. Lin, J.-H. Wang, W.-C. Wei, T.-W. Chang, T.-C. Chang, *et al.* "24.1 A 1Mb multibit ReRAM computing-in-memory macro with 14.6 ns parallel MAC computing time for CNN based AI edge processors". In: *2019 IEEE International Solid-State Circuits Conference-(ISSCC)*. IEEE. 2019, pp. 388–390.

[49] W.-H. Chen, C. Dou, K.-X. Li, W.-Y. Lin, P.-Y. Li, J.-H. Huang, J.-H. Wang, W.-C. Wei, C.-X. Xue, Y.-C. Chiu, *et al.* "CMOS-integrated memristive non-volatile computing-in-memory for AI edge processors". In: *Nature, 2019* ().

[50] W. Wan, R. Kubendran, S. B. Eryilmaz, W. Zhang, Y. Liao, D. Wu, S. Deiss, B. Gao, P. Raina, S. Joshi, *et al.* "33.1 A 74 TMACS/W CMOS-RRAM neurosynaptic core with dynamically reconfigurable dataflow and in-situ transposable weights for probabilistic graphical models". In: *2020 IEEE International Solid-State Circuits Conference-(ISSCC)*. IEEE. 2020, pp. 498–500.

[51] Q. Liu, B. Gao, P. Yao, D. Wu, J. Chen, Y. Pang, W. Zhang, Y. Liao, C.-X. Xue, W.-H. Chen, *et al.* "33.2 A fully integrated analog ReRAM based 78.4 TOPS/W compute-in-memory chip with fully parallel MAC computing". In: *International Solid-State Circuits Conference-(ISSCC)*. 2020, pp. 500–502.

[52] W. Wan, R. Kubendran, B. Gao, S. Joshi, P. Raina, H. Wu, G. Cauwenberghs, and H. P. Wong. "A voltage-mode sensing scheme with differential-row weight mapping for energy-efficient RRAM-based in-memory computing". In: *2020 IEEE Symposium on VLSI Technology*. IEEE. 2020, pp. 1–2.

[53] J.-M. Hung, C.-X. Xue, H.-Y. Kao, Y.-H. Huang, F.-C. Chang, S.-P. Huang, T.-W. Liu, C.-J. Jhang, C.-I. Su, W.-S. Khwa, *et al.* "A four-megabit compute-in-memory macro with eight-bit precision based on CMOS and resistive random-access memory for AI edge devices". In: *Nature Electronics* 4.12 (2021), pp. 921–930.

[54] C.-X. Xue, Y.-C. Chiu, T.-W. Liu, T.-Y. Huang, J.-S. Liu, T.-W. Chang, H.-Y. Kao, J.-H. Wang, S.-Y. Wei, C.-Y. Lee, *et al.* "A CMOS-integrated compute-in-memory macro based on resistive random-access memory for AI edge devices". In: *Nature Electronics* 4.1 (2021), pp. 81–90.

[55] S. Jung *et al.* "A crossbar array of magnetoresistive memory devices for in-memory computing". In: *Nature* (2022).

[56] Y.-C. Chiu, C.-S. Yang, S.-H. Teng, H.-Y. Huang, F.-C. Chang, Y. Wu, Y.-A. Chien, F.-L. Hsieh, C.-Y. Li, G.-Y. Lin, *et al.* "A 22nm 4Mb STT-MRAM data-encrypted near-memory computation macro with a 192GB/s read-and-decryption bandwidth and 25.1-55.1 TOPS/W 8b MAC for AI operations". In: *2022 IEEE International Solid-State Circuits Conference (ISSCC)*. Vol. 65. IEEE. 2022, pp. 178–180.

[57] M. Ishii, S. Kim, S. Lewis, A. Okazaki, J. Okazawa, M. Ito, M. Rasch, W. Kim, A. Nomura, U. Shin, *et al.* "On-chip trainable 1.4 M 6T2R PCM synaptic array with 1.6 K stochastic LIF neurons for spiking RBM". In: *2019 IEEE International Electron Devices Meeting (IEDM)*. IEEE. 2019, pp. 14–2.

[58] P. Narayanan, S. Ambrogio, A. Okazaki, K. Hosokawa, H. Tsai, A. Nomura, T. Yasuda, C. Mackin, S. C. Lewis, A. Friz, *et al.* "Fully on-chip MAC at 14 nm enabled by accurate row-wise programming of PCM-based weights and parallel vector-transport in duration-format". In: *IEEE Transactions on Electron Devices* 68.12 (2021), pp. 6629–6636.

[59] M. Le Gallo, R. Khaddam-Aljameh, M. Stanisavljevic, A. Vasilopoulos, B. Kersting, M. Dazzi, G. Karunaratne, M. Brändli, A. Singh, S. M. Mueller, *et al.* "A 64-core mixed-signal in-memory compute chip based on phase-change memory for deep neural network inference". In: *Nature Electronics* (2023), pp. 1–14.

[60] W.-S. Khwa, Y.-C. Chiu, C.-J. Jhang, S.-P. Huang, C.-Y. Lee, T.-H. Wen, F.-C. Chang, S.-M. Yu, T.-Y. Lee, and M.-F. Chang. "A 40-nm, 2M-cell, 8b-precision, hybrid SLC-MLC PCM computing-in-memory macro with 20.5-65.0 TOPS/W for tiny-Al edge devices". In: *International Solid-State Circuits Conference-(ISSCC)*. Vol. 65. 2022, pp. 1–3.

[61] S. Sakhare *et al.* "Enablement of STT-MRAM as last level cache for high performance computing domain at 5nm node". In: *IEDM*. 2018.

[62] R. Khaddam-Aljameh, M. Stanisavljevic, J. F. Mas, G. Karunaratne, M. Brändli, F. Liu, A. Singh, S. M. Müller, U. Egger, A. Petropoulos, *et al.* "HERMES-core—A 1.59-TOPS/mm 2 PCM on 14-nm CMOS in-memory compute core using 300-ps/LSB linearized CCO-based ADCs". In: *Journal of Solid-State Circuits* 57.4 (2022), pp. 1027–1038.

[63] R. Waser. "Redox-based resistive switching memories". In: *Journal of nanoscience and nanotechnology* 12.10 (2012), pp. 7628–7640.

[64] I. Giannopoulos *et al.* "In-Memory Database Query". In: *Advanced Intelligent Systems* 2.12 (2020), p. 2000141.

[65] A. Singh, M. Zahedi, T. Shahroodi, M. Gupta, A. Gebregiorgis, M. Komalan, R. V. Joshi, F. Catthoor, R. Bishnoi, and S. Hamdioui. "CIM-based robust logic accelerator using 28nm STT-MRAM characterization chip tape-out". In: *AICAS*. 2022.

[66] A. Singh, R. Bishnoi, R. V. Joshi, and S. Hamdioui. "Referencing-in-array scheme for RRAM-based CIM architecture". In: *DATE*. 2022.

[67] A. Singh, M. Fieback, R. Bishnoi, F. Bradarić, A. Gebregiorgis, R. V. Joshi, and S. Hamdioui. "Accelerating RRAM Testing with a Low-cost Computation-in-Memory based DFT". In: *ITC*. 2022.

[68] A. Singh, R. Bishnoi, R. Joshi, and S. Hamdioui. "MOXOR-CIM: Energy-efficient Multi-operand XOR Logic based Computation-in-Memory Accelerator". In: *Submitted in DATE* 4 (2024), p. 100025.

[69] A. Singh, M. A. Lebdeh, A. Gebregiorgis, R. Bishnoi, R. V. Joshi, and S. Hamdioui. "SRIF: Scalable and reliable integrate and fire circuit ADC for memristor-based CIM architectures". In: *TCAS-I* (2021).

[70] A. Singh, R. Bishnoi, A. Kaichouhi, S. Diware, R. V. Joshi, and S. Hamdioui. "A 115.1 TOPS/W, 12.1 TOPS/mm 2 Computation-in-Memory using Ring-Oscillator based ADC for Edge AI". In: *2023 IEEE 5th International Conference on Artificial Intelligence Circuits and Systems (AICAS)*. IEEE. 2023, pp. 1–5.

[71] A. Singh, R. Bishnoi, A. Kaichouhi, S. Diware, R. V. Joshi, and S. Hamdioui. "Efficient Computation-in-memory using complementary data for low-power Edge-AI". In: *Yet to be submitted*. IEEE. 2024, pp. x–x.

[72] F. Akopyan, J. Sawada, A. Cassidy, R. Alvarez-Icaza, J. Arthur, P. Merolla, N. Imam, Y. Nakamura, P. Datta, G.-J. Nam, *et al.* "Truenorth: Design and tool flow of a 65 mw 1 million neuron programmable neurosynaptic chip". In: *IEEE transactions on computer-aided design of integrated circuits and systems* 34.10 (2015), pp. 1537–1557.

[73] Y. Xiang and H. Kim. "Pipelined data-parallel CPU/GPU scheduling for multi-DNN real-time inference". In: *2019 IEEE Real-Time Systems Symposium (RTSS)*. IEEE. 2019, pp. 392–405.

[74] Y. E. Wang, G.-Y. Wei, and D. Brooks. "Benchmarking TPU, GPU, and CPU platforms for deep learning". In: *arXiv preprint arXiv:1907.10701* (2019).

[75] H. A. D. Nguyen, J. Yu, M. A. Lebdeh, M. Taouil, S. Hamdioui, and F. Catthoor. "A classification of memory-centric computing". In: *ACM Journal on Emerging Technologies in Computing Systems (JETC)* 16.2 (2020), pp. 1–26.

[76] H. Fujiwara, H. Mori, W.-C. Zhao, M.-C. Chuang, R. Naous, C.-K. Chuang, T. Hashizume, D. Sun, C.-F. Lee, K. Akarvardar, *et al.* "A 5-nm 254-TOPS/W 221-TOPS/mm 2 fully-digital computing-in-memory macro supporting wide-range dynamic-voltage-frequency scaling and simultaneous MAC and write operations". In: *2022 IEEE International Solid-State Circuits Conference (ISSCC)*. Vol. 65. IEEE. 2022, pp. 1–3.

[77] H. Mori, W.-C. Zhao, C.-E. Lee, C.-F. Lee, Y.-H. Hsu, C.-K. Chuang, T. Hashizume, H.-C. Tung, Y.-Y. Liu, S.-R. Wu, *et al.* "A 4nm 6163-TOPS/W/b 4790-TOPS/mm^2 SRAM Based Digital-Computing-in-Memory Macro Supporting Bit-Width Flexibility and Simultaneous MAC and Weight Update". In: *2023 IEEE International Solid-State Circuits Conference (ISSCC)*. IEEE. 2023, pp. 132–134.

[78] J. Chang, Y.-H. Chen, H. Cheng, W.-M. Chan, H.-J. Liao, Q. Li, S. Chang, S. Natarajan, R. Lee, P.-W. Wang, *et al.* "A 20nm 112Mb SRAM in High-K metal-gate with assist circuitry for low-leakage and low-V MIN applications". In: *ISSCC*. 2013.

[79] G. Snider. "Computing with hysteretic resistor crossbars". In: *Applied Physics A* 80.6 (2005), pp. 1165–1172.

[80] S. Kvatinsky, D. Belousov, S. Liman, G. Satat, N. Wald, E. G. Friedman, A. Kolodny, and U. C. Weiser. "MAGIC—Memristor-aided logic". In: *IEEE Transactions on Circuits and Systems II: Express Briefs* 61.11 (2014), pp. 895–899.

[81] S. Kvatinsky, G. Satat, N. Wald, E. G. Friedman, A. Kolodny, and U. C. Weiser. "Memristor-based material implication (IMPLY) logic: Design principles and methodologies". In: *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 22.10 (2014), pp. 2054–2066.

[82] L. Xie, H. A. Du Nguyen, M. Taouil, S. Hamdioui, and K. Bertels. "Fast boolean logic mapped on memristor crossbar". In: *2015 33rd IEEE International Conference on Computer Design (ICCD)*. IEEE. 2015, pp. 335–342.

[83] P.-E. Gaillardon, L. Amarú, A. Siemon, E. Linn, R. Waser, A. Chattopadhyay, and G. De Micheli. "The programmable logic-in-memory (PLiM) computer". In: *2016 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. Ieee. 2016, pp. 427–432.

[84] L. Xie, H. A. Du Nguyen, J. Yu, A. Kaichouhi, M. Taouil, M. AlFailakawi, and S. Hamdioui. “Scouting logic: A novel memristor-based logic design for resistive computing”. In: *2017 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*. IEEE. 2017, pp. 176–181.

[85] S. Jain, A. Ranjan, K. Roy, and A. Raghunathan. “Computing in memory with spin-transfer torque magnetic ram”. In: *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 26.3 (2018), pp. 470–483.

[86] N. Xu *et al.* “STT-MRAM design technology co-optimization for hardware neural networks”. In: *IEDM*. 2018.

[87] R. Khaddam-Aljameh, P.-A. Francese, L. Benini, and E. Eleftheriou. “An SRAM-Based Multibit In-Memory Matrix-Vector Multiplier With a Precision That Scales Linearly in Area, Time, and Power”. In: *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* (2020).

[88] Z. Jiang, S. Yin, J.-S. Seo, and M. Seok. “C3SRAM: An in-memory-computing SRAM macro based on robust capacitive coupling computing mechanism”. In: *IEEE Journal of Solid-State Circuits* 55.7 (2020), pp. 1888–1897.

[89] J. Lee, H. Valavi, Y. Tang, and N. Verma. “Fully row/column-parallel in-memory computing SRAM macro employing capacitor-based mixed-signal computation with 5-b inputs”. In: *2021 Symposium on VLSI Circuits*. IEEE. 2021, pp. 1–2.

[90] V. Garcia and M. Bibes. “Ferroelectric tunnel junctions for information storage and processing”. In: *Nature communications* 5.1 (2014), pp. 1–12.

[91] S. Slesazeck, T. Ravsher, V. Havel, E. T. Breyer, H. Mulaosmanovic, and T. Mikolajick. “A 2TnC ferroelectric memory gain cell suitable for compute-in-memory and neuromorphic application”. In: *2019 IEEE International Electron Devices Meeting (IEDM)*. IEEE. 2019, pp. 38–6.

[92] T. Cao, Z. Zhang, W. L. Goh, C. Liu, Y. Zhu, and Y. Gao. “A Ternary Weight Mapping and Charge-mode Readout Scheme for Energy Efficient FeRAM Crossbar Compute-in-Memory System”. In: *2023 IEEE 5th International Conference on Artificial Intelligence Circuits and Systems (AICAS)*. IEEE. 2023, pp. 1–5.

[93] X. Peng, W. Chakraborty, A. Kaul, W. Shim, M. S. Bakir, S. Datta, and S. Yu. “Benchmarking monolithic 3D integration for compute-in-memory accelerators: overcoming ADC bottlenecks and maintaining scalability to 7nm or beyond”. In: *2020 IEEE International Electron Devices Meeting (IEDM)*. IEEE. 2020, pp. 30–4.

[94] D. B. Strukov, G. S. Snider, D. R. Stewart, and R. S. Williams. “The missing memristor found”. In: *nature* 453.7191 (2008), p. 80.

[95] L. Chua. “Memristor-the missing circuit element”. In: *IEEE Transactions on circuit theory* 18.5 (1971), pp. 507–519.

[96] X. Bi, C. Zhang, H. Li, Y. Chen, and R. E. Pino. “Spintronic memristor based temperature sensor design with CMOS current reference”. In: *2012 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE. 2012, pp. 1301–1306.

[97] J.-M. Lee, C.-M. Lee, L.-X. Ye, J.-P. Su, and T.-H. Wu. "Switching properties for mgo-based magnetic tunnel junction devices driven by spin-transfer torque in the nanosecond regime". In: *IEEE Transactions on magnetics* 47.3 (2011), pp. 629–632.

[98] B. K. You, M. Byun, S. Kim, and K. J. Lee. "Self-structured conductive filament nanoheater for chalcogenide phase transition". In: *ACS nano* 9.6 (2015), pp. 6587–6594.

[99] F. Oboril, R. Bishnoi, M. Ebrahimi, and M. B. Tahoori. "Evaluation of hybrid memory technologies using SOT-MRAM for on-chip cache hierarchy". In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 34.3 (2015), pp. 367–380.

[100] S. Salahuddin, K. Ni, and S. Datta. "The era of hyper-scaling in electronics". In: *Nature Electronics* 1.8 (2018), pp. 442–450.

[101] M. Fieback. "Testing RRAM and Computation-in-Memory Devices: Defects, Fault Models, and Test Solutions". In: (2022).

[102] J. S. Meena, S. M. Sze, U. Chand, and T.-Y. Tseng. "Overview of emerging nonvolatile memory technologies". In: *Nanoscale research letters* 9 (2014), pp. 1–33.

[103] I. Giannopoulos, A. Sebastian, M. Le Gallo, V. Jonnalagadda, M. Sousa, M. Boon, and E. Eleftheriou. "8-bit precision in-memory multiplication with projected phase-change memory". In: *2018 IEEE International Electron Devices Meeting (IEDM)*. IEEE. 2018, pp. 27–7.

[104] J. Borghetti, G. S. Snider, P. J. Kuekes, J. J. Yang, D. R. Stewart, and R. S. Williams. "'Memristive'switches enable 'stateful'logic operations via material implication". In: *Nature* 464.7290 (2010), p. 873.

[105] S. Kvatinsky, D. Belousov, S. Liman, G. Satat, N. Wald, E. G. Friedman, A. Kolodny, and U. C. Weiser. "MAGIC—Memristor-aided logic". In: *IEEE Transactions on Circuits and Systems II: Express Briefs* 61.11 (2014), pp. 895–899.

[106] L. Xie *et al.* "Scouting logic: A novel memristor-based logic design for resistive computing". In: *2017 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*. IEEE. 2017, pp. 176–181.

[107] S. Li, C. Xu, Q. Zou, J. Zhao, Y. Lu, and Y. Xie. "Pinatubo: A processing-in-memory architecture for bulk bitwise operations in emerging non-volatile memories". In: *Proceedings of the 53rd Annual Design Automation Conference*. ACM. 2016, p. 173.

[108] S. Jain *et al.* "Computing in memory with spin-transfer torque magnetic RAM". In: *TVLSI Systems* (2017).

[109] M. Imani, Y. Kim, and T. Rosing. "MPIM: Multi-purpose in-memory processing using configurable resistive memory". In: *ASP-DAC*. 2017.

[110] M. Lee, W. Tang, B. Xue, J. Wu, M. Ma, Y. Wang, Y. Liu, D. Fan, V. Narayanan, H. Yang, *et al.* "FeFET-based low-power bitwise logic-in-memory with direct write-back and data-adaptive dynamic sensing interface". In: *International Symposium on Low Power Electronics*. 2020.

[111] M.-F. Chang, P.-F. Chiu, and S.-S. Sheu. “Circuit design challenges in embedded memory and resistive RAM (RRAM) for mobile SoC and 3D-IC”. In: *16th ASP-DAC*. IEEE. 2011, pp. 197–203.

[112] W. Kim, A. Chattopadhyay, A. Siemon, E. Linn, R. Waser, and V. Rana. “Multistate memristive tantalum oxide devices for ternary arithmetic”. In: *Scientific reports, 2016* ().

[113] H. Koike *et al.* “1T1MTJ STT-MRAM cell array design with an adaptive reference voltage generator for improving device variation tolerance”. In: *IMW*. 2015.

[114] Y. Xie, X. Xue, J. Yang, Y. Lin, Q. Zou, R. Huang, and J. Wu. “A logic resistive memory chip for embedded key storage with physical security”. In: *TCAS II: Express Briefs, 2015* ().

[115] D. Ly, J. Noel, B. Giraud, P. Royer, E. Esmanhotto, N. Castellani, T. Dalgaty, J.-F. Nodin, C. Fenouillet-Beranger, E. Nowak, *et al.* “Novel 1T2R1T RRAM-based Ternary Content Addressable Memory for Large Scale Pattern Recognition”. In: *IEDM, 2019*.

[116] X. Si, Y.-N. Tu, W.-H. Huang, J.-W. Su, P.-J. Lu, J.-H. Wang, T.-W. Liu, S.-Y. Wu, R. Liu, Y.-C. Chou, *et al.* “15.5 A 28nm 64Kb 6T SRAM computing-in-memory macro with 8b MAC operation for AI edge chips”. In: *International Solid-State Circuits Conference-(ISSCC)*. 2020, pp. 246–248.

[117] M. Caselli, D. Bhattacharjee, A. Mallik, P. Debacker, and D. Verkest. “Tiny ci-SAR A/D Converter for Deep Neural Networks in Analog in-Memory Computation”. In: *2022 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE. 2022, pp. 1823–1827.

[118] Z. Chen, Z. Yu, Q. Jin, Y. He, J. Wang, S. Lin, D. Li, Y. Wang, and K. Yang. “CAP-RAM: A charge-domain in-memory computing 6T-SRAM for accurate and precision-programmable CNN inference”. In: *IEEE Journal of Solid-State Circuits* 56.6 (2021), pp. 1924–1935.

[119] A. Biswas, H. Sanghvi, M. Mehendale, and G. Preet. “An area-efficient 6T-SRAM based compute-in-memory architecture with reconfigurable SAR ADCs for energy-efficient deep neural networks in edge ML applications”. In: *2022 IEEE Custom Integrated Circuits Conference (CICC)*. IEEE. 2022, pp. 1–2.

[120] M. E. Sinangil, B. Erbagci, R. Naous, K. Akarvardar, D. Sun, W.-S. Khwa, H.-J. Liao, Y. Wang, and J. Chang. “A 7-nm compute-in-memory SRAM macro supporting multi-bit input, weight and output and achieving 351 TOPS/W and 372.4 GOPS”. In: *Journal of Solid-State Circuits* 56.1 (2020), pp. 188–198.

[121] S. Yin *et al.* “High-Throughput In-Memory Computing for Binary Deep Neural Networks With Monolithically Integrated RRAM and 90-nm CMOS”. In: *IEEE Transactions on Electron Devices* 67 (2020), pp. 4185–4192.

[122] J.-H. Yoon, M. Chang, W.-S. Khwa, Y.-D. Chih, M.-F. Chang, and A. Raychowdhury. “A 40-nm, 64-Kb, 56.67 TOPS/W voltage-sensing computing-in-memory/digital RRAM macro supporting iterative write with verification and online read-disturb detection”. In: *IEEE Journal of Solid-State Circuits* 57.1 (2021), pp. 68–79.

[123] C.-X. Xue, J.-M. Hung, H.-Y. Kao, Y.-H. Huang, S.-P. Huang, F.-C. Chang, P. Chen, T.-W. Liu, C.-J. Jhang, C.-I. Su, *et al.* "16.1 A 22nm 4Mb 8b-precision ReRAM computing-in-memory macro with 11.91 to 195.7 TOPS/W for tiny AI edge devices". In: *International Solid-State Circuits Conference-(ISSCC)*. Vol. 64. 2021, pp. 245–247.

[124] J.-W. Su, Y.-C. Chou, R. Liu, T.-W. Liu, P.-J. Lu, P.-C. Wu, Y.-L. Chung, L.-Y. Hung, J.-S. Ren, T. Pan, *et al.* "16.3 A 28nm 384kb 6T-SRAM computation-in-memory macro with 8b precision for AI edge chips". In: *2021 IEEE International Solid-State Circuits Conference (ISSCC)*. Vol. 64. IEEE. 2021, pp. 250–252.

[125] C. Liu, B. Yan, C. Yang, L. Song, Z. Li, B. Liu, Y. Chen, H. Li, Q. Wu, and H. Jiang. "A spiking neuromorphic design with resistive crossbar". In: *2015 52nd ACM/EDAC/IEEE Design Automation Conference (DAC)*. IEEE. 2015, pp. 1–6.

[126] C. Liu, Q. Yang, B. Yan, J. Yang, X. Du, W. Zhu, H. Jiang, Q. Wu, M. Barnell, and H. Li. "A memristor crossbar based computing engine optimized for high speed and accuracy". In: *2016 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*. IEEE. 2016, pp. 110–115.

[127] W.-H. Chen, K.-X. Li, W.-Y. Lin, K.-H. Hsu, P.-Y. Li, C.-H. Yang, C.-X. Xue, E.-Y. Yang, Y.-K. Chen, Y.-S. Chang, *et al.* "A 65nm 1Mb nonvolatile computing-in-memory ReRAM macro with sub-16ns multiply-and-accumulate for binary DNN AI edge processors". In: *International Solid-State Circuits Conference-(ISSCC)*. 2018, pp. 494–496.

[128] C.-X. Xue, T.-Y. Huang, J.-S. Liu, T.-W. Chang, H.-Y. Kao, J.-H. Wang, T.-W. Liu, S.-Y. Wei, S.-P. Huang, W.-C. Wei, *et al.* "15.4 A 22nm 2Mb ReRAM compute-in-memory macro with 121-28TOPS/W for multibit MAC computing for tiny AI edge devices". In: *2020 IEEE International Solid-State Circuits Conference-(ISSCC)*. IEEE. 2020, pp. 244–246.

[129] A. Siemon, T. Breuer, N. Aslam, S. Ferch, W. Kim, J. van den Hurk, V. Rana, S. Hoffmann-Eifert, R. Waser, S. Menzel, *et al.* "Realization of Boolean Logic Functionality Using Redox-Based Memristive Devices". In: *Advanced functional materials* 25.40 (2015), pp. 6414–6423.

[130] N. Talati, S. Gupta, P. Mane, and S. Kvatinsky. "Logic design within memristive memories using memristor-aided loGIC (MAGIC)". In: *IEEE Transactions on Nanotechnology* 15.4 (2016), pp. 635–650.

[131] L. Cheng, Y. Li, K.-S. Yin, S.-Y. Hu, Y.-T. Su, M.-M. Jin, Z.-R. Wang, T.-C. Chang, and X.-S. Miao. "Functional Demonstration of a Memristive Arithmetic Logic Unit (MemALU) for In-Memory Computing". In: *Advanced Functional Materials* (2019).

[132] M. Hu, C. E. Graves, C. Li, Y. Li, N. Ge, E. Montgomery, N. Davila, H. Jiang, R. S. Williams, J. J. Yang, *et al.* "Memristor-based analog computation and neural network classification with a dot product engine". In: *Advanced Materials* 30.9 (2018), p. 1705914.

[133] V. Joshi, M. Le Gallo, S. Haefeli, I. Boybat, S. Nandakumar, C. Piveteau, M. Dazzi, B. Rajendran, A. Sebastian, and E. Eleftheriou. “Accurate deep neural network inference using computational phase-change memory”. In: *Nature Communications* 11 (2020), p. 2473.

[134] P. M. Sheridan, F. Cai, C. Du, W. Ma, Z. Zhang, and W. D. Lu. “Sparse coding with memristor networks”. In: *Nature Nanotechnology* 12.8 (2017), p. 784.

[135] M. Le Gallo, A. Sebastian, R. Mathis, M. Manica, H. Giefers, T. Tuma, C. Bekas, A. Curioni, and E. Eleftheriou. “Mixed-precision in-memory computing”. In: *Nature Electronics* 1.4 (2018), p. 246.

[136] G. Karunaratne, M. Le Gallo, G. Cherubini, L. Benini, A. Rahimi, and A. Sebastian. “In-memory hyperdimensional computing”. In: *Nature Electronics* (2020). https://doi.org/10.1038/s41928-020-0410-3.

[137] N. Potti and J. M. Patel. “DAQ: A New Paradigm for Approximate Query Processing”. In: *Proc. VLDB Endow.* 8.9 (May 2015), pp. 898–909. ISSN: 2150-8097. DOI: 10.14778/2777598.2777599.

[138] P. Bakkum and K. Skadron. “Accelerating SQL Database Operations on a GPU with CUDA”. In: *Proceedings of the 3rd Workshop on General-Purpose Computation on Graphics Processing Units*. GPGPU-3. Pittsburgh, Pennsylvania, USA: Association for Computing Machinery, 2010, pp. 94–103. ISBN: 9781605589350. DOI: 10.1145/1735688.1735706. URL: https://doi.org/10.1145/1735688.1735706.

[139] D. Patterson, T. Anderson, N. Cardwell, R. Fromm, K. Keeton, C. Kozyrakis, R. Thomas, and K. Yelick. “A case for intelligent RAM”. In: *IEEE Micro* 17.2 (1997), pp. 34–44.

[140] M. Imani, S. Gupta, S. Sharma, and T. S. Rosing. “NVQuery: Efficient Query Processing in Nonvolatile Memory”. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 38.4 (2019), pp. 628–639.

[141] F. Xiong, A. D. Liao, D. Estrada, and E. Pop. “Low-power switching of phase-change materials with carbon nanotube electrodes”. In: *Science* 332.6029 (2011), pp. 568–570.

[142] K.-S. Li, C. Ho, M.-T. Lee, M.-C. Chen, C.-L. Hsu, J. Lu, C. Lin, C. Chen, B. Wu, Y. Hou, *et al.* “Utilizing sub-5 nm sidewall electrode technology for atomic-scale resistive memory fabrication”. In: *Proceedings of the Symposium on VLSI Technology*. IEEE. 2014, pp. 1–2.

[143] J. J. Yang, D. B. Strukov, and D. R. Stewart. “Memristive devices for computing”. In: *Nature Nanotechnology* 8.1 (2013), p. 13.

[144] I. Vourkas and G. C. Sirakoulis. “Emerging memristor-based logic circuit design approaches: A review”. In: *IEEE Circuits and Systems Magazine* 16.3 (2016), pp. 15–30.

[145] J. Borghetti, G. S. Snider, P. J. Kuekes, J. J. Yang, D. R. Stewart, and R. S. Williams. “Memristive switches enable stateful logic operations via material implication”. In: *Nature* 464.7290 (2010), p. 873.

[146] E. Linn, R. Rosezin, S. Tappertzhofen, U. Böttger, and R. Waser. "Beyond von Neumann: Logic operations in passive crossbar arrays alongside memory operations". In: *Nanotechnology* 23.30 (2012), p. 305205.

[147] D. S. Jeong, K. M. Kim, S. Kim, B. J. Choi, and C. S. Hwang. "Memristors for energy-efficient new computing paradigms". In: *Advanced Electronic Materials* 2.9 (2016), p. 1600090.

[148] E. Lehtonen, J. H. Poikonen, and M. Laiho. "Applications and limitations of memristive implication logic". In: *2012 13th International Workshop on Cellular Nanoscale Networks and their Applications*. 2012, pp. 1–6.

[149] Y. Sun, Y. Wang, and H. Yang. "Energy-efficient SQL query exploiting RRAM-based process-in-memory structure". In: *2017 IEEE 6th Non-Volatile Memory Systems and Applications Symposium (NVMSA)*. 2017, pp. 1–6.

[150] S. Li, C. Xu, Q. Zou, J. Zhao, Y. Lu, and Y. Xie. "Pinatubo: A processing-in-memory architecture for bulk bitwise operations in emerging non-volatile memories". In: *Proceedings of the Design Automation Conference (DAC)*. ACM. 2016, p. 173.

[151] L. Xie, H. A. Du Nguyen, J. Yu, A. Kaichouhi, M. Taouil, M. AlFailakawi, and S. Hamdioui. "Scouting logic: A novel memristor-based logic design for resistive computing". In: *Proceedings of the Computer Society Annual Symposium on VLSI (ISVLSI)*. IEEE. 2017, pp. 176–181.

[152] S. Hamdioui, H. A. Du Nguyen, M. Taouil, A. Sebastian, M. Le Gallo, S. Pande, S. Schaafsma, F. Catthoor, S. Das, F. G. Redondo, *et al.* "Applications of computation-in-memory architectures based on memristive devices". In: *Proceedings of the Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE. 2019, pp. 486–491.

[153] Y. Liu, Z. Wang, A. Lee, F. Su, C. Lo, Z. Yuan, C. Lin, Q. Wei, Y. Wang, Y. King, C. Lin, P. Khalili, K. Wang, M. Chang, and H. Yang. "4.7 A 65nm ReRAM-enabled non-volatile processor with 6× reduction in restore time and 4× higher clock frequency using adaptive data retention and self-write-termination nonvolatile logic". In: *2016 IEEE International Solid-State Circuits Conference (ISSCC)*. 2016, pp. 84–86.

[154] H. Li, T. F. Wu, A. Rahimi, K. Li, M. Rusch, C. Lin, J. Hsu, M. M. Sabry, S. B. Eryilmaz, J. Sohn, W. Chiu, M. Chen, T. Wu, J. Shieh, W. Yeh, J. M. Rabaey, S. Mitra, and H. P. Wong. "Hyperdimensional computing with 3D VRRAM in-memory kernels: Device-architecture co-design for energy-efficient, error-resilient language recognition". In: *2016 IEEE International Electron Devices Meeting (IEDM)*. 2016, pp. 16.1.1–16.1.4.

[155] W.-H. Chen, W.-J. Lin, L.-Y. Lai, S. Li, C.-H. Hsu, H.-T. Lin, H.-Y. Lee, J.-W. Su, Y. Xie, S.-S. Sheu, *et al.* "A 16Mb dual-mode ReRAM macro with sub-14ns computing-in-memory and memory functions enabled by self-write termination scheme". In: *Proceedings of the International Electron Devices Meeting (IEDM)*. IEEE. 2017, pp. 28–2.

[156] G. W. Burr, R. S. Shenoy, K. Virwani, P. Narayanan, A. Padilla, B. Kurdi, and H. Hwang. "Access devices for 3D crosspoint memory". In: *Journal of Vacuum Science & Technology B, Nanotechnology and Microelectronics: Materials, Processing, Measurement, and Phenomena* 32.4 (2014), p. 040802.

[157] J. Y. Seok, S. J. Song, J. H. Yoon, K. J. Yoon, T. H. Park, D. E. Kwon, H. Lim, G. H. Kim, D. S. Jeong, and C. S. Hwang. "A review of three-dimensional resistive switching cross-bar array memories from the integration and materials property points of view". In: *Advanced Functional Materials* 24.34 (2014), pp. 5316–5339.

[158] S. Slesazeck and T. Mikolajick. "Nanoscale resistive switching memory devices: a review". In: *Nanotechnology* 30.35 (2019), p. 352003.

[159] P. O. Vontobel, W. Robinett, P. J. Kuekes, D. R. Stewart, J. Straznicky, and R. S. Williams. "Writing to and reading from a nano-scale crossbar memory based on memristors". In: *Nanotechnology* 20.42 (2009), p. 425204.

[160] J. Liang and H.-S. P. Wong. "Cross-point memory array without cell selectors: Device characteristics and data storage pattern dependencies". In: *IEEE Transactions on Electron Devices* 57.10 (2010), pp. 2531–2538.

[161] M. A. Zidan, H. Omran, R. Naous, A. Sultan, H. A. H. Fahmy, W. D. Lu, and K. N. Salama. "Single-Readout High-Density Memristor Crossbar". In: *Scientific Reports* 6.1 (2016), p. 18863. ISSN: 2045-2322.

[162] A. Dozortsev, I. Goldshtein, and S. Kvatinsky. "Analysis of the row grounding technique in a memristor-based crossbar array". In: *International Journal of Circuit Theory and Applications* 46.1 (2018), pp. 122–137.

[163] I. Giannopoulos, A. Sebastian, M. Le Gallo, V. Jonnalagadda, M. Sousa, M. Boon, and E. Eleftheriou. "8-bit precision in-memory multiplication with projected phase-change memory". In: *Proceedings of the International Electron Devices Meeting (IEDM)*. IEEE. 2018, pp. 27–7.

[164] J. O. Irwin. "On the frequency distribution of the means of samples from a population having any law of frequency with finite moments, with special reference to Pearson's type II". In: *Biometrika* 19.3-4 (Dec. 1927), pp. 225–239. DOI: 10.1093/biomet/19.3-4.225.

[165] J. Marengo, D. Farnsworth, and L. Stefanic. "A Geometric Derivation of the Irwin-Hall Distribution". In: *International Journal of Mathematics and Mathematical Sciences* 2017 (Sept. 2017), pp. 1–6. DOI: 10.1155/2017/3571419.

[166] M. Le Gallo, D. Krebs, F. Zipoli, M. Salinga, and A. Sebastian. "Collective Structural Relaxation in Phase-Change Memory Devices". In: *Advanced Electronic Materials* 4.9 (2018), p. 1700627. DOI: 10.1002/aelm.201700627.

[167] D. Dua and C. Graff. *UCI Machine Learning Repository*. 2017. URL: http://archive.ics.uci.edu/ml.

[168] V. Seshadri, K. Hsieh, A. Boroum, D. Lee, M. A. Kozuch, O. Mutlu, P. B. Gibbons, and T. C. Mowry. "Fast Bulk Bitwise AND and OR in DRAM". In: *IEEE Computer Architecture Letters* 14.2 (2015), pp. 127–131. DOI: 10.1109/LCA.2015.2434872.

[169] N. Chandoke, N. Chitkara, and A. Grover. “Comparative analysis of sense amplifiers for SRAM in 65nm CMOS technology”. In: *Proceedings of the International Conference on Electrical, Computer and Communication Technologies (ICECCT)*. IEEE. 2015, pp. 1–7.

[170] M. Komalan *et al.* “Cross-layer design and analysis of a low power, high density STT-MRAM for embedded systems”. In: *ISCAS*. 2017.

[171] Y. Pan *et al.* “A multilevel cell STT-MRAM-based computing in-memory accelerator for binary convolutional neural network”. In: *Transactions on Magnetics* (2018).

[172] L. Zhang *et al.* “A high-reliability and low-power computing-in-memory implementation within STT-MRAM”. In: *Microelect.* (2018).

[173] S. Resch *et al.* “Pimball: Binary neural networks in spintronic memory”. In: *ACM TACO* (2019).

[174] R. Bishnoi, M. Ebrahimi, F. Oboril, and M. B. Tahoori. “Read disturb fault detection in STT-MRAM”. In: *ITC*. 2014.

[175] S. Rao *et al.* “STT-MRAM array performance improvement through optimization of Ion Beam Etch and MTJ for Last-Level Cache application”. In: *IMW*. 2021.

[176] M. A. Lebdeh *et al.* “Memristive device based circuits for computation-in-memory architectures”. In: *ISCAS*. 2019.

[177] P. C. Santos, G. F. Oliveira, D. G. Tomé, M. A. Alves, E. C. Almeida, and L. Carro. “Operand size reconfiguration for big data processing in memory”. In: *DATE, 2017*.

[178] M. Bocquet, T. Hirztlin, J.-O. Klein, E. Nowak, E. Vianello, J.-M. Portal, and D. Querlioz. “In-memory and error-immune differential RRAM implementation of binarized deep neural networks”. In: *IEDM*. 2018.

[179] L. Wang, W. Ye, C. Dou, X. Si, X. Xu, J. Liu, D. Shang, J. Gao, F. Zhang, Y. Liu, *et al.* “Efficient and Robust Nonvolatile Computing-In-Memory Based on Voltage Division in 2T2R RRAM With Input-Dependent Sensing Control”. In: *TCAS II: Express Briefs, 2021* ().

[180] J. Wang, L. Pan, B. Gao, D. Wu, J. Tang, H. Wu, and H. Qian. “A Novel Page-Forming Scheme with Ultra-Low Bit-Error-Rate and High Reliability on a 1Mb RRAM Chip”. In: *ICSICT, 2020*.

[181] Z. Zhou, P. Huang, Y. Xiang, W. Shen, Y. Zhao, Y. Feng, B. Gao, H. Wu, H. Qian, L. Liu, *et al.* “A new hardware implementation approach of BNNs based on nonlinear 2T2R synaptic cell”. In: *IEDM, 2018*.

[182] J. Borghetti, G. S. Snider, P. J. Kuekes, J. J. Yang, D. R. Stewart, and R. S. Williams. “Memristive switches enable ‘stateful’ logic operations via material implication”. In: *Nature, 2010* ().

[183] S. Kvatinsky *et al.* “MAGIC—Memristor-aided logic”. In: *IEEE Transactions on Circuits and Systems II: Express Briefs* 61.11 (2014), pp. 895–899.

[184] S. Yu and P.-Y. Chen. “Emerging memory technologies: Recent trends and prospects”. In: *Solid-State Circuits Magazine* (2016).

[185] F. Cai, J. M. Correll, S. H. Lee, Y. Lim, V. Bothra, Z. Zhang, M. P. Flynn, and W. D. Lu. “A fully integrated reprogrammable memristor–CMOS system for efficient multiply–accumulate operations”. In: *Nature Electronics* 2.7 (2019), pp. 290–299.

[186] M. Fieback, G. C. Medeiros, A. Gebregiorgis, H. Aziza, M. Taouil, and S. Hamdioui. “Intermittent Undefined State Fault in RRAMs”. In: *ETS*. 2021.

[187] M. Fieback, M. Taouil, and S. Hamdioui. “Testing resistive memories: Where are we and what is missing?” In: *ITC*. 2018.

[188] X. Cui, M. Zhang, Q. Lin, X. Cui, and A. Pang. “Design and test of the in-array build-in self-test scheme for the embedded RRAM array”. In: *TED* (2019).

[189] S. Kannan, J. Rajendran, R. Karri, and O. Sinanoglu. “Sneak-path testing of crossbar-based nonvolatile random access memories”. In: *IEEE Trans. on Nanotechnology* (2013).

[190] C.-Y. Chen, H.-C. Shih, C.-W. Wu, C.-H. Lin, P.-F. Chiu, S.-S. Sheu, and F. T. Chen. “RRAM defect modeling and failure analysis based on march test and a novel squeeze-search scheme”. In: *TC* (2014).

[191] Y.-X. Chen and J.-F. Li. “Fault modeling and testing of 1T1R memristor memories”. In: *VTS*. 2015.

[192] P. Liu, Z. You, J. Kuang, Z. Hu, H. Duan, and W. Wang. “Efficient March test algorithm for 1T1R cross-bar with complete fault coverage”. In: *Elect. Letters* (2016).

[193] Y. Luo, X. Cui, M. Luo, and Q. Lin. “A high fault coverage march test for 1T1R memristor array”. In: *EDSSC*. 2017.

[194] P. Liu, J. Wu, Z. You, M. Elimu, W. Wang, and S. Cai. “Defect analysis and parallel march test algorithm for 3D hybrid CMOS-memristor memory”. In: *ATS*. 2018.

[195] T. Copetti, T. Gemmeke, and L. B. Poehls. “Validating a DFT Strategy’s Detection Capability regarding Emerging Faults in RRAMs”. In: *VLSI-SoC*. 2021.

[196] V. A. Hongal, R. Kotikalapudi, Y.-B. Kim, and M. Choi. “A novel “divide and conquer” testing technique for memristor based lookup table”. In: *MWSCAS*. 2011.

[197] P. Liu, Z. You, J. Kuang, Z. Hu, and W. Wang. “Logic operation-based DFT method and 1R memristive crossbar march-like test algorithm”. In: *IEICE* (2015).

[198] P. Liu, Z. You, J. Kuang, M. Elimu, S. Cai, and W. Wang. “Logic operation-based Design for Testability method and parallel test algorithm for 1T1R crossbar”. In: *Elec. Letters* (2017).

[199] T. Li, X. Bi, N. Jing, X. Liang, and L. Jiang. “Sneak-path based test and diagnosis for 1R RRAM crossbar using voltage bias technique”. In: *DAC*. 2017.

[200] N. Z. Haron and S. Hamdioui. “DfT schemes for resistive open defects in RRAMs”. In: *DATE*. 2012.

[201] S. Hamdioui, M. Taouil, and N. Z. Haron. “Testing open defects in memristor-based memories”. In: *IEEE Trans. on Computers* (2013).

[202] S. N. Mozaffari, S. Tragoudas, and T. Haniotakis. “More efficient testing of metal-oxide memristor–based memory”. In: *TCAD* (2016).

[203] M. N. I. Khan and S. Ghosh. “Test challenges and solutions for emerging non-volatile memories”. In: *VTS*. 2018.

[204] P. Liu, Z. You, J. Wu, B. Liu, Y. Han, and K. Chakrabarty. “Fault modeling and efficient testing of memristor-based memory”. In: *TCASI* (2021).

[205] H.-S. P. Wong, H.-Y. Lee, S. Yu, Y.-S. Chen, Y. Wu, P.-S. Chen, B. Lee, F. T. Chen, and M.-J. Tsai. “Metal–oxide RRAM”. In: *Proceedings IEEE* (2012).

[206] J. Yang, X. Xue, X. Xu, Q. Wang, H. Jiang, J. Yu, D. Dong, F. Zhang, H. Lv, and M. Liu. “24.2 A 14nm-FinFET 1Mb Embedded 1T1R RRAM with a 0.022 μ m 2 Cell Size Using Self-Adaptive Delayed Termination and Multi-Cell Reference”. In: *ISSCC*. 2021.

[207] E. Esmanhotto, L. Brunet, N. Castellani, D. Bonnet, T. Dalgaty, L. Grenouillet, D. Ly, C. Cagli, C. Vizioz, N. Allouti, *et al.* “High-density 3D monolithically integrated multiple 1T1R multi-level-cell for neural networks”. In: *IEDM*. 2020.

[208] M. Fieback, G. C. Medeiros, L. Wu, H. Aziza, R. Bishnoi, M. Taouil, and S. Hamdioui. “Defects, Fault Modeling, and Test Development Framework for RRAMs”. In: *JETC* (2022).

[209] N. Z. Haron and S. Hamdioui. “On defect oriented testing for hybrid CMOS/memristor memory”. In: *Asian Test Symposium*. 2011.

[210] M. Fieback, L. Wu, G. C. Medeiros, H. Aziza, S. Rao, E. J. Marinissen, M. Taouil, and S. Hamdioui. “Device-aware test: A new test approach towards DPPB level”. In: *ITC*. 2019.

[211] M. R. Stan. “Synchronous up/down counter with clock period independent of counter size”. In: *ARITH*. 1997.

[212] R. Aitken, N. Dogra, D. Gandhi, and S. Becker. “Redundancy, repair, and test features of a 90nm embedded SRAM generator”. In: *DFT*. 2003.

[213] F. Cüppers, S. Menzel, C. Bengel, A. Hardtdegen, M. Von Witzleben, U. Böttger, R. Waser, and S. Hoffmann-Eifert. “Exploiting the switching dynamics of HfO2-based ReRAM devices for reliable analog memristive behavior”. In: *APL materials* (2019).

[214] A. J. Van De Goor. “Using march tests to test SRAMs”. In: *DTC* (1993).

[215] A. Razavieh, P. Zeitzoff, and E. J. Nowak. “Challenges and limitations of CMOS scaling for FinFET and beyond architectures”. In: *Trans. on Nanotechnology* (2019).

[216] E. J. Nowak, I. Aller, T. Ludwig, K. Kim, R. V. Joshi, C.-T. Chuang, K. Bernstein, and R. Puri. “Turning silicon on its edge [double gate CMOS/FinFET technology]”. In: *Circuits and Devices Magazine* (2004).

[217] Y. Wang, H. Yu, D. Sylvester, and P. Kong. “Energy efficient in-memory AES encryption based on nonvolatile domain-wall nanowire”. In: *DATE*. 2014.

[218] M. Ma, Y. Zhu, Z. Zhu, R. Yuan, J. Liu, L. Xu, Y. Yang, and Y. Wang. “Efficient In-Memory AES Encryption Implementation Using a General Memristive Logic: Surmounting the data movement bottleneck”. In: *Nanotechnology Magazine* (2022).

[219] J.-M. Hung, C.-J. Jhang, P.-C. Wu, Y.-C. Chiu, and M.-F. Chang. “Challenges and trends of nonvolatile in-memory-computation circuits for AI edge devices”. In: *Open JSSC Society* (2021).

[220] W.-H. Chen, W.-J. Lin, L.-Y. Lai, S. Li, C.-H. Hsu, H.-T. Lin, H.-Y. Lee, J.-W. Su, Y. Xie, S.-S. Sheu, *et al.* “A 16Mb dual-mode ReRAM macro with sub-14ns computing-in-memory and memory functions enabled by self-write termination scheme”. In: *IEDM*. 2017.

[221] R. Liu, X. Zhang, X. Chen, Y. Han, and M. Tang. “FeMIC: Multi-operands in-memory computing based on FeFETs”. In: *ASP-DAC*. 2022.

[222] J. Yu, H. A. Du Nguyen, M. A. Lebdeh, M. Taouil, and S. Hamdioui. “Enhanced scouting logic: A robust memristive logic design scheme”. In: *NANOARCH*. 2019.

[223] D. Reis, H. Geng, M. Niemier, and X. S. Hu. “IMCRYPTO: An in-memory computing fabric for AES encryption and decryption”. In: *TVLSI* (2022).

[224] M. Ezzadeen, A. Majumdar, M. Bocquet, B. Giraud, J.-P. Noël, F. Andrieu, D. Querlioz, and J.-M. Portal. “Low-overhead implementation of binarized neural networks employing robust 2T2R resistive RAM bridges”. In: *ESSCIRC*. 2021.

[225] B. Li, Y. Shan, M. Hu, Y. Wang, Y. Chen, and H. Yang. “Memristor-based approximated computation”. In: *International Symposium on Low Power Electronics and Design (ISLPED)*. IEEE. 2013, pp. 242–247.

[226] C. Liu, Q. Yang, C. Zhang, H. Jiang, Q. Wu, and H. H. Li. “A memristor-based neuromorphic engine with a current sensing scheme for artificial neural network applications”. In: *2017 22nd Asia and South Pacific Design Automation Conference (ASP-DAC)*. IEEE. 2017, pp. 647–652.

[227] M. A. Zidan *et al.* “Memristor based memory: The sneak paths problem and solutions”. In: *Microelectronics Journal* (2013).

[228] Y. Jeong *et al.* “Parasitic Effect Analysis in Memristor-Array-Based Neuromorphic Systems”. In: *IEEE Transactions on Nanotechnology* (2018).

[229] L. Kull, T. Toifl, M. Schmatz, P. A. Francese, C. Menolfi, M. Braendli, M. Kossel, T. Morf, T. M. Andersen, and Y. Leblebici. “A 3.1 mW 8b 1.2 GS/s single-channel asynchronous SAR ADC with alternate comparators for enhanced speed in 32 nm digital SOI CMOS”. In: *IEEE Journal of Solid-State Circuits* 48.12 (2013), pp. 3049–3058.

[230] A. Shafiee *et al.* “ISAAC: A convolutional Neural Network Accelerator with In-Situ Analog Arithmetic in Crossbars”. In: *ACM SIGARCH Computer Architecture News* 44.3 (2016), pp. 14–26.

[231] A. Shafiee, A. Nag, N. Muralimanohar, R. Balasubramonian, J. P. Strachan, M. Hu, R. S. Williams, and V. Srikumar. “ISAAC: A convolutional neural network accelerator with in-situ analog arithmetic in crossbars”. In: *ACM SIGARCH Computer Architecture News* 44.3 (2016), pp. 14–26.

[232] P. Chi, S. Li, C. Xu, T. Zhang, J. Zhao, Y. Liu, Y. Wang, and Y. Xie. "Prime: A novel processing-in-memory architecture for neural network computation in reram-based main memory". In: *ACM SIGARCH Computer Architecture News* 44.3 (2016), pp. 27–39.

[233] T. Chou *et al.* "Cascade: Connecting rrams to extend analog dataflow in an end-to-end in-memory processing paradigm". In: *Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture*. 2019, pp. 114–125.

[234] L. Song *et al.* "Pipelayer: A pipelined reram-based accelerator for deep learning". In: *2017 IEEE International Symposium on High Performance Computer Architecture (HPCA)*. IEEE. 2017, pp. 541–552.

[235] B. S. Amrutur and M. A. Horowitz. "A replica technique for wordline and sense control in low-power SRAM's". In: *IEEE Journal of solid-state circuits* 33.8 (1998), pp. 1208–1219.

[236] M.-F. Chang *et al.* "A high-speed 7.2-ns read-write random access 4-Mb embedded resistive RAM (ReRAM) macro using process-variation-tolerant current-mode read schemes". In: *IEEE Journal of Solid-State Circuits* 48.3 (2012), pp. 878–891.

[237] M.-F. Chang *et al.* "Challenges and circuit techniques for energy-efficient on-chip nonvolatile memory using memristive devices". In: *IEEE Journal on Emerging and Selected Topics in Circuits and Systems* 5.2 (2015), pp. 183–193.

[238] Q. Dong *et al.* "Rectified-linear and recurrent neural networks built with spin devices". In: *2017 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE. 2017, pp. 1–4.

[239] A. Hardtdegen, C. La Torre, F. Cüppers, S. Menzel, R. Waser, and S. Hoffmann-Eifert. "Improved switching stability and the effect of an internal series resistor in HfO 2/TiO x Bilayer ReRAM cells". In: *IEEE Transactions on Electron Devices* 65.8 (2018), pp. 3229–3236.

[240] Y. Chen, H. Lee, P. Chen, P. Gu, C. Chen, W. Lin, W. Liu, Y. Hsu, S. Sheu, P. Chiang, *et al.* "Highly scalable hafnium oxide memory with improvements of resistive distribution and read disturb immunity". In: *2009 IEEE International Electron Devices Meeting (IEDM)*. IEEE. 2009, pp. 1–4.

[241] P. Nuzzo, C. Nani, C. Armiento, A. Sangiovanni-Vincentelli, J. Craninckx, and G. Van der Plas. "A 6-bit 50-MS/s threshold configuring SAR ADC in 90-nm digital CMOS". In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 59.1 (2011), pp. 80–92.

[242] S. Hong, H. Kang, J. Kim, and K. Cho. "Low voltage time-based matrix multiplier-and-accumulator for neural computing system". In: *Electronics* 9.12 (2020), p. 2138.

[243] D. S. Kang and S. Yu. "Time-Based Compute-in-Memory for Cryogenic Neural Network With Successive Approximation Register Time-to-Digital Converter". In: *Journal on Exploratory Solid-State Computational Devices and Circuits* 8.2 (2022), pp. 128–133.

[244] A. Kneip and D. Bol. “Impact of analog non-idealities on the design space of 6T-SRAM current-domain dot-product operators for in-memory computing”. In: *Transactions on Circuits and Systems I: Regular Papers* 68.5 (2021), pp. 1931–1944.

[245] H.-H. Hsu, T.-H. Wen, W.-H. Huang, W.-S. Khwa, Y.-C. Lo, C.-J. Jhang, Y.-H. Chin, Y.-C. Chen, C.-C. Lo, R.-S. Liu, *et al.* “A Nonvolatile AI-Edge Processor With SLC–MLC Hybrid ReRAM Compute-in-Memory Macro Using Current–Voltage-Hybrid Readout Scheme”. In: *JSSC* (2023).

[246] M. Baert and W. Dehaene. “A 5-GS/s 7.2-ENOB time-interleaved VCO-based ADC achieving 30.5 fJ/cs”. In: *JSSC* (2019).

CURRICULUM VITÆ

Abhairaj SINGH

14.02.1991 Born in Chandigarh, India

EDUCATION

2019-2024	Pursuing PhD. degree in Computer Engineering *Delft University of Technology, the Netherlands*
Thesis:	Logic and Arithmetic Computation-in-Memory Accelerators
Promotors:	Prof. Said Hamdioui, Prof. Rajiv V. Joshi, Dr. Rajendra K. Bishnoi

2017-2019	M.Sc. degree in Electrical Engineering, Micro-electronics *Delft University of Technology, the Netherlands*
Thesis:	Circuit-design for Memristor-based In-memory Computing
Supervisors:	Prof. Said Hamdioui

2009-2013	B.Sc. degree in Electronics and Communication Engineering *Delhi Technological University, India*

WORK EXPERIENCE

2013-2017	Design Engineer *ARM, India*
Profile:	Circuit design engineer for SRAM, ROM compilers

List of Publications & Patents

Publications

13. **Abhairaj Singh**, Rajendra Bishnoi, Yashvarhan Biyani, Rajiv V. Joshi, Said Hamdioui, "A 206 TOPS/W Computation-in-Memory using a Novel Self-Biasing Scheme", *submitted* in ESSERC 2024

12. **Abhairaj Singh**, Rajendra Bishnoi, Rajiv V. Joshi, Said Hamdioui, "MOXOR-CIM: Energy-efficient Multi-operand XOR Logic based Computation-in-Memory Accelerator", *submitted* in DAC 2024

11. **Abhairaj Singh**, Rajendra Bishnoi, Ali Kaichouhi, Sumit Diware, Rajiv V. Joshi, Said Hamdioui, "A 115.1 TOPS/W, 12.1 TOPS/mm2 Computation-in-Memory using Ring-Oscillator based ADC for Edge AI", in AICAS 2023: 1-5

10. **Abhairaj Singh**, Moritz Fieback, Rajendra Bishnoi, Filip Bradaric, Anteneh Gebregiorgis, Rajiv V. Joshi, Said Hamdioui, "Accelerating RRAM Testing with a Low-cost Computation-in-Memory based DFT", in ITC 2022: 400-409

9. **Abhairaj Singh**, Rajendra Bishnoi, Rajiv V. Joshi, Said Hamdioui, "Referencing-in-Array Scheme for RRAM-based CIM Architecture", in DATE 2022: 1413-1418

8. **Abhairaj Singh**, Mahdi Zahedi, Taha Shahroodi, Mohit Gupta, Anteneh Gebregiorgis, Manu Komalan, Rajiv V. Joshi, Francky Catthoor, Rajendra Bishnoi, Said Hamdioui, "CIM-based Robust Logic Accelerator using 28 nm STT-MRAM Characterization Chip Tape-out", in AICAS 2022: 451-454

7. **Abhairaj Singh**, Sumit Diware, Anteneh Gebregiorgis, Rajendra Bishnoi, Francky Catthoor, Rajiv V. Joshi, Said Hamdioui, "Low-Power Memristor-Based Computing for Edge-AI Applications", in ISCAS 2021: 1-5

6. **Abhairaj Singh**, Muath Abu Lebdeh, Anteneh Gebregiorgis, Rajendra Bishnoi, Rajiv V. Joshi, Said Hamdioui, "SRIF: Scalable and Reliable Integrate and Fire Circuit ADC for Memristor-Based CIM Architectures", in IEEE Trans. Circuits Syst. I Regul. Pap. 68(5): 1917-1930 (2021)

5. Sumit Diware, **Abhairaj Singh**, Anteneh Gebregiorgis, Rajiv V. Joshi, Said Hamdioui, Rajendra Bishnoi, "Accurate and Energy-Efficient Bit-Slicing for RRAM-Based Neural Networks", in IEEE Trans. Emerg. Top. Comput. Intell. 7(1): 164-177 (2023)

4. Anteneh Gebregiorgis, **Abhairaj Singh**, Amir Yousefzadeh, Dirk Wouters, Rajendra Bishnoi, Francky Catthoor, Said Hamdioui, "Tutorial on memristor-based computing for smart edge applications", in Memories-Materials, Devices, Circuits and Systems, 4, 100025, 2023

3. Mahta Mayahinia, **Abhairaj Singh**, Christopher Bengel, Stefan Wiefels, Muath Abu Lebdeh, Stephan Menzel, Dirk J. Wouters, Anteneh Gebregiorgis, Rajendra Bishnoi, Rajiv V. Joshi,

Said Hamdioui, "A Voltage-Controlled, Oscillation-Based ADC Design for Computation-in-Memory Architectures Using Emerging ReRAMs", in ACM J. Emerg. Technol. Comput. Syst. 18(2): 32:1-32:25 (2022)

2. Anteneh Gebregiorgis, **Abhairaj Singh**, Sumit Diware, Rajendra Bishnoi, Said Hamdioui, "Dealing with Non-Idealities in Memristor Based Computation-In-Memory Designs", in VLSI-SoC 2022: 1-6

1. Iason Giannopoulos, **Abhairaj Singh**, Manuel Le Gallo, Vara Prasad Jonnalagadda, Said Hamdioui, Abu Sebastian, "In-Memory Database Query", in Adv. Intell. Syst. 2(12): 2000141 (2020)

PATENTS

2. **Abhairaj Singh**, Rajendra Bishnoi, Said Hamdioui, "Complementary Data Storage for Accurate and Efficient Computation-in-Memory Accelerators", *patent pending* 2022

1. Yashvarhan Biyani, **Abhairaj Singh**, Rajendra Bishnoi, Said Hamdioui, "Dradnats: An Ultra-low-power Memristor-based Computation-in-Memory architecture using a serial current driver approach", *patent pending* 2022