Testing RRAM and Computation-in-Memory Devices
Defects, Fault Models, and Test Solutions

Fieback, M.

DOI
10.4233/uuid:71e1cbf8-ce02-4ce8-a09b-883e6f84e996

Publication date
2022

Document Version
Final published version

Citation (APA)
Fieback, M. (2022). *Testing RRAM and Computation-in-Memory Devices: Defects, Fault Models, and Test Solutions*. [Dissertation (TU Delft), Delft University of Technology]. https://doi.org/10.4233/uuid:71e1cbf8-ce02-4ce8-a09b-883e6f84e996

TESTING RRAM AND COMPUTATION-IN-MEMORY DEVICES

DEFECTS, FAULT MODELS, AND TEST SOLUTIONS

TESTING RRAM AND COMPUTATION-IN-MEMORY DEVICES

DEFECTS, FAULT MODELS, AND TEST SOLUTIONS

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 vrijdag 8 juli 2022 om 15:00 uur

door

Moritz Christiaan Reiner FIEBACK

Master of Science in Electrical Engineering
Technische Universiteit Delft, Delft, Nederland,
geboren te Leiderdorp, Nederland.

Dit proefschrift is goedgekeurd door de

promotor: prof. dr. ir. S. Hamdioui
copromotor: dr. ir. M. Taouil

Samenstelling promotiecommissie:

Rector Magnificus,	voorzitter
Prof. dr. ir. S. Hamdioui,	Technische Universiteit Delft
Dr. ir. M. Taouil,	Technische Universiteit Delft
Onafhankelijke leden:	
Prof. dr. ir. F.P. Widdershoven,	Technische Universiteit Delft
Prof. dr. J.-F. Li,	National Central University, Taiwan
Dr. S.D. Cotofana,	Technische Universiteit Delft
Dr. H. Aziza,	Aix-Marseille Université, France
Prof. dr. ir. R.E. Kooij,	Technische Universiteit Delft, reservelid

Keywords: resistive RAM, RRAM, memory test, device-aware test, defect modeling, fault modeling, test development, computation-in-memory, CIM, reliability

Printed by: Ipskamp Printing, the Netherlands

Front & Back: “Buitenbeentje” by M.C.R. Fieback

ISBN 978-94-6384-356-0

CONTENTS

SUMMARY

Resistive random access memory (RRAM) is a promising emerging memory technology that offers dense, non-volatile memories that do not consume any static power. Furthermore, RRAM devices can be written and read out in nanoseconds, and it is possible to use them to perform computation-in-memory (CIM). These benefits make this technology a potential replacement for Flash or even dynamic random access memory (DRAM). This is also clearly seen by the community; both universities and companies are prototyping RRAMs, and there are already some commercial RRAMs available. In order to deliver high-quality products, the RRAMs need to be tested properly so that a manufacturer can guarantee the quality. This dissertation focuses on test development for RRAMs.

Traditionally, production defects in memories, such as DRAM, are modeled as linear resistors in or between two nodes of the circuit. In literature, many researchers have applied a similar approach for RRAMs. However, we demonstrate that this method of modeling defects is inappropriate, because the models fail to describe the defective behavior of the RRAM device. Instead, those models describe defects in the interconnections that surround the RRAM device. To overcome this, we propose the Device-Aware Test (DAT) approach that consists of three steps. First, the approach models the actual physics of defective devices and thus leads to realistic defect models. Second, the defect models are used to perform accurate fault modeling and analysis. Third, the results from this step are used to develop high-quality RRAM tests.

We do this by first characterizing the defect. We analyze the complete production process of a RRAM. During this analysis, we identify what can go wrong in every step, and in what kind of defects this may result. All identified defects need to be properly modeled, so that a high-quality test can be developed. Next, we analyze how it affects the performance of a defect-free device, and incorporate the resulting defective behavior in a compact defect model. This model is calibrated and accurately describes the effects of the defect. Second, we apply this defect model in a RRAM circuit to perform fault modeling and analysis. We systematically define the complete space of all faults that could occur, and then apply an analysis methodology to validate which faults actually occur in the circuit. Third, we develop a test for the validated faults. This test only needs to detect faults that are actually sensitized, and thus is shorter than generic tests, while it also has a better fault coverage.

We apply the DAT approach to RRAM forming defects and RRAM intermittent undefined state faults. The results show that these two defect models sensitize different faults than the traditional defect models do. Since the DAT defect models describe the actual physics of the defects, we can conclude that the traditional approach will lead to low-quality tests that generate test escapes and reduce the production yield. Furthermore, we demonstrate that the faults cannot easily be detected by existing test algorithms, and that special tests need to be developed to detect them.

We also apply the DAT approach to a RRAM-based computation-in-memory (CIM) architecture to develop a test for it. We shows that a CIM device needs to be tested both in its memory and computation configuration, as there are unique faults in both configurations. We define the complete fault space for CIM faults and validate it using the DAT approach. Subsequently, we develop a test that detects the faults in both configurations. Furthermore, we study how process, voltage and temperature variations affect the performance of the CIM architecture. We demonstrate that certain operations are more susceptible to these variations than other ones.

SAMENVATTING

Resistive Random Access Memory (RRAM) is een veelbelovende opkomende geheugentechnologie die dichte, niet-vluchtige geheugens biedt die geen statisch vermogen verbruiken. Bovendien kunnen RRAM-cellen in nanoseconden worden geschreven en uitgelezen en is het mogelijk om ze te gebruiken om gegevensverwerking-in-geheugenarchitecturen (GIG) uit te voeren. Deze voordelen maken deze technologie een potentiële vervanging voor Flash of zelfs Dynamic Random Access Memory (DRAM). Dit wordt ook duidelijk gezien door de gemeenschap; zowel universiteiten als bedrijven maken prototypes van RRAM's en er zijn al enkele commerciële RRAM's beschikbaar. Om producten van hoge kwaliteit te kunnen leveren, moeten de RRAM's goed worden getest zodat een fabrikant de kwaliteit kan garanderen. Dit proefschrift richt zich op testontwikkeling voor RRAM's.

In standaard geheugentechnologieën, zoals DRAM, worden productiefouten gemodelleerd als lineaire weerstanden tussen twee knooppunten in een circuit. In de wetenschappelijke literatuur wordt een gelijke aanpak gebruikt voor RRAM's. In dit werk laten wij echter zien dat deze manier van modellering niet afdoende is, omdat zulke modellen het foutieve gedrag van de RRAM-cel niet goed kunnen beschrijven; deze modellen beschrijven immers productiefouten in de verbindingen rondom de RRAM-cel. Om dit probleem op te lossen, introduceren wij de Componentbewuste Test (CBT)-benadering die uit drie stappen bestaat. In de eerste stap wordt de echte fysica van de productiefout gemodelleerd, zodat realistische productiefoutmodellen ontwikkeld kunnen worden. In de tweede stap worden deze modellen gebruikt om accuraat geheugenfouten te modelleren en te analyseren. In de derde stap gebruiken we de resultaten uit de tweede stap om tests van hoge kwaliteit te ontwikkelen voor RRAM's.

Eerst karakteriseren we de productiefout. We analyseren het complete productieproces van een RRAM om vast te stellen wat er in iedere fase fout kan gaan, en in wat voor productiefouten dit kan resulteren. Alle productiefouten die aan het licht komen moeten op de juiste wijze gemodelleerd worden, zodat we test van hoge kwaliteit kunnen ontwikkelen. Vervolgens bestuderen we hoe een productiefout de parameters van een foutenvrije RRAM-cel beïnvloedt en modelleren we dat gedrag in een compact productiefoutmodel. Dit model is gekalibreerd en beschrijft op nauwkeurige wijze de gevolgen van de productiefout. Daarna gebruiken we dit model om geheugenfouten te modelleren en te analyseren. We definiëren op een systematische manier de complete geheugenfoutenruimte van geheugenfouten die theoretisch zouden kunnen ontstaan. Vervolgens analyseren we geheugenfoutenruimte om te valideren welke fouten daadwerkelijk in het circuit op kunnen treden. Ten derde ontwikkelen we test enkel voor de gevalideerde geheugenfouten. Omdat deze test alleen echte fouten moet detecteren, zal deze test sneller zijn en een hogere foutdetectie hebben dan reguliere geheugentesten.

We passen de CBT-benadering toe op vormingsfouten en op periodieke ongedefinieerde toestandsfouten in RRAM's. Uit de resultaten blijkt dat deze twee productie-

fouten andere geheugenfouten veroorzaken dan de standaard productiefoutmodellen doen. Doordat de modellen die met de CBT-benadering ontwikkeld zijn de echte fysica van een productiefout beschrijven kunnen we concluderen dat de traditionele foutmodelleringsmethode enkel tot tests van lage kwaliteit kunnen leiden. Het gebruik van deze tests leidt vervolgens tot het niet-detecteren van foutieve RRAM's en het onnodig detecteren van correcte RRAM's. Daarnaast demonstreren we dat de gevalideerde geheugenfouten niet eenvoudig met bestaande testalgoritmen gedetecteerd kunnen worden en dat er dus speciale tests nodig zijn om ze te detecteren.

We passen de CBT-benadering ook toe om een test te ontwikkelen voor een RRAM die GIG kan uitvoeren. We tonen aan dat een GIG-geheugen zowel in de geheugenstaat als in de gegevensverwerkingsstaat getest moet worden, omdat er in beide staten unieke fouten kunnen optreden. We beschrijven de volledige geheugenfoutenruimte voor GIG-fouten en valideren deze middels de CBT-benadering. Vervolgens ontwikkelen we tests die de fouten in beide staten kunnen detecteren. Daarnaast bestuderen we hoe proces-, spannings- en temperatuurvariaties de prestaties van een GIG-geheugen beïnvloeden. De resultaten tonen aan dat bepaalde logische operaties veel gevoeliger zijn voor deze variaties dan andere.

ACKNOWLEDGEMENTS

After four years and a few months, I have finally come to the completion of my PhD. There are many people who supported me in one way or another and whom I would like to thank for that.

First, I am thanking my promotor and supervisor **prof. dr. ir. Said Hamdioui**. Said, in the past years I have learned an amazing amount of things from you. You taught me to think critically and communicate effectively. I did not understand the importance of this last skill four years ago, but I am now definitely convinced of its importance. We have had some long and demanding sessions to improve my scientific communications, but now I can say that it was worth it to spend the time. Thank you for that. I also value a lot the trust that you give me. From early on in my PhD, you allowed me to supervise students, collaborate with others, and to steer my research in the directions I deemed right. Later, this was followed by even larger teaching responsibilities, and the organization of larger and smaller events and activities. I want to thank you for these opportunities, as they made me a better scientist and engineer. Next to this, I also want to thank you for sharing your industrial knowledge, which allowed me to put my work in a broader perspective. I hope that the future keeps following this trend. Thank you again!

Next, I would like to thank my copromotor **dr. ir. Mottaqiallah Taouil**. Motta, you have guided me through my master degree, which led to this PhD degree. Throughout both, you provided me with critical insights on my work, while also helping to move it forward. Next to that, you kept me motivated when the work was hard. Thank you for that. Besides these serious points, I also value how you bring fun and laughter in the office. Maybe it is time that we actually work out one of your joke ideas. Besides that, I also thank you for giving me the opportunities to become a better teacher, by giving me responsibilities and discussing about proper teaching methods. Thank you!

I also want to thank my academic collaborators. First, **dr. Hassen Aziza** for teaching me the basics of RRAM design, for critically discussing my work, and for providing important measurement data. I really appreciate that you took the time to teach me in person in Marseille. I hope we can keep this fruitful collaboration going on. Secondly, I am thanking **prof. dr. Mehdi Tahoori** and **Christopher Münch MSc.** for their collaboration. I really enjoyed the technical discussions that we had about CIM, and the differences between STT-MRAM and RRAM., but also talking about less research-related topics during our calls and in Barcelona. Next, I thank **dr. Leticia Bolzani Poehls** and **dr. Thiago Copetti** for our cooperation. We have studied interesting subjects together and have developed a solid basis for further work. Thank you all. Besides the collaborations with researchers in other universities, I have also worked with great researchers in TU Delft. Thank you **dr. Rajendra Bishnoi** and **dr. Anteneh Gebregiorgis** for helping we with my research and sharing your valuable insights.

Thanks as well to both my paranymphs, **dr. Guilherme Cardoso Medeiros** and **ir. Rob Bootsman**. Guilherme, we started around the same time with our PhDs and completed them at a similar time as well. However, we are not just colleagues, I think we have become great friends as well. I want to thank you for your moral support on mental health Wednesdays, your feedback on drafts, your great company over a beer, your fierce competition during squash and many other events. Thanks, Timber! Rob, since the start of my university life, I know you. We have done so many things together, ranging from ordering Chinese food out of the /Pub via partying on galas via powernaps in unnamed locations to dinner discussions and Mario kart. Thank you!

The day-to-day life of my PhD was made a lot better by having the right office mates. **Daniël**, **Lizhou** and **Guilherme** thanks for providing a nice office environment. It was *waardeloos* unless you shout *Device-Aware Test* loudly and frequently. Next to office mates, I also had the pleasure to share my workday with many other people in the department. **Abdullah**, **Abdulqader**, **Abhairaj**, **Anh**, **Anteneh**, **Arwa**, **Cezar**, **Haji**, **Innocent**, **Jintao**, **Lei**, **Mahdi**, **Mark**, **Michael**, **Muath**, **Oscar**, **Shayesteh**, **Simon**, and **Troya**, thanks for the nice lunches that we had together, the daily coffee walks, board game nights, survival runs, ice skating, go-karting, sailing, bouldering, and many more events. I wish you all a good future. **Joyce**, **Leon**, and **Pascal**, thank you for organizing the department events with me and Guilherme. Let's have a drink soon. I would also like to thank the support staff of the QCE department. **Joyce**, **Lidwina**, **Trisha**, **Laura**, **Francis**, and **Paul**, thank you for keeping everything in order and for ensuring that my work could be performed effectively. Next, I also want to thank **Erik** for your great help in setting up the simulation environments. A part of the tasks of a PhD candidate is to convey knowledge to other students. However, I did not know that teaching someone else would teach me a lot as well. Therefore, I want to thank **Surya**, **Tijs**, **Filip**, and **Mathijs** for providing me with this important experience.

Although working in a department with so many great people really helped me in the past four years, I must also thank my friends outside of it. Thank you **Ludo**, **Marc**, **Ralph**, and **Kevin** from 142 for the great times in München and France, as well as the always eventful graduation parties, the countless borrels, galas, rallies, coffees, diners, and much more. I hope we keep making these amazing events in the future. Next, I want to thank Complex. Thank you **Arjen**, **Dennis**, **Eric**, **Lieuwe**, and **Maarten** for great companionship on regular evenings, but definitely also abroad in Germany, Belgium, Zeeland, and even Terschelling. It's time to visit the Hesburger in Hamburg. Of course, I also thank Hoezo? Ouzo. Thanks **Abel**, **Arjen**, **Bert**, **Marco**, and **Pieter** for the endless support when deciding which B.A. Quadrupel is the best one and deciding whether Delphi is really worse than Olympia. Thank you **Soufiane** for the interesting discussions we always have. They inspired me and even led to some of the propositions in this thesis. Next, thank you **Aron** for providing the necessary balance by not being an engineer in engineering environments.

I have received enormous amounts of support from my family during the last four years. Uit Haarlem wil ik graag mijn opa **Chris** en tante **Petra** bedanken. Lieve opa Chris, ik ben blij dat het "nachtcafé" altijd tot laat open is voor een belletje en ik ben je zeer dankbaar voor je hulp aan mijn huis. Lieve Petra, bedankt voor je motivatie en lekkere diners. Beiden, bedankt! Meine Familie aus Bremen, Oma **Liselotte**, Tante **Bettina**, Unkel **Georg**, **Philip** und **Charlotte**, ist mir auch sehr wichtig. Liebe Oma, vielen dank für die Wochenenden die für mich immer wie Ferien anfühlen. Auch vielen Dank für die Unterstützung für meine Wohnung; jedes Mahl wenn wir anrufen wird es etwas weiter sein. Ich danke die Familie Von Estorff für die Interesse die ihr immer habt in meiner Arbeit. Danke! Maar ook mijn Rotterdamse familie mag niet vergeten worden. **Hans**, **Ines** en **Peter**, bedankt voor jullie goede adviezen en interessante discussies rondom mijn promotie, een dak boven mijn hoofd in Rotterdam, en het opnemen van mij in jullie familie. Dankjulliewel! Lieve **Bente** en **Casper**, bedankt voor de stroom aan schattige kattenfilmpjes, de inside jokes, en jullie altijd gezellige aanwezigheid! Tot slot wil ik mijn ouders **Esther** en **Norbert** bedanken voor alles wat jullie voor mij hebben gedaan. Jullie zijn altijd open en bereikbaar voor mij en hebben me perfect ondersteund om hier te komen. Het begon met het verdragen van Legolawaai op zondagmorgens, via universiteitsbezoeken door heel Nederland, tot de uitvoerige gesprekken over mijn werk. Bedankt ook voor jullie hulp aan mijn woning. Bedankt! Ik hou van jullie allemaal, ich liebe euch allen!

Als laatste dank ik **Eva**. Eva, je hebt mij de afgelopen (bijna) vijf jaar door dik en dun gesteund. Vakanties op het water en in de bergen, lummelen in het weekend, en samen bouwprojecten doen, niks verveelt. Als ik het zwaar heb, ben jij er om me er doorheen te slepen. Als ik een succes ervaar, ben jij er om het met me te vieren. Ik ben ongelooflijk blij en dankbaar dat wij samen zijn. Jij bent er altijd voor mij, en ik ook voor jou. Ik hou van je en ik hou je.

Moritz
Delft, June 2022

LIST OF FIGURES

List of Tables

1

INTRODUCTION

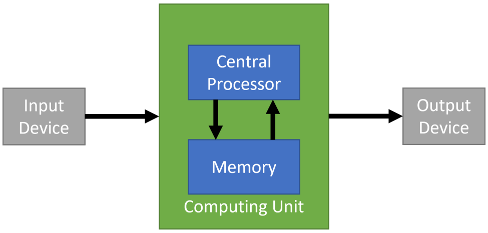


Figure 1.1: Von Neumann computer architecture

1.1. MOTIVATION

This section introduces the setting of this dissertation. First, we describe how computing systems work and how they make use of memory. Next, we present three walls that limit the performance of the computing systems nowadays, and we present how emerging memories can address these.

1.1.1. TRADITIONAL COMPUTING AND MEMORY

Currently, most computing systems have a Von Neumann architecture that splits the computation and storage. Computation is performed in a processor, and the data for the computation are stored in memory. In this section, we first introduce the Von Neumann architecture, and subsequently, the the memory hierarchy.

VON NEUMANN ARCHITECTURE

Fig. 1.1 shows the Von Neumann architecture. The green block is the computing unit. In there, the central processor fetches data from the memory, i.e., both instructions and data to apply these instructions on. Furthermore, it takes input from the outside world via input devices, e.g., a keyboard or an internet connection, and it outputs to the outside world via the output device, e.g., a monitor or a speaker. The benefit of separating processing and memory is that the central processor is independent of the memory contents. In other words, its structure is always the same and does not need to change for different data. Furthermore, the Von Neumann architecture allows to develop these components separately.

MEMORY HIERARCHY

The speed of the memory does not increase as fast as the speed of the central processor. This limits the overall computation speed, as the processor has to wait for the memory to load and store the data. This bottleneck is know as the Von Neumann bottleneck. To mitigate it, the memory is divided into layers that form a memory hierarchy. The aim of this hierarchy is to have just enough fast and expensive memory close to the central processor, while having slower and cheaper memory further from the processor. This memory hierarchy is shown in Fig. 1.2 and consists of five layers. The top layers are the fastest but also the most expensive per bit, while the lower layers are slower but cheapest per bit.

- **Registers:** The registers are placed directly next to the arithmetic unit in the central processor. They can store and load data fast (usually within a single clock cycle).

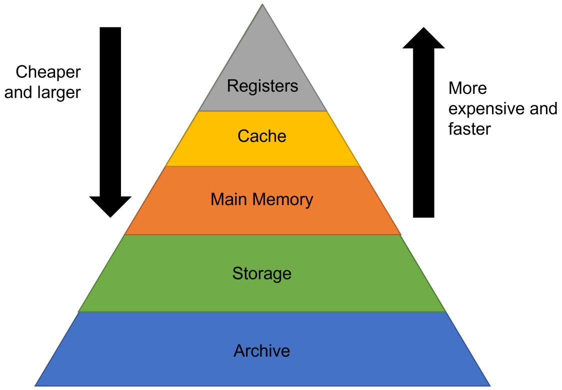


Figure 1.2: Memory Hierarchy

However, registers require the most transistors per bit. Registers lose their data when powered off, i.e., they are volatile.

- **Cache:** Cache is placed on the same chip as the the central processor. It is implemented as static random access memory SRAM, which costs six transistors per bit. This is cheaper than registers, but also slower. Typically, multiple levels of cache are present, e.g., level 1, level 2 and level 3 cache, where each higher level contains more bits but is also slower than the lower levels. SRAMs are also volatile memory.
- **Main memory:** The main memory is typically not placed on the same die as the central processor, it is connected via a fast but instead. Main memory is implemented as dynamic random access memory (DRAM) and uses only one transistor and one capacitor per bit. DRAMs are also volatile memory.
- **Storage:** Storage class memory stores the bulk of data local to the computing unit, also when powered down, i.e., it is non-volatile. This memory is implemented as Flash memory or a hard disk drive (HDD). Flash is faster than a HDD and can store multiple bits using a single transistor. HDDs store data on a magnetic disk which is lower than Flash but also allows for even more and cheaper storage capacity.
- **Archive:** Archive memory is memory that is stored far away from the computing unit. This can be, for example, magnetic tape, or cloud storage on a server.

1.1.2. THREE WALLS

The umber of transistors on a chip double every two years, as predicted by Moore [2]. This prediction is known as Moore's law. The technological implementation, e.g., doping concentrations, transistor threshold voltage, frequency, etc., followed a scaling pattern that was predicted by Dennard, Gaensslen, Yu, *et al.* [3]. This prediction was valid until the 2000s. At that time, it became impossible to scale transistors down further, while also scaling their performance in a similar manner. This has led to the identification of three walls that limit the development of computing devices [1], [4]. These are discussed next.

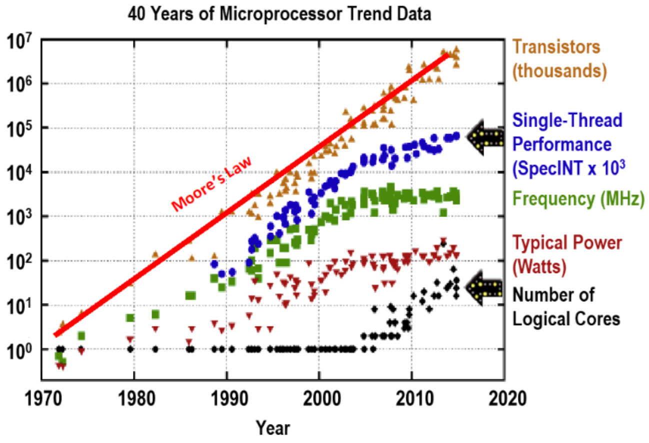


Figure 1.3: Microprocessor trends [1]

Power Wall Due to Moore's law, the power density of a chip increases and thus the heat generated in the chip increases. This is visible in the typical power graph in Fig. 1.3. Around 2005, chips started to produce too much heat. From that point on, it became too expensive to enable all transistors on a chip or increase the switching frequency further, as shown in the frequency graph in Fig. 1.3.

Memory Wall Memory speeds have not scaled in the same pace as processor speeds. As such, the overall speed of a computer is more and more determined by the speed of the memory. To overcome this, more levels of cache are introduced. An alternative is to move away from the Von Neumann architecture and move computations into the memory to mitigate the need for data transfer completely. This is called computation-in-memory (CIM).

Instruction Level Parallelism Wall In the past, microprocessors the performance of processors was improved by performing more instructions in parallel, e.g., via instruction level parallelism, where one instruction is applied to multiple registers at once. This has increased the performance of a single processor, as shown in the single-thread performance graph in Fig. 1.3. However, the number of operations that can be performed in parallel is limited in many algorithms. As such, increasing instruction parallelism further results only in marginal further improvements in a single processor. To overcome this, multiple logical cores were introduced on a single chip, as shown in the number of logical cores graph in Fig 1.3. Here, each core can run a separate program and thereby achieve better overall computational efficiency.

1.1.3. EMERGING MEMORY TECHNOLOGIES

To address the walls, emerging memory technologies can be used, such as phase change random access memory memory (PCRAM, also Phase Change Memory, PCM), spin-

Table 1.1: Comparison of mainstream and emerging memories [5]

	Mainstream Memories				Emerging Memories		
	SRAM	DRAM	NOR Flash	NAND Flash	PCRAM	STT-MRAM	RRAM
Cell area	$> 100F^2$	$6F^2$	$10F^2$	$< 4F^2$	$6 - 50F^2$	$4 - 30F^2$	$4 - 12F^2$
Multibit	1	1	2	3	2	1	2
Voltage	< 1 V	< 1 V	> 10 V	> 10 V	< 3 V	< 1.5 V	< 3 V
Read time	1 ns	10 ns	50 ns	10 µs	< 10 ns	< 10 ns	< 10 ns
Write time	1 ns	10 ns	10 µs – 1 ms	100 µs – 1 ms	50 ns	< 10 ns	< 10 ns
Retention	None	ms	years	years	years	years	years
Endurance	$> 10^{16}$	$> 10^{16}$	$> 10^{5}$	$> 10^{4}$	$> 10^{9}$	$> 10^{15}$	$> 10^{6}$
Write energy per bit	1 fJ	10 fJ	100 pJ	10 fJ	10 pJ	100 fJ	100 fJ

transfer torque random access memory (STT-MRAM), and resistive random access memory (RRAM or ReRAM). These memories store data as a resistance, rather than as charge. These technologies have several benefits over mainstream memory technologies, such as SRAM, DRAM and Flash. Table 1.1 lists key features of all of these technologies [5]. In this table, for the cell area, F denotes the minimum feature size; multibit indicates how many bits can be stored in a cell; retention indicates how long data remain valid after power-off; endurance indicates how many times a cell can be written. From the table it follows that the emerging memories can be made smaller than SRAM and in some cases also DRAM and NOR Flash. In combination with multibit storage, this leads to denser memories, which reduces the memory latency. They also operate on lower voltages than Flash, which makes them more easily suitable for embedded applications. The read and write times are comparable to DRAM and significantly lower than for Flash, reducing the memory latency. Next, they are non-volatile, in contrast to SRAM and DRAM. This allows for lower static energy consumption, and thus addresses the power wall. Their endurance is higher than the endurance of Flash, but lower than for SRAM and DRAM. Finally, the write energy per bit is higher than most mainstream memories. The emerging technologies can also be effectively applied to perform CIM due to the fact that they store data as a resistance.

Next, we will discuss the basic concepts of these three technologies.

Phase Change Random Access Memory Fig. 1.4(c) and 1.4(d) show two PCRAM device structures. The concept of a PCRAM device is that the phase of the phase-change material, and consequently the resistance, is changed by heating it up. The melting of the material requires high currents, and thus writing a PCRAM cell has a high power consumption. Furthermore, the melting process is relatively slow. A positive feature of PCRAMs is that the resistance of the material can be tuned so that multiple bits can be stored in a single cell.

Spin-Transfer Torque Magnetic Random Access Memory Fig. 1.4(a) and 1.4(b) show two magnetic tunnel junctions (MTJs) that are used in STT-MRAMs. MTJs work on the concept of magnetic spin. They have a pinned layer where the spin is always in the same direction and a free layer where the spin can be parallel or anti-parallel to the spin of the free layer. The spin can be changed by driving a current through the MTJ. If the spins are

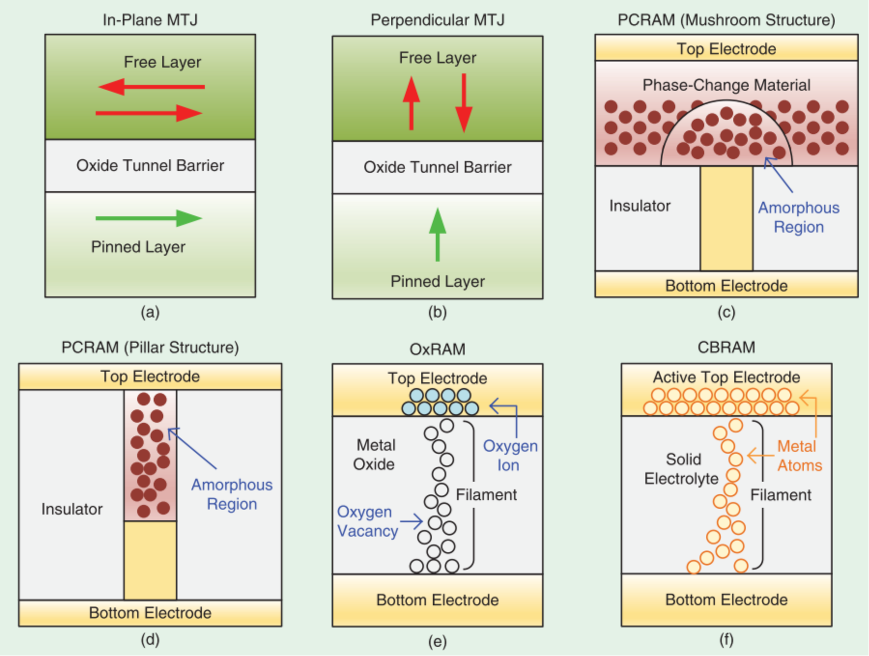


Figure 1.4: Overview of emerging memory structures [5]

parallel, the resistance is low, while the resistance is high if the spins are anti-parallel. Because there are only two spin combinations, an MTJ can only store one bit.

Resistive Random Access Memory Fig. 1.4(e) and 1.4(f) show two RRAM structures. RRAMs have a filament that connects two electrodes. The shape, e.g., length, width, etc, of the filament determines the resistance of the device. The shape of the filament can be changed by applying voltages to the electrodes. There are two kinds of RRAM that are based on the atomic structure of the filament: oxide-based RAM (OxRAM) has a filament made of oxygen ions, and conductive bridge RAM (CBRAM) has a filament made of metal ions. RRAM devices can be manufactured densely. Furthermore, the shape of the filament can be controlled, so that multiple bits can be stored in one device. However, the endurance is limited in comparison to STT-MRAM.

Based on the analysis above, it is expected that STT-MRAM can be used to replace lower levels of cache or the main memory, while PCRAM and RRAM are expected to replace slower main memories but mainly storage class memory technologies such as Flash. However, currently, these emerging memories are only used in smaller nice markets, and the mainstream memory technologies remain widely used. This is partially due to the fact that the emerging memories require novel production process and materials. These lead to new failure modes and require the development of new specialized tests. For STT-MRAM, interesting developments have been presented in [6]. Since PCRAM is rather slow and has a high energy consumption, in this thesis, we focus on defect and fault modeling, and test development for RRAM, more specifically OxRAMs.

1.2. RRAM DEFECTS, FAULTS, AND TESTING

Bushnell and Agrawal [7] defines defects, errors, faults as follows. A defect is an unintended deviation in the actual manufactured chip from the design, e.g., a short circuit between two metal wires. This defect can lead to an error in the operation of the circuit, e.g., a counter produces a wrong result for certain values. A fault is the abstraction of a defect on the functional level, e.g., there is a count up fault. To detect the defect, the defect needs to be modeled and the sensitized faults need to be understood. Next, a test solution can be developed that detects these faults. Traditionally, for mainstream memories, defects are modeled as linear resistors between two nodes or within a connection [8], [9]. Subsequently, the faults that are sensitized by these defect models are identified, and a test to detect them is developed. For RRAM, researchers have followed a similar approach. They identified unique RRAM faults, and confirmed the existence of traditional memory faults in RRAM. Subsequently, tests are presented to detect these faults.

Ginez, Portal, and Muller are the first to address testing of RRAMs [10]. They study bridge defects between the bit and word lines of a RRAM. Their analysis results in the confirmation of coupling faults in RRAM. Next, Haron and Hamdioui studied more bridge defects and identified the RRAM unique undefined write fault and confirmed the traditional transition, stuck-at and incorrect read faults [11]. Their follow-up work in [12], [13] presents the first test solution to detect faults by modifying the write operation. Kannan, Rajendran, Karri, *et al.* identified two other RRAM unique faults, namely the deep-0 and deep-1 state. Furthermore, the authors also present a test solution to detect these based on the inherent sneak paths in a RRAM [14]–[16]. Mozaffari, Tragoudas, and Haniotakis introduced an alternative test that speeds up the testing process [17], [18]. Further progress on identifying defects and faults has been made by Chen, Shih, Wu, *et al.* [19]. They identified the over-forming defect and the read-1 disturb fault it sensitizes. The authors present a test algorithm to detect all RRAM faults. Finally, Chen and Li again use the resistive bridge defect model to identify and develop a test for destructive write faults [20]. Recently, Liu, You, Wu, *et al.* confirmed two additional coupling faults [21]. Finally, Copetti, Gemmeke, and Bolzani Poehls developed a test circuit to detect undefined state faults in a more efficient manner [22].

There has also been a focus of the research community on making the tests more efficient or applicable to different RRAM architectures. In [23], Liu, You, Kuang, *et al.* present a test algorithm that detects all RRAM faults with the exception of the destructive write faults. Luo, Cui, Luo, *et al.* improved this test to also include those faults [24]. Next, Liu, You, Kuang, *et al.* reduced the test time of this algorithm by including a special test structure that allows to read multiple cells at once [25]. Finally, Liu, You, Wu, *et al.* developed a test for another RRAM architecture, again based on the linear resistor defect models [26].

The above test solutions are all based on the linear resistors defect model that affects the interconnections of a circuit that *surround* the RRAM device. However, it is clear that this defect model does not describe defects *in* the RRAM device. As such, it can be expected that these tests will not be able to detect all RRAM faults and therefore will lead to low-quality RRAMs.

1.3. RESEARCH TOPICS

From the previous sections, it has become clear that RRAM is a suitable technology to implement future memories. However, in order to guarantee high quality products, better test solutions need to be developed. These tests need to take into account the actual physics of RRAM defects, rather than defects in the surrounding circuitry. The research that needs to be done to achieve this can be divided in three steps: 1) accurate defect modeling, 2) accurate fault modeling and analysis, 3) high-quality test development. We will discuss these steps next.

1.3.1. ACCURATE DEFECT MODELING

Accurate defect modeling is the key to obtain high-quality tests. If the defect model is not accurate, the faults that are sensitized will not be realistic, and thus the test will be of low-quality. Hence, an approach to develop these accurate defect models for RRAM is needed. First, the production process of a RRAM needs to be analyzed and the potential defects that may occur there need to be identified. Next, measurements of these defects need to be obtained, to understand what causes them, and how they affect the circuit. Then, a defect model can be developed that accurately describes the effects of the defect. Furthermore, this procedure leads to a better understanding of RRAM devices and even new defects may be identified based on it. There exist CIM architectures that are based on RRAMs. Hence, also this production process needs to be analyzed in order to identify the defects that may occur there.

1.3.2. ACCURATE FAULT MODELING AND ANALYSIS

Once the accurate defect models are developed, they can be used to perform accurate fault modeling. The fault modeling needs to be done in a systematic way that ensures that all faults that can occur in a circuit are properly sensitized. Therefore, first the complete fault space needs to be defined. This definition needs to take into account the unique RRAM behavior. Next, the fault space must be validated using the accurate defect models. The faults that are sensitized are then proven to be realistic and accurate faults and allow to develop high-quality tests. CIM architectures are not only a memory, but also a computational unit. As such, it is expected that CIM architectures that are based on RRAMs suffer from unique RRAM computing faults. These faults also need to be modeled and analyzed in a systematic manner.

1.3.3. HIGH-QUALITY TEST DEVELOPMENT

Once it is clear which defects introduce which faults in the RRAM, high-quality test solutions can be developed to detect these. There are two important factors when developing a test for RRAM: 1) the test needs to detect all RRAM faults, even those that are unique and may proof difficult to detect; 2) the test needs to be efficient in terms of test time and hardware resources used. The first criterion requires that for every fault the detection criteria are determined. It may turn out that need specialized test circuitry is needed to guarantee detection of the fault. The second criterion ensures that the cost to detect all these faults is as low as possible. For this aim, methodologies need to be developed that optimize the test solutions.

1.4. CONTRIBUTIONS

The contributions of this thesis address contribute to all three research topics that were presented in the previous section.

Investigation and Analysis of Existing RRAM Test Solutions We have performed a literature study to investigate the existing RRAM test solutions. It turns out that many of the existing solutions are based on linear resistor defect models, and as such do not describe realistic RRAM defects. This has been published in [27].

Investigation of RRAM Production Process and Identification of Defects We have investigated the complete production process of a RRAM in order to identify all the defects that may occur there. From this investigation, we identified that defects can occur in the transistors, the interconnections, and in the RRAM devices. We describe in detail what causes the defects and how the defects will affect the RRAM. The investigation has resulted in a list of defects that form the complete RRAM defect space for the considered production process, i.e., the list describes all RRAM defects that may occur. This list can be used for subsequent defect modeling. This has been published in [27]–[29].

Development of the Device-Aware Test Approach In order to develop realistic defect models, we have developed the Device-Aware Test (DAT) approach. The approach consists of three steps: 1) device-aware defect modeling; 2) device-aware fault modeling and analysis; and 3) device-aware test development. The core of the approach is to model the actual physics of a defect and incorporate the effects of the defect in a defect model that can be used to perform accurate fault modeling and analysis. During the device-aware fault modeling and analysis step, first the fault space is systematically defined. Then, it is systematically validated to obtain a set of faults that can occur in a circuit. The third step, device-aware test development, uses the validated faults to develop high-quality test solutions. This has been published in [28], [29].

Application of the DAT Approach to Forming Defects We apply the DAT approach to model forming defects in RRAMs. We develop a device-aware defect model, perform the device-aware defect modeling, and develop a device-aware test for them. From the results it follows that traditional test approaches will result in tests that do not detect many of the faults that are sensitized and thus lead to test escapes. Furthermore, they will test for unrealistic faults and thus lead to yield loss. In contrast, we develop a test that detects the forming defects accurately without resulting in yield loss or test escapes. This has been published in [28], [29].

Identification of and Application of the DAT Approach to Intermittent Undefined State Faults We identify the intermittent undefined state fault based on silicon measurements. This fault changes the switching mechanism of the RRAM device intermittently, which leads to faults. We apply the DAT approach to develop a defect model that can sensitize this fault. The subsequent fault analysis shows that this fault is difficult to detect with regular RRAM tests. This has been published in [30].

Device-Aware Test Development and Reliability Analysis for RRAM-Based CIM We have applied the DAT approach on RRAM-based CIM architectures. We demonstrate that the fault space for such an architecture is different from that of a regular memory and that there exist faults that are uniquely due to the computational capabilities. To detect these faults, we present test solutions that make use of the computing capabilities. Furthermore, we analyze how susceptible RRAM-based CIM is to process, voltage, and temperature (PVT) variations. This has been published in [31], [32]

1.5. THESIS ORGANIZATION

This thesis is organized as follows.

Chapter 2 presents background information on RRAMs. First, the basics of the device structure and of the switching principles of a RRAM device are explained. Subsequently, we give an overview of RRAM device modeling and compare nine compact models. Compact models can be used to model a RRAM in a circuit, e.g., a circuit that is simulated using a SPICE simulator. Next, we present the RRAM architectures that are used in this thesis: a regular RRAM, and a modified version of it that allows for CIM. Finally, we give an overview of existing RRAM chips.

Chapter 3 explains the details of the RRAM structure and its production process. We analyze this process and identify all manufacturing defects that can occur there, both in the RRAM device and the other components on the chip. A selection of these defects will be modeled later in the thesis. Next, the knowledge of the device structure is applied to explain the concepts of RRAM reliability.

Chapter 4 presents the traditional RRAM defect modeling, fault modeling, and test development approach. This approach models all defects in the circuit as linear resistors between two nodes or within a connection. We apply this method to model all interconnection defects in a single cell. Next, we define and validate the complete fault space for RRAMs and validate it using the traditional defect models. Furthermore, we present the RRAM faults and tests that have been presented in literature. We apply the existing test solutions to detect the faults that we validated.

Chapter 5 introduces the Device-Aware Test Approach. This approach models the actual physics of a defective device in order to obtain tests for realistic faults. The DAT approach consists of three steps: device-aware defect modeling, device-aware fault modeling, and device-aware test development. We apply these steps on the interconnection defects that are modeled in the previous chapter.

Chapter 6 applies the DAT approach to model the RRAM forming defect. This defect affects the forming of the initial RRAM filament and thus has a strong impact on quality of the device. First, we characterize the defect and perform the physical modeling of the defect, resulting in a defect model. This defect model is used during fault analysis, to validate which faults actually occur in the RRAM. Only for those faults, we develop a high-quality test.

Chapter 7 applies the DAT approach to model the RRAM intermittent undefined state fault. This faults causes the RRAM device to switch into an undefined state at intermittent periods. First, we characterize the fault based on measurements of defective devices. Next, we develop a defect model based on the physics of the defect. This model is then used for fault analysis. The results of this are used to develop test solutions that can detect this fault.

Chapter 8 applies the DAT approach to develop test for CIM architectures and it studies the effects of process, voltage and temperature on the performance of CIM. We show that CIM architectures suffer from different faults than regular memories. Thus, we define the fault space for CIM faults and validate it. Subsequently, we develop a test that detects these faults that are unique for the CIM architecture. Finally, we demonstrate that CIM architectures are susceptible to PVT variations and address possible solutions to prevent this impact.

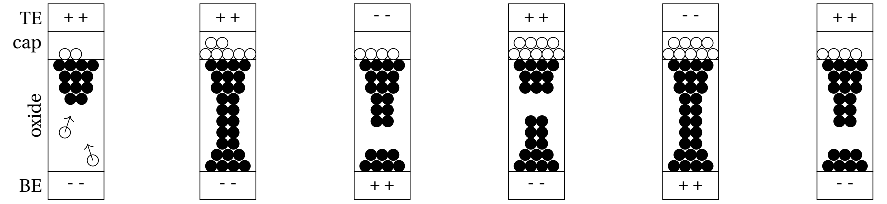
Chapter 9 concludes this thesis and presents some insights on future work.

2

BACKGROUND

This chapter presents a background on RRAMs. First, it presents the basic RRAM device structure and it explains three resistive switching mechanisms: bipolar, complementary, and unipolar switching. Next, an overview of RRAM compact models is presented. These models can be included in a netlist and simulated using a SPICE simulator. Subsequently, a complete RRAM simulation model is presented. It consists of a cell array, drivers, and sense amplifiers. Next, this simulation model is extended so that it can perform computation-in-memory. Finally, the chapter concludes with an overview of existing RRAM prototype and commercial chips.

This chapter is partially based on [29], [30], [33].



(a) Forming (b) BS SET (c) BS RESET (d) CS RESET (e) CS SET (f) US RESET

Figure 2.1: Evolution of the conductive filament

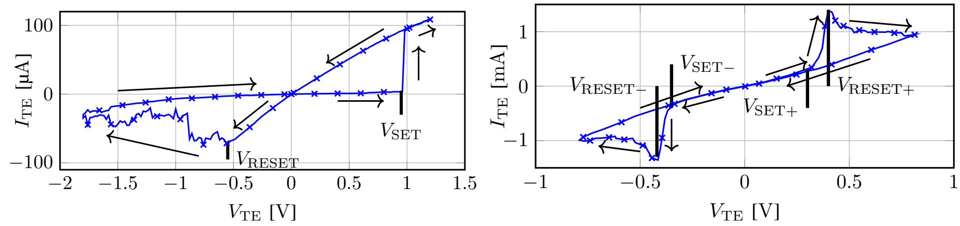
Figure 2.2: RRAM cell resistance and corresponding logic values

2.1. RRAM DEVICE TECHNOLOGY

This section introduces the RRAM device technology. First, we present an overview of the device. Then we introduce three switching mechanisms for a RRAM device.

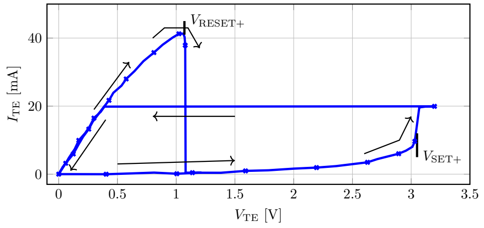
2.1.1. OVERVIEW

Fig. 2.1a illustrates a RRAM device; it consists of two electrodes, the top electrode (TE) and bottom electrode (BE), in between which a metallic oxide is sandwiched (commonly, HfO_x, TaO_x, or TiO_x) with an additional capping layer [5], [34], [35]. Chapter 3 describes the structure in more detail. After fabrication, a voltage V_{form} is applied to the oxide to break some of the metal and oxygen ions bonds [34], [35]. Fig. 2.1a shows how the (negatively charged) oxygen ions (white circles) move towards the positive electrode and enter into a oxygen exchange layer at the TE interface under the influence of the electric field and bond there [35], [36]. Note that in some devices this layer is a specially deposited capping layer (cap). This leaves behind a chain of conducting vacancies (black circles) known as Conductive Filament (CF), as shown in Fig. 2.1b. When the voltage is removed, the CF remains, making the RRAM device non-volatile. The shape (e.g., radius) of the CF determines the resistance of the device [36]; it can be changed by applying voltages to the oxide. When using the RRAM device as a binary storage device, resistance ranges have to be defined that correspond to a logical value. This is shown in Fig. 2.2. Here, a low resistance and high conductance correspond to a logical '1', a high resistance and low conductance correspond to a logical '0'. Between these two ranges, the resistance is in the undefined 'U' state. When the resistance is below the valid '1' range, the device is in extreme high state 'H'. Similarly, when the resistance is above the valid '0' range, the device is in extreme low state 'L'. There exist three switching mechanisms in RRAM: Bipolar Switching (BS), Complementary Switching (CS), and Unipolar Switching (US). These will be explained next.



(a) Bipolar switching (BS)

(b) Complementary switching (CS)



(c) Unipolar switching (US) redrawn from [37]

Figure 2.3: Resistance switching mechanisms in RRAM.

Table 2.1: BS RRAM key parameters

Technological Parameters		Electrical Parameters	
t_{ox}	Oxide thickness	V_{RESET}	RESET threshold
l_{gap}	CF gap length	V_{SET}	SET threshold
ϕ_{T}	CF thinnest width	R_{HRS}	RESET resistance
		R_{LRS}	SET resistance

2.1.2. BIPOLAR SWITCHING

The BS mechanism relies on the formation and dissolution of the CF by movement of *oxygen ions* [34], [38]. Its electrical behavior is illustrated in Fig. 2.3a. When a negative voltage V_{SET} less than a threshold V_{reset} is applied to the TE, then some of the oxygen ions drift back into the oxide and re-oxidize the CF leaving a gap in the CF, as shown in Fig. 2.1c. This movement increases the resistance of the device and is called a RESET operation, resulting in the high resistance state (HRS), with has a resistance of R_{HRS}. When a positive voltage higher than a threshold V_{SET} is applied, the bonds between the metal and oxygen ions break again, and the CF regrows, as shown in Fig. 2.1b. The oxygen vacancy-rich capping layer collects the free oxygen ions [35]. This process is called a SET operation and results in a lower device resistance, the low resistance state (LRS), which has a resistance of R_{LRS}. Note that the opposing sign of the switching voltages gives BS its name.

Fig. 2.4 and Table 2.1 show and list the key technology parameters of BS RRAMs. The technology parameters are: the thickness of the oxide (t_{ox}) that affects the voltage switching thresholds, the length of the CF gap (l_{gap}) that affects the resistance in the RESET state, and the thinnest width of the filament (ϕ_{T}) that affects the Resistance of the device in both the SET and RESET state. The electrical parameters are the voltages of the

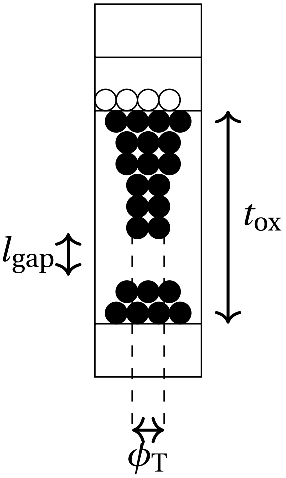


Figure 2.4: BS RRAM key technology parameters

switching thresholds to RESET (V_{RESET}) or SET (V_{SET}), and the resistance in RESET (high resistance state (HRS), R_{HRS}), and the resistance in SET (low resistance state (LRS), R_{LRS}).

2.1.3. COMPLEMENTARY SWITCHING

The CS mechanism relies on the formation and dissolution of the CF by movement of *oxygen vacancies* [39], [40]. Its electrical behavior is illustrated in Fig. 2.3b. After initial forming, a normal BS RESET is performed, resulting in the CF shown in Fig. 2.1c. Now, by applying a voltage exceeding the switching threshold V_{SET+}, a SET operation will take place (similar to BS) that results in the breaking of the bonds between the oxygen and metal ions and the forming of the CF as shown in Fig. 2.1b. However, when the voltage is further increased and exceeds the threshold V_{RESET+}, the negative oxygen ions cannot all be bound in the capping layer. Instead, they will remain in the oxide and the positive oxygen vacancies in the CF will start to move along the electric field towards the negative electrode. This disrupts the just created CF and brings the device in a RESET state, as shown in Fig. 2.1d. Note that both SET and RESET occur at the same voltage direction, they complement each other. When a negative voltage is applied less than the threshold V_{SET-}, first the oxygen vacancies will move back into the oxide and form a CF, see Fig. 2.1e. With further decreasing of the voltage below threshold V_{RESET-}, the CF will be broken again by the oxygen ions that move back into the oxide from the capping layer, leading to the state shown in Fig. 2.1c. The key technology parameters for a CS device are similar to those of a BS device, except that there are more switching thresholds and resistance states for the former than for the latter.

2.1.4. UNIPOLAR SWITCHING

The US switching mechanism relies on the formation of and dissolution of the CF by the movement of *oxygen ions*, similar to BS. However, for US, the process is controlled by *diffusion* of the ions due to thermal heating [34], [41]. It electrical switching behavior is illustrated in Fig. 2.3c [37]. After initial forming, the device will be in a state similar to BS SET, as shown in Fig. 2.1b. Now, by applying a positive voltage that exceeds the threshold $V_{\mathrm{RESET}+}$, there will be so much current flowing that that the device heats. Due to this heat, the oxygen ions *diffuse* away from the TE and re-oxidize the CF, resulting in Fig. 2.1f. The US SET operation is similar to the BS SET operation; by applying again a positive voltage to the TE that is higher than the threshold, the oxygen ions break away from the metal ions and are attracted by the electric filed to the oxygen exchange layer. Note that US occurs at a single polarity. However, it is also possible to have US with opposite switching polarity. The key technology parameters for an US device are similar to those of a BS device, except that there are more switching thresholds and resistance states for the former than for the latter.

2.1.5. VARIATIONS

The growth and dissolution of the CF is a stochastic process [35]. Therefore, there are variations in the resistance of different devices, called device-to-device (D2D) variations, as well as variations per SET/RESET cycle within one device, called cycle-to-cycle (C2C) variations. D2D variations are caused by variations in the production process of the RRAM devices, e.g., variations in the crystal structure of the oxide, or in the forming current [35], [42]. C2C variations are caused by the random growth and dissolution of the CF in a cycle, e.g., variations in the radius of the filament [42]. Chapter 3 explains the sources of variation in detail.

2.2. RRAM DEVICE MODELING

IN this section, we present models that can describe the RRAM behavior. First, we introduce the different levels on which a RRAM device can be modeled. After that, we present an overview of compact models and compare them.

2.2.1. MODELING LEVELS

Lanza, Wong, Pop, *et al.* in [43] classify RRAM models in four categories based on the accuracy and simulation complexity, as shown in Fig. 2.5. In general, higher modeling accuracy requires longer simulation time. The model categories are: microscopic models, kinetic Monte Carlo or finite element models, and compact models. These will be discussed next.

Microscopic models contain both ab initio models and Monte Carlo models. These models describe the interactions of individual particles with each other in the RRAM device. These interactions determine the electrical behavior of a RRAM device. For example, these models can be used to make predictions on the growth and the shape of a CF for a wide range of materials. Ab initio models have the highest physical accuracy as they describe the quantum interactions of the particles. Monte Carlo models have a lower accuracy due to a simplification of some of the interactions, but they are able to model more elements. Microscopic models allow to investigate the impact of even the smallest changes in the RRAM device structure, but this requires a long simulation time that prohibits their usage in circuit simulations.

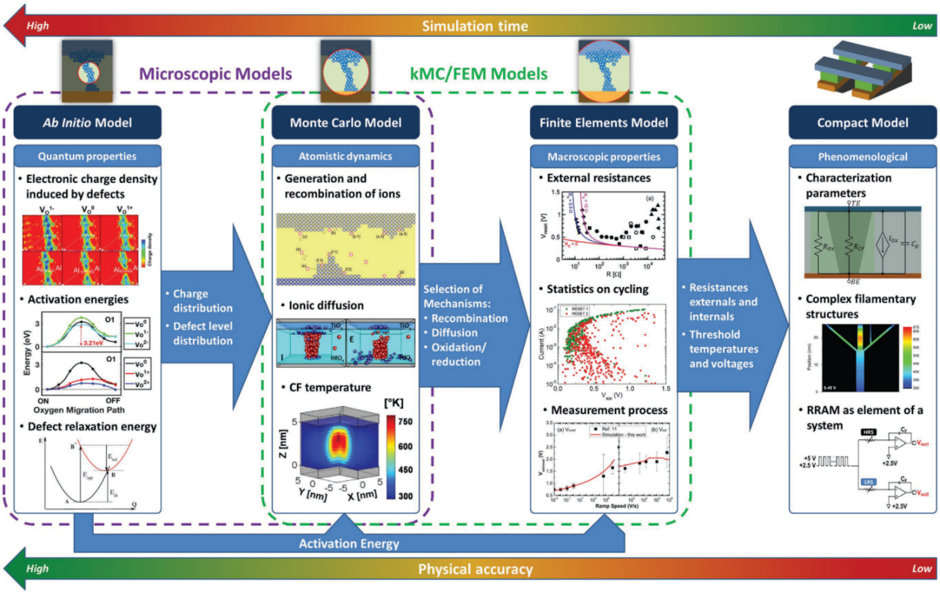


Figure 2.5: RRAM modeling levels [43]

Kinetic Monte Carlo or finite element models describe the electrical behavior of a RRAM device directly, while also considering the effects of the material. These models generalize and simplify some of the quantum interactions of the microscopic models in order to speed up the simulation. Kinetic Monte Carlo models take the contributions of individual particles in consideration, while finite element models generalize the physical behavior, e.g., using differential equations to describe the electric field in the RRAM device. The benefit of kinetic Monte Carlo is that they also describe variation effects, while finite element models are faster to simulate. Both these models have a lower physical accuracy than microscopic models, but they have a shorter simulation time. However, the simulation time is still inhibits the usage in larger circuit simulations.

Compact models also describe the electrical behavior of a RRAM device, but simplify the underlying physics further so that the model can be simulated quickly. For example, these models assume that there is a a CF with a certain shape that always changes in the same way when similar voltages are applied to it. The generalization that these models make allows to simulate them quickly. Hence, these models are used in circuit simulations. In the remainder of this thesis, we focus on compact models because these can be used in the circuit simulations that need to be done for test development.

2.2.2. COMPACT MODELS

Hajri, Aziza, Mansour, *et al.* in [54] compare eight RRAM compact models for BS on multiple metrics. Table 2.2 shows the results of this comparison and adds an additional RRAM compact model that was not present at the time of publishing (the model in [53]) to the comparison. Note that only the metrics relevant to this thesis are included. The

Table 2.2: Comparison of different RRAM compact models, partially based on [44]

Metric	Model								
	Linear [45]	Non-linear [46]	Simmons [47]	TEAM [48]	VTEAM [49]	Stanford [50]	SPICE [51]	IM2NP [52]	JART v1B [53]
Genericity	No	No	No	Yes	Yes	Yes	Yes	Yes	Yes
Non-linearity	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Hard SET and soft RESET	No	No	No	Yes	Yes	Yes	Yes	Yes	Yes
Forming	No	No	No	No	No	No	No	No	No
High frequency	No	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Switching thresholds	No	No	No	Yes	Yes	Yes	Yes	Yes	Yes
Pulse voltage dependence	No	No	No	No	Yes	Yes	Yes	Yes	Yes
Pulse timing dependence	No	No	No	No	No	Yes	No	Yes	Yes
Temperature dependence	No	No	No	No	No	Yes	No	Yes	Yes
Variability	No	No	No	No	No	Yes	No	Yes	Yes

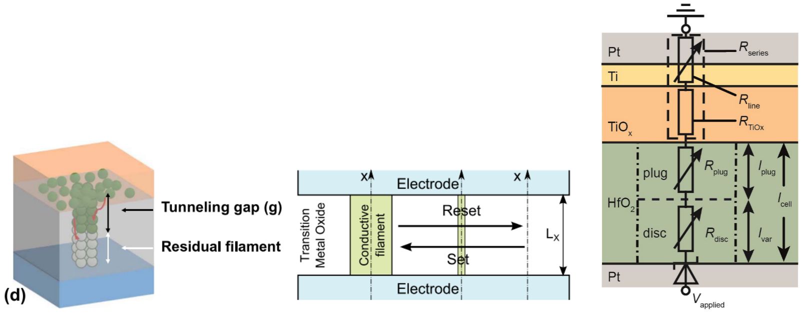
metrics are described next.

- Genericity: Can different types of RRAM devices be modeled as well, e.g., other materials, other structures, etc.
- Non-linearity: Do the models describes the switching process as a non-linear process.
- Hard SET and soft RESET: Does the model describe the SET process as an abrupt switching event and the RESET process as a gradual switching process.
- Forming: Does the model describe the CF forming process.
- High frequency: Can the model be used with high frequency inputs.
- Switching thresholds: Does the model incorporate the switching thresholds.
- Pulse voltage dependence: Does the model take into account the voltage dependence when a pulse is applied. That is, do higher voltages lead to faster switching.
- Pulse timing dependence: Does the model take into account the time dependence when a pulse is applied. That is, do longer pulses lead to a lower resistance after switching.
- Temperature dependence: Is the current through the device dependent on temperature.
- Variability: Can the model describe variability in the switching process.

From the Table, it can be seen that the Stanford model [50], the IM2NP [52], and the JART v1B [53] model perform best on the criteria listed above. Note that in [44] it is noted that the IM2NP model also describes the forming process, but that this is not documented in [52]. We discuss these models in more detail next.

STANFORD MODEL

The Stanford model in [50] models the CF as a current conducting cylinder, as shown in Fig. 2.6a. When a RESET operation is applied, the cylinder shortens and a tunneling gap develops that limits the current through the device. When a SET operation is applied, the tunneling gap shortens (and thus, the cylinder grows taller) until it is gone. The current



(a) Stanford model [50] (b) IM2NP model [52] (c) JART v1B model [53]

Figure 2.6: Basic concept of three compact RRAM models.

is then limited by the size of the cylinder. For every time step in the simulation, the local temperature, the change in the filament length, and the resulting current through the device are calculated. Variability is included by adding a stochastic component to the growth and dissolution rate of the CF.

IM2NP MODEL

The IM2NP model in [52] models the CF as a cylinder with a radius that varies when applying a SET or RESET operation, as shown in Fig. 2.6b. A positive voltage increases the radius and brings the device in the LRS, while a negative voltage reduces the radius of the CF and brings it in the HRS. In LRS, the conductance is determined by the ohmic resistance of the cylinder, while in HRS, the current is determined by the leakage through the oxide. For every time step in the simulation, the local temperature, the change in the filament radius, and the resulting current through the device are calculated. Variability is modeled by adding randomness to the radius parameters.

JART V1B MODEL

The JART v1B model in [53] models the RRAM device as a device with two regions, the well-conducting plug region and the less conducting disc region, as shown in Fig. 2.6c. The conductance of the disc region changes when applying voltages to the device. A SET operation increases the vacancy density of the disc, while a RESET operation decreases the vacancy density of the disc. For every time step in the simulation, the local temperature, the change in the disc vacancy density, and the resulting current through the device are calculated. Variability is modeled by adding a stochastic component to the radius, length, and vacancy density limits of the disc.

All the above models are calibrated to measurements of actual RRAM devices. In the remainder of this thesis we will use the Stanford [50] and JART v1B [53] models to model the RRAM device.

2.3. MEMORY ARCHITECTURE

This section introduces a regular RRAM architecture. First, we provide an overview of the architecture. Second, we analyze the components in this architecture.

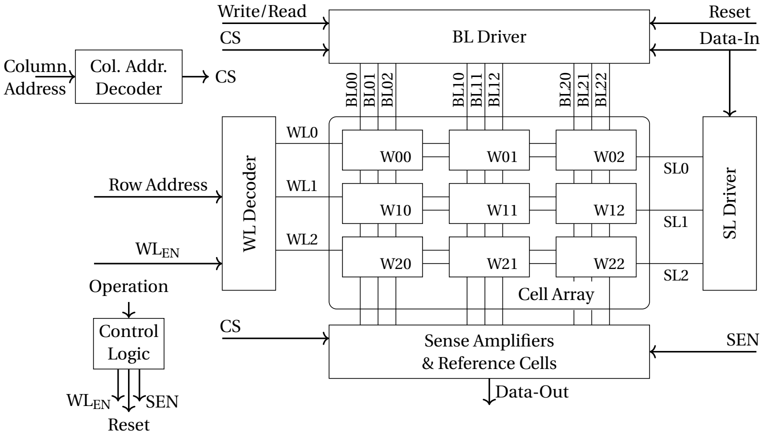


Figure 2.7: RRAM architecture

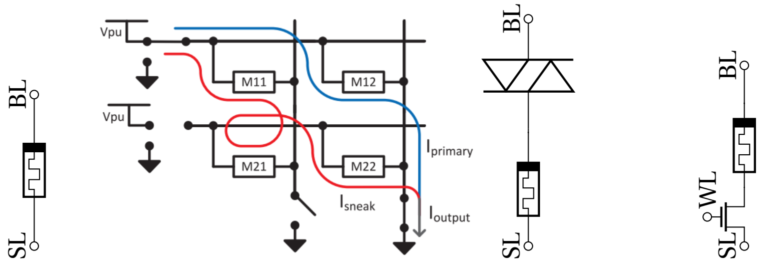
2.3.1. OVERVIEW

Fig. 2.7 shows an example of a RRAM architecture. It consists of several components and several signals that connect them together. The data are stored in *cells* in the *cell array*. Every cell stores one bit of information. Three cells together form a *word*, indicated as Wxx with $x \in \{0,1,2\}$. The words on a single row all share a *word line* (WL) and a *select line* (SL). In every column, every cell shares a *bit line* (BL). A word can uniquely be addressed by setting these lines to appropriate values. This is done by the *WL Decoder*, the *BL Driver*, and the *SL Driver*. The *Sense Amplifiers* (SAs) and the *Reference Cells* are used to read the values that are stored in the cells. The *Col. Address Decoder* decodes a column address to a *column select* (CS) signal that is used to select individual columns. Finally, the *Control Logic* provides, based on the selected operation, i.e., *Write/Read*, the timing for the internal signals: *word line enable* (WL_{EN}), the *Reset* signal, and *sense amplifier enable* (SEN). Next, we explain the details of the cell array, the decoders, the BL driver, the SL driver, the SAs, and the reference cells.

2.3.2. CELL ARRAY

The cell array contains all the cells that store data. For RRAMs, there are a few commonly used cell designs: one RRAM device (1R), one diode and one RRAM device (1D1R), and one transistor and one RRAM device (1T1R). This work employs 1T1R cells, but the other cell designs are explained next as well.

1R Cell Fig. 2.8a shows a 1R cell. It consists only of a RRAM device that can be accessed via the BL and the SL. When BL is high and SL low, the device will be SET, while the device will be RESET when BL is low and the SL is high. This cell design has the benefit that it can be manufactured densely in a so-called crossbar structure. In this structure, the BLs and SLs are manufactured perpendicular to each other and on every junction a RRAM device is placed, as is shown in Fig. 2.8b. Unfortunately, this architecture suffers from a major drawback, sneak paths. When cell M12 in the figure is accessed via the BL and the



(a) 1R (b) Sneak Paths [55] (c) 1D1R (d) 1T1R

Figure 2.8: RRAM cell designs and sneak paths.

SL, e.g., to read the cell, the corresponding BL is set to Vdd and the corresponding SL to GND, a current will flow through the cell, I_{primary}. However, there will also flow a current through the sneak path formed by the cells M11, M21, and M22, I_{sneak}. The total current will be higher than when only cell M12 would have been accessed. This limits the read margin and may lead to read errors.

1D1R Cell To reduce the sneak path severity, a bidirectional diode can be put in series with the RRAM device. The diodes introduce a voltage drop that limits the sneak path current. However, when writing a cell, also a higher voltage is required. This increases the power consumption of the RRAM and may even be infeasible for smaller technology nodes.

1T1R Cell Another solution to overcome the sneak path problem is to put a transistor in series with the RRAM device, as shown in Fig. 2.8d. This design has the benefit that one RRAM device can be selected individually. The drawbacks are: an additional WL is required to control the transistor; the cell density is lower due to the additional transistor; and there is a voltage drop over the transistor when the SL is high and the BL is low. The first two problems cannot be solved, but the voltage drop problem can partially be addressed by increasing the WL voltage in those cases.

Because the 1T1R cell offers the best control, this cell design is adapted in this thesis.

2.3.3. DECODERS

The Column Address and WL Decoders convert the address to the appropriate CS and WL signals, respectively. Fig. 2.9 shows an address decoder that decodes a four bit address m that consists of the bits A0, A1, A2, A3 to an internal signal line$m$A. For the Column Address Decoder, this is the CS signal. For the WL decoder, this signal is subsequently fed through an AND gate with the WL$_{\text{EN}}$ signal to the word line WLm. It is also fed to the SL named WL Input. The signal WL$_{\text{EN}}$ gives exact control over when the WL is enabled. This prevents, for example, selecting wrong WL lines when the address changes.

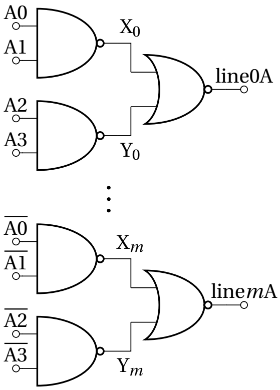


Figure 2.9: Address Decoder

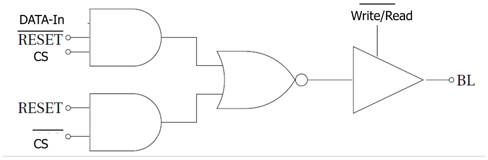


Figure 2.10: BL driver

2.3.4. BL DRIVERS

Fig. 2.10 shows a BL driver. When the Write/Read signal is low, a write operation is performed. This enables the tri-state buffer so that the BL can be written. When the Write/Read signal is high, the the tri-state buffer is disabled so that the BL Driver will not interfere with the read operation.

2.3.5. SL DRIVERS

Fig. 2.11 shows the SL Driver. It is always low, unless the Reset signal is high and the WL Input is set high by the WL driver.

2.3.6. SENSE AMPLIFIER

The Sense Amplifier senses the resistance of a cell and compares it to a reference cell. If the resistance is lower than that of the reference, it outputs ‘1’, if it is higher it outputs ‘0’. It is shown in Fig. 2.12. When the SEN signal is low, nodes A and B are precharged to Vdd. When a read operation is performed, the address is translated and the correct cells are enabled, and thus Cs is enabled as well. Then, the SEN signal is enabled to start the sensing operation. The two nodes will be discharged based on the resistance of the connected cell or reference cell. If the resistance of the cell is lower than that of the reference cell, node A will be discharged to GND, which recharges node B to Vdd. In the alternative case, node A is recharged to Vdd and node B is is discharged to GND. The value of the cell is now present at node B. When SEN is low again, nodes A nd B will both be precharged again.

Figure 2.11: SL driver

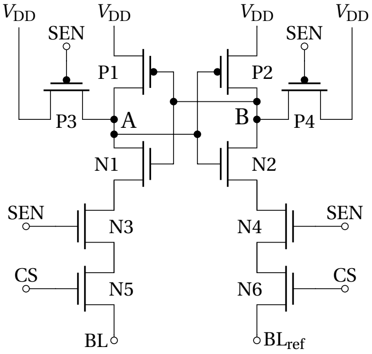


Figure 2.12: Sense Amplifier

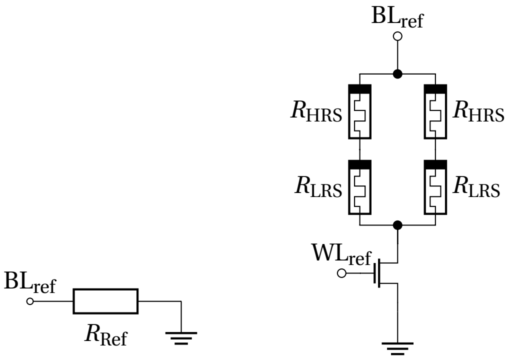
2.3.7. REFERENCE CELLS

Fig. 2.13 shows two reference cell designs that can be used with the previously discussed SA. The cell in Fig. 2.13a uses a simple resistor with a resistance set between R_{LRS} and R_{HRS}. Although this design is simple, it is also rather large when produced on the wafer. Furthermore, it cannot be adapted to, for example, process variations. To overcome these issues, the cell in Fig. 2.13b can be used. The cell uses two RRAM device in series in parallel with two other RRAM devices in series. In one link, one RRAM device is set to R_{LRS} and the other one to R_{HRS}. This sets the equivalent cell resistance exactly in the middle of these two values. The benefit of this design is that the cell is smaller than an on-chip resistor, and that process variations will have the same impact on the reference cell as on the actual memory cells.

2.4. COMPUTATION-IN-MEMORY

2.4.1. CONCEPT AND CLASSIFICATION

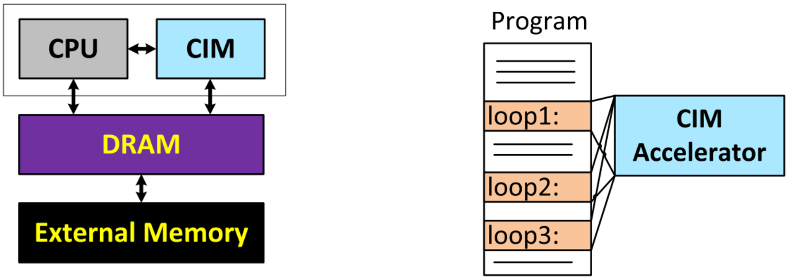
The CIM architecture is based on integrating the processing units and the storage in the same physical memory location. A realistic implementation that many researchers are prototyping is shown in Fig. 2.14a [56]–[58]; the CIM core may consist of very dense memristive crossbar array and CMOS peripheral circuitry. The CIM die takes over the memory-intensive computation parts from the processor, thus significantly speeding up the execution and reducing the energy consumption by eliminating large amounts of data transfers. Fig. 2.14b illustrates a program that could be executed efficiently on a CIM architecture; multiple loops can be executed within the CIM core while the other parts of the program can be executed on the conventional core. Each time a loop is invoked, the CPU sends a macro-instruction to the CIM core which decodes and executes it locally, and returns the final results.



(a) Reference using a resistor

(b) Reference cell composed of RRAM devices

Figure 2.13: Reference cells



(a) CIM core

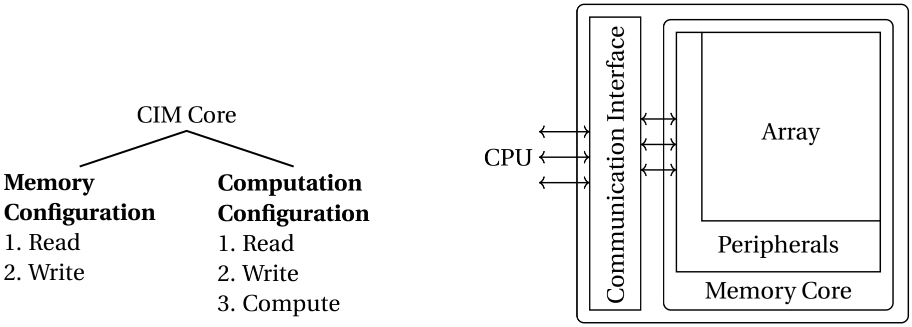
(b) CIM program

Figure 2.14: CIM as an accelerator [56]

As already mentioned, computing in the CIM core takes actually place within the memory. Hence, the CIM core can operate in two different configurations: *memory* and *computation* configuration. Fig. 2.15a shows these configurations, as well as the operations that each configuration requires. Since the computation configuration also uses read and write operations, it is a superset of the memory configuration. Fig. 2.15b shows a block diagram of a CIM die. In addition to the memory core, it consists also of a communication interface. It is worth noting that computations in the CIM core take place *within* the memory core. Because a memory core consists of a *memory array* and *peripheral circuits*, and depending on *where* the result of the computation is produced, CIM architectures can be divided into two classes [59]:

- *CIM-Array (CIM-A):* In CIM-A, the computation result is produced within the array. Examples of such architectures are PLiM [60], ReVAMP [61], CIM device [62], etc. The CIM-A core typically requires a significant redesign of the memory array to support computing, as conventional memory cell layouts are typically optimized for storage functionality only.

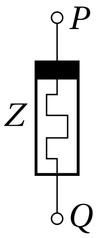
- *CIM-Periphery (CIM-P):* In CIM-P, the computation result is produced within the



(a) CIM configurations

(b) CIM block diagram

Figure 2.15: CIM configurations and block diagram



P	Q	Z	Z_{new}
0	0	0	0
0	1	0	0
1	0	0	1
1	1	0	0
0	0	1	1
0	1	1	0
1	0	1	1
1	1	1	1

Figure 2.16: Majority logic

peripheral circuitry. Examples of such architectures are PRIME [63], Pinatubo [64], CIM-Accelerator [65], etc. This architecture focuses on special circuits in the peripherals to realize, for example, bit-wise logic operations [64], [66], matrix-vector multiplication [67], [68], etc. Even though the computational results are produced in the peripheral circuits, the memory array could be a significant component in the computations. For example, to perform bit wise logic operations, multiple rows in the array need to be simultaneously activated.

As CIM performs operations within the memory core, at least part of the operands should be stored in the memory array. In other words, the operator being executed within the memory needs to have *all operands* stored in the array (as *resistive*) or *only* part of the operands is stored in the array and the other part is received via the memory port(s) (hence their logic values are *hybrid*, i.e., resistive and voltage). This results in four sub-classes: CIM-Ar, CIM-Ah, CIM-Pr and CIM-Ph; the additional letters ‘r’ and ‘h’ denote the nature of the inputs (operands), namely resistive and hybrid, respectively. An example of CIM-Ah and CIM-Pr will be discussed next.

CIM-Ah: Majority Logic The majority logic gate [69] shown in Fig. 2.16 is an implementation example of the CIM-Ah class using a memristive device Z. It has three inputs: P and Q supplied as voltages from the *peripherals*, and Z stored in the *array*; The output

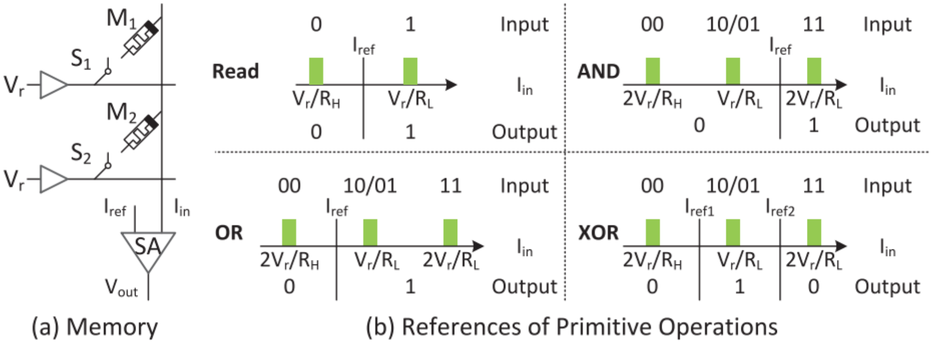


Figure 2.17: Scouting logic operations [66]

Z_{new} is produced after a majority operation is performed. The output is '1' if the majority of the inputs P, $\overline{Q}$, and Z are '1', as described in the truth table. Here $\overline{Q}$ denotes the negation of Q. The state of Z can only change to another state for a limited amount of input combinations P and Q, as shown in the truth table of the figure. This function can be used to develop other logic functions, like imply or inversion.

CIM-Pr: Scouting Logic Scouting logic [66], used as the CIM architecture discussed in this work, is an implementation example of CIM-Pr. It executes bit-wise logic OR, AND, and XOR within the memory core (e.g., RRAM). Unlike a standard memory where a single bit-cell (or word) in a column is read at a time, Scouting logic simultaneously selects two bit-cells (or words) in a column (i.e., two operands) on which the operations are performed. Fig. 2.17 illustrates the concept on two bit cells, M1 and M2 [66]. To perform a memory read operation, a suitable voltage V_r is applied to the input terminal of selected bit (e.g, M1) and the current I_{in} representing the resistance value of the selected bit cell is sensed by the SA. To perform an OR operation on the two bits (stored in M1 and M2), both devices are selected while applying V_r to the input terminal of each device. The resulting current, I_{in} which depends on the the equivalent parallel resistances of M1 and M2, is sensed by the SA. By comparing this current with an appropriate reference current I_{ref}, the result value is generated. In a similar manner, by changing I_{ref} of the SA, different gates such as AND and XOR can be realized.

2.4.2. ARCHITECTURE OVERVIEW

In this thesis, we use Scouting logic [66] as the CIM architecture. Fig. 2.18 shows the architecture for this CIM implementation. The architecture is a modification of the regular RRAM architecture that was presented in the previous section and is depicted in Fig. 2.7. Because Scouting logic is CIM-Pr, some changes to the periphery need to be made to implement it. However, no changes to the cell array are needed. The peripheral changes are: enabling the WL decoders to select multiple WLs, and modification and addition of the reference cells. All other peripheral circuits remain unchanged. Next, we will discuss the modifications of these components.

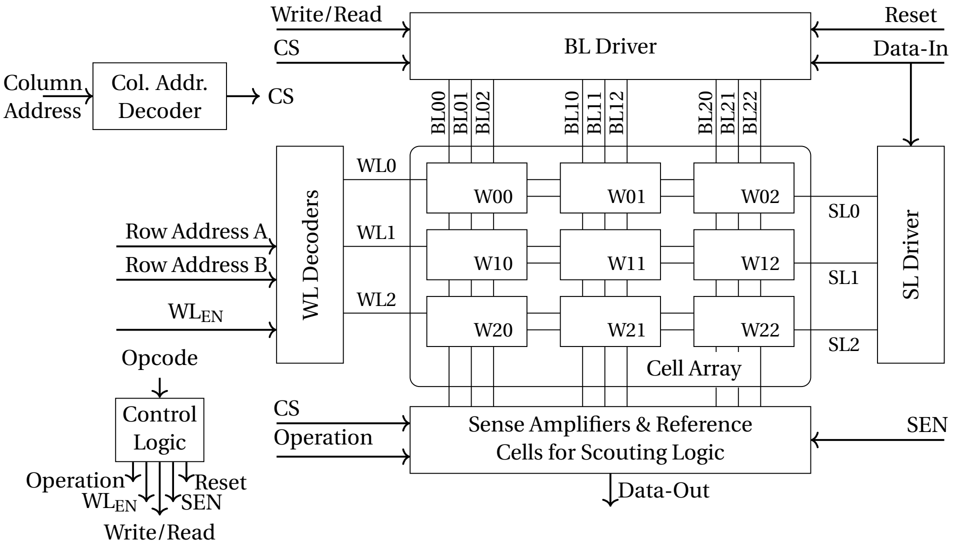


Figure 2.18: CIM architecture for Scouting logic

2.4.3. WL DECODERS

For Scouting logic, two rows need to be selected at the same time. This requires that a second row decoder needs to be added to the circuit. This allows to select any arbitrary combination of two rows to operate on. Fig. 2.19 shows the modified WL decoders. It consists of two row decoders, similar to the one in the regular RRAM array, shown in Fig. 2.9. The output signals are fed through an OR-gate. The result is subsequently through an AND gate with the WL_{EN} signal.

2.4.4. REFERENCE CELLS

Scouting logic requires two references, one for the OR operation and one for the AND operation. Fig. 2.20 schematically shows the resistance of the these references relative to the equivalent cell resistance. From the figure it follows that the reference for the AND operation (R_{AND}) is closer to the equivalent cell resistances than the reference for the OR operation (R_{OR}). Note that $R_{read} = R_{OR}$, as this resistance is both above ‘1’ and below ‘00’. Fig. 2.21 shows how the two references are implement using RRAM devices. Again, this reference cell design has the benefit to be smaller than an on-chip resistor, and the cells will suffer in a similar way from fabrication variations as the memory cells.

2.5. EXISTING RRAMS

This section presents an overview of existing RRAM chips. We focus both on prototypes by companies and research institutes, as well as on commercially available RRAMs.

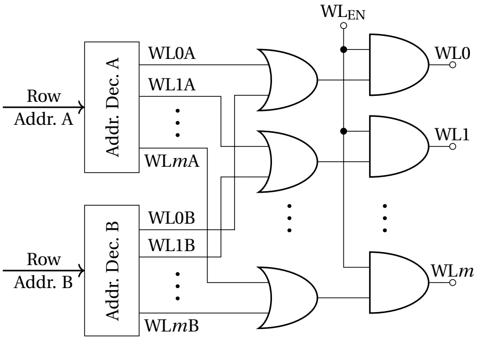


Figure 2.19: Scouting logic WL decoders

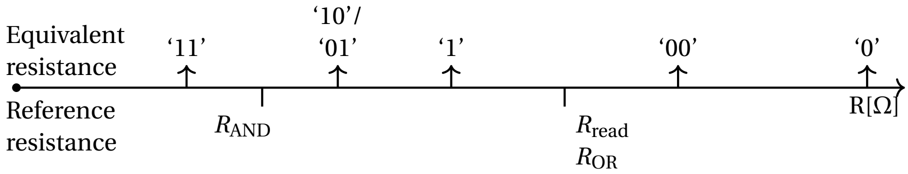


Figure 2.20: Resistances for Scouting logic

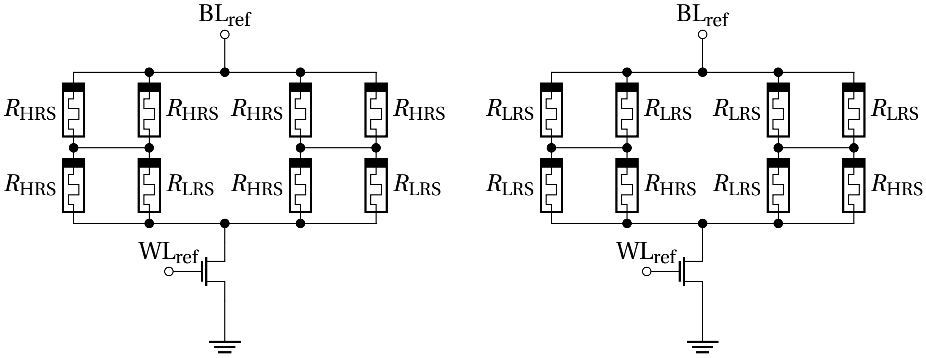
2.5.1. PROTOTYPES

Table 2.3 lists all RRAM prototypes that are published to date. If available, the table lists the year of publication, the company or institute that developed it, the process node in which the RRAM is made, the oxide (oxide RRAM) or electrolyte (CBRAM) that is used, the capacity of the memory, the write and read speed, the write and read latency, the cell architecture, the ratio $R_{\text{HRS}}/R_{\text{LRS}}$, and the endurance of the cell, i.e., how many cycles the cell can be written. The table shows that both companies an research institutes have an interest in developing RRAM. Next, we discuss some interesting observations from this table.

Cell selectors Almost all prototypes apply some kind of selector device, i.e., 1T1R, 1D1R, etc. cells. Interestingly enough, some cells introduce FinFET or FDSOI transistors in the process. This shows that RRAM is also suitable with these newer technologies.

Memory capacity The table shows that the peak memory capacity was achieved in 2013, where a prototype of 32 GB was presented using 24 nm technology. After this, only smaller memories have been presented. This may be caused by the fact that larger memories have a limited cycle endurance or to the relative low write and read speed of the memory. Alternatively, the manufacturing of CBRAM might be difficult than oxide RRAM.

Process node Over time, the prototypes make use of smaller feature sizes. This again shows that RRAM is compatible with modern technologies and is a useful technology in



(a) Reference cell for OR

(b) Reference cell for AND

Figure 2.21: Reference cells for Scouting logic CIM

future chips.

$R_{\text{HRS}}/R_{\text{LRS}}$ The ratio of $R_{\text{HRS}}/R_{\text{LRS}}$ is somewhat decreasing over time. This might show that the sensing quality improves. This means that the chips become more reliable.

Endurance The table shows that the endurance of the cells was generally higher for older technology nodes, while it is constantly lower for more modern technology nodes. In combination with the $R_{\text{HRS}}/R_{\text{LRS}}$ observations, one might conclude that modern RRAMs are more susceptible to write failures, which limits the endurance of these devices.

Faster read than write Finally, the table shows that read operations can generally be performed much faster than write operations. This is due to the fact that for a read operation only a short sensing operation is needed, while a write operation requires to grow or disrupt the CF.

2.5.2. COMMERCIAL RRAMS

Commercial RRAMs are less present than prototypes. They can be divided into two categories: RRAM used as an external memory for embedded systems, or RRAM as intellectual property (IP) that can be integrated in custom chips.

Embedded RRAM Panasonic [98] and Fujitsu [99] provide RRAM chips for use in embedded systems. The memory capacity for these devices is up to 62 kB, and up to 12 Mbit, respectively. The main features of these memories are the low power consumption of RRAM due to its non-volatility.

RRAM IP Dialog Semiconductor [100] and TSMC [101] provide RRAM as IP that can be included in custom designed chips. Dialog offers CBRAM technology, while TSMC offers oxide RRAM technology.

Table 2.3: RRAM prototypes.

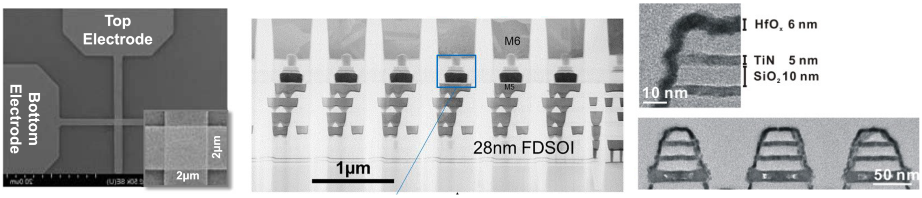
Year	Company/Institute	Reference	Process Node	Oxide/Electrolyte	Capacity	Write speed	Read speed	Write latency SET/RESET	Read latency	Cell	$R_{\text{HRS}}/R_{\text{LRS}}$	Endurance
2007	Qimonda	[70]	90 nm	Ag, CBRAM	2 MB			9 ns	50 ns	1T1R	$1.00 \cdot 10^7$	$1.00 \cdot 10^7$ cycles
2009	NTHU	[71]	90 nm	TiN/TiON				100 ns/10 µs		1T1R	36	$1.00 \cdot 10^6$ cycles
2009	National Central University/ITRI	[72]	180 nm	HfO2	1 kB			5 ns	8.5 ns	1T1R	$1.00 \cdot 10^3$	$1.00 \cdot 10^8$ cycles
2010	Unity Semiconductor	[73]	130 nm		64 MB		100 MB/s		100 µs	1R		
2011	ITRI/NTHU	[74]	180 nm	HfO2	4 MB			8 ns	8 ns	1T1R	100	
2011	Sony	[75]	180 nm	CuTe, CBRAM	4 MB	216 MB/s	2.3 GB/s			1T1R	100	
2011	Hynix	[76]	56 nm	TiOx/Al2O3	256 kB	100 MB/s				1T1R	100	
2012	Panasonic	[77]	180 nm	TaOx	8 MB	443 MB/s			25 ns	1T1R	150	
2012	Hynix	[78]	54 nm	Ta2O5/TiOx	2 MB			10 ns		1R	10	
2012	NTHU	[79]	28 nm	TiON				500 ns/100 µs		1T1R		$1.00 \cdot 10^6$ cycles
2012	NTHU	[80]	65 nm	TiON	4 MB				45 ns	1T1R	10	$1.00 \cdot 10^8$ cycles
2013	NTHU/TSMC	[81]	180 nm	TaOx	512kB					1T1R		$1.00 \cdot 10^7$ cycles
2013	Adesto	[82]	130 nm	CBRAM	1 MB			250 ns	20 ns	1T1R		$1.00 \cdot 10^5$ cycles
2013	Sandisk/Toshiba	[83]	24 nm		32 GB	8.4 MB/s	48.8 MB/s	230 µs	40 µs	1D1R		
2014	Micron/Sony	[84]	27 nm	Cu, CBRAM	16 GB	200 MB/s	1 GB/s			1T1R		$1.00 \cdot 10^5$ cycles
2014	NTHU/TSMC	[85]	28 nm	TiON	1 MB			500 ns/ 100 µs	6.8 ns	1T1R		
2015	NTHU/TSMC	[86]	16 nm	HfO2	1 kB					1FinFET1R	>5	$1.00 \cdot 10^5$ cycles
2015	Crossbar Inc	[87]								1R		$1.00 \cdot 10^6$ cycles
2015	Renesas	[88]	90 nm	Ta2O5/Ru							1.5	$1.00 \cdot 10^6$ cycles
2017	Chinese Academy of Sciences	[89]	5 nm	TaOx	10 MB			100 ns	300 ns		100	$1.00 \cdot 10^7$ cycles
2017	Tsinghua University	[90]	130 nm	HfOx	16 MB					1T1R		$1.00 \cdot 10^6$ cycles
2017	Sony	[91]	180 nm	Cu, CBRAM	4 MB	1.2 MB/s		100 ns		2T 1S1R	100	$1.00 \cdot 10^7$ cycles
2018	TSMC	[92]	40 nm	HfOx	11 MB				9 ns	1T1R		
2019	Intel	[93]	22 nm		3.6 MB			10 µs	5 ns	1FinFET1R		
2020	TSMC	[94]	22 nm		13.5 MB			6.5 ns/10 ns		1T1R/4T3R?		$1.00 \cdot 10^4$ cycles
2020	Chinese Academy of Sciences	[95]	28 nm		1.5 MB				3.3 ns	1T2R	50	$1.00 \cdot 10^5$ cycles
2021	Chinese Academy of Sciences	[96]	14 nm		1 MB				9.5 ns	1FinFET1R	10	$1.00 \cdot 10^5$ cycles
2021	CEA-LETI/UGA	[97]	28 nm	HfO2	16 kB			1 µs		1T(FDSOI)1R	10	$1.00 \cdot 10^5$ cycles

3

RRAM STRUCTURE, PRODUCTION PROCESS, DEFECTS, AND RELIABILITY

The production of RRAM devices requires a change of of the regular CMOS manufacturing process. This is due to the fact that RRAM devices are typically not fabricated directly on the silicon wafer where the transistors are placed, but rather on top of it between the metal layers or even in a dedicated stack structure. These differences introduce new defects that are not seen in regular CMOS production. Hence, this chapter discusses the production process of RRAMs and the defects that may occur there. First, we give a detailed overview of the RRAM device placement and its structure. Second, we give an overview of the RRAM production process that consists of the front-end-of-line process, the back-end-of-line, and the filament forming. Next, we discuss these sub-processes and the defects that may occur there. Finally, we discuss RRAM reliability and how defects affect this.

This chapter is partially based on [27]–[29].



(a) Cross structure [102]. (b) Between metal layers [103]. (c) Dedicated 3D structure [104].

Figure 3.1: RRAM device placement structures.

3.1. STRUCTURE OF A RRAM DEVICE

This section describes the structure of a RRAM device in more detail. In this dissertation, we focus mainly on oxide-based RRAMs (OxRAMs).

3.1.1. PLACEMENT

Typically, RRAM devices are manufactured on top of the transistors on the wafer. The three possibilities for the placement of the devices are depicted in Fig. 3.1. They are: 1) cross structure (Fig. 3.1a [102]), 2) between the metal layers (Fig. 3.1b [103]), and 3) via a dedicated 3D structure (Fig. 3.1c [104]). Next, we will discuss these in more detail.

CROSS STRUCTURE

The cross structure is typically used for small RRAMs that are used in research. The production process is simpler than for RRAM devices manufactured between metal layers or dedicated 3D structures. The purpose of these RRAMs is, for example, to study new materials [102], or specific switching behaviors [105], [106]. The cross structure allows to connect pads directly to the top and bottom electrode that in turn can be connected to probes of measuring equipment.

BETWEEN METAL LAYERS

RRAM devices that are fabricated directly between the metal layers allow for denser RRAMs than the cross structure. Furthermore, it is easier to combine a single RRAM device with a transistor on the underlying wafer to obtain a 1T1R cell. Hence, this placement possibility results in reliable dense RRAM structures [103], [107].

DEDICATED 3D STRUCTURE

The dedicated 3D structure allows for even denser memories than RRAM devices that are fabricated between metal layers. However, the production process deviates more from a regular CMOS process [104], [108]. This makes it more difficult to manufacture these devices.

The remainder of this thesis focuses on vertically stacked RRAM cells, i.e., on RRAM devices in cross structures and manufactured between metal layers, because this type of structure is seen more frequently in literature than dedicated cross structures, which makes the work more general applicable.

3.1.2. STACKED STRUCTURE

Fig. 3.1b shows an example of a cross section of a RRAM device that is fabricated between the metal layers. The materials that are used in the stack determine the switching behavior of the device, i.e., whether it is bipolar, complementary, or unipolar switching. Next, we discuss the structural details of the oxide, the electrodes, and the capping layer for all three switching modes that were explained in Section 2.1.

BIPOLAR SWITCHING

For bipolar switching, it is required that there exists an asymmetric oxygen binding profile in the RRAM device, i.e., one side of the RRAM should be able to attract more oxygen vacancies than the other side [40], [109], [110]. This can be achieved by using one noble material as an electrode and one reactive metal, or by introducing a capping layer [109]. In the former case, the oxygen ions are bound by the reactive metal, while in the latter case, the ions are bound by the capping layer. Examples of structures that use one noble and one reactive electrode are (BE/oxide/TE): Pt/HfO_2/TiN [40], Pt/TiO_2/TiN [111]. Examples of structures that use a capping layer are (BE/oxide/capping layer/TE): Pt/HfO_2/Hf/Pt [112], TiN/HfO_2 /Ti/TiN [113], TiN/HfO_2/Hf/TiN [114].

COMPLEMENTARY SWITCHING

Complementary switching also occurs in asymmetric oxygen binding profile stacks similar to bipolar stacks. However, for complementary switching to occur, the capacity of the device to bind the free oxygen ions should be limited. This causes the oxygen vacancies to shift, rather than the oxygen ions [40], [41]. For example, the capping layer should be thin, or the CF should be wide so that many oxygen ions are present in the RRAM device. It is shown that complementary and bipolar switching can exist in the same devices. An example that uses one noble and one reactive electrode is (BE/oxide/TE): Pt/HfO_2/TiN [40]. Examples of structures (BE/oxide/capping layer/TE) that use a capping layer are: TiN/HfO_x/Ti/TiN [115].

UNIPOLAR SWITCHING

Unipolar switching typically occurs in devices that have a symmetric oxygen binding profile [34], [109]. This can be achieved by using two noble metals as electrodes and not including a capping layer. In these devices, the oxygen ions can move to any of the electrodes, depending on the voltage polarity. Examples of structures that use two noble materials for both electrodes are (BE/oxide/TE): Pt/TiO_2/Pt [37], Ag/TiO_2/Ag [116].

3.2. OVERVIEW OF RRAM PRODUCTION PROCESS

A generalized RRAM production flow for stacked devices can be split into three steps: front-end-of-line (FEOL), back-end-of-line (BEOL), and forming of the CF [113], [114], [117]. The manufacturing flow is schematically depicted in Fig. 3.2 [113], [114], [117]. The process starts with the FEOL phase in which transistors are fabricated on the wafer. This step is the same as the conventional CMOS production flow. In the following BEOL phase, the metal layers are fabricated. Somewhere between these layers the memristive devices are placed, e.g., between M4 and M5 [113]. After the manufacturing process is completed, an initial CF needs to be formed in the finalizing forming step.

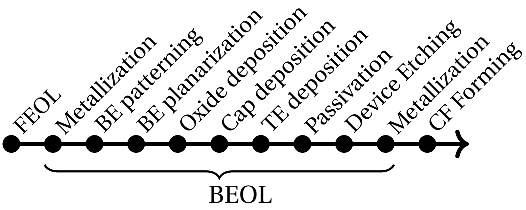


Figure 3.2: RRAM manufacturing process [113], [114], [117].

Table 3.1: RRAM defect classification.

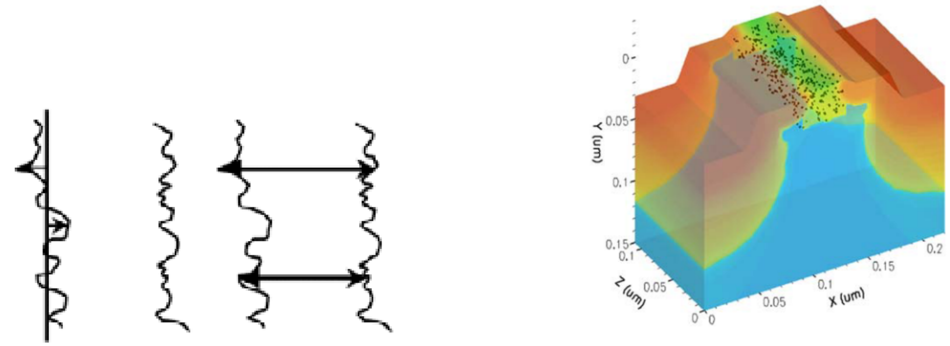
FEOL	BEOL	
Transistor	Interconnection	RRAM Device
Patterning proximity	Opens	Electrode roughness
Line roughness	Shorts	Polish variations
Polish variations	Line roughness	Varying defect density
Anneal	Irregular shapes	Dimensional variations
Strain	Big bubbles	Lower capping layer binding capacity
Gate granularity	Small particles	Material redeposition
Dielectric variations		Over-forming
		Under-forming

The defects that can occur during the RRAM production can be classified into transistor, interconnection, and RRAM device defects. Table 3.1 lists the RRAM defects according to this classification. The following sections describe the production process and the potential defects in more detail.

3.3. FRONT-END-OF-LINE

During the FEOL production phase, transistors are fabricated on the wafer. The production process consists of several steps that are repeated to make the different structures of the transistor. First, a silicon oxide is grown, or a material, e.g., an other oxide for the gate, or metal for the interconnections, is deposited. This layer covers the complete wafer. Next, a photoresistive layer is applied over the wafer. Subsequently, lithography is used to pattern the photoresistive layer. After this step, either the parts of the photoresistive layer that have been patterned, or those that have not been patterned, are removed. This gives access to the underlying material. Now, this underlying layer can be etched or the material can be doped by injecting ions in to modify the material properties. These steps are repeated for the different parts of the transistor fabrication.

During the transistor manufacturing, several defects can occur. Kuhn, Giles, Becher, *et al.* divide transistor defects into two categories: historical and emerging defects [119]. Historical defects include [118]: *patterning proximity effects*, where parts outside of the pattern, i.e., that should not receive any photons, also receive photons; *line-edge* and *line-width roughness*, where the edge of a pattern is not smooth but rough due to photon



(a) Line edge roughness (left) and line width roughness (right) [118]

(b) Random dopants in a transistor [119].

Figure 3.3: FEOL defects

variations, as shown in Fig. 3.3a; *polish variations* that affect the thickness of the polished layers; and *variations in the gate dielectric* due to defects in the gate oxide material.

Emerging defects are observed in smaller technology nodes. They are [119]: *random dopant* and *anneal* fluctuations, where small deviations in the doping have a significant effect on the transistor's performance due to the small number of dopants in the device, as shown in Fig. 3.3b; *strains*, where small structures put mechanical stress on other parts of the wafer; and *gate material granularity*, where the gate is not a uniform structure but a varying one.

3.4. BACK-END-OF-LINE

The next production phase is the BEOL where the metal layers as well as the RRAM devices are fabricated. All layers a manufactured using the same steps of deposition and lithography that were used during the FEOL. Metallization of the lower layers again is equal to that of a standard production process, and thus the same defects may occur here. For example, misalignment or *small particles* may lead to poor connections that increase the resistance of a wire. Lithographic issues such as *line-edge roughness* may attribute to the formation of *irregular shapes* and affect the wire resistance and capacitance [120], which in turn reduces the RRAM performance [121], [122]. For example, higher wire resistance will reduce the voltage over the RRAM device, and thus limit its switching capabilities, while increased capacitance will lead to cross-talk.

After the lower metal layers are deposited, the RRAM device can be constructed, as shown in Fig. 3.1b. This step starts with the deposition and patterning of the BE that connects to the underlying metal layer. The deposition process leaves a *rough surface* on the BE. Due to this roughness, there will be many surface defects between the BE and the RRAM oxide that increase the variability as well as the probability of a hard oxide breakdown [124], [125]. Therefore, the BE is planarized by a chemical-mechanical polishing step to reduce the electrode roughness [123], [125]. The results of this are shown in Fig. 3.4. The polishing step also needs to be well-controlled, in order to prevent the generation of *polish variations*.

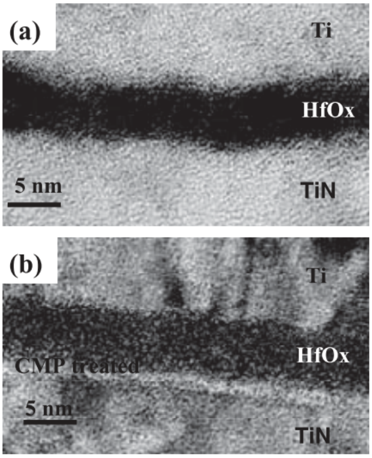


Figure 3.4: Unpolished (A) and polished (B) electrodes [123]

After the completion of the BE, the oxide can be deposited on top of it. To minimize device variations, the *thickness of the oxide layer* as well as the *amount of defects in the oxide and at the oxide interface* need to be controlled. For these reasons, the oxide is typically deposited by an atomic layer deposition process rather than physical vapor deposition process, since the former process provides a tighter control [50], [117]. The deposited oxide can have a (poly-) crystalline or an amorphous structure. The edges of a crystalline structure are called grain boundaries (GBs) and contain a higher number of defects than the inside of the crystals. Therefore, CFs tend to form along these edges [126], [127]. Fig. 3.5 shows that the current through an HfO_2 oxide flows mainly along the GBs [126]. Because the dimensions of a crystal can vary, the number of poly-crystals in an oxide will vary, and hence, the crystalline oxides will have a wider resistance distribution than amorphous oxides [128]. However, the amorphous oxides have a smaller HRS/LRS ratio than crystalline structures which makes it more difficult to distinguish the two states [128].

Next, the capping layer is fabricated. If no capping layer is desired, this step will be skipped and the production process continues with the TE. The thickness of the capping layer is less of importance than the thickness of the oxide, as the capping layer will conduct significantly better than the oxide. Therefore, less precise depositing methods such as physical vapor deposition can be used [114]. However, the oxygen binding capabilities of the capping layer play an important role. Hence, the capping layer should be thick enough and doped enough so that all oxygen ions can bind that come free when forming the CF [129]. The defect that may occur here is *low capping layer binding capacity*. Subsequently, the TE is deposited in a similar way that the BE was deposited, although the polishing step is not always required here.

The stack of materials is then etched to separate the individual memristive devices. Lithographic issues may again lead to *variations of the device dimensions*, affecting the performance of the finalized device [132]. Etching may lead to *redeposition of material*

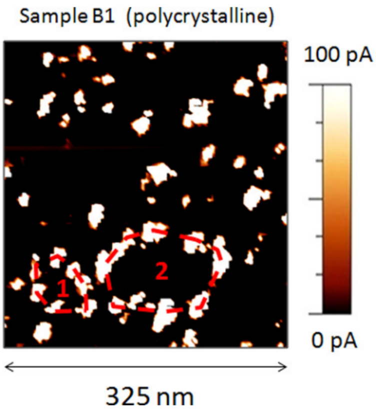
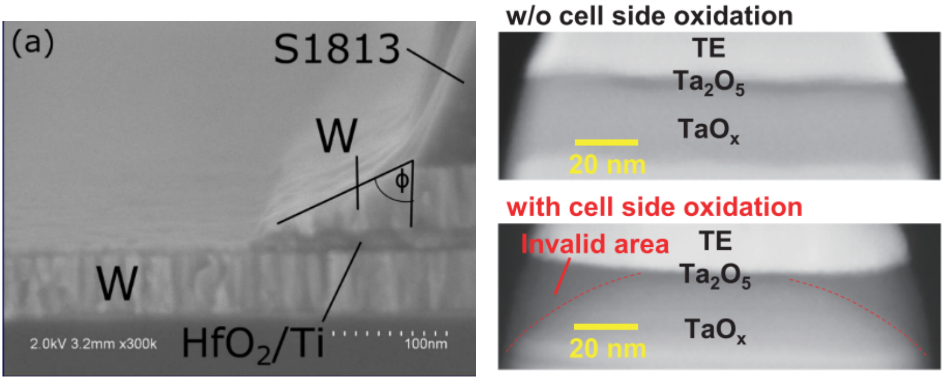


Figure 3.5: Grain boundaries conduct current through the oxide [126]



(a) Redeposition after etching [130]

(b) Partial invalidation of sidewall [131]

Figure 3.6: Sidewall redeposition and mitigation.

along the sidewalls of the RRAM device [130], [133], as shown in Fig. 3.6a. This forms a parasitic leakage path that lowers the resistance. Furthermore, the forming voltage is lowered and more resistive variation is observed [130]. Redepositioning can be prevented by optimizing the production process so that the via connecting to the BE is smaller than and buried by the electrode, preventing redeposition of metallic species [133]. Sidewall leakage effects due to atomic redeposition can be further reduced by optimizing the etch processes so that the etching happens as vertically as possible [130]. Hayakawa, Himeno, Yasuhara, *et al.* have shown that the uniformity of cells suffering from sidewall oxygen ions can be improved by invalidating a strip around next to the sidewall [131], as shown in Fig. 3.6b. This invalidated area can then be covered with a protecting layer in order to minimize the impact of sidewall redeposition effects.

Finally, the memristive device is isolated from its surrounding structures and the production flow continues with the next metal layer, similar to the standard CMOS BEOL.

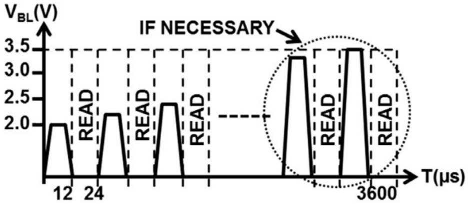


Figure 3.7: Controlled forming process using short pulses and verifying read operations [134]

3.5. Forming of Conductive Filament

In the final production step of an RRAM, a CF is formed in the oxide. The forming conditions have a strong impact on the CF shape. Typically, the forming process requires a voltage V_{form} that is higher than V_{set} or V_{reset} [135]. Higher V_{form} will lead to a quicker formation of the CF, but it may also lead to a complete breakdown of the oxide [136]. To prevent this, a forming scheme can be applied that measures the forming of the CF and adapts V_{form} accordingly [134], [137]. One of these schemes is shown in Fig. 3.7 [134]. It applies short pulses and verifies the resistance of the device after every pulse. If the desired resistance is not yet reached, another pulse is given with a slightly higher voltage.

Apart from V_{form}, the forming current (I_{form}) flowing through RRAM devices also needs to be taken into account. In general, higher forming currents will result in wider CF structures and hence in a lower overall resistance and less variation, while lower currents will lead to thinner CFs [42], [113], [135], [138], [139]. Furthermore, I_{form} needs to be kept as constant as possible, as I_{form} fluctuation will lead to variations in the device resistance later on [140]. In addition to the forming conditions, the device geometry also plays an important role. Smaller devices will have a higher resistance because the probability of forming a CF is lower, while larger devices have a higher forming probability [114], [136], [139].

Two defects may get introduced by the forming step in the RRAM. It is possible that the CF will barely or not at all form, and thus the memristive device will always remain in HRS. This is referred to as an *under-forming* defect. In contrast, the formation of the CF may also be too strong, for example due to a variable forming current. This will push the resistance of the device below its design specifications and render the device overformed, i.e., an *over-forming* defect [19], [141]. Even stronger formation may lead to a breakdown of the oxide, causing the device to be always stuck at LRS, unable to switch back to HRS [136]. Fig. 3.8 [105] shows a RRAM device in which the oxide is broken down.

3.6. Reliability of RRAMs

The reliability of a device is a metric that describes how well the device performs over time. This performance is related to production defects and test, because an undetected defect may result in low reliability of a device, which shortens its lifetime and usability. For RRAMs, there are two major reliability characteristics: cycle-to-cycle variations, and cycle endurance. These are discussed next.

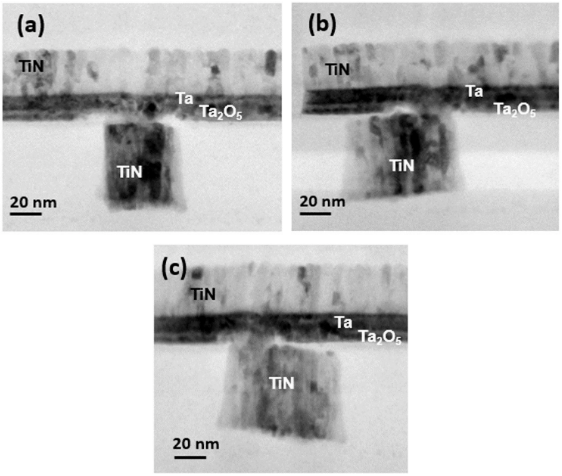


Figure 3.8: Hard breakdown of the oxide [105]

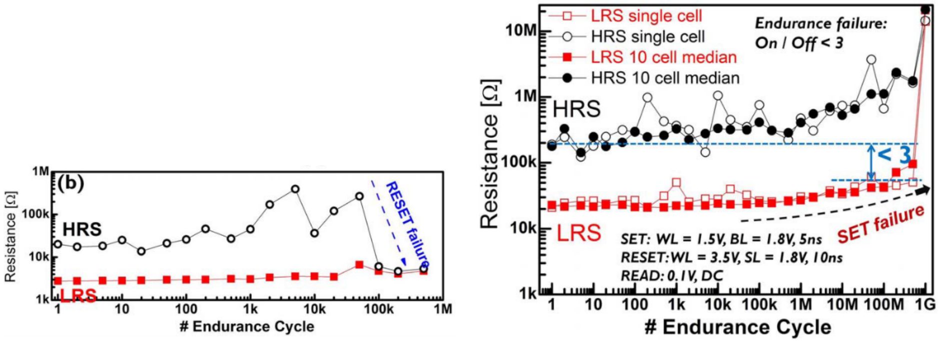
3.6.1. CYCLE-TO-CYCLE VARIATIONS

The growth and rupture of the CF is a stochastic process. Not every oxygen ion will move away in every SET cycle, nor will the same vacancy be filled in every RESET cycle. Because the CF changes every cycle, the resistance of the RRAM device changes per write cycle as well. This phenomenon is known as cycle-to-cycle variation. If these variations become too severe, the performance of a RRAM is impacted, e.g., a cell is not written to an appropriate value. Typically, the variation of a device in the LRS is lower than in the HRS [113]. This due to the fact that in LRS, the CF is wider and thus single ions have less impact on the overall resistance, while in HRS the opposite is true [113], [136].

Production process variations and defects can have an amplifying effect on the severity of the cycle-to-cycle variations. For example, the forming current has a strong impact on the initial CF. Hence, lower forming current will lead to smaller CFs, that are more susceptible to variations [113], [135]. Also polish variations can impact the variability, as these lead to defects at the oxide interface, that affect the growth and dissolution of the filament [125]. Temperature has a minor effect on cycle-to-cycle variations [142].

3.6.2. CYCLE ENDURANCE

The number of switching cycles that a RRAM device can withstand without failing, is called the cycle endurance. Ideally, this number should be as high as possible, to guarantee a long useful life of the RRAM. The growth and dissolution of the CF capabilities of a RRAM device deteriorate over time, leading to a RRAM device that is unable to switch and remains either in the LRS, or in HRS [125], [143]. The former is caused by the fact that with every SET operation the CF grows wider, which makes it harder to switch it back. Similarly, the latter is caused by a RESET operation that is too strong. Fig. 3.9 shows how the cycle endurance of a RRAM device is limited by SET or RESET failures [143].



(a) SET failure

(b) RESET failure

Figure 3.9: Endurance limitations leading to SET and RESET failures [143]

There are multiple defects that reduce the cycle endurance. For example, polishing variations can lead to SET failures [125]. Also, the crystalline structure of the RRAM device plays has an impact; it is found that poly-crystalline structures show better endurance characteristics than amorphous structures [128].

Besides optimizing the production process to achieve a maximal cycle endurance, it is also possible to add verification schemes into a design [144], [145]. These verification schemes measure the resistance of a RRAM cell after or during every write operation and adapt it if it is out of bounds. This can extend the cycle endurance significantly.

4

TRADITIONAL DEFECT MODELING, FAULT MODELING, AND TEST DEVELOPMENT

The aim of a test is to validate whether a manufactured chip matches the design. Traditional test development uses linear resistors to model defects that can occur during the manufacturing. These resistors are then added to a defect-free design and simulated to observe what faults are sensitized. Subsequently, a test solution, e.g., a March test or design-for-test schemes (DFT), is developed that will detect the sensitized faults. This chapter studies these three steps using the traditional defect modeling approach of using linear resistors.

This chapter is partially based on [27], [29].

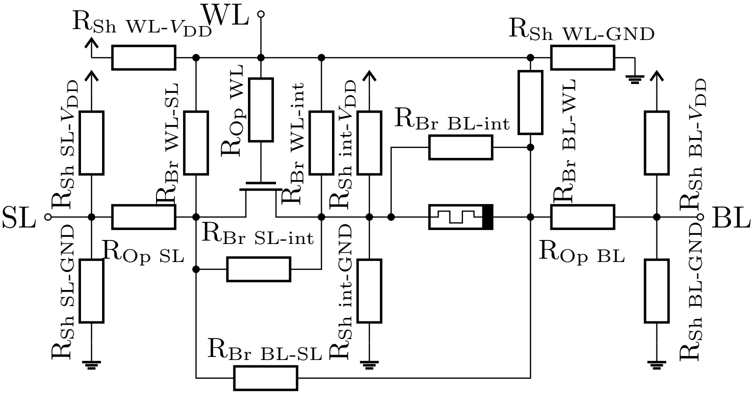


Figure 4.1: All possible shorts, bridges, and opens in a single 1T1R cell

4.1. Defect Modeling

To develop high-quality test, the defect that can occur in a circuit need to be understood and they need to properly modelled. The resulting defect model can then be used to determine how the defect affects the performance of the circuit. Traditionally, in SRAMs and DRAMs, spot defects in the interconnections are modeled as linear resistors [8], [9], [146]. A spot defect is an unwanted bridge connection between two nodes of the circuit, an unwanted short circuit connection between a node and $\mathrm{V_{DD}}$ or ground, or an open connection in a wire.

Several papers have studied spot defects in RRAM as well [10], [11], [20], [26]. However, none of these study all possible spot defects that can exist in a single cell. Fig. 4.1 shows all possible spot defects modeled as shorts ($\mathrm{R_{Sh}}$), bridges ($\mathrm{R_{Br}}$), and opens ($\mathrm{R_{Op}}$) that can occur in a single 1T1R cell. To illustrate, $\mathrm{R_{Sh\ int\text{-}GND}}$ indicates a short defect between the internal node and ground, and $\mathrm{R_{Op\ WL}}$ indicates an open in the word line of the cell. The resistance of each resistor determines the strength of the defects, e.g., a bridge with a lower resistance is more severe than a higher resistance. To develop a test, all these defect models must be injected in the circuit and simulated to see which faults are sensitized. The sensitized faults need to be analyzed and included in a test.

4.2. Fault Modeling and Analysis

A fault abstracts the behavior of a defect to the functional level of the memory [7]. That is, it describes the defect's behavior in terms of memory operations that *sensitize* the fault and the logic values that are stored in the cell after sensitization of the fault. To illustrate, when the resistance of defect $\mathrm{R_{Op\ WL}}$ in Fig. 4.1 is above a certain threshold value, i.e., $\mathrm{R_{Op\ WL}} > R_{\mathrm{thresh}}$, the cell cannot be accessed anymore. There may be multiple defects that increase the resistance so beyond this limit. A fault abstracts the resulting erroneous behavior of all these defects as a fault in which the cell cannot be accessed. This section first describes the fault modeling process. Here, all faults that can be sensitized in a RRAM are systematically defined. Second, the fault space needs to be analyzed to see

Figure 4.2: RRAM resistance range

which faults are actually sensitized and require a test to detect them.

4.2.1. FAULT MODELING

FAULT SPACE DEFINITION

Note that multiple operations may be needed to sensitize the fault, or that multiple cells can be affected by the fault. Hence, we can classify the faults based on the number of operations n that are needed to sensitize the fault, and on the number of cells $\#C$ that are involved in the fault [147]. For the number of operations involved, if $n \leq 1$, then the fault is static, if $n > 1$, then the fault is dynamic. For the number of cells involved, if $\#C = 1$, then the fault is a single-cell fault, if $\#C \geq 1$, then the fault is a multi-cell $\#C$-coupling fault. Note that a fault is always classified in both categories, e.g., there can exist static coupling faults, as well as dynamic single-cell faults. In the following sections we present the static and dynamic fault space for single-cell and multi-cell coupling faults.

SINGLE-CELL FAULTS

As mentioned in the previous section, a fault is described by its sensitizing sequence and the effect on the data storing cell. A fault is typically described as a fault primitive (FP) described by the following notation: ⟨S/F/R⟩ [27], [148] which extends the scheme in [147]. In this notation:

- S describes the sensitizing operation of the fault. It is denoted as $x_0 O_1 x_1 \ldots O_n x_n$, where $x \in \{0,1\}$ denotes the value that is stored in the cell, $O \in \{\mathrm{r},\mathrm{w}\}$ denotes the operation that is performed (i.e., read (r) or write (w)), and n denotes the amount of operations. Examples are: S = 0r0 where a read operation is performed on a cell containing a ‘0’ and the expected output is also ‘0’, and S = 1w0r0 where a value of ‘0’ is written to a cell that initially contains a ‘1’ and is subsequently read out.

- F describes the state that is stored in the cell after the sensitizing operation. For digital memories (e.g. SRAM), it holds that $\mathrm{F} \in \{0,1\}$. Because the RRAM device is an analog device, more states can be defined than just ‘1’ and ‘0’. This concept is shown schematically in Fig. 4.2. In this figure, 1 and 0 denote the resistance ranges for a logical ‘1’ and ‘0’, U denotes the undefined state, and L and H denote respectively an extremely low and an extremely high conductance state. In the ‘L’ state, the resistance of the RRAM device is higher than the HRS range, and, in ‘H’ the resistance of the cell is lower than the LRS range. Logically, the ‘L’ and ‘H’ state behave as a ‘0’ or ‘1’ respectively. The thresholds between these ranges depend on the circuit that is investigated. For RRAMs, F needs to be extended: $\mathrm{F} \in \{0,1,\mathrm{L},\mathrm{H},\mathrm{U}\}$.

- R describes the output after a read operation is performed, and $\mathrm{R} \in \{0,1,?,-\}$. Here, ‘−’ denotes no output when no read operation is involved, and ‘?’ denotes a random output value, i.e., sometimes ‘0’, sometimes ‘1’. It should be noted that ‘U’ is not equal to ‘?’ in R; ‘?’ defines the outcome of a read operation, while ‘U’ defines the resistance state of the cell.

Some faults have a similar behavior, e.g., ⟨1w0/1/-⟩ and ⟨0w1/0/-⟩ both describe a failed transitioning operation. The set of faults that describe the same faulty behavior is called a functional fault model (FFM). To be able to group all individual faults into functional fault models, we propose to name the faults according to their behavior.

If $n = 0$, then the fault is a state fault. These faults are denoted as FP = $\mathrm{S}\{ini\}\mathrm{F}\{fin\}$, where *ini* denotes the initial state of the cell, and *fin* describes the cell contents after the sensitizing operation is completed; it is equal to the F value in the ⟨S/F/R⟩ notation. To illustrate, the ⟨0/U/-⟩ fault is named S0FU. For $n = 1$, the FP are named according to the following scheme: FP = $\{out\}\{opn\}\{opd\}\{eff\}\mathrm{F}\{fin\}$. The fields in this scheme have the following meaning:

- *out* describes the behavior of a read operation (it is omitted in case of a write operation), i.e., $out \in \{\mathrm{i,r,d}\}$. Here, i denotes an incorrect read output, r a random read output, and d a deceptive read output. A deceptive read outputs the correct value but flips the cell’s contents, e.g., ⟨0r0/1/0⟩ is a deceptive read fault.
- *opn* denotes the operation that is performed and triggers the fault in the cell, i.e., $opn \in \{\mathrm{R,W}\}$. Here R denotes a read operation and W a write operation.
- *opd* denotes the operand of the operation, i.e., $opd \in \{0,1\}$.
- *eff* describes the fault effect, i.e., $eff \in \{\mathrm{T,D,N}\}$. Here, T means a transition operation (i.e., S = 1w0 or S = 0w1), D means that the operation is destructive (it changes the cell contents), and N is non-destructive (it does not affect the cell’s contents).
- *fin* describes the cell contents after the sensitizing operation is completed; it is equal to the F value in the ⟨S/F/R⟩ notation.

As an illustration, consider the following faults and their names. The ⟨0w1/0/-⟩ fault is named W1TF0, the ⟨0r0/U/?⟩ is named rR0DFU, and the ⟨1r1/1/0⟩ fault is named iR1NF1. Dynamic faults ($n \geq 2$) follow the same naming scheme but get an additional prefix $\{n\mathrm{d}-\}$. The name of the fault is based on the last operation in S, e.g., the ⟨0r0w1/L/-⟩ fault is named 2d-W1TFL.

With this information, we can define the complete fault space for single-cell faults. Table 4.1 lists all static single-cell faults and the corresponding FFMs that can occur in an RRAM. It shows that there exist more single-cell static faults in RRAM than in traditional digital memories [147]. This is due to the three additional states (‘U’, ‘L’, ‘H’) that a RRAM device may be in. Dynamic faults require more than one sensitizing operation, i.e. $n \geq 2$. The total number of sensitizing sequences (#S) for a certain n is described by:

$$\#\mathrm{S} = \sum_{i=0}^{n} 2 \cdot 3^{i} \,. \tag{4.1}$$

The correctness of this equation can be demonstrated by observing that for $i = 0$, there are only two possible S: S = 0, S = 1. Further, for every $i > 0$, a sensitizing sequence can

Table 4.1: Single-cell static fault primitives and functional fault models

#	S	F	R	Name	FFM	#	S	F	R	Name	FFM
1	0	L	–	S0FL	State fault	27	0r0	1	0	dR0DF1	Deceptive read destructive fault
2	0	U	–	S0FU	State fault	28	0r0	H	0	dR0DFH	Deceptive read destructive fault
3	0	1	–	S0F1	State fault	29	0r0	L	1	iR0DFL	Incorrect read destructive fault
4	0	H	–	S0FH	State fault	30	0r0	0	1	iR0NF0	Incorrect read fault
5	1	L	–	S1FL	State fault	31	0r0	U	1	iR0DFU	Incorrect read destructive fault
6	1	0	–	S1F0	State fault	32	0r0	1	1	iR0DF1	Incorrect read destructive fault
7	1	U	–	S1FU	State fault	33	0r0	H	1	iR0DFH	Incorrect read destructive fault
8	1	H	–	S1FH	State fault	34	0r0	L	?	rR0DFL	Random read destructive fault
9	0w0	L	–	W0DFL	Write destructive fault	35	0r0	0	?	rR0NF0	Random read fault
10	0w0	U	–	W0DFU	Write destructive fault	36	0r0	U	?	rR0DFU	Random read destructive fault
11	0w0	1	–	W0DF1	Write destructive fault	37	0r0	1	?	rR0DF1	Random read destructive fault
12	0w0	H	–	W0DFH	Write destructive fault	38	0r0	H	?	rR0DFH	Random read destructive fault
13	0w1	L	–	W1TFL	Write transition fault	39	1r1	L	0	iR1DFL	Incorrect read destructive fault
14	0w1	0	–	W1TF0	Write transition fault	40	1r1	0	0	iR1DF0	Incorrect read destructive fault
15	0w1	U	–	W1TFU	Write transition fault	41	1r1	U	0	iR1DFU	Incorrect read destructive fault
16	0w1	H	–	W1TFH	Write transition fault	42	1r1	1	0	iR1NF1	Incorrect read fault
17	1w0	L	–	W0TFL	Write transition fault	43	1r1	H	0	iR1DFH	Incorrect read destructive fault
18	1w0	U	–	W0TFU	Write transition fault	44	1r1	L	1	dR1DFL	Deceptive read destructive fault
19	1w0	1	–	W0TF1	Write transition fault	45	1r1	0	1	dR1DF0	Deceptive read destructive fault
20	1w0	H	–	W0TFH	Write transition fault	46	1r1	U	1	dR1DFU	Deceptive read destructive fault
21	1w1	L	–	W1DFL	Write destructive fault	47	1r1	H	1	dR1DFH	Deceptive read destructive fault
22	1w1	0	–	W1DF0	Write destructive fault	48	1r1	L	?	rR1DFL	Random read destructive fault
23	1w1	U	–	W1DFU	Write destructive fault	49	1r1	0	?	rR1DF0	Random read destructive fault
24	1w1	H	–	W1DFH	Write destructive fault	50	1r1	U	?	rR1DFU	Random read destructive fault
25	0r0	L	0	dR0DFL	Deceptive read destructive fault	51	1r1	1	?	rR1NF1	Random read fault
26	0r0	U	0	dR0DFU	Deceptive read destructive fault	52	1r1	H	?	rR1DFH	Random read destructive fault

only be extended by subsequently performing a w0, w1, or a rx with x the same logical value as in the previous operation. Because the total number of FPs grows exponentially with n, we omit listing them as was done for the static ones in Table 4.1, but the methodology to generate this table is similar.

MULTIPLE-CELLS COUPLING FAULTS

Multi-cell faults can also be described with an FP. The FP then needs to describe the fault effect on the victim cell that is caused by $\#C - 1$ aggressor cells. In order to do so, the ⟨S/F/R⟩ notation scheme has to be extended to $\langle S_{a,1};\ldots;S_{a,z};\ldots;S_{a,\#C-1};S_v/\mathrm{F}/\mathrm{R}\rangle$ [147]. Here, $\mathrm{S_a}$,z, with z < $\#C$ denotes the sensitizing sequence that is applied to an aggressor cell, $\mathrm{S_v}$ denotes the sensitizing sequence that is applied to the victim cell, F and R refer to the victim cell as well. With this extended notation, the complete fault space for static and dynamic multi-cell faults can be defined as well by following the same approach as was done for the single-cell faults. In the remainder of this work, we focus on single-cell faults.

4.2.2. FAULT ANALYSIS

The next step is to analyze which faults can actually be sensitized in a circuit. Only for the faults that are sensitized, a test needs to be developed. In this section, first we present the RRAM faults that are described in literature. Second, we perform a complete fault analysis using the linear resistor defect model.

EXISTING RRAM FAULT MODELS

In literature, several RRAM fault have been identified. Some of the faults are conventional memory faults that are also seen in other memory technologies (e.g., SRAM, DRAM), such as transition faults and stuck-at faults, while other faults are unique, due to the operating principles of the RRAM device. These faults have been identified by simulating linear resistor defect models and analyzing the resulting behavior, e.g., [11], [15], or by experimental measurement data analysis [19].

Conventional Fault Models

- Stuck-at-Fault (SAF) [11]: the cell is always in LRS (Stuck-at-1, SA1) or in HRS (Stuck-at-0, SA0). Defects that can cause this fault can be, for example, a missing WL connection, an open in the BL, or complete failure of the forming step which does not result in any CF formation.
- Transition Fault (TF) or Slow Write Fault (SWF) [11]: the cells fail to undergo a RESET or SET operation in the allowed time. The fault may occur only in one transition direction, i.e. ‘1’ to ‘0’ and ‘0’ to ‘1’. These faults are caused by resistive defects on the metal lines, weak access transistors, improper capping layer deposition or improper stack etching.
- State Coupling Fault (CFst) [10], [18], [20]: the state of an aggressor cell alters the state of a victim cell. This fault can occur when there is a bridge between two cells, bit lines or word lines. A write operation on the aggressor not only alters its state, but due to bridge also the victim cell flips its state [10], [18], [20].
- Write Disturbance Fault (WDF) [20]: these faults are coupling-like faults and are caused by a defective transistor. If the access transistor is stuck-at-ON, e.g. due to a short of the gate with V_{dd}, a writing operation on an aggressor cell sharing the bit or source line with the victim cell can result in an unintentional write to the victim cell. If this fault is sensitized in 1 cycle, the fault is called static (WDF), if it requires several consecutive cycles it is called dynamic (dWDF) [20].
- Read-1 Disturb Fault (R1DF) [19]: the cell returns a correct logic value when a read operation is performed, while the data that is stored by the cell is flipped by the read operation from ‘1’ to ‘0’. Note that the fault may also exist and flip ‘0’ to ‘1’. These faults are mainly weak faults and occur in weak cells, i.e., when LRS (HRS) is larger (smaller) than the nominal value. The small bias current during a read operation is sufficient to complete a RESET (SET) operation.
- Incorrect Read Fault (IRF) [12]: the cell returns an incorrect logic value when a read operation is performed, while the data stored by the cell is correct and not affected by the read operation. These faults are mainly strong faults, caused by the presence of resistive defects in the memory cell, e.g., opens in the BL or SL.

Unique RRAM Fault Models

- Undefined Write Fault (UWF) [11]: after a writing operation, the cell is brought into an undefined state ‘U’. This fault is caused by a lack of bias during the writing operation in weaker cells, e.g., due to an open in the SL. If now a read operation is performed on this cell, a random logic value will be read [11].
- Deep State Fault (Deep) [15]: The resistance of the cell is beyond the limits for ‘1’ or ‘0’ [15]. This weak fault can for example be caused by overforming.
- Unknown Read Fault (URF) [14]: a read operation is performed on a cell that stores an ‘U’. The resistance is too close to the reference of the SA, and therefore a random output will be generated.

Unifying Terminology As can be seen above, not all faults that have been observed in RRAMs are defined using the FP notation scheme. For clarity, we analyze these faults and define them according to our naming scheme to unify the fault terminology.

- *SAF* [11], [15]: Stuck-at Faults are defined as a cell always being in a certain state, independent of the sensitizing sequence [147]. The fault can therefore be described by multiple FPs, e.g. a SAF-1 can be represented by a combination of: S0F1, W0TF1, iR0DF1, etc.
- *Deep 0/1* [15]: Deep Faults are defined as a cell being in the ‘L’ or ‘H’ state. Again, multiple FPs can describe this fault: W0DFL, rR1DFH, etc.
- *SWF* [15]: Slow Write Faults are defined as a cell that fails to transition in the allotted time and is brought into the ‘U’ state. Therefore, this fault can be described as W1TFU or W0TFU.
- *URF* [15], [20]: Undefined Read Faults are defined as performing a read operation on a cell that is in a ‘U’ state, thus resulting in a random read outcome. Because S cannot contain a ‘U’ state, this FP cannot exist by itself. An appropriate FP for the URF is rR1DFU.
- *R1DF* [19]: Read-1 Disturb Faults are defined as reading operations causing a cell’s state to change. Therefore, this fault belongs to the read destructive faults, e.g., iR1DF0, dR1DF0, etc.

LINEAR RESISTOR DEFECT MODEL FAULT ANALYSIS

In this section, we perform a complete static fault analysis, i.e., $n \leq 1$ using all linear resistor defect models that were explained in Section 4.1. We apply the defcts in the circuit that was shown in Fig. 2.7. We analyze every defect and vary the defect strength from 1 Ω to 100 MΩ with logarithmically increasing step sizes. Table 4.2 summarizes the obtained results. For some defects, multiple faults are sensitized for different defect strengths. For example, $R_{\text{Op BL}}$ sensitizes both ⟨1w0/1/-⟩ and ⟨1w0/U/-⟩. The superscripted numbers indicate the defect strength range (i.e., the resistance range) in which this fault is sensitized. These ranges are unique for every defect, e.g., range 1 for $R_{\text{Br BL-int}}$ is not necessarily the same as range 1 for $R_{\text{Sh SL-GND}}$. To guarantee detection of every defect strength, every fault needs to be detected with a test.

Table 4.2: Static faults sensitized by all linear resistor defect models from Fig. 4.1

S	$R_{\text{Br BL-SL}}$	$R_{\text{Br BL-WL}}$	$R_{\text{Br BL-int}}$	$R_{\text{Br SL-WL}}$	$R_{\text{Br SL-int}}$	$R_{\text{Br WL-int}}$	$R_{\text{Op BL}}$	$R_{\text{Op SL}}$	$R_{\text{Op WL}}$
0									
1									
0w0									
0w1									
0r0									
1w0	$\langle 1w0/U/\text{-}\rangle^{1}$		$\langle 1w0/U/\text{-}\rangle^{1}$				$\langle 1w0/1/\text{-}\rangle^{1}$, $\langle 1w0/U/\text{-}\rangle^{2}$		
1w1									
1r1		$\langle 1r1/1/0\rangle^{1}$		$\langle 1r1/1/0\rangle^{1}$		$\langle 1r1/1/0\rangle^{1}$	$\langle 1r1/1/0\rangle^{1}$	$\langle 1r1/1/0\rangle^{1}$	$\langle 1r1/1/0\rangle^{1}$

S	$R_{\text{Sh BL-GND}}$	$R_{\text{Sh BL-}V_{\text{DD}}}$	$R_{\text{Sh SL-GND}}$	$R_{\text{Sh SL-}V_{\text{DD}}}$	$R_{\text{Sh WL-GND}}$	$R_{\text{Sh WL-}V_{\text{DD}}}$	$R_{\text{Sh int-GND}}$	$R_{\text{Sh int-}V_{\text{DD}}}$	
0									
1									
0w0							$\langle 0w0/U/\text{-}\rangle^{1}$		
0w1									
0r0									
1w0			$\langle 1w0/U/\text{-}\rangle^{1}$		$\langle 1w0/1/\text{-}\rangle^{1}$, $\langle 1w0/U/\text{-}\rangle^{2}$		$\langle 1w0/U/\text{-}\rangle^{1}$		
1w1									
1r1		$\langle 1r1/1/0\rangle^{1}$		$\langle 1r1/1/0\rangle^{1}$		$\langle 1r1/L/0\rangle^{1}$		$\langle 1r1/1/0\rangle^{1}$	

4.3. TEST DEVELOPMENT

After the faults are analyzed and it has become clear which faults can exist in a circuit, a test can be developed. To tests memories, typically, March tests are used. These test are algorithms of read and write operations that are applied to a memory in a particular order, so that faults will be sensitized and detected. The benefit of these tests is that the existing hardware of the memory is used and that the test is performed on the actual operating speed of the memory. However, not all faults can be detected by using just the regular memory operations. That is, the fault coverage of such a test is not 100%. This may lead to some faulty devices escaping the test and being sold as good devices. These faulty devices are test escapes. Detecting these faults requires additional effort in the form of DFT schemes. In this section, we will discuss both tests for RRAM. Subsequently, we will describe a test solution that can detect the faults that were sensitized using the linear res istor defect model in the previous section.

4.3.1. MARCH TESTS

March tests are used in all kinds of computer memories for decades, e.g., SRAM [149] and DRAM [146]. They can also be applied in RRAMs. First, we present an overview of existing march tests for RRAMs. Then, we present the details of these tests.

OVERVIEW

In literature, sever March tests for RRAM have been defined. Typically, they are derived from more generic March tests for other memory technologies, such as March C- [149], and extended to detect the RRAM unique faults that were described in Section 4.2.2. Table 4.3 lists all algorithms for 1R and 1T1R arrays, as well as their fault coverage. Note that there are also March tests for RRAM that make use of DFT. These are discussed in the next section.

Table 4.3: Test Algorithms for RRAMs. If an algorithm detects a fault only partially, it is not added to the final "coverage" column.

Algorithm		Conventional						Unique			Coverage
Year	Name	SAF	TF	WDF	IRF	R1DF	CFst	UWF	URF	Deep	
2015	March C* [19]	Y	Y	N	Y	Y	Y	N	N	N	5/9
2015	March 1T1R [20]	Y	Y	Y	Y	N	Y	N	N	Partially	4/9
2016	March C*-1T1R [23]	Y	Y	N	Y	Y	Y	N	N	Y	6/9
2017	March W-1T1R [24]	Y	Y	Y	Y	Y	Y	N	N	Y	7/9

DETAILED ANALYSIS

March tests use regular memory operations, i.e., read and write operations, to sensitize and detect faults in the memory. The order of these operations is determined by the sensitization and detection conditions of a fault. To illustrate, consider an the fault ⟨0w1/0/-⟩ and the MATS+ algorithm in Equation 4.2 [149]:

$$\text{MATS+} = \{\Updownarrow(\text{w}0)\,;\Uparrow(\text{r}0,\text{w}1)\,;\Downarrow(\text{r}1,\text{w}0)\}\,. \tag{4.2}$$

In this equation, $\Uparrow$, $\Downarrow$, and $\Updownarrow$ indicate the addressing order, increasing addresses, decreasing addresses, and either in- or decreasing addresses, respectively. Every block of parentheses ((...)) denotes a March element. It is applied with the given addressing order. Every March element can contain read (r) or write (w) operation. The data that is read or written is noted after the operation, i.e., '1' or '0'. The algorithm detects the fault as follows. The fault can be sensitized by first initializing a cell to '0' and subsequently performing a w1 operation. This is achieved in the first two March elements. The fault is then detected by performing a r1 operation in the last March element. A March test is developed by identifying all sensitizing and detection conditions of all faults that can occur in a circuit and combining these operations into an algorithm. This algorithm should be as fast as possible to limit test time. Next, we analyze the existing March test for RRAM in more detail.

March C* [19] This is a modified version of the standard March C- algorithm [149]. The algorithm is modified so that it also detects the R1DF by adding a a r1 operation after an other r1 operation.

March 1T1R [20] This is newly developed March algorithm that detects the WDF and dWDF by repeating the write operations.

March C*-1T1R [23] This is a modified version of the March C* algorithm. The algorithm includes detection of deep faults by performing double write operations and subsequently a complementary write operation and read operation.

March W-1T1R [24] This is a newly developed March algorithm to detect WDF as well as IRF and R1DF.

Clearly, March tests alone cannot detect all RRAM faults. Therefore, their detection capabilities need to be improved by including DFT schemes.

Table 4.4: DFT for RRAMs. If a DFT detects a fault only partially, it is not added to the final “coverage” column

DFT		Conventional						Unique			Coverage
Year	Name	SAF	TF	WDF	IRF	R1DF	CFst	UWF	URF	Deep	
2012, 2015	Weak write [12], [13]	N	N	N	Y	N	N	Y	Y	Y	4/9
2013, 2014	Sneak Path [15], [16]	Y	Y	Partial	N	N	Partial	N	Y	Y	4/9
2015, 2017	Fast write [17], [18]	Y	Y	N	N	N	N	Y	Y	Y	5/9
2017	MAGIC NOR [25]	Y	Y	N	Y	Y	Y	N	N	Y	6/9

4.3.2. DESIGN-FOR-TEST SCHEMES

Design-for-test schemes increase the test coverage by providing other sensitization and detection possibilities than only regular memory operations. However, adding DFTs to a circuit comes at the cost of increased chip area or power usage.

OVERVIEW

There exist several DFT schemes for RRAM. They can be classified into two categories: DFTs to increase the fault coverage, and DFTs to increase the test execution speed. Table 4.4 lists all existing DFT schemes and their coverage when combined with the accompanying March algorithm (if present). The table shows that there is no test solution that can detect all RRAM faults.

DETAILED ANALYSIS

Next, we analyze all DFT schemes in detail.

Weak write [12], [13] This scheme weakens the write operation so that defective cells can be detected. The weak write operation causes defective cells to fail a transitioning operation, while normal cells will succeed. A subsequent read operation can then be used to detect the fault. The weakening is performed by reducing the write voltage or by reducing the write time duration and it can be calibrated.

Sneak path [15], [16] This scheme uses the sneak paths to detect the state of multiple cells at once in a test region to reduce the test execution time. The DFT allows to access multiple cells at once, and to measure the resulting current. Consider that a test region should contain only ‘0’. If one of the cells stores a ‘1’, the current will be lower than the expected current for a region than only contains ‘0’s, and thus the fault can be detected.

Fast write [17], [18] This scheme proposes to reduce the time for a w0 (RESET) operation, as this is the slowest operation in a RRAM. As a shorter write time will put the cells closer to ‘U’, the scheme also introduce‘s two additional references to detect whether a cell is in ‘U’.

MAGIC NOR [25] This scheme applies the MAGIC computation-in-memory NOR design [150] and a parallel write operation to speed up the test execution. A NOR operation is performed in parallel on a complete row to parallelize the r1 operation. That is, if one of the cells in the row is ‘0’, the NOR operation will output 1. The parallel write operation speeds up the test execution further.

4.3.3. TEST FOR ALL LINEAR RESISTOR MODEL DEFECTS

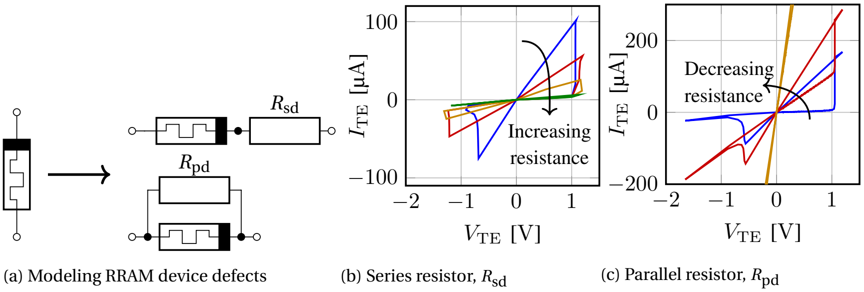
The results of the fault analysis using linear resistor defect models are now used to develop a test that can detect them. From the Table 4.2, it can be seen that all FPs belong to the following RRAM faults from Section 4.2.2: UWFs, IRFs, TFs, and Deep faults. Next, from Table 4.3 and 4.4 it follows that a test for all these faults should use the March W-1T1R algorithm in combination with the weak write DFT.

5

DEVICE-AWARE TEST APPROACH

The previous chapter presented the traditional defect modeling, fault modeling and analysis, and test development approach. It models spot defects in the wiring as linear resistors in or between two nodes of a circuit. However, we have shown that spot defects do not occur in the RRAM device itself. Therefore, it is unsuitable to use a linear resistor to model defects within the RRAM device and doing so will lead to test-escapes and thereby low-quality devices. The Device-Aware Test approach overcomes this problem by modeling the actual physics of a defective device and incorporating the defective behavior in the device model itself. Subsequently, the resulting defect model is used for fault analysis, and test development. Because, the actual physics of the defect are modeled, the resulting test will be able to detect defects in the RRAM device itself, rather than in the wiring that surrounds it. Therefore, tests developed using this approach will be of high-quality and the number of test escapes is reduced significantly. This chapter describes the Device-Aware Test approach.

This chapter is partially based on [27]–[30].



(a) Modeling RRAM device defects (b) Series resistor, R_{sd} (c) Parallel resistor, R_{pd}

Figure 5.1: Linear resistor defect modeling

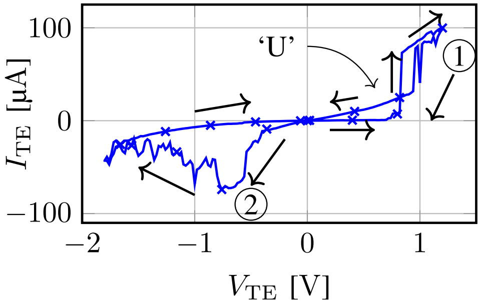


Figure 5.2: Defective RRAM device.

5.1. SHORTCOMINGS OF THE TRADITIONAL APPROACH

In Chapter 3, we analyzed the production process of a RRAM device and the defect that may occur there. It became clear that there are many defects in the RRAM device that have not been seen in traditional memories, such as SRAM and DRAM, or even in generic CMOS processes. Then, we showed in Chapter 4 that the traditional test development approach models spot defects as linear resistors in or between nodes of a circuit. However, it is clear that RRAM device defects are not spot defects. Hence, using linear resistor defect models to model RRAM device defects is wrong and will lead to low-quality tests. Next, we will illustrate this in more detail.

Fig. 5.1a shows how the traditional linear resistor defect modeling approach models defects in the RRAM device; the only options are putting a resistor in series (R_{sd}) or in parallel (R_{pd}) with the unmodified, defect-free RRAM device model. Fig. 5.1b and 5.1c show the resulting I-V curves. As can be expected, the series resistor decreases the voltage over the RRAM device, and thus limits its switching capabilities, while the parallel resistor only increases the current through the cell, without affecting the actual switching of the device. Now, consider Fig. 5.2 that shows the I-V curve of a defective RRAM device.[1] It is immediately visible that a series or parallel resistor is unable to model this defect. As such, when a test for RRAM is developed using only linear resistor defect models, the resulting test will not be able to detect all possible faults.

---

[1] This device suffers from an intermittent undefined state fault and will be discussed in detail in Chapter 7.

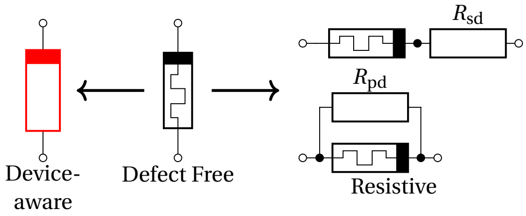


Figure 5.3: Device-aware and resistive defective-device models.

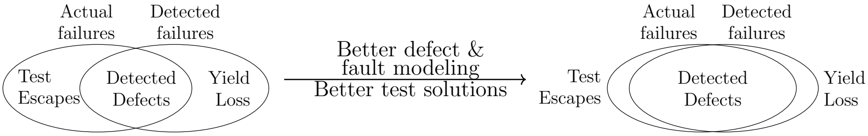


Figure 5.4: Aim of the device-aware test approach

In stead of using linear resistors *around* a defect-free RRAM device to model defects *in* the RRAM device, the RRAM device model itself should be modified so that it describes the actual physics of a defect. Kannan, Karri, and Sinanoglu have tried to do this by changing the area and cell length parameters of a RRAM model to mimic a defect in the RRAM device [55]. However, it is unclear how these parameters and their corresponding changes map to an actual defect. For example, it is unclear how a thinner capping layer will affect these parameters. Hence, this defect modeling approach may lead to faults that cannot exist in real RRAMs, leading again to low-quality tests. To overcome these problems, we propose the Device-Aware test approach that will be discussed in the remainder of this chapter.

5.2. OVERVIEW OF THE DEVICE-AWARE TEST APPROACH

We have established that the defective behavior of RRAM device cannot be modeled using linear resistors in series or in parallel with a defect-free device model. Hence, tests developed based on these models are unable to detect all the *realistic* device failures, leading to test escapes, or they test for *unrealistic* faults, increasing the yield loss. To overcome this, we introduce the Device-Aware Test (DAT) approach that relies on accurate defect and fault modeling in order to develop better test solutions for RRAMs. Fig. 5.3 shows the core idea of this approach. The aim of DAT is to incorporate the physics of a defective device into the RRAM device model, rather than modeling the defects as linear resistors that surround a defect-free RRAM device model. Fig. 5.4 shows that appropriate defect and fault modeling leads to the detection of more realistic faults, while decreasing the detection of unrealistic faults, thus increasing test quality. In this section, we describe the approach. It is illustrated in Fig. 5.5 and consists of three steps that are explained next:

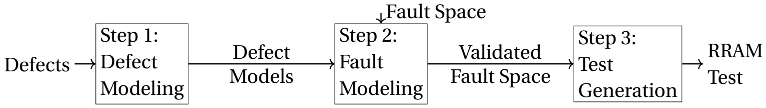


Figure 5.5: Device-aware Test approach overview

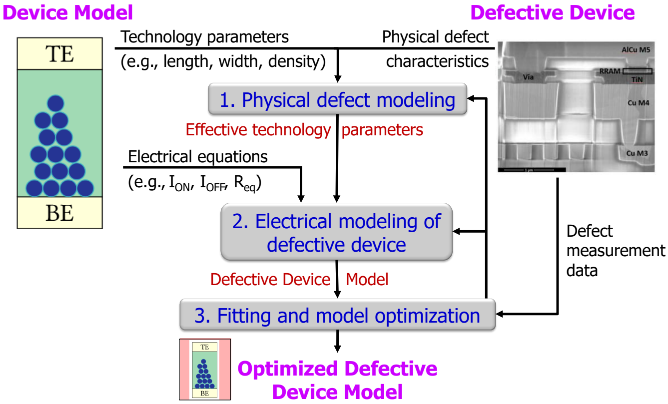


Figure 5.6: Device-Aware defect modeling

1. Defect modeling This is the most important step of the framework, as the quality of the resulting defect models determines the final test quality. In this step, all possible defects that may occur in a circuit are identified and analyzed. Then, the physics of these defects are modeled and incorporated in an electrical defect model that can be used in circuit simulations. Because the physics of the defect are modeled, these electrical defect models will properly describe the defective behavior and thus can be used to develop accurate tests.

2. Fault modeling In this step, the defect is described as a fault on the functional level of the memory [7]. First the fault space is classified and defined to describe all theoretically possible faults. Then, the defect models from the first step are incorporated in the netlist and the circuit is simulated. The resulting behavior is analyzed and, if found erroneous, labeled as a fault. This is repeated for all defects and results in the validated fault space. This is a list of faults that can realistically occur in the circuit and for which a test needs to be developed.

3. Test generation In this step, test solutions are developed that can detect all faults from the previously validated fault space. The resulting test solutions can be, for example, march algorithms, special DFT schemes, or stress tests.

Next, we discuss these steps in detail.

5.3. DEFECT MODELING

Inaccurate defect modeling may result in poor fault models, thereby limiting the effectiveness of proposed test solutions and DFT designs, not only in terms of defect coverage but also in terms of test time. For example, a test targeting a fault model that does not represent any real defect will not increase the defect coverage while still consuming test time. To accurately model physical defects, the device model should incorporate the way the defect impacts the technology parameters (e.g., length, width, density) and thereafter the electrical parameters (e.g., the switching thresholds) of the device; this is exactly what device defect modeling of DAT does. Fig. 5.6 shows the flow of such modeling approach; its inputs are 1) the electrical model of a device, and 2) the defect under investigation. The output is an optimized (parameterized) model of a defective device. Note that a device can be a planar or FinFET transistor, an MTJ device, a RRAM device, a PCM device, etc. The approach consists of the following three steps.

1. Physical defect analysis and modeling Given a set of physical defects $\mathbf{D} = \{\mathrm{d}_1, \mathrm{d}_2, \ldots, \mathrm{d}_\mathrm{n}\}$ that can occur during the manufacturing process of the device, each defect d_i has to be analyzed to fully understand the defect mechanism and identify its impact on each (key) technology parameter of the device. Due to such a defect, one or more technology parameters will be modified from their defect-free values (Tp_df), resulting in what we refer to as an *effective technology parameter* Tp_eff. This can be described by the following abstract function:

$$Tp_\mathrm{eff}(\boldsymbol{S_i}) = f_i(Tp_\mathrm{df}, \boldsymbol{S_i}) \tag{5.1}$$

where Tp_df is the defect-free technology parameter, f_i is a mapping function corresponding to defect d_i ($i \in [1,\mathrm{n}]$), and $\boldsymbol{S_i} = \{x_1, x_2, \cdots, x_\mathrm{t}\}$ is a set of parameters representing the size or strength of defect d_i. It is worth noting that each defect may impact one or more technology parameters.

2. Electrical modeling of the defective device In this step, the impact of the altered technology parameters from Step 1 on each of the key electrical parameters of the device is identified. The resulting electrical parameters are therefore qualified to describe the electrical behavior of the defective device with defect d_i. This is done by modifying the defect-free device electrical model and converting it into a defect-parameterized model by integrating Equation 5.1 for each involved technology parameter. This step gives an uncalibrated defective-device model with the effective electrical output parameters.

3. Fitting and model optimization To guarantee the accuracy of the defective-device model, the model needs to be calibrated. Therefore, real-world defective devices need to be measured. If any physical or electrical parameters of the defective model do not accurately match the characterization data, then it is necessary to keep optimizing the device model until an acceptable accuracy is obtained. By performing silicon data fitting and model optimization, we can derive an optimized defective-device model, which enables accurate circuit simulation for fault modeling.

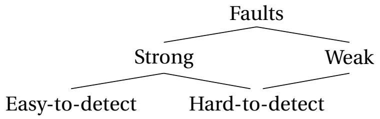


Figure 5.7: Detectability of different faults

5.4. FAULT MODELING AND ANALYSIS

5.4.1. FAULT MODELING

Fault modeling is similar to the traditional approach, as discussed in Section 4.2. As such, all faults and functional fault models in Table 4.1 are applicable as well.

5.4.2. FAULT ANALYSIS

Fault analysis is the validation of the fault space to see which faults can realistically occur in a circuit. Only those faults that can occur need to be detected by a test. This relates the fault analysis step closely to the test development step.

The fault analysis methodology aims to determine what faults can realistically occur in a circuit, and to determine which faults can be efficiently detected by a test. The former aim reduces the amount of test escapes, and decreases the yield loss due to wrongly modeled faults, as shown in Fig. 5.4. For the latter aim, the detectability of the faults has to be identified, as shown in Fig. 5.7. A fault can be a strong or a weak fault. A strong fault is *always* sensitized by a certain sequence of operations and it can be described by an FP, e.g., W0DFU, rR1NF1, or dR0DF1. A strong fault will thus always affect the cell's state or the read output, or both. These faults *may* be detected by, for example, a march test that only uses regular memory operations. In contrast, a weak fault does not cause functional errors but parametric deviations instead and cannot be described by an FP, as the cell's contents or read output are unaffected. Examples are, a decrease of bit line swing below the nominal values when reading a certain cell, but not causing a wrong or random read output, or when the switching thresholds shift outside of their nominal ranges, but switch does occur. Strong faults that are *guaranteed* to be sensitized and detected by regular memory operations are called Easy-to-Detect (EtD) faults, e.g., W1TF0 and iR1NF1; both these faults will be detected by performing a regular read operation that always outputs a wrong value. Note that not all strong faults in Table 4.1 are EtD. For example, it *cannot* be guaranteed that the random read output caused by rR1NF1 is detected by performing a normal read operation, as its output has a *probability* to be either '1' or '0'. Hence, a regular read operation will only in some cases detect these faults. These strong faults together with all weak faults are called Hard-to-Detect (HtD) faults. To detect these faults, additional effort is required, e.g., by using DFT schemes. Hence, a strong fault can be further classified as EtD or strong HtD (sHtD), while weak faults are always weak HtD (wHtD) faults. Note that these two fault types both are hard faults, as they will always occur given the same circumstances; there is no random component involved in their sensitization.

The fault analysis methodology is shown in Fig. 5.8 and consists of seven steps: 1) circuit generation, 2) defect injection, 3) stimuli generation, 4) circuit simulation, 5) fault

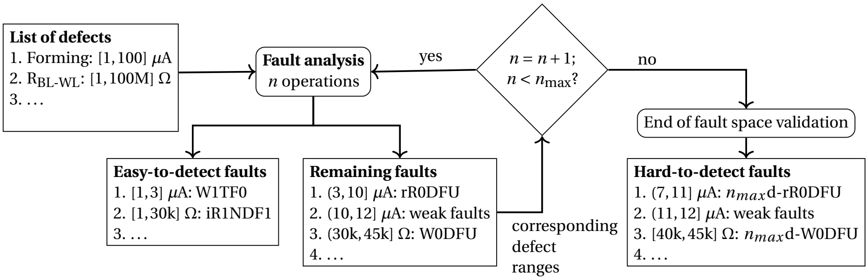


Figure 5.8: Fault analysis methodology

analysis, 6) fault primitives identification, 7) defect strength sweeping, and repetition of steps 2 to 7 until all defects and defect strengths are covered. The methodology starts with a list of defects along with their strengths. These defects are injected one by one in the netlist to which stimuli for at most one operation, i.e., $n \leq 1$, are applied. Then, the behavior of the circuit is analysed. If the fault can be described by an FP, then the fault is strong. For these faults the detectability is determined, i.e., whether it is an EtD or sHtD fault. If the parameters of the circuit are outside of its specifications but are not described by an FP, then the fault is weak, i.e., a wHtD fault. If the parameters of the circuit are within the specification, then the there is no fault present for that value of n.

Defects with certain defect strength ranges that sensitize EtD faults do not require further analysis, as they can already be easily detected. They are listed as EtD faults. However, analyzing them further allows to better optimize tests, as will be shown in Section 5.5. The remaining defect strength ranges are added to the remaining faults list. They are simulated again, but now with a longer sensitizing sequence with length $n = n+1$. This process is repeated until all defects and sizes are in the EtD list, or when the n passes a maximum n_{max}. These remaining HtD faults are dynamic faults that are sensitized when $n = n_{\text{max}}$, e.g., a n_{max}d-rR0DFU fault. The choice of n_{max} is an economic choice that depends on the allotted total simulation time and the desired fault coverage. Equation 4.1 showed that the total time increases exponentially with n, hence a higher *guaranteed* fault coverage comes at a higher development cost. Note that n_{max} is the upper limit and that the test development cost can be reduced; e.g., if it becomes clear that no additional faults are sensitized with increasing n, the process can end earlier. Some defect strength ranges do not sensitize any fault when $n = n_{\text{max}}$ and the circuit could be labeled fault-free. However, these defects may form a reliability risk, e.g., electromigration poses a higher risk in thinner wires. Hence, a test developer needs to make a choice between increasing n_{max} to increase the fault coverage, and to minimize development costs.

The resultant fault lists are used for test development. The EtD faults can be detected by applying march tests, while the HtD faults require special test solutions, e.g., DFT schemes or stress tests. Note that a single defect may sensitize multiple faults, e.g., an open in the WL may sensitize W1TF0, iR1NF1, etc. All these faults together form a single fault class (FC). When developing a test to detect this defect, it suffices to detect only one fault per FC, which eases the test development further.

The total worst-case analysis time ($t_{\text{analysis,w.c.}}$) using this fault analysis methodology is determined by $n_{\max}$ (which affects #S, see Equation 4.1), the total number of defects ($\#D_{\text{tot}}$), and the number of defect strengths per defect ($\#D_{\text{strength}}$). It is described by:

$$t_{\text{analysis,w.c.}} = \left(\sum_{i=0}^{n_{\max}} 2 \cdot 3^{i} \right) \cdot \#D_{\text{tot}} \cdot \#D_{\text{strength}}. \tag{5.2}$$

This equation assumes that all sensitizing sequences need to be applied to all defects for all strengths. Hence, it gives an upper limit on $t_{\text{analysis,w.c.}}$. In reality, the total analysis time will be lower, as many defect strengths will sensitize EtD faults that do not need to be studied further.

5.4.3. FAULT ANALYSIS OF INTERCONNECTION DEFECTS

We demonstrate the fault analysis method by applying it to all interconnection defects that can occur in a 1T1R cell, as shown in Fig. 4.1. We inject the defect in the RRAM architecture from Fig. 2.7. We study one defect at a time and vary its defect strength from 1 Ω up to 100 MΩ in 81 logarithmically spaced steps. We apply all sensitizing sequences up to $n_{\max} = 3$ operations. The results of this analysis for the defect $\text{R}_{\text{Br BL-int}}$ are shown in Table 5.1. The table shows per defect strength and sensitizing operation what class of faults is sensitized. The gray color indicates fault-free behavior, green indicates an EtD fault, yellow an sHtD fault (i.e., it can be described by an FP), and cyan a wHtD fault.

The table shows that faults are sensitized for lower defect strengths, which is expected for a bridge defect. Furthermore, it can be seen that defects that sensitize EtD faults for lower defect strengths, will sensitize strong HtD faults for increased defect strengths, e.g., see S = 1w0. To illustrate, the defect range $R_{\text{Br BL}-\text{int}} \in [1, 10\text{k}]$ sensitizes EtD faults, while the range $R_{\text{Br BL}-\text{int}} \in (10\text{k}, 316\text{k}]$ sensitizes sHtD ones. The table also shows that increasing the length of the sensitizing sequence increases the amount of EtD faults that is sensitized. For example, performing a r0 operation after the previous 1w0 operation (i.e., S = 1w0r0) sensitizes an EtD fault for the defect range $R_{\text{Br BL}-\text{int}} \in [1, 501\text{k}]$. Hence, when developing a test, this sequence is to be chosen over S = 1w0, because a wider defect range will sensitize EtD faults. Although extending the length of the sensitizing sequence does increase the amount of sensitized EtD faults, there are still defect strengths that only sensitize strong HtD faults. These faults cannot be guaranteed to be detected by regular memory operations alone. Therefore, special measures have to be taken to detect these faults and guarantee the highest fault coverage possible.

Table 5.2 and 5.3 summarize the results for the remaining defects. The tables are generated by analyzing the fault maps similar to those in Table 5.1 and is used further during test development. The tables show for every defect (column) the fault classes (FCs) (digits after the FP) that are sensitized and the corresponding FPs (in which S denotes the applied sensitizing sequence). For clarity, only the FCs that belong to the largest number of defect strengths are listed. A black digit indicates that the FP in this FC is EtD, while a red digit indicates an sHtD fault. Remember that to detect all defects, at least one FP per FC needs to be detected. To illustrate the construction of the table and the FCs, consider the short defect $\text{R}_{\text{Br BL-int}}$ in Table 5.1. The fault map shows that S ∈ {1w0r0, 0w1w0r0, 1w1w0r0, 1r1w0r0} all sensitize EtD faults for the defect strength up to 50.1 kΩ. However, for the same range of defect strengths, there are also sHtD faults sensitized, e.g., by S ∈ {1w0, 1w1w0, ...}. Because the defect can be detected if any of these faults is detected, these two sets form a single FC, labeled ‘1’ in Table 5.2 and 5.3. From Table 5.1 it follows that S = 1w0 uniquely sensitizes sHtD faults from 50.1 kΩ up to 316.2 kΩ. Hence, this also forms a single FC, labeled with a red ‘2’ in the table. From the table, we now can see that to detect $\mathrm{R}_{\mathrm{Br\ BL\text{-}int}}$, only two FCs need to be detected. Further, it tells that FC 1 is EtD and thus can be detected by regular memory operations, while the guaranteed detection of sHtD FC 2 requires additional effort. Note that no wHtD faults are sensitized by this defect without also sensitizing an EtD or sHtD fault. Therefore, these faults are not listed in the table.

5.5. TEST DEVELOPMENT

5.5.1. TEST DEVELOPMENT ALGORITHM

After the fault analysis is performed, a test can be developed that detect the faults that have been sensitized. The design of the test depends on the detectability of the faults that are sensitized, as well as on the potential to combine the sensitization of multiple faults so to achieve efficient tests. The detectability of the fault determines whether March algorithms alone are enough to deliver a sufficient test coverage, i.e., when there are only EtD faults, or whether additional test measures need to be included, such as the DFTs from Section 4.3.2. Not all, faults need to be detected separately; it suffices to detect only one fault from a fault class. Furthermore, Some faults may be part of multiple fault classes, and thus detection of such a fault leads to the detection of many corresponding defects. We have developed an algorithm to minimize the selection of faults that need to be included in a test to maximize the defect coverage.

The algorithm works as follows. We consider two linear resistor defect models, Defect 1 and Defect 2, and vary the resistance from 1 Ω to 100 MΩ. We apply only sensitization sequences with one operation to the circuit. The fault analysis could result then result in Table 5.4. This table lists a 1 for every defect with a specific strength $j \in [1, \ldots, q]$ that sensitizes a fault with sensitizing sequence $i \in [1, \ldots, p]$. If no fault is sensitized, the defect with its corresponding strength is not included in the table. To illustrate, Defect 1, with a strength of 100 MΩ sensitizes faults when S is 1r1 or 1w1. The right column shows how many sensitizing sequences will sensitize a fault for the given defect strength. Here, $a_{i,j}$ is an entry in the table. The algorithm will now minimize the number of sensitizing sequences that are included in a test, while considering that every defect strength needs to be detected. This can be formulated as an integer-linear programming (ILP) problem as follows:

$$\min_{\mathrm{S_{selected}}} \sum_{i=1}^{p} c_i \cdot \mathrm{S}_{\mathrm{selected},i} \qquad \text{subject to:} \sum_{i=1}^{p} a_{i,j} \cdot \mathrm{S}_{\mathrm{selected},i} \geq 1 \text{ for all rows } j. \tag{5.3}$$

Here, $\mathrm{S}_{\mathrm{selected},i} \in \{0, 1\}$ is a binary value that indicates whether the ith sensitizing sequence is selected and c_i is a weight for this sequence, normally set to $c_i = 1$. The first statement ensures that the amount of selected Ss is minimized, while the second statement ensures that every defect strength is covered by at least one sensitizing sequence. This problem can be solved using various tools, e.g., Matlab [151].

To illustrate, Table 5.4 shows that 0r0 needs to be included in the test, as this is the only S to sensitize a fault for defect strength $j = 3$. Therefore, $\mathrm{S}_{\mathrm{selected}}, 1 = 1$. The same applies for S = 1r1 and S = 1w0. To sensitize defect strength $q-1$, both S = 0w0 and S = 0w1 can be selected. To minimize the test length, only one of them is selected, in this case S = 0w0. Therefore, all defects can be detected by including all $\text{S}_{\text{selected}}, i = 1$.

The set of selected sensitizing sequences can now combined with detecting operations, e.g., if $\text{S}_{\text{selected}}$ = 1w0, a subsequent r0 is required to detect the fault. This combined set can now be merged into a March algorithm. This algorithm is of minimal test complexity, while it detects the maximal number of defects. This results in more efficient tests than using standard March algorithms, such as those described in Section 4.3.1, because the traditional algorithms detect all possible faults. In contrast, the test that results from this approach detects only the faults that are required to maximize test coverage. Note that it is possible to automatically develop such a March algorithm [152].

5.5.2. TRADE-OFFS AND CONSIDERATIONS

There are some considerations and trade-offs when using the test development algorithm. These are: preference for certain operations, better EtD detection, and detection difficulty.

Operation preference It is possible to reformulate the problem in order to optimize the set of sensitizing sequences for different goals. For example, in Equation 5.3, it is assumed that writing (t_W) takes as much time as reading (t_R), as is reported in [153]. However, there exist also RRAM designs where $t_w \neq t_r$, as reported in [154]. For the latter design, the test time can be optimized by favouring the operation that is shorter by changing the weights c_i of sensitizing sequences that contain more of the faster operations. To illustrate, when $t_w = 2 \cdot t_r$, $c_w = 2 \cdot c_r$ as well. Hence, when solving the problem in Equation 5.3, it becomes harder to minimize the first sum when a write operation is selected and thus read operations are favored.

Better EtD Detection The fault analysis algorithm in Fig. 5.8 does not analyze defects that sensitize EtD faults further to save test development time. However, it is possible that a longer sensitizing sequence leads to the sensitization of an EtD fault that is present in more fault classes. Including this fault in the test may lead to a shorter test overall at the cost of increased test development time.

Detection Difficulty The weights c_i can also be modified so that the cost of the detecting operation is included as well, e.g., to add a r0 operation after a sensitizing sequence 1w0. Furthermore, when using a certain DFT, this cost can also change, e.g., a DFT that introduces parallel operations changes the cost of some operations.

5.5.3. TEST DEVELOPMENT FOR INTERCONNECTION DEFECTS

We now apply the test development algorithm for the faults in Tables 5.2 and 5.3 that are sensitized by interconnection defects. First, we apply the algorithm to develop a test for EtD faults, subsequently we develop a test for EtD and sHtD faults, We applied MathWorks's MATLAB R2019b [151] to solve this problem for the EtD faults. The solver finds the set $\text{S}_{EtD,int/cont} \in \{1\text{r}1, 1\text{w}0\text{r}0, 1\text{r}1\text{w}0\}$ that sensitizes all EtD faults. Since all faults are

EtD, they can be detected by performing a read operation after the sensitization if the last operation is not an incorrect read operation already. For example, S = 1r1w0 can be detected by adding a r0 operation, while S = 1w0r0 sensitizes and detects the faults. All EtD faults sensitized by interconnect and contact defects can be detected by the following March algorithm:

$$\text{March-EtD, int/cont} = \left\{ \Updownarrow (\text{w1}) ; \Updownarrow (\text{r1}, \text{w0}, \text{r0}) \right\}. \tag{5.4}$$

The test time for this algorithm is $2T_w + 2T_r$, where T_w and T_r denote the time to write and read respectively the complete address space once. It can be seen that the algorithm is minimized in number of operations, as 1r1 can be included in 1r1w0, and then only 1w0r0 needs to be added. We developed this algorithm manually, but it is possible to do this automatically as well if the set of required Ss grows larger [152].

To detect all faults in the RRAM, also the sHtD faults need to be detected by a test. There are two ways to implement such a test: 1) a standalone test is developed for the sHtD faults only and performed after the EtD test, or 2) a test is generated considering both EtD and sHtD faults at the same time. The first option is easier, but results in less optimized tests, while the second option requires slightly more effort, but results in an optimized test. For this latter reason, we develop a test for both EtD and sHtD faults at the same time.

We again apply the test development algorithm, but now we also add sHtD faults to the table. This results in the following set of sensitizing sequences that will sensitize all EtD and sHtD faults: $\text{S}_{EtD-sHtD,int/cont} \in \{1\text{w0}, 1\text{r1}, 0\text{w0w0w0}, 1\text{r1w0r0}\}$. The EtD faults can again be detected by performing a read operation after the sensitizing sequence, if required. However, this does not apply for the sHtD faults. For example, when reading a cell that suffers from a W0TFL fault, a subsequent read operation will result in '0', thus hiding the faulty state of the cell. Therefore, DFT is required to detect these faults.

Hamdioui, Taouil, and Haron have presented a DFT scheme for RRAMs that modifies the write operations in order to detect cells in the 'U' state [13]. The paper proposes to shorten the duration of the write operation, or to decrease the write voltage. These *weak write* operations are denoted as $\hat{\text{w}}$.

We apply the weak write DFT scheme [13] (see Section 4.3.2) to detect the sHtD faults. The idea behind the $\hat{\text{w}}$ operations is that defect-free cells will be able to switch to the desired state, but that defective cells do not have enough time or driving voltage to switch and thus will remain in a wrong state. As an illustration, consider the W0TFL fault. It can be sensitized and detected by the following sequence: $\Updownarrow (\text{w0}, \hat{\text{w}}1, \text{r1})$. The $\hat{\text{w}}1$ operation is unable to switch the defective cell to '1' and the cell remains in 'L' or '0', while a defect-free cell can be switched to '1'. Hence, the subsequent r1 will output '1' for a defect-free cell and '0' for a defective cell.

The $\hat{\text{w}}$ operations can also be included in a march algorithm. To do that, we analyze the faults that are sensitized by the set $\text{S}_{EtD-sHtD,int/cont}$. For example, S = 1w0 sensitizes W0TF1 EtD faults as well as the sHtD faults W0TFU and W0TFL. The sHtD faults can be detected by replacing the regular write operation w by $\hat{\text{w}}$ and performing a subsequent read operation. A similar argumentation applies for the remaining sensitizing sequences in $\text{S}_{EtD-sHtD,int/cont}$. This leads to the following march algorithm that detects the EtD and sHtD faults with a test time of $8T_w + 5T_r$:

$$\begin{aligned}\text{March-EtD-sHtD, int/cont} = \{&\Updownarrow(\mathrm{w1});\Updownarrow(\mathrm{r1},\hat{\mathrm{w}}0,\mathrm{r0});\\ &\Updownarrow(\mathrm{w0},\mathrm{w0},\mathrm{w0},\hat{\mathrm{w}}1);\Updownarrow(\mathrm{r1},\mathrm{w0},\mathrm{r0},\hat{\mathrm{w}}1,\mathrm{r1})\}.\end{aligned} \tag{5.5}$$

5.6. ADVANTAGES AND CHALLENGES OF THE DAT APPROACH

5.6.1. ADVANTAGES

The DAT approach introduces the following advantages.

Reduction of Test Escapes and Test Quality Improvement Because the DAT approach models the actual physics of a defect, realistic faults will be sensitized. Subsequently, only these faults are included in a test. This increases the quality of the test, reduces the number of test escapes, and minimizes the test time.

Fast Yield Learning Thanks to the fact that there is a single defect model per defect, rather than a linear resistor that lumps several defects at once, it is possible to make a fingerprint of faults for each defects. This fingerprint can be used for faster yield learning, as it points immediately and uniquely to a single defect.

Generic and Scalable The approach is generic and can be applied for any IC technology, as long as the physics of the device and the defect are understood. For example, Wu, Taouil, Rao, *et al.* have used the DAT approach to model pinhole defects in STT-MRAM [155]. Next to that, the method is scalable, because once a defect model is developed, it can be re-used in other designs that use the same device. Also, the fault analysis and test development algorithm are flexible and easily modified. For example, if the DAT approach is used for multi-level cells, only the fault modeling needs to be updated, while defect modeling, fault analysis, and test development remain unchanged.

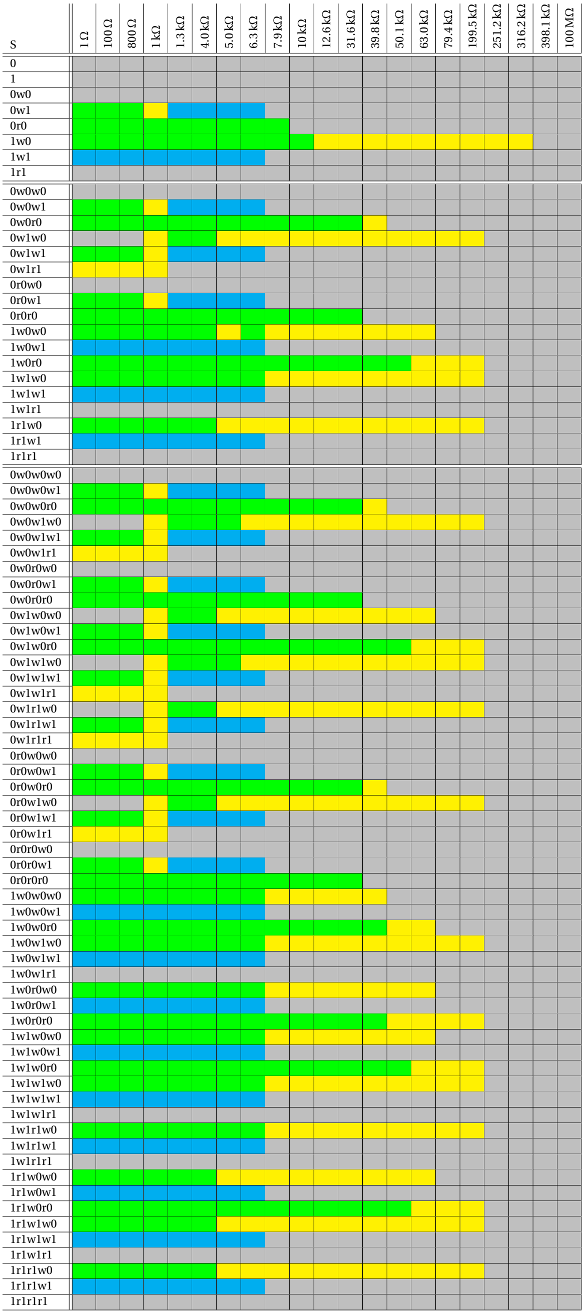
5.6.2. CHALLENGES

Besides the above benefits, there are also some challenges for the DAT approach.

Lack of Defect Data In the public domain, there are not many papers that share data of defective RRAM devices. This makes it harder to calibrate the defect models. However, even in the case when none of these data are available, the defect models can still be used, because they model the actual physics of the device.

Specialized Knowledge Defect modeling requires deep and specialized knowledge about the device that is studied. It may be difficult for some companies to find employees who can do this. This could be overcome if semiconductor manufacturing companies develop the defect models and sell these also to their customers.

Table 5.1: Fault map for $R_{\text{Br BL-int}}$ ⬜ Fault-free 🟩 EtD 🟨 HtD strong 🟦 HtD weak



S	1 Ω	100 Ω	800 Ω	1 kΩ	1.3 kΩ	4.0 kΩ	5.0 kΩ	6.3 kΩ	7.9 kΩ	10 kΩ	12.6 kΩ	31.6 kΩ	39.8 kΩ	50.1 kΩ	63.0 kΩ	79.4 kΩ	199.5 kΩ	251.2 kΩ	316.2 kΩ	398.1 kΩ	100 MΩ
0	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜
1	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜
0w0	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜
0w1	🟩	🟩	🟩	🟨	🟦	🟦	🟦	🟦	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜
0r0	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜
1w0	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	⬜	⬜
1w1	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜
1r1	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜
0w0w0	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜
0w0w1	🟩	🟩	🟩	🟨	🟦	🟦	🟦	🟦	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜
0w0r0	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟨	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜
0w1w0	⬜	⬜	⬜	🟨	🟩	🟩	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	⬜	⬜	⬜	⬜
0w1w1	🟩	🟩	🟩	🟨	🟦	🟦	🟦	🟦	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜
0w1r1	🟨	🟨	🟨	🟨	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜
0r0w0	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜
0r0w1	🟩	🟩	🟩	🟨	🟦	🟦	🟦	🟦	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜
0r0r0	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜
1w0w0	🟩	🟩	🟩	🟩	🟩	🟩	🟨	🟩	🟨	🟨	🟨	🟨	🟨	🟨	🟨	⬜	⬜	⬜	⬜	⬜	⬜
1w0w1	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜
1w0r0	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟨	🟨	🟨	⬜	⬜	⬜	⬜
1w1w0	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	⬜	⬜	⬜	⬜
1w1w1	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜
1w1r1	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜
1r1w0	🟩	🟩	🟩	🟩	🟩	🟩	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	⬜	⬜	⬜	⬜
1r1w1	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜
1r1r1	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜
0w0w0w0	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜
0w0w0w1	🟩	🟩	🟩	🟨	🟦	🟦	🟦	🟦	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜
0w0w0r0	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟨	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜
0w0w1w0	⬜	⬜	⬜	🟨	🟩	🟩	🟩	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	⬜	⬜	⬜	⬜
0w0w1w1	🟩	🟩	🟩	🟨	🟦	🟦	🟦	🟦	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜
0w0w1r1	🟨	🟨	🟨	🟨	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜
0w0r0w0	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜
0w0r0w1	🟩	🟩	🟩	🟨	🟦	🟦	🟦	🟦	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜
0w0r0r0	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜
0w1w0w0	⬜	⬜	⬜	🟨	🟩	🟩	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	⬜	⬜	⬜	⬜	⬜	⬜
0w1w0w1	🟩	🟩	🟩	🟨	🟦	🟦	🟦	🟦	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜
0w1w0r0	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟨	🟨	🟨	⬜	⬜	⬜	⬜
0w1w1w0	⬜	⬜	⬜	🟨	🟩	🟩	🟩	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	⬜	⬜	⬜	⬜
0w1w1w1	🟩	🟩	🟩	🟨	🟦	🟦	🟦	🟦	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜
0w1w1r1	🟨	🟨	🟨	🟨	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜
0w1r1w0	⬜	⬜	⬜	🟨	🟩	🟩	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	⬜	⬜	⬜	⬜
0w1r1w1	🟩	🟩	🟩	🟨	🟦	🟦	🟦	🟦	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜
0w1r1r1	🟨	🟨	🟨	🟨	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜
0r0w0w0	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜
0r0w0w1	🟩	🟩	🟩	🟨	🟦	🟦	🟦	🟦	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜
0r0w0r0	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟨	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜
0r0w1w0	⬜	⬜	⬜	🟨	🟩	🟩	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	⬜	⬜	⬜	⬜
0r0w1w1	🟩	🟩	🟩	🟨	🟦	🟦	🟦	🟦	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜
0r0w1r1	🟨	🟨	🟨	🟨	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜
0r0r0w0	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜
0r0r0w1	🟩	🟩	🟩	🟨	🟦	🟦	🟦	🟦	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜
0r0r0r0	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜
1w0w0w0	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟨	🟨	🟨	🟨	🟨	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜
1w0w0w1	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜
1w0w0r0	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟨	🟨	⬜	⬜	⬜	⬜	⬜	⬜
1w0w1w0	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	⬜	⬜	⬜	⬜
1w0w1w1	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜
1w0w1r1	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜
1w0r0w0	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟨	🟨	🟨	🟨	🟨	🟨	🟨	⬜	⬜	⬜	⬜	⬜	⬜
1w0r0w1	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜
1w0r0r0	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟨	🟨	🟨	🟨	⬜	⬜	⬜	⬜
1w1w0w0	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟨	🟨	🟨	🟨	🟨	🟨	🟨	⬜	⬜	⬜	⬜	⬜	⬜
1w1w0w1	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜
1w1w0r0	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟨	🟨	🟨	⬜	⬜	⬜	⬜
1w1w1w0	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	⬜	⬜	⬜	⬜
1w1w1w1	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜
1w1w1r1	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜
1w1r1w0	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	⬜	⬜	⬜	⬜
1w1r1w1	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜
1w1r1r1	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜
1r1w0w0	🟩	🟩	🟩	🟩	🟩	🟩	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	⬜	⬜	⬜	⬜	⬜	⬜
1r1w0w1	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜
1r1w0r0	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟨	🟨	🟨	⬜	⬜	⬜	⬜
1r1w1w0	🟩	🟩	🟩	🟩	🟩	🟩	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	⬜	⬜	⬜	⬜
1r1w1w1	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜
1r1w1r1	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜
1r1r1w0	🟩	🟩	🟩	🟩	🟩	🟩	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	⬜	⬜	⬜	⬜
1r1r1w1	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜
1r1r1r1	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜

Table 5.2: Part 1: Sensitized fault classes (FCs) per interconnection defect. Black FCs are EtD, red FCs are sHtD.

S	$R_{\text{Br BL-SL}}$	$R_{\text{Br BL-WL}}$	$R_{\text{Br BL-int}}$	$R_{\text{Br SL-WL}}$	$R_{\text{Br SL-int}}$	$R_{\text{Br WL-int}}$	$R_{\text{Op BL}}$	$R_{\text{Op SL}}$	$R_{\text{Op WL}}$
0									
1									
0w0									
0w1									
0r0									
1w0	⟨S/U/⟩-1, ⟨S/U/⟩-2		⟨S/U/⟩-1, ⟨S/U/⟩-2				⟨S/1/⟩-1, ⟨S/U/⟩-2		
1w1									
1r1		⟨S/1/0⟩-1		⟨S/1/0⟩-1		⟨S/1/0⟩-1	⟨S/1/0⟩-1,	⟨S/1/0⟩-1	⟨S/1/0⟩-1
0w0w0									
0w0w1									
0w0r0	⟨S/0/1⟩-1,								
0w1w0	⟨S/U/⟩-1,		⟨S/U/⟩-1,				⟨S/U/⟩-1,		
0w1w1									
0w1r1				⟨S/1/0⟩-1		⟨S/1/0⟩-1		⟨S/1/0⟩-1	⟨S/1/0⟩-1
0r0w0									
0r0w1									
0r0r0	⟨S/0/1⟩-1,								
1w0w0			⟨S/U/⟩-1,				⟨S/U/⟩-1,		
1w0w1									
1w0r0	⟨S/U/1⟩-1, ⟨S/U/0⟩-2		⟨S/U/1⟩-1,				⟨S/U/0⟩-1, ⟨S/U/0⟩-2		
1w1w0	⟨S/U/⟩-1,		⟨S/U/⟩-1,				⟨S/U/⟩-1,		
1w1w1									
1w1r1				⟨S/1/0⟩-1		⟨S/1/0⟩-1		⟨S/1/0⟩-1	⟨S/1/0⟩-1
1r1w0	⟨S/U/⟩-1,		⟨S/U/⟩-1,				⟨S/U/⟩-1,		
1r1w1									
1r1r1				⟨S/1/0⟩-1		⟨S/1/0⟩-1		⟨S/1/0⟩-1	
0w0w0w0					⟨S/L/⟩-1				
0w0w0w1									
0w0w0r0	⟨S/0/1⟩-1,								
0w0w1w0	⟨S/U/⟩-1,		⟨S/U/⟩-1,				⟨S/U/⟩-1,		
0w0w1w1									
0w0w1r1				⟨S/1/0⟩-1		⟨S/1/0⟩-1		⟨S/1/0⟩-1	⟨S/1/0⟩-1
0w0r0w0									
0w0r0w1									
0w0r0r0	⟨S/0/?⟩-1,								
0w1w0w0			⟨S/U/⟩-1,				⟨S/U/⟩-1,		
0w1w0w1									
0w1w0r0	⟨S/U/1⟩-1,		⟨S/U/1⟩-1,				⟨S/U/0⟩-1,		
0w1w1w0	⟨S/U/⟩-1,		⟨S/U/⟩-1,				⟨S/U/⟩-1,		
0w1w1w1									
0w1w1r1				⟨S/1/0⟩-1		⟨S/1/0⟩-1		⟨S/1/0⟩-1	⟨S/1/0⟩-1
0w1r1w0	⟨S/U/⟩-1,		⟨S/U/⟩-1,				⟨S/U/⟩-1,		
0w1r1w1									
0w1r1r1		⟨S/1/0⟩-1		⟨S/1/0⟩-1		⟨S/1/0⟩-1		⟨S/1/0⟩-1	
0r0w0w0									
0r0w0w1									
0r0w0r0	⟨S/0/1⟩-1,								
0r0w1w0	⟨S/U/⟩-1,		⟨S/U/⟩-1,				⟨S/U/⟩-1,		
0r0w1w1									
0r0w1r1				⟨S/1/0⟩-1		⟨S/1/0⟩-1		⟨S/1/0⟩-1	⟨S/1/0⟩-1
0r0r0w0									
0r0r0w1									
0r0r0r0	⟨S/0/1⟩-1,								
1w0w0w0					⟨S/L/⟩-1		⟨S/U/⟩-1,		
1w0w0w1									
1w0w0r0	⟨S/0/1⟩-1,		⟨S/U/0⟩-1,				⟨S/U/0⟩-1,		
1w0w1w0	⟨S/U/⟩-1,		⟨S/U/⟩-1,				⟨S/U/⟩-1,		
1w0w1w1									
1w0w1r1				⟨S/1/0⟩-1		⟨S/1/0⟩-1		⟨S/1/0⟩-1	⟨S/1/0⟩-1
1w0r0w0			⟨S/U/⟩-1,				⟨S/U/⟩-1,		
1w0r0w1									
1w0r0r0	⟨S/U/1⟩-1, ⟨S/U/0⟩-2		⟨S/U/0⟩-1,				⟨S/U/0⟩-1, ⟨S/U/0⟩-2		
1w1w0w0			⟨S/U/⟩-1,				⟨S/U/⟩-1,		
1w1w0w1									
1w1w0r0	⟨S/U/1⟩-1,		⟨S/U/1⟩-1,				⟨S/U/0⟩-1,		
1w1w1w0	⟨S/U/⟩-1,		⟨S/U/⟩-1,				⟨S/U/⟩-1,		
1w1w1w1									
1w1w1r1				⟨S/1/0⟩-1		⟨S/1/0⟩-1		⟨S/1/0⟩-1	⟨S/1/0⟩-1
1w1r1w0	⟨S/U/⟩-1,		⟨S/U/⟩-1,				⟨S/U/⟩-1,		
1w1r1w1									
1w1r1r1		⟨S/1/0⟩-1		⟨S/1/0⟩-1		⟨S/1/0⟩-1		⟨S/1/0⟩-1	
1r1w0w0			⟨S/U/⟩-1,				⟨S/U/⟩-1,		
1r1w0w1									
1r1w0r0	⟨S/U/1⟩-1,		⟨S/U/1⟩-1,				⟨S/U/0⟩-1,		
1r1w1w0	⟨S/U/⟩-1,		⟨S/U/⟩-1,				⟨S/U/⟩-1,		
1r1w1w1									
1r1w1r1				⟨S/1/0⟩-1		⟨S/1/0⟩-1		⟨S/1/0⟩-1	⟨S/1/0⟩-1
1r1r1w0	⟨S/U/⟩-1,		⟨S/U/⟩-1,				⟨S/U/⟩-1,		
1r1r1w1									
1r1r1r1		⟨S/1/0⟩-1		⟨S/1/0⟩-1		⟨S/1/0⟩-1		⟨S/1/0⟩-1	

Table 5.3: Part 2: Sensitized fault classes (FCs) per interconnection defect. Black FCs are EtD, red FCs are sHtD.

S	$R_{\text{Sh BL-GND}}$	$R_{\text{Sh BL-}V_{DD}}$	$R_{\text{Sh SL-GND}}$	$R_{\text{Sh SL-}V_{DD}}$	$R_{\text{Sh WL-GND}}$	$R_{\text{Sh WL-}V_{DD}}$	$R_{\text{Sh int-GND}}$	$R_{\text{Sh int-}V_{DD}}$
0								
1								
0w0							⟨S/U/⟩-1,	
0w1								
0r0								
1w0			⟨S/U/⟩-1, ⟨S/U/⟩-2		⟨S/1/⟩-1, ⟨S/U/⟩-2		⟨S/U/⟩-1,	
1w1								
1r1		⟨S/1/0⟩-1		⟨S/1/0⟩-1		⟨S/L/0⟩-1		⟨S/1/0⟩-1
0w0w0							⟨S/U/⟩-1,	
0w0w1								
0w0r0	⟨S/0/1⟩-1						⟨S/U/0⟩-1,	
0w1w0			⟨S/U/⟩-1,		⟨S/U/⟩-1,		⟨S/U/⟩-1,	
0w1w1								
0w1r1		⟨S/1/0⟩-1		⟨S/1/0⟩-1				⟨S/1/0⟩-1
0r0w0							⟨S/U/⟩-1,	
0r0w1								
0r0r0	⟨S/0/1⟩-1							
1w0w0			⟨S/U/⟩-1,		⟨S/U/⟩-1,		⟨S/U/⟩-1,	
1w0w1								
1w0r0	⟨S/0/1⟩-1		⟨S/U/1⟩-1,		⟨S/1/1⟩-1, ⟨S/U/0⟩-2	⟨S/L/0⟩-1	⟨S/U/0⟩-1, ⟨S/U/0⟩-2	
1w1w0			⟨S/U/⟩-1,		⟨S/U/⟩-1,		⟨S/U/⟩-1,	
1w1w1								
1w1r1		⟨S/1/0⟩-1		⟨S/1/0⟩-1				⟨S/1/0⟩-1
1r1w0			⟨S/U/⟩-1,		⟨S/U/⟩-1,	⟨S/L/⟩-1	⟨S/1/⟩-1, ⟨S/U/⟩-2	
1r1w1								
1r1r1		⟨S/1/0⟩-1		⟨S/1/0⟩-1		⟨S/L/0⟩-1		⟨S/1/0⟩-1
0w0w0w0							⟨S/U/⟩-1,	
0w0w0w1								
0w0w0r0	⟨S/0/1⟩-1						⟨S/U/0⟩-1,	
0w0w1w0			⟨S/U/⟩-1,		⟨S/U/⟩-1,		⟨S/U/⟩-1,	
0w0w1w1								
0w0w1r1		⟨S/1/0⟩-1		⟨S/1/0⟩-1				⟨S/1/0⟩-1
0w0r0w0							⟨S/U/⟩-1,	
0w0r0w1								
0w0r0r0	⟨S/0/1⟩-1						⟨S/U/0⟩-1,	
0w1w0w0			⟨S/U/⟩-1,		⟨S/U/⟩-1,		⟨S/U/⟩-1,	
0w1w0w1								
0w1w0r0	⟨S/0/1⟩-1		⟨S/U/?⟩-1,		⟨S/U/?⟩-1,		⟨S/U/0⟩-1,	
0w1w1w0			⟨S/U/⟩-1,		⟨S/U/⟩-1,		⟨S/U/⟩-1,	
0w1w1w1								
0w1w1r1		⟨S/1/0⟩-1		⟨S/1/0⟩-1				⟨S/1/0⟩-1
0w1r1w0			⟨S/U/⟩-1,		⟨S/U/⟩-1,		⟨S/U/⟩-1,	
0w1r1w1								
0w1r1r1		⟨S/1/0⟩-1		⟨S/1/0⟩-1				⟨S/1/0⟩-1
0r0w0w0							⟨S/U/⟩-1,	
0r0w0w1								
0r0w0r0	⟨S/0/1⟩-1						⟨S/U/0⟩-1,	
0r0w1w0			⟨S/U/⟩-1,		⟨S/U/⟩-1,		⟨S/U/⟩-1,	
0r0w1w1								
0r0w1r1		⟨S/1/0⟩-1		⟨S/1/0⟩-1				⟨S/1/0⟩-1
0r0r0w0							⟨S/U/⟩-1,	
0r0r0w1								
0r0r0r0	⟨S/0/1⟩-1							
1w0w0w0			⟨S/U/⟩-1,		⟨S/U/⟩-1,		⟨S/U/⟩-1,	
1w0w0w1								
1w0w0r0	⟨S/0/1⟩-1		⟨S/U/0⟩-1,		⟨S/U/1⟩-1,	⟨S/L/0⟩-1	⟨S/U/0⟩-1,	
1w0w1w0			⟨S/U/⟩-1,		⟨S/U/⟩-1,		⟨S/U/⟩-1,	
1w0w1w1								
1w0w1r1		⟨S/1/0⟩-1		⟨S/1/0⟩-1				⟨S/1/0⟩-1
1w0r0w0			⟨S/U/⟩-1,		⟨S/U/⟩-1,	⟨S/L/⟩-1	⟨S/U/⟩-1,	
1w0r0w1								
1w0r0r0	⟨S/0/1⟩-1		⟨S/U/1⟩-1,		⟨S/1/1⟩-1, ⟨S/U/0⟩-2	⟨S/L/0⟩-1	⟨S/U/0⟩-1, ⟨S/U/0⟩-2	
1w1w0w0			⟨S/U/⟩-1,		⟨S/U/⟩-1,		⟨S/U/⟩-1,	
1w1w0w1								
1w1w0r0	⟨S/0/1⟩-1		⟨S/U/?⟩-1,		⟨S/U/1⟩-1,		⟨S/U/0⟩-1,	
1w1w1w0			⟨S/U/⟩-1,		⟨S/U/⟩-1,		⟨S/U/⟩-1,	
1w1w1w1								
1w1w1r1		⟨S/1/0⟩-1		⟨S/1/0⟩-1				⟨S/1/0⟩-1
1w1r1w0			⟨S/U/⟩-1,		⟨S/U/⟩-1,		⟨S/U/⟩-1,	
1w1r1w1								
1w1r1r1		⟨S/1/0⟩-1		⟨S/1/0⟩-1				⟨S/1/0⟩-1
1r1w0w0			⟨S/U/⟩-1,		⟨S/U/⟩-1,	⟨S/L/⟩-1	⟨S/1/⟩-1,	
1r1w0w1								
1r1w0r0	⟨S/0/1⟩-1		⟨S/U/?⟩-1,		⟨S/U/?⟩-1,	⟨S/L/0⟩-1	⟨S/1/1⟩-1, ⟨S/U/0⟩-2	
1r1w1w0			⟨S/U/⟩-1,		⟨S/U/⟩-1,		⟨S/1/⟩-1,	
1r1w1w1								
1r1w1r1		⟨S/1/0⟩-1		⟨S/1/0⟩-1				⟨S/1/0⟩-1
1r1r1w0			⟨S/U/⟩-1,		⟨S/U/⟩-1,	⟨S/L/⟩-1	⟨S/1/⟩-1, ⟨S/U/⟩-2	
1r1r1w1								
1r1r1r1		⟨S/1/0⟩-1		⟨S/1/0⟩-1		⟨S/L/0⟩-1		⟨S/1/0⟩-1

Table 5.4: Example to illustrate the test development algorithm

			S						$\sum_{i=1}^{p} a_{i,j}$
			1	2	…	i	…	p	
			0r0	1r1	0w0	0w1	1w0	1w1	
Defect 1	1	1 Ω	1	0	0	1	1	0	3
	2	10 Ω	1	0	0	1	1	0	3
	3	100 Ω	1	0	0	0	0	0	1
	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
	$j-1$	1 MΩ	1	0	0	0	0	0	1
	j	10 MΩ	0	1	0	0	0	0	1
	$j+1$	100 MΩ	0	1	0	0	0	1	2
Defect 2	$j+2$	1 Ω	0	1	0	0	0	0	1
	$j+3$	10 Ω	0	1	0	0	0	0	1
	$j+4$	100 Ω	0	0	0	0	1	0	1
	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
	$q-2$	1 MΩ	0	0	0	0	1	0	1
	$q-1$	10 MΩ	0	0	1	1	0	0	2
	q	100 MΩ	0	0	1	1	0	1	3
	$\mathrm{S}_{\mathrm{selected}}$		1	1	1	0	1	0	

6

DEVICE-AWARE TEST FOR FORMING DEFECTS

During the CF formation, two defects may occur: over-forming and under-forming. In the former case, the CF grows too wide and the device-resistance is too low, while in the latter case the CF is too thin and the resistance is too high. If the defect is even more severe, the RRAM device is unable to switch. Clearly, forming defects are of prime importance for RRAM. This chapter applies the DAT approach to model the forming defects, performing fault analysis with the obtained defect models, and develops a test for it. It also compares the DAT approach with the traditional approach and shows that the DAT approach is superior.

This chapter is partially based on [28], [29].

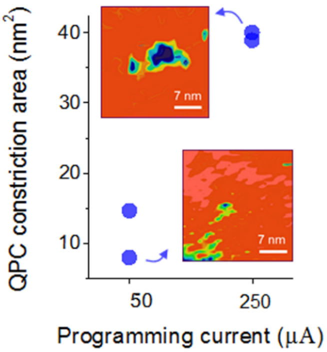


Figure 6.1: Thinnest point of a CF for two forming currents [36]

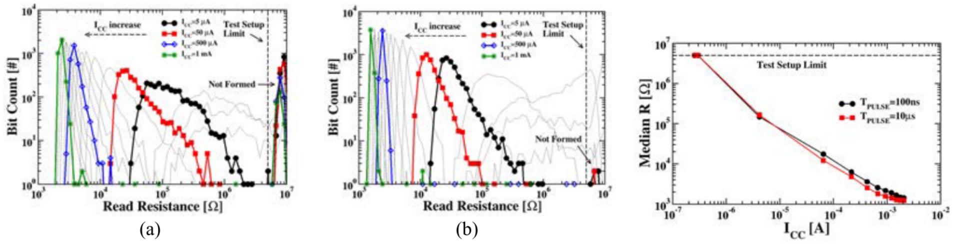
6.1. DEFECT MECHANISM AND CHARACTERIZATION

6.1.1. DEFECT MECHANISM

After manufacturing, there is no CF in the oxide present and it needs to be formed during the forming process. This process works as follows. A voltage is applied over the device that causes an electric field inside the oxide. When this electric filed passes a certain strength, the oxygen and metal ions break apart, and the negatively charged oxygen ions are attracted by the electric field, leaving behind a vacancy. This vacancy can carry a current, and thus a current starts to flow through the oxide. The current heats the oxide, which enhances the bond breaking process even further and thus results in even more bond breaking [156], [157]. This positive feedback loop only stops when the applied voltage is lowered or removed completely. This can be done by limiting the duration of the forming pulse, or by limiting the current through the device, e.g., with a diode or a transistor. If the process is not stopped in time, over-forming will occur, which causes the CF to grow too wide. This leads to an extremely low device resistance. Alternatively, the forming process may be stopped too early. This will leave the CF thin, and the device resistance too high. Fig. 6.1 shows the CF for two forming (programming) currents at its thinnest point [36]. It can be seen that the current has a strong impact on the shape of the CF.

6.1.2. CHARACTERIZATION

Fig. 6.2a shows the resistance distributions of multiple cells directly after forming with two different pulse lengths and varying forming current [113]. The figure shows indeed that higher forming currents lead to lower device resistances as well as less variation between the different devices. Fig. 6.2b shows the mean resistance after forming for varying forming currents [113]. Again, it can be seen that higher forming currents result in lower device resistances.



(a) Resistance distribution after forming for two forming pulse lengths: (a) 100 ns and (b) 10 μs [113]

(b) Mean resistance after forming for two forming pulse lengths [113]

Figure 6.2: Resistances after forming

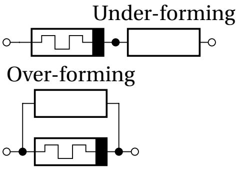


Figure 6.3: Traditional defect modeling of forming defects

6.2. TRADITIONAL DEFECT MODELING APPROACH

The traditional defect modeling approach models forming defects using linear resistors in series or in parallel with the defect-free device, as was shown in Chapter 4. Fig. 6.3 shows how forming defects could be modeled using this approach. Over-forming defects reduce the device resistance, and thus could be modeled with a parallel resistor, while under-forming defects increase the device resistance, and thus could be modeled with a series resistor.

In Chapter 5, we presented the faults analysis results for all interconnect defects in a 1T1R cell, including the series and parallel resistor that are now used to model the forming defects, $\mathrm{R_{Op\,BL}}$ and $\mathrm{R_{Br\,BL\text{-}int}}$, respectively. The fault classes that are sensitized with these defect models are listed in Table 6.1. From the Table it follows that ⟨1w0r0/U/1⟩ for $\mathrm{R_{Br\,BL\text{-}int}}$, and ⟨1w0/1/-⟩ for $\mathrm{R_{Op\,BL}}$ sensitize the EtD faults that cover the widest defect range. Similarly, ⟨1w0/U/-⟩ for $\mathrm{R_{Br\,BL\text{-}int}}$, and ⟨1w0/1/-⟩ or ⟨1r1/1/0⟩ for $\mathrm{R_{Op\,BL}}$ sensitize the sHtD faults that cover the widest defect range. Based on this analysis, we can develop two March algorithms that detect these defects. Note that for the sHtD test, we again apply the weak write DFT from [13].

$$\text{March-LR, forming, EtD} = \{\Updownarrow(\text{w1});\Updownarrow(\text{w0},\text{r0})\}, \tag{6.1}$$

$$\text{March-LR, forming, sHtD} = \{\Updownarrow(\text{w1});\Updownarrow(\text{r1},\hat{\text{w}}0,\text{r0})\}. \tag{6.2}$$

These tests have test time of $2T_w + 1T_r$ and $2T_w + 2t_r$, respectively.

6.3. DEVICE-AWARE DEFECT MODELING

We will now apply the DAT approach to model the forming defects based on the forming current (I_{form}). In Chapter 5, we have shown that this step consists of three substeps:

Table 6.1: Fault classes sensitized by linear resistor forming defect modeling. EtD FCs are black, while sHtD FCs are red.

S	$R_{\text{Br BL-int}}$	$R_{\text{Op BL}}$	S	$R_{\text{Br BL-int}}$	$R_{\text{Op BL}}$	S	$R_{\text{Br BL-int}}$	$R_{\text{Op BL}}$
0			0w0w0w1			1w0w0w1		
1			0w0w0r0			1w0w0r0	⟨S/U/0⟩-1,	⟨S/U/0⟩-1,
0w0			0w0w1w0	⟨S/U/⟩-1,	⟨S/U/⟩-1,	1w0w1w0	⟨S/U/⟩-1,	⟨S/U/⟩-1,
0w1			0w0w1w1			1w0w1w1		
0r0			0w0w1r1			1w0w1r1		
1w0	⟨S/U/⟩-1, ⟨S/U/⟩-2	⟨S/1/⟩-1, ⟨S/U/⟩-2	0w0r0w0			1w0r0w0	⟨S/U/⟩-1,	⟨S/U/⟩-1,
1w1			0w0r0w1			1w0r0w1		
1r1		⟨S/1/0⟩-1,	0w0r0r0			1w0r0r0	⟨S/U/0⟩-1,	⟨S/U/0⟩-1, ⟨S/U/0⟩-2
0w0w0			0w1w0w0	⟨S/U/⟩-1,	⟨S/U/⟩-1,	1w1w0w0	⟨S/U/⟩-1,	⟨S/U/⟩-1,
0w0w1			0w1w0w1			1w1w0w1		
0w0r0			0w1w0r0	⟨S/U/1⟩-1,	⟨S/U/0⟩-1,	1w1w0r0	⟨S/U/1⟩-1,	⟨S/U/0⟩-1,
0w1w0	⟨S/U/⟩-1,	⟨S/U/⟩-1,	0w1w1w0	⟨S/U/⟩-1,	⟨S/U/⟩-1,	1w1w1w0	⟨S/U/⟩-1,	⟨S/U/⟩-1,
0w1w1			0w1w1w1			1w1w1w1		
0w1r1			0w1w1r1			1w1w1r1		
0r0w0			0w1r1w0	⟨S/U/⟩-1,	⟨S/U/⟩-1,	1w1r1w0	⟨S/U/⟩-1,	⟨S/U/⟩-1,
0r0w1			0w1r1w1			1w1r1w1		
0r0r0			0w1r1r1			1w1r1r1		
1w0w0	⟨S/U/⟩-1,	⟨S/U/⟩-1,	0r0w0w0			1r1w0w0	⟨S/U/⟩-1,	⟨S/U/⟩-1,
1w0w1			0r0w0w1			1r1w0w1		
1w0r0	⟨S/U/1⟩-1,	⟨S/U/0⟩-1, ⟨S/U/0⟩-2	0r0w0r0			1r1w0r0	⟨S/U/1⟩-1,	⟨S/U/0⟩-1,
1w1w0	⟨S/U/⟩-1,	⟨S/U/⟩-1,	0r0w1w0	⟨S/U/⟩-1,	⟨S/U/⟩-1,	1r1w1w0	⟨S/U/⟩-1,	⟨S/U/⟩-1,
1w1w1			0r0w1w1			1r1w1w1		
1w1r1			0r0w1r1			1r1w1r1		
1r1w0	⟨S/U/⟩-1,	⟨S/U/⟩-1,	0r0r0w0			1r1r1w0	⟨S/U/⟩-1,	⟨S/U/⟩-1,
1r1w1			0r0r0w1			1r1r1w1		
1r1r1			0r0r0r0			1r1r1r1		
0w0w0w0			1w0w0w0		⟨S/U/⟩-1,			

physical defect modeling, electrical defect modeling, and fitting and model optimization.

6.3.1. PHYSICAL DEFECT MODELING:

The resistance of a RRAM device is mainly affected by the length of the tunneling gap l_{gap} and the width of the top of the filament ϕ_{T} [36], [158]. Therefore, the effective values ($l_{\text{gap,eff}}$ and $\phi_{\text{T,eff}}$) for these parameters have to be determined under influence of I_{form}. This is described as:

$$l_{\text{gap,eff}} = \frac{p_1 \cdot R_\mu \left(I_{\text{form}}\right)^2 + p_2 \cdot R_\mu \left(I_{\text{form}}\right) + p_3}{R_\mu \left(I_{\text{form}}\right)^2 + p_4 \cdot R_\mu \left(I_{\text{form}}\right) + p_5} \cdot p_6 + p_7, \tag{6.3}$$

$$\phi_{\text{T,eff}} = \frac{q_1 \cdot R_\mu \left(I_{\text{form}}\right)^2 + q_2 \cdot R_\mu \left(I_{\text{form}}\right) + q_3}{R_\mu \left(I_{\text{form}}\right)^2 + q_4 \cdot R_\mu \left(I_{\text{form}}\right) + q_5} \cdot q_6 + q_7. \tag{6.4}$$

In the above two equations, $R_\mu \left(I_{\text{form}}\right)$ is described by the median resistance in [113] as:

$$R_\mu \left(I_{\text{form}}\right) = r_1 + \frac{r_2}{1 + \left(r_3 \cdot I_{\text{form}}\right)^{r_4}}. \tag{6.5}$$

Note that in the above equations, $p_{1,2,3,4,5,6,7}$, $q_{1,2,3,4,5,6,7}$, and $r_{1,2,3,4}$ are fitting parameters.

6.3.2. ELECTRICAL DEFECT MODELING:

The RRAM device model in [159] takes l_{gap} and ϕ_{T} as input parameters. Hence, we can include the effective technology parameters directly in the model and observe their effects on the electrical parameters (see Table 2.1) by simulating it.

Table 6.2: Fitting parameters for the DAT forming defect model.

Parameter	p_1	p_2	p_3	p_4	p_5	p_6	p_7
Value	1895	$3.26 \cdot 10^6$	$-2.45 \cdot 10^7$	2652	$3.68 \cdot 10^5$	$1.5 \cdot 10^{-12}$	$2.00 \cdot 10^{-8}$
Parameter	q_1	q_2	q_3	q_4	q_5	q_6	q_7
Value	1895	$3.26 \cdot 10^6$	$-2.45 \cdot 10^7$	2652	$3.68 \cdot 10^5$	$-1 \cdot 10^{-11}$	$1.50 \cdot 10^{-10}$
Parameter	r_1	r_2	r_3	r_4			
Value	824.7073	$2.22 \cdot 10^{10}$	$1.39 \cdot 10^{10}$	1.049 237			

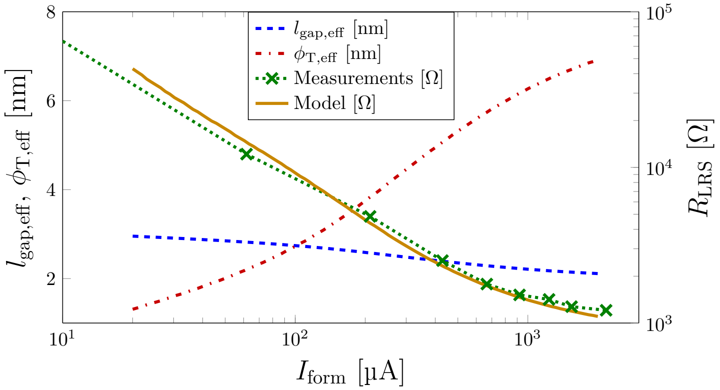


Figure 6.4: DAT forming defect model performance compared to measurements in [113].

6.3.3. FITTING AND MODEL OPTIMIZATION:

In this step, the electrical model is fitted to match the behavior of the real defective devices. We fitted the model to the measurements in [113] by changing the values of parameters $p_{1,2,3,4,5,6,7}$, $q_{1,2,3,4,5,6,7}$, and $r_{1,2,3,4}$ using MathWorks's MATLAB R2019b [151], which resulted in the values listed in Table 6.2. First, we have fitted Equation 6.5, then we analyzed how the resistance of the model in [159] is affected by changing the parameters l_{gap} and ϕ_{T}. Subsequently, we fitted Equations 6.3 and 6.4 to match to Equation 6.5. Fig. 6.4 shows the values of $l_{\text{gap,eff}}$ and $\phi_{\text{T,eff}}$, as well as the resistance after forming from [113] and the model prediction. It can be seen that higher forming currents lead to a small decrease in the tunneling gap length ($l_{\text{gap,eff}}$) and a major increase in filament width ($\phi_{\text{T,eff}}$), which is also observed by [36]. Furthermore, it can be seen that the model matches the resistive measurements from [113].

The obtained defect model describes both under and over-forming defects, because its input I_{form} is continuous. In the following, we refer to an under-forming defect if the forming current is less than the nominal forming current, while we refer to an over-forming defect if the forming current is more than the nominal current.

6.4. DEVICE-AWARE FAULT ANALYSIS

We perform fault analysis using the DAT defect model that was developed in the previous section. We replace the defect-free RRAM device in a 1T1R cell with the defect model in the architecture from Fig. 2.7. Subsequently, we sweep I_{form} and apply all sensitizing

(a) $\mathrm{R}_{\mathrm{Op\ BL}}$ and under-forming defect

(b) $\mathrm{R}_{\mathrm{Br\ BL\text{-}int}}$ and over-forming defect

Figure 6.5: Number of faults sensitized by traditional and DAT forming defect modeling for $0 \leq n \leq 3$

sequences with $n_{\mathrm{max}} = 3$. The resulting fault maps are shown in Table 6.3 for the under-forming defects, and in Table 6.4 for the over-forming defect. A gray cell indicates fault-free behavior, a green cell an EtD fault, a yellow cell and sHtD fault, and a cyan cell a wHtD fault. Table 6.5 lists the fault classes that are sensitized using this defect model.

The fault maps show that the defects mainly sensitize sHtD faults. These are all related to cells being in an unwanted state, e.g., ⟨0w0/L/-⟩ or ⟨0w1/H/-⟩. This is expected, as lower forming currents will lead to higher device resistances than desired, and higher forming currents lead to lower device resistances. Furthermore, Table 6.4 shows that EtD faults are only sensitized when $n \geq 3$, thus again proving that extending the length of S can lead to sensitization of more EtD faults.

We can conclude that forming defects cannot be properly modeled with linear resistor defect models by comparing the fault analysis results from the defects $\mathrm{R}_{\mathrm{Br\ BL\text{-}int}}$ and $\mathrm{R}_{\mathrm{Op\ BL}}$ in Table 6.1 with those of the forming defects in Table 6.5. Since forming defects manifest themselves as a lowered device resistance, one could think that this might be modeled using the $\mathrm{R}_{\mathrm{Op\ BL}}$ or $\mathrm{R}_{\mathrm{Br\ BL\text{-}int}}$ defect models. However, from the table it follows that this is not sufficient, as these defect models are unable to show the RRAM cells switch into the ‘L’ or ‘H’ state. This is to be expected, because the defect does not affect the properties of the RRAM device directly, but rather its surrounding interconnections and contacts. Fig. 6.5b shows the amount of faults that are sensitized using these two different defect models for $0 \leq n \leq 3$. It becomes clear that only a limited number of faults overlaps. Without using device-aware defect models and only the $\mathrm{R}_{\mathrm{Op\ BL}}$ defect model, only 36 *realistic* faults for this defect will be included in the test, while 73 *unrealistic* faults are included as well, leading to test overhead. Furthermore, the test will not detect 106 realistic faults, leading to *test escapes*. Similarly, without using device-aware defect models and only the $\mathrm{R}_{\mathrm{Br\ BL\text{-}int}}$ defect model, only 16 *realistic* faults for this defect will be included in the test, while 87 *unrealistic* faults are included as well, leading to a test overhead. Furthermore, the test will not detect 36 realistic faults, leading to *test escapes*.

6.5. DEVICE-AWARE TEST DEVELOPMENT

For the forming defect test development, we follow again the DAT test development algorithm. First, we present a test for only the EtD faults. Second, we present a test for both the EtD and sHtD faults. Subsequently, we compare the generated test with those generated by the traditional test approach.

6.5.1. TEST FOR EtD FAULTS

For this test, we again solve the ILP problem for the under-forming and over-forming defect model results. It follows that the set $\mathrm{S}_{EtD,forming} \in \{1\mathrm{r}1, 0\mathrm{w}1\mathrm{w}0\mathrm{r}0\}$ is able to sensitize and detect all EtD forming faults. These two can be included in a march algorithm in the following way with a test time of $3T_w + 2T_r$:

$$\text{March-EtD, forming} = \left\{\Updownarrow(\mathrm{w}1, \mathrm{r}1)\,; \Updownarrow(\mathrm{w}1, \mathrm{w}0, \mathrm{r}0)\right\}. \tag{6.6}$$

6.5.2. TEST FOR EtD AND STRONG HtD FAULTS

The test development procedure for the EtD and sHtD faults sensitized by forming defects is similar to that of the faults sensitized by interconnection and contact defects. From Table 6.5 it follows that almost any S is able to sensitize all FCs for the forming defects. For example, the set $\mathrm{S}_{EtD-sHtD,forming} \in \{1\mathrm{r}1, 1\mathrm{w}1, 0\mathrm{w}1, 0\mathrm{w}1\mathrm{w}0\}$ is able to sensitize all forming defect FCs. For their detection, we again make use of the weak write DfT from [13]. After the weak write operation, a read operation needs to be performed in order to detect the defective cell. This results in the following march algorithm with a test time of $4T_w + 2T_r$:

$$\text{March-EtD-sHtd, forming} = \left\{\Updownarrow(\mathrm{w}0, \mathrm{w}1, \hat{\mathrm{w}}0)\,; \Updownarrow(\mathrm{r}0)\,; \Updownarrow(\hat{\mathrm{w}}1, \mathrm{r}1)\right\}. \tag{6.7}$$

6.5.3. COMPARISON WITH TRADITIONAL APPROACH

A comparison of the tests above with the tests developed in Section 6.2 shows that the tests designed using the DAT approach have a higher test time than those developed specifically for the forming defect. However, when we analyze the detection capabilities of these tests on the forming defect FCs in Table 6.5, we observe the following. *March-LR, forming, EtD* is unable to detect any of the forming defects FCs. *March-LR, forming, sHtD* is able to detect FCs 1 and 2, for the under-forming defect, but unable to detect any of the over-forming FCs. We can conclude that modeling forming defects as linear resistors is wrong and will lead to low-quality tests that do not detect the correct faults. In contrast, DAT defect modeling allows to design tests that actually detect the forming defects.

Table 6.3: Fault map for the under-forming defects ⬜ Fault-free 🟩 EtD 🟨 HtD strong 🟦 HtD weak

S	20 µA	21 µA	28 µA	29 µA	30 µA	42 µA	44 µA	46 µA	48 µA	50 µA	53 µA	126 µA	132 µA	138 µA	145 µA	174 µA	182 µA	191 µA	200 µA	209 µA	219 µA	230 µA
0	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜
1	🟩	🟩	🟩	🟩	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	⬜
0w0	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	⬜	⬜	⬜	⬜	⬜
0w1	🟩	🟩	🟩	🟩	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	⬜
0r0	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜
1w0	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	⬜	⬜	⬜	⬜	⬜	⬜
1w1	🟩	🟩	🟩	🟩	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	⬜
1r1	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	⬜
0w0w0	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	⬜	⬜	⬜
0w0w1	🟩	🟩	🟩	🟩	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	⬜
0w0r0	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	⬜	⬜	⬜	⬜	⬜
0w1w0	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	⬜	⬜	⬜	⬜	⬜	⬜
0w1w1	🟩	🟩	🟩	🟩	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	⬜
0w1r1	🟩	🟩	🟩	🟩	🟩	🟩	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	⬜
0r0w0	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	⬜	⬜	⬜	⬜	⬜
0r0w1	🟩	🟩	🟩	🟩	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	⬜
0r0r0	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜
1w0w0	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	⬜	⬜	⬜	⬜
1w0w1	🟩	🟩	🟩	🟩	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	⬜
1w0r0	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	⬜	⬜	⬜	⬜	⬜	⬜
1w1w0	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	⬜	⬜	⬜	⬜	⬜	⬜
1w1w1	🟩	🟩	🟩	🟩	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	⬜
1w1r1	🟩	🟩	🟩	🟩	🟩	🟩	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	⬜
1r1w0	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	⬜	⬜	⬜	⬜	⬜	⬜
1r1w1	🟩	🟩	🟩	🟩	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	⬜
1r1r1	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	⬜
0w0w0w0	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	⬜	⬜
0w0w0w1	🟩	🟩	🟩	🟩	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	⬜
0w0w0r0	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	⬜	⬜	⬜
0w0w1w0	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	⬜	⬜	⬜	⬜	⬜	⬜
0w0w1w1	🟩	🟩	🟩	🟩	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	⬜
0w0w1r1	🟩	🟩	🟩	🟩	🟩	🟩	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	⬜
0w0r0w0	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	⬜	⬜	⬜
0w0r0w1	🟩	🟩	🟩	🟩	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	⬜
0w0r0r0	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	⬜	⬜	⬜	⬜	⬜
0w1w0w0	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	⬜	⬜	⬜	⬜
0w1w0w1	🟩	🟩	🟩	🟩	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	⬜
0w1w0r0	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	⬜	⬜	⬜	⬜	⬜	⬜
0w1w1w0	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	⬜	⬜	⬜	⬜	⬜	⬜
0w1w1w1	🟩	🟩	🟩	🟩	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	⬜
0w1w1r1	🟩	🟩	🟩	🟩	🟩	🟩	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	⬜
0w1r1w0	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	⬜	⬜	⬜	⬜	⬜	⬜
0w1r1w1	🟩	🟩	🟩	🟩	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	⬜
0w1r1r1	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	⬜
0r0w0w0	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	⬜	⬜	⬜
0r0w0w1	🟩	🟩	🟩	🟩	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	⬜
0r0w0r0	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	⬜	⬜	⬜	⬜	⬜
0r0w1w0	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	⬜	⬜	⬜	⬜	⬜	⬜
0r0w1w1	🟩	🟩	🟩	🟩	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	⬜
0r0w1r1	🟩	🟩	🟩	🟩	🟩	🟩	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	⬜
0r0r0w0	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	⬜	⬜	⬜	⬜	⬜
0r0r0w1	🟩	🟩	🟩	🟩	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	⬜
0r0r0r0	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜
1w0w0w0	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	⬜	⬜
1w0w0w1	🟩	🟩	🟩	🟩	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	⬜
1w0w0r0	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	⬜	⬜	⬜	⬜
1w0w1w0	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	⬜	⬜	⬜	⬜	⬜	⬜
1w0w1w1	🟩	🟩	🟩	🟩	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	⬜
1w0w1r1	🟩	🟩	🟩	🟩	🟩	🟩	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	⬜
1w0r0w0	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	⬜	⬜	⬜	⬜
1w0r0w1	🟩	🟩	🟩	🟩	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	⬜
1w0r0r0	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	⬜	⬜	⬜	⬜	⬜	⬜
1w1w0w0	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	⬜	⬜	⬜	⬜
1w1w0w1	🟩	🟩	🟩	🟩	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	⬜
1w1w0r0	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	⬜	⬜	⬜	⬜	⬜	⬜
1w1w1w0	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	⬜	⬜	⬜	⬜	⬜	⬜
1w1w1w1	🟩	🟩	🟩	🟩	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	⬜
1w1w1r1	🟩	🟩	🟩	🟩	🟩	🟩	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	⬜
1w1r1w0	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	⬜	⬜	⬜	⬜	⬜	⬜
1w1r1w1	🟩	🟩	🟩	🟩	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	⬜
1w1r1r1	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	⬜
1r1w0w0	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	⬜	⬜	⬜	⬜
1r1w0w1	🟩	🟩	🟩	🟩	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	⬜
1r1w0r0	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	⬜	⬜	⬜	⬜	⬜	⬜
1r1w1w0	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	⬜	⬜	⬜	⬜	⬜	⬜
1r1w1w1	🟩	🟩	🟩	🟩	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	⬜
1r1w1r1	🟩	🟩	🟩	🟩	🟩	🟩	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	⬜
1r1r1w0	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	⬜	⬜	⬜	⬜	⬜	⬜
1r1r1w1	🟩	🟩	🟩	🟩	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	⬜
1r1r1r1	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	⬜

Table 6.4: Fault map for the over-forming defects ⬜ Fault-free 🟩 EtD 🟨 HtD strong 🟦 HtD weak

S	230 µA	240 µA	251 µA	263 µA	276 µA	289 µA	303 µA	317 µA	332 µA	348 µA	364 µA	381 µA	399 µA	418 µA	438 µA	~	2.00 mA
0	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜
1	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜
0w0	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜
0w1	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	🟦	🟦	🟦	🟨	🟨	🟨	🟨	🟨
0r0	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜
1w0	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜
1w1	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	🟦	🟦	🟦	🟨	🟨	🟨	🟨	🟨
1r1	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜
0w0w0	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜
0w0w1	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	🟦	🟦	🟦	🟨	🟨	🟨	🟨	🟨
0w0r0	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜
0w1w0	⬜	⬜	⬜	⬜	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨
0w1w1	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	🟦	🟦	🟦	🟨	🟨	🟨	🟨	🟨
0w1r1	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	🟨	🟨	🟨	🟨	🟨
0r0w0	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜
0r0w1	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	🟦	🟦	🟦	🟨	🟨	🟨	🟨	🟨
0r0r0	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜
1w0w0	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜
1w0w1	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	🟦	🟦	🟦	🟨	🟨	🟨	🟨	🟨
1w0r0	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜
1w1w0	⬜	⬜	⬜	⬜	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨
1w1w1	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	🟦	🟦	🟦	🟨	🟨	🟨	🟨	🟨
1w1r1	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	🟨	🟨	🟨	🟨	🟨
1r1w0	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜
1r1w1	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	🟦	🟦	🟦	🟨	🟨	🟨	🟨	🟨
1r1r1	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜
0w0w0w0	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜
0w0w0w1	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	🟦	🟦	🟦	🟨	🟨	🟨	🟨	🟨
0w0w0r0	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜
0w0w1w0	⬜	⬜	⬜	⬜	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨
0w0w1w1	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	🟦	🟦	🟦	🟨	🟨	🟨	🟨	🟨
0w0w1r1	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	🟨	🟨	🟨	🟨	🟨
0w0r0w0	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜
0w0r0w1	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	🟦	🟦	🟦	🟨	🟨	🟨	🟨	🟨
0w0r0r0	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜
0w1w0w0	⬜	⬜	⬜	⬜	⬜	⬜	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨
0w1w0w1	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	🟦	🟦	🟦	🟨	🟨	🟨	🟨	🟨
0w1w0r0	⬜	⬜	⬜	⬜	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟩	🟩	🟩
0w1w1w0	⬜	⬜	⬜	⬜	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨
0w1w1w1	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	🟦	🟦	🟦	🟨	🟨	🟨	🟨	🟨
0w1w1r1	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	🟨	🟨	🟨	🟨	🟨
0w1r1w0	⬜	⬜	⬜	⬜	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨
0w1r1w1	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	🟦	🟦	🟦	🟨	🟨	🟨	🟨	🟨
0w1r1r1	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	🟨	🟨	🟨	🟨	🟨
0r0w0w0	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜
0r0w0w1	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	🟦	🟦	🟦	🟨	🟨	🟨	🟨	🟨
0r0w0r0	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜
0r0w1w0	⬜	⬜	⬜	⬜	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨
0r0w1w1	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	🟦	🟦	🟦	🟨	🟨	🟨	🟨	🟨
0r0w1r1	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	🟨	🟨	🟨	🟨	🟨
0r0r0w0	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜
0r0r0w1	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	🟦	🟦	🟦	🟨	🟨	🟨	🟨	🟨
0r0r0r0	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜
1w0w0w0	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜
1w0w0w1	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	🟦	🟦	🟦	🟨	🟨	🟨	🟨	🟨
1w0w0r0	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜
1w0w1w0	⬜	⬜	⬜	⬜	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨
1w0w1w1	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	🟦	🟦	🟦	🟨	🟨	🟨	🟨	🟨
1w0w1r1	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	🟨	🟨	🟨	🟨	🟨
1w0r0w0	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜
1w0r0w1	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	🟦	🟦	🟦	🟨	🟨	🟨	🟨	🟨
1w0r0r0	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜
1w1w0w0	⬜	⬜	⬜	⬜	⬜	⬜	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨
1w1w0w1	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	🟦	🟦	🟦	🟨	🟨	🟨	🟨	🟨
1w1w0r0	⬜	⬜	⬜	⬜	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟩	🟩	🟩
1w1w1w0	⬜	⬜	⬜	⬜	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨
1w1w1w1	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	🟦	🟦	🟦	🟨	🟨	🟨	🟨	🟨
1w1w1r1	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	🟨	🟨	🟨	🟨	🟨
1w1r1w0	⬜	⬜	⬜	⬜	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨
1w1r1w1	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	🟦	🟦	🟦	🟨	🟨	🟨	🟨	🟨
1w1r1r1	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	🟨	🟨	🟨	🟨	🟨
1r1w0w0	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜
1r1w0w1	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	🟦	🟦	🟦	🟨	🟨	🟨	🟨	🟨
1r1w0r0	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜
1r1w1w0	⬜	⬜	⬜	⬜	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨
1r1w1w1	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	🟦	🟦	🟦	🟨	🟨	🟨	🟨	🟨
1r1w1r1	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	🟨	🟨	🟨	🟨	🟨
1r1r1w0	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜
1r1r1w1	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	🟦	🟦	🟦	🟨	🟨	🟨	🟨	🟨
1r1r1r1	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜

Table 6.5: Fault classes sensitized by the forming defect model. EtD FCs are black, while sHtD FCs are red.

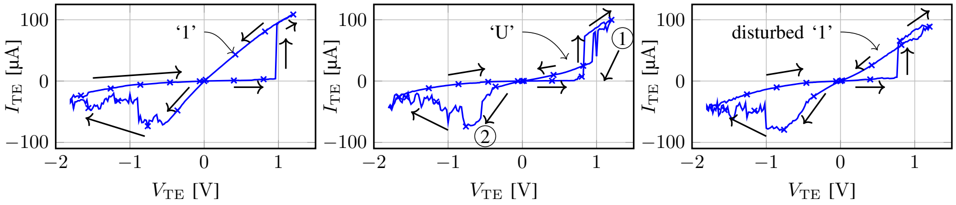
S	Under-forming	Over-forming	S	Under-forming	Over-forming	S	Under-forming	Over-forming
0	⟨S/L/⟩-1, ⟨S/L/⟩-2,		0w0w0w1	⟨S/U/⟩-1, ⟨S/U/⟩-2, ⟨S/U/⟩-3	⟨S/H/-⟩^{2}	1w0w0w1	⟨S/U/⟩-1, ⟨S/U/⟩-2, ⟨S/U/⟩-3	⟨S/H/-⟩^{2}
1	⟨S/U/⟩-1, ⟨S/U/⟩-2, ⟨S/U/⟩-3		0w0w0r0	⟨S/L/0⟩-1, ⟨S/L/0⟩-2,		1w0w0r0	⟨S/L/0⟩-1, ⟨S/L/0⟩-2,	
0w0	⟨S/L/⟩-1, ⟨S/L/⟩-2,		0w0w1w0	⟨S/L/⟩-1, ⟨S/L/⟩-2,	⟨S/U/-⟩^{1,2}	1w0w1w0	⟨S/L/⟩-1, ⟨S/L/⟩-2,	⟨S/U/-⟩^{1,2}
0w1	⟨S/U/⟩-1, ⟨S/U/⟩-2, ⟨S/U/⟩-3	⟨S/H/-⟩^{2}	0w0w1w1	⟨S/U/⟩-1, ⟨S/U/⟩-2, ⟨S/U/⟩-3	⟨S/H/-⟩^{2}	1w0w1w1	⟨S/U/⟩-1, ⟨S/U/⟩-2, ⟨S/U/⟩-3	⟨S/H/-⟩^{2}
0r0	⟨S/L/0⟩-1, ⟨S/L/0⟩-2,		0w0w1r1	⟨S/U/1⟩-1, ⟨S/U/1⟩-2, ⟨S/U/1⟩-3	⟨S/H/1⟩^{2}	1w0w1r1	⟨S/U/1⟩-1, ⟨S/U/1⟩-2, ⟨S/U/1⟩-3	⟨S/H/1⟩^{2}
1w0	⟨S/L/⟩-1, ⟨S/L/⟩-2,		0w0r0w0	⟨S/L/⟩-1, ⟨S/L/⟩-2,		1w0r0w0	⟨S/L/⟩-1, ⟨S/L/⟩-2,	
1w1	⟨S/U/⟩-1, ⟨S/U/⟩-2, ⟨S/U/⟩-3	⟨S/H/-⟩^{2}	0w0r0w1	⟨S/U/⟩-1, ⟨S/U/⟩-2, ⟨S/U/⟩-3	⟨S/H/-⟩^{2}	1w0r0w1	⟨S/U/⟩-1, ⟨S/U/⟩-2, ⟨S/U/⟩-3	
1r1	⟨S/U/0⟩-1, ⟨S/U/0⟩-2, ⟨S/U/1⟩-3		0w0r0r0	⟨S/L/0⟩-1, ⟨S/L/0⟩-2,		1w0r0r0	⟨S/L/0⟩-1, ⟨S/L/0⟩-2,	⟨S/H/-⟩^{2}
0w0w0	⟨S/L/⟩-1, ⟨S/L/⟩-2,		0w1w0w0	⟨S/L/⟩-1, ⟨S/L/⟩-2,	⟨S/U/-⟩^{2}	1w1w0w0	⟨S/L/⟩-1, ⟨S/L/⟩-2,	⟨S/U/-⟩^{2}
0w0w1	⟨S/U/⟩-1, ⟨S/U/⟩-2, ⟨S/U/⟩-3	⟨S/H/-⟩^{2}	0w1w0w1	⟨S/U/⟩-1, ⟨S/U/⟩-2, ⟨S/U/⟩-3	⟨S/H/-⟩^{2}	1w1w0w1	⟨S/U/⟩-1, ⟨S/U/⟩-2, ⟨S/U/⟩-3	⟨S/H/-⟩^{2}
0w0r0	⟨S/L/0⟩-1, ⟨S/L/0⟩-2,		0w1w0r0	⟨S/L/0⟩-1, ⟨S/L/0⟩-2,	⟨S/U/0⟩^{1}, ⟨S/U/1⟩^{2}	1w1w0r0	⟨S/L/0⟩-1, ⟨S/L/0⟩-2,	⟨S/U/0⟩^{1}, ⟨S/U/1⟩^{2}
0w1w0	⟨S/L/⟩-1, ⟨S/L/⟩-2,	⟨S/U/-⟩^{1,2}	0w1w1w0	⟨S/L/⟩-1, ⟨S/L/⟩-2,	⟨S/U/-⟩^{1,2}	1w1w1w0	⟨S/L/⟩-1, ⟨S/L/⟩-2,	⟨S/U/-⟩^{1,2}
0w1w1	⟨S/U/⟩-1, ⟨S/U/⟩-2, ⟨S/U/⟩-3	⟨S/H/-⟩^{2}	0w1w1w1	⟨S/U/⟩-1, ⟨S/U/⟩-2, ⟨S/U/⟩-3	⟨S/H/-⟩^{2}	1w1w1w1	⟨S/U/⟩-1, ⟨S/U/⟩-2, ⟨S/U/⟩-3	⟨S/H/-⟩^{2}
0w1r1	⟨S/U/1⟩-1, ⟨S/U/1⟩-2, ⟨S/U/1⟩-3	⟨S/H/1⟩^{2}	0w1w1r1	⟨S/U/1⟩-1, ⟨S/U/1⟩-2, ⟨S/U/1⟩-3	⟨S/H/1⟩^{2}	1w1w1r1	⟨S/U/1⟩-1, ⟨S/U/1⟩-2, ⟨S/U/1⟩-3	⟨S/H/1⟩^{2}
0r0w0	⟨S/L/⟩-1, ⟨S/L/⟩-2,		0w1r1w0	⟨S/L/⟩-1, ⟨S/L/⟩-2,	⟨S/U/-⟩^{1,2}	1w1r1w0	⟨S/L/⟩-1, ⟨S/L/⟩-2,	⟨S/U/-⟩^{1,2}
0r0w1	⟨S/U/⟩-1, ⟨S/U/⟩-2, ⟨S/U/⟩-3	⟨S/H/-⟩^{2}	0w1r1w1	⟨S/U/⟩-1, ⟨S/U/⟩-2, ⟨S/U/⟩-3	⟨S/H/-⟩^{2}	1w1r1w1	⟨S/U/⟩-1, ⟨S/U/⟩-2, ⟨S/U/⟩-3	⟨S/H/-⟩^{2}
0r0r0	⟨S/L/0⟩-1, ⟨S/L/0⟩-2,		0w1r1r1	⟨S/U/0⟩-1, ⟨S/U/1⟩-2, ⟨S/U/1⟩-3	⟨S/H/1⟩^{2}	1w1r1r1	⟨S/U/0⟩-1, ⟨S/U/1⟩-2, ⟨S/U/1⟩-3	⟨S/H/1⟩^{2}
1w0w0	⟨S/L/⟩-1, ⟨S/L/⟩-2,		0r0w0w0	⟨S/L/⟩-1, ⟨S/L/⟩-2,		1r1w0w0	⟨S/L/⟩-1, ⟨S/L/⟩-2,	
1w0w1	⟨S/U/⟩-1, ⟨S/U/⟩-2, ⟨S/U/⟩-3	⟨S/H/-⟩^{2}	0r0w0w1	⟨S/U/⟩-1, ⟨S/U/⟩-2, ⟨S/U/⟩-3	⟨S/H/-⟩^{2}	1r1w0w1	⟨S/U/⟩-1, ⟨S/U/⟩-2, ⟨S/U/⟩-3	⟨S/H/-⟩^{2}
1w0r0	⟨S/L/0⟩-1, ⟨S/L/0⟩-2,		0r0w0r0	⟨S/L/0⟩-1, ⟨S/L/0⟩-2,		1r1w0r0	⟨S/L/0⟩-1, ⟨S/L/0⟩-2,	
1w1w0	⟨S/L/⟩-1, ⟨S/L/⟩-2,	⟨S/U/-⟩^{1,2}	0r0w1w0	⟨S/L/⟩-1, ⟨S/L/⟩-2,	⟨S/U/-⟩^{1,2}	1r1w1w0	⟨S/L/⟩-1, ⟨S/L/⟩-2,	⟨S/U/-⟩^{1,2}
1w1w1	⟨S/U/⟩-1, ⟨S/U/⟩-2, ⟨S/U/⟩-3	⟨S/H/-⟩^{2}	0r0w1w1	⟨S/U/⟩-1, ⟨S/U/⟩-2, ⟨S/U/⟩-3	⟨S/H/-⟩^{2}	1r1w1w1	⟨S/U/⟩-1, ⟨S/U/⟩-2, ⟨S/U/⟩-3	⟨S/H/-⟩^{2}
1w1r1	⟨S/U/1⟩-1, ⟨S/U/1⟩-2, ⟨S/U/1⟩-3	⟨S/H/1⟩^{2}	0r0w1r1	⟨S/U/1⟩-1, ⟨S/U/1⟩-2, ⟨S/U/1⟩-3	⟨S/H/1⟩^{2}	1r1w1r1	⟨S/U/1⟩-1, ⟨S/U/1⟩-2, ⟨S/U/1⟩-3	⟨S/H/1⟩^{2}
1r1w0	⟨S/L/⟩-1, ⟨S/L/⟩-2,		0r0r0w0	⟨S/L/⟩-1, ⟨S/L/⟩-2,		1r1r1w0	⟨S/L/⟩-1, ⟨S/L/⟩-2,	
1r1w1	⟨S/U/⟩-1, ⟨S/U/⟩-2, ⟨S/U/⟩-3	⟨S/H/-⟩^{2}	0r0r0w1	⟨S/U/⟩-1, ⟨S/U/⟩-2, ⟨S/U/⟩-3	⟨S/H/-⟩^{2}	1r1r1w1	⟨S/U/⟩-1, ⟨S/U/⟩-2, ⟨S/U/⟩-3	⟨S/H/-⟩^{2}
1r1r1	⟨S/U/1⟩-1, ⟨S/U/1⟩-2, ⟨S/U/1⟩-3		0r0r0r0	⟨S/L/0⟩-1, ⟨S/L/0⟩-2,		1r1r1r1	⟨S/U/0⟩-1, ⟨S/U/1⟩-2, ⟨S/U/1⟩-3	
0w0w0w0	⟨S/L/⟩-1, ⟨S/L/⟩-2,		1w0w0w0	⟨S/L/⟩-1, ⟨S/L/⟩-2,				

7

DEVICE-AWARE TEST FOR INTERMITTENT UNDEFINED STATE FAULTS

This chapter details a new RRAM unique fault, named the intermittent undefined state fault. When it occurs, a cell that is normally bipolar switching suddenly is complementary switching when a SET operation is applied. The faulty behavior is intermittent, as it shows up randomly for the duration of a few cycles only. This fault can occur when there is not enough capacity to bind all the oxygen ions that break free when performing a SET operation. This can be caused by two defects: over-forming, which causes the CF to be too wide, or reduced capping layer effectiveness, which limits the oxygen binding capabilities. We model these defects and perform a fault analysis. Subsequently, we propose a test to detect this fault. We also compare the results with the traditional defect modeling approach and show that the traditional approach falls short for this fault.

This chapter is partially based on [30].



(a) Defect-free device (b) Faulty device (c) Weak device

Figure 7.1: Comparing defect-free, faulty, and weak devices

7.1. DEFECT MECHANISM AND CHARACTERIZATION

7.1.1. DEFECT MECHANISM

Chapter 2 explained the switching mechanics of bipolar and complementary switching. It is summarized here. In a bipolar switching device, the switching relies on oxygen vacancies [34], [38]. During a SET operation, oxygen ions are attracted to the capping layer or top electrode by the electric field. During a RESET operation, the ions move back in the oxide and bond again with the metal ions. In a complementary switching device, the switching relies on the movement of oxygen vacancies [39], [40]. Initially, the oxygen ions also move to the capping layer or TE. However, not all ions can be bound there and instead the oxygen vacancies start to move with the electric field, thereby breaking the CF and putting it in HRS. A negative subsequent voltage pulse will first restore the CF to LRS, before the oxygen ions move back and the CF is dissolved again. Some researchers have shown that these two switching modes can co-exist in a single device by controlling the current through the device [39], [40]. That is, higher currents lead to CS, while lower currents keep the device BS.

However, it is also possible that a BS device performs CS due to a production defect that limits the oxygen binding capacities of the device. Two such defects exist: overforming, which results in a wider CF than anticipated and thus requires more oxygen ions to be bound, or low capping layer binding capacity, e.g., due to low doping, which also limits how many oxygen ions can be bound.

7.1.2. CHARACTERIZATION

We measured the electrical characteristics of $7 \times 7 = 49$ 1T1R *BS* RRAM devices on a single wafer during 936 RESET-SET cycles. The RRAM devices are manufactured at ST Microelectronics and have the following structure (BE, oxide, cap, TE) = (TiN, HfO$_2$(10 nm), Ti(10 nm), TiN). An ST Microelectronics 130 nm technology NMOS high-voltage thick oxide transistor is placed in series to control the current through the device. The switching in a nominal defect-free device is bipolar, where logic '1' is represented by the SET state with $R_{\mathrm{SET}} < 25\,\mathrm{k\Omega}$, and logic '0' by the RESET state with $100\,\mathrm{k\Omega} < R_{\mathrm{RESET}} < 1\,\mathrm{M\Omega}$. The range $[25\,\mathrm{k\Omega}, 100\,\mathrm{k\Omega}]$ is called an *undefined state* ('U'). An illustrative measured I-V graph of a defect-free device is shown in Fig. 7.1a. Typical nominal values for V_{SET} and V_{RESET} are 0.82 V and −0.88 V.

While analyzing the measurement data, we noticed that some of the devices showed a strange switching characteristic! After a number of cycles, the switching resembles CS

Table 7.1: Occurrence probability of CS (P_{CS}) in % and maximal duration suffering from CS ($D_{\max}$) in absolute numbers.

		WL 0	WL 1	WL 2	WL 3	WL 4	WL 5	WL 6
BL 0	P_{CS}	5.983	0.427	0	0	0.427	0	0
	$D_{\max}$	3	2	0	0	1	0	0
BL 1	P_{CS}	0	1.282	0	1.175	0.107	1.709	0.641
	$D_{\max}$	0	2	0	1	1	2	2
BL 2	P_{CS}	0	0	0	0.107	0.107	0	1.282
	$D_{\max}$	0	0	0	1	1	0	1
BL 3	P_{CS}	0	0.427	0	0	0	0.107	0
	$D_{\max}$	0	1	0	0	0	1	0
BL 4	P_{CS}	0	0	0.427	0.214	2.564	0.214	0.855
	$D_{\max}$	0	0	1	1	3	1	3
BL 5	P_{CS}	0	0	2.03	0	0	0.107	0
	$D_{\max}$	0	0	2	0	0	1	0
BL 6	P_{CS}	0.427	0.107	0.427	0.321	1.389	0	0
	$D_{\max}$	1	1	1	1	3	0	0

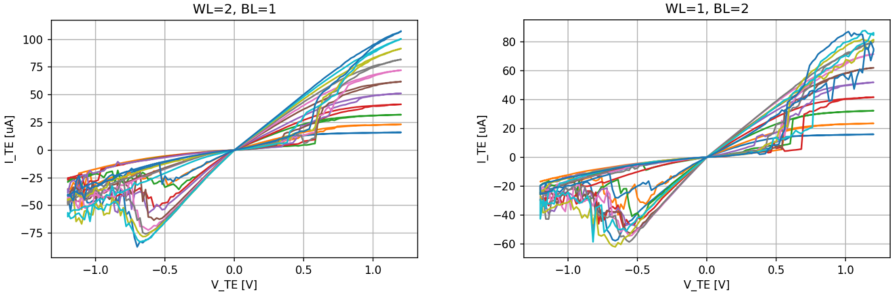
when performing SET, even though the devices are BS. The observed undesired CS in all faulty devices can be classified into two groups: *faulty* and *weak* devices. Fig. 7.1b represents the I-V graph of the faulty devices. A SET event occurs when the voltage increases from 0 V. However, as the voltage increases even further, the current through the device suddenly decreases at ①, leading to an *undesired* RESET putting the device in an *undefined* state of $R_{\text{SET}}=33\,\text{k}\Omega$. The subsequent RESET operation at ② also differs from a nominal operation, as the current first increases before decreasing, which again resembles CS. Hence, in this group of devices, a SET operation ends in undefined state. Contrarily, Fig. 7.1c represents the group of weak devices that shows a similar switching behavior, but instead the operation ends in a *disturbed '1' state*.

Table 7.1 lists the occurrence probability (P_{CS}.) of undesired CS in the measured devices (using the WL and BL to indicate them); this is the percentage of cycles in which CS occurs given a total of 936 cycles, and includes both groups of faulty and weak devices. Additionally, the table lists the maximal duration ($D_{\max}$) of the undesired behavior, expressed in the maximal number of *consecutive* cycles in which the device shows this behavior. For example, if the undesired CS occurs first in cycles 54, 55 and 56, and later in cycles 141 and 142, then $P_{\text{CS}} = 5/936 = 0.534\,\%$ and $D_{\max} = 3$ cycles. From the table it follows that undesired CS occurs in 25 devices, in up to 5.983 % of the cycles, and with a maximal duration of 3 cycles. Hence, the faulty behavior is *intermittent*, and it is a serious concern for BS RRAM devices. As the first group of faulty devices ends in *undefined state* after a SET operation, we will refer to this fault as *Intermittent Undefined State Fault (IUSF)*.

Table 7.2 lists the occurrence probability (P_{IUSF}) of the IUSF in the 49 investigated devices over all 936 RESET-SET cycles. It can be seen that depending on the defect strength, the failure rate varies from 0.107 % up to 1.068 % of the cycles, all resulting in an IUSF. Also, the probability of weak CS can be easily derived using $P_{\text{weak}} = P_{\text{CS}} - P_{\text{IUSF}}$, and can

Table 7.2: Occurrence probability (P_{IUSF}) of IUSF in %

	WL 0	WL 1	WL 2	WL 3	WL 4	WL 5	WL 6
BL 0	0.962	0.427	0	0	0.107	0	0
BL 1	0	0.107	0	0.107	0	0.427	0
BL 2	0	0	0	0	0	0	0.107
BL 3	0	0.107	0	0	0	0	0
BL 4	0	0	0.107	0	1.068	0	0
BL 5	0	0	0.321	0	0	0	0
BL 6	0	0	0.321	0	0.748	0	0



(a) Defect-free device

(b) Faulty device

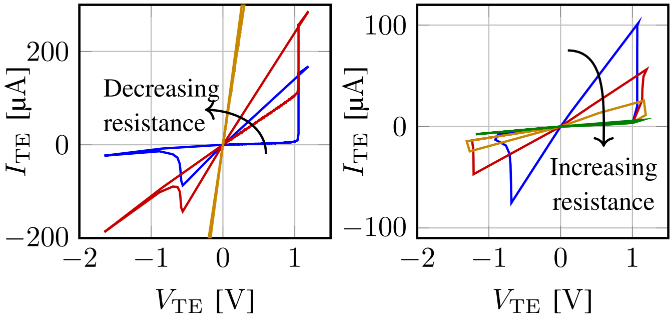
Figure 7.2: I-V curves for increasing gate voltage from 1.0 V to 2.0 V

thus be a high as 5.021 %. Note that the weak devices do not result in logic faulty behavior at time=0, but can result in reliability concerns as the weak devices may become logic faults in field, e.g., due to aging, the weak fault may become an IUSF.

Fig. 7.2 shows that the occurrence of the IUSF is indeed related to a defect that limits oxygen ion binding capability in the RRAM device. Fig. 7.2a shows a defect-free device in which the currents increase with increasing transistor gate voltage, but no IUSF occurs. In contrast, Fig. 7.2b shows that the IUSF occurs when the gate voltage is increased above 1.8 V. Because the maximal current through the transistors is the same in both figures, the CFs should be similar. Hence, the only explanation for the IUSF in Fig. 7.2b can be a defect that limits the oxygen binding capabilities.

7.2. TRADITIONAL DEFECT MODELING APPROACH

Chapter 4 explained the traditional defect modeling approach that models defects in a RRAM as linear resistor. Hence, when modeling the IUSF, there are only two possible options; the resistor can either be in parallel or in series with the device. Fig. 7.3a and Fig. 7.3b show how these defect models affect the behavior of the RRAM device for varying defect strength (i.e., the resistance of the resistor). Clearly none of these two defect models can describe the IUSF properly; a parallel resistor shows that the defective device behaves as a low ohmic resistor for low defect values, while a series resistor shows



(a) Parallel resistor (b) Series resistor

Figure 7.3: Linear resistor defect models

Table 7.3: Sensitized faults using $\text{R}_{\text{Br BL-int}}$ and $\text{R}_{\text{Op BL}}$ defect models

Defect	Defect strength	Sensitized faults
$\text{R}_{\text{Br BL-int}}$	0 Ω - 794 Ω	⟨0w1/0/-⟩
	794 Ω-1 kΩ	⟨0w1/U/-⟩
	1 kΩ-6.3 kΩ	Weak fault
	6.3 kΩ-∞	Fault-free
$\text{R}_{\text{Op BL}}$	0 Ω-5 kΩ	Fault-free
	5 kΩ-125 kΩ	Weak fault
	125 kΩ-251 kΩ	⟨0w1/U/-⟩
	251 kΩ-∞	⟨0w1/0/-⟩

that the device behaves as a high ohmic resistor for high defect sizes. Linear resistors are unable to show that the switching process of the device changes from BS to CS. Furthermore, when using the traditional modeling approach, the defects are always present. Therefore, these defect models cannot describe the intermittent behavior of the IUSF.

This can also be shown by performing a fault analysis using the linear resistor defect models. It is clear that the IUSF can only occur when a device is in HRS and a subsequent SET operation is applied, i.e., S = 1w0 is the only applicable sensitizing sequence. We model the parallel and series resistors as $\text{R}_{\text{Br BL-int}}$ and $\text{R}_{\text{Op BL}}$ from Fig. 4.1, respectively. Table 7.3 shows the faults that are sensitized for varying defect strengths. The sHtD FP ⟨0w1/U/-⟩ is sensitized, but it does not contain the temporal component of the fault, i.e., this fault will always be present for the given defect strengths. Because of this, a test for this fault will also miss the temporal behavior of the fault. Therefore, the detection probability of such a test will be low. As the fault is HtD, it cannot be detected using only March algorithms and additional DFT is required. Again, the weak write DFT from [13] can be used to detect the fault using the following algorithm:

$$\text{March-IUSF-LR} = \{\Updownarrow(\text{w0});\Updownarrow(\text{w1},\hat{\text{w}}0,\text{r1})\}. \tag{7.1}$$

Here, the weak ŵ0 operation is used to push the faulty cell from ‘U’ to ‘0’, while keeping the fault-free cells in ‘1’. Hence, the final r1 operation will detect the fault.

Table 7.4: Model parameters for JART VCM v2 [160]

Symbol	Value	Symbol	Value
l_1	8.75 nm	a	0.6 nm
l_{cell}	10 nm	$R_{\text{series},0}$	2 kΩ
l_2	$l_{\text{cell}} - l_1$	N_{max}	$3 \cdot 10^{27}\,\text{m}^{-3}$
N_{init1}	$2 \cdot 10^{27}\,\text{m}^{-3}$	μ_{n0}	$1.8 \cdot 10^{-5}\,\text{m}^2/(\text{Vs})$
N_{init2}	$3.5 \cdot 10^{25}\,\text{m}^{-3}$	ν_0	$4 \cdot 10^{12}$ Hz
$e\phi_{\text{Bn0,1}}$	0.35 eV	ΔW_{A}	0.9 eV
$e\phi_{\text{Bn0,1}}$	0.1 eV	ϵ	$17 \cdot \epsilon_0$
r_{fil}	18 nm	$\epsilon_{\phi_{\text{B}}}$	$5.5 \cdot \epsilon_0$
ΔE_{ac}	0.04 eV	$\alpha_{\text{T,series}}$	$4 \cdot 10^{-3}\,\text{K}^{-1}$
$R_{\text{th,eff}}$	$6 \cdot 10^6\,\text{KW}^{-1}$		

7.3. DEVICE-AWARE DEFECT MODELING

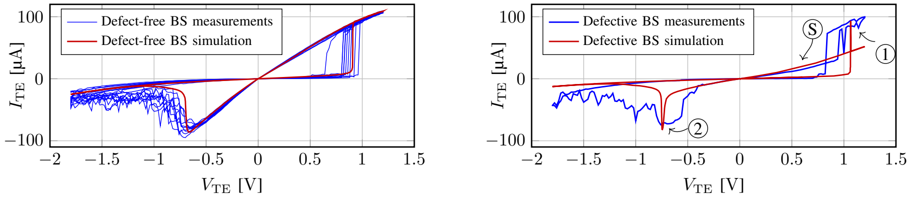
We apply the DAT defect modeling approach from Chapter 5 to model the IUSF.

7.3.1. PHYSICAL DEFECT MODELING

In this step, the effects of the defect on the physical parameters of the RRAM device are analyzed and modeled. Section 7.1 showed that a shortage of ions that can bind free oxygen ions causes CS in BS devices, which leads to the IUSF. To appropriately model this phenomenon, we use the physics-based HfO_x RRAM model for CS, JART VCM v2, from [160]. We first modify the model to describe a defect-free BS RRAM. Thereafter, we change the resulting model so that it sensitizes the IUSF. The model describes CS as the exchange of oxygen vacancies between two regions (region 1 and 2, see Fig. 2.1a) in the oxide; the paper also describes how the model can be adapted to perform BS by introducing asymmetry between these regions via their lengths, barrier height, and electron mobility. Based on this, we adapt the model so that region 1 becomes the BS CF, and that the switching depends solely on this region. Then, we include the unwanted CS by changing two of the model parameters: the initial oxygen vacancy concentration in region 1 (parameter N_{init1}), and the maximal vacancy concentration in the oxide (parameter N_{max}). The former controls the number of oxygen vacancies that form the CF in region 1, while the latter controls the overall number of oxygen vacancies in the oxide. Since the model is based on the exchange of vacancies between region 1 and 2, the ratio $N_{\text{init1}}/N_{\text{max}}$ determines the number of vacancies that can be in region 2. The lower this ratio, the more vacancies can move to region 2 and thus the stronger the CS effect.

7.3.2. ELECTRICAL DEFECT MODELING

The next step is to incorporate the affected physical parameters of the device (e.g., the filament radius) into its electrical parameters (e.g., V_{SET}, R_{RESET}). JART VCM v2 is written in Verilog-A. Hence, it can directly be integrated in a SPICE simulator to derive the electrical behavior of the memory in the presence of the modeled defects.



(a) Simulation vs. measurements for defect-free device.

(b) Simulation vs. measurements for defective device

Figure 7.4: Comparing defect-free and defective device simulation model

7.3.3. FITTING AND MODEL OPTIMIZATION

The model is fitted to match the measurements from Section 7.1. The parameter values used for the JART VCM v2 model are listed in Table 7.4. Fig. 7.4a shows the simulation results for the BS device model as well as the measurements of defect-free devices for different SET-RESET cycles. The model matches the measurements well, except for the RESET current decrease, which is steeper than the measured current decrease. This is because the model does not include the randomness of the filament growth and rupture, and it is based on the CS model assuming that vacancies always shift between the two switching regions; in a real RRAM device, the exchange is between the capping layer and the oxide, leading to a less abrupt switching.

Fig. 7.4b shows the simulation results (of the calibrated model) and the measurement data for a defective BS RRAM device suffering from an IUSF; we use the same parameters as in Table 7.4 but with $N_{\text{init1}} = 1.3 \cdot 10^{27}\,\text{m}^{-3}$ and $N_{\text{max}} = 8.0 \cdot 10^{27}\,\text{m}^{-3}$ so that the ratio $N_{\text{init1}}/N_{\text{max}}$ is rather small while still being physically realistic. It can be seen that the simulation and the measurements match well, and the the simulation can predict the switching moments (①, ②), and the final resistance states (Ⓢ). Note that that the simulation has marginal deviations compared with the data, e.g., some shifts in SET and RESET voltages. This is again caused by the used physical model that favors the abrupt switching over the more gradual switching. Nonetheless, the obtained device model is accurate enough (e.g., accurate in the prediction of the intermediate state Ⓢ) to be used for fault modeling and test development.

The model is also calibrated to describe the behavior of weak devices modeling such as that shown in Fig. 7.1c, e.g., using $N_{\text{init1}} = 1.8 \cdot 10^{27}\,\text{m}^{-3}$ and $N_{\text{max}} = 8.0 \cdot 10^{27}\,\text{m}^{-3}$. The results show a very good matching between measurements and simulations.

7.4. DEVICE-AWARE FAULT ANALYSIS

In order to analyze the IUSF effects on RRAM, a SPICE-based simulation is built; it consists of a 1T1R cell (with same dimensions as those devices used during the characterization) and the required circuitry to drive appropriate voltages in the three control lines WL, BL and SL. The defect-free device is now replaced with the model of the defective RRAM device obtained in the previous section. The defect strength is governed by the ratio $N_{\text{init1}}/N_{\text{max}}$; N_{init1} is varied from $1.2 \cdot 10^{27}\,\text{m}^{-3}$ to $2.0 \cdot 10^{27}\,\text{m}^{-3}$, and N_{max} from $3.0 \cdot 10^{27}\,\text{m}^{-3}$ to $8.0 \cdot 10^{27}\,\text{m}^{-3}$ to fit within realistic physical limits [53] and ensure proper functioning of the model. Similar to the characterization, the voltages are applied via the

Table 7.5: Fault Analysis Results for IUSF.

$N_{\text{init1}}/N_{\text{max}}$ [−]	N_{max} [$10^{27}\,\text{m}^{-3}$]	N_{init1} [$10^{27}\,\text{m}^{-3}$]	Fault
0.15	8.0	1.2	⟨0w1/U/-⟩
0.18	8.0	1.4	⟨0w1/U/-⟩
0.20	8.0	1.6	Weak fault
0.20	60	12	Weak
0.23	8.0	1.8	Weak fault
0.23	60	14	Weak
0.25	8.0	2.0	Weak fault
0.27	6.0	1.6	Fault free
0.30	6.0	1.8	Fault free
0.33	60	20	Fault free
0.40	30	12	Fault free
0.47	30	14	Fault free
0.53	3.0	1.6	Fault free
0.60	30	18	Fault free
0.67	3.0	2.0	Fault free

BL and are swept from 0 V to 1.2 V back to 0 V for SET, as this is the only operation that can sensitize the fault. While performing SPICE simulations, we inspect both the final resulting resistance of the defective device (e.g., R_{SET} after SET operation) and the I-V graph; these are used to derive the behavior of the memory in the presence of the modeled defects. For example, if the final device resistance is outside of the R_{SET} range, then F = U, and if the undesired CS occurs but only disturbs the R_{SET} value without putting it outside the spec, then it is a weak fault.

Table 7.5 shows the obtained results. It can be seen that faults are sensitized when the ratio $N_{\text{init1}}/N_{\text{max}}$ decreases; strong faults IUSFs are sensitized for ratios below 0.2, while weak faults are caused for ratio's between 0.20 and 0.25. A lower ratio gives more room for vacancies to move from region 1 into region 2. This makes it easier for the device to perform the undesired CS.

7.5. DEVICE-AWARE TEST DEVELOPMENT

As shown in Table 7.5, the defect causes strong and weak faults. A straightforward test for the strong fault IUSF would be a *March test*. However, due to the nature of the faults (being intermittent and causing the cell to switch into 'U'), a March test cannot guarantee the detection of such a fault. It will rather *probabilistically* detect the fault; reading this state will sometimes result in '1' and sometimes in '0'. The following march algorithm can be used:

$$\text{March-IUSF} = \left\{ \Updownarrow (\text{w0}, \text{w1}, \text{r1})^k \right\}. \tag{7.2}$$

If we assume that reading a cell in 'U' state results the same probability of getting '1' or '0' (i.e., 50 %), and that the occurrence probability of IUSF is P_{IUSF}, then the detection probability is: P_{d}=$1-(1-(P_{\mathrm{IUSF}}\cdot 50\,\%))^{k}$. Assuming that P_{IUSF}=1.068 % (see Section 7.1) results in k = 560 to realize a fault coverage of 95 %, and in k = 1291 to 99 %. Hence, realizing high fault coverage needs long test time; not to mention the potential impact of repeating memory accesses on the endurance. Note that the detection capabilities might be further improved by adding additional stress factors, e.g., by using back-to-back operations and special data backgrounds.

To reduce the test time while providing high fault coverage, design-for-test (DFT) schemes can be used. For example, the authors in [19] change the references of the SA to perform a binary search to find the resistance of the RRAM device. This idea can be modified to detect the IUSF by setting the reference of the SA directly at the boundary between the '1' and 'U' state, making the read operation deterministic; i.e., if IUSF occurs, then the SA will *always* output '0', and detect it. This results into $P_{\mathrm{d}} = 1-(1-P_{\mathrm{IUSF}})^{k}$. In this case realizing a fault coverage of 95 % requires $k = 279$ and 99 % requires $k = 644$; a reduction in test time of about ≈2X as compared with march test only.

To further decrease the test time while keeping high fault coverage, specialized DFTs are required. Such schemes could aim at increasing P_{IUSF} (and thus decreasing k) by e.g., increasing the current through the device during SET. This will lead to a wider CF which increases P_{IUSF} and thus decreases P_{weak} [39]. This could be done by boosting the WL or BL voltage during SET in test mode. A drawback of this scheme is that it may lead to lower R_{SET}, which will increase power consumption. Alternatively, additional DFT schemes could be introduced that continuously monitor and verify write operations, for example by performing a read operation using the above mentioned idea from [19] after every SET operation. This will increase the write latency and energy consumption, but it will significantly boost the fault coverage.

8

TESTING AND RELIABILITY OF COMPUTATION-IN-MEMORY

Modern computing architectures based on the von Neumann architecture suffer from major memory bottlenecks that the data that can be processed by the processor. To overcome these problems, researchers have proposed to move the computation for memory intensive operations to the memory itself, thus alleviating the bottlenecks. RRAMs are a suitable candidate to implement a Computation-in-Memory (CIM) architecture. However, such a new architecture introduce new faults and failure mechanisms and thus require specialized test solutions. This chapter introduces the Device-Aware Test approach for RRAM-based CIM architectures. We demonstrate that there are unique faults that are not seen in regular memories, and that CIM architectures need to be tested both as a memory and computing device. Furthermore, we investigate the reliability of RRAM-based CIM architectures. We analyze how well it can withstand the effects of process, voltage and temperature variations.

This chapter is partially based on [31]–[33].

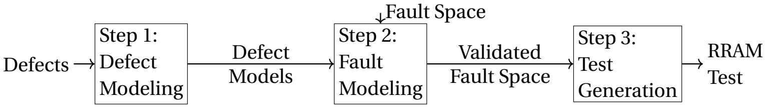


Figure 8.1: Device-Aware Test approach overview

8.1. Device-Aware Test for Computation-in-Memory Overview

Computing in the CIM core takes actually place within the memory. Hence, the CIM core can operate in two different configurations: the *memory* and the *computation* configuration. Fig. 2.15a shows these configurations, as well as the operations that each configuration requires. Since the computation configuration also uses read and write operations, it is a superset of the memory configuration. Note that at least part of the CIM hardware used in the computation configuration is not used in the memory configuration. Hence, CIM cores cannot be tested only as regular memories. CIM cores have to be tested for both configurations. As the computation configuration makes use of the memory, the latter has to be tested first. Testing CIM cores has to be performed as follows:

1. **Memory Configuration Test:** In this case, the memory functionality is tested; i.e., only the hardware that is required to perform memory operations is enabled and tested. Obviously, common memory test solutions applicable to the type of memory can be used, e.g., the test solutions from Chapters 4, 5, 6, 7. Note that testing CIM in this configuration is an independent step, and does not test all hardware involved in the computation configuration.

2. **Computation Configuration Test:** In this case, the hardware responsible for all the computing functionalities is tested. This hardware strongly depends on the CIM architecture and the computing features it enables. For example, testing a CIM die with (analog) vector matrix multiplication features could be different than testing for logic bit-wise operations.

For test development, we follow again the Device-Aware Test approach from Chapter 5, as illustrated again in Fig. 8.1. First, the *defects* must be understood and adequately modeled. The resulting *defect models* are injected into the electrical netlist of the design. Second, this netlist is simulated and the faulty behaviors are observed and compiled into *fault models*. Ideally, before the fault analysis, the complete *fault space* should be defined (when applicable). During the *fault analysis*, the fault space is verified by injecting every defect in the netlist, which results in a set of realistic faults for that specific design or layout. Third, test solutions for the realistic faults are *generated*. Applying the above test development approach to CIM would mean applying it two times; once for each CIM configuration, i.e., the memory and computation configuration.

Test development for CIM as memory: the memory core of CIM can be any kind of memory such as conventional ones (SRAM, DRAM) as well as emerging ones (RRAM, PCM, STT-MRAM). Although testing of SRAM and DRAM is very mature, testing of emerging memories is still under investigation. They may need radically new approaches in defect modeling; a defective non-linear device (e.g., an RRAM device) cannot be accurately modeled with a linear resistor in series or in parallel with a perfect device [28], [161].

Test development for CIM as computing unit: As already mentioned, testing CIM in this configuration is strongly dependent on the design of the architecture. Defining what to test for implies the identification of the modified or new blocks integrated with the memory core to realize the computing functionality. To illustrate this, we will briefly analyze two examples of CIM architectures that are explained in Section 2.4.1: CIM-Ah majority Logic [69] and CIM-Pr Scouting Logic [66].

CIM-Ah Majority Logic: realizing such functionality within e.g. RRAM crossbar will need the modification of the following memory components: a) Memory array, b) BL and SL drivers, and c) Control logic. CIM-A architectures require always a redesign of the memory cells, as the conventional memory cell dimensions and their embedding in the bit and word line structure do not allow them to be used for logic. A conventional memory cell is namely heavily optimized in terms of processing stack and layout. Therefore, any modifications of the array require a new cell design and characterization process for the new control voltages, currents, etc. In addition, modifications in the periphery are needed to support the changes in the cell. In case of CIM-Ah Majority Logic, the write drivers and the control circuitry have to be redesigned to support the required functionality; e.g., the control logic needs to assure that the output of the sense amplifier can be fed back into the array via the drivers for operations on data from multiple cells. Therefore, testing CIM-Ah Majority Logic requires the guarantee of testing the memory array, BL and SL drivers, and the control logic. Note that the memory array is tested both in the memory configuration as well as in computation configuration; an access to the memory during computation could lead to an erroneous bit flip of the cell.

CIM-Pr Scouting Logic: As Fig. 2.17 shows that realizing such functionality, for example within RRAM crossbars, will need the modification of the following memory components: a) Memory array, b) Word line decoders, c) Sense amplifiers, and d) Control logic. Even though the computational results are produced in the peripheral circuits, the memory array for CIM-P is a substantial component in the computation. As the peripheral circuits are modified, the currents and voltages applied to the memory array are typically different than in the conventional memory. Obviously, the majority of the changes take place in the peripheral circuits and minimal to medium changes are required in the memory array. CIM-Pr Scouting Logic activates two or more (but not many) rows of a memory array simultaneously (similar to multi-port memories) during computations. Hence, in addition to a customized sense amplifier to perform the logic operation, this architecture also requires modifications in the address decoder to activate several rows at the same time. Note, however, that modifications in the cell array could be minimal as the total read current is still small. Therefore, testing CIM-Pr Scouting Logic requires to test the memory array, sense amplifiers, the decoders, and the control logic. Note also here that the memory array is tested both in the Memory Configuration as well as in Computation configuration; e.g., simultaneous access of the memory array during computing may lead to a fault in a cell.

8.2. DEVICE-AWARE DEFECT MODELING

8.2.1. MEMORY CONFIGURATION

In a RRAM-based CIM core, defects may occur in transistors, interconnections, or in the RRAM devices, as explained in Chapter 4. Defects in transistors (e.g., lithographic variations, polish variations, material impurity, pinholes, etc. [119]) and interconnects (e.g., line edge roughness, irregular shapes, small particles, etc.) are modeled as linear resistors between related nodes. All possible linear resistor defects are shown in Fig. 4.1. The strength of these resistive defects R_{def} is varied from 1 Ω to 100 MΩ in 11 logarithmically spaced points. Defects in RRAM devices are related to the oxide and electrode structures and have a strong effect on the conductive filament, as described in Chapter 3. Due to the non-linear behavior of the RRAM device, these defects cannot be accurately modeled by linear resistors [27]. Therefore, these *forming defects* are modeled using *device-aware* defect modeling approach that takes into account the actual physical behavior of a defective device [28]. We use the forming model that was developed in Chapter 6. In this work, the forming current value is considered between 1 µA and 35 µA in steps of 1 µA.

8.2.2. COMPUTATION CONFIGURATION

The defects that may occur in the computation configuration devices are equal to the defects that may occur in the memory configuration. Therefore, for the computation configuration test we apply the same defect models as we applied for the memory configuration test development. That is, transistor and interconnection defects are modeled with linear resistors, and RRAM device defects are modeled with the forming model from Chapter 6.

8.3. DEVICE-AWARE FAULT MODELING

8.3.1. MEMORY CONFIGURATION

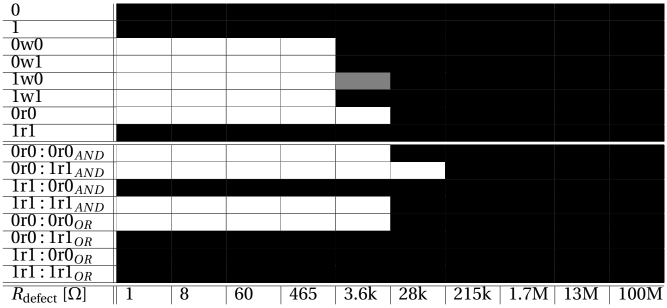
Fault modeling consists of defining the fault space and verifying it via defect model injection and circuit simulation.

FAULT MODELING

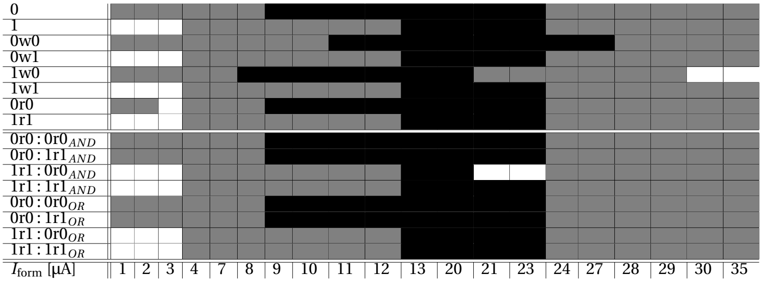
We will define the fault space for the memory array, address decoders, and sense amplifiers.

Array The fault space for the memory array is similar to the one defined in Chapter 5. Again, we make use of the FP notation to describe faults in the memory array.

Address Decoders AD faults in a single decoder are well studied and can be grouped in *static* and *dynamic/delay* faults. Static AD faults (AFs) always lead to errors, and consist of [162]: no access (AFna), multiple cells (AFmc), multiple addresses (AFma), and other cells (AFoc). Delay faults (ADFs), which affect the timing of the AD, consists of two faults [163]: Activation delay fault (ActD), or Deactivation delay fault (DeactD).



(a) Sensitized FPs for defect $R_{\text{Br BL-SL}}$



(b) Sensitized FPs for forming defect

Figure 8.2: Fault maps for defect $R_{\text{Br BL-SL}}$ and forming defect. A white square indicates an ETD fault, a gray one an sHTD fault, and a black one fault-free

Sense Amplifiers SA faults can be grouped in static and dynamic faults. Static faults are SA stuck-at faults (SASF) forcing the SA to produce a fixed output value irrespective of applied memory operation. Dynamic SA faults are slow SA faults (SSAFs) where the SA operation is too slow [164].

8

FAULT ANALYSIS

Array The top part of Fig. 8.2a graphically gives the simulation results for defect $R_{\text{Br BL-SL}}$, while the top part of Fig. 8.2b does the same for the forming defect (the bottom parts shows the results for CC configuration that will be discussed in Section 8.3.2). The colors in these graphs relate to the nature of the validated faults; black indicates that no FP was sensitized (correct operation), white indicates an *easy-to-detect (ETD)* FP, and gray a *hard-to-detect (sHTD)* FP, as was explained in Chapter 5. ETD faults are those whose detection can be always *guaranteed* by applying write and read operations, while sHTD faults are those for which this is not the case, although they may be detected (for example the FP ⟨1r1/U/?⟩ resulting in a random read). The top parts of Fig. 8.2 shows that depending on the strength of the defect (x-axis) and the applied sensitizing operation (y-axis), a fault may or may not be sensitized. In addition, it reveals that a single defect with a *fixed* strength can sensitize multiple faults; e.g., $R_{\text{Br BL-SL}}$ with a strength of 465 Ω sensitizes different FPs for S = 0w0, 0w1, 1w0, 1w1, and 0r0. These multiple faults form together what we call a *fault class (FC)*. For example, the forming defect for $I_{\text{form}} = 1\,\mu\text{A}$ sensitizes both ETD FPs (due to $S \in \{0, 0w1, 1w1, 1r1\}$) and sHTD FPs (due to

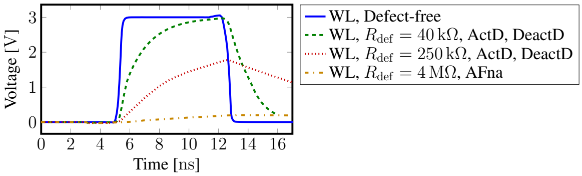


Figure 8.3: WL decoder faults

$S \in \{0, 0w0, 1w0, 0r0\}$) belonging to a single FC.

The results of the fault analysis form the basis for test development. The obtained faults in the previous section are classified as ETD faults (which can easily be detected by e.g. March test [162]) and sHTD faults (which may or may not be detected by March test and guaranteeing their detection requires additional effort such as special DfT circuitry). In addition, for each FC (fault class) of FPs (caused by a single defect with a fixed strength) only a *single* FP needs to be detected. We develop FC tables similar to those in Section 5.4.3 in Table 8.1. Each row gives the minimum number of FPs that have to be considered per defect (i.e, left column of the table) in order to maximize the *defect coverage*. These FPs may belong to the same or different FCs, indicated with the super-scripts. FPs with the same super-script belong to the same FC; e.g., the first row consists of two FCs indicated with 1 and 2, while the last row consists of 5 FCs. Note that for a maximal fault coverage, only one FP per FC should be detected. Including FPs belonging to the same FC in a single row aims at providing freedom to select the right sensitizing operation in order to also detect additional FPs caused by other defects, while optimizing the test time. For example, the detection of FC 1 of $\mathrm{R}_{\text{Br BL-int}}$ can be done by the detection of ⟨0r0/0/1⟩ or ⟨1w0/1/1⟩. However, targeting ⟨0r0/0/1⟩ allows also for the detection of faults caused by $\mathrm{R}_{\text{Br BL-WL}}$, $\mathrm{R}_{\text{Br BL-SL}}$, $\mathrm{R}_{\text{Op WL}}$, $\mathrm{R}_{\text{Sh BL-GND}}$, and $\mathrm{R}_{\text{Sh BL-Vdd}}$; and part of faults due to $\mathrm{R}_{\text{Sh int-Vdd}}$ and $\mathrm{R}_{\text{Sh WL-GND}}$ (see column 0r0 in the table). Note that the faults reported in the *left part* of Table 8.1 consist of ETD faults given in black font and HTD faults given in red one.

Address Decoders Fault analysis for address decoders has been studied also very well by assuming a linear resistor as defect model [163], [164]. For example, Fig. 8.3 illustrates how an open defect in a WL can cause AFna or ADFs, depending on the defect size.

Sense Amplifiers Sense amplifier faults in semiconductor memories have been well studied [9], [164]. They can be divided into static and dynamic faults. Static faults are assumed to be caused by complete opens, low ohmic shorts to V_{DD} or GND, or low ohmic bridges [162]; they consist of the traditional *SA Stuck-at fault* (SASF), an SASF means that the SA always outputs the same value, independent of its inputs. Dynamic faults are caused by partial opens and shorts and consist of two faults: 1) *Unbalanced SA fault* (USAF) [9]: the SA has a continuous tendency to switch to a certain value under equal input conditions, rather than being balanced, and 2) *Slow SA fault* (SSAF) [164]: the SA is too slow to switch, which may result in incorrect read values.

Fault analysis for SAs has been performed by assuming that any defect can be mod-

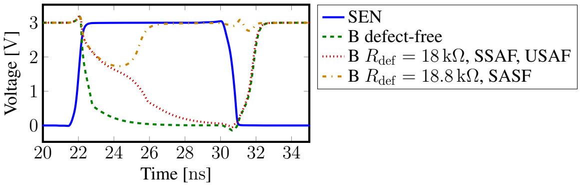


Figure 8.4: SA faults

eled as a linear resistor. Fig. 8.4 illustrates the faults that may occur when performing a r0 operation in an SA in the presence of an open defect (R_{def}) between transistors N2 and N4 of Fig. 2.12. This defect leads to a slower discharge of node B, as the path to GND now has a higher resistance. It can be seen that when $R_{\text{def}} < 18.8\,\text{k}\Omega$, the defect causes an unbalance in the SA and thus is slowing down the sensing operation causing USAFs and SSAFs. When $R_{\text{def}} \geq 18.8\,\text{k}\Omega$, the SA will always switch to the wrong value, thus leading to an SASF.

8.3.2. COMPUTATION CONFIGURATION

In this section, we define and verify the fault space in the CC for the array, ADs, and SAs.

FAULT MODELING

Array In Scouting logic, a compute operation can be seen as a special read operation in which two cells are read *simultaneously* by a single (modified) SA. This dual access may cause new faults in the memory array as it is the case for dual-port memories [165]. To define the fault space for these faults, the $\langle$S/F/R$\rangle$ notation scheme needs to be extended to: $\langle S_1 : S_2 \ / \ F_1 : F_2 \ / \ R\rangle_{op}$. In this notation, S_1 and S_2 describe the two simultaneous sensitizing sequence applied to cell 1 and 2, respectively; ":" indicates that these sensitizing operations are applied in *parallel*. If there is no operation applied to a cell, then this is denoted with n. F_1 and F_2 denote the state of the cells after the sensitizing operation, R indicates the read output, and $_{op}$ indicates the operation being performed (e.g., AND, OR). For example, $\langle 1r1_1 : 0r0_2 \ / \ H_1 : 0_2 \ / \ 1\rangle_{AND}$ denotes an FP where, due to an AND operation performed on two cells containing '1' and '0', the value of cell 1 is flipped from '1' to 'H', while the produced result at the output is a wrong '1'. This extended FP notation can be used to define the fault space for all faults in the memory array in both configurations.

Address Decoders The two row decoders operate simultaneously to select the operands for the compute operation; their behavior is the same as in dual-port memories. It has been shown that defects in address decoders for dual-port memories can lead to *port interference faults* (AFpi) [166]; e.g., one of the row decoder erroneously selects an additional world line when the inputs of both selected decoders have certain defined values. These faults do not occur when the ADs operate sequentially. Note that AD faults in MC can be seen as a subset of AD faults in CC.

Sense Amplifiers The (modified) SAs in the CC suffer from the same faults as in the MC, except that these faults can occur for *every* computing operation (e.g., AND, OR). Thus, the SA fault space in the CC is a superset of that of the MC.

FAULT ANALYSIS

Array The *bottom* part of Fig. 8.2a and 8.2b graphically presents faults that are sensitized in the CC. The simulation is based on a single defect at a time approach; in this case cell 1 is defective when applying logic operations on cell 1 and cell 2. The figures show similar trends. That is, depending on the strength of the defect and the applied logic operations, a fault may or may not be sensitized. In addition, two important observations can be made when comparing the results of the MC and those of the CC (in the same figures):

- A defect with some strength may not sensitize an FP in the MC but it will in CC. For example, in the MC, $R_{\text{Br BL-SL}}$ causes faults *only* for 1 Ω to 3.6 kΩ. In the CC, however, the same defect can cause faults even for higher values such as 28 kΩ. Hence, testing the CIM die in the MC is *not enough* and testing it in CC is a must.
- A defect may cause faults only in the MC, e.g., see defect $R_{\text{Sh WL-Vdd}}$. Hence, testing only the CC will lead to test escapes.
- A defect may cause HTD faults in MC and ETD faults in CC. For example, the forming defect for $21\,\mu\text{A} \leq I_{form} \leq 23\,\mu\text{A}$ causes ETD faults when $S = 1r1_1 : 0r0_{2\,AND}$, while it causes only HTD faults in MC for S = 1w0. Hence, testing in CC will increase the fault coverage at low cost.
- A defect may cause HTD faults in the CC while ETD faults in the MC; see the forming defect for $30\,\mu\text{A} \leq I_{form} \leq 35\,\mu\text{A}$. Hence, both configurations need to be considered for high quality and optimal test solutions.
- Many defects that sensitize faults in the MC also sensitize faults in the CC, e.g., defect $R_{\text{Br BL-SL}}$. This increases the amount of faults in the FC. A test needs only to detect one of these and thus can become more efficient.

Address Decoders The Scouting logic computation configuration requires both address decoders to act simultaneously to select the appropriate word lines. This configuration may give rise to unique address decoder faults, and is quite similar to dual-port memories [166]; also here two addresses should be selected simultaneously. Hence, the same fault space and fault models can apply. Such faults are called *port interference faults* and are due to potential interference/bridges between the two decoders (between wires of the two different decoders). They differ from single AFs in the sense that they only occur when two decoders are accessed simultaneously, and not when operating sequentially. For example, one of the decoders erroneously selects an additional word line when the inputs of both decoders have defined values. Consider Fig. 2.19 and assume the two addresses $A_1A_2A_1A_0$='1111' and $B_3B_2B_1B_0$='1110' are selected in a 4-bit WL decoders; these will drive WL0A and WL1B simultaneously. If now a low ohmic bridge defect exist between the node Y_1 of the decoder circuit driving WLB1 and the node X_2 of

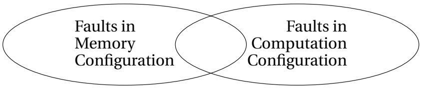


Figure 8.5: Faults in Scouting logic-based CIM die

the decoder circuit driving WLA2, then the simultaneous selection of WL0A and WL1B will result in erroneous selection of WL2A, i.e., WL0, WL1, and WL2 will be activated.

Sense Amplifiers The modified SA in the computation configuration may suffer from similar faults as the SA in the memory configuration. These faults (consisting of SASF, USAF and SSAF) can take place in each of the computing configurations of the SA including OR, AND, and XOR; note that the modified SA uses different reference currents to perform the different logic operations.

From the above we can conclude that faults in Scouting logic-based CIM dies consist of: 1) faults sensitized only in the CC, 2) faults sensitized only in the MC, and 3) faults sensitized in both configurations as illustrated in Fig. 8.5. High-quality and optimal test solutions have to take this into consideration.

8.4. DEVICE-AWARE TEST DEVELOPMENT

8.4.1. MEMORY CONFIGURATION

TEST FOR ETD FAULTS

Analyzing the ETD memory array faults in the *left part* of Table 8.1 reveals that detecting faults sensitized by all defects requires a test which detects the following 6 FPS: ⟨0r0/0/1⟩, ⟨1r1/1/0⟩, ⟨0w1/0/-⟩, ⟨1w0/1/-⟩, ⟨0w1; 0/1/-⟩, and ⟨1w0/H/-⟩. These can be detected by some known March test algorithms such as March C- [162], or by dedicated March tests optimized for such faults. Note that such test will also cover the ETD (i.e., static) SA faults in MC configurations, as these are modelled as stuck-at-faults. For the detection of ETD faults in the ADs, a test should satisfy the following conditions [162]: $\Uparrow(\mathrm{r}x,\ldots,\mathrm{w}\bar{x})$ and $\Downarrow(\mathrm{r}\bar{x},\ldots,\mathrm{w}x)$ for x=0 or x=1 ($\bar{x}$ denotes the negation of x. $\Uparrow$ ($\Downarrow$) denotes an increasing (decreasing) address order. Taking all requirements for ETD faults in MC configuration results in the following March test with a test length of $3T_w + 3T_r$:

$$\text{March-ETD-MC} = \{\Updownarrow(\mathrm{w0});\Uparrow(\mathrm{r0},\mathrm{w1});\ \Downarrow(\mathrm{r1},\mathrm{w0});\Updownarrow(\mathrm{r0})\} \tag{8.1}$$

It can be easily verified that the test guarantees the detection of all considered ETD faults (array, SAs, ADs). In addition, the test *may* detect *some* of HTD faults such as array faults with undefined state, e.g.,⟨0w1/U/-⟩. In this case, a subsequent read operation may return ‘0’ instead of ‘1’.

Note that the CIM die consists of two row decoders. Hence, both should be tested for ETD faults, e.g., by applying the test twice. However, this can be done more efficiently by dividing the columns equally over both row decoders. For example, if the array has C columns, decoder A is used for all columns less than $C/2$ and decoder B for all remaining columns.

Test for HTD Faults

Detecting the HTD faults in the array requires more effort than detecting the ETD ones, e.g., by including DFT schemes. Looking again at the left part of Table 8.1 reveals that detection of all HTD faults in the array requires the detection of 3 FPs: ⟨1w0/U/-⟩, ⟨1w0/L/-⟩, and ⟨0w1/U/-⟩ (or any other FP of FC 2 of the forming defect). We propose to use the weak write DFT scheme from [13] (see Section 4.3.2). The write operation is modified so that it pushes defective cells in a wrong state but defect-free cells remain unaffected. For example, the fault ⟨0w1/U/-⟩ can be detected by first performing a normal w1 operation followed by a $\hat{\mathrm{w}}0$ operation; the latter puts the defective cell into '0'. A defect-free cell does not have enough time to switch back to '0' and remains in '1'. When a r1 is performed thereafter, the defective cell can be detected. Cells in 'L' and 'H' can be detected in a similar way, e.g., the fault ⟨1w1/H/-⟩ can be detected by performing a $\hat{\mathrm{w}}0$ operation following a normal w1 operation so that a defect-free cell switches to '0', while the defective cell remains in 'H' or '1'. Combining the weak write scheme with the required memory operations yields the following March-HTD-MC algorithm with test time of $5T_w + 3T_r$:

$$\begin{aligned}\text{March-HTD-MC} = \{&\Updownarrow(\mathrm{w0})\,;\Downarrow(\mathrm{r0},\mathrm{w1},\hat{\mathrm{w}}0)\,;\\ &\Uparrow(\mathrm{r1},\mathrm{w0})\,;\Updownarrow(\hat{\mathrm{w}}1,\mathrm{r0})\}\,.\end{aligned} \tag{8.2}$$

The HTD faults in the peripherals are delay faults. Traditional march tests may or may not sensitize these faults, and thus the detection probability is low. However, in [163], it has been shown that ActD and DeactD in ADs can be sensitized by applying specific *sensitizing sequences* with specific *addressing transitioning order*, e.g. by applying an $\Updownarrow^{H1}$ (Hamming distance between two addresses of 1) addressing order. Hence, the fault coverage of March-HTD-MC can be improved by including these sensitizations. For high test efficiency, the column addresses should again be split between both row decoders. In the SA, SSAFs can be sensitized applying a back-to-back stressing sequence with fast row access to the SA that forces the output to switch quickly between '1' and '0' [164]. This can, for example, be done by adding the following sequence to March-HTD-MC: $\Updownarrow(\mathrm{r0},\mathrm{w1},\mathrm{r1})$.

8.4.2. Computation Configuration

This section presents a test for the CC ETD and HTD faults.

Test for ETD Faults

A test for the ETD faults in the CC can be developed in the same way as for the MC. However, when developing a test to cover all ETD memory array faults in *both* configurations, two approaches can be used: 1) extend the test for the MC faults to cover additional CC faults; or 2) develop a test (as we did it for MC) but then by considering faults of *both* configurations at the same time. The benefits of the first way are easier test development, and better defect characterization thanks to the clear split between the configurations. The benefit of test co-development for both configurations is that it results in more compact and efficient tests. Next we will use the second approach as it can be done in a systematic way (as done for MC faults), and results in low test complexity algorithms.

The right side of Table 8.1 lists the compute faults that were sensitized per defect. The entries are derived in the same way as was done for the MC. The test for the ETD array faults in both configurations is developed by selecting the minimal amount of Ss, in the MC *and* CC, that sensitize all FCs in Table 8.1. For example, the selection of the four sensitizing sequences $S \in \{1w0, 1r1, 0r0:1r1_{\text{AND}}, 1r1:0r0_{\text{AND}}, 1r1:0r0_{\text{OR}}\}$ results in the following $5T_w + 5T_r$ test detecting all ETD array faults:

$$\begin{aligned}\text{March-ETD} = \Big\{&\Updownarrow(\text{w0});\Uparrow(\text{r0},\text{w1});\Downarrow(\text{r1},\text{w0});\\ &\Updownarrow(\text{w1}_r:\text{n},\text{r1}_r:\text{r0}_{r+1},\text{r0}_{r+1}:\text{r1}_r,\text{r1}_r:\text{r0}_{r+1},\\ &\text{w0}_r:\text{n})_{AND,AND,OR}\Big\}.\end{aligned} \tag{8.3}$$

The test switches between memory (i.e., the first three march elements) and compute (i.e., last march element) configurations; n denotes no operation. Note that for the computing operation access two cells in the same column *simultaneously* (i.e., rows r and $r+1$). In addition to all ETD memory array faults, it can be easy seen that the test also detects SA and AD faults in MC. However, the detection of SA and AD faults in CC require additional attention. To detect the CC SA faults, two elements that contain $(\text{r1}:\text{r1})_{AND}$ and $(\text{r0}:\text{r0})_{OR}$ need to be added. To detect CC AD faults, an additional test such as those used for dual-port memories can be used [166].

The computing features can be further used for test optimization. For example, instead of performing $\Updownarrow(\text{r0}_r:\text{n}_{r+1})_R$ to all memory cells, one can perform the OR operation to each two adjacent rows, i.e., $\Updownarrow(\text{r0}_r:\text{r0}_{r+1})_{OR}$. This will result in 50% less execution time. The same philosophy can be applied to $\Updownarrow(\text{r1}_r:\text{n}_{r+1})_R$ with regards to AND. With this addressing, the test time is sped up 5%.

Test for HTD Faults

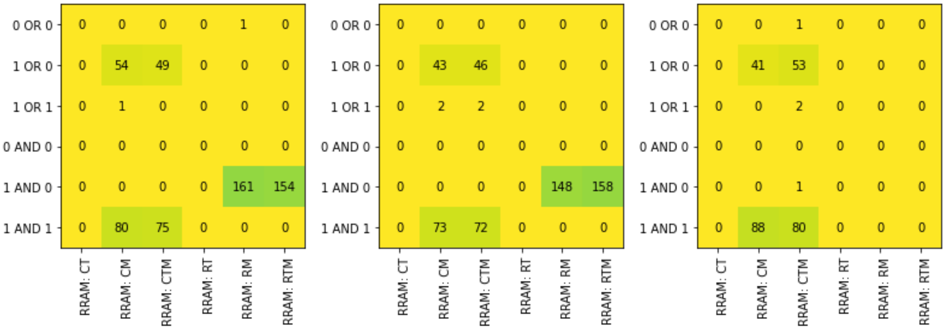
The test for all HTD array faults is developed similarly to the ETD test. From Table 8.1, it follows that all faults in the array can be sensitized by $S \in \{1w0, 1r1, 0r0:1r1_{\text{AND}}, 1r1:0r0_{\text{OR}}\}$. To detect the HTD faults in the array in both configurations, we again apply the weak write operations DFT [13]. This results in the following March-HTD algorithm applied to every column with a test time of $6T_w + 5T_r$:

$$\begin{aligned}\text{March-HTD} = \{&\Uparrow(\text{w1});\Uparrow(\text{r1},\hat{\text{w}}1,\text{r1});\\ &\Downarrow(\text{w0},\hat{\text{w}}1);\Downarrow(\text{r0}_r:\text{n},\text{w1}_r:\text{n},\\ &\text{r1}_r:\text{r0}_{r+1},\text{r0}_{r+1}:\text{r1}_r,\text{w0}_r:\text{n})_{R,OR,AND}\Big\}.\end{aligned} \tag{8.4}$$

To detect all CC SA faults the same additional sequences as for March-ETD need to be included. To further improve the fault coverage for SA faults, an additional back-to-back stressing sequence with fast row addressing can be added to March-HTD as well [164]. Further, all AD faults can be detected by combining *March AF2* from [166] with the specialized sensitizing and addressing sequences from [163].

8.5. Computation-in-Memory Reliability

Next, we analyze how robust the Scouting logic RRAM-based CIM implementation is to process, voltage, and temperature (PVT) variations.



	RRAM: CT	RRAM: CM	RRAM: CTM	RRAM: RT	RRAM: RM	RRAM: RTM
0 OR 0	0	0	0	0	1	0
1 OR 0	0	54	49	0	0	0
1 OR 1	0	1	0	0	0	0
0 AND 0	0	0	0	0	0	0
1 AND 0	0	0	0	0	161	154
1 AND 1	0	80	75	0	0	0

(a) $T = 300\,\mathrm{K}$, $V_{\mathrm{DD}} = 1.5\,\mathrm{V}$, nominal

	RRAM: CT	RRAM: CM	RRAM: CTM	RRAM: RT	RRAM: RM	RRAM: RTM
0 OR 0	0	0	0	0	0	0
1 OR 0	0	43	46	0	0	0
1 OR 1	0	2	2	0	0	0
0 AND 0	0	0	0	0	0	0
1 AND 0	0	0	0	0	148	158
1 AND 1	0	73	72	0	0	0

(b) $T = 273\,\mathrm{K}$, $V_{\mathrm{DD}} = 1.5\,\mathrm{V}$

	RRAM: CT	RRAM: CM	RRAM: CTM	RRAM: RT	RRAM: RM	RRAM: RTM
0 OR 0	0	0	1	0	0	0
1 OR 0	0	41	53	0	0	0
1 OR 1	0	0	2	0	0	0
0 AND 0	0	0	0	0	0	0
1 AND 0	0	0	1	0	0	0
1 AND 1	0	88	80	0	0	0

(c) $T = 358\,\mathrm{K}$, $V_{\mathrm{DD}} = 1.5\,\mathrm{V}$

Figure 8.6: Number of operation errors for varying temperature. Each data point refers to 1000 Monte Carlo simulations points.

To study the impact of resistive variations on the CIM performance, we perform all possible 2-bit AND and OR operations, with varying operating conditions using the Scouting logic architecture from Section 2.4. We vary the *ambient temperature* (T) and the *supply voltage* (V_{DD}). Based on the nominal corner at 300 K and V_{DD} of 1.5 V, we investigate the fixed voltage corners with varying temperature (273 K and 358 K) and the fixed temperature corners with varying V_{DD} (1.4 V and 1.6 V). For every combination, we perform a sensitivity analysis to study the influence of six different component PV combinations in the circuit: cell transistor only (CT), cell memristive device (CM), cell transistor and memristive device (CTM), reference transistor (RT), reference memristive devices (RM), and reference transistor and memristive device (RTM). For every temperature, voltage and component variation combination, we perform 1000 Monte Carlo (MC) simulations. For the transistor variations, we use the variation models included in the TSMC 40nm LP process development kit, while for RRAM device variations, we set up the models according to [53]. We study both D2D and C2C variations. We initialize the cells to the opposite value of the desired value to include the D2D variation. Subsequently, we write the correct value to the cell before we perform the logic operation to include the C2C variation. To illustrate, if the desired operation is 1 AND 1, we initialize the cells to ‘0’, then write a ‘1’ to them and perform the CIM operation.

For every MC iteration, we record three metrics: 1) Number of operation errors (number of operations that fail to produce the correct answer) 2) Energy consumption (energy consumption of the complete circuit during the read operation) and 3) Sensing delay (the delay between the start of the sensing operation and the output crossing $V_{\mathrm{DD}}/2$).

First, we present the results for the nominal case. Second, we present the results for the temperature variations. Third, we present the results for the voltage variations.

8.5.1. NOMINAL CASE: NO VOLTAGE/TEMPERATURE VARIATIONS

NUMBER OF OPERATION ERRORS

Fig. 8.6a lists the number of operation errors for the nominal case, i.e., $T = 300\,\text{K}$ and $V_{\text{DD}} = 1.5\,\text{V}$. The x-axis lists the cell technology with the varied component combination, e.g., *RRAM: CTM* denotes that the components that are varied are the cell transistor and the cell RRAM device. The y-axis lists the performed operations, e.g., 1 OR 0. We observe the following:

- More errors are observed when the equivalent cell resistance is closer to the reference resistance.
- There are no RRAM access transistor faults.

They will be discussed in detail below. We can see that the closer the equivalent resistance is to a reference, more errros will be generated. To illustrate, the equivalent resistance of two cells storing a ‘1’ and ‘0’ is closer to the OR reference resistance than two cells storing ‘1’ and ‘1’. Hence, variations will cause more operation errors in those cases. Finally, the figure shows that there are no RRAM access transistor faults. This is caused by the fact that this transistor is relatively large, and thus PV here has a relatively small effect on the performance [167].

ENERGY CONSUMPTION

Table 8.2 lists the mean energy consumption results in columns A. The table shows that the lower the equivalent and reference resistance is, the more energy is consumed. This can be explained by the fact that lower resistance will discharge the nodes of the SA further before the difference is sensed than higher resistances.

SENSING DELAY

Table 8.3 lists the mean sensing delay results in columns A. We chose to only list the sensing delay per operation, as there was more variation here than per component combination. The table shows that the sensing delay is the shortest when the equivalent cell and reference resistance are farther from each other. This can be explained by the larger voltage difference that will develop during the sensing phase, which will lead to a faster sensing operation.

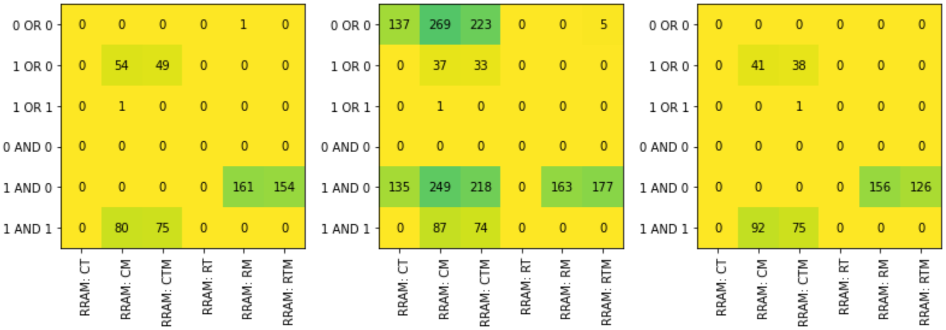
8.5.2. TEMPERATURE DEPENDENCE

NUMBER OF OPERATION ERRORS

The number of operation errors for the three studied temperatures is shown in Fig. 8.6. From the figures the following observation can be made.

- Temperature has a limited effect on RRAM cells.

This will be discussed next. The reduced number of errors in RRAM cells for higher temperatures can be attributed to the fact that with higher temperatures there is less resistance variation for RRAM cells [142]. Hence, the impact of these temperature variations is marginal for RRAM cells.



	RRAM: CT	RRAM: CM	RRAM: CTM	RRAM: RT	RRAM: RM	RRAM: RTM
0 OR 0	0	0	0	0	1	0
1 OR 0	0	54	49	0	0	0
1 OR 1	0	1	0	0	0	0
0 AND 0	0	0	0	0	0	0
1 AND 0	0	0	0	0	161	154
1 AND 1	0	80	75	0	0	0

(a) $T = 300\,\mathrm{K}$, $V_{\mathrm{DD}} = 1.5\,\mathrm{V}$, nominal

	RRAM: CT	RRAM: CM	RRAM: CTM	RRAM: RT	RRAM: RM	RRAM: RTM
0 OR 0	137	269	223	0	0	5
1 OR 0	0	37	33	0	0	0
1 OR 1	0	1	0	0	0	0
0 AND 0	0	0	0	0	0	0
1 AND 0	135	249	218	0	163	177
1 AND 1	0	87	74	0	0	0

(b) $T = 300\,\mathrm{K}$, $V_{\mathrm{DD}} = 1.4\,\mathrm{V}$

	RRAM: CT	RRAM: CM	RRAM: CTM	RRAM: RT	RRAM: RM	RRAM: RTM
0 OR 0	0	0	0	0	0	0
1 OR 0	0	41	38	0	0	0
1 OR 1	0	0	1	0	0	0
0 AND 0	0	0	0	0	0	0
1 AND 0	0	0	0	0	156	126
1 AND 1	0	92	75	0	0	0

(c) $T = 300\,\mathrm{K}$, $V_{\mathrm{DD}} = 1.6\,\mathrm{V}$

Figure 8.7: Number of operation errors for varying V_{DD}. Each data point refers to 1000 Monte Carlo simulations points.

ENERGY CONSUMPTION

The results on the energy consumption are shown in Table 8.2 in columns A and B. The table shows that, due to the change in transistor threshold voltage, less energy is consumed when the temperature decreases, and more energy is consumed when it increases. Furthermore, the same trends from the nominal case are seen at these two temperature corners as well.

SENSING DELAY

The results on the sensing delay are shown in Table 8.3 in columns A and B. Lower temperatures decrease the sensing delay, as the SA is able to operate faster. Further, the operation dependability seen in the nominal case still applies.

8.5.3. VOLTAGE DEPENDENCE

NUMBER OF OPERATION ERRORS

The number of operation errors for varying V_{DD} is shown in Fig. 8.7. The following observation can be made:

- Lower voltages lead to more operation errors.

This will be explained next. Lower voltages introduce more operation errors, as the transistor current is reduced. This increases the equivalent resistance of the cells and thus lowers the difference with the reference resistance, leading to operation errors. For RRAM, lowering V_{DD} also decreases the voltage over the RRAM device, which increases its switching time. This is more severe for the RESET process, i.e., writing ‘1’, and thus more errors will occur when an operation contains a ‘1’. Note that the difference between the OR reference resistance and the equivalent resistance of two cells that store ‘1’ is large enough so that no errors occur for this operation. Higher voltages do not affect the RRAM performance, as there is sufficient voltage to reliably write the cells.

ENERGY CONSUMPTION

The effects of voltage variations on the energy consumption are listed in Table 8.2 in columns A and C. As expected, it decreases when the voltage decreases, and it increases when the voltage increases. Furthermore, the operation dependability is similar to the nominal case.

SENSING DELAY

The effects of voltage variations on the sensing delay are listed in Table 8.3 in columns A and C. Lower voltages will result in a slower SA, and thus increase the sensing delay, while higher voltages will speed up the sensing and thus reduce the sensing delay. Furthermore, the operation dependability is similar to the nominal case.

8.5.4. POTENTIAL IMPROVEMENTS

This section discusses improvements to reduce the impact of PVT variations on the CIM performance.

- Increase the LRS/HRS ratio: Increasing this ratio increases the difference between the two references and the closest equivalent cell resistances. This can be implemented in RRAM by changing the cell drivers that are used for writing.

- Change the operation: The number of operation errors depends strongly on the difference between the reference and the equivalent cell resistance. Hence, many errors can be mitigated if the operation with the largest difference is favored. Here, this means that the AND operations should be preferred for RRAM. Additionally, changing the operand encoding to HRS/LRS=0/1 allows reusing the initial OR operation to reliably implement the AND operation.

- Modify cell structure: In [168], the authors propose to add a second access transistor, word and bit line, to increase the difference between the equivalent cell and reference resistance. This can be effective, but decreases the density of the CIM capable memory, which increases its cost.

Table 8.1: Sensitized FPs that cover maximal amount of defect strengths in an FC. FCs indicated by -x, $x \in \mathbb{N}$, red FPs are HTD.

	Sensitizing Sequence (S) for Memory Configuration								**Sensitizing Sequence (S) for Computation Configuration**							
Defect	**0**	**1**	**0w0**	**0w1**	**1w0**	**1w1**	**0r0**	**1r1**	**0r0:0r0 AND**	**0r0:1r1 AND**	**1r1:0r0 AND**	**1r1:1r1 AND**	**0r0:0r0 OR**	**0r0:1r1 OR**	**1r1:0r0 OR**	**1r1:1r1 OR**
$R_{\text{Br BL-int}}$					$\langle S/1/-\rangle^{1}$ $\langle S/U/-\rangle^{2}$		$\langle S/0/1\rangle^{1}$			$\langle S/0:1/1\rangle^{1}$	$\langle S/1:0/1\rangle^{1}$		$\langle S/0:0/1\rangle^{1}$			
$R_{\text{Br BL-WL}}$							$\langle S/1/0\rangle^{1}$	$\langle S/1/0\rangle^{1}$						$\langle S/1:1/0\rangle^{1}$		$\langle S/1:1/0\rangle^{1}$
$R_{\text{Br int-WL}}$								$\langle S/L/0\rangle^{1}$ $\langle S/1/0\rangle^{1}$				$\langle S/1:1/0\rangle^{1}$			$\langle S/1:0/0\rangle^{1}$	
$R_{\text{Br WL-SL}}$				$\langle S/0/-\rangle^{1}$				$\langle S/1/0\rangle^{2}$				$\langle S/1:1/0\rangle^{1,2}$			$\langle S/1:0/0\rangle^{1,2}$	
$R_{\text{Br SL-int}}$			$\langle S/L/-\rangle^{1}$		$\langle S/L/-\rangle^{1}$											
$R_{\text{Br BL-SL}}$							$\langle S/0/1\rangle^{1}$		$\langle S/0:0/1\rangle^{1}$	$\langle S/0:1/1\rangle^{1,2}$		$\langle S/1:1/0\rangle^{1}$	$\langle S/0:0/1\rangle^{1}$			
$R_{\text{Op BL}}$					$\langle S/1/-\rangle^{1}$										$\langle S/1:0/0\rangle^{2}$	
$R_{\text{Op WL}}$							$\langle S/0/1\rangle^{1}$								$\langle S/1:0/0\rangle^{1}$	
$R_{\text{Op SL}}$								$\langle S/1/0\rangle^{1}$								
$R_{\text{Sh int-GND}}$			$\langle S/1/-\rangle^{1}$ $\langle S/U/-\rangle^{2}$		$\langle S/1/-\rangle^{1}$ $\langle S/U/-\rangle^{2}$											
$R_{\text{Sh int-Vdd}}$							$\langle S/1/1\rangle^{1}$	$\langle S/L/0\rangle^{2}$	$\langle S/1:1/1\rangle^{1,2}$		$\langle S/1:1/1\rangle^{1,2}$		$\langle S/1:1/1\rangle^{1,2}$		$\langle S/1:1/1\rangle^{1,2}$	
$R_{\text{Sh BL-GND}}$							$\langle S/0/1\rangle^{1}$			$\langle S/1:1/1\rangle^{1}$	$\langle S/1:0/1\rangle^{1}$		$\langle S/0:0/1\rangle^{1}$			
$R_{\text{Sh BL-Vdd}}$							$\langle S/1/0\rangle^{1}$	$\langle S/1/0\rangle^{1}$	$\langle S/1:1/1\rangle^{1}$	$\langle S/1:1/1\rangle^{1}$	$\langle S/1:1/1\rangle^{1}$	$\langle S/1:1/0\rangle^{1}$	$\langle S/1:1/1\rangle^{1}$		$\langle S/1:1/1\rangle^{1}$	
$R_{\text{Sh WL-Vdd}}$			$\langle S;0/1/-\rangle^{1}$	$\langle S;0/1/-\rangle^{1}$	$\langle S;0/1/-\rangle^{1}$	$\langle S;0/1/-\rangle^{1}$										
$R_{\text{Sh WL-GND}}$				$\langle S/0/-\rangle^{1}$	$\langle S/1/-\rangle^{1}$ $\langle S/U/-\rangle^{2}$		$\langle S/0/1\rangle^{1}$					$\langle S/1:1/0\rangle^{1}$			$\langle S/1:0/0\rangle^{1}$	
$R_{\text{Sh SL-GND}}$			$\langle S;0/1/-\rangle^{1}$	$\langle S;0/1/-\rangle^{1}$	$\langle S;0/1/-\rangle^{1}$	$\langle S;0/1/-\rangle^{1}$										
$R_{\text{Sh SL-Vdd}}$								$\langle S/1/0\rangle^{1}$				$\langle S/1:1/0\rangle^{1}$			$\langle S/1:0/0\rangle^{1}$	
Forming		$\langle S/L/-\rangle^{1}$ $\langle S/0/-\rangle^{1}$ $\langle S/U/-\rangle^{2}$		$\langle S/L/-\rangle^{1}$ $\langle S/0/-\rangle^{1}$ $\langle S/U/-\rangle^{2}$	$\langle S/U/-\rangle^{3,4}$ $\langle S/H/-\rangle^{5}$	$\langle S/L/-\rangle^{1}$ $\langle S/U/-\rangle^{2}$		$\langle S/L/0\rangle^{1}$ $\langle S/0/0\rangle^{1}$ $\langle S/U/0\rangle^{1}$ $\langle S/U/1\rangle^{2}$	$\langle S/H:0/1\rangle^{4,5}$	$\langle S/H:0/1\rangle^{4,5}$	$\langle S/L:0/0\rangle^{1}$ $\langle S/U:0/0\rangle^{2}$ $\langle S/1:0/1\rangle^{3}$ $\langle S/H:0/1\rangle^{4,5}$	$\langle S/L:1/0\rangle^{1}$ $\langle S/U:1/1\rangle^{2}$ $\langle S/H:0/1\rangle^{4,5}$	$\langle S/H:0/1\rangle^{4,4}$	$\langle S/H:0/1\rangle^{4,5}$	$\langle S/L:0/0\rangle^{1}$ $\langle S/U:0/1\rangle^{2}$ $\langle S/H:0/1\rangle^{4,5}$	$\langle S/L:0/0\rangle^{1}$ $\langle S/U:1/1\rangle^{2}$ $\langle S/H:0/1\rangle^{4,5}$

Table 8.2: Mean CIM Energy Consumption [pJ]

Experiment set	A	B	B	C	C
T [K]	300	273	358	300	300
V_{DD} [V]	1.5	1.5	1.5	1.4	1.6
0 OR 0	7.41	7.29	7.94	6.15	8.82
1 OR 0	7.19	6.95	7.76	5.88	8.56
1 OR 1	6.40	6.16	6.97	5.22	7.72
0 AND 0	6.42	6.28	7.05	5.14	7.73
1 AND 0	6.27	6.03	6.96	5.10	7.58
1 AND 1	6.08	5.81	6.69	4.97	7.39

Table 8.3: Mean CIM Sensing Delay [ps]

Experiment set	A	B	B	C	C
T [K]	300	273	358	300	300
V_{DD} [V]	1.5	1.5	1.5	1.4	1.6
0 OR 0	708	664	746	784	651
1 OR 0	737	709	777	833	672
1 OR 1	606	585	628	681	563
0 AND 0	620	581	645	676	574
1 AND 0	627	607	650	684	580
1 AND 1	638	619	661	700	592

9

CONCLUSION

9.1. SUMMARY

9.1.1. CHAPTER 1: INTRODUCTION

This chapter introduced this thesis. We explained the Von Neumann architecture, the memory hierarchy and the three walls it faces. These walls are: the power wall, the memory walls, and the instruction level parallelism wall. We explained what emerging memories are, how they could replace current mainstream memories, and can be used to mitigate the problems the walls pose. After that, we presented the state of the art in RRAM testing. We concluded that existing RRAM test solutions do not detect actual defects in the RRAM device, but instead defects in the interconnections. After this, we formulated the research topics of this thesis. These are: accurate defect modeling, accurate fault modeling and analysis, and high-quality test development. We listed the contributions of this thesis and detailed how they address the research topics.

9.1.2. CHAPTER 2: BACKGROUND

This chapter introduced the fundamentals of RRAM. We have shown that there exist three switching modes for RRAM devices: bipolar switching, unipolar switching, and complementary switching. We compared several compact RRAM models superficially and studied three models in more depth. These models can be used in the RRAM architecture that was explained next. We explained the structure and workings of all the components. After that, we introduces the CIM concept and showed how the regular RRAM architecture needs to be adapted to perform Scouting logic. Finally, we presented an overview of existing RRAM chips, both prototypes and commercially available ones.

9.1.3. CHAPTER 3: RRAM STRUCTURE, PRODUCTION PROCESS, DEFECTS, AND RELIABILITY

This chapter presented the production process of a RRAM, the defects that may occur, and the related reliability issues. First, we presented the structural details of an OxRAM RRAM device. After that, we introduced the three steps of the manufacturing process: front-end-of-line, back-end-of-line, and the CF forming. We explained the details of these three steps and what defects may occur there. Finally, we presented the major reliability problems for RRAMs and related these to the production process as well.

9.1.4. CHAPTER 4: TRADITIONAL DEFECT MODELING, FAULT MODELING, AND TEST DEVELOPMENT

In this chapter, we presented and demonstrated the traditional test development approach. We applied it to model spot defects in the interconnections of a RRAM. First, we showed which interconnections suffer from which defects. Then, we performed fault modeling and analysis. We defined the complete fault space and investigated the faults that already exist in literature. Further, we validated the fault space for the spot defects. After this, we showed which RRAM tests exist in literature and how they can detect the faults that were sensitized.

9.1.5. CHAPTER 5: DEVICE-AWARE TEST APPROACH

This chapter introduces the DAT approach. We demonstrate that the traditional defect modeling approach that makes use of linear resistors falls short to model RRAM device defects. The DAT approach aims to model the actual physics of a defective device and consists of three steps: 1) device-aware defect modeling, 2) device-aware fault modeling, and 3) device-aware test development. Step 1) is the most important, as this step forms the bridge between actual measured defects and an accurate defect model that can be used for test development. In, step 2) we presented a method to perform fault analysis in such a way that optimal test can be developed. During step 3), we presented a test development algorithm that results in test algorithms of minimal test time and maximal coverage. We demonstrate the complete DAT approach to the interconnection defects.

9.1.6. CHAPTER 6: DEVICE-AWARE TEST FOR FORMING DEFECTS

This chapter applied the DAT approach to forming defects. First, we explained and characterized the failure mechanism. After this, we applied the DAT approach and developed a device-aware defect model. This defect model shows good correspondence to the characterization data and is subsequently used to perform fault modeling and analysis. The results of this steps are used to develop a device-aware test that can detect the defects. Finally, we compare the results with what the traditional approach would have resulted in, and we can conclude that the traditional approach is unable to sensitize the correct faults. This will result in low-quality test and thereby test escapes and yield loss.

9.1.7. CHAPTER 7: DEVICE-AWARE TEST FOR INTERMITTENT UNDEFINED STATE FAULTS

This chapter applied the DAT approach to the intermittent undefined state fault. Again, we first explain and characterize the failure mechanism. Next, we demonstrate that the traditional defect modeling approach falls short to model this defect properly. After this, we developed a device-aware defect model that can sensitize the fault. We perform fault modeling and analysis, and develop test solutions to detect the fault.

9.1.8. CHAPTER 8: TESTING AND RELIABILITY OF COMPUTATION-IN-MEMORY

This chapter applied the DAT approach on RRAM-based Scouting logic CIM and it studied the effects of PVT variations on the reliability of this CIM architecture. We demonstrate that CIM architectures need to be tested both in their memory and computation configuration. We define the fault space for this architecture and validate it using interconnection and forming defect models.From the fault analysis it follows that there are unique computation configuration faults, and thus, that a CIM chip needs to be tested in both configurations. After this, we analyze the PVT effects on the CIM reliability. We show that the most errors are caused by the fact that the resistive difference between the AND reference and the equivalent cell resistance is smaller than the difference with the OR reference. As such, the AND operation is more prone to reliability issues.

9.2. FUTURE WORK

In this thesis, we demonstrated the usefulness of the DAT approach for developing high-quality tests for RRAMs. Future work should focus on the following points:

Modeling More RRAM Defects From Chapter 3, it follows that there are many RRAM unique defects that are not modeled yet, even though they will have a major impact on the RRAM performance. Some examples are: electrode roughness and material redeposition. Both have a strong impact on the forming process, and thus on the performance of the fabricated device.

Improving RRAM Test Solutions We have shown that many hard-to-detect faults for RRAM are caused by the device being in either ‘L’, ‘H’, or ‘U’ state. Furthermore, it has become clear that the currently existing test solutions are not optimized to detect these faults efficiently. This is especially true for the intermittent undefined state fault, as its occurrence is not certain during test time. To address this, smarter DFTs needs to be developed that allow for in-field detection of these kinds of faults. For example, by modifying the sense amplifier.

Applying DAT to Other Technologies It has been shown that STT-MRAM also benefits from the DAT approach [155], [169]. Hence, it makes sense to believe that it can also be applied to emerging memory technologies, such as PCRAM, but also to established technologies, such as DRAM, Flash, and FinFET transistors. Furthermore, we demonstrated in this work that the approach is also applicable to different architectures than just a regular RRAM. As such, in the future, the DAT approach should also be applied to other circuits that make use of RRAM devices.

BIBLIOGRAPHY

[1] F. Balestra, M. Graef, Y. Hayashi, H. Ishiuchi, T. Conte, and P. Gargini, “International Roadmap for Devices and Systems (IRDS™) 2021 Edition”, IRDS/IEEE, Tech. Rep., 2021. [Online]. Available: https://irds.ieee.org/editions/2021.

[2] G. E. Moore, “Cramming more components onto integrated circuits, Reprinted from Electronics, volume 38, number 8, April 19, 1965, pp.114 ff.”, *IEEE Solid-State Circuits Soc. Newsl.*, vol. 11, no. 3, pp. 33–35, Feb. 2009, ISSN: 1098-4232. DOI: 10.1109/N-SSC.2006.4785860.

[3] R. H. Dennard, F. H. Gaensslen, H. N. Yu, V. L. Rideout, E. Bassous, and A. R. Leblanc, “Design of Ion-Implanted MOSFET’s With Very Small Physical Dimensions”, *IEEE J. Solid-State Circuits*, vol. 9, no. 5, pp. 256–268, 1974, ISSN: 1558173X. DOI: 10.1109/JSSC.1974.1050511.

[4] D. A. Patterson, “Future of Computer Architecture”, in *Berkeley EECS Annu. Res. Symp.*, 2006.

[5] S. Yu and P.-Y. Chen, “Emerging Memory Technologies: Recent Trends and Prospects”, *IEEE Solid-State Circuits Mag.*, vol. 8, no. 2, pp. 43–56, 2016. DOI: 10.1109/MSSC.2016.2546199. [Online]. Available: http://ieeexplore.ieee.org/document/7495087/.

[6] L. Wu, “Testing STT-MRAM: Manufacturing Defects, Fault Models, and Test Solutions”, 2021, ISBN: 978-94-6384-199-3. DOI: 10.4233/uuid:088a3991-4ea9-48a0-9b92-cc763748868c.

[7] M. L. Bushnell and V. D. Agrawal, *Essentials of Electronic Testing for Digital Memory & Mixed-Signal VLSI Circuits*. New York, New York, USA: Springer Science+Business Media, 2000, ISBN: 9781475781427.

[8] S. Hamdioui and A. Van De Goor, “An experimental analysis of spot defects in SRAMs: realistic fault models and tests”, in *Proc. Ninth Asian Test Symp.*, IEEE Comput. Soc, 2000, pp. 131–138, ISBN: 0-7695-0887-1. DOI: 10.1109/ATS.2000.893615. [Online]. Available: http://ieeexplore.ieee.org/document/893615/.

[9] K. Zarrineh, A. Deo, and R. Adams, “Defect analysis and realistic fault model extensions for static random access memories”, in *Rec. IEEE Int. Work. Mem. Technol. Des. Test.*, IEEE Comput. Soc, 2000, pp. 119–124, ISBN: 0-7695-0689-5. DOI: 10.1109/MTDT.2000.868625. [Online]. Available: http://ieeexplore.ieee.org/document/868625/.

[10] O. Ginez, J.-M. Portal, and C. Muller, "Design and Test Challenges in Resistive Switching RAM (ReRAM): An Electrical Model for Defect Injections", in *ETS 2009*, IEEE, May 2009, pp. 61–66, ISBN: 978-0-7695-3703-0. DOI: 10.1109/ETS.2009.23. [Online]. Available: http://ieeexplore.ieee.org/document/5170460/.

[11] N. Z. Haron and S. Hamdioui, "On Defect Oriented Testing for Hybrid CMOS/Memristor Memory", in *2011 Asian Test Symp.*, IEEE, Nov. 2011, pp. 353–358, ISBN: 978-1-4577-1984-4. DOI: 10.1109/ATS.2011.66. [Online]. Available: http://ieeexplore.ieee.org/document/6114755/.

[12] N. Z. Haron and S. Hamdioui, "DfT schemes for resistive open defects in RRAMs", in *DATE 2012*, IEEE, Mar. 2012, pp. 799–804, ISBN: 978-1-4577-2145-8. DOI: 10.1109/DATE.2012.6176603. [Online]. Available: http://ieeexplore.ieee.org/document/6176603/.

[13] S. Hamdioui, M. Taouil, and N. Z. Haron, "Testing Open Defects in Memristor-Based Memories", *IEEE Trans. Comput.*, vol. 64, no. 1, pp. 247–259, Jan. 2015. DOI: 10.1109/TC.2013.206. [Online]. Available: http://ieeexplore.ieee.org/document/6636885/.

[14] S. Kannan, J. Rajendran, R. Karri, and O. Sinanoglu, "Sneak-Path Testing of Crossbar-Based Nonvolatile Random Access Memories", *IEEE Trans. Nanotechnol.*, vol. 12, no. 3, pp. 413–426, May 2013, ISSN: 1536-125X. DOI: 10.1109/TNANO.2013.2253329. [Online]. Available: http://ieeexplore.ieee.org/document/6482250/.

[15] ——, "Sneak-path Testing of Memristor-based Memories", in *2013 26th Int. Conf. VLSI Des. 2013 12th Int. Conf. Embed. Syst.*, IEEE, Jan. 2013, pp. 386–391, ISBN: 978-1-4673-4639-9. DOI: 10.1109/VLSID.2013.219. [Online]. Available: http://ieeexplore.ieee.org/document/6472671/.

[16] S. Kannan, N. Karimi, R. Karri, and O. Sinanoglu, "Detection, diagnosis, and repair of faults in memristor-based memories", in *VTS 2014*, IEEE, Apr. 2014, pp. 1–6, ISBN: 978-1-4799-2611-4. DOI: 10.1109/VTS.2014.6818762. [Online]. Available: http://ieeexplore.ieee.org/document/6818762/.

[17] S. N. Mozaffari, S. Tragoudas, and T. Haniotakis, "Fast march tests for defects in resistive memory", in *Proc. 2015 IEEE/ACM Int. Symp. Nanoscale Archit.*, IEEE, Jul. 2015, pp. 88–93, ISBN: 978-1-4673-7849-9. DOI: 10.1109/NANOARCH.2015.7180592. [Online]. Available: http://ieeexplore.ieee.org/document/7180592/.

[18] ——, "More Efficient Testing of Metal-Oxide Memristor-Based Memory", *IEEE Trans. Comput. Des. Integr. Circuits Syst.*, vol. 36, no. 6, pp. 1018–1029, Jun. 2017, ISSN: 02780070. DOI: 10.1109/TCAD.2016.2608863. [Online]. Available: http://ieeexplore.ieee.org/document/7565639/.

[19] 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", *IEEE Trans. Comput.*, vol. 64, no. 1, pp. 180–190, Jan. 2015, ISSN: 00189340. DOI: 10.1109/TC.2014.12. [Online]. Available: http://ieeexplore.ieee.org/document/6725492/.

[20] Y.-X. Chen and J.-F. Li, "Fault modeling and testing of 1T1R memristor memories", in *VTS 2015*, IEEE, Apr. 2015, pp. 1–6, ISBN: 978-1-4799-7597-6. DOI: 10.1109/VTS.2015.7116247. [Online]. Available: http://ieeexplore.ieee.org/document/7116247/.

[21] P. Liu, Z. You, J. Wu, B. Liu, Y. Han, and K. Chakrabarty, "Fault Modeling and Efficient Testing of Memristor-Based Memory", *IEEE Trans. Circuits Syst. I Regul. Pap.*, vol. 68, no. 11, pp. 4444–4455, Nov. 2021, ISSN: 15580806. DOI: 10.1109/TCSI.2021.3098639.

[22] T. S. Copetti, T. Gemmeke, and L. M. Bolzani Poehls, "Validating a DFT Strategy's Detection Capability regarding Emerging Faults in RRAMs", *2021 IFIP/IEEE 29th Int. Conf. Very Large Scale Integr.*, pp. 1–6, Oct. 2021. DOI: 10.1109/VLSI-SOC53125.2021.9606993. [Online]. Available: https://ieeexplore.ieee.org/document/9606993/.

[23] 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", *Electron. Lett.*, vol. 52, no. 18, pp. 1520–1522, Sep. 2016, ISSN: 0013-5194. DOI: 10.1049/el.2016.1693. [Online]. Available: http://digital-library.theiet.org/content/journals/10.1049/el.2016.1693.

[24] Y. Luo, X. Cui, M. Luo, and Q. Lin, "A high fault coverage march test for 1T1R memristor array", in *EDSSC 2017*, IEEE, Oct. 2017, pp. 1–2, ISBN: 978-1-5386-2907-9. DOI: 10.1109/EDSSC.2017.8126415. [Online]. Available: http://ieeexplore.ieee.org/document/8126415/.

[25] 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", *Electron. Lett.*, vol. 53, no. 25, pp. 1631–1632, Dec. 2017. DOI: 10.1049/el.2017.2424. [Online]. Available: http://digital-library.theiet.org/content/journals/10.1049/el.2017.2424.

[26] P. Liu, Z. You, J. Wu, M. Elimu, W. Wang, S. Cai, and Y. Han, "Defect analysis and parallel testing for 3D hybrid CMOS-memristor memory", *IEEE Trans. Emerg. Top. Comput.*, vol. 9, no. 2, pp. 745–758, Apr. 2021. DOI: 10.1109/TETC.2020.2982830.

[27] M. Fieback, M. Taouil, and S. Hamdioui, "Testing Resistive Memories: Where are We and What is Missing?", in *2018 IEEE Int. Test Conf.*, IEEE, Oct. 2018, pp. 1–9, ISBN: 978-1-5386-8382-8. DOI: 10.1109/TEST.2018.8624895. [Online]. Available: https://ieeexplore.ieee.org/document/8624895/.

[28] M. Fieback, L. Wu, G. Cardoso Medeiros, H. Aziza, S. Rao, E. J. Marinissen, M. Taouil, and S. Hamdioui, "Device-Aware Test: A New test Approach Towards DPPB Level", in *Proc. 50th Int. Test Conf. 2019*, 2019.

[29] M. Fieback, G. C. Medeiros, L. Wu, H. Aziza, R. Bishnoi, M. Taouil, and S. Hamdioui, "Defect and Fault Modeling , and Test Development Framework for RRAM", *ACM J. Emerg. Technol. Comput. Syst.*, vol. 00, no. 0, pp. 1–26, 2022.

[30] M. Fieback, G. C. Medeiros, A. Gebregiorgis, H. Aziza, M. Taouil, and S. Hamdioui, “Intermittent Undefined State Fault in RRAMs”, in *Proc. Eur. Test Work.*, vol. 2021-May, Institute of Electrical and Electronics Engineers Inc., May 2021, ISBN: 9781665418492. DOI: 10.1109/ETS50041.2021.9465401.

[31] M. Fieback, S. Nagarajan, R. Bishnoi, M. Tahoori, M. Taouil, and S. Hamdioui, “Testing Scouting Logic-Based Computation-in-Memory Architectures”, in *Proc. Eur. Test Work.*, vol. 2020-May, Institute of Electrical and Electronics Engineers Inc., May 2020, ISBN: 9781728143125. DOI: 10.1109/ETS48528.2020.9131604.

[32] M. Fieback, M. Christopher, and A. Grebegiorgis, “PVT Analysis for RRAM and STT-MRAM-based Logic Computation-in-Memory”, in *Eur. Test Symp.*, 2022.

[33] S. Hamdioui, M. Fieback, S. Nagarajan, and M. Taouil, “Testing Computation-in-Memory Architectures Based on Emerging Memories”, in *ITC 2019*, 2019, pp. 1–10, ISBN: 9781728148236.

[34] 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”, *Proc. IEEE*, vol. 100, no. 6, pp. 1951–1970, Jun. 2012, ISSN: 0018-9219. DOI: 10.1109/JPROC.2012.2190369. [Online]. Available: http://ieeexplore.ieee.org/document/6193402/.

[35] R. Degraeve, A. Fantini, N. Raghavan, L. Goux, S. Clima, B. Govoreanu, A. Belmonte, D. Linten, and M. Jurczak, “Causes and consequences of the stochastic aspect of filamentary RRAM”, *Microelectron. Eng.*, vol. 147, pp. 171–175, Nov. 2015, ISSN: 0167-9317. DOI: 10.1016/J.MEE.2015.04.025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167931715002348%7B%5C#%7Dbi0005.

[36] U. Celano, L. Goux, R. Degraeve, A. Fantini, O. Richard, H. Bender, M. Jurczak, and W. Vandervorst, “Imaging the three-dimensional conductive channel in filamentary-based oxide resistive switching memory”, *Nano Lett.*, vol. 15, no. 12, pp. 7970–7975, Dec. 2015, ISSN: 15306992. DOI: 10.1021/acs.nanolett.5b03078. [Online]. Available: https://pubs.acs.org/doi/10.1021/acs.nanolett.5b03078.

[37] X. Cao, X. Li, W. Yu, Y. Zhang, R. Yang, X. Liu, J. Kong, and W. Shen, “Structural characteristics and resistive switching properties of thermally prepared TiO2 thin films”, *J. Alloys Compd.*, vol. 486, no. 1-2, pp. 458–461, Nov. 2009, ISSN: 0925-8388. DOI: 10.1016/J.JALLCOM.2009.06.175.

[38] Y. Yang, X. Zhang, L. Qin, Q. Zeng, X. Qiu, and R. Huang, “Probing nanoscale oxygen ion motion in memristive systems”, *Nat. Commun.*, vol. 8, no. 1, p. 15 173, Aug. 2017, ISSN: 2041-1723. DOI: 10.1038/ncomms15173. [Online]. Available: http://www.nature.com/articles/ncomms15173.

[39] F. Nardi, S. Balatti, S. Larentis, D. C. Gilmer, and D. Ielmini, “Complementary switching in oxide-based bipolar resistive-switching random memory”, *IEEE Trans. Electron Devices*, vol. 60, no. 1, pp. 70–77, 2013, ISSN: 00189383. DOI: 10.1109/TED.2012.2226728.

[40] X. Chen, W. Hu, Y. Li, S. Wu, and D. Bao, "Complementary resistive switching behaviors evolved from bipolar TiN/HfO2/Pt device", *Appl. Phys. Lett.*, vol. 108, no. 5, p. 053 504, Feb. 2016, ISSN: 00036951. DOI: 10.1063/1.4941287. [Online]. Available: http://aip.scitation.org/doi/10.1063/1.4941287.

[41] F. Nardi, S. Balatti, S. Larentis, and D. Ielmini, "Complementary switching in metal oxides: Toward diode-less crossbar RRAMs", in *Tech. Dig. - Int. Electron Devices Meet. IEDM*, 2011, ISBN: 9781457705052. DOI: 10.1109/IEDM.2011.6131647.

[42] A. Fantini, L. Goux, R. Degraeve, D. Wouters, N. Raghavan, G. Kar, A. Belmonte, Y.-Y. Chen, B. Govoreanu, and M. Jurczak, "Intrinsic switching variability in HfO2 RRAM", in *IMW 2013*, IEEE, May 2013, pp. 30–33, ISBN: 978-1-4673-6169-9. DOI: 10.1109/IMW.2013.6582090. [Online]. Available: http://ieeexplore.ieee.org/document/6582090/.

[43] M. Lanza, H.-S. P. Wong, E. Pop, D. Ielmini, D. Strukov, B. C. Regan, L. Larcher, M. A. Villena, J. J. Yang, L. Goux, A. Belmonte, Y. Yang, F. M. Puglisi, J. Kang, B. Magyari-Köpe, E. Yalon, A. Kenyon, M. Buckwell, A. Mehonic, A. Shluger, H. Li, T.-H. Hou, B. Hudec, D. Akinwande, R. Ge, S. Ambrogio, J. B. Roldan, E. Miranda, J. Suñe, K. L. Pey, X. Wu, N. Raghavan, E. Wu, W. D. Lu, G. Navarro, W. Zhang, H. Wu, R. Li, A. Holleitner, U. Wurstbauer, M. C. Lemme, M. Liu, S. Long, Q. Liu, H. Lv, A. Padovani, P. Pavan, I. Valov, X. Jing, T. Han, K. Zhu, S. Chen, F. Hui, and Y. Shi, "Recommended Methods to Study Resistive Switching Devices", *Adv. Electron. Mater.*, p. 1 800 143, Sep. 2018, ISSN: 2199160X. DOI: 10.1002/aelm.201800143. [Online]. Available: http://doi.wiley.com/10.1002/aelm.201800143.

[44] B. Hajri, M. M. Mansour, A. Chehab, and H. Aziza, "Oxide-based RRAM models for circuit designers: A comparative analysis", in *DTIS 2017*, IEEE, Apr. 2017, pp. 1–6, ISBN: 978-1-5090-6377-2. DOI: 10.1109/DTIS.2017.7930176. [Online]. Available: http://ieeexplore.ieee.org/document/7930176/.

[45] D. B. Strukov, G. S. Snider, D. R. Stewart, and R. S. Williams, "The missing memristor found", *Nature*, vol. 453, no. 7191, pp. 80–83, May 2008. DOI: 10.1038/nature06932. [Online]. Available: http://www.nature.com/articles/nature06932.

[46] D. B. Strukov, ·. R. S. Williams, D. B. Strukov, and R. S. Williams, "Exponential ionic drift: fast switching and low volatility of thin-film memristors", *Appl. Phys. A 2008 943*, vol. 94, no. 3, pp. 515–519, Nov. 2008, ISSN: 1432-0630. DOI: 10.1007/S00339-008-4975-3. [Online]. Available: https://link-springer-com.tudelft.idm.oclc.org/article/10.1007/s00339-008-4975-3.

[47] M. D. Pickett, D. B. Strukov, J. L. Borghetti, J. J. Yang, G. S. Snider, D. R. Stewart, and R. S. Williams, "Switching dynamics in titanium dioxide memristive devices", *J. Appl. Phys.*, vol. 106, no. 7, 2009. DOI: 10.1063/1.3236506. [Online]. Available: http://jap.aip.org/jap/copyright.jsp.

[48] S. Kvatinsky, E. G. Friedman, A. Kolodny, and U. C. Weiser, “TEAM: ThrEshold Adaptive Memristor Model”, *IEEE Trans. Circuits Syst. I Regul. Pap.*, vol. 60, no. 1, pp. 211–221, Jan. 2013. DOI: `10.1109/TCSI.2012.2215714`. [Online]. Available: `http://ieeexplore.ieee.org/document/6353604/`.

[49] S. Kvatinsky, M. Ramadan, E. G. Friedman, and A. Kolodny, “VTEAM: A General Model for Voltage-Controlled Memristors”, *IEEE Trans. Circuits Syst. II Express Briefs*, vol. 62, no. 8, pp. 786–790, Aug. 2015, ISSN: 1549-7747. DOI: `10.1109/TCSII.2015.2433536`. [Online]. Available: `http://ieeexplore.ieee.org/document/7110565/`.

[50] Z. Jiang, Y. Wu, S. Yu, L. Yang, K. Song, Z. Karim, and H.-S. P. Wong, “A Compact Model for Metal-Oxide Resistive Random Access Memory With Experiment Verification”, *IEEE Trans. Electron Devices*, vol. 63, no. 5, pp. 1884–1892, May 2016. DOI: `10.1109/TED.2016.2545412`. [Online]. Available: `http://ieeexplore.ieee.org/document/7448912/`.

[51] Y. V. Pershin and M. Di Ventra, “SPICE model of memristive devices with threshold”, Apr. 2012, [Online]. Available: `https://arxiv.org/abs/1204.2600v5`.

[52] M. Bocquet, H. Aziza, W. Zhao, Y. Zhang, S. Onkaraiah, C. Muller, M. Reyboz, D. Deleruyelle, F. Clermidy, and J. M. Portal, “Compact Modeling Solutions for Oxide-Based Resistive Switching Memories (OxRAM)”, *J. Low Power Electron. Appl.*, vol. 4, no. 1, pp. 1–14, Jan. 2014, ISSN: 2079-9268. DOI: `10.3390/JLPEA4010001`. [Online]. Available: `https://www.mdpi.com/2079-9268/4/1/1/htm%20https://www.mdpi.com/2079-9268/4/1/1`.

[53] C. Bengel, A. Siemon, F. Cüppers, S. Hoffmann-Eifert, A. Hardtdegen, M. Von Witzleben, L. Hellmich, R. Waser, and S. Menzel, “Variability-Aware Modeling of Filamentary VCM based Bipolar Resistive Switching Cells Using SPICE Level Compact Models”, *IEEE Trans. CIRCUITS Syst.*, pp. 1–13, 2020. DOI: `10.1109/TCSI.2020.3018502`.

[54] B. Hajri, H. Aziza, M. M. Mansour, and A. Chehab, “RRAM Device Models: A Comparative Analysis with Experimental Validation”, *IEEE Access*, vol. 7, pp. 168 963–168 980, 2019, ISSN: 21693536. DOI: `10.1109/ACCESS.2019.2954753`.

[55] S. Kannan, R. Karri, and O. Sinanoglu, “Sneak path testing and fault modeling for multilevel memristor-based memories”, in *ICCD 2013*, IEEE, Oct. 2013, pp. 215–220, ISBN: 978-1-4799-2987-0. DOI: `10.1109/ICCD.2013.6657045`. [Online]. Available: `http://ieeexplore.ieee.org/document/6657045/`.

[56] S. Hamdioui, H. A. Du Nguyen, M. Taouil, A. Sebastian, M. L. Gallo, S. Pande, S. Schaafsma, F. Catthoor, S. Das, F. G. Redondo, G. Karunaratne, A. Rahimi, and L. Benini, “Applications of Computation-In-Memory Architectures based on Memristive Devices”, in *2019 Des. Autom. Test Eur. Conf. Exhib.*, IEEE, Mar. 2019, pp. 486–491, ISBN: 978-3-9819263-2-3. DOI: `10.23919/DATE.2019.8715020`. [Online]. Available: `https://ieeexplore.ieee.org/document/8715020/`.

[57] A. Sebastian, T. Tuma, N. Papandreou, M. Le Gallo, L. Kull, T. Parnell, and E. Eleftheriou, “Temporal correlation detection using computational phase-change memory”, *Nat. Commun.*, vol. 8, no. 1, p. 1115, Dec. 2017, ISSN: 2041-1723. DOI: `10.1038/s41467-017-01481-9`. [Online]. Available: `http://www.nature.com/articles/s41467-017-01481-9`.

[58] B. Chen, F. Cai, J. Zhou, W. Ma, P. Sheridan, and W. D. Lu, “Efficient in-memory computing architecture based on crossbar arrays”, in *2015 IEEE Int. Electron Devices Meet.*, IEEE, Dec. 2015, pp. 17.5.1–17.5.4, ISBN: 978-1-4673-9894-7. DOI: `10.1109/IEDM.2015.7409720`. [Online]. Available: `http://ieeexplore.ieee.org/document/7409720/`.

[59] M. A. Lebdeh, U. Reinsalud, H. A. D. Nguyen, S. Wong, and S. Hamdioui, “Memristive Device Based Circuits for Computation-in-Memory Architectures”, in *2019 IEEE Int. Symp. Circuits Syst.*, IEEE, May 2019, pp. 1–5, ISBN: 978-1-7281-0397-6. DOI: `10.1109/ISCAS.2019.8702542`. [Online]. Available: `https://ieeexplore.ieee.org/document/8702542/`.

[60] P.-E. Gaillardon, L. Amaru, A. Siemon, E. Linn, R. Waser, A. Chattopadhyay, and G. De Micheli, “The Programmable Logic-in-Memory (PLiM) computer”, in *2016 Des. Autom. Test Eur. Conf. Exhib.*, 2016, pp. 427–432.

[61] D. Bhattacharjee, R. Devadoss, and A. Chattopadhyay, “ReVAMP: ReRAM based VLIW architecture for in-memory computing”, in *Proc. 2017 Des. Autom. Test Eur. DATE 2017*, Institute of Electrical and Electronics Engineers Inc., May 2017, pp. 782–787, ISBN: 9783981537093. DOI: `10.23919/DATE.2017.7927095`.

[62] S. Hamdioui, K. L. M. Bertels, and M. Taouil, “Computing device for big data applications using memristors”, in *US Patent 9,824,753*, Nov. 2017.

[63] 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 Comp. Arch. News*, vol. 44, 2016, pp. 27–39.

[64] 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 *DAC ’16*, New York, New York, USA: ACM Press, 2016, pp. 1–6, ISBN: 9781450342360. DOI: `10.1145/2897937.2898064`. [Online]. Available: `http://dl.acm.org/citation.cfm?doid=2897937.2898064`.

[65] H. D. Nguyen, J. Yu, L. Xie, M. Taouil, S. Hamdioui, and D. Fey, “Memristive devices for computing: Beyond CMOS and beyond von Neumann”, in *2017 IFIP/IEEE Int. Conf. Very Large Scale Integr.*, IEEE, Oct. 2017, pp. 1–10, ISBN: 978-1-5386-2880-5. DOI: `10.1109/VLSI-SoC.2017.8203479`. [Online]. Available: `http://ieeexplore.ieee.org/document/8203479/`.

[66] L. Xie, H. D. Nguyen, J. Yu, A. Kaichouhi, M. Taouil, M. AlFailakawi, and S. Hamdioui, “Scouting Logic: A Novel Memristor-Based Logic Design for Resistive Computing”, in *ISVLSI 2017*, IEEE, Jul. 2017, pp. 176–181, ISBN: 978-1-5090-6762-6. DOI: `10.1109/ISVLSI.2017.39`. [Online]. Available: `http://ieeexplore.ieee.org/document/7987515/`.

[67] 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”, *Nat. Elec.*, vol. 1, no. 4, p. 246, 2018.

[68] D. Fujiki, S. Mahlke, R. Das, D. Fujiki, S. Mahlke, and R. Das, “In-Memory Data Parallel Processor”, in *Proc. Twenty-Third Int. Conf. Archit. Support Program. Lang. Oper. Syst. - ASPLOS '18*, vol. 53, New York, New York, USA: ACM Press, 2018, pp. 1–14, ISBN: 9781450349116. DOI: `10.1145/3173162.3173171`. [Online]. Available: `http://dl.acm.org/citation.cfm?doid=3173162.3173171`.

[69] E. Linn, R. Rosezin, S. Tappertzhofen, U. Böttger, and R. Waser, “Beyond von Neumann-logic operations in passive crossbar arrays alongside memory operations”, *Nanotechnology*, vol. 23, 2012. DOI: `10.1088/0957-4484/23/30/305205`. [Online]. Available: `http://iopscience.iop.org/article/10.1088/0957-4484/23/30/305205/pdf`.

[70] S. Dietrich, M. Angerbauer, M. Ivanov, D. Gogl, H. Hoenigschmid, M. Kund, C. Liaw, M. Markert, R. Symanczyk, L. Altimime, S. Bournat, and G. Mueller, “A nonvolatile 2-mbit cbram memory core featuring advanced read and program control”, *IEEE Journal of Solid-State Circuits*, vol. 42, no. 4, pp. 839–845, 2007. DOI: `10.1109/JSSC.2007.892207`.

[71] Y. H. Tseng, C.-E. Huang, C. .-.-H. Kuo, Y. .-.-D. Chih, and C. J. Lin, “High density and ultra small cell size of contact reram (cr-ram) in 90nm cmos logic technology and circuits”, in *2009 IEEE International Electron Devices Meeting (IEDM)*, 2009, pp. 1–4. DOI: `10.1109/IEDM.2009.5424408`.

[72] S.-S. Sheu, P.-C. Chiang, W.-P. Lin, H.-Y. Lee, P.-S. Chen, Y.-S. Chen, T.-Y. Wu, F. T. Chen, K.-L. Su, M.-J. Kao, K.-H. Cheng, and M.-J. Tsai, “A 5ns fast write multi-level non-volatile 1 k bits rram memory with advance write scheme”, in *2009 Symposium on VLSI Circuits*, 2009, pp. 82–83.

[73] C. J. Chevallier, C. H. Siau, S. F. Lim, S. R. Namala, M. Matsuoka, B. L. Bateman, and D. Rinerson, “A 0.13um 64mb multi-layered conductive metal-oxide memory”, in *2010 IEEE International Solid-State Circuits Conference - (ISSCC)*, 2010, pp. 260–261. DOI: `10.1109/ISSCC.2010.5433945`.

[74] S.-S. Sheu, M.-F. Chang, K.-F. Lin, C.-W. Wu, Y.-S. Chen, P.-F. Chiu, C.-C. Kuo, Y.-S. Yang, P.-C. Chiang, W.-P. Lin, C.-H. Lin, H.-Y. Lee, P.-Y. Gu, S.-M. Wang, F. T. Chen, K.-L. Su, C.-H. Lien, K.-H. Cheng, H.-T. Wu, T.-K. Ku, M.-J. Kao, and M.-J. Tsai, “A 4mb embedded slc resistive-ram macro with 7.2ns read-write random-access time and 160ns mlc-access capability”, in *2011 IEEE International Solid-State Circuits Conference*, 2011, pp. 200–202. DOI: `10.1109/ISSCC.2011.5746281`.

[75] W. Otsuka, K. Miyata, M. Kitagawa, K. Tsutsui, T. Tsushima, H. Yoshihara, T. Namise, Y. Terao, and K. Ogata, “A 4mb conductive-bridge resistive memory with 2.3gb/s read-throughput and 216mb/s program-throughput”, in *2011 IEEE International Solid-State Circuits Conference*, 2011, pp. 210–211. DOI: `10.1109/ISSCC.2011.5746286`.

[76] J. Yi, H. Choi, S. Lee, J. Lee, D. Son, S. Lee, S. Hwang, S. Song, J. Park, S. Kim, W. Kim, J.-Y. Kim, S. Lee, J. Moon, J. You, M. Joo, J. Roh, S. Park, S.-W. Chung, J. Lee, and S.-J. Hong, "Highly reliable and fast nonvolatile hybrid switching reram memory using thin al2o3 demonstrated at 54nm memory array", in *2011 Symposium on VLSI Technology - Digest of Technical Papers*, 2011, pp. 48–49.

[77] A. Kawahara, R. Azuma, Y. Ikeda, K. Kawai, Y. Katoh, Y. Hayakawa, K. Tsuji, S. Yoneda, A. Himeno, K. Shimakawa, T. Takagi, T. Mikawa, and K. Aono, "An 8 mb multi-layered cross-point reram macro with 443 mb/s write throughput", *IEEE Journal of Solid-State Circuits*, vol. 48, no. 1, pp. 178–185, 2013. DOI: 10.1109/JSSC.2012.2215121.

[78] H. D. Lee, S. G. Kim, K. Cho, H. Hwang, H. Choi, J. Lee, S. H. Lee, H. J. Lee, J. Suh, S.-O. Chung, Y. S. Kim, K. S. Kim, W. S. Nam, J. T. Cheong, J. T. Kim, S. Chae, E.-R. Hwang, S. N. Park, Y. S. Sohn, C. G. Lee, H. S. Shin, K. J. Lee, K. Hong, H. G. Jeong, K. M. Rho, Y. K. Kim, S. Chung, J. Nickel, J. J. Yang, H. S. Cho, F. Perner, R. S. Williams, J. H. Lee, S. K. Park, and S.-J. Hong, "Integration of 4f2 selectorless crossbar array 2mb reram based on transition metal oxides for high density memory applications", in *2012 Symposium on VLSI Technology (VLSIT)*, 2012, pp. 151–152. DOI: 10.1109/VLSIT.2012.6242506.

[79] W. C. Shen, C. Y. Mei, Y.-D. Chih, S.-S. Sheu, M.-J. Tsai, Y.-C. King, and C. J. Lin, "High-k metal gate contact rram (crram) in pure 28nm cmos logic process", in *2012 International Electron Devices Meeting*, 2012, pp. 31.6.1–31.6.4. DOI: 10.1109/IEDM.2012.6479146.

[80] M.-F. Chang, C.-W. Wu, C.-C. Kuo, S.-J. Shen, K.-F. Lin, S.-M. Yang, Y.-C. King, C.-J. Lin, and Y.-D. Chih, "A 0.5v 4mb logic-process compatible embedded resistive ram (reram) in 65nm cmos using low-voltage current-mode sensing scheme with 45ns random read time", in *2012 IEEE International Solid-State Circuits Conference*, 2012, pp. 434–436. DOI: 10.1109/ISSCC.2012.6177079.

[81] A. Kawahara, K. Kawai, Y. Ikeda, Y. Katoh, R. Azuma, Y. Yoshimoto, K. Tanabe, Z. Wei, T. Ninomiya, K. Katayama, R. Yasuhara, S. Muraoka, A. Himeno, N. Yoshikawa, H. Murase, K. Shimakawa, T. Takagi, T. Mikawa, and K. Aono, "Filament scaling forming technique and level-verify-write scheme with endurance over 107 cycles in reram", in *2013 IEEE International Solid-State Circuits Conference Digest of Technical Papers*, 2013, pp. 220–221. DOI: 10.1109/ISSCC.2013.6487708.

[82] J. R. Jameson, P. Blanchard, C. Cheng, J. Dinh, A. Gallo, V. Gopalakrishnan, C. Gopalan, B. Guichet, S. Hsu, D. Kamalanathan, D. Kim, F. Koushan, M. Kwan, K. Law, D. Lewis, Y. Ma, V. McCaffrey, S. Park, S. Puthenthermadam, E. Runnion, J. Sanchez, J. Shields, K. Tsai, A. Tysdal, D. Wang, R. Williams, M. N. Kozicki, J. Wang, V. Gopinath, S. Hollmer, and M. Van Buskirk, "Conductive-bridge memory (cbram) with excellent high-temperature retention", in *2013 IEEE International Electron Devices Meeting*, 2013, pp. 30.1.1–30.1.4. DOI: 10.1109/IEDM.2013.6724721.

[83] T.-y. Liu, T. H. Yan, R. Scheuerlein, Y. Chen, J. K. Lee, G. Balakrishnan, G. Yee, H. Zhang, A. Yap, J. Ouyang, T. Sasaki, A. Al-Shamma, C. Chen, M. Gupta, G. Hilton, A. Kathuria, V. Lai, M. Matsumoto, A. Nigam, A. Pai, J. Pakhale, C. H. Siau, X. Wu, Y. Yin, N. Nagel, Y. Tanaka, M. Higashitani, T. Minvielle, C. Gorla, T. Tsukamoto, T. Yamaguchi, M. Okajima, T. Okamura, S. Takase, H. Inoue, and L. Fasoli, "A 130.7-mm2 2-layer 32-gb reram memory device in 24-nm technology", *IEEE Journal of Solid-State Circuits*, vol. 49, no. 1, pp. 140–153, 2014. DOI: 10.1109/JSSC.2013.2280296.

[84] S. Sills, S. Yasuda, J. Strand, A. Calderoni, K. Aratani, A. Johnson, and N. Ramaswamy, "A copper reram cell for storage class memory applications", in *2014 Symposium on VLSI Technology (VLSI-Technology): Digest of Technical Papers*, 2014, pp. 1–2. DOI: 10.1109/VLSIT.2014.6894368.

[85] M.-F. Chang, J.-J. Wu, T.-F. Chien, Y.-C. Liu, T.-C. Yang, W.-C. Shen, Y.-C. King, C.-J. Lin, K.-F. Lin, Y.-D. Chih, S. Natarajan, and J. Chang, "19.4 embedded 1mb reram in 28nm cmos with 0.27-to-1v read using swing-sample-and-couple sense amplifier and self-boost-write-termination scheme", in *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, 2014, pp. 332–333. DOI: 10.1109/ISSCC.2014.6757457.

[86] H. W. Pan, K. P. Huang, S. Y. Chen, P. C. Peng, Z. S. Yang, C.-H. Kuo, Y.-D. Chih, Y.-C. King, and C. J. Lin, "1kbit finfet dielectric (find) rram in pure 16nm finfet cmos logic process", in *2015 IEEE International Electron Devices Meeting (IEDM)*, 2015, pp. 10.5.1–10.5.4. DOI: 10.1109/IEDM.2015.7409670.

[87] S. H. Jo, T. Kumar, C. Zitlaw, and H. Nazarian, "Self-limited rram with on/off resistance ratio amplification", in *2015 Symposium on VLSI Technology (VLSI Technology)*, 2015, T128–T129. DOI: 10.1109/VLSIT.2015.7223715.

[88] M. Ueki, K. Takeuchi, T. Yamamoto, A. Tanabe, N. Ikarashi, M. Saitoh, T. Nagumo, H. Sunamura, M. Narihiro, K. Uejima, K. Masuzaki, N. Furutake, S. Saito, Y. Yabe, A. Mitsuiki, K. Takeda, T. Hase, and Y. Hayashi, "Low-power embedded reram technology for iot applications", in *2015 Symposium on VLSI Technology (VLSI Technology)*, 2015, T108–T109. DOI: 10.1109/VLSIT.2015.7223640.

[89] Q. Luo, X. Xu, T. Gong, H. Lv, D. Dong, H. Ma, P. Yuan, J. Gao, J. Liu, Z. Yu, J. Li, S. Long, Q. Liu, and M. Liu, "8-layers 3d vertical rram with excellent scalability towards storage class memory applications", in *2017 IEEE International Electron Devices Meeting (IEDM)*, 2017, pp. 2.7.1–2.7.4. DOI: 10.1109/IEDM.2017.8268315.

[90] Z. Chen, H. Wu, B. Gao, D. Wu, N. Deng, H. Qian, Z. Lu, B. Haukness, M. Kellam, and G. Bronner, "Performance improvements by sl-current limiter and novel programming methods on 16mb rram chip", in *2017 IEEE International Memory Workshop (IMW)*, 2017, pp. 1–4. DOI: 10.1109/IMW.2017.7939097.

[91] S. Yasuda, K. Ohba, T. Mizuguchi, H. Sei, M. Shimuta, K. Aratani, T. Shiimoto, T. Yamamoto, T. Sone, S. Nonoguchi, J. Okuno, A. Kouchiyama, W. Otsuka, and K. Tsutsui, "A cross point cu-reram with a novel ots selector for storage class mem-

ory applications”, in *2017 Symposium on VLSI Technology*, 2017, T30–T31. DOI: 10.23919/VLSIT.2017.7998189.

[92] C.-C. Chou, Z.-J. Lin, P.-L. Tseng, C.-F. Li, C.-Y. Chang, W.-C. Chen, Y.-D. Chih, and T.-Y. J. Chang, “An n40 256k×44 embedded rram macro with sl-precharge sa and low-voltage current limiter to improve read and write performance”, in *2018 IEEE International Solid - State Circuits Conference - (ISSCC)*, 2018, pp. 478–480. DOI: 10.1109/ISSCC.2018.8310392.

[93] P. Jain, U. Arslan, M. Sekhar, B. C. Lin, L. Wei, T. Sahu, J. Alzate-vinasco, A. Vangapaty, M. Meterelliyoz, N. Strutt, A. B. Chen, P. Hentges, P. A. Quintero, C. Connor, O. Golonzka, K. Fischer, and F. Hamzaoglu, “13.2 a 3.6mb 10.1mb/mm2 embedded non-volatile reram macro in 22nm finfet technology with adaptive forming/set/reset schemes yielding down to 0.5v with sensing time of 5ns at 0.7v”, in *2019 IEEE International Solid- State Circuits Conference - (ISSCC)*, 2019, pp. 212–214. DOI: 10.1109/ISSCC.2019.8662393.

[94] C.-C. Chou, Z.-J. Lin, C.-A. Lai, C.-I. Su, P.-L. Tseng, W.-C. Chen, W.-C. Tsai, W.-T. Chu, T.-C. Ong, H. Chuang, Y.-D. Chih, and T.-Y. J. Chang, “A 22nm 96kx144 rram macro with a self-tracking reference and a low ripple charge pump to achieve a configurable read window and a wide operating voltage range”, in *2020 IEEE Symposium on VLSI Circuits*, 2020, pp. 1–2. DOI: 10.1109/VLSICircuits18222.2020.9163014.

[95] J. Yang, X. Xue, X. Xu, H. Lv, F. Zhang, X. Zeng, M.-F. Chang, and M. Liu, “A 28nm 1.5mb embedded 1t2r rram with 14.8 mb/mm2 using sneaking current suppression and compensation techniques”, in *2020 IEEE Symposium on VLSI Circuits*, 2020, pp. 1–2. DOI: 10.1109/VLSICircuits18222.2020.9163035.

[96] 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 0.022um2 cell size using self-adaptive delayed termination and multi-cell reference”, in *2021 IEEE International Solid- State Circuits Conference (ISSCC)*, vol. 64, 2021, pp. 336–338. DOI: 10.1109/ISSCC42613.2021.9365945.

[97] L. Grenouillet, N. Castellani, A. Persico, V. Meli, S. Martin, O. Billoint, R. Segaud, S. Bernasconi, C. Pellissier, C. Jahan, C. Charpin-Nicolle, P. Dezest, C. Carabasse, P. Besombes, S. Ricavy, N.-P. Tran, A. Magalhaes-Lucas, A. Roman, C. Boixaderas, T. Magis, M. Bedjaoui, M. Tessaire, A. Seignard, F. Mazen, S. Landis, E. Vianello, G. Molas, F. Gaillard, J. Arcamone, and E. Nowak, “16kbit 1t1r oxram arrays embedded in 28nm fdsoi technology demonstrating low ber, high endurance, and compatibility with core logic transistors”, in *2021 IEEE International Memory Workshop (IMW)*, 2021, pp. 1–4. DOI: 10.1109/IMW51353.2021.9439607.

[98] Panasonic, *MN101LR05D : 8bit Ultra Low Power MN101L*, Accessed on 23-02-2022, 2014. [Online]. Available: https://industrial.panasonic.com/ww/products/pt/mn101l/models/MN101LR05D.

[99] Fujitsu, *Fujitsu Launches 12Mbit ReRAM – Largest Memory Density in ReRAM Family*, Accessed on 23-03-2022, 2021. [Online]. Available: `https://www.fujitsu.com/jp/group/fsm/en/products/reram/spi-12m-mb85as12mt.html`.

[100] D. Semiconductor, *CBRAM Technology - Breakthrough Resistive RAM (ReRAM) Technology*, Accessed on 23-03-2022, 2022. [Online]. Available: `https://www.dialog-semiconductor.com/products/memory/cbram-technology`.

[101] TSMC, *eFlash*, Accessed on 23-03-2022, 2022. [Online]. Available: `https://www.tsmc.com/english/dedicatedFoundry/technology/specialty/eflash`.

[102] W. Kim, S. Menzel, D. J. Wouters, R. Waser, and V. Rana, "3-Bit Multilevel Switching by Deep Reset Phenomenon in Pt/W/TaOX/Pt-ReRAM Devices", *IEEE Electron Device Lett.*, vol. 37, no. 5, pp. 564–567, May 2016, ISSN: 07413106. DOI: `10.1109/LED.2016.2542879`.

[103] L. Grenouillet, N. Castellani, A. Persico, V. Meli, S. Martin, O. Billoint, R. Segaud, S. Bernasconi, C. Pellissier, C. Jahan, C. Charpin-Nicolle, P. Dezest, C. Carabasse, P. Besombes, S. Ricavy, N. P. Tran, A. Magalhaes-Lucas, A. Roman, C. Boixaderas, T. Magis, M. Bedjaoui, M. Tessaire, A. Seignard, F. Mazen, S. Landis, E. Vianello, G. Molas, F. Gaillard, J. Arcamone, and E. Nowak, "16kbit 1T1R OxRAM arrays embedded in 28nm FDSOI technology demonstrating low BER, high endurance, and compatibility with core logic transistors", in *2021 IEEE Int. Mem. Work. IMW 2021 - Proc.*, Institute of Electrical and Electronics Engineers Inc., May 2021, ISBN: 9781728185170. DOI: `10.1109/IMW51353.2021.9439607`.

[104] Kai-Shin Li, C. Ho, Ming-Taou Lee, Min-Cheng Chen, Cho-Lun Hsu, J. M. Lu, C. H. Lin, C. C. Chen, B. W. Wu, Y. F. Hou, C. Y. Lin, Y. J. Chen, T. Y. Lai, M. Y. Li, I. Yang, C. S. Wu, and Fu-Liang Yang, "Utilizing Sub-5 nm sidewall electrode technology for atomic-scale resistive memory fabrication", in *2014 Symp. VLSI Technol. Dig. Tech. Pap.*, IEEE, Jun. 2014, pp. 1–2, ISBN: 978-1-4799-3332-7. DOI: `10.1109/VLSIT.2014.6894402`. [Online]. Available: `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6894402`.

[105] S. Mei, M. Bosman, K. Shubhakar, N. Raghavan, L. Ming, and K. L. Pey, "3D characterization of hard breakdown in RRAM device", *Microelectron. Eng.*, vol. 216, Aug. 2019, ISSN: 01679317. DOI: `10.1016/j.mee.2019.111042`. [Online]. Available: `www.elsevier.com/locate/mee`.

[106] H. Zhang, S. Yoo, S. Menzel, C. Funck, F. Cüppers, D. J. Wouters, C. S. Hwang, R. Waser, and S. Hoffmann-Eifert, "Understanding the Coexistence of Two Bipolar Resistive Switching Modes with Opposite Polarity in Pt/TiO2/Ti/Pt Nanosized ReRAM Devices", *ACS Appl. Mater. Interfaces*, vol. 10, no. 35, pp. 29 766–29 778, Sep. 2018, ISSN: 19448252. DOI: `10.1021/acsami.8b09068`. [Online]. Available: `www.acsami.org`.

[107] S. Yoneda, S. Ito, Y. Hayakawa, Z. Wei, S. Muraoka, R. Yasuhara, K. Kawashima, A. Himeno, and T. Mikawa, "Newly developed process integration technologies for highly reliable 40 nm ReRAM", *Jpn. J. Appl. Phys.*, vol. 58, no. SB, SBBB06, Feb.

2019, ISSN: 13474065. DOI: 10.7567/1347-4065/aafd8d. [Online]. Available: https://doi.org/10.7567/1347-4065/aafd8d.

[108] C.-w. Hsu, I.-t. Wang, C.-l. Lo, M.-c. Chiang, and W.-y. Jang, "Self-Rectifying Bipolar TaOx/TiO2 RRAM with Superior Endurance over 10ˆ12 Cycles for 3D High-Density Storage-Class Memory", *2013 Symp. VLSI Technol. Dig. Tech. Pap.*, T166–T167, 2013, ISSN: 0743-1562.

[109] Y. Chen, "ReRAM: History, Status, and Future", *IEEE Trans. Electron Devices*, vol. 67, no. 4, pp. 1420–1433, Apr. 2020, ISSN: 15579646. DOI: 10.1109/TED.2019.2961505.

[110] G. Bersuker, D. C. Gilmer, D. Veksler, J. Yum, H. Park, S. Lian, L. Vandelli, A. Padovani, L. Larcher, K. McKenna, A. Shluger, V. Iglesias, M. Porti, M. Nafria, W. Taylor, P. D. Kirsch, and R. Jammy, "Metal oxide RRAM switching mechanism based on conductive filament microscopic properties", in *2010 Int. Electron Devices Meet.*, IEEE, Dec. 2010, pp. 19.6.1–19.6.4, ISBN: 978-1-4424-7418-5. DOI: 10.1109/IEDM.2010.5703394. [Online]. Available: http://ieeexplore.ieee.org/document/5703394/.

[111] Y. H. Do, J. S. Kwak, Y. C. Bae, J. H. Lee, Y. Kim, H. Im, and J. P. Hong, "TiN electrode-induced bipolar resistive switching of TiO2 thin films", *Curr. Appl. Phys.*, vol. 10, no. 1, e71–e74, Jan. 2010, ISSN: 1567-1739. DOI: 10.1016/J.CAP.2009.12.017.

[112] A. Hardtdegen, C. La Torre, F. Cuppers, S. Menzel, R. Waser, and S. Hoffmann-Eifert, "Improved switching stability and the effect of an internal series resistor in HfO2/TiOx Bilayer ReRAM Cells", *IEEE Trans. Electron Devices*, vol. 65, no. 8, pp. 3229–3236, Aug. 2018, ISSN: 00189383. DOI: 10.1109/TED.2018.2849872.

[113] A. Grossi, E. Nowak, C. Zambelli, C. Pellissier, S. Bernasconi, G. Cibrario, K. E. Hajjam, R. Crochemore, J. Nodin, P. Olivo, and L. Perniola, "Fundamental variability limits of filament-based RRAM", in *IEDM 2016*, IEEE, Dec. 2016, pp. 4.7.1–4.7.4, ISBN: 978-1-5090-3902-9. DOI: 10.1109/IEDM.2016.7838348. [Online]. Available: http://ieeexplore.ieee.org/document/7838348/.

[114] B. Govoreanu, G. Kar, Y.-Y. Chen, V. Paraschiv, S. Kubicek, A. Fantini, I. Radu, L. Goux, S. Clima, R. Degraeve, N. Jossart, O. Richard, T. Vandeweyer, K. Seo, P. Hendrickx, G. Pourtois, H. Bender, L. Altimime, D. Wouters, J. Kittl, and M. Jurczak, "10x10nm2 Hf/HfOx crossbar resistive RAM with excellent performance, reliability and low-energy operation", in *2011 Int. Electron Devices Meet.*, IEEE, Dec. 2011, pp. 31.6.1–31.6.4, ISBN: 978-1-4577-0505-2. DOI: 10.1109/IEDM.2011.6131652. [Online]. Available: http://ieeexplore.ieee.org/document/6131652/.

[115] S. Balatti, S. Ambrogio, D. C. Gilmer, and D. Ielmini, "Set Variability and Failure Induced by Complementary Switching in Bipolar RRAM", *IEEE Electron Device Lett.*, vol. 34, no. 7, pp. 861–863, Jul. 2013, ISSN: 0741-3106. DOI: 10.1109/LED.2013.2261451. [Online]. Available: http://ieeexplore.ieee.org/document/6530615/.

[116] Y. Hu, D. Perello, M. Yun, D. H. Kwon, and M. Kim, “Variation of switching mechanism in TiO2 thin film resistive random access memory with Ag and graphene electrodes”, *Microelectron. Eng.*, vol. 104, pp. 42–47, Apr. 2013, ISSN: 0167-9317. DOI: 10.1016/J.MEE.2012.11.009.

[117] G. Kar, A. Fantini, Y.-Y. Chen, V. Paraschiv, B. Govoreanu, H. Hody, N. Jossart, H. Tielens, S. Brus, O. Richard, T. Vandeweyer, D. Wouters, L. Altimime, and M. Jurczak, “Process-improved RRAM cell performance and reliability and paving the way for manufacturability and scalability for high density memory application”, in *VLSIT 2012*, IEEE, Jun. 2012, pp. 157–158, ISBN: 978-1-4673-0847-2. DOI: 10.1109/VLSIT.2012.6242509. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6242509.

[118] K. Kuhn, C. Kenyon, A. Kornfeld, M. Liu, A. Maheshwari, W.-k. Shih, S. Sivakumar, G. Taylor, P. VanDerVoorn, and K. Zawadzki, “Managing Process Variation in Intel’s 45nm CMOS Technology”, *Intel J. Technol.*, vol. 12, no. 45, pp. 77–85, 2008. DOI: 10.1535/itj.1201.

[119] K. J. Kuhn, M. D. Giles, D. Becher, P. Kolar, A. Kornfeld, R. Kotlyar, S. T. Ma, A. Maheshwari, and S. Mudanai, “Process Technology Variation”, *IEEE Trans. Electron Devices*, vol. 58, no. 8, pp. 2197–2208, Aug. 2011, ISSN: 0018-9383. DOI: 10.1109/TED.2011.2121913. [Online]. Available: http://ieeexplore.ieee.org/document/5755188/.

[120] M. Stucchi, M. Bamal, and K. Maex, “Impact of line-edge roughness on resistance and capacitance of scaled interconnects”, *Microelectron. Eng.*, vol. 84, no. 11, pp. 2733–2737, Nov. 2007, ISSN: 0167-9317. DOI: 10.1016/J.MEE.2007.05.038. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167931707005631.

[121] H. Aziza, P. Canet, J. Postel-Pellerin, M. Moreau, J.-M. Portal, and M. Bocquet, “ReRAM ON/OFF resistance ratio degradation due to line resistance combined with device variability in 28nm FDSOI technology”, in *EUROSOI 2017*, IEEE, Apr. 2017, pp. 35–38, ISBN: 978-1-5090-5313-1. DOI: 10.1109/ULIS.2017.7962594. [Online]. Available: http://ieeexplore.ieee.org/document/7962594/.

[122] J. Liang, S. Yeh, S. S. Wong, and H.-S. P. Wong, “Effect of Wordline/Bitline Scaling on the Performance, Energy Consumption, and Reliability of Cross-Point Memory Array”, *ACM J. Emerg. Technol. Comput. Syst.*, vol. 9, no. 1, pp. 1–14, Feb. 2013, ISSN: 15504832. DOI: 10.1145/2422094.2422103. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2422094.2422103.

[123] P.-S. Chen, Y.-S. Chen, H.-Y. Lee, T.-Y. Wu, P.-Y. Gu, F. Chen, and M.-J. Tsai, “Impact of Flattened TiN Electrode on the Memory Performance of HfO2 Based Resistive Memory”, *Electrochem. Solid-State Lett.*, vol. 15, no. 4, H136, Jan. 2012, ISSN: 10990062. DOI: 10.1149/2.001205esl. [Online]. Available: http://esl.ecsdl.org/cgi/doi/10.1149/2.001205esl.

[124] S. K. Nandi, X. Liu, D. K. Venkatachalam, and R. G. Elliman, “Effect of Electrode Roughness on Electroforming in HfO2 and Defect-Induced Moderation of Electric-Field Enhancement”, *Phys. Rev. Appl.*, vol. 4, no. 6, p. 064 010, Dec. 2015. DOI: `10.1103/PhysRevApplied.4.064010`. [Online]. Available: `https://link.aps.org/doi/10.1103/PhysRevApplied.4.064010`.

[125] H. Y. Lee, Y. S. Chen, P. S. Chen, P. Y. Gu, Y. Y. Hsu, S. M. Wang, W. H. Liu, C. H. Tsai, S. S. Sheu, P. C. Chiang, W. P. Lin, C. H. Lin, W. S. Chen, F. T. Chen, C. H. Lien, and M.-J. Tsai, “Evidence and solution of over-RESET problem for HfOX based resistive memory with sub-ns switching speed and high endurance”, in *2010 Int. Electron Devices Meet.*, IEEE, Dec. 2010, pp. 19.7.1–19.7.4, ISBN: 978-1-4424-7418-5. DOI: `10.1109/IEDM.2010.5703395`. [Online]. Available: `http://ieeexplore.ieee.org/document/5703395/`.

[126] M. Lanza, K. Zhang, M. Porti, M. Nafría, Z. Y. Shen, L. F. Liu, J. F. Kang, D. Gilmer, and G. Bersuker, “Grain boundaries as preferential sites for resistive switching in the HfO2 resistive random access memory structures”, *Appl. Phys. Lett.*, vol. 100, no. 12, p. 123 508, Mar. 2012, ISSN: 0003-6951. DOI: `10.1063/1.3697648`. [Online]. Available: `http://aip.scitation.org/doi/10.1063/1.3697648`.

[127] V. Iglesias, M. Porti, M. Nafría, X. Aymerich, P. Dudek, and G. Bersuker, “Dielectric breakdown in polycrystalline hafnium oxide gate dielectrics investigated by conductive atomic force microscopy”, *J. Vac. Sci. Technol. B, Nanotechnol. Microelectron. Mater. Process. Meas. Phenom.*, vol. 29, no. 1, 01AB02, Jan. 2011, ISSN: 2166-2746. DOI: `10.1116/1.3532945`. [Online]. Available: `http://avs.scitation.org/doi/10.1116/1.3532945`.

[128] A. Grossi, E. Perez, C. Zambelli, P. Olivo, and C. Wenger, “Performance and reliability comparison of 1T-1R RRAM arrays with amorphous and polycrystalline HfO2”, in *EUROSOI 2016*, IEEE, Jan. 2016, pp. 80–83, ISBN: 978-1-4673-8609-8. DOI: `10.1109/ULIS.2016.7440057`. [Online]. Available: `http://ieeexplore.ieee.org/document/7440057/`.

[129] Z. Fang, X. P. Wang, J. Sohn, B. B. Weng, Z. P. Zhang, Z. X. Chen, Y. Z. Tang, G. Q. Lo, J. Provine, S. S. Wong, H. S. Wong, and D. L. Kwong, “The role of ti capping layer in HfOx-Based RRAM Devices”, *IEEE Electron Device Lett.*, vol. 35, no. 9, pp. 912–914, 2014, ISSN: 07413106. DOI: `10.1109/LED.2014.2334311`.

[130] K. Beckmann, J. Holt, W. Olin-Ammentorp, Z. Alamgir, J. Van Nostrand, and N. C. Cady, “The effect of reactive ion etch (RIE) process conditions on ReRAM device performance”, *Semicond. Sci. Technol.*, vol. 32, no. 9, p. 095 013, Sep. 2017. DOI: `10.1088/1361-6641/aa7eed`. [Online]. Available: `http://stacks.iop.org/0268-1242/32/i=9/a=095013?key=crossref.c28efe83f8f4a3e411ae7b040097cbdc`.

[131] Y. Hayakawa, A. Himeno, R. Yasuhara, W. Boullart, E. Vecchio, T. Vandeweyer, T. Witters, D. Crotti, M. Jurczak, S. Fujii, S. Ito, Y. Kawashima, Y. Ikeda, A. Kawahara, K. Kawai, Z. Wei, S. Muraoka, K. Shimakawa, T. Mikawa, and S. Yoneda, “Highly reliable TaOx ReRAM with centralized filament for 28-nm embedded application”, in *VLSI Tech. 2015*, IEEE, Jun. 2015, T14–T15, ISBN: 978-4-86348-501-3. DOI:

10.1109/VLSIT.2015.7223684. [Online]. Available: http://ieeexplore.ieee.org/document/7223684/.

[132] D. Niu, Y. Chen, C. Xu, and Y. Xie, “Impact of process variations on emerging memristor”, in *DAC ’10*, New York, New York, USA: ACM Press, 2010, p. 877, ISBN: 9781450300025. DOI: 10.1145/1837274.1837495. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1837274.1837495.

[133] J. H. Nickel, J. P. Strachan, M. D. Pickett, C. Tom Schamp, J. J. Yang, J. A. Graham, and R. S. Williams, “Memristor structures for high scalability: Non-linear and symmetric devices utilizing fabrication friendly materials and processes”, *Microelectron. Eng.*, vol. 103, pp. 66–69, Mar. 2013. DOI: 10.1016/J.MEE.2012.09.007. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167931712005114.

[134] A. Grossi, C. Zambelli, P. Olivo, E. Miranda, V. Stikanov, C. Walczyk, and C. Wenger, “Electrical characterization and modeling of pulse-based forming techniques in RRAM arrays”, *Solid. State. Electron.*, vol. 115, pp. 17–25, Jan. 2016. DOI: 10.1016/J.SSE.2015.10.003. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0038110115002828.

[135] C. Nguyen, C. Cagli, G. Molas, B. Sklenard, C. Nail, K. El Hajjam, J.-F. Nodin, C. Charpin, S. Bernasconi, and G. Reimbold, “Study of Forming Impact on 4Kbit RRAM Array Performances and Reliability”, in *IMW 2017*, IEEE, May 2017, pp. 1–4, ISBN: 978-1-5090-3274-7. DOI: 10.1109/IMW.2017.7939105. [Online]. Available: http://ieeexplore.ieee.org/document/7939105/.

[136] N. Raghavan, “Performance and reliability trade-offs for high-κ RRAM”, *Microelectron. Reliab.*, vol. 54, no. 9-10, pp. 2253–2257, Sep. 2014. DOI: 10.1016/J.MICROREL.2014.07.135. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0026271414003370.

[137] P. Lorenzi, R. Rao, and F. Irrera, “Forming Kinetics in HfO2 -Based RRAM Cells”, *IEEE Trans. Electron Devices*, vol. 60, no. 1, pp. 438–443, Jan. 2013. DOI: 10.1109/TED.2012.2227324. [Online]. Available: http://ieeexplore.ieee.org/document/6365260/.

[138] Y.-S. Chen, W.-H. Liu, H.-Y. Lee, P.-S. Chen, S.-M. Wang, C.-H. Tsai, Y.-Y. Hsu, P.-Y. Gu, W.-S. Chen, F. Chen, C.-H. Lien, and M.-J. Tsai, “Impact of compliance current overshoot on high resistance state, memory performance, and device yield of HfOx based resistive memory and its solution”, in *Proc. 2011 Int. Symp. VLSI Technol. Syst. Appl.*, IEEE, Apr. 2011, pp. 1–2, ISBN: 978-1-4244-8493-5. DOI: 10.1109/VTSA.2011.5872252. [Online]. Available: http://ieeexplore.ieee.org/document/5872252/.

[139] H.-L. Chang, H.-C. Li, C. W. Liu, F. Chen, and M.-J. Tsai, “Physical mechanism of HfO2-based bipolar resistive random access memory”, in *Proc. 2011 Int. Symp. VLSI Technol. Syst. Appl.*, IEEE, Apr. 2011, pp. 1–2, ISBN: 978-1-4244-8493-5. DOI: 10.1109/VTSA.2011.5872253. [Online]. Available: http://ieeexplore.ieee.org/document/5872253/.

[140] A. Kalantarian, G. Bersuker, D. Gilmer, D. Veksler, B. Butcher, A. Padovani, O. Pirrotta, L. Larcher, R. Geer, Y. Nishi, and P. Kirsch, "Controlling uniformity of RRAM characteristics through the forming process", in *IRPS 2012*, IEEE, Apr. 2012, pp. 6C.4.1–6C.4.5, ISBN: 978-1-4577-1680-5. DOI: 10.1109/IRPS.2012.6241874. [Online]. Available: http://ieeexplore.ieee.org/document/6241874/.

[141] E. I. Vatajelu, P. Prinetto, M. Taouil, and S. Hamdioui, "Challenges and Solutions in Emerging Memory Testing", *IEEE Trans. Emerg. Top. Comput.*, pp. 1–1, 2017. DOI: 10.1109/TETC.2017.2691263. [Online]. Available: http://ieeexplore.ieee.org/document/7894207/.

[142] H. García, J. Boo, G. Vinuesa, Ó. G. Ossorio, B. Sahelices, S. Dueñas, H. Castán, M. B. González, F. Campabadal, and F. Khateb, "Influences of the Temperature on the Electrical Properties of HfO2-Based Resistive Switching Devices", *Electron. 2021, Vol. 10, Page 2816*, vol. 10, no. 22, p. 2816, Nov. 2021, ISSN: 2079-9292. DOI: 10.3390/ELECTRONICS10222816. [Online]. Available: https://www.mdpi.com/2079-9292/10/22/2816/htm%20https://www.mdpi.com/2079-9292/10/22/2816.

[143] Y. Y. Chen, R. Degraeve, S. Clima, B. Govoreanu, L. Goux, A. Fantini, G. S. Kar, G. Pourtois, G. Groeseneken, D. J. Wouters, and M. Jurczak, "Understanding of the endurance failure in scaled HfO2-based 1T1R RRAM through vacancy mobility degradation", in *IEDM 2012*, IEEE, Dec. 2012, pp. 20.3.1–20.3.4, ISBN: 978-1-4673-4871-3. DOI: 10.1109/IEDM.2012.6479079. [Online]. Available: http://ieeexplore.ieee.org/document/6479079/.

[144] 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", *IEEE J. Solid-State Circuits*, pp. 1–1, Aug. 2021, ISSN: 0018-9200. DOI: 10.1109/JSSC.2021.3101209.

[145] A. Ghofrani, M. A. Lastras-Montaño, S. Gaba, M. Payvand, W. Lu, L. Theogarajan, and K. T. Cheng, "A low-power variation-aware adaptive write scheme for access-transistor-free memristive memory", *ACM J. Emerg. Technol. Comput. Syst.*, vol. 12, no. 1, pp. 1–18, Jul. 2015, ISSN: 15504840. DOI: 10.1145/2717313. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2810396.2717313.

[146] Z. Al-Ars, S. Hamdioui, and A. A. van de Goor, "Space of DRAM Fault Models and Corresponding Testing", in *Proc. Des. Autom. Test Eur. Conf.*, IEEE, 2006, pp. 1–6, ISBN: 3-9810801-1-4. DOI: 10.1109/DATE.2006.244080. [Online]. Available: http://ieeexplore.ieee.org/document/1657087/.

[147] A. Van de Goor and Z. Al-Ars, "Functional memory faults: a formal notation and a taxonomy", in *Proc. 18th IEEE VLSI Test Symp.*, IEEE Comput. Soc, 2000, pp. 281–289, ISBN: 0-7695-0613-5. DOI: 10.1109/VTEST.2000.843856. [Online]. Available: http://ieeexplore.ieee.org/document/843856/.

[148] S. Hamdioui, *Testing Static Random Access Memories*, ser. Frontiers in Electronic Testing. Boston, MA: Springer US, 2004, vol. 26, p. 221, ISBN: 978-1-4419-5430-5. DOI: 10.1007/978-1-4757-6706-3. [Online]. Available: http://link.springer.com/10.1007/978-1-4757-6706-3.

[149] A. van De Goor, "Using march tests to test SRAMs", *IEEE Des. Test Comput.*, vol. 10, no. 1, pp. 8–14, Mar. 1993. DOI: 10.1109/54.199799. [Online]. Available: http://ieeexplore.ieee.org/document/199799/.

[150] S. Kvatinsky, D. Belousov, S. Liman, G. Satat, N. Wald, E. G. Friedman, A. Kolodny, and U. C. Weiser, "MAGIC–Memristor-Aided Logic", *IEEE Trans. Circuits Syst. II Express Briefs*, vol. 61, no. 11, pp. 895–899, Nov. 2014. DOI: 10.1109/TCSII.2014.2357292. [Online]. Available: http://ieeexplore.ieee.org/document/6895258/.

[151] MathWorks, *MATLAB R2019b*, 2019. [Online]. Available: https://mathworks.com/products/matlab.html.

[152] A. Benso, A. Bosio, S. Di Carlo, G. Di Natale, and P. Prinetto, "Automatic march tests generation for static and dynamic faults in SRAMs", in *Proc. 10th IEEE Eur. Test Symp. ETS 2005*, vol. 2005, 2005, pp. 122–127. DOI: 10.1109/ETS.2005.8.

[153] Q. Luo, X. Xu, T. Gong, H. Lv, D. Dong, H. Ma, P. Yuan, J. Gao, J. Liu, Z. Yu, J. Li, S. Long, Q. Liu, and M. Liu, "8-Layers 3D vertical RRAM with excellent scalability towards storage class memory applications", in *IEDM 2017*, IEEE, Dec. 2017, pp. 2.7.1–2.7.4, ISBN: 978-1-5386-3559-9. DOI: 10.1109/IEDM.2017.8268315. [Online]. Available: http://ieeexplore.ieee.org/document/8268315/.

[154] T. F. Wu, B. Q. Le, R. Radway, A. Bartolo, W. Hwang, S. Jeong, H. Li, P. Tandon, E. Vianello, P. Vivet, E. Nowak, M. K. Wootters, H. .-.-S. P. Wong, M. M. S. Aly, E. Beigne, and S. Mitra, "14.3 A 43pJ/Cycle Non-Volatile Microcontroller with 4.7μs Shutdown/Wake-up Integrating 2.3-bit/Cell Resistive RAM and Resilience Techniques", in *2019 IEEE Int. Solid-State Circuits Conf. -*, IEEE, Feb. 2019, pp. 226–228, ISBN: 978-1-5386-8531-0. DOI: 10.1109/ISSCC.2019.8662402. [Online]. Available: https://ieeexplore.ieee.org/document/8662402/.

[155] L. Wu, M. Taouil, S. Rao, E. J. Marinissen, and S. Hamdioui, "Electrical Modeling of STT-MRAM Defects", in *2018 IEEE Int. Test Conf.*, IEEE, Oct. 2018, pp. 1–10, ISBN: 978-1-5386-8382-8. DOI: 10.1109/TEST.2018.8624749. [Online]. Available: https://ieeexplore.ieee.org/document/8624749/.

[156] L. Larcher, O. Pirrotta, F. M. Puglisi, A. Padovani, P. Pavan, and L. Vandelli, "Progresses in Modeling HfOx RRAM Operations and Variability", *ECS Trans.*, vol. 64, no. 14, pp. 49–60, Aug. 2014, ISSN: 1938-6737. DOI: 10.1149/06414.0049ecst. [Online]. Available: http://ecst.ecsdl.org/cgi/doi/10.1149/06414.0049ecst.

[157] A. Padovani, L. Larcher, O. Pirrotta, L. Vandelli, and G. Bersuker, "Microscopic modeling of HfOx RRAM operations: From forming to switching", *IEEE Trans. Electron Devices*, vol. 62, no. 6, pp. 1998–2006, Jun. 2015, ISSN: 00189383. DOI: 10.1109/TED.2015.2418114.

[158] E. A. Miranda, C. Walczyk, C. Wenger, and T. Schroeder, "Model for the Resistive Switching Effect in HfO2 MIM Structures Based on the Transmission Properties of Narrow Constrictions", *IEEE Electron Device Lett.*, vol. 31, no. 6, pp. 609–611, Jun. 2010, ISSN: 0741-3106. DOI: 10.1109/LED.2010.2046310. [Online]. Available: http://ieeexplore.ieee.org/document/5454260/.

[159] H. Li, P. Huang, B. Gao, B. Chen, X. Liu, and J. Kang, "A SPICE Model of Resistive Random Access Memory for Large-Scale Memory Array Simulation", *IEEE Electron Device Lett.*, vol. 35, no. 2, pp. 211–213, Feb. 2014. DOI: 10.1109/LED.2013.2293354. [Online]. Available: http://ieeexplore.ieee.org/document/6684558/.

[160] C. La Torre, A. F. Zurhelle, T. Breuer, R. Waser, and S. Menzel, "Compact modeling of complementary switching in oxide-based ReRAM devices", *IEEE Trans. Electron Devices*, vol. 66, no. 3, pp. 1265–1275, Mar. 2019, ISSN: 00189383. DOI: 10.1109/TED.2019.2892997. [Online]. Available: https://ieeexplore.ieee.org/document/8630083.

[161] L. Wu, S. Rao, G. Cardoso Medeiros, M. Taouil, E. J. Marinissen, F. Yasin, S. Couet, S. Hamdioui, and G. Sankar Kar, "Pinhole Defect Characterization and Fault Modeling for STT-MRAM Testing", in *Proc. 24th Eur. Test Symp. 2019*, 2019.

[162] A. J. van de Goor, *Testing Semiconductor Memories - Theory and Practice*. John Wiley & Sons, 1991, ISBN: 0471925861.

[163] S. Hamdioui, Z. Al-Ars, and A. van de Goor, "Opens and Delay Faults in CMOS RAM Address Decoders", *IEEE Trans. Comput.*, vol. 55, no. 12, pp. 1630–1639, Dec. 2006, ISSN: 0018-9340. DOI: 10.1109/TC.2006.203. [Online]. Available: http://ieeexplore.ieee.org/document/1717393/.

[164] A. van de Goor, S. Hamdioui, and R. Wadsworth, "Detecting faults in the peripheral circuits and an evaluation of SRAM tests", in *2004 Int. Conferce Test*, IEEE, 2004, pp. 114–123, ISBN: 0-7803-8580-2. DOI: 10.1109/TEST.2004.1386943. [Online]. Available: http://ieeexplore.ieee.org/document/1386943/.

[165] S. Hamdioui and A. J. van de Goor, "Efficient tests for realistic faults in dual-port SRAMs", in *IEEE TC*, 2002. [Online]. Available: https://dl.acm.org/citation.cfm?id=3130553.

[166] S. Hamdioui and A. Van De Goor, "Address decoder faults and their tests for two-port memories", in *Proceedings. Int. Work. Mem. Technol. Des. Test.*, IEEE Comput. Soc, 1998, pp. 97–103, ISBN: 0-8186-8494-1. DOI: 10.1109/MTDT.1998.705954. [Online]. Available: http://ieeexplore.ieee.org/document/705954/.

[167] M. J. Pelgrom, A. C. Duinmaijer, and A. P. Welbers, "Matching Properties of MOS Transistors", *IEEE J. Solid-State Circuits*, vol. 24, no. 5, pp. 1433–1439, 1989, ISSN: 1558173X. DOI: 10.1109/JSSC.1989.572629.

[168] J. Yu, H. A. Du Nguyen, M. A. Lebdeh, M. Taouil, and S. Hamdioui, "Enhanced Scouting Logic: A Robust Memristive Logic Design Scheme", *NANOARCH 2019 - 15th IEEE/ACM Int. Symp. Nanoscale Archit. Proc.*, Jul. 2019. DOI: 10.1109/NANOARCH47378.2019.181296.

[169] L. Wu, S. Rao, M. Taouil, G. Cardoso Medeiros, M. Fieback, E. J. Marinissen, G. S. Kar, and S. Hamdioui, “Defect and Fault Modeling Framework for STT-MRAM Testing”, *IEEE Trans. Emerg. Top. Comput.*, pp. 1–1, 2019, ISSN: 2168-6750. DOI: `10.1109/TETC.2019.2960375`. [Online]. Available: `https://ieeexplore.ieee.org/document/8935208/`.

Curriculum Vitæ

Moritz Christiaan Reiner Fieback

28-09-1993	Born in Leiderdorp, The Netherlands.

Education

2011–2015	B.Sc. degree in Electrical Engineering Delft University of Technology, Delft, The Netherlands
2015–2017	M.Sc. degree in Electrical Engineering Delft University of Technology, Delft, The Netherlands *Thesis:* DRAM Reliability: Aging Analysis and Reliability Prediction Model *Supervisor:* Prof. dr. ir. S. Hamdioui
2018–2022	PhD. degree in Computer Engineering Delft University of Technology, Delft, The Netherlands *Thesis:* Testing RRAM and Computation-in-Memory Devices: Defects, Fault Models, and Test Solutions *Promotor:* Prof. dr. ir. S. Hamdioui

Awards and Honors

2018	Best Paper Award at LATS 2018
2020	Nominated Best Paper Award at ETS 2020
2021	Best Paper Award at ETS 2021
2022	Nominated Best Paper Award at ETS 2022

LIST OF PUBLICATIONS

22. **Moritz Fieback**, Christopher Münch, Anteneh Grebegiorgis, Guilherme Cardoso Medeiros, Mottaqiallah Taouil, Said Hamdioui, Mehdi Tahoori, *"PVT Analysis for RRAM and STT-MRAM-based Logic Computation-in-Memory"*, presented at 2022 European Test Symposium (ETS), 2022. **Nominated for Best Paper Award**

21. **M. Fieback**, G. C. Medeiros, L. Wu, H. Aziza, R. Bishnoi, M. Taouil, S. Hamdioui, *"Defects, Fault Modeling, and Test Development Framework for RRAMs"*, ACM Journal on Emerging Technologies in Computing Systems, Volume 18, Issue 3, 2022, doi: 10.1145/3510851.

20. E. Brum, **M. Fieback**, T. S. Copetti, H. Jiayi, S. Hamdioui, F. Vargas, L. M. Bolzani Poehls, *"Evaluating the Impact of Process Variation on RRAMs"*, 2021 IEEE 22nd Latin American Test Symposium (LATS), 2021, pp. 1-6, doi: 10.1109/LATS53581.2021.9651789.

19. L.M.B. Poehls, **M.C.R. Fieback**, S Hoffmann-Eifert, T. Copetti, E. Brum, S. Menzel, S. Hamdioui, T. Gemmeke, *"Review of Manufacturing Process Defects and Their Effects on Memristive Devices"*, J Electron Test 37, 427–437, 2021, dot: 10.1007/s10836-021-05968-8.

18. G. Cardoso Medeiros, **M. Fieback**, T.S. Copetti, A. Gebregiorgis, M. Taouil, L.B. Poehls, S. Hamdioui, *"Improving the Detection of Undefined State Faults in FinFET SRAMs"*, 2021 16th International Conference on Design & Technology of Integrated Systems in Nanoscale Era (DTIS), 2021, pp. 1-6, doi: 10.1109/DTIS53253.2021.9505130.

17. **M. Fieback**, G. C. Medeiros, A. Gebregiorgis, H. Aziza, M. Taouil and S. Hamdioui, *"Intermittent Undefined State Fault in RRAMs"*, 2021 IEEE European Test Symposium (ETS), 2021, pp. 1-6, doi: 10.1109/ETS50041.2021.9465401. **Best Paper Award**

16. G. C. Medeiros, **M. Fieback**, A. Gebregiorgis, M. Taouil, L. B. Poehls and S. Hamdioui, *"Detecting Random Read Faults to Reduce Test Escapes in FinFET SRAMs"*, 2021 IEEE European Test Symposium (ETS), 2021, pp. 1-6, doi: 10.1109/ETS50041.2021.9465441.

15. G. Cardoso Medeiros, **M. Fieback**, L. Wu, M. Taouil, L. M. Bolzani Poehls and S. Hamdioui, *"Hard-to-Detect Fault Analysis in FinFET SRAMs"*, in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 29, no. 6, pp. 1271-1284, June 2021, doi: 10.1109/TVLSI.2021.3071940.

14. H. Aziza, S. Hamdioui, **M. Fieback**, M. Taouil and M. Moreau, *"Density Enhancement of RRAMs using a RESET Write Termination for MLC Operation"*, 2021 Design, Automation & Test in Europe Conference & Exhibition (DATE), 2021, pp. 1877-1880, doi: 10.23919/DATE51398.2021.9473967.

13. H. Aziza, S. Hamdioui, **M. Fieback**, M. Taouil, M. Moreau, P. Girard, A. Virazel, K. Coulié, *"Multi-Level Control of Resistive RAM (RRAM) Using a Write Termination to Achieve 4 Bits/Cell in High Resistance State"*, Electronics 2021, 10, 2222. doi: 10.3390/electronics10182222.

12. H. Aziza, M. Moreau, **M. Fieback**, M. Taouil and S. Hamdioui, *"An Energy-Efficient Current-Controlled Write and Read Scheme for Resistive RAMs (RRAMs)"*, in IEEE Access, vol. 8, pp. 137263-137274, 2020, doi: 10.1109/ACCESS.2020.3011647.
11. L. Wu, **M. Fieback**, M. Taouil and S. Hamdioui, *"Device-Aware Test for Emerging Memories: Enabling Your Test Program for DPPB Level"*, 2020 IEEE European Test Symposium (ETS), 2020, pp. 1-2, doi: 10.1109/ETS48528.2020.9131559.
10. **M. Fieback**, S. Nagarajan, R. Bishnoi, M. Tahoori, M. Taouil and S. Hamdioui, *"Testing Scouting Logic-Based Computation-in-Memory Architectures"*, 2020 IEEE European Test Symposium (ETS), 2020, pp. 1-6, doi: 10.1109/ETS48528.2020.9131604. **Nominated for Best Paper Award**
9. Rajendra Bishnoi, Lizhou Wu, **Moritz Fieback**, Christopher Münch, Sarath Mohanachandran Nair, Mehdi Tahoori, Ying Wang, Huawei Li, Said Hamdioui, *"Special Session – Emerging Memristor Based Memory and CIM Architecture: Test, Repair and Yield Analysis"*, 2020 IEEE 38th VLSI Test Symposium (VTS), 2020, pp. 1-10, doi: 10.1109/VTS48691.2020.9107595.
8. Guilherme Cardoso Medeiros, Cemil Cem Gürsoy, Lizhou Wu, **Moritz Fieback**, Maksim Jenihhin, Mottaqiallah Taouil, Said Hamdioui, *"A DFT Scheme to Improve Coverage of Hard-to-Detect Faults in FinFET SRAMs"*, 2020 Design, Automation & Test in Europe Conference & Exhibition (DATE), 2020, pp. 792-797, doi: 10.23919/DATE48585.2020.9116278.
7. Lizhou Wu, Siddharth Rao, Mottaqiallah Taouil, Guilherme Cardoso Medeiros, **Moritz Fieback**, Erik Jan Marinissen, Gouri Sankar Kar, Said Hamdioui, *"Defect and Fault Modeling Framework for STT-MRAM Testing"*, in IEEE Transactions on Emerging Topics in Computing, vol. 9, no. 2, pp. 707-723, 1 April-June 2021, doi: 10.1109/TETC.2019.2960375.
6. **Moritz Fieback**, Lizhou Wu, Guilherme Cardoso Medeiros, Hassen Aziza, Siddharth Rao, Erik Jan Marinissen, Mottaqiallah Taouil, Said Hamdioui, *"Device-Aware Test: A New Test Approach Towards DPPB Level"*, 2019 IEEE International Test Conference (ITC), 2019, pp. 1-10, doi: 10.1109/ITC44170.2019.9000134.
5. S. Hamdioui, **M. Fieback**, S. Nagarajan and M. Taouil, *"Testing Computation-in-Memory Architectures Based on Emerging Memories"*, 2019 IEEE International Test Conference (ITC), 2019, pp. 1-10, doi: 10.1109/ITC44170.2019.9000117.
4. A. Bosio, I. O'Connor, G.S. Rodrigues, F.K. Lima, E.I. Vatajelu, G. Di Natale, L. Anghel, S. Nagarajan, **M.C.R. Fieback**, S. Hamdioui, *"Rebooting Computing: The Challenges for Test and Reliability"*, 2019 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT), 2019, pp. 8138-8143, doi: 10.1109/DFT.2019.8875270.
3. G. C. Medeiros, M. Taouil, **M. Fieback**, L. B. Poehls and S. Hamdioui, *"DFT Scheme for Hard-to-Detect Faults in FinFET SRAMs"*, 2019 IEEE European Test Symposium (ETS), 2019, pp. 1-2, doi: 10.1109/ETS.2019.8791517.
2. **M. Fieback**, M. Taouil and S. Hamdioui, *"Testing Resistive Memories: Where are We and What is Missing?"*, 2018 IEEE International Test Conference (ITC), 2018, pp. 1-9, doi: 10.1109/TEST.2018.8624895.
1. **M. Fieback**, M. Taouil, S. Hamdioui and M. Rovatti, *"Ionizing radiation modeling in DRAM transistors"*, 2018 IEEE 19th Latin-American Test Symposium (LATS), 2018, pp. 1-6, doi: 10.1109/LATW.2018.8349678. **Best Paper Award**