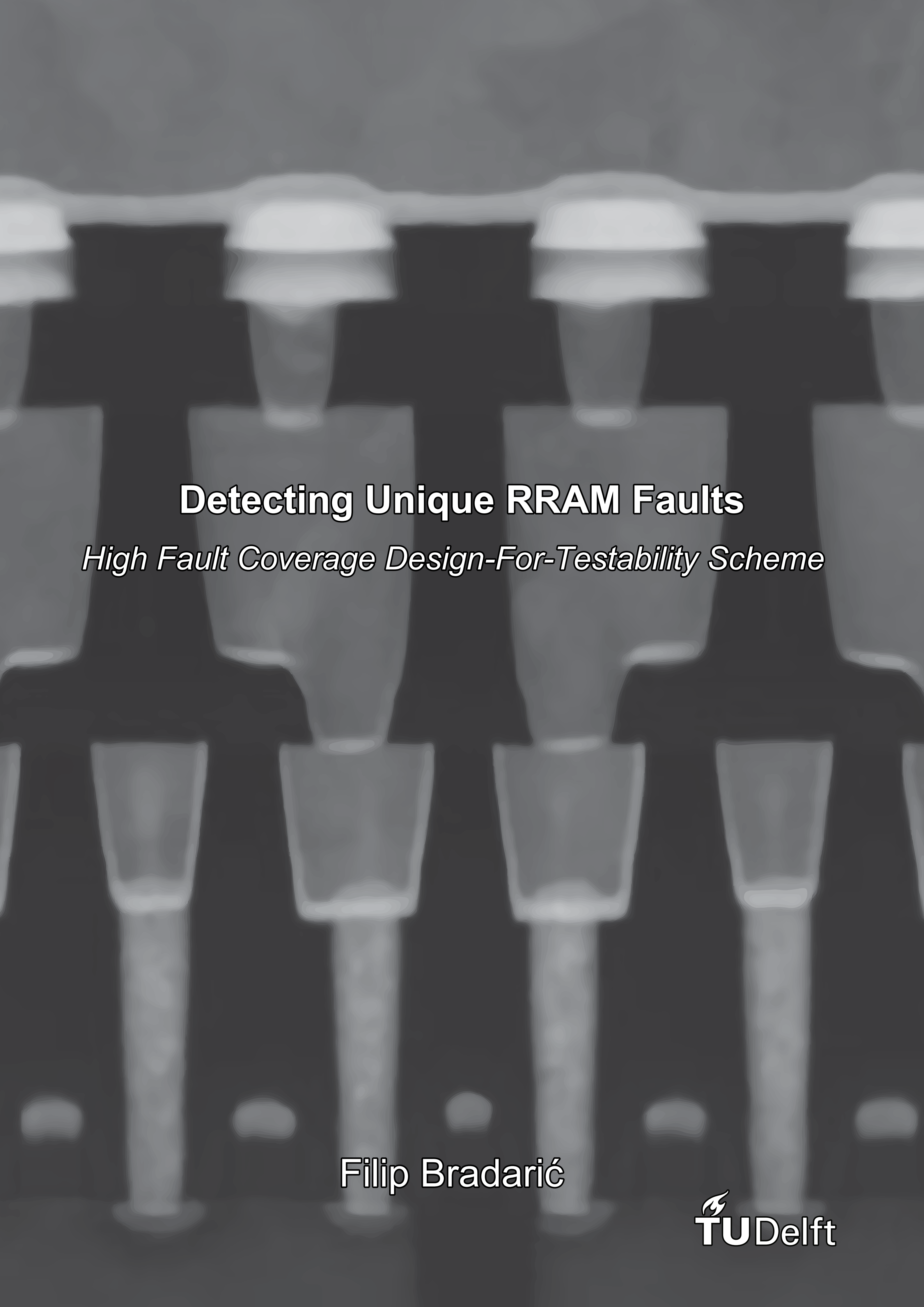


Detecting Unique RRAM Faults

High Fault Coverage Design-For-Testability Scheme

Filip Bradarić

TU Delft

[This Page Intentionally Left Blank]

Detecting Unique RRAM Faults

High Fault Coverage Design-For-Testability Scheme

by

Filip Bradarić
(4643615)

In partial fulfilment of the requirements for the degree of

Master of Science
In Embedded Systems

Thesis Committee:

Prof. Dr. Ir. Said Hamdioui,
Dr. Ir. Nick van der Meijs,
Dr. Ir. Mottaqiallah Taouil,
Ir. Moritz Fieback

Thesis defence:

16 June 2022

Project duration:

29 November 2021 – 16 June 2022

Cover:

Adapted from [1]

Delft University of Technology
Faculty of Electrical Engineering, Mathematics and Computer Science
Embedded Systems Programme
Computer Architecture Specialisation

[This Page Intentionally Left Blank]

Abstract

Resistive Random-Access Memory (RRAM) is an emerging memory technology that has the possibility to compete with mainstream memory technologies such as Dynamic Random-Access Memory (DRAM) and flash memory. What sets RRAM apart from the mainstream memory technologies is that data is stored in terms of resistance instead of charge, which means that RRAM does not suffer from scaling limitations related to charge storage. Furthermore, RRAM is a Non-Volatile Memory (NVM) with a high memory density, low power consumption and Back-End-Of-Line (BEOL) compatibility with the standard Complementary Metal-Oxide-Semiconductor (CMOS) process.

The reason why RRAM has not seen mass adoption yet is due to its defect-prone nature. Since defects are complex to analyse, because they are low-level physical deviations, fault models have been developed that abstract from the physical aspect and only focus on the high-level effect of the defect. Next to conventional faults, that are present in every CMOS process, RRAM also suffers from unique faults specific to its fabrication process and working principle. RRAM is not a binary device but an analogue device. The resistance of RRAM can assume any resistance value within its operating range. For this reason, the operating range of RRAM can be divided into five states instead of the regular two logic states. Conventional test techniques are incapable of detecting unique faults due to their inability to distinguish between the five cell states, resulting in a large number of test escapes. Therefore, new test methods, such as Design-For-Testability (DFT), need to be developed to reduce the number of test escapes and ensure customer satisfaction.

Throughout the past years, multiple DFTs have been developed to reduce the number of test escapes. However, the state-of-the-art DFTs are still incapable of detecting all the identified faults. For example, the ability to detect intermittent faults is still missing. Moreover, the current state of the art is lacking in a few regions. To be more precise, the lack of CMOS implementation details, oversimplification during the design stage and missing proper validation of the implemented design are gaps that are currently present in the state of the art.

This work aims to bridge the aforementioned gaps in the state of the art by proposing two new DFTs: Parallel-Reference Read (PRR) and Closed-Loop Write (CLW). The PRR DFT is a replacement for the regular read circuit, which enables the detection of all five cell states, while the CLW DFT is an addition to the regular write circuit, which introduces feedback during the write operation. From these two DFTs, the PRR DFT is selected for further development and its design is elaborated. Furthermore, complete validation of the PRR DFT is performed.

From the validation, it is concluded that the PRR DFT can detect all five cell states. Moreover, under process variations, the PRR DFT will provide the correct output in 95.90% of the cases. Furthermore, the PRR DFT improves the overall resistive-defect detection capability by 14.79% when compared to a regular read circuit. Finally, the PRR DFT offers 100% identified fault coverage while only requiring $4N$ write operations, $5N$ read operations and an area overhead of $14N_c$ transistors, where N and N_c are the total number of cells and the total number of columns in the RRAM array, respectively. On top of the quantifiable improvements to the state of the art, the PRR DFT also delivers incalculable improvements. By acting as a replacement of the read circuit, the PRR DFT can be used for in-field testing which enables the detection of intermittent faults. Furthermore, it can easily be adapted to the needs of the customer due to its modularity and support for different architectures and different emerging memory technologies. All in all, it can safely be concluded that this work achieved its initial goal of filling the gaps in the state of the art.

[This Page Intentionally Left Blank]

Contents

1. Introduction

This chapter introduces the thesis topic. First, motivation is given behind the importance of the chosen topic. After that, the current state of the art is provided and the contribution of this thesis is presented. Finally, the outline of the thesis is shown.

1.1 Motivation

The most important components of any computational device are the Central Processing Unit (CPU) and the memory. The CPU is used to process data which is provided by the memory. Without the memory, the computational device would be rendered useless since the performed instructions would be of no value.

Today, the most prominent memories are Static Random-Access Memory (SRAM), Dynamic Random-Access Memory (DRAM) and flash memory [2]. These memories are also classified as mainstream memory technologies. The mainstream memory technologies share one thing in common, which is that they all store data (bits) in terms of charge. However, memories based on charge storage are expected to run into serious scaling limitations in the foreseeable future [3].

The concern about scaling limitations of mainstream memory technologies resulted in the research and development of non-charge based memories. These memories are also classified as emerging memory technologies which includes Ferroelectric Random-Access Memory (FeRAM), Spin-Transfer Torque Magnetic Random-Access Memory (STT-MRAM), Phase-Change Random-Access Memory (PCRAM) and Resistive Random-Access Memory (RRAM) [4]. Next to the fact that emerging memory technologies are not based on charge storage, they are also Non-Volatile Memories (NVMs) since data is not lost when the power is turned off. Amongst the emerging memory technologies, RRAM stands out due to a lower power consumption than PCRAM, a greater density than STT-MRAM, a lower latency than FeRAM and Back-End-Of-Line (BEOL) compatibility with the standard Complementary Metal-Oxide-Semiconductor (CMOS) process [4], [5].

RRAM is based on storing data in terms of resistance [5]. A Conductive Filament (CF) can be formed or ruptured inside a metal oxide resulting in a Low Resistance State (LRS) and High Resistance State (HRS), respectively. However, the formation and rupturing of the CF is not deterministic in nature, causing device-to-device and cycle-to-cycle variations. Furthermore, the fabrication of RRAM during the BEOL stage of the CMOS process gives rise to additional defects previously not present in mainstream memory technologies [6].

High-fault-coverage testing after fabrication is essential if the satisfaction of customers wants to be ensured. For mainstream memory technologies, a sequence of read and write operations, called a March test, is most commonly used to test the memories. On top of March tests, DFTs have been developed to further increase the fault coverage. However, the testing methods that are currently being used for mainstream memory technologies prove to be inadequate to provide the same high fault coverage for RRAM as they do for mainstream memory technologies [6]. The main reason is that RRAM introduces unique faults which are not present in mainstream memory technologies, hence they are not covered by their tests [6].

To ensure high-fault-coverage testing of RRAM, new testing methods need to be developed that are specifically designed to cover unique RRAM faults. These new testing methods, also called Design-For-Testability (DFT) schemes, will get the emerging memory technologies one step closer to becoming mainstream.

1.2 State of the art

In the recent years, multiple DFTs have been developed with the aim of increasing the fault coverage and/or reducing the testing time. Hongal *et al.* in [7] introduce an algorithm to efficiently locate the defective cell without having to individually check every single cell inside a RRAM array. With this approach, the testing time can be improved. Liu *et al.* in [8], [9] developed a DFT which allows all the RRAM cells to be checked if they are in HRS at the same time. The added benefit of this DFT is the decrease in time it takes to perform a read 0 operation on the whole RRAM array. Kannan *et al.* in [10] and Li *et al.* in [11] created DFTs that use sneak-paths to their advantage in order to test multiple cells by only performing a read operation on a single cell. Sneak-paths are undesirable and unintended paths within an electrical circuit that should be avoided during normal operation. However, their usefulness is discovered for testing purposes. Hamdioui *et al.* in [12], [13] and Mozaffari *et al.* in [14] take a different approach by introducing DFTs that shorten the write operation in order to improve the unique fault coverage and reduce the testing time. Copetti *et al.* in [15] came with a DFT that includes an on-chip sensor for every cell. With this sensor, the voltage of the internal node can be measured in order to determine the state of the cell. Liu *et al.* in [16] developed DFTs which employ multiple reference currents in order to ease the detection of unique faults. The reference currents are generated in parallel. However, only one reference current is selected based on the operation.

Even though a lot of progress has been made, the state of the art is still lacking in a few regions. First, RRAM introduces many unique faults that are currently not detectable by the state of the art. For example, all the DFTs assume that a fault is always present and can be detected by checking for it only once after fabrication. However, RRAM is also prone to intermittent faults that cannot be detected by the current state of the art [17]. Thus, new DFTs are required that can detect all unique faults and provide the required high fault coverage. On top of the aforementioned issue with the state of the art, most of the DFTs are lacking implementation details. The theoretical working principle is explained and a high-level overview is given for every DFT. However, the actual design in CMOS technology and its validation are left out. Furthermore, simplifications are made that do not consider process variations which makes the implementability of some DFTs quite challenging.

1.3 Contribution

In this work, two new DFTs are proposed: PRR and CLW. The PRR DFT is based on replacing the read circuit with four current comparators in order to detect the actual state of the cell during every read operation. In this way, unique faults, that cause the cell to end up in a non-logic state, can be detected solely by performing one read operation. On the other hand, the CLW DFT is based on introducing feedback to the write circuit in order to prevent or detect faults during every write operation. Adding feedback to the write operation allows for dynamic write time instead of static write time. Using a dynamic write time instead of a static write time prevents faults that occur due to weak cells and process variations while, by imposing a maximum write time, faults that occur due to other defects can be detected.

The contribution of this work can be summarised as follows:

- Analysis of the state-of-the-art test solutions and identification of their drawbacks.
- Identification of DFT targets and possible methods to achieve those targets.
- Proposal of two DFTs that employ different methods to achieve higher fault coverage.
- Fully documented CMOS design and comprehensive validation of the PRR DFT.
- Open-source custom-developed validation framework that can be used to verify the results of this thesis or adapted for own research purposes.

1.4 Outline

The remainder of this thesis is organised as follows. In Chapter 2, the background information that is necessary to understand the contribution of this thesis is provided. Chapter 3 places the focus on RRAM testing by covering and comparing the state-of-the-art testing techniques. In Chapter 4, the possibility of detecting faults through read and write operations is explored and the two new DFTs are proposed. Chapter 5 provides all the details about the test circuit and the PRR DFT design. In Chapter 6, the PRR DFT is validated by analysing its detection capability and resilience against process variations. Finally, a discussion about the work is presented in Chapter 7, while a conclusion is drawn in Chapter 8.

2. Background

This chapter provides the background information necessary to understand the contribution of this thesis. First, the memristor is introduced as the theoretical foundation of RRAM. After that, a basic overview of the emerging memory technologies is presented. Finally, a deep-dive into RRAM is performed.

2.1 Memristor

In the theory of linear circuits, the four fundamental circuit variables are presented as the current i, the voltage v, the charge q and the flux ϕ. The discoveries of the resistor, the capacitor and the inductor have introduced relationships between the four variables, as can be seen in Figure 2.1. However, one relationship was still missing, the relationship between the charge q and the flux ϕ. In 1971, Leon Chua noticed the missing relationship and argued that there should be a fourth fundamental circuit element, called the memristor, to bridge this gap [18]. Chua's contribution was theoretical and it took 37 years for the first physical memristor to be presented. In 2008, HP Labs demonstrated a metal-oxide-metal device with memristive properties [19]. Today, such a device is known as RRAM.

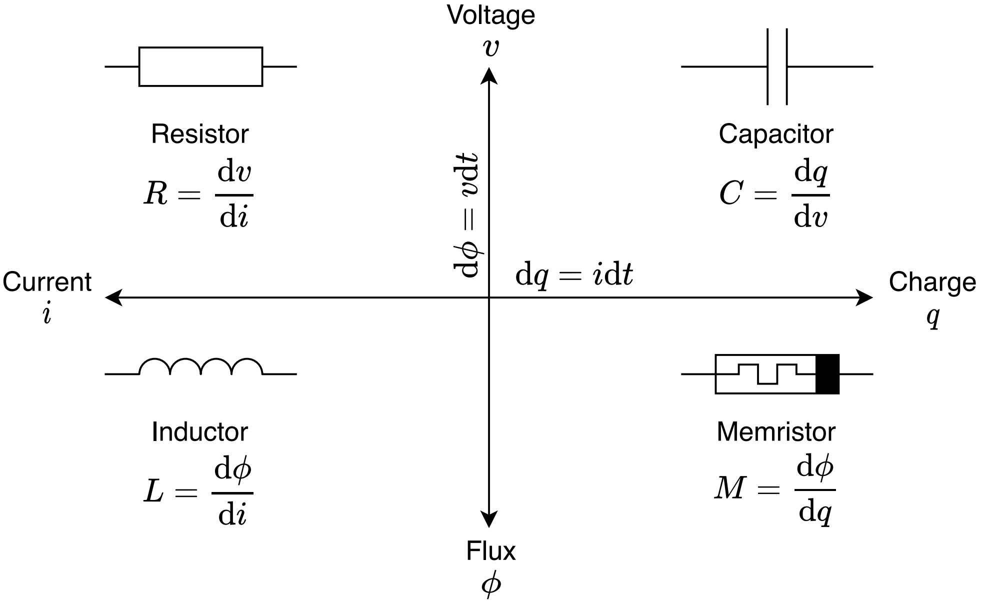

Voltage
v
Resistor
$R = \frac{\mathrm{d}v}{\mathrm{d}i}$
Capacitor
$C = \frac{\mathrm{d}q}{\mathrm{d}v}$
$\mathrm{d}\phi = v\mathrm{d}t$
$\mathrm{d}q = i\mathrm{d}t$
Current
i
Charge
q
Inductor
$L = \frac{\mathrm{d}\phi}{\mathrm{d}i}$
Memristor
$M = \frac{\mathrm{d}\phi}{\mathrm{d}q}$
Flux
ϕ

Figure 2.1: The four fundamental circuit elements. Adapted from [19]

2.2 Emerging memory technologies

Mainstream memory technologies include SRAM, DRAM and flash memory. These memory technologies are based on storing data (bits) in terms of charge. SRAM stores the charge in the nodes of the cross-coupled inverters, DRAM in the cell capacitor and flash memory in the floating gate of the Metal–Oxide–Semiconductor Field-Effect Transistor (MOSFET) [2]. Even though charge storage has been used extensively throughout the years, it is expected to reach scaling limitations in the foreseeable future [3]. For DRAM, as the feature size is reduced, the cell capacitor's area should remain constant in order to ensure reliable sensing. This is achieved by using the third dimension (deep trench capacitor) to gain enough area while reducing the lateral dimensions. However, reducing the feature size to sub-30 nm introduces serious leakage currents [3]. For flash memory, high writing voltages are used to ensure a large barrier for non-volatile retention. However, reducing the feature size to sub-20 nm is extremely difficult since the maximum gate-source voltage V_{gs} is reduced as well [3].

In order to solve the scaling problem, emerging memory technologies are being extensively researched. This includes Ferroelectric Random-Access Memory (FeRAM), Spin-Transfer Torque Magnetic Random-Access Memory (STT-MRAM), Phase-Change Random-Access Memory (PCRAM) and Resistive Random-Access Memory (RRAM) [4]. What makes the emerging memory technologies interesting is that, next to the fact that they are not based on charge storage, they are also Non-Volatile Memories (NVMs). This means that the stored data are not lost when the power is turned off.

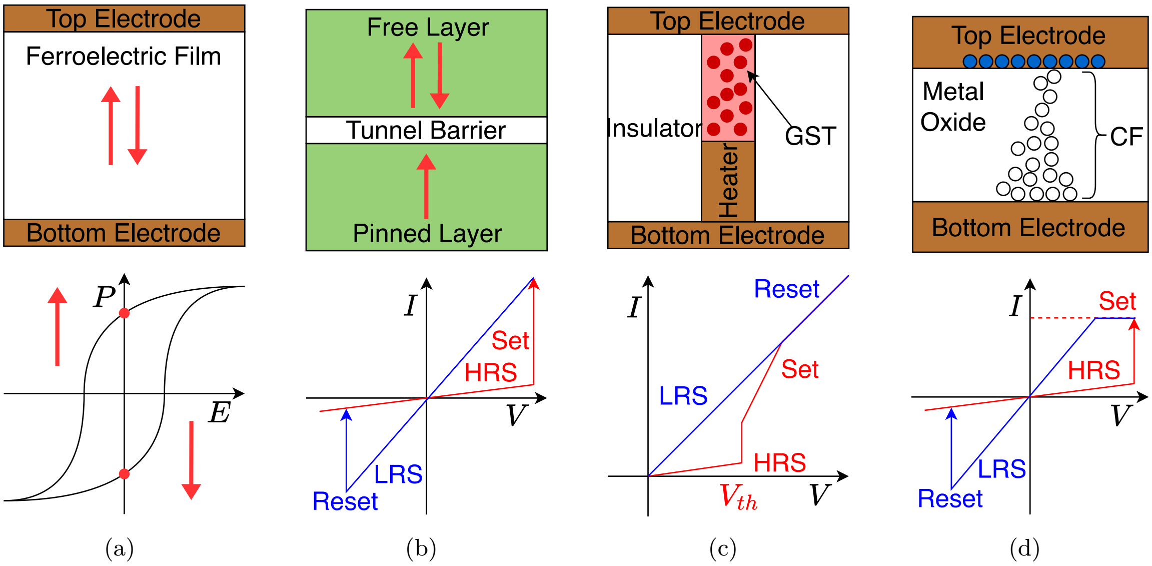


Figure 2.2: Structures and working principles of emerging memory technologies. (a) FeRAM. (b) STT-MRAM. (c) PCRAM. (d) RRAM. Adapted from [2]

2.2.1 FeRAM

FeRAM is based on the Metal-Ferroelectric-Metal (MFM) capacitor structure [20], as can be seen in Figure 2.2(a). The Top Electrode (TE) and Bottom Electrode (BE) are made out of metal while the ferroelectric film is made out of $\mathrm{PbZr_xTi_{1\text{-}x}O_3}$ (PZT) or $\mathrm{SrBi_2Ta_2O_9}$ (SBT) material [21]. The FeRAM device utilises the remanent polarisation of the ferroelectric film in order to store data. The working principle can be seen in Figure 2.2(a), where E denotes the electric field and P denotes the polarisation. If the ferroelectric film is upwards polarised, the FeRAM device is in state 0 while, if the ferroelectric film is downwards polarised, the FeRAM device is in state 1. The data are read out using polarisation reversal current, which is a destructive operation, meaning that the content of the device has to be rewritten after a read operation [20].

2.2.2 STT-MRAM

Similar to FeRAM, STT-MRAM is based on the dipole moment [22]. However, instead of the dipole moment being electric, as is the case with FeRAM, it is magnetic. In Figure 2.2(b), the Magnetic Tunnel Junction (MTJ) structure of STT-MRAM can be seen. It consists of an oxide (MgO) Tunnel Barrier (TB) sandwiched between two ferromagnetic layers called the Free Layer (FL) and the Pinned Layer (PL) [2] or Reference Layer (RL) [22]. The difference between those two layers is that the PL has fixed (pinned) magnetisation while the magnetisation of the FL can be switched. STT-MRAM has two distinguishable states that are based on the magnetisation of the FL. If the magnetisation of the FL has the same orientation as the PL, then the device is in the Parallel (P) state while, if the magnetisation has the opposite orientation, the device is in the Anti-Parallel (AP) state. Distinguishing between the P and AP state during a read operation is not difficult since the magnetisation of the layers affects the resistance of the device, called the Tunnelling MagnetoResistance (TMR) [22]. If the device is in the P state, the resistance is low while, if the device is in the AP state, the resistance is high.

2.2.3 PCRAM

The basic principle of PCRAM revolves around the use of chalcogenide materials, of which GeSbTe (GST) materials are the most common [2]. The structure of a PCRAM can be seen in Figure 2.2(c). By heating the GST material to different temperatures at different speeds, the phase of the material can be changed from crystalline to amorphous and vice versa [23]. If the GST material is in the crystalline phase, it has a low resistance and is said to be in the Low Resistance State (LRS). On the other hand, if the GST material is in the amorphous phase, it has a high resistance and is said to be in the High Resistance State (HRS). These two distinguishable resistance states can be used to store data in the device.

2.2.4 RRAM

RRAM is based on the Metal-Insulator-Metal (MIM) structure, as can be seen in Figure 2.2(d). The MIM structure consists of a metal oxide insulator sandwiched between two metal electrodes (TE and BE) [2]. Similar to PCRAM, there are two distinguishable resistance states, LRS and HRS. However, these two resistance states are achieved through the forming and rupture of Conductive Filaments (CFs) [5]. Since RRAM is the primary focus of this thesis, a more elaborate explanation will be provided in Section 2.3.

2.3 Principles of RRAM

RRAM is an emerging NVM technology that is based on storing data in terms of resistance [5]. To be more precise, the length and width of the CF are responsible for the resistance of the device. By applying voltages of different magnitude and/or polarity to the TE and BE, the CF can be ruptured and regrown, changing the resistance of the device.

Based on the type of CF, RRAM can be divided into two subcategories: Oxide Random-Access Memory (OxRAM) and Conductive Bridge Random-Access Memory (CBRAM) [24]. The difference between OxRAM and CBRAM is that the CF of OxRAM is made out of oxygen vacancies in the metal oxide, while the CF of CBRAM is made out of metal atoms in the solid electrolyte. Both subcategories have similar characteristics. However, OxRAM is the more dominant one. This may be attributed to the fact that OxRAM offers better endurance than CBRAM [2], [25]. In this thesis, OxRAM will be considered when discussing RRAM.

2.3.1 Switching process

There are three fundamental switching modes in RRAM: unipolar, bipolar and complementary [24], [26]. The simplified I–V curves of the switching modes can be seen in Figure 2.3. With unipolar switching, the set and reset voltages are of the same polarity while, with bipolar switching, the set and reset voltages are of different polarity. Complementary switching is similar to unipolar switching, in the sense that the set and reset voltages are of the same polarity, however, their magnitudes are reversed. From the three switching modes, bipolar switching is the most common [24] and will therefore be used in this thesis.

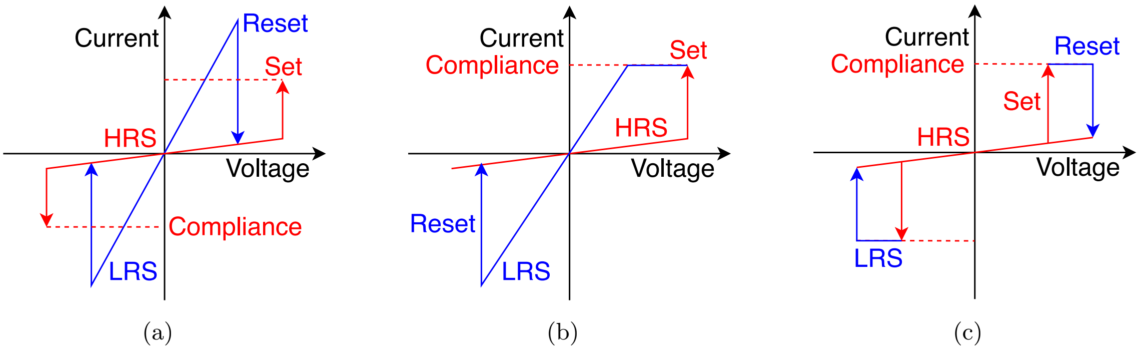


Figure 2.3: RRAM switching modes. (a) Unipolar. (b) Bipolar. (c) Complementary. Adapted from [5]

During its lifetime, RRAM goes through six stages of operation: pre-forming, forming, LRS, reset, HRS and set. A visualisation can be seen in Figure 2.4.

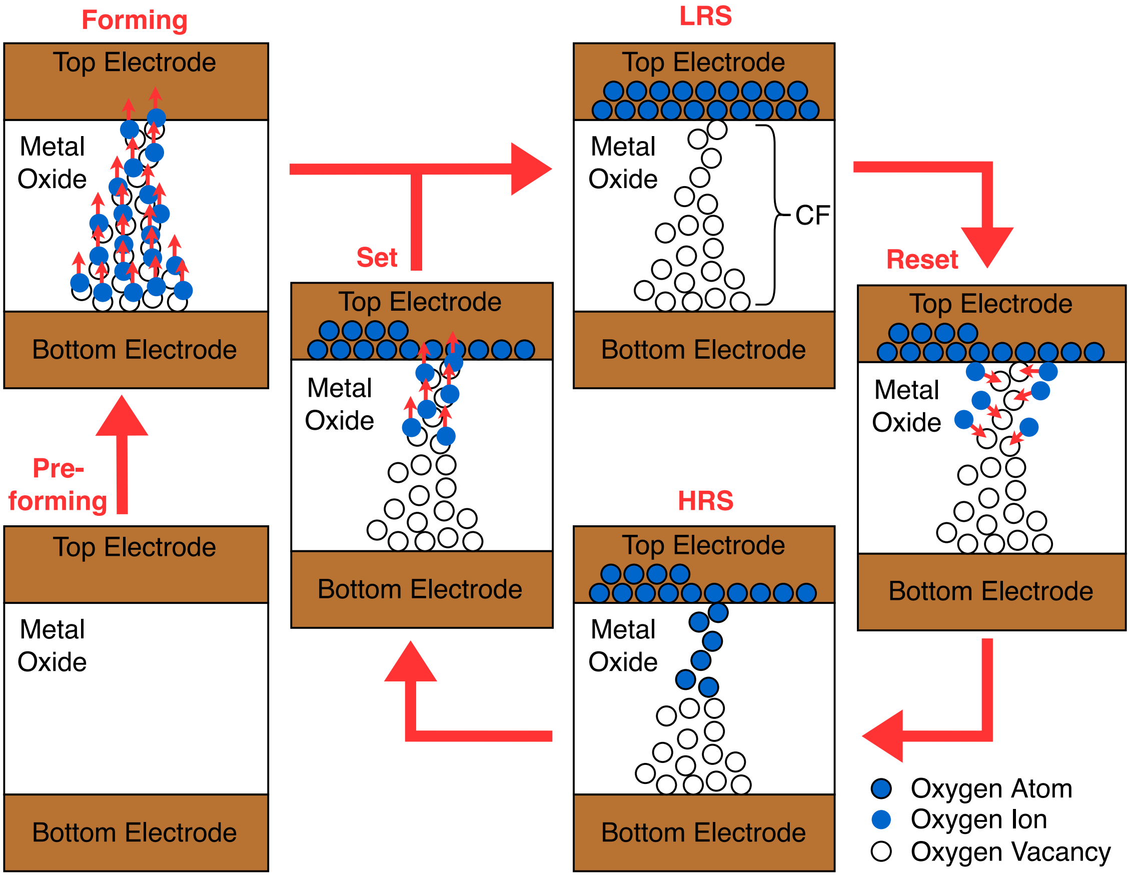


Figure 2.4: RRAM switching process. Adapted from [5]

Pre-forming

After fabrication, the metal oxide has a polycrystalline phase which contains a relatively low number of oxygen vacancies on the grain boundaries [5]. The initially present oxygen vacancies are not enough to form the CF. In this stage, the resistance of the RRAM device is the highest.

Forming

During the forming process, a relatively high positive voltage V_{form} is applied to the TE to cause a soft dielectric breakdown [5]. As a result of the high voltage, a high electric field will be present in the metal oxide causing negative oxygen ions (anions) to drift towards the interface between the TE and the metal oxide, also known as the “oxygen reservoir” [5]. The oxygen anions will leave behind oxygen vacancies in the metal oxide resulting in a conductive path between the TE and BE, called the CF.

Forming is not a mandatory stage. When the forming voltage V_{form} is lower or equal to the set voltage V_{set}, the device is called “forming-free” and the forming stage can be skipped. This is achieved by making the metal oxide layer thinner or introducing more oxygen vacancies during fabrication [24].

LRS

In the LRS, also known as logic state 1, the current flows through the CF resulting in a low resistance of the RRAM device. In this stage, the resistance can be read by applying a read voltage V_{read} to the TE. The read voltage V_{read} should be small enough to ensure that the RRAM device does not change its state.

Reset

During the reset process, a negative voltage lower than V_{reset} is applied to the TE causing the oxygen anions to migrate back to the metal oxide where they recombine with the oxygen vacancies. This recombination will cause a rupture in the CF near the TE since only the oxygen vacancies closest to the TE will be reoxidised [5]. The remaining oxygen vacancies are referred to as the “virtual cathode”.

HRS

In the HRS, also known as logic state 0, there is a gap between the TE and the CF. This gap makes it more difficult for electrons to travel between the TE and BE, resulting in a high resistance of the RRAM device [5]. In this stage, the resistance can be read by applying a read voltage V_{read} to the TE. The read voltage V_{read} should be small enough to ensure that the RRAM device does not change its state.

Set

During the set process, a positive voltage larger than V_{set} is applied to the TE causing the oxygen anions to migrate back to the interface between the TE and the metal oxide. The newly created oxygen vacancies restore the CF to its original shape. After the set process, the RRAM device is back in the LRS and the switching cycle can be repeated.

2.3.2 States

Ideally, the RRAM device will either be in logic state 0 (HRS) or logic state 1 (LRS). However, RRAM is not a binary device but an analogue device. This means that it is capable of assuming any resistance value within its operating range. For this reason, the continuous resistance range is divided into five states [27], as can be seen in Figure 2.5. State 0 and state 1 are the regular logic state 0 and logic state 1, respectively. State U is called the undefined state, which incorporates the resistance range in between the two logic states. In practice, the undefined state is commonly defined as the region from 40% to 60% between the LRS and HRS [12]. State L and state H represent the extremely low and extremely high conductance state, respectively. State L incorporates resistance values higher than the resistance range of state 0, while state H incorporates resistance values lower than the resistance range of state 1. Moreover, state L is also known as the deep 0 state, while state H is also known as the deep 1 state.

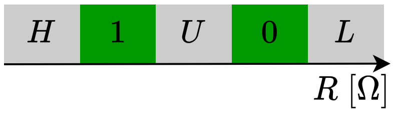


Figure 2.5: RRAM states. Adapted from [27]

2.3.3 Models

In the last two decades, multiple models have been developed in order to capture the behaviour of RRAM devices [19], [28]–[36]. These models range from simple first-order approximations to physics-based models. In [37], the authors performed a comparative analysis of eight models based on a number of metrics that they deemed important. To that comparison, two more models have been added as well as the complementary switching metric. Including complementary switching as a metric gives extra insight in which model to chose. For example, if it is desired to model intermittent faults, then choosing a model that incorporates complementary switching is necessary. This addition extends the comparison to the following models:

- Electrical Switch Circuit model [28]
- Linear Ion Drift model [19]
- Non-Linear Ion Drift model [29]
- Simmons Tunnel Barrier model [30]
- ThrEshold Adaptive Memristor (TEAM) model [31]
- Voltage ThrEshold Adaptive Memristor (VTEAM) model [32]
- Stanford model [33]
- SPICE model [34]
- IM2NP model [35]
- JART VCM v1b [36]

The full comparison can be seen in Table 2.1. From this comparison, it can be concluded that every model serves a different purpose. There is no universal model and, based on the focus of the research, the appropriate model can be chosen through the help of the provided metrics. However, some models are more realistic than others and cover more metrics. For this thesis, the JART VCM v1b [36] model is selected since it covers almost all the metrics while still allowing for acceptable simulation time.

Table 2.1: Comparison of RRAM models. Adapted from [37]

	Model [28] (Switch)	Model [19] (Linear)	Model [29] (Non-linear)	Model [30] (Simmons)	Model [31] (TEAM)	Model [32] (VTEAM)	Model [33] (Stanford)	Model [34] (SPICE)	Model [35] (IM2NP)	Model [36] (JART VCM v1b)
Type of the model	1st order	Ideal physics	Physics-based	Physics-based	Physics-based	Physics-based	Physics-based	Analytical	Physics-based	Physics-based
Efficient use in RRAM arrays	N	N	N	N	N	Y	Y	N	Y	Y
Bipolar switching	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Complementary switching	N	N	N	N	N	N	N	N	N	Y (v2)
Low complexity	Y	Y	N	N	Y	Y	N	Y	Y	Y
Matching the actual memristive behavior	N	N	N	Y	Y	Y	Y	N	Y	Y
Genericity	N	N	N	N	Y	Y	Y	Y	Y	Y
Non-linearity	N	N	Y	Y	Y	Y	Y	Y	Y	Y
Symmetric	Y	Y	N	N	N	N	N	Y	N	N
Voltage-controlled	Y	N	Y	N	N	Y	Y	Y	Y	Y
Hard set	N	N	N	N	Y	Y	Y	Y	Y	Y
Soft reset	N	N	N	N	Y	Y	N	Y	Y	Y
Electro Forming	N	N	N	N	N	N	N	N	Y	N
Support of high frequencies	N	N	N	Y	Y	Y	Y	Y	Y	Y
Threshold	N	N	N	N	Y	Y	Y	Y	Y	Y
Pulse-programming Voltage dependence	N	N	N	N	N	Y	Y	Y	Y	Y
Pulse-programming Timing dependence	N	N	N	N	N	N	Y	N	Y	Y
Temperature dependence	N	N	N	N	N	N	Y	N	Y	Y
Variability dependence	N	N	N	N	N	N	Y	N	Y	Y (v1b var)
Retention	N	N	N	N	N	N	Y	N	N	Y (v2)

2.3.4 Array architectures

Using a single RRAM device as memory will not suffice. Therefore, many RRAM devices are used to form an array. Different architectures exist to realise a RRAM array: One-Resistor (1R), One-Diode-One-Resistor (1D1R), One-Transistor-One-Resistor (1T1R) and CMOS Molecular (CMOL) [5], [38]. These architectures can be seen in Figure 2.6.

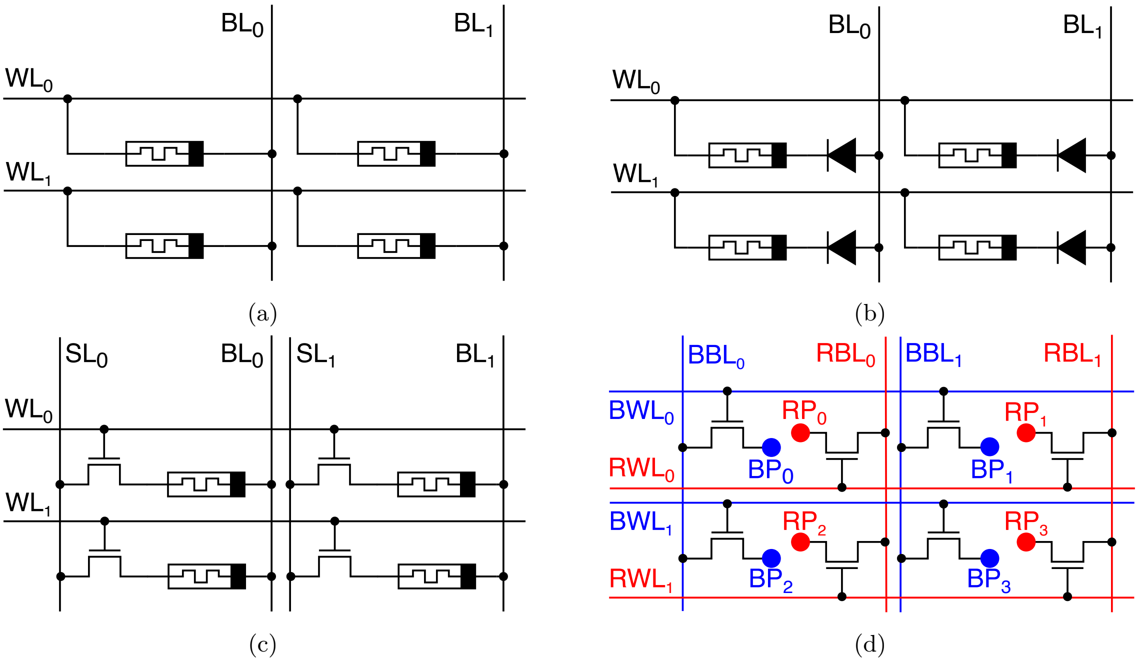


Figure 2.6: RRAM array architectures. (a) 1R. (b) 1D1R. (c) 1T1R. (d) CMOL. Adapted from [2], [10], [28], [39]

1R architecture

The 1R or cross-point architecture consists of two sets of perpendicular nanowires, containing RRAM devices at the cross-points [40]. The horizontal nanowires, called Word Lines (WLs), are used to select rows, while the vertical nanowires, called Bit Lines (BLs), are used to select columns. By selecting a WL and BL, every individual RRAM device can be accessed separately. The advantage of this architecture is that a high memory density can be achieved since every memory cell consists of only a single RRAM device. However, the disadvantage of this architecture is that sneak-paths exist [10].

Sneak-paths are undesirable and unintended paths within an electrical circuit. The current going through these sneak-paths, called the sneak-current, corrupts the output current and may cause incorrect read and write operations. In Figure 2.7, the sneak-paths of a 2×2 1R RRAM array can be seen. The current $I_{primary}$ is the desired output current that goes through the selected cell and is used to determine the logic state of the memristor. However, the actual output current I_{output} is corrupted by the sneak-current I_{sneak} that flows through multiple memristors which can be larger than $I_{primary}$. Therefore, sneak-paths are undesired and should be eliminated during operation.

1D1R architecture

The 1D1R architecture is similar to the 1R architecture with only one difference. At every cross-point, a diode is connected in series with the RRAM device [5]. The addition of the diode prevents sneak-paths since the current can only flow in one direction through the RRAM device. However, this introduces the limitation that only unipolar RRAM devices can be used with this architecture. Next to that, the memory density is lower compared to the 1R architecture.

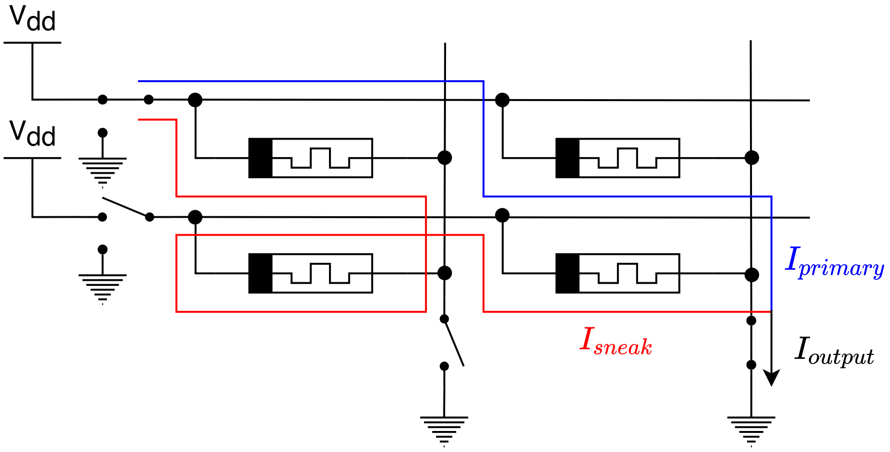


Figure 2.7: Sneak paths in a 2×2 1R RRAM array. Adapted from [10]

1T1R architecture

The 1T1R architecture replaces the diode in the 1D1R architecture with a Negative-Channel Metal-Oxide-Semiconductor (NMOS) transistor [2]. Since NMOS transistors are three-terminal devices, an additional set of nanowires, called the Source Lines (SLs), are required. In the 1T1R architecture, RRAM devices are accessed by activating the desired row using the WL and then selecting the desired column through its BL and SL. Contrary to the 1D1R architecture, the 1T1R architecture blocks sneak-paths while allowing the use of bipolar RRAM devices. However, this comes at the cost of a lower memory density compared to the 1D1R architecture. Another benefit of using an NMOS transistor is the ability to control the maximum current going through the RRAM device, also known as the compliance current. This is achieved by adjusting the WL voltage. By changing the compliance current, the strength of the CF can be controlled [5].

CMOL architecture

The CMOL architecture improves the memory density of the 1T1R architecture by allowing one transistor to select multiple RRAM devices [39]. This is achieved by having two sets of bitlines, Blue Bit Lines (BBLs) and Red Bit Lines (RBLs), and two sets of wordlines, Blue Word Lines (BWLs) and Red Word Lines (RWLs). The BWLs and BBLs can be used to select a specific Blue Point (BP), while the RWLs and RBLs can be used to select a specific Red Point (RP). One BP is connected to the BE of multiple RRAM devices, while one RP is connected to the TE of multiple RRAM devices. However, the combination of one BP and one RP uniquely selects only one RRAM device. Even though the CMOL architecture allows for a higher memory density than the 1T1R architecture, to the best knowledge of the author, it is not used often in practice. A possible explanation is that using two sets of bitlines and wordlines introduces additional complexity in the control circuit and wiring.

2.3.5 Fabrication process

The fabrication process of RRAM consists of three main stages: Front-End-Of-Line (FEOL), Back-End-Of-Line (BEOL) and CF forming [6]. In the FEOL stage, the CMOS transistors are fabricated on the wafer while, in the BEOL stage, the metal layers and RRAM devices are fabricated. Finally, the CF is formed in the CF forming process. A visualisation of the fabrication process can be seen in Figure 2.8.

FEOL

The FEOL stage is identical to the conventional CMOS process [6]. This stage includes multiple steps that are required to fully form isolated CMOS transistors [41]. First, the type of silicon wafer is selected. Next, shallow trench isolation is performed by adding dielectric material between active regions. Then,

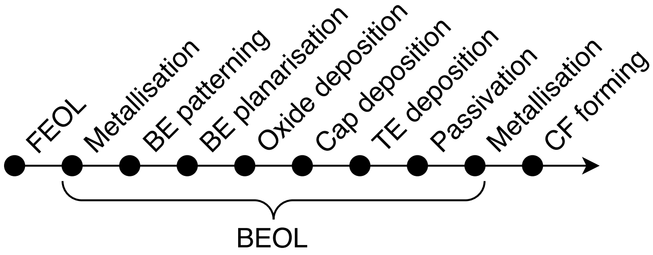


Figure 2.8: RRAM fabrication process. Adapted from [6]

the wells are formed on the silicon wafer, followed by the deposition of the gate oxide and the gate electrodes. Finally, the drain and source of the transistors are formed.

BEOL

The BEOL stage starts identical to the conventional CMOS process by fabricating the bottom metal layers [6]. After the bottom metal layers are fabricated, the fabrication of the RRAM device is started. First, a metal layer is deposited and etched to form the BE. Since the metal has a rough surface, a planarisation step is performed where chemical mechanical polishing is used. In the next step, the oxide is deposited on top of the BE followed by an optional capping layer. Finally, a metal layer is deposited and etched to form the TE. Before continuing with the fabrication of the remaining metal layers, the RRAM device is isolated from its surroundings.

CF forming

After the BEOL stage, the RRAM devices are fully fabricated. However, they cannot be used yet due to a low amount of oxygen vacancies inside the oxide. This is the reason why CF forming is considered the last step of the fabrication process [6].

3. RRAM testing

In this chapter, the focus is placed on RRAM testing. First, the possible defects are presented together with two different methods to model them. Next, the theory behind fault models is explained and the identified fault models are presented. After that, different testing techniques are investigated. Finally, a comparison of the state-of-the-art test solutions is performed.

3.1 Defects

Defects are unintended differences between the fabricated device and its design [42]. Their presence can cause erroneous behaviour or impact the nominal operation of the device. Defects can be categorised based on the step of the fabrication process in which they occur [6], [43]. A summary of all the defects is shown in Table 3.1.

Table 3.1: RRAM defects. Adapted from [27]

FEOL		BEOL	
Historical	Emerging	Interconnect	RRAM
Patterning proximity	Random dopant fluctuations	Opens	Electrode roughness
Line roughness	Anneals	Shorts	Polish variations
Polish variations	Strains	Line roughness	Varying oxygen vacancy density
Dielectric variations	Gate granularity	Irregular shapes	Dimensional variations
		Big bubbles	Material redeposition
		Small particles	Over-forming
			Under-forming

3.1.1 FEOL

FEOL defects include all defects related to the fabrication of the CMOS transistors. In [44], the authors have categorised transistor defects into two categories:

- Historical
  - Patterning proximity effects
  - Line-edge and line-width roughness
  - Polish variations for shallow trench isolation
  - Variations in gate dielectric
- Emerging
  - Random dopant fluctuations
  - Anneals
  - Strains
  - Material granularity

Since FEOL defects are present in all chips containing transistors, the aforementioned defects are commonly known and testing techniques have already been developed to detect these defects.

3.1.2 BEOL

BEOL defects can be categorised into interconnect defects and RRAM defects [6]. Interconnect defects are defects based around the fabrication of the metal layers. These defects are also present in the conventional CMOS process. The interconnect defects are:

- Incomplete wiring
- Incomplete via fills

- Line-edge roughness
- Wire misalignment

RRAM defects are defects based around the fabrication of the RRAM device. These defects can be further divided based on the specific layer in which they occur:

- BE
  - Areal variations
  - Rough surface
- Oxide
  - Non-uniform oxide thickness
  - Variations in oxygen vacancy density
- TE
  - Metal redeposition along the sidewalls

Just like FEOL defects, the interconnect defects are commonly known and testing techniques have already been developed to detect these defects. On the other hand, RRAM defect analysis is in its early stages and the testing techniques necessary to detect these defects are still actively being developed.

3.1.3 CF forming

During CF forming, the over-forming or under-forming defect can occur [6]. The over-forming defect is caused by a too strong CF, resulting in a lower mean resistance of the device. Contrary to the over-forming defect, the under-forming defect is caused by a too weak CF, resulting in a higher mean resistance of the device. In the extreme cases, over-forming can result in a hard dielectric breakdown while under-forming can result in an unformed device.

3.2 Defect models

In order to have a better understanding of the impact of defects on the operation of the device, the defect should be modelled on an electrical level. The most common way of modelling defects is through the Resistive Defect (RD) model [6], shown in Figure 3.1(a). In this model, a defect is modelled as a linear resistance in series or in parallel with the defective device, while the device itself is assumed to be normal. The strength of the defect is represented by the resistance value of the resistor. Though simple, the RD model is not accurate since it fails to capture the non-linear behaviour of the RRAM device [6].

Another way of modelling defects is through the Device-Aware (DA) model [45], shown in Figure 3.1(b). In this model, the technology parameters of the RRAM device are modified such that they mimic the defective device. These modified technology parameters are then fed into an electrical model to obtain the electrical parameters of the device. Finally, real measurements of a defective device can be used to fine-tune the electrical model.

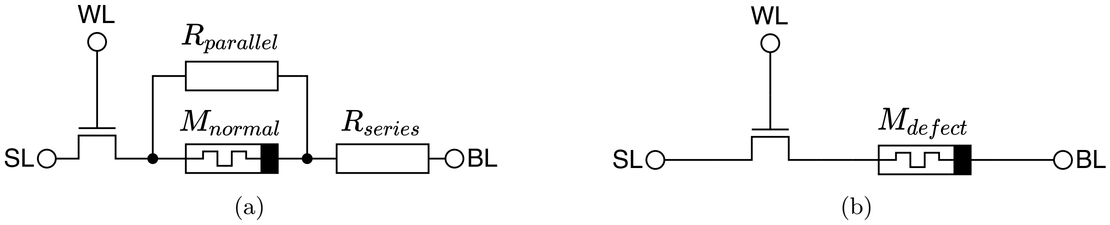


Figure 3.1: Defect models. (a) RD. (b) DA. Adapted from [6]

3.3 Fault models

Fault models are an abstraction of physical defects that capture their effect on the functionality of the device. They were developed in order to make testing more algorithmic rather than empirical. Introducing a level of abstraction solely based on the functionality of the device allows for easier target identification and analysis. Furthermore, it allows for easy comparison between testing techniques based on the amount of faults that they can detect, called the fault coverage.

3.3.1 Classification

Faults can be divided into multiple categories [6], [45], based on a specific aspect:

- Number of operations
  - Static faults - at most one operation is required to sensitise the faults
  - Dynamic faults - more than one operation is required to sensitise the faults
- Number of cells
  - Single-cell faults - the faults involves only one cell
  - Multi-cell coupling faults - the faults involve more than one cell
- Impact
  - Strong faults - the faults cause functional errors
  - Weak faults - the faults cause parametric deviations
- Ease of detection
  - Easy-to-detect faults - the faults can be sensitised and detected by regular memory operations
  - Hard-to-detect faults - the faults require additional detecting techniques

A visual representation of the fault classification can be seen in Figure 3.2.

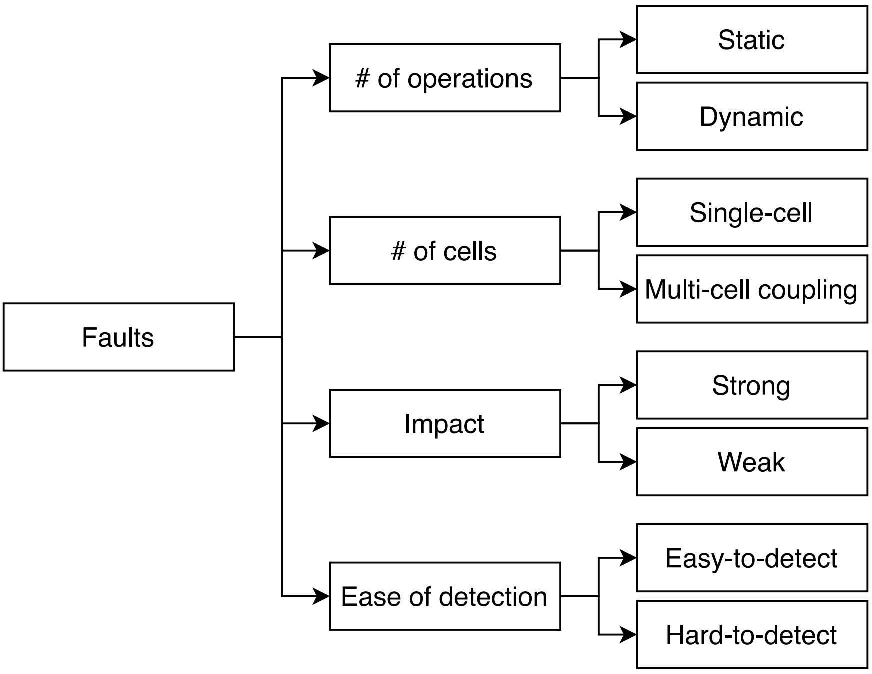


Figure 3.2: Fault classification.

3.3.2 Fault Primitive

In order to represent memory faults in a compact way, Fault Primitives (FPs) were developed. An FP describes the difference between the observed and the expected memory behaviour [42]. It has the following structure $\langle S/F/R \rangle$, where:

- S represents the sequence of operations that sensitises the fault.
- F represents the value of the cell after the sensitisation sequence S.
- R represents the output of the read operation, if the last operation in S was a read operation.

The sensitisation sequence S is of the from $S = x_0 O_1 x_1 ... O_i x_i ... O_n x_n$, where "$O_i$" represents the i-th operation and "x_i" represents the i-th cell value. The operation can be either a read operation ($O_i = r$) or a write operation ($O_i = w$). The cell value can either be a logical 0 ($x_i = 0$) or logical 1 ($x_i = 1$).

The faulty cell F can have five distinctive values $F \in \{0, 1, L, H, U\}$. The values "0" and "1" represent the standard logic 0 (HRS) and logic 1 (LRS), respectively. "L" denotes the extremely low conductance state (deep 0), "H" the extremely high conductance state (deep 1), while "U" denotes the undefined state.

The read output R can have four distinctive values $R \in \{0, 1, ?, -\}$. The values "0" and "1" represent the standard logic 0 and logic 1, respectively. "?" denotes a random output value, while "$-$" denotes no output, meaning that no read operation was performed.

In case of multi-cell coupling faults, a sensitising sequence applied to an aggressor cell influences the state of a victim cell. These faults can also be represented by an FP of the form $\langle S_{a,1}; ...; S_{a,n}; S_v/F/R \rangle$. Here, "$S_{a,i}$" denotes the sensitising sequence of the i-th aggressor cell, while "S_v" denotes the sensitising sequence of the victim cell.

3.3.3 Identified fault models

Throughout history, multiple fault models have been identified for RRAM [6]. These fault models can be divided into two categories: conventional faults and unique faults. Conventional faults are faults which are also present in mainstream memory technologies, while unique faults are faults which are only present in emerging memory technologies.

Conventional faults

The following faults are considered conventional faults that have been identified in RRAM [6]:

- Stuck-at-Fault (SAF): the state of the cell cannot be changed. It is either always in LRS or HRS which is denoted as Stuck-at-1 (SA1) and Stuck-at-0 (SA0), respectively.
- Transition Fault (TF) or Slow Write Fault (SWF): during a set or reset operation, the cell fails to reach its final state.
- State Coupling Fault (CFst): the state of an aggressor cell is coupled to the state of a victim cell. In other words, changing the state of the aggressor cell will also change the state of the victim cell.
- Write Disturbance Fault (WDF): similar to CFst, writing a value to an aggressor cell will also write the same value to the victim cell. This fault can either be sensitised in one cycle (static) or in more than one consecutive cycles (Dynamic Write Disturbance Fault (dWDF)).
- Read Disturb Fault (RDF): performing a read operation on a cell will cause the state of the cell to flip while giving the correct value at the output.
- Incorrect Read Fault (IRF): performing a read operation results in an incorrect value at the output, while the data stored in the cell is correct.

Unique faults

The following faults are considered unique faults that have been identified in RRAM [6]:

- Undefined Write Fault (UWF): performing a write operation will cause the cell to go into the undefined state "U".
- Deep State Fault (Deep): the resistance of the cell is out-of-boundaries. In other words, the cell has a resistance lower than the LRS resistance or higher than the HRS resistance.
- Undefined Read Fault (URF): performing a read operation will result in a random value at the output.
- Intermittent Undefined State Fault (IUSF) [17]: the cell will sporadically (intermittently) change from bipolar switching to complementary switching, causing the cell to end up in the undefined state after a write operation.
- Undefined Coupling Fault (CFud) [16]: similar to CFst, the state of an aggressor cell is coupled to the state of a victim cell. However, the coupling is weak, resulting in the victim cell going into the undefined state instead of the same state as the aggressor cell.

It should be noted that the aforementioned conventional and unique fault models do not represent all the possible faults that can happen in RRAM. Fault models incorporate an extensive set of faults that are most commonly observed and have been reported in literature. If it is desired to test for all possible faults, then using FPs and generating all possible combinations of S, F and R is the correct approach to take. However, the set of FPs increases exponentially with the addition of operations to the sensitisation sequence S [27]. For this reason, using conventional and unique fault models is a good compromise between complexity and fault coverage, since one fault model covers multiple FPs.

3.4 Testing techniques

In order to detect defective RRAM devices, multiple testing techniques have been developed. These testing techniques are based on the aforementioned fault models with the aim of providing the highest fault coverage. The following testing techniques exist:

- March test: applying a sequence of read and write operations on the memory to sensitise a fault and detect it.
- DFT: adding extra hardware to the device to facilitate better, cheaper and/or faster testing.
- Built-In Self-Test (BIST): adding extra hardware to the device so that the device can test itself.
- Built-In Self-Repair (BISR): adding extra hardware to the device so that the device can repair itself.

March test and DFT will be discussed in detail, while BIST and BISR are just provided for completeness.

3.5 March test

March tests are able to test the functionality of the chip without requiring the presence of any additional hardware on the chip. They solely rely on write and read operations to sensitise the fault and detect it in the form of an incorrect value at the output. A March test is composed out of a sequence of march elements [46]. Every march element contains a sequence of operations that is applied to every memory cell. An operation can either be a read "r" or write "w" operation with the operand being either "0" or "1".

After applying all the operations to a given cell, the march element will be applied to the next cell. The addressing order determines the address of the next cell. There are three different addressing orders most commonly used in March tests: up-addressing, down-addressing and arbitrary-addressing [46].

With up-addressing, the address is incremented (denoted with $\Uparrow$), while with down-addressing, the address is decremented (denoted with $\Downarrow$). Arbitrary-addressing is denoted with $\Updownarrow$ and it specifies that the addressing order is irrelevant.

3.5.1 RRAM March tests

Throughout the years, many March tests for RRAM have been developed. Most of the March tests are incremental improvements on a previous March test that introduce coverage for new faults. To the best knowledge of the author, the following March tests are used for RRAM:

- March C– [46]: made for mainstream memory technologies to detect SAF, CFst and TF.

$$\text{March C–: } \{\Updownarrow(w0); \Uparrow(r0,w1); \Uparrow(r1,w0); \Downarrow(r0,w1); \Downarrow(r1,w0); \Uparrow(r0);\}$$

- March–MOM [10]: made specifically for RRAM to detect SAF, Deep, SWF and CFst.

$$\text{March–MOM: } \{\Updownarrow(w0); \Uparrow(r0,w0,w1); \Uparrow(r1); \Downarrow(w1); \Downarrow(r1,w0); \Downarrow(r0);\}$$

- March C* [47]: improves upon March C– to cover RDF.

$$\text{March C*: } \{\Uparrow(r0,w1); \Uparrow(r1,r1,w0); \Downarrow(r0,w1); \Downarrow(r1,w0); \Uparrow(r0);\}$$

- March–1T1R [48]: improves upon March–MOM to cover WDF and dWDF.

$$\text{March–1T1R: } \{\Updownarrow(w0); \Uparrow(r0,w1,r1,(w1)^{a-1}); \Uparrow(r1,(w0)^{b}); \Downarrow(r0,(w1)^{a}); \Downarrow(r1,(w0)^{b});\}$$

- March C*–1T1R [49]: improves upon March C* to cover Deep faults.

$$\text{March C*–1T1R: } \{\Updownarrow(w0); \Uparrow(r0,w1); \Downarrow(r1,r1,w0); \Downarrow(r0,w1); \Uparrow(r1,w1,w0); \Uparrow(r0);\}$$

- March–CMOL [39]: parallel March tests made specifically for the CMOL architecture.

$$\begin{aligned}\text{March–CMOL: } \{ &\Updownarrow(w1); \Uparrow(r\bar{a},wa)_{bc}; \Uparrow(r1,w0)_{mc}; \Uparrow(ra,ra,wa,w\bar{a})_{bc};\\ &\Uparrow(r0,r0,w0,w1)_{mc}; \Downarrow(r1,w0)_{mc}; \Downarrow(r\bar{a},wa)_{bc}; \Downarrow(r0,w1)_{mc};\\ &\Downarrow(ra,w\bar{a})_{bc}; \Uparrow(r\bar{b},wb)_{bc}; \Downarrow(rb,w\bar{b})_{bc}; \Downarrow(r\bar{b},wb)_{bc}; \Uparrow(rb,w\bar{b})_{bc};\}\end{aligned}$$

- March W–1T1R [50]: improves upon March C*–1T1R to cover WDF and dWDF.

$$\begin{aligned}\text{March W–1T1R: } \{ &\Updownarrow(w0); \Uparrow(r0,w1,r1,w1); \Uparrow(r1,w0,r0,w0);\\ &\Downarrow(r0,w1,w1); \Downarrow(r1,r1,w0,w0); \Updownarrow(r0);\}\end{aligned}$$

A comparison between March tests can be seen in Table 3.2. In this table, under the listed faults, "Y" denotes that the fault is covered, "N" denotes that the fault is not covered and "P" denotes that the fault is partially covered. For the test time, "N" denotes the total number of cells in the RRAM array, while "a" and "b" denote the number of consecutive $w1$ and $w0$ operations to detect dWDF, respectively.

In Table 3.2, March tests are compared based on their fault coverage and test time. It can be concluded that March tests have a good coverage for conventional faults. However, they prove inadequate for unique faults.

Table 3.2: March test comparison. Inspired by [6]

		Conventional						Unique						Test Time	
Year	Name	SAF	TF	WDF	IRF	RDF	CFst	UWF	URF	Deep	IUSF	CFud	Coverage	Write	Read
1993	March C– [46]	Y	Y	N	Y	N	Y	N	N	N	N	N	36%	5N	5N
2013	March–MOM [10]	Y	Y	P	N	N	P	N	Y	Y	N	N	36%	5N	4N
2015	March–1T1R [48]	Y	Y	Y	N	N	Y	N	N	P	N	N	36%	(1+2a+2b)N	5N
2015	March C* [47]	Y	Y	N	Y	Y	Y	N	N	N	N	N	45%	4N	6N
2016	March C*–1T1R [49]	Y	Y	N	Y	Y	Y	N	N	Y	N	N	55%	6N	6N
2018	March–CMOL [39]	Y	Y	N	Y	Y	Y	N	N	Y	N	N	55%	N.A.	N.A.
2017	March W–1T1R [50]	Y	Y	Y	Y	Y	Y	N	N	Y	N	N	64%	9N	8N

3.6 Design-For-Testability (DFT)

Throughout the years, multiple DFTs have been developed to improve the testing process of RRAM. Even though all DFTs have the same goal, they use a different method of achieving it. In Figure 3.3, one possible way of categorising DFTs is presented. First, DFTs can be categorised based on their target, which can either be test time reduction or fault coverage improvement. Once the target is determined, the method to achieve this target can be specified. For test time reduction, the method can be shortening the write time, shortening the read time or reducing the number of memory accesses. For fault coverage improvement, the method can be modifying the write circuit, modifying the read circuit or performing other modifications to the architecture. It should be noted that a single DFT can use multiple methods to achieve its target or even have two targets.

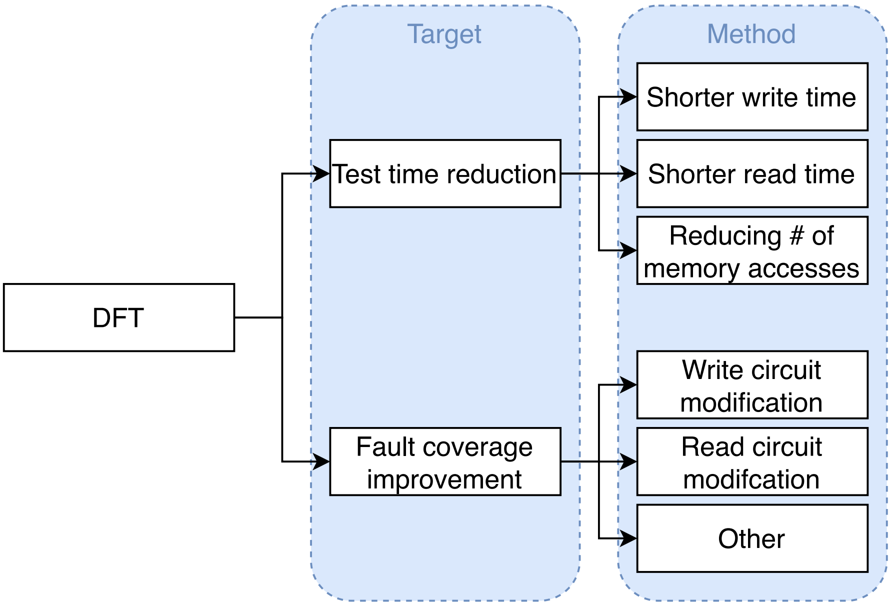


Figure 3.3: DFT categorisation.

In the remainder of this section, state-of-the-art DFTs will be provided and briefly explained.

3.6.1 Divide and Conquer

The Divide and Conquer testing technique is an algorithm which is used to efficiently locate the defective memristor [7]. In Computer Science, this algorithm is also known as the binary search algorithm. Normally, for an RRAM array of N memristors, a maximum of N read operations need to be performed to find the defective memristor. In the case of the Divide and Conquer technique, the maximum number of read operations to find the defective memristor is $2\log_2 N$. This is achieved by using the following algorithm, illustrated in Figure 3.4:

1. Measure the current sum of the entire RRAM array and compare it to the ideal current.
2. If the measured current matches the ideal current, the RRAM array is defect-free. If the measured current deviates from the ideal current, continue with the next step.
3. Split the region into two equal halves and measure the current sum of those halves separately.
4. If the measured current of a halve deviates from the ideal current, repeat step 3 on that halve until all defective memristors are identified.

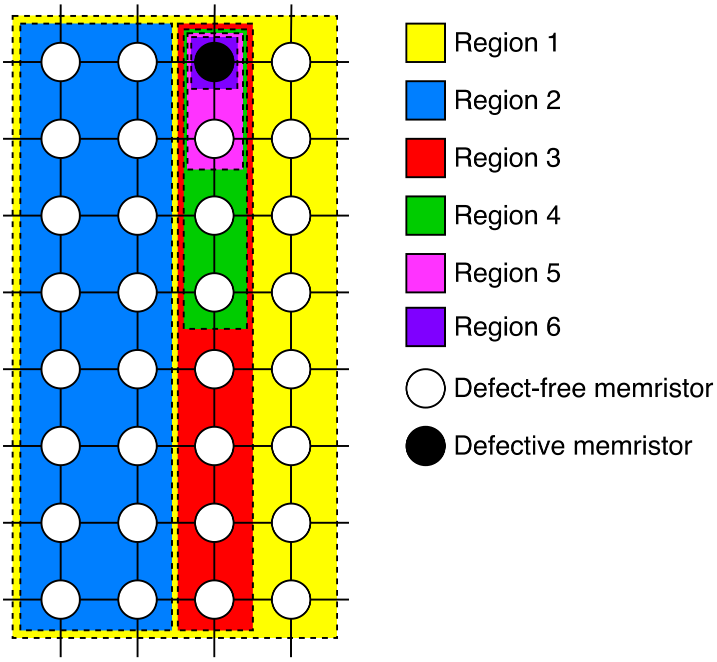


Figure 3.4: Divide and Conquer illustration. Adapted from [7]

Drawbacks

Even though this DFT reduces the testing time, it has the following drawbacks:

- Too simplistic assumptions: the DFT does not take into account device variations. When multiple memristors are read at the same time, their output current is summed. This summation will cause all the individual variations to add up, resulting in a larger current variation at the output.
- The need for multiple reference currents: for every differently sized region, a separate reference current should be provided. Next to that, a different reference current is required when testing for SA0 or SA1 faults.
- Address decoder modification: the address decoder should be modified to introduce the ability of selecting different sized regions during the test phase.
- High current (power) consumption: performing read operations on regions with many memristors will cause a large current to flow through the circuit. For example, when a large region is tested for SA0 and all the memristors are in the LRS.

3.6.2 MAGIC NOR

The MAGIC NOR DFT [8], [9] uses the Memristor-Aided loGIC (MAGIC) NOR operation [51] to check whether all memristors are in the HRS at the same time. Normally, performing a read-0 operation on an entire RRAM array would require N reading operations, where N is the total number of memristors in the RRAM array.

The MAGIC NOR gate consists of multiple input memristors and one output memristor. The input memristors are connected in parallel with the same polarity. The output memristor is connected in series with the input memristors with the opposite polarity, as depicted in Figure 3.5.

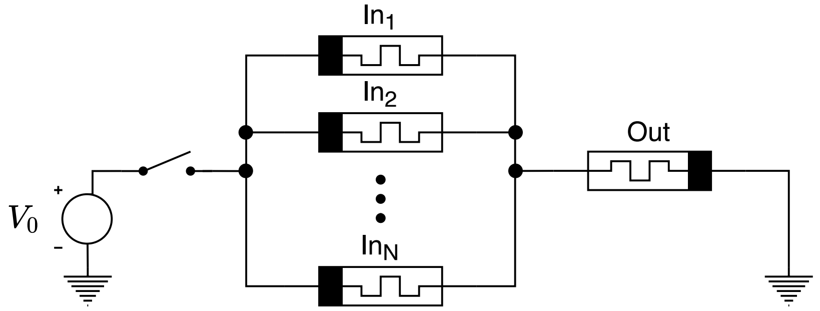


Figure 3.5: Schematic of an N-input MAGIC NOR gate. Adapted from [51]

Performing the logic NOR operation consist of two steps. First, the output memristor is initialised to LRS. Finally, a voltage V_0 is applied to all the input memristors. If all the input memristors are in HRS, most of the voltage will be dropped across the input memristors. This leaves an insufficient voltage across the output resistor to make it change its state and the output resistor stays in LRS. In the case that at least one of the input memristors is in LRS, the voltage drop across the input memristors will be lower. This will result in a higher voltage across the output resistor, causing it to change its state to HRS. The state of the output memristors provides the result of the logic NOR operation.

In order for the MAGIC NOR operation to be successful, the applied voltage V_0 should fulfil three requirements. First, the voltage V_0 should be low enough such that it does not cause a set operation on the input memristors. Secondly, the voltage V_0 should be low enough such that it does not cause a reset operation on the output memristor when the input memristors are all in HRS. Finally, the voltage V_0 should be high enough such that it causes a reset operation on the output memristor when at least one of the input memristors is in LRS. The mentioned requirements can be represented as an inequality presented in Equation 3.1 [8].

$$\frac{|V_{reset}|}{R_{LRS}} \cdot \left[R_{LRS} + \left(\frac{R_{HRS}}{N-1}\right) \parallel R_{LRS}\right] < V_0 < \min\left[|V_{reset}| \cdot \left(1 + \frac{R_{HRS}}{NR_{LRS}}\right), \left(1 + \frac{NR_{LRS}}{R_{HRS}}\right) \cdot |V_{set}|\right] \tag{3.1}$$

The MAGIC NOR DFT is easily implemented by adding an extra row of memristors to the RRAM array and using those memristors as the output of the MAGIC NOR operation. Implementations for both 1R [8] and 1T1R [9] architectures have been developed.

Liu *et al.* in [9] have provided a dedicated March test for the MAGIC NOR DFT called Parallel March, shown in Equation 3.2. In their March test, they use B to denote that the march element is executed in parallel.

$$\text{Parallel March: } \{B(w0); \Uparrow (r0, w1); B(r1); \Downarrow (r1, w0); \Downarrow (r0, w1); B(w1); \Uparrow (r1, w0); \Uparrow (r0); \} \tag{3.2}$$

It should be noted that, in Equation 3.2, the authors use “0” to represent LRS and “1” to represent HRS.

Drawbacks

Even though this DFT reduces the testing time, it has the following drawbacks:

- Unintentional set operation: due to variability, V_0 could be too high and perform a set operation on one or more input memristors.

- Failed reset operation: due to variability, V_0 could be too low to perform a reset operation on the output memristor.

- Only works for specific parameters: for example, using the default parameters from JART VCM v1b [36]: $R_{LRS} = 2$ kΩ, $R_{HRS} = 100$ kΩ, $V_{set} = 0.592$ V, $V_{reset} = -0.943$ V and $N = 64$, results in the inequality 1.36 V $< V_0 <$ 1.35 V which does not hold.

3.6.3 Sneak-path Testing

Sneak-path Testing [10] utilises sneak-paths, explained in Subsection 2.3.4, during testing to detect faults in multiple memory elements at the same time. By performing a read operation on a single memristor and measuring the output current, multiple memristors can be tested due to the presence of sneak-paths. However, sneak-paths should only be allowed during testing since they are undesired during normal operation. To achieve this, the 1T1R architecture is used. During normal operation, only a single memristor is selected by turning on its access transistor. On the other hand, during testing, multiple memristors are selected by turning on their respective access transistors. The memristors that are being tested in parallel are referred to as the Region-of-Detection (RoD).

In Figure 3.6, an example is shown of how RoDs can be tiled in order to test the whole RRAM array. Every square represents a memristor. The dark grey squares represent the memristors that are being addressed while the light grey squares represent the memristors that are part of the RoD. It can be seen that, in order to test the whole RRAM array, only a subset of the memristors need to be accessed. The larger the RoD, the less read operations are necessary to test the whole RRAM array which results in test time improvement. However, the size of the RoD is limited by the precision of the Current-Sense Amplifier (CSA) and variability of the memristors.

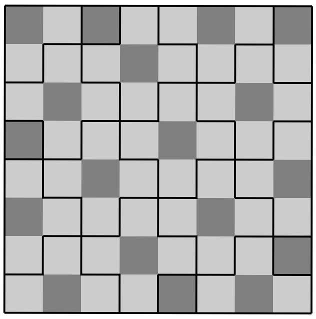


Figure 3.6: Tiling of the RoDs to test the whole RRAM array. Adapted from [10]

In order to perform Sneak-path Testing, two reference currents $I_{idealOFF}$ and $I_{idealON}$ should be added, as can be seen in Figure 3.7. The $I_{idealOFF}$ current is the output current when all the memristor in the RoD are in HRS. Similarly, the $I_{idealON}$ current is the output current when all the memristors in the RoD are in LRS. By setting all the memristors to HRS or LRS and using $I_{idealOFF}$ or $I_{idealON}$ as the reference current during a read operation respectively, defective memristors within the RoD can be detected.

Kannan *et al.* in [10] have provided a dedicated March test for the Sneak-path Testing DFT, shown in Equation 3.3. The symbols "$\uparrow_c$", "$\downarrow_c$" "$\uparrow_d$" are used to denote the new RoD addressing sequences.

$$\{\Updownarrow (w0); \uparrow_c (r0, w0, w1); \uparrow_d (r1); \Downarrow (w1); \downarrow_c (r1, w0); \uparrow_d (r0); \} \tag{3.3}$$

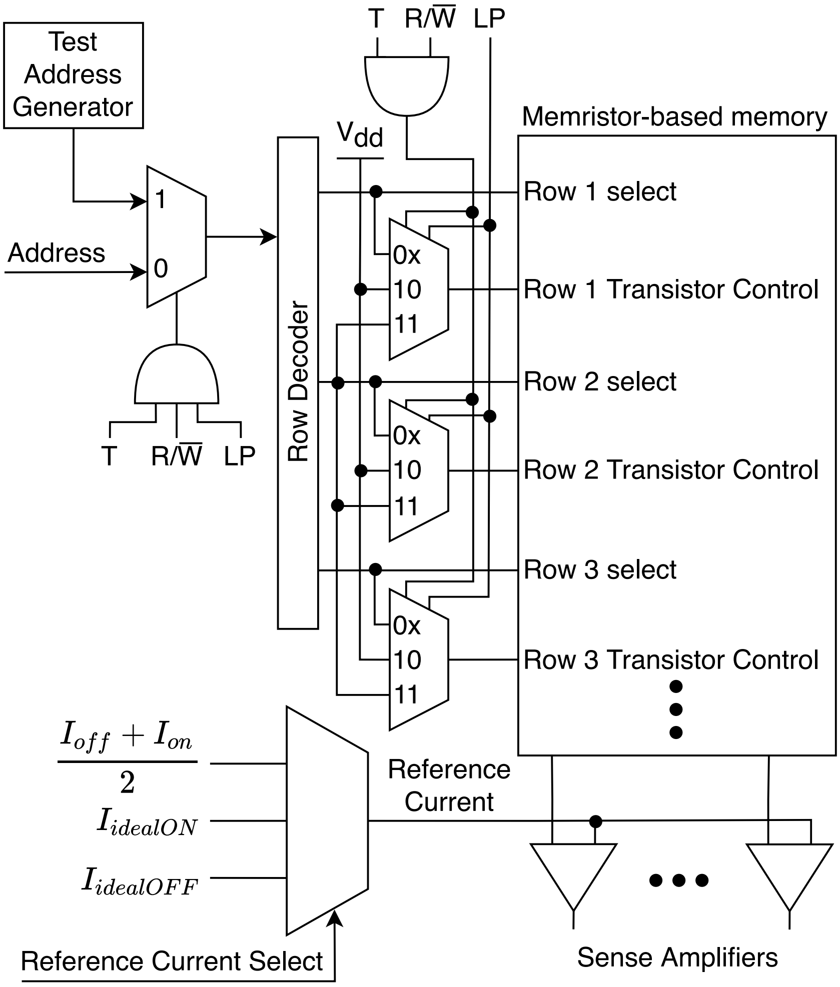


Figure 3.7: Sneak-path Testing DFT. Adapted from [10]

Drawbacks

Even though this DFT reduces the testing time, it has the following drawbacks:

- The need for multiple reference currents: in addition to the $\frac{I_{off}+I_{on}}{2}$ reference current, $I_{idealON}$ and $I_{idealOFF}$ should be provided to the CSA.
- Address decoder modification: the address decoder should be modified such that it can select all cells in the RoD during testing.
- Can only be used for the 1T1R architecture: access transistors are necessary to control the RoD and number of sneak-paths. In the 1R architecture, the number of sneak-paths is unbounded resulting in a large power consumption.

3.6.4 Sneak-path Testing using Voltage Bias

As mentioned in Subsection 3.6.3, the Sneak-path Testing DFT cannot be used for the 1R architecture because transistors are required to limit the RoD. In order to solve this problem, Li *et al.* in [11] developed the Sneak-path Testing using Voltage Bias DFT. This DFT is based on voltage biasing every WL and BL such that the undesired sneak-paths are eliminated while the desired sneak-paths are magnified.

The undesired sneak-paths can be eliminated by biasing all the WLs and BLs to the same voltage. In this way, the current cannot flow through the unselected memristors since there is no voltage drop over the unselected memristors. On the other hand, the desired sneak-paths can be magnified by biasing the WLs and BLs such that a voltage drop is induced over the memristors in the desired sneak-path that amplifies the current through that sneak-path.

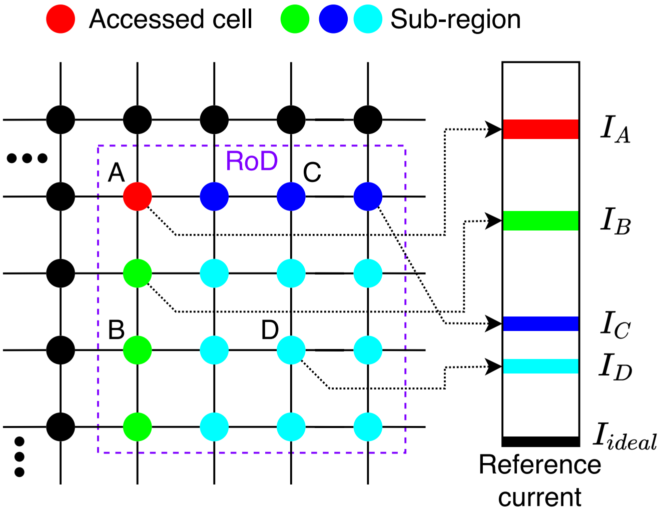


Figure 3.8: Influence of the SA1 fault's location on the output current. Adapted from [11]

The Sneak-path Testing using Voltage Bias DFT uses a 4×4 RoD which is divided into four sub-regions: A, B, C and D. An example of the RoD for the detection of SA1 faults can be seen in Figure 3.8. Sub-region A consists of only one memristor, which is the only memristor being accessed during the read operation. If there is a SA1 fault in sub-region A, the output current will have the largest variance with respect to the ideal output current I_{ideal} and it can be detected by using I_A as the reference current. If there is a SA1 fault in sub-region B, the output current will have the second largest variance since many sneak-paths have to pass through memristors in sub-region B to reach the output. The SA1 fault in sub-region B can be detected by using I_B as the reference current. Finally, the same applies for a SA1 fault in sub-region C and D which can be distinguished from each other and detected by using I_C and I_D as reference current, respectively. In this way, by performing a single read operation, the whole 4×4 RoD can be tested for faults and the location of the defective memristor can be narrowed to its sub-region. If the exact location of the defective memristor is required, the whole procedure can be repeated by using the size of the sub-region as the new RoD.

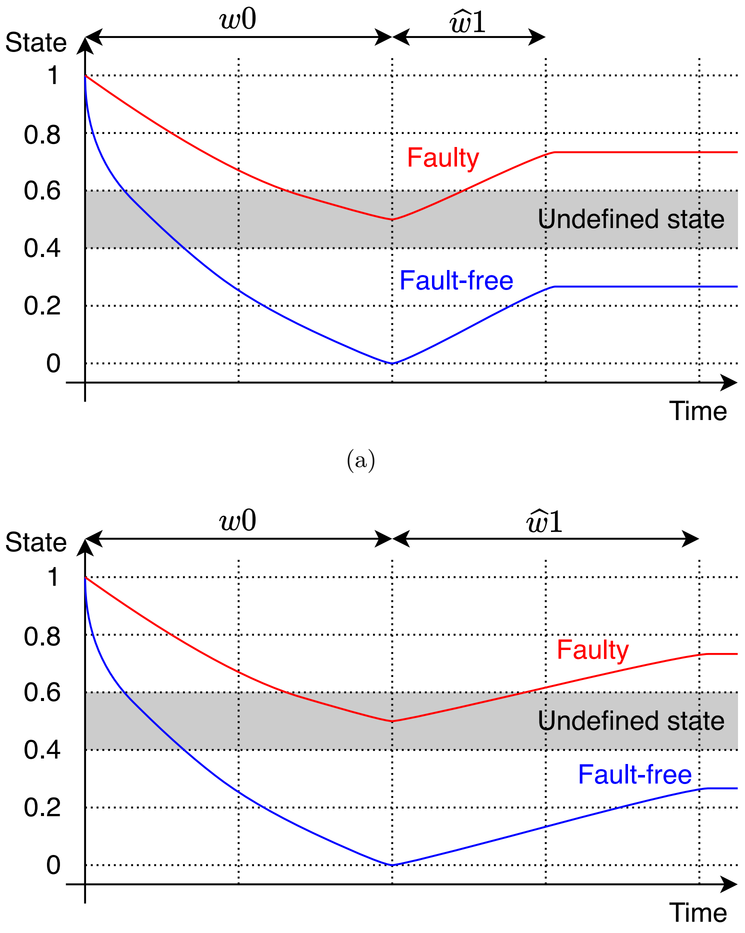
Drawbacks

Even though this DFT reduces the testing time, it has the following drawbacks:

- Different voltages need to be generated at every WL and BL: if not already available, Digital-to-Analogue Converters (DACs) need to be placed at every WL and BL.
- The need for multiple reference currents: one reference current per detection region is required: I_A, I_B, I_C and I_D. Moreover, those four reference currents are not constant but they differ per fault. For one fault, a different set of reference currents needs to be used than for another fault.
- Address decoder modification: the address decoder should be modified such that it can select all cells in the RoD during testing.

3.6.5 Weak-Write - Short Write Time and Low Write Voltage

The Weak-Write DFT [12], [13] is specifically developed for the detection of UWFs. The detection is achieved by introducing a new operation, called weak-write $\widehat{w}$, which causes the faulty memristors to shift from an undefined state to the incorrect logic state while fault-free memristors remain in their correct logic state.



(a)

(b)

Figure 3.9: Effect of the weak-write operation. (a) SWT. (b) LWV. Adapted from [12]

Haron *et al.* in [13] introduce two DFTs for realising the weak-write operation: Short Write Time (SWT) and Low Write Voltage (LWV), shown in Figure 3.10. The SWT DFT realises the weak-write operation by using the same voltage as the regular write operation but for a shorter period of time.

Since the write time is shorter, a fault-free memristor will not have enough time to leave its current state while a faulty memristor will go from the undefined state to the incorrect state, as can be seen in Figure 3.9(a). On the other hand, the LWV DFT realises the weak-write operation by using the same period of time as the regular write operation but with a lower voltage. The effect of using a lower write voltage on a fault-free and faulty memristor can be seen in Figure 3.9(b). Even though Figure 3.9 only shows the weak-write 1 operation, the same principle applies for the weak-write 0 operation. However, it should be noted that Haron *et al.* found that the weak-write 0 operation causes fault-free memristors to go into the undefined state, resulting in possible yield loss.

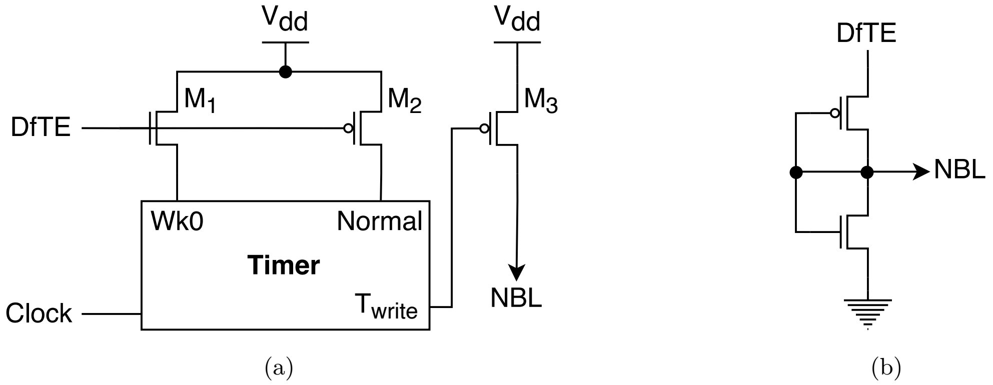


Figure 3.10: Weak-Write DFTs. (a) SWT. (b) LWV. Adapted from [12]

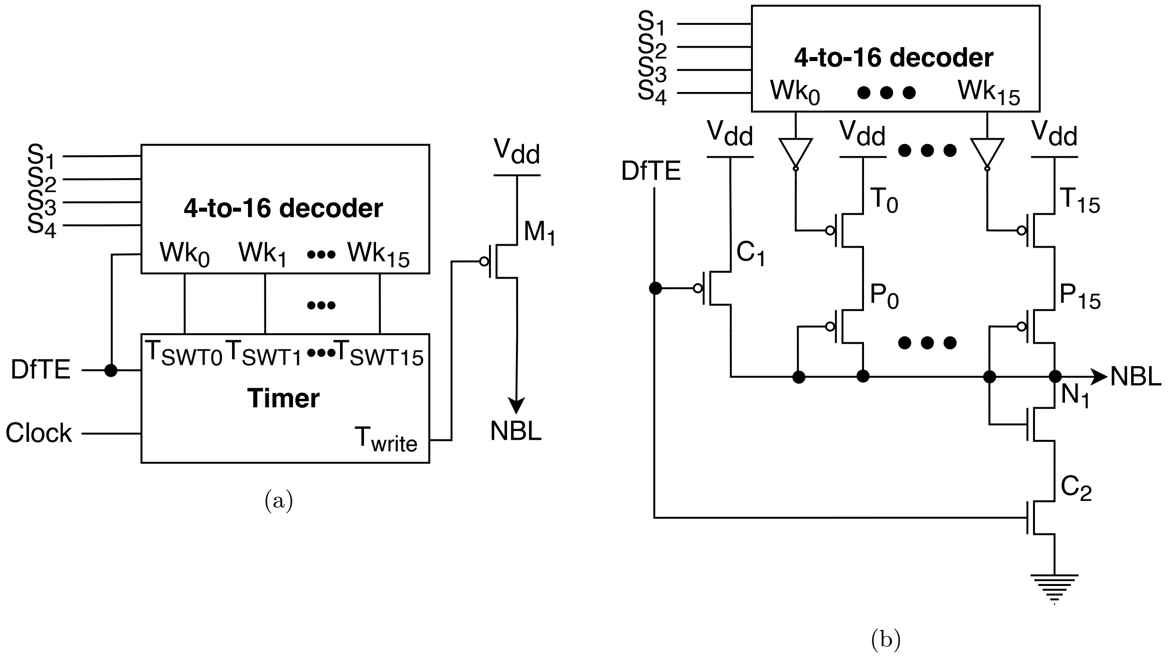


Figure 3.11: Programmable Weak-Write DFTs. (a) PSWT. (b) PLWV. Adapted from [12]

The SWT and LWV DFTs have a major limitation since they can only target a single stress strength, which is determined during the design stage [12]. Moreover, process variations during fabrication are difficult to take into account due to their stochastic nature. Using a fixed stress strength, may result in test escapes and/or yield loss since undesirable under-stressing or over-stressing can occur during testing.

To cope with this limitation, Hamdioui *et al.* in [12] introduce the Programmable Short Write Time (PSWT) and Programmable Low Write Voltage (PLWV) DFTs, shown in Figure 3.11. These are the programmable versions of the SWT and LWV DFTs, which provide the usage of multiple distinct values of write time and write voltage, respectively. In this way, during testing, the desired stress strength can be selected to reduce the number of test escapes and reduce the yield loss.

Drawbacks

Even though these DFTs increase the fault coverage, they have the following drawbacks:

- Yield loss: performing a weak-write 0 operation causes fault-free memristors to end up in the undefined state.
- Restoring the state of the memristor: after a weak-write operation, the memristor is in a weakened logic state. Performing a regular write operation after a weak-write operation could cause the memristor to go into the deep state.
- Variability in short-write time or low-write voltage: variability in the employed short-write time or low-write voltage can cause test escapes and/or yield loss.
- Challenging calibration: for the programmable versions of the DFTs, the determination of the distinct short-write time or low-write voltage values can be challenging.

3.6.6 Fast-Write

Due to the hard (abrupt) set and soft (gradual) reset property of RRAM, performing a set operation takes significantly less time than performing a reset operation [14]. Based on this knowledge, it can be concluded that the write 0 operation dominates the test time of existing March tests [14]. In order to reduce test time, Mozaffari *et al.* in [14] developed the Fast-Write DFT, which introduces the fast-write 0 ($fw0$) operation. A comparison between a regular write operation and the fast-write operation can be seen in Figure 3.12. In this figure, the solid line represents the memristance over time when a regular write 0 operation is performed, while the dashed line represents the memristance over time when a fast-write 0 operation is performed.

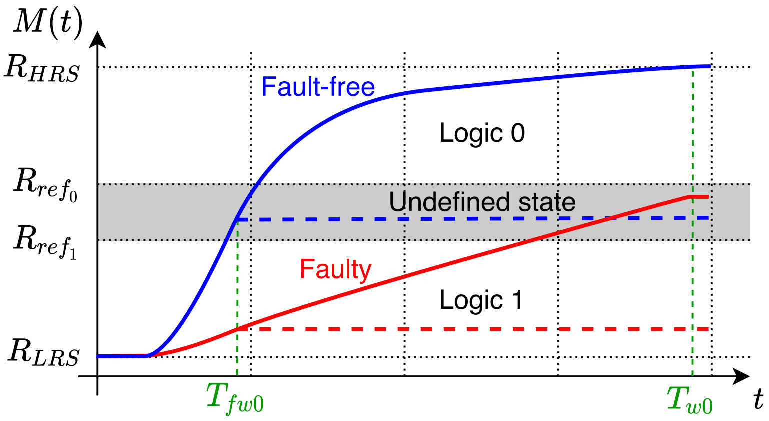


Figure 3.12: Comparison between regular write and fast-write operations. Adapted from [14]

During a reset operation, the resistance increases rapidly in the first instance and, as the resistance continues to increase, the rate of change slows down. Mozaffari *et al.* in [14] have noticed that it is not necessary to perform a full write operation in order to detect a faulty memristor. By prematurely ending the write operation and using a new set of reference currents, a faulty memristor can be distinguished from a fault-free memristor in less time than if a regular write operation was used.

The Fast-Write DFT uses two extra references I_{ref_0} and I_{ref_1} when performing a read 0 and read 1 operation during testing, respectively. These references are located on the boundaries of the undefined state, as shown in Figure 3.12. During normal operation, the standard reference current, in the middle of the undefined state, is used.

The Fast-Write DFT can be seen in Figure 3.13. A test enable signal is used in order to specify if the write timer should use the normal write time or the fast-write time during a write operation.

Furthermore, a multiplexer is added which allows the selection of the desired reference current during a read operation.

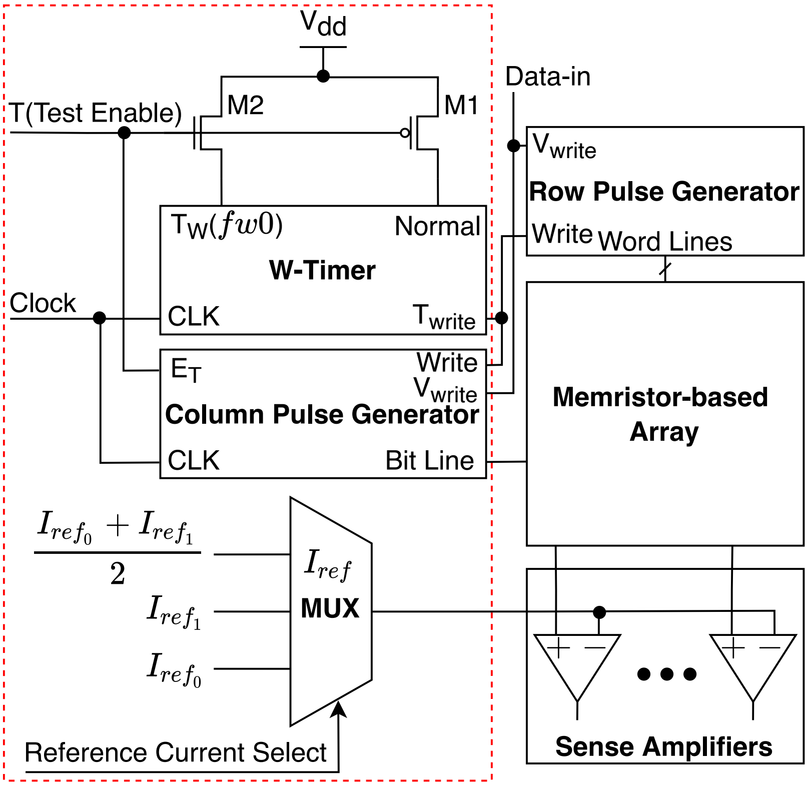


Figure 3.13: Non-programmable Fast-Write DFT. Adapted from [14]

To account for process variations, Mozaffari *et al.* in [14] proposed a programmable version of the Fast-Write DFT, shown in Figure 3.14. The programmable DFT provides the ability to calibrate the write times during testing through the use of the selection signals.

Mozaffari *et al.* in [14] have provided a dedicated March test for the Fast-Write DFT called Fast March Test (FMT), shown in Equation 3.4. In FMT, "$\|$" denotes parallel addressing, "$fw0$" denotes the fast-write 0 operation and "r_{ref_1}" denotes a read operation using I_{ref_1} as the reference current. On top of that, the "$r0$" and "$r1$" operations during testing are performed using I_{ref_0} and I_{ref_1} as reference currents, respectively.

$$\text{FMT: } \{\| (w0); \Uparrow (r0, w1, r1); \Downarrow (r1, fw0, r_{ref_1}); \| (w1); \Downarrow (r1, fw0); \Uparrow (r_{ref_1}, w1); \} \tag{3.4}$$

Drawbacks

Even though this DFT improves fault coverage and reduces the test time, it has the following drawbacks:

- Restoring the state of the memristor: after a fast-write operation, the memristor is in the undefined state. Performing a regular write operation after a fast-write operation could cause the memristor to go into the deep state.
- Variability in fast-write time: variability in the employed fast-write time can cause test escapes and/or yield loss.

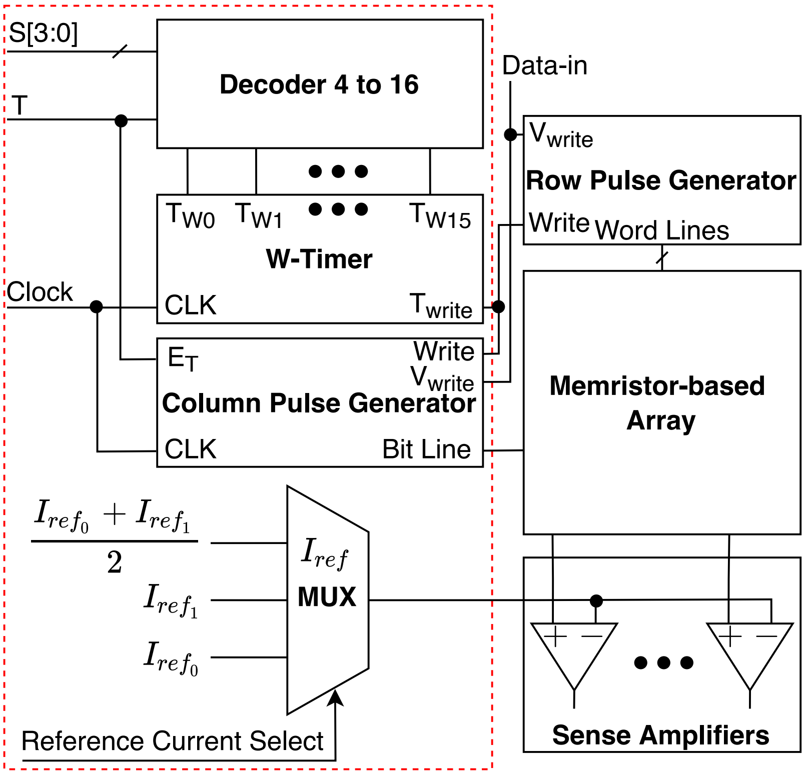


Figure 3.14: Programmable Fast-Write DFT. Adapted from [14]

3.6.7 On-chip Sensor

Copetti *et al.* in [15] take a completely different approach to detecting faults by introducing the On-Chip Sensor DFT, shown in Figure 3.16. The DFT revolves around measuring the voltage at the node between the access transistor and the memristor during a read operation, denoted as MEM. This voltage is then compared to the voltage at the reference node (REF). If the voltage at the MEM node is higher than the voltage at the REF node, the sensor output (SO) will be high. In the other case, the SO will be low. The voltage at the REF node can have two distinct values: Low Resistance Reference (LRR) or High Resistance Reference (HRR). Which one of the two references will be used depends on the VHRR and VLRR signals. In Figure 3.15, a visualisation of the MEM node voltages and REF node voltages can be seen.

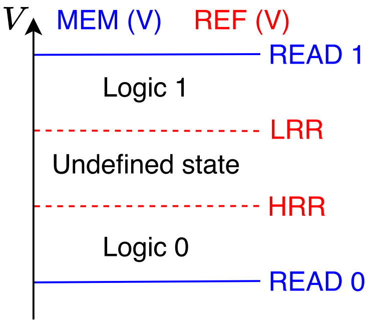


Figure 3.15: Visualisation of the MEM node voltages and the REF node voltages during read operations. Adapted from [15]

During a read 0 operation, the sensor checks if the MEM node voltage is lower than HRR. If this is the case, then the memristor is in the correct state. On the other hand, if the MEM node voltage is higher than HRR, then the memristor is either in the undefined state or logic 1 state. During a read 1 operation, the sensor checks if the MEM node voltage is higher than LRR. If this is the case, then the memristor is in the correct state. On the other hand, if the MEM node voltage is lower than LRR, then the memristor is either in the undefined state or logic 0 state. In order to determine if the memristor is in the undefined state, two read operations should be performed: one with HRR and the other with LRR. It is also possible to use the DFT to detect Deep faults, however, this requires the addition of two extra references: one below the read 0 voltage and one above the read 1 voltage.

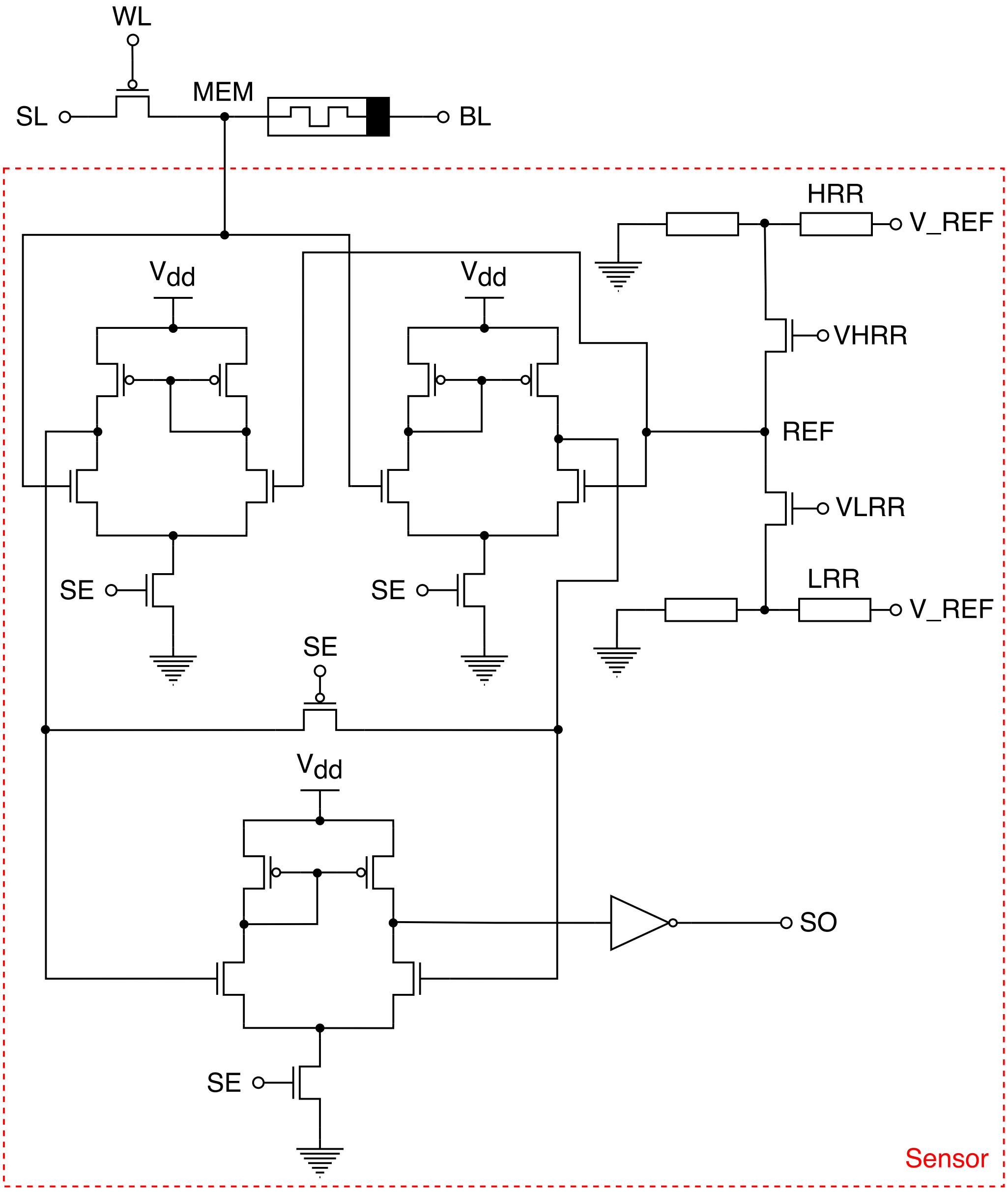


Figure 3.16: On-chip Sensor DFT. Adapted from [15]

Drawbacks

Even though this DFT improves fault coverage, it has the following drawbacks:

- Large area overhead: in this DFT, it is required to use one sensor per memristor.
- Additional (complex) wiring: in a regular RRAM architecture, there is no direct access to the MEM node.
- Reference voltages cannot be compared in parallel: in order to detect UWF, two read operations need to be performed.
- Deep faults not originally covered: two additional reference voltages need to be added to detect Deep faults.

3.6.8 Enhanced March and March RC

Liu *et al.* in [16] introduce two new faults: Undefined Coupling Fault (CFud) and Dynamic Undefined Coupling Fault (dCFud). A representation of the two new faults can be seen in Figure 3.17.

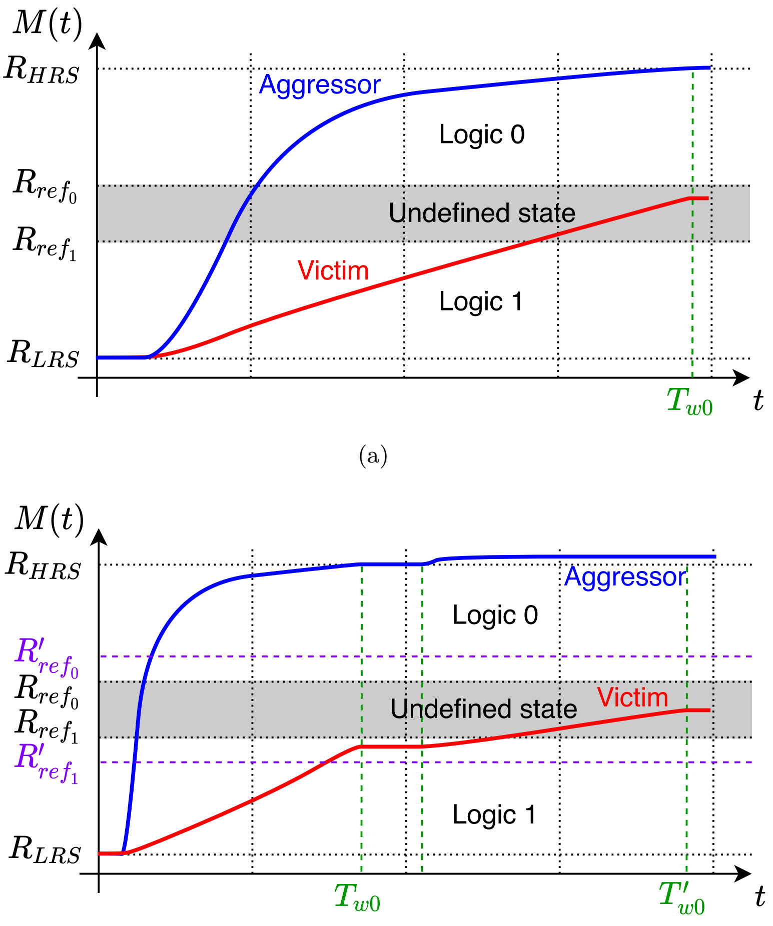


Figure 3.17: Representation of the two new faults. (a) CFud. (b) dCFud. Adapted from [16]

The CFud is similar to the regular CFst, in the sense that there is coupling between adjacent cells (memristors). The difference between CFud and CFst is that, instead of ending up in the same state as the aggressor cell, the victim cell ends up in the undefined state. The dCFud is practically the same as

CFud, with the only difference that it takes more than one write operation to get the victim cell into the undefined state.

To deal with these new faults, Liu *et al.* in [16] introduce two new DFTs: Enhanced March and March RC. The Enhanced March DFT, shown in Figure 3.18, is based on adding two more reference currents, I_{ref_0} and I_{ref_1}, that are located on the boundaries of the undefined state. By performing read operations using these new reference currents, the undefined state can be detected. The Enhanced March DFT comes with its own March test, shown in Equation 3.5.

$$\begin{aligned}\text{Enhanced March: } \{ &\Uparrow (r_{ref_1}, w0, w0); \Uparrow (r0, r_{ref_0}, w1, w1); \\ &\Downarrow (r_{ref_1}, w0, r_{ref_0}, w0); \Downarrow (r_{ref_0}, w1, r_{ref_1}, w1); \}\end{aligned} \tag{3.5}$$

The Enhanced March test can detect both CFud and dCFud. CFud is detected by using the new reference currents, I_{ref_0} and I_{ref_1}, while performing a read operation. On the other hand, dCFud is detected by performing two write 0 and two write 1 operations before performing a read 0 and read 1 operation, respectively. It should be noted that the Enhanced March test can only detect dCFuds that are sensitised within two operations.

In order to reduce the test time of the Enhanced March DFT, Liu *et al.* in [16] developed the March RC DFT, shown in Figure 3.19. Firstly, the test time is reduced by modifying the row and column decoders such that a parallel write operation can be performed. With this parallel write operation, all the memristors can be set or reset at the same time. Finally, the test time is reduced by using new reference currents, I'_{ref_0} and I'_{ref_1}, which are located slightly away from the boundaries of the undefined state. In this way, the two-write-operation sensitised dCFud can be detected by only performing one write operation.

The March RC DFT comes with its own March test, shown in Equation 3.6. The "$\|$" symbol denotes parallel addressing.

$$\begin{aligned}\text{March RC: } \{ &\Uparrow (r'_{ref_1}, w0); \Uparrow (r0, r'_{ref_0}, w1); \| (w1); \\ &\Downarrow (r'_{ref_1}, w0); \| (w0); \Downarrow (r'_{ref_0}, w1); \Updownarrow (r1); \}\end{aligned} \tag{3.6}$$

Drawbacks

Even though these DFTs improve fault coverage and reduce the test time, they have the following drawbacks:

- Large current (power consumption): during the parallel write operation, all the memristors are written to at the same time. The circuit should be designed such that it can handle such a large current and dissipate the heat that is produced during the parallel write operation.
- Bending the definition of the undefined state: placing reference currents slightly away from the boundary of the undefined state in order to catch dynamic faults is theoretically possible but it fails to grasp the reason why the undefined state exists in the first place, which is to ensure that there is enough read margin around the reference current.

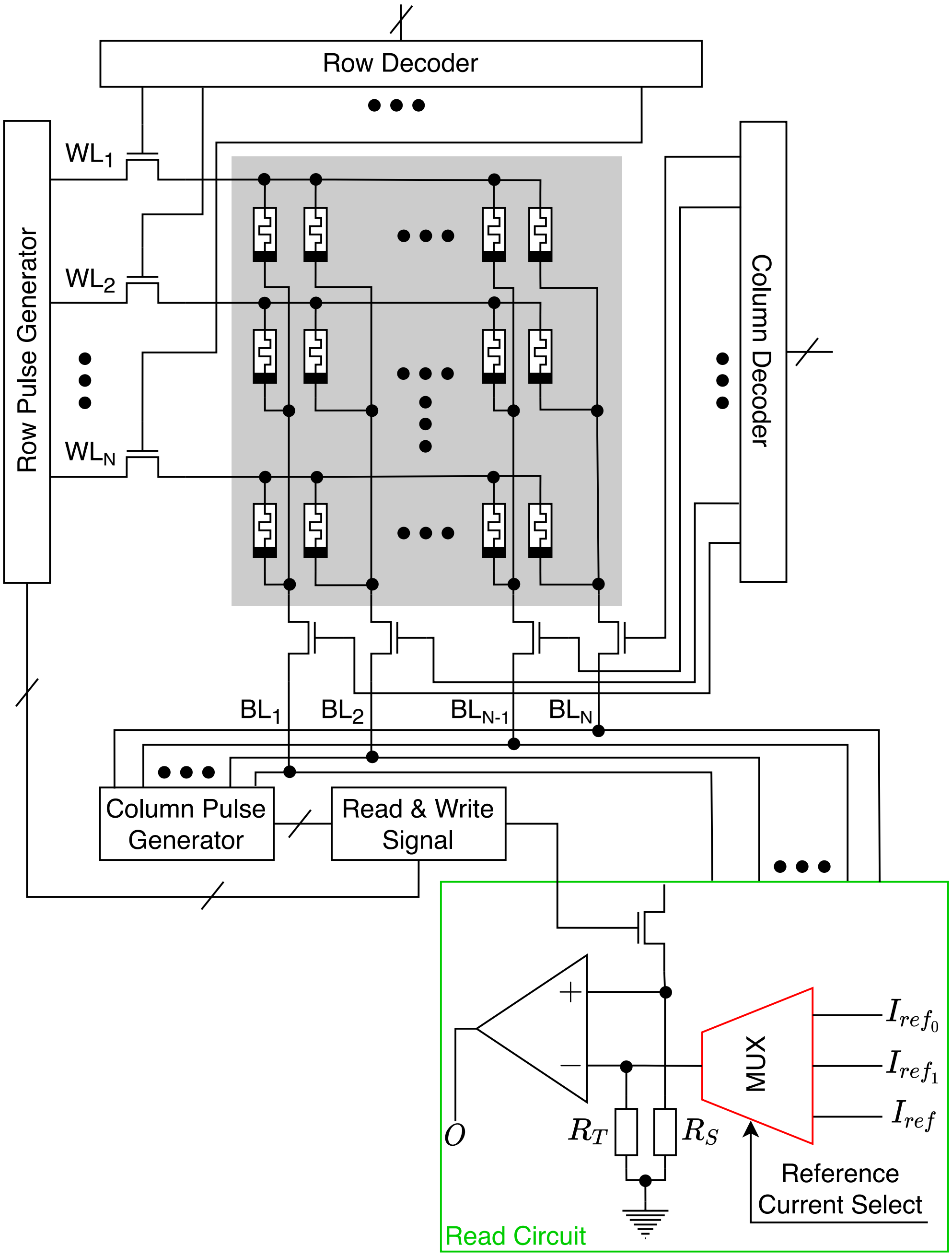


Figure 3.18: Enhanced March DFT. Adapted from [16]

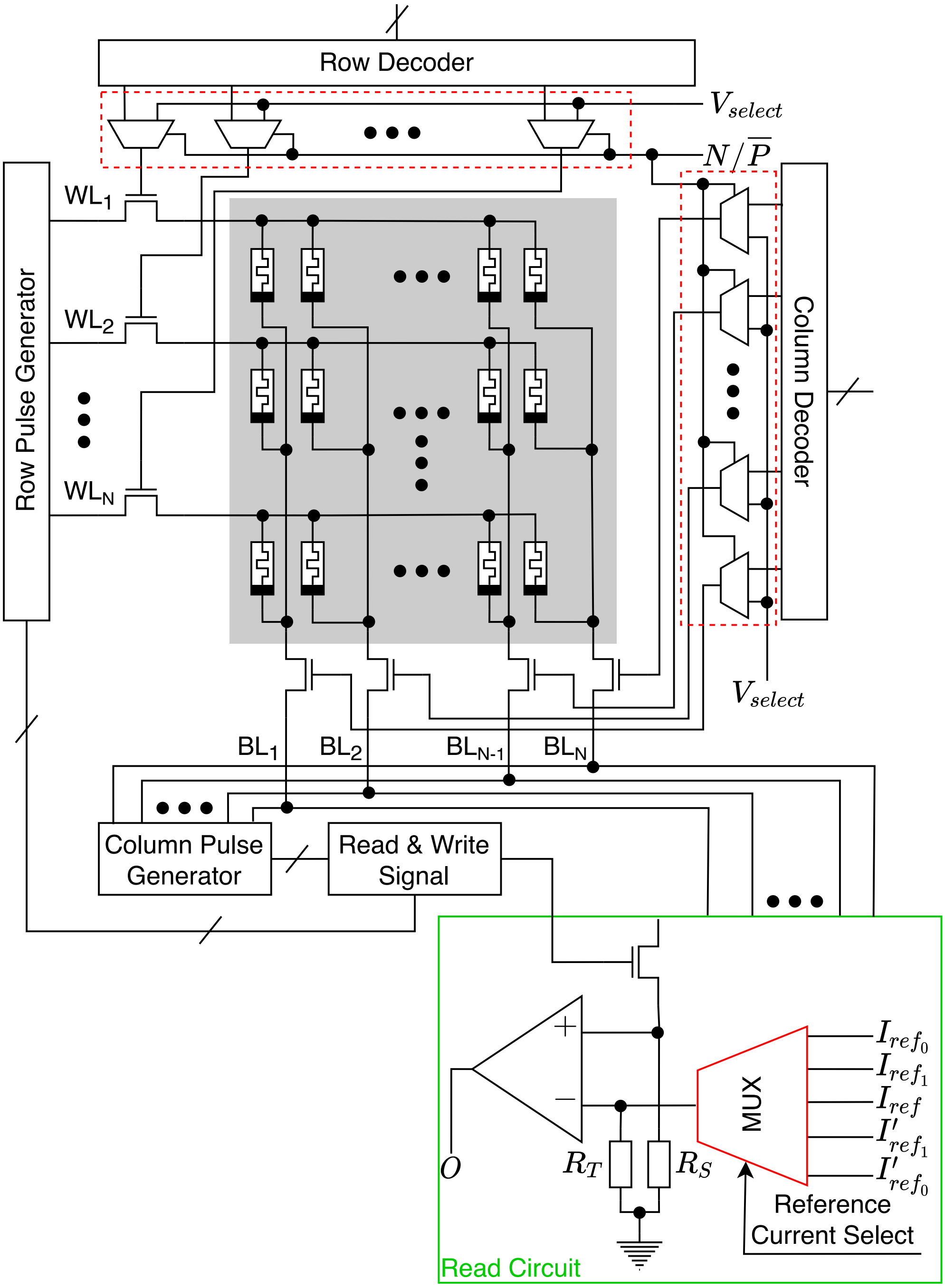


Figure 3.19: March RC DFT. Adapted from [16]

3.7 Comparison

As discussed in Section 3.6, DFTs can be categorised based on their target. In Table 3.3, a comparison between the DFTs based on the target can be seen. It can be concluded that most of the DFTs are either targeting test time reduction or fault coverage improvement. However, there are some DFTs that target both.

Table 3.3: Comparison between DFTs based on the target.

Year	DFT	Test Time reduction	Fault Coverage improvement
2015	MAGIC NOR 1R [8]	Y	N
2017	MAGIC NOR 1T1R [9]	Y	N
2013	Sneak-path Testing [10]	Y	N
2017	Sneak-path Testing using Voltage Bias [11]	Y	N
2012	Weak-Write – SWT [12], [13]	N	Y
2017	Fast-Write [14]	Y	Y
2021	On-chip Sensor [15]	N	Y
2021	Enhanced March [16]	N	Y
2021	March RC [16]	Y	Y

Increasing the fault coverage or improving the test time cannot be achieved without a cost. Most of the time, the cost is manifested in the form of additional area overhead. In Table 3.4, a comparison between DFTs based on area overhead is presented. In this table, N represents the total number of memristors, N_r the number of rows and N_c the number of columns in the RRAM array.

Table 3.4: Comparison between DFTs based on the area overhead.

Year	DFT	Area overhead (# of transistors)
2015	MAGIC NOR 1R [8]	$13(2N_c + N_r)$ [16]
2017	MAGIC NOR 1T1R [9]	$13(N_c + 3N_r)$ [16]
2013	Sneak-path Testing [10]	$28 + 26N_r$
2017	Sneak-path Testing using Voltage Bias [11]	$28 + 26N_r$
2012	Weak-Write – SWT [12], [13]	$24 + 18N_r$ [10]
2017	Fast-Write [14]	$24 + 18N_r + 26$
2021	On-chip Sensor [15]	$20N$
2021	Enhanced March [16]	26
2021	March RC [16]	$13(N_c + N_r + 4)$

Finally, the two most important metrics, fault coverage and test time are considered. A comparison between the March tests and DFTs, based on fault coverage and test time, can be seen in Table 3.5. In this table, under the listed faults, “Y” denotes that the fault is covered, “N” denotes that the fault is not covered and “P” denotes that the fault is partially covered. For the test time, “N” denotes the total number of cells in the RRAM array, “T” the tile size, “x” the ratio between the $w0$ and $w1$ duration, while “a” and “b” denote the number of consecutive $w1$ and $w0$ operations to detect dWDF, respectively. From the comparison, it can be concluded that DFTs offer much better fault coverage for unique faults than March tests. However, there is still no DFT that can detect IUSF, due to its sporadic nature.

Table 3.5: March test and DFT comparison. Inspired by [6]

		Conventional						Unique						Test Time	
Year	Name	SAF	TF	WDF	IRF	RDF	CFst	UWF	URF	Deep	IUSF	CFud	Coverage	Write	Read
1993	March C– [46]	Y	Y	N	Y	N	Y	N	N	N	N	N	36%	5N	5N
2013	March–MOM [10]	Y	Y	P	N	N	P	N	Y	Y	N	N	36%	5N	4N
2015	March–1T1R [48]	Y	Y	Y	N	N	Y	N	N	P	N	N	36%	(1+2a+2b)N	5N
2015	March C* [47]	Y	Y	N	Y	Y	Y	N	N	N	N	N	45%	4N	6N
2016	March C*–1T1R [49]	Y	Y	N	Y	Y	Y	N	N	Y	N	N	55%	6N	6N
2018	March–CMOL [39]	Y	Y	N	Y	Y	Y	N	N	Y	N	N	55%	N.A.	N.A.
2017	March W–1T1R [50]	Y	Y	Y	Y	Y	Y	N	N	Y	N	N	64%	9N	8N
2017	Parallel March [9]	Y	Y	N	Y	Y	Y	N	N	Y	N	N	55%	4(N+1)	5N+N$_r$
2013	Sneak-path Testing [10]	Y	Y	P	N	N	P	N	Y	Y	N	N	36%	7N	5/3N
2012	Weak-Write [12], [13]	N	N	N	Y	N	N	Y	Y	Y	N	N	36%	No March	
2017	Fast-Write [14]	Y	Y	N	N	N	N	Y	Y	Y	N	N	45%	(4+1/T+x/T)N	6N
2021	On-chip Sensor [15]	Y	Y	Y	Y	Y	N	Y	Y	Y*	N	N	73%	No March	
2021	Enhanced March [16]	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	Y	91%	8N	7N
2021	March RC [16]	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	Y	91%	4N+2	6N

4. DFT formulation

In this chapter, new DFTs are formulated. First, two possible ways of detecting each identified fault model are shown. Finally, two DFTs, based on different detection methods, are proposed and their high-level overviews are given.

4.1 Fault detection methods

As the purpose of this thesis is to develop a new DFT that covers all the identified fault models, the target for the DFT is set to fault coverage improvement. Looking at Figure 3.3, it can be concluded that there are three possible methods to consider: write circuit modification, read circuit modification or performing other modifications to the RRAM architecture. Since performing other modifications to the RRAM architecture requires mayor changes and incurs potentially high-overhead, e.g. On-chip Sensor [15], only read circuit modification and write circuit modification are considered.

In the remainder of this section, the application of the two methods is explored per identified fault model.

4.1.1 Conventional faults

Conventional faults are easily detectable using March tests, rendering the DFT useless in terms of fault coverage improvement. However, the impact of the DFT on the test time should not be ignored. Reducing the test time means an overall reduction in testing costs since more chips can be tested during the same time frame.

Stuck-at-Fault (SAF)

SAF can be detected by performing a regular read operation directly after a write operation, as can be seen in Figure 4.1(a). During the write operation, the memristance (purple line) does not change. Since the value of the write operation is known (0 or 1), a read operation using the standard reference (red dotted line) is used to test for the presence of SAF.

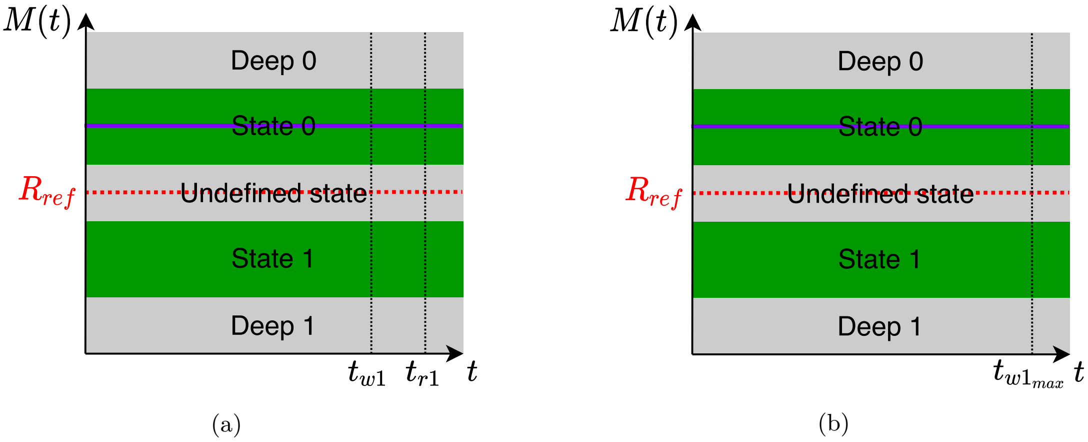
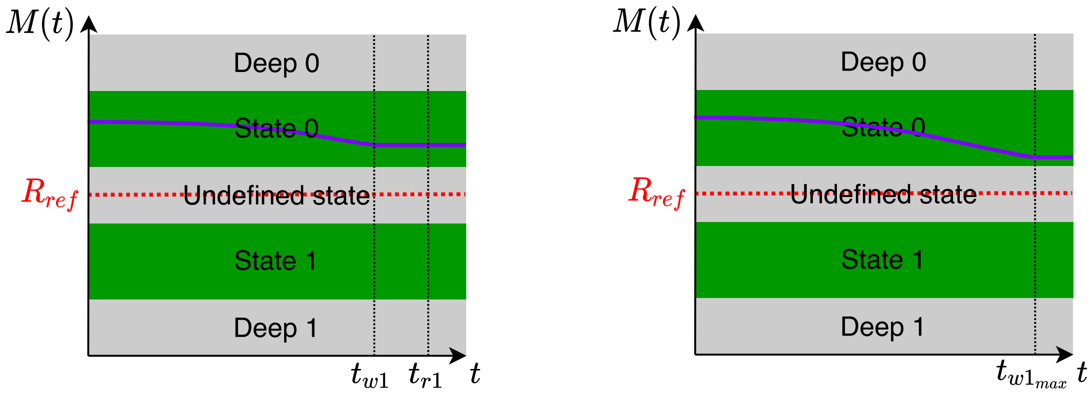


Figure 4.1: Visualisation of SAF detection. (a) During read operation. (b) During write operation.

Another way of testing for the presence of SAF can be during the write operation, as can be seen in Figure 4.1(b). By adding feedback to the write circuit, the write operation is stopped when the memristance reaches its final value instead of using a fixed write time. However, since the memristance does not change with the presence of SAF, the write operation will continue indefinitely. In order to solve this issue, a maximum write time limit can be used. If the memristance does not reach its final value within the maximum write time, the write operation is stopped and SAF is detected.

Transition Fault (TF) or Slow Write Fault (SWF)

TF or SWF is similar to SAF, hence it has the same principle of detection. The difference is that for SAF the memristance value does not change during a write operation, while for TF or SWF the memristance values does change but does not reach the final state. In Figure 4.2(a), the detection during read operation is shown, while in Figure 4.2(b), the detection during a write operation is shown.



(a)

(b)

Figure 4.2: Visualisation of TF or SWF detection. (a) During read operation. (b) During write operation.

State Coupling Fault (CFst) and Write Disturbance Fault (WDF)

CFst and WDF involve two or more cells, hence the detection cannot be established by performing a read operation on the aggressor cell. To the best knowledge of the author, March tests remain the de facto standard for detecting these types of faults during read operations.

During write operations, a technique similar to [52] could be employed. By measuring the difference between the SL current and the BL current, a bridge that causes WDF can be detected. If the SL current and BL current are the same, then the write operation is only affecting the selected memristor. However, if the SL current and BL current are different, then the write operation is also affecting one or more other memristors. In this way, the fault can be detected during the write operation on the aggressor cell without having to perform a read operation on the victim cell.

Read Disturb Fault (RDF)

As the name suggests, RDF can only be sensitised using a read operation. No DFT is required since the fault can easily be detected by performing a second read operation.

Incorrect Read Fault (IRF)

Similar to RDF, IRF can only be sensitised by performing a read operation. However, both sensitisation and detection happen during that single read operation, meaning that there is no need for a second read operation.

4.1.2 Unique faults

As already mentioned in Section 3.7, March tests are inadequate in detecting unique faults. This is the reason why the development of DFTs is crucial for achieving high fault coverage of emerging memory technologies.

Undefined Write Fault (UWF)

UWF is difficult to detect using a standard single-reference read circuit, since the output value will be random. However, if two references are employed that are placed on the boundaries between the logic states and the undefined state, as shown in Figure 4.3(a), the undefined state can easily be detected.

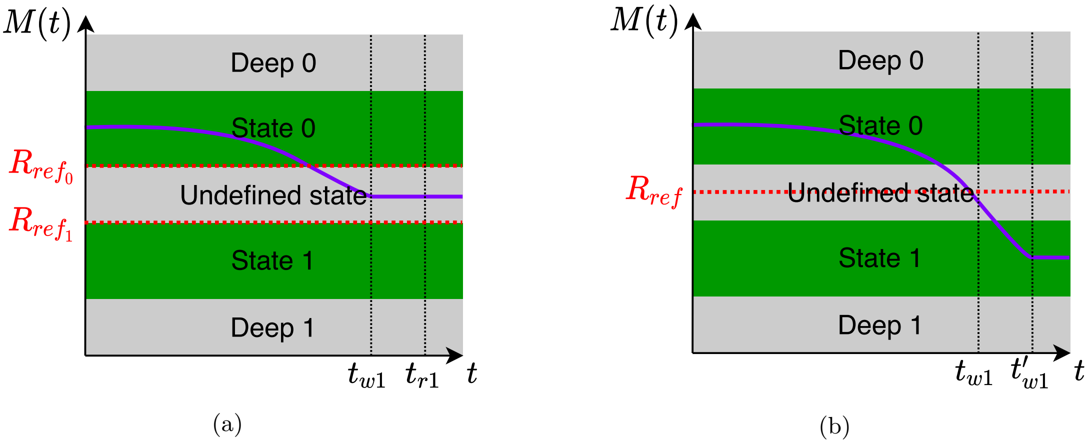


Figure 4.3: Visualisation of UWF detection and prevention. (a) During read operation. (b) During write operation.

On the other hand, UWF can be prevented by adding feedback to the write operation. Instead of using a fixed write time, the write operation is finished once the memristance reaches its final value. The visualisation of this method is shown in Figure 4.3(b). With a regular write circuit, the write operation would be stopped at t_{w1}, resulting in the memristor entering the undefined state. However, with the modified write circuit, the write operation will be sustained until the memristor reaches its final value at t'_{w1}. In this way, UWF is prevented for weaker memristors by extending the write time. It should be noted that, just as with SAF, a maximum write time limit is imposed to prevent the write operation from performing indefinitely.

Unknown Read Fault (URF)

URF is mostly caused by UWF, so it shares the same principle of detection and prevention.

Undefined Coupling Fault (CFud)

Similar to CFst, CFud cannot be detected by performing a read operation on the aggressor cell. On top of that, contrary to CFst, CFud also cannot be detected by performing a read operation on the victim cell. This is caused by the fact that the victim cell ends up in the undefined state, which is not easily detectable using a standard single-reference read circuit. However, if a read circuit with two references located on the boundaries of the undefined state is used, then the detection of CFud is possible through a March test. The principal of detection is the same as for UWF, shown in Figure 4.3(a), with the difference that the write operation is performed on the aggressor cell instead of the victim cell.

During a write operation, a technique similar to [52] could be employed. By measuring the difference between the SL current and the BL current, a bridge that causes CFud can be detected. If the SL current and BL current are the same, then the write operation is only affecting the selected memristor. However, if the SL current and BL current are different, then the write operation is also affecting one or more other memristors. In this way, the fault can be detected during the write operation on the aggressor cell without having to perform a read operation on the victim cell.

Deep State Fault (Deep)

Similar to UWF, Deep is difficult to detect using a standard single-reference read circuit, since the deep 0 and deep 1 state cannot be distinguished from logic state 0 and logic state 1, respectively. However, if two references are employed on the boundaries between the logic states and the deep states, as shown in Figure 4.4(a), the deep states can easily be detected.

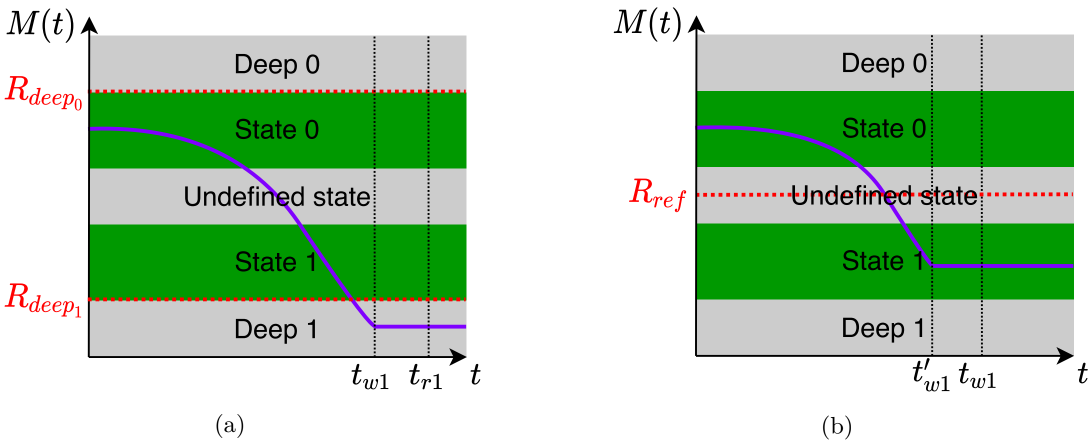


Figure 4.4: Visualisation of Deep detection and prevention. (a) During read operation. (b) During write operation.

Similar to UWF, Deep can also be prevented by adding feedback to the write operation. By making the write time dynamic, the write operation will be stopped as soon as the memristor reaches its final state, preventing the memristor from going into the deep state. A visualisation can be seen in Figure 4.4(b). Here, the dynamic write time t'_{w1} is shorter than the fixed write time t_{w1}, resulting in the prevention of Deep.

Intermittent Undefined State Fault (IUSF)

The difficulty of detecting IUSF lies in its intermittent nature. Solely testing the RRAM devices after fabrication is not sufficient to detect IUSF, resulting in the need for in-field solutions. By using a read circuit with two references on the boundaries of the undefined state, instead of one reference in the middle of the undefined state, the undefined state can be detected with every read operation. A visualisation can be seen in Figure 4.5(a). Since the new references are replacing the old reference, in-field IUSF detection is guaranteed.

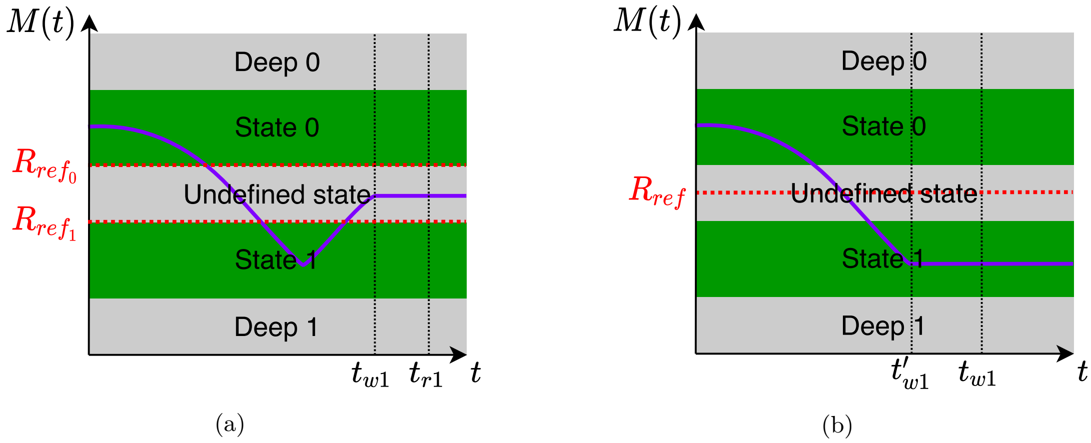


Figure 4.5: Visualisation of IUSF detection and prevention. (a) During read operation. (b) During write operation.

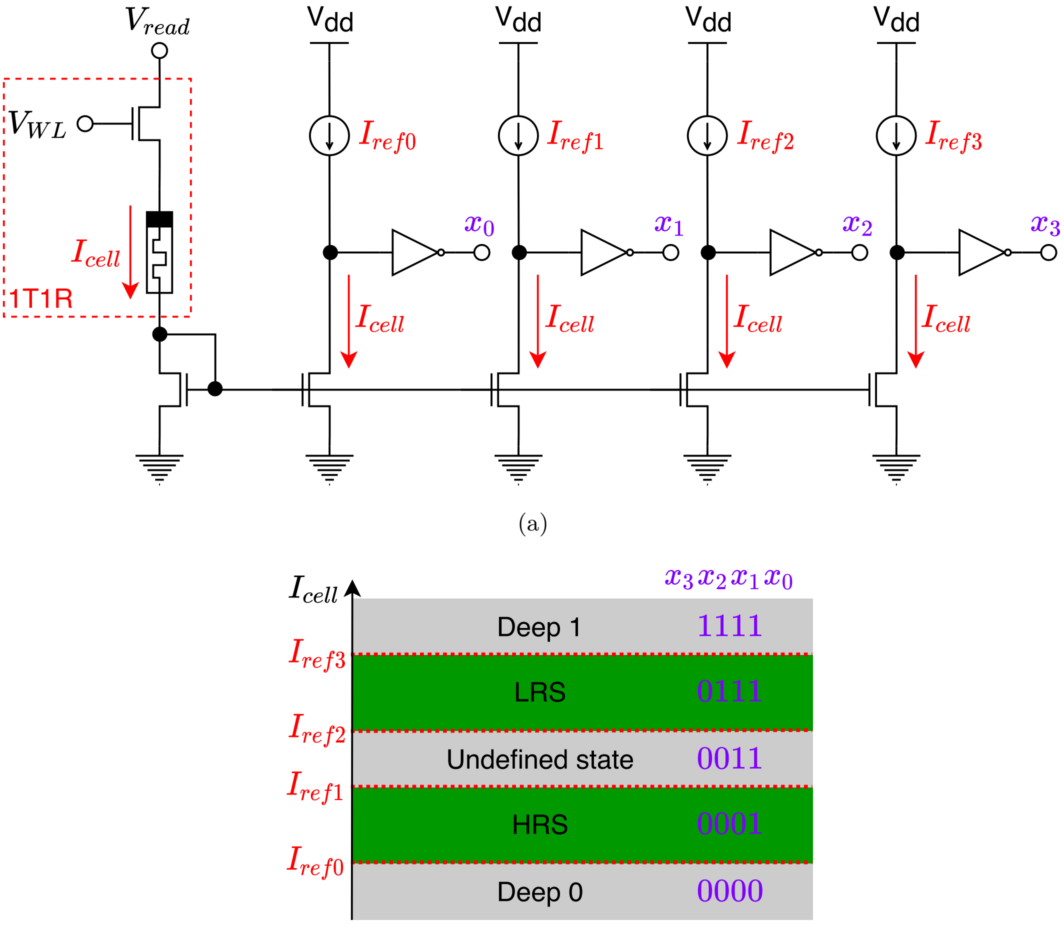
On the other hand, IUSF can be prevented by adding feedback to the write operation, as can be seen in Figure 4.5(b). By making the write operation dynamic, the write operation will be stopped as soon as the memristor reaches its final state and before complementary switching occurs that causes a (partial) reset operation.

4.2 Proposed DFTs

Based on the analysis in Section 4.1, two new DFTs are proposed: Parallel-Reference Read (PRR) and Closed-Loop Write (CLW). The PRR DFT is based on read circuit modification, while the CLW DFT is based on write circuit modification.

4.2.1 Parallel-Reference Read (PRR)

The PRR DFT, shown in Figure 4.6(a), revolves around using multiple references in order to detect all the five memristor states during a read operation. However, what sets it apart from other DFTs that utilise multiple references is that the references are checked in parallel (at the same time). This ensures that only one read operation is necessary in order to determine the state of the memristor. Another benefit is that, since the PRR DFT replaces the standard single-reference read circuit, the PRR DFT incurs less (dead) area overhead than other single-use DFTs. Finally, since the PRR DFT is a replacement rather than an addition to the RRAM architecture, the PRR DFT is capable of in-field fault detection.



(a)

(b)

Figure 4.6: PRR DFT. (a) High-level overview. (b) Reference placement and output encoding.

The working principle of the PRR DFT consists of three simple steps. Firstly, the current going through the memristor I_{cell} during a read operation is duplicated four times through the use of current mirrors. Secondly, a current comparison is performed by taking the difference between the duplicates of the memristor current I_{cell} and the four reference current I_{ref0-3}. Finally, the output of every comparison is combined and presented as a thermometer code $x_3x_2x_1x_0$, whose encoding is shown in Figure 4.6(b). The thermometer code output can further be processed by basic logic gates to obtain the read output in the desired form, e.g. `out` $= x_1$ and `fault` $= x_3 + \overline{x_2}x_1 + \overline{x_0}$.

4.2.2 Closed-Loop Write (CLW)

The CLW DFT is inspired by the Dynamic Write Termination circuit in [53]. Aziza *et al.* in [53] introduce feedback in the write circuit in order to make the write operation shorter, which results in less energy consumption. A similar principle can be used for fault detection, as discussed in Section 4.1, which is the basis of the CLW DFT.

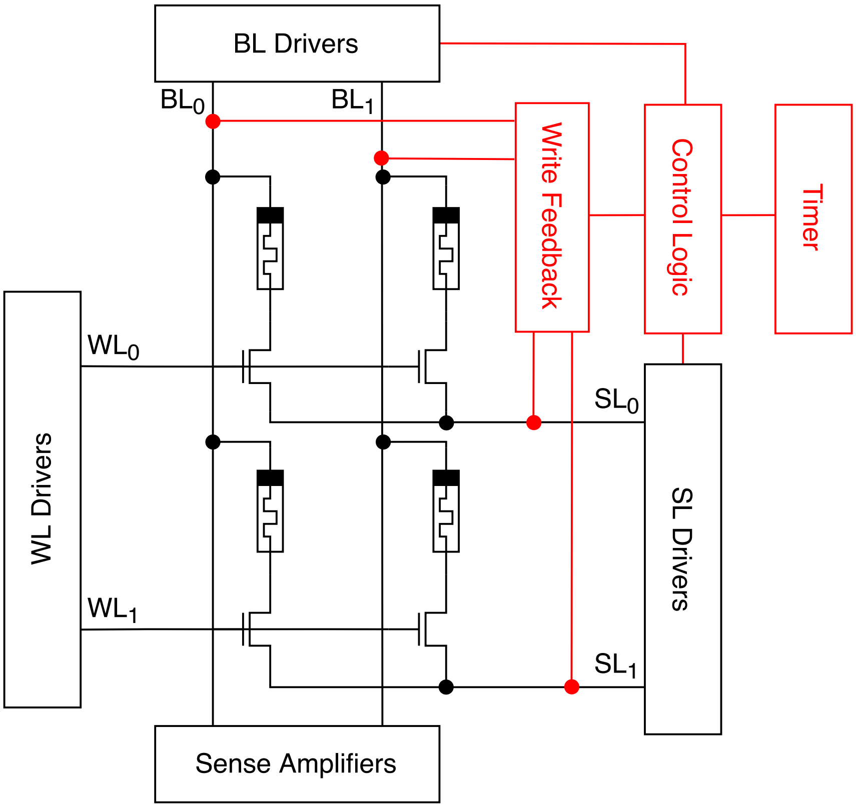


Figure 4.7: CLW DFT high-level overview.

In Figure 4.7, the high-level overview of the CLW DFT can be seen. To every BL and SL, a write feedback circuit is added, shown in Figure 4.8. By using two current comparators, similar to the current comparators used in the PRR DFT, an output signal can be provided when the memristor reaches the desired state (LRS or HRS). As discussed in Section 4.1, the addition of write feedback ensures the prevention of some faults. However, a maximum write limit should be imposed to ensure the detection of all the faults, e.g. SAF. A maximum write limit is imposed by implementing a control logic circuit which starts a timer at the start of every write operation. If the write operation is completed, determined through the $done_{set}$ or $\overline{done_{reset}}$ signals, before the timer reaches the maximum write time, the write operation is labelled as succeeded. On the other hand, if the write operation is still active when the timer reaches the maximum write time, the write operation is stopped and labelled as failed.

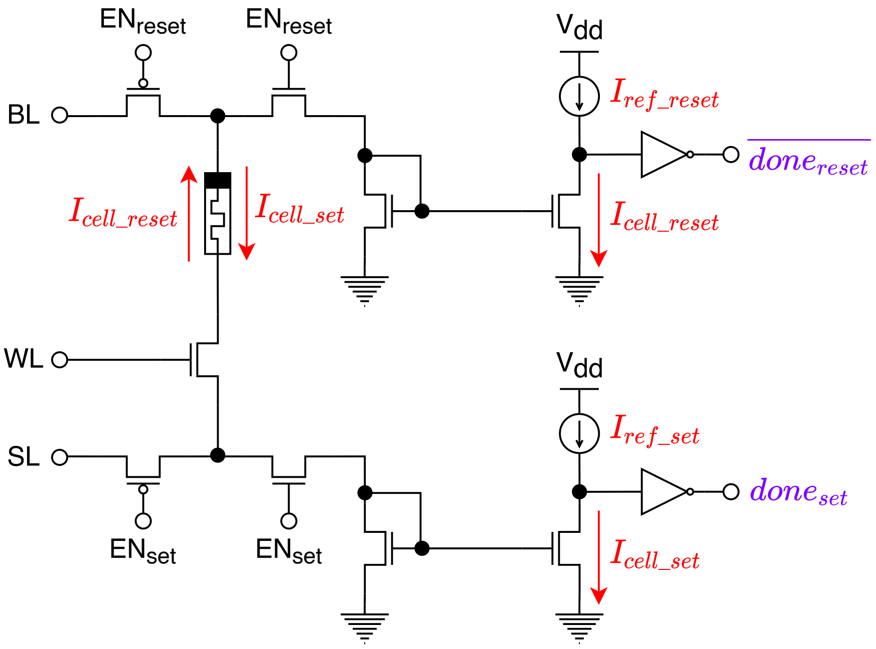


Figure 4.8: CLW DFT write feedback high-level overview.

4.2.3 Final candidate

Even though, in theory, both DFTs are equally effective, in practice, this is not the case. Both DFTs rely on current mirrors in order to create duplicates of the current going through the memristor during a read or a write operation. Ideally, it is desired that the voltage over the memristor is irrespective of its resistance such that the state of the memristor can be determined solely based on the current. However, by introducing a diode-connected transistor in series with the memristor, the voltage over the memristor becomes dependent on the current going through the memristor, i.e. the state of the memristor. This dependence can be compensated for by adjusting the references such that correct boundaries between the memristor states are still ensured. However, there are limitations in the effectiveness of this compensation method.

In a diode-connected transistor, the gate of the transistor is connected to its drain. This results in the drain voltage being equal to the gate voltage, which is, in turn, dependent on the drain current. The higher the drain current, the higher the drain voltage will be. This can be confirmed by looking at Equation 4.1 which shows the relationship between drain current I_d and gate-source voltage V_{gs} for an NMOS transistor in saturation mode. Since the diode-connected transistor is connected in series with the memristor, the voltage over the memristor will become dependent on the current going through the memristor.

$$I_d = \frac{1}{2}\mu_n C_{ox}\frac{W}{L}(V_{gs} - V_{th})^2(1 + \lambda V_{ds}) \tag{4.1}$$

There are two straightforward methods to compensate for this dependence. The first method is to increase the read or write voltage such that desired voltage is present over the memristor while the rest of the voltage is being dropped over the diode-connected transistor. This methods is not ideal since it will increase power consumption and force the usage of transistors that can handle higher voltages. The second method is to increase the width of the transistors such that the influence of the drain current on the drain (gate) voltage is reduced, see Equation 4.1. However, increasing the width of the transistors results in higher capacitance and lower switching speeds.

In the case of the PRR DFT, the diode-connected transistor is only present during a read operation. Since the read operation requires a much lower voltage than a write operation and the current during a read operation is much smaller than during a write operation, the voltage over the diode-connected transistor can decently be compensated by increasing the read voltage and the transistor width.

In the case of the CLW DFT, the diode-connected transistor is present during a write operation for both the set and reset directions. Since the write operation is performed at the V_{dd} voltage and the current during a write operation is much larger than the current during a read operation, compensating the voltage over the diode-connected transistor is challenging. Firstly, increasing the write voltage means that transistors should be used that can handle a higher nominal voltage, which is not desired due to the limitations they impose on scaling. Secondly, since the write current is much larger than the read current, a much larger transistor width is required to compensate for the voltage dependence in the CLW DFT than in the PRR DFT. Aziza *et al.* in [53] face the same issue with their Dynamic Write Termination circuit. Even though Aziza *et al.* state that the nominal set voltage for the employed RRAM cell is 1.2 V, in the simulation and results, a 2.8 V set voltage is used (2.3× the nominal set voltage). Next to that, a 0.13 µm High Voltage CMOS technology with a nominal voltage of 3.3 V is used to be able to handle such a large supply voltage.

Based on the aforementioned limitations, it was decided that the PRR DFT is the best candidate for further development. The lower supply voltage and smaller transistor width makes it more implementable and it facilitates easier future scaling.

5. Test circuit & PRR DFT design

This chapter describes the details of the design. First, the design of the peripheral circuitry (test circuit), which is necessary to be able to validate the PRR DFT, is considered. Next, the PRR DFT's design is developed. Finally, a summary of all the design parameters is given.

5.1 Test circuit

To be able to validate the PRR DFT, a minimal working example of a RRAM chip is designed. In Figure 5.1, the high-level overview of the design can be seen. The whole design revolves around the 1T1R cell, which is used for data storage. The presence of the 1T1R cell is not sufficient since there needs to be a way to change the content of the cell. For this purpose, a write driver is designed, which provides the ability to perform set and reset operations on the 1T1R cell. Next to the write driver, a read driver is developed whose sole purpose is providing the desired voltage to the 1T1R cell during a read operation. Finally, the PRR DFT is designed which replaces the read circuit (CSA) of a standard RRAM chip.

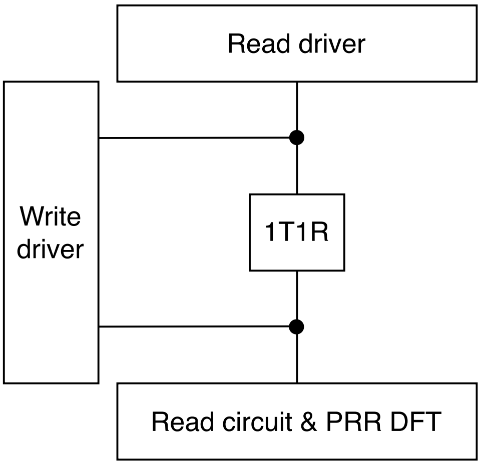


Figure 5.1: Test circuit & PRR DFT high-level overview.

The whole design process is performed in Cadence Virtuoso, while the simulations are performed using Cadence Spectre. Furthermore, for all transistors, the Taiwan Semiconductor Manufacturing Company (TSMC) 40 nm 2.5 V model library is used. In the following subsections, the design of the 1T1R cell, write driver and read driver is elaborated.

5.1.1 1T1R cell

The 1T1R cell, as the name implies, consists of an access transistor in series with a memristor. The memristor and the access transistor are first considered separately before combining them into the 1T1R cell.

Memristor characterisation

Based on the comparison in Table 2.1, the JART VCM v1b [36] model is chosen to describe the behaviour of the memristor. Before integrating the memristor into the 1T1R cell, the characterisation of the memristor is performed. In this way, all the necessary information to design the peripheral circuit around the 1T1R cell is obtained.

In Figure 5.2, the I–V curve of the memristor can be seen. The I–V curve is obtained by applying the voltage V to the TE of the memristor while grounding the BE. The current I represents the current going through the memristor. For the simulation of Figure 5.2, a triangular voltage sweep between -1.5 V and 1.5 V with a period of 6 seconds is used. Through inspection, it can be confirmed that the obtained I–V curve captures the expected memristor behaviour. The two resistive states, LRS and HRS, are represented by the two lines of different slope that cross at the $V = 0$ V point. Moreover, the switching between the resistive states is captured by the two vertical lines, where set happens abruptly at a positive TE voltage, while reset happens gradually at a negative TE voltage.

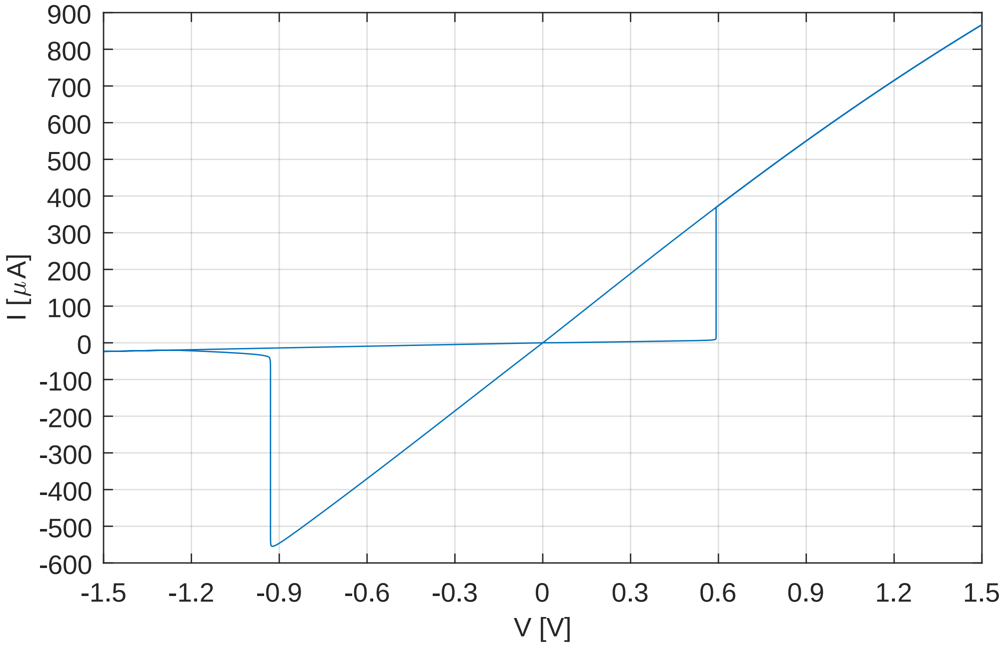


Figure 5.2: Memristor I–V curve.

From the performed simulation, some key parameters can be extracted. These key parameters are shown in Table 5.1. It should be noted that, in this simulation, the memristor is being set and reset to the limiting values specified in the Verilog–A description of JART VCM v1b [36]. In practice, a smaller range within those limits is used to properly simulate the behaviour of the memristor and, at the same time, allowing room for deep states. Moreover, performing a reset operation from one limiting value to the other results in a substantial reset time t_{reset}, which is not desired for fast memory operations. For this reason, R_{HRS} and R_{LRS} in Table 5.1 should be seen as the upper and lower resistance limit, respectively. Furthermore, the set voltage V_{set} and the reset voltage V_{reset} are dependent on the period of the triangular voltage sweep. If a lower period for the simulation is used, the set and reset voltages would be larger in magnitude. Therefore, the magnitude of V_{set} and V_{reset} in Table 5.1 should be seen as the lowest value at which a set and reset occurs, respectively.

Table 5.1: Important memristor parameters.

V_{set}	V_{reset}	R_{LRS}	R_{HRS}	t_{set}@1.5 V	t_{reset}@-1.5 V
592 mV	-930 mV	1.59 kΩ	104.71 kΩ	67 ps	793 µs

LRS & HRS determination

As explained in [36], the chosen LRS and HRS have a substantial impact on the duration of the reset process. The reset process can be dissected into three different phases [36]. In the first phase, called the delay phase, the resistance (state) of the memristor stays almost constant, even though a reset voltage is applied across the memristor. The delay phase is state dependent, meaning that a higher LRS resistance results in a significantly lower delay time. The second phase is the abrupt reset phase whose duration is solely dependent on the applied reset voltage. Finally, the third phase is the gradual reset phase which happens due to the increase in resistance causing lower heating that slows down the reset process. The gradual reset phase is state dependent, meaning that a lower HRS resistance results in a significantly lower reset time.

It can be concluded that, to achieve lower reset time, the LRS and HRS should be as close to each other as possible while still allowing easy distinction between the two states during a read operation. This can be achieved by going for an on/off resistance ratio of around 10, which is still in the standard OxRAM range of 10–100, according to [2].

All things considered, the choice is made to use $R_{LRS} = 4$ kΩ and $R_{HRS} = 45$ kΩ for the LRS and HRS, respectively. Through the use of these values, the reset time is decreased by approximately 100× and room is made for the deep states, while still achieving an adequate on/off resistance ratio of approximately 11. However, since the memristor is an analogue device instead of a binary device, its continuous operating range needs to be divided into multiple states. In Figure 5.3, the chosen boundary values between the states can be seen. The explanation behind the choice of these values is provided in Subsection 5.2.2.

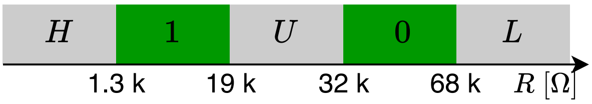


Figure 5.3: Memristor states and their boundary values.

Access transistor type & placement

For the access transistor, there are multiple choices based on the type and placement, which are shown in Figure 5.4. Based on the type, the access transistor can be either Negative-Channel Metal-Oxide-Semiconductor (NMOS) or Positive-Channel Metal-Oxide-Semiconductor (PMOS). The main deciding factor is area consumption since a smaller access transistor will result in a higher memory density. NMOS transistors have a smaller area for the same driving power than PMOS transistors. The reason for this difference is the fact that in NMOS the majority carriers are electrons, while in PMOS the majority carriers are holes [54]. Since the electron mobility is higher than the hole mobility, NMOS transistor consume a lower area for the same driving power than PMOS transistors. This is the reason why NMOS is chosen for the access transistor type.

Based on the placement, the access transistor can be either on the SL or the BL. When an NMOS is placed on the SL, as shown in Figure 5.4(a), the access transistor is favouring the set operation since its source is grounded during the set operation. On the other hand, when the NMOS is placed on the BL, as shown in Figure 5.4(c), the access transistor is favouring the reset operation. Since the reset operation is the slowest operation of the two, placing the NMOS on the BL is the preferred option.

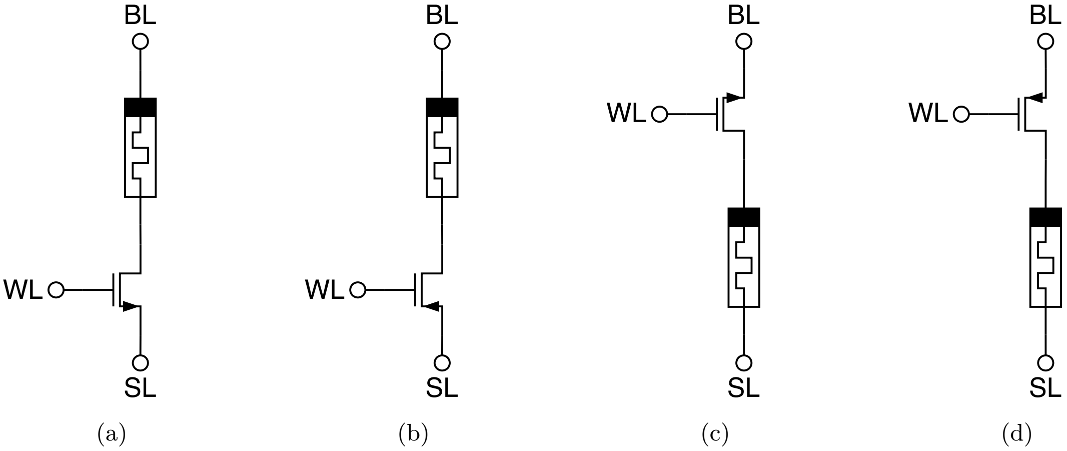


Figure 5.4: Access transistor type and placement. (a) NMOS at SL. (b) PMOS at SL. (c) NMOS at BL. (d) PMOS at BL.

NMOS characterisation

Before integrating the NMOS transistor into the 1T1R cell, the characterisation of the NMOS transistor from the TSMC 40 nm 2.5 V library is performed. An NMOS transistor has three operating regions [54]:

- Cut-off region:

$$I_d \approx 0 \quad \text{for } [V_{gs} \leq V_{th}] \tag{5.1}$$

- Triode, ohmic or linear region:

$$I_d = \mu_n C_{ox} \frac{W}{L} (V_{gs} - V_{th} - \frac{V_{ds}}{2}) V_{ds} \quad \text{for } [V_{gs} - V_{th} \geq V_{ds} \geq 0] \tag{5.2}$$

- Saturation, active or pinch-off region:

$$I_d = \frac{1}{2} \mu_n C_{ox} \frac{W}{L} (V_{gs} - V_{th})^2 (1 + \lambda V_{ds}) \quad \text{for } [V_{ds} \geq V_{gs} - V_{th} \geq 0] \tag{5.3}$$

where I_d is the drain current, V_{gs} is the gate-source voltage, V_{ds} is the drain-source voltage, V_{th} is the threshold voltage, μ_n is the electron mobility, C_{ox} is the gate oxide capacitance per unit area, λ is the Channel Length Modulation (CLM) parameter, W is the gate width and L is the gate length.

Since TSMC does not publicly disclose their transistor parameters, I–V curves should be obtained through simulations to characterise the NMOS transistor. In Figure 5.5, the dependence of the drain current I_d on the gate-source voltage V_{gs} can be seen for the minimum transistor size $L = 270$ nm and $W = 400$ nm. From this I–V curve, an important parameter can be extracted which is that $V_{th} \approx 500$ mV.

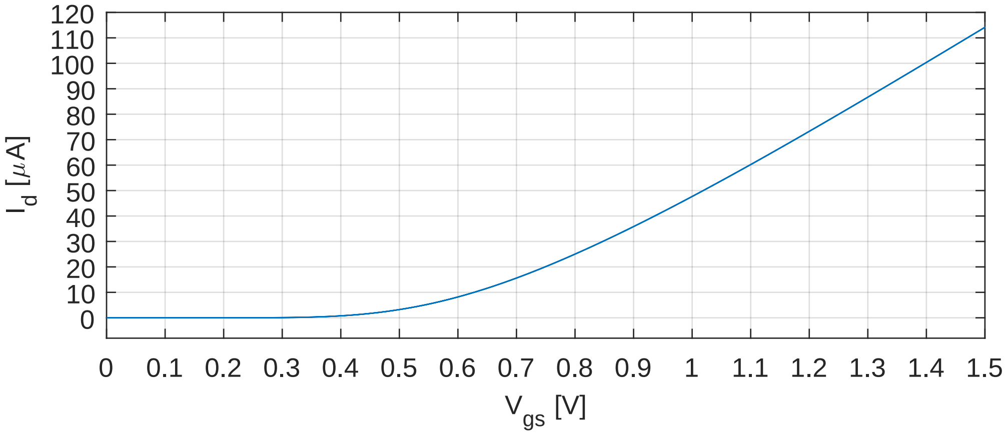


Figure 5.5: NMOS I_d vs V_{gs} curve for $V_{ds} = 1.5$ V.

Another important relationship is between the drain current I_d and the drain-source voltage V_{ds}, which is shown in Figure 5.6. In this figure, curves can be seen for different gate-source voltages V_{gs} when the minimum transistor size, $L = 270$ nm and $W = 400$ nm, is used. Finally, the relationship between the drain current I_d and the drain-source voltage V_{ds} for different transistor widths is shown in Figure 5.7. In this figure, curves can be seen for different transistor widths when the minimum length is used $L = 270$ nm, while $V_{gs} = 1.5$ V is used for the gate-source voltage. These I–V curves are referenced throughout the design process to determine the approximate sizing of the transistors before fine-tuning through simulation is performed.

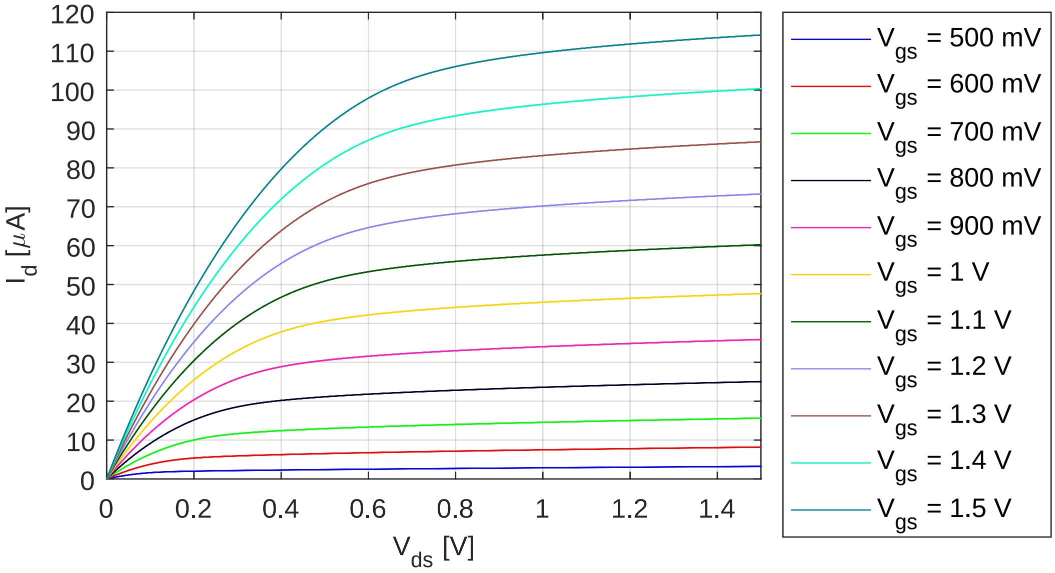


Figure 5.6: NMOS I_d vs V_{ds} curve for different V_{gs}.

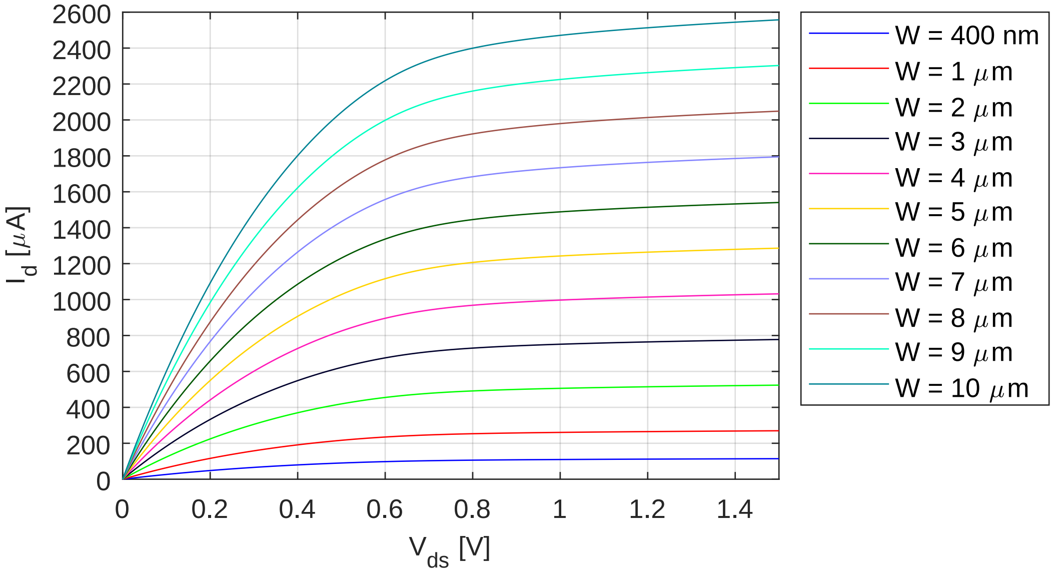


Figure 5.7: NMOS I_d vs V_{ds} curve for different transistor widths.

Access transistor sizing

The sizing of the access transistor has a large effect on the performance of the 1T1R cell. Most importantly, it impacts the set compliance current, write time and memory density. Regarding the write time, using a wider transistor results in faster set and reset operations, which is a valuable property for memories. However, using a wider transistor also results in lower memory density, which is not desired. A trade-off needs to be made between memory density and write time.

The TSMC 40 nm 2.5 V model library offers a minimum transistor length of 270 nm and a minimum transistor width of 400 nm. For the access transistor, the length is kept at the minimum value since there is no benefit in increasing the length. On the other hand, increasing the width of the transistor can be beneficial due to its impact on the reset time. Through simulation, it is determined that increasing the access transistor width to 500 nm allows for sub–10 µs reset operations while still limiting additional area consumption.

5.1.2 Write driver

The write driver is responsible for performing the set and reset operation on the memristor. To reiterate, a set operation is performed by providing V_{dd} to the BL while grounding the SL. On the other hand, a reset operation is performed by providing V_{dd} to the SL while grounding the BL. This means that the write driver should be capable of providing V_{dd} to both the BL and SL, as well as grounding the BL and SL. Moreover, the write driver should also be capable of disconnecting from the BL and SL when a read operation is performed. In this way, no additional current paths are introduced that could interfere with the read operation. The mentioned requirements can easily be fulfilled by using two tri-state buffers, one on the BL and one on the SL, as can be seen in Figure 5.8.

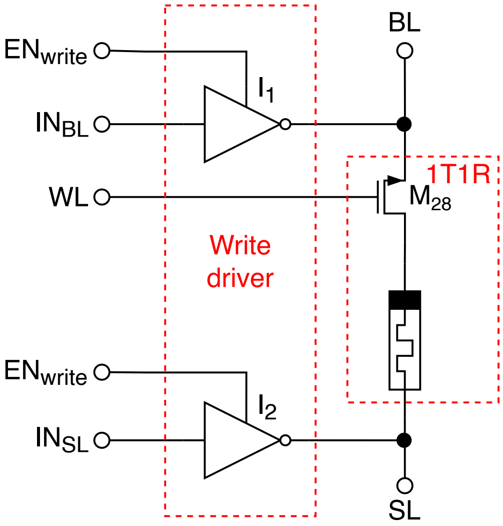


Figure 5.8: Write driver.

Tri-state buffer

The implementation of a tri-state buffer can be seen in Figure 5.9. A tri-state buffer has a similar working principle as an inverter, in the sense that the signal at the output `OUT` is the inverted version of the signal on the input `IN`. The difference lies in the fact that the tri-state buffer has an additional enable input `EN`. If `EN` is high, then the tri-state buffer functions just as a regular inverter. However, if `EN` is low, then the output is in High-Z mode, i.e. disconnected.

In terms of the low-level implementation, the transistors `I1-2.M1` and `I1-2.M2` form a basic inverter which is used to create the inverted version of the `EN` signal. The `EN` and $\overline{\texttt{EN}}$ signals are provided to the transistors `I1-2.M5` and `I1-2.M4`, respectively. The transistors `I1-2.M4` and `I1-2.M5` are used to disconnect the output `OUT`, resulting in the High-Z mode. Finally, the transistors `I1-2.M3` and `I1-2.M6` form a basic inverter which provides the inverted version of the input signal `IN` at the output `OUT`.

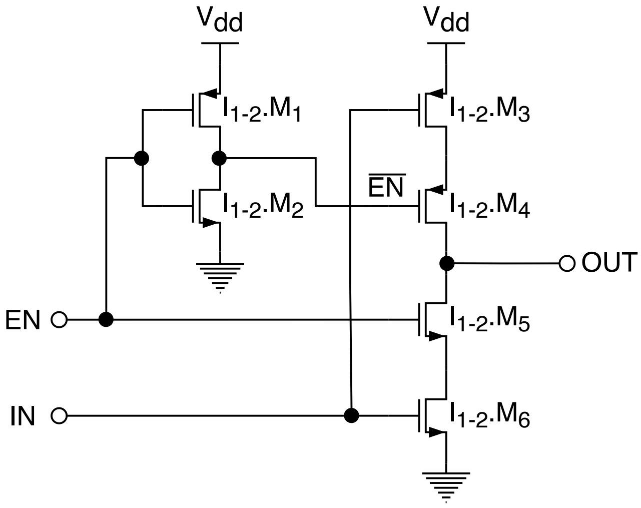


Figure 5.9: Tri-state buffer.

Tri-state buffer sizing

The inverter, consisting of transistors `I1-2.M1` and `I1-2.M2`, is only used to drive the gate of the transistor `I1-2.M4`. Therefore, it is unnecessary to increase the length or width of transistor `I1-2.M2` from the minimum value of 270 nm and 400 nm, respectively. In this way, the area consumption of the inverter is kept at a minimum. On the other hand, the width of the transistor `I1-2.M1` needs to be increased in order to compensate for the lower driving power of the PMOS transistor compared to the NMOS transistor. Through simulation, it is found that making the PMOS width three times larger than the NMOS width results in equal driving powers and a midpoint voltage of $V_M = \frac{V_{dd}}{2}$ for the inverter.

Transistors `I1-2.M1`, `I1-2.M2`, `I1-2.M3` and `I1-2.M4` are used to drive the memristor during a write operation. For this purpose, the width of the transistors should be properly sized so that the required write current can be achieved. In the worst case, during a reset operation, the memristor is in LRS with a resistance of 4 kΩ while a write voltage of $V_{dd} = 1.5$ V is applied across the memristor. This results in a write current of 375 µA. Through inspection of Figure 5.7 and additional fine-tuning, it is found that using 10 µm for the width of the NMOS transistors `I1-2.M5` and `I1-2.M6`, while using 30 µm for the width of the PMOS transistors `I1-2.M3` and `I1-2.M4`, results in the required driving power while keeping the area consumption limited.

5.1.3 Read driver

The read driver is responsible for providing the read voltage to the BL during a read operation. The provided voltage on the BL will produce a current going through the memristor that flows into the read circuit, which is connected to the SL. However, the read driver should be capable of disconnecting from the BL when a write operation is performed. The simplest way of achieving this is through the use of a single NMOS transistor, as shown in Figure 5.10.

NMOS sizing

The NMOS transistor of the read driver is designed to match the width and length of the NMOS transistors of the write driver. To be more precise, the length is set to the minimum value of 270 nm, while the width is set to 10 µm.

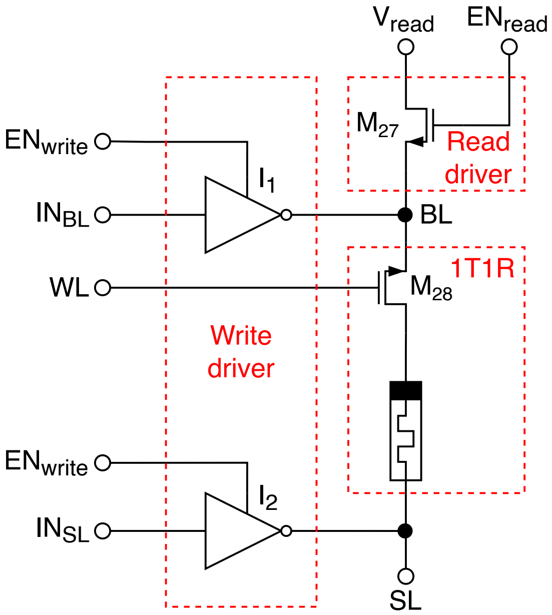


Figure 5.10: Read driver.

5.2 Read circuit & PRR DFT

The PRR DFT, shown in Figure 5.13, is designed to replace the read circuit of a standard RRAM chip. This implies that, next to the fact that it should provide an output signal during a read operation, it should be capable of disconnecting from the SL during a write operation. Disconnecting the PRR DFT from the SL is achieved using the NMOS transistor `M1` and the $\mathtt{EN_{read}}$ signal.

During a read operation, multiple things happen in parallel. First, four duplicates of the current flowing through the memristor are generated by transistors `M2`, `M3`, `M4`, `M5` and `M6`, which together form an NMOS current mirror. Next, the four reference currents, I_{ref0}, I_{ref1}, I_{ref2}, and I_{ref3} are generated by the NMOS transistors `M9`, `M14`, `M19` and `M24`, respectively. These reference currents are then copied using four PMOS current mirrors `M7-8`, `M12-13`, `M17-18` and `M22-23` into their designated branch where the comparison between a reference current and a duplicate of the memristor current is made. Finally, the four inverters `M10-11`, `M15-16`, `M20-21` and `M25-26` provide the output of the four current comparisons in the form of a thermometer code $x_3x_2x_1x_0$. If $x_i = 0$, the memristor current is lower than the i-th reference current. On the other hand, if $x_i = 1$, the memristor current is larger than the i-th reference current.

5.2.1 Read current dependence

Ideally, the voltage over the memristor V_{mem} during a read operation should be independent of its resistance (memristance) and equal to the read voltage V_{read}. In this way, the state of the memristor can easily be determined based on the current going through it, due to the linear relationship between current and resistance. However, the introduction of the diode-connected transistor `M2` in series with the memristor creates a non-linear dependence between the voltage over the memristor V_{mem} and the current flowing through the memristor I_{mem}, as shown in Figure 5.11.

The introduced dependence results in non-linearity of the read current with respect to the resistance. This non-linearity can be decreased by increasing the width of the diode-connected transistor `M2`. In this way, the same change in the drain current of the diode-connected transistor will result in a smaller change in the gate (drain) voltage, as can be seen in Figure 5.12. However, increasing the width of the transistor results in a larger read time due to the larger gate capacitance which causes slower transistor switching speeds. On top of the introduced non-linearity, the diode-connected transistor `M2` decreases the voltage over the memristor by at least $V_{th} \approx 500$ mV, as can be seen in Figure 5.5, since a lower

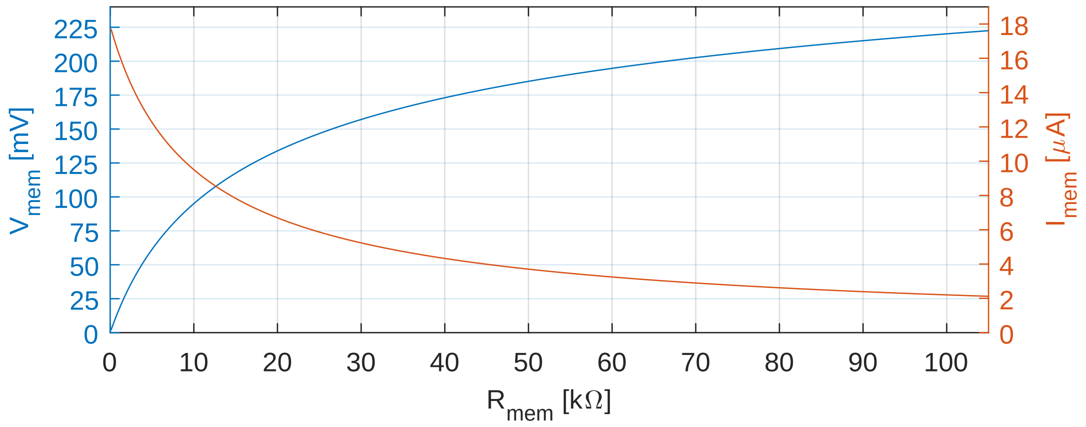


Figure 5.11: Voltage and current dependence on the memristance.

voltage would result in the transistor entering the cut-off region. The minimum voltage drop across the diode-connected transistor can be compensated for by increasing the read voltage by the same amount. For the aforementioned reason, $V_{read} = 750$ mV was selected as the read voltage.

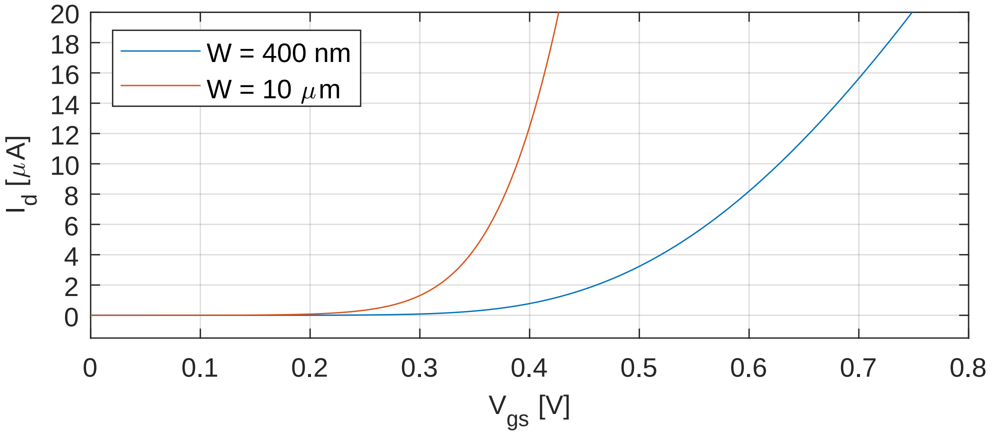


Figure 5.12: Relationship between the drain current I_d and the gate-source voltage V_{gs}.

5.2.2 Current references selection

The four reference currents, I_{ref0}, I_{ref1}, I_{ref2}, and I_{ref3}, should be selected such that every reference current is placed on a boundary between two memristor states. In practice, the undefined state is commonly defined as the region from 40% to 60% between the R_{LRS} and R_{HRS} [12]. Since $R_{LRS} = 4$ kΩ and $R_{HRS} = 45$ kΩ, the boundaries of the undefined state should be in the vicinity of $R_{ref2} \approx 20$ kΩ and $R_{ref1} \approx 29$ kΩ. For simplicity, the reference currents are rounded to $I_{ref1} = 5$ µA and $I_{ref2} = 7$ µA resulting in the undefined state covering the resistance values from 19 kΩ to 32 kΩ.

For the boundaries of the Deep states, the current references are chosen such to provide enough margin around the LRS and HRS, while still allowing room for the Deep states to occur. Through inspection of Figure 5.11, $I_{ref0} = 3$ µA and $I_{ref3} = 16$ µA are chosen as the reference currents for the Deep states.

5.2.3 Transistor sizing

In order to achieve correct functioning of the PRR DFT circuit, it is necessary to properly size the transistors.

NMOS current mirror

The sizing of the NMOS current mirror, consisting of transistors `M2`, `M3`, `M4`, `M5` and `M6`, is mainly determined by the aforementioned voltage drop over the diode-connected transistor `M2` and the tracking accuracy of the current mirror.

For a current mirror, or a current source in general, it is desired that the output current is independent of the output voltage [55]. In this way, optimal current tracking of the current mirror is assured. The problem with using small-feature-size NMOS transistors as current sources is that, in the saturation region, the drain current is not independent of the drain voltage. This is caused by effectively shortening the channel length due to an increase in the voltage at the drain of the transistor, i.e. CLM. The effect of CLM on the output current of a current mirror can be seen in Equation 5.4 [55]. In this equation, I_{IN} is the input current, I_{OUT} is the output current, $(W/L)_{IN}$ is the width-to-length ratio of the transistor at the input side, $(W/L)_{OUT}$ is the width-to-length ratio of the transistor at the output side, $V_{ds_{IN}}$ is the drain-source voltage of the transistor at the input side and $V_{ds_{OUT}}$ is the drain-source voltage of the transistor at the output side. The CLM parameter λ is used to capture the strength of CLM and it is inversely proportional to the length of the transistor. In other words, transistors of larger length suffer from less severe CLM than transistors of shorter length. Through simulations, it is determined that using a length of 600 nm for the NMOS current mirror results in accurate current tracking while keeping the additional area consumption limited.

$$I_{OUT} = \frac{(W/L)_{OUT}}{(W/L)_{IN}} I_{IN} \left(1 + \lambda(V_{ds_{OUT}} - V_{ds_{IN}})\right) \tag{5.4}$$

In order to decrease the influence of the diode-connected transistor `M2` on the voltage over the memristor during a read operation, the width of the diode-connected transistor should be increased. Since it is desired to have a 1-to-1 duplication of the current going through the memristor, all the transistors of the NMOS current mirror should have the same $\frac{W}{L}$ ratio. Through simulations, it is determined that using a width of 10 µm for the NMOS current mirror provides the ability to distinguish between all the memristor states based on the current going through it, while keeping the area consumption and read time manageable.

PMOS current mirrors

Similar to the design of an inverter, in order for the current comparison to be accurate, the driving power of the PMOS current mirrors should match the driving power of the NMOS current mirror. For this reason, the length of the four PMOS current mirrors `M7-8`, `M12-13`, `M17-18` and `M22-23` is set to 600 nm, while the width is set to $3 \cdot 10 = 30$ µm.

Reference currents

The reference currents are generated by simply providing V_{dd} to the gates of the NMOS transistors `M9`, `M14`, `M19` and `M24`. The length of every transistor is set so that its drain current matches one of the four reference current I_{ref0}, I_{ref1}, I_{ref2} and I_{ref3}, while the width is kept at the minimum value of 400 nm. To be more precise, the length of `M9`, `M14`, `M19` and `M24` is set to 10.44 µm, 6.245 µm, 4.445 µm and 1.945 µm, respectively.

Inverters

The four inverters `M10-11`, `M15-16`, `M20-21` and `M25-26` are used to generate the output logic signals x_{0-3}. Since they do not require large driving power, the NMOS transistor `M11`, `M16`, `M21` are kept at the minimum length of 270 nm and the minimum width of 400 nm. The PMOS transistors `M10`, `M15`, `M20` are kept at the same minimum length of 270 nm. However, their width is increased to 1.2 µm to match the driving power of the NMOS transistors.

5.3 Design parameters

In this section, a summary of all the design parameters is given. In Table 5.2, the important input parameters for the circuit are presented. In this table, V_{dd} represents the power supply (write) voltage, V_{read} the read voltage, V_{WL} the WL voltage, t_{set} the duration of the set operation, t_{reset} the duration of the reset operation and t_{read} the duration of the read operation.

Table 5.2: Input parameters.

V_{dd}	V_{read}	V_{WL}	t_{set}	t_{reset}	t_{read}
1.5 V	750 mV	1.8 V; 2.5 V (reset)	40 ns	7.22 µs	60 ns

In Table 5.3, a summary of the type, length and width for every transistor can be seen. The transistors that share the same parameters are merged into one row to reduce the table size.

Table 5.3: Transistor parameters.

Name	Type	Length	Width
M1; M27	NMOS	270 nm	10 µm
M2-6	NMOS	600 nm	10 µm
M7-8; M12-13; M17-18; M22-23	PMOS	600 nm	30 µm
M10; M15; M20; M25	PMOS	270 nm	1.2 µm
M11; M16; M21; M26	NMOS	270 nm	400 nm
M9	NMOS	10.44 µm	400 nm
M14	NMOS	6.245 µm	400 nm
M19	NMOS	4.445 µm	400 nm
M24	NMOS	1.945 µm	400 nm
M28	NMOS	270 nm	500 nm
I1-2.M1	PMOS	270 nm	1.2 µm
I1-2.M2	NMOS	270 nm	400 nm
I1-2.M3-4	PMOS	270 nm	30 µm
I1-2.M5-6	NMOS	270 nm	10 µm

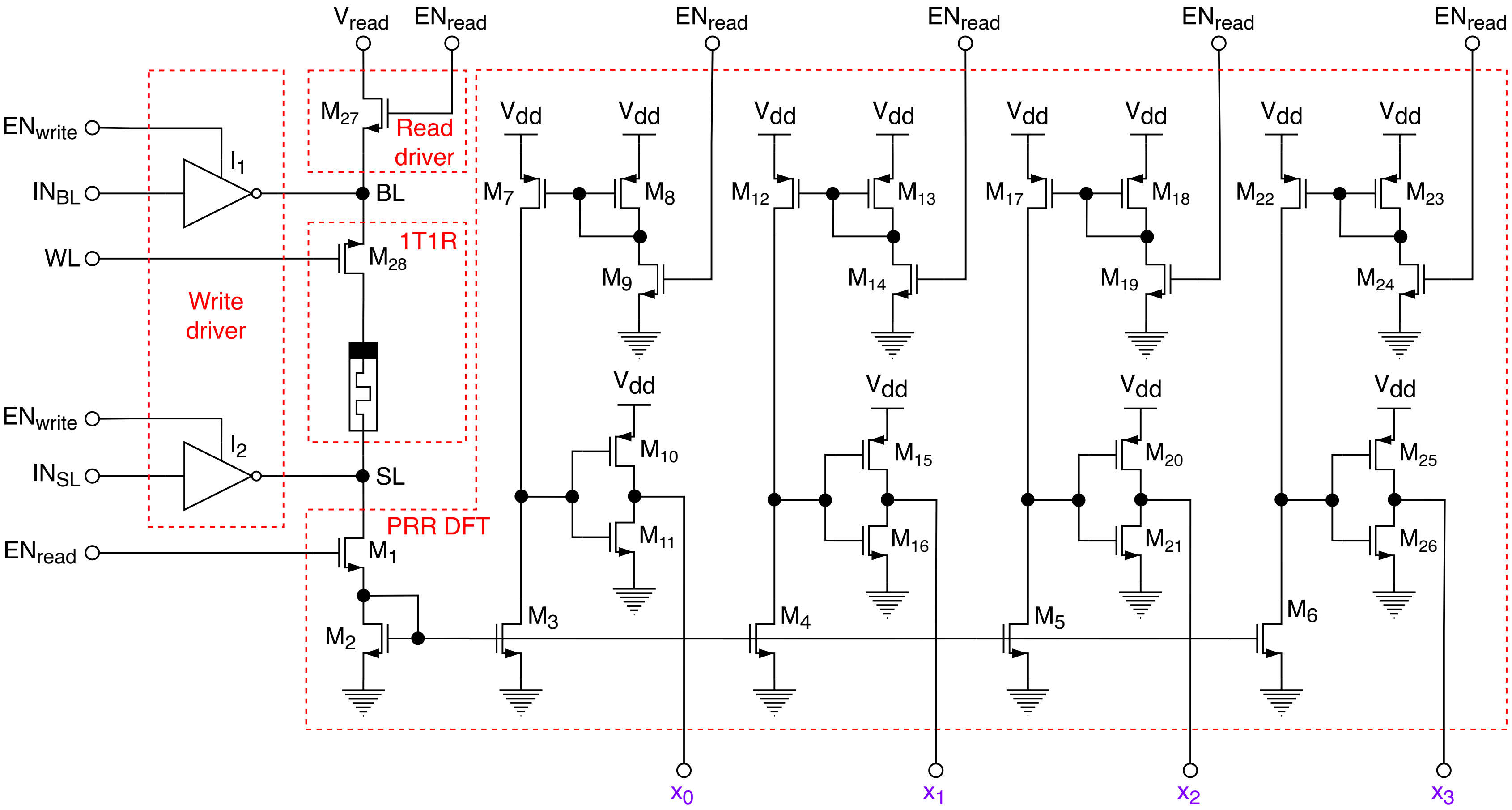


Figure 5.13: PRR DFT.

6. PRR DFT validation & test development

In this chapter, the validation of the PRR DFT is performed. First, the experimental setup is provided. After that, the ability of detecting all the five memristor states is validated. Next, the resilience against process variations during normal operation is measured. After that, resistive defect injection into the 1T1R cell is performed and the detection capability of the PRR DFT is charted. Finally, a March test is devised that covers all the identified fault models. For increased transparency and ability to verify the obtained results, all the source code for the performed validations is made publicly available [56].

6.1 Experimental setup

To properly validate the PRR DFT, an experimental setup is created consisting of three main parts:

- Nominal setup
- Process variations setup
- Defect injection setup

The nominal setup represents an ideal RRAM chip with no defects and is used to validate that the PRR DFT can detect all five memristor states during nominal operation. For this setup, the circuit in Figure 5.13 is implemented in Cadence Virtuoso using the JART VCM v1b model [36] for the memristor and the TSMC 40 nm 2.5 V model library for the CMOS transistors. For improved realism, 150 fF capacitors are added to the WL, SL and BL nodes which represent the line capacitances. The input signals for the circuit are generated using piecewise linear voltage sources with a rise time and fall time of 1 ns. The exact value of an input signal for a specific operation can be seen in Table 6.1. Furthermore, the simulation is performed using a transient analysis in Cadence Spectre with `errpreset` set to conservative.

Table 6.1: Utilised input signals per operation.

	EN_{write}	EN_{read}	IN_{BL}	IN_{SL}	WL
NOP	GND	GND	V_{dd}	V_{dd}	GND
Set	V_{dd}	GND	GND	V_{dd}	1.8 V
Reset	V_{dd}	GND	V_{dd}	GND	2.5 V
Read	GND	V_{dd}	V_{dd}	V_{dd}	1.8 V

The process variation setup is based on the nominal setup with the difference that the transistors are replaced with variants that include chip-to-chip and device-to-device mismatch. In this way, 1000 Monte Carlo iterations are performed using Cadence SpectreMDL to capture the impact of process variations on the accuracy of the PRR DFT.

The defect injection setup is similar circuit-wise to the nominal setup with only one difference. The difference lies in the fact that the 1T1R cell is replaced with a defective 1T1R cell that contains 17 resistive defects, as can be seen in Figure 6.10. These resistive defects are not all activated at the same time but are considered one at a time. In terms of simulation, scripts are made that utilise Cadence SpectreMDL to sweep the strength of every defect separately. Next to the fact that the strength of every defect is swept, a sweep is also performed for all sensitisation sequences up to two operations that end with a read. A visualisation of the defect injection setup can be seen in Figure 6.1.

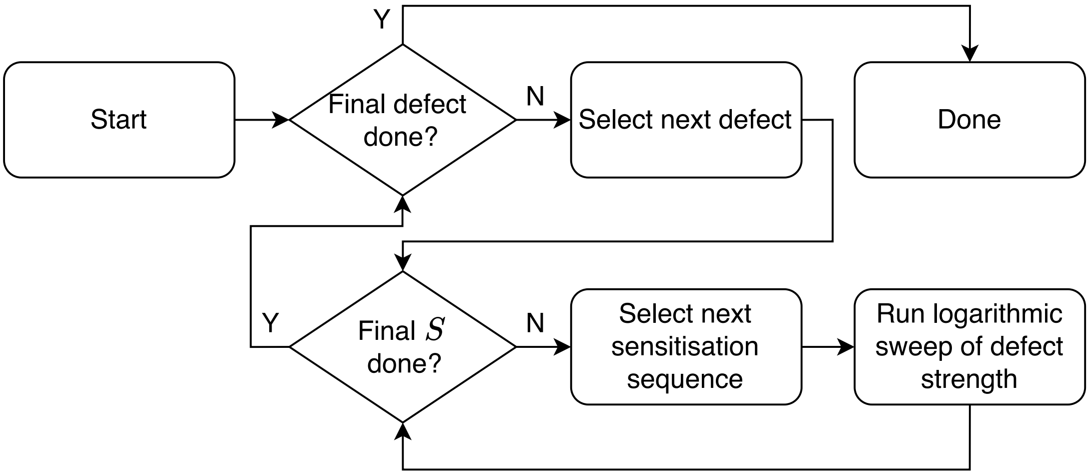

Start
Final defect done?
Y
N
Select next defect
Done
Final S done?
Y
N
Select next sensitisation sequence
Run logarithmic sweep of defect strength

Figure 6.1: Flow chart of the defect injection setup.

6.2 Memristor state detection

The first validation step is to confirm that the PRR DFT is capable of detecting all the five memristor states. For this purpose, the memristor is replaced with a regular resistor and its value is swept. In Figure 6.3, the output signals x_0, x_1, x_2 and x_3 as a function of the memristor current can be seen. From this figure, it can be concluded that every output signal switches exactly at its reference current, as specified in Subsection 5.2.2.

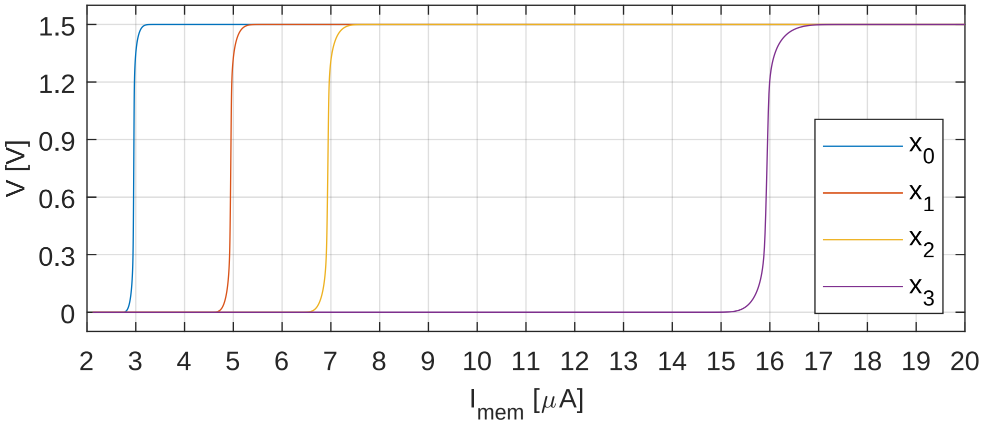


Figure 6.2: PRR DFT output as a function of the memristor current.

To further validate if the memristor current is representative of its memristance, Figure 6.3 shows the output signals as a function of the memristance. Here, it can be seen that all five memristor states are uniquely defined by a combination of the output signals.

Now that the correctness of the PRR DFT output is validated by means of a sweep, the PRR DFT is simulated during normal operation. For this purpose, the sensitisation sequence $S = 0r0w1r1w0r0$ is used. In Figure 6.4, the voltage over the memristor V_{mem} and the four output signals x_0, x_1, x_2 and x_3 can be seen. At $t = 4$ µs, the first read operation is performed, which is indicated by the positive 60 ns pulse in V_{mem}. Since the memristor is initialised in the HRS, only the x_0 output signal is high during the first read operation. At $t = 12$ µs, the memristor is switched from HRS to LRS (set operation), which is indicated by the positive 40 ns pulse in V_{mem}. During the set operation, all the output signals are low since the PRR DFT is disconnected from the SL. At $t = 20$ µs, the second read operation is

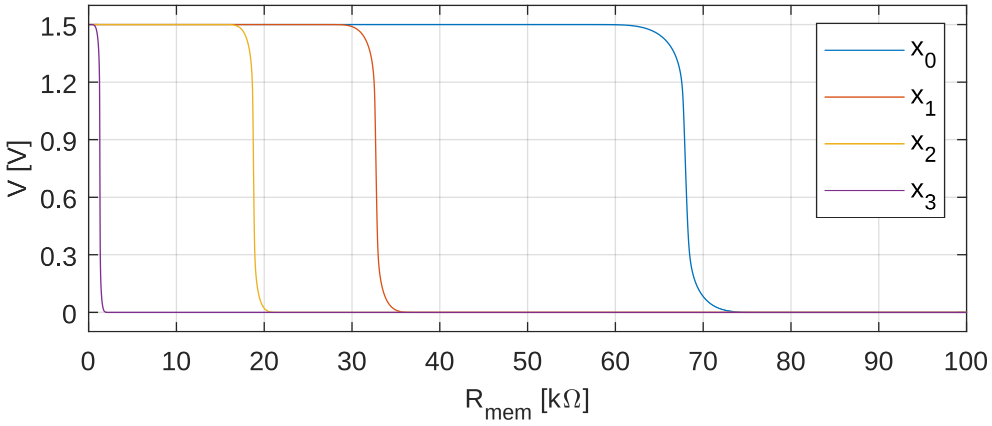


Figure 6.3: PRR DFT output as a function of the memristance.

performed, which is indicated by the positive 60 ns pulse in V_{mem}. During the second read operation, the outputs x_0, x_1 and x_2 are high, which confirms that the memristor is indeed in the LRS. At $t = 28$ µs, the memristor is switched from LRS to HRS (reset operation), which is indicated by the negative 7.22 µs pulse in V_{mem}. Just as before, there is no output from the PRR DFT during the reset operation. Finally, at $t = 36$ µs, the third read operation is performed, which is indicated by the positive 60 ns pulse in V_{mem}. Since the memristor is brought back into HRS, only the output x_0 is high during the third read operation. Based on data from Figure 6.4, it can be concluded that the PRR DFT behaves as expected during normal operation.

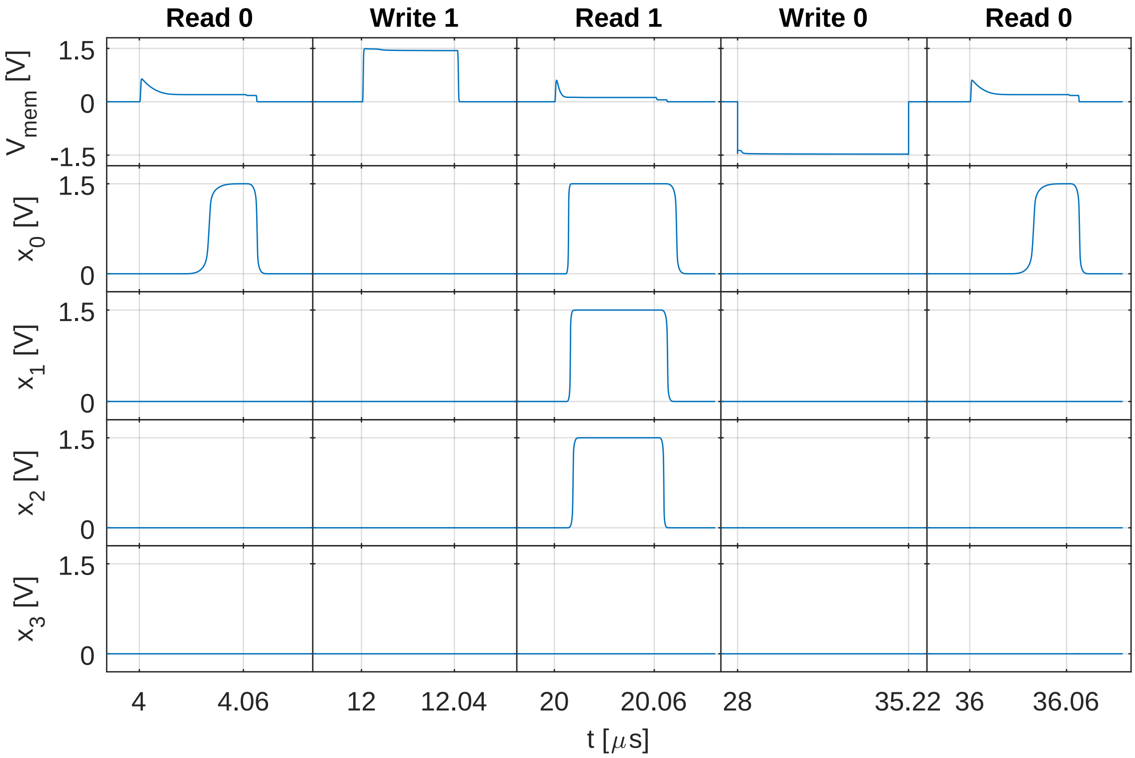


Figure 6.4: Regular operation.

In order to test the detection of the undefined state, the same sensitisation sequence $S = 0r0w1r1w0r0$ is used. However, this time the duration of the reset operation is reduced from 7.22 µs to 400 ns. By prematurely ending the reset operation, the memristor will go from LRS to the undefined state instead of HRS. The results of this simulation can be seen in Figure 6.5. At $t = 28$ µs, the reset operation is performed for just 400 ns instead of the required 7.22 µs, which can be seen by the negative 400 ns pulse in V_{mem}. After that, at $t = 36$ µs, the output signals x_0 and x_1 are high, indicating that the memristor is indeed in the undefined state. With this simulation, the detection of the undefined state during normal operation is confirmed.

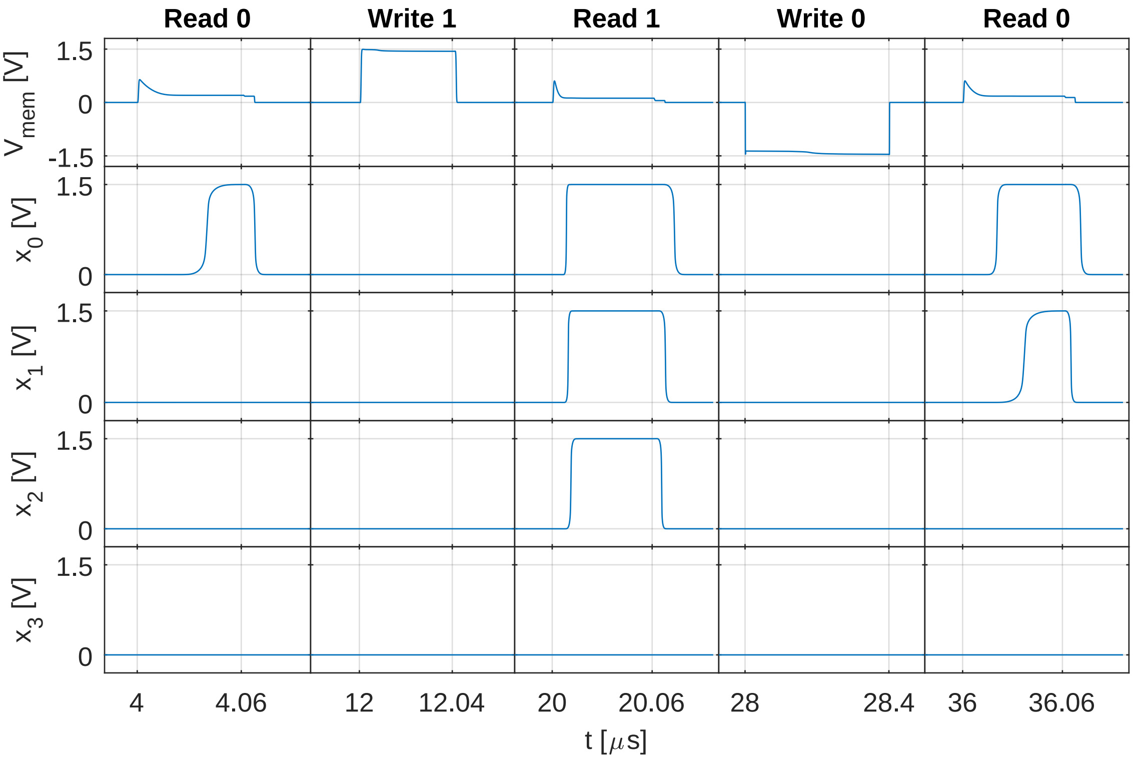


Figure 6.5: Forcing the memristor into the undefined state.

In order to test the detection of the deep 1 state, the same sensitisation sequence $S = 0r0w1r1w0r0$ is used. However, this time the WL voltage is increased from 1.8 V to 2.5 V during the set operation. The increase in WL voltage will result in a higher set compliance current, causing the memristor to go into the deep 1 state. The results of this simulation can be seen in Figure 6.6. At $t = 12$ µs, a set operation is performed using the aforementioned higher WL voltage. After that, at $t = 20$ µs, all the output signals are high, indicating that the memristor is indeed in the deep 1 state. Moreover, at $t = 36$ µs, all the output signals are still high even after a reset operation has been performed at $t = 28$ µs. The reason behind this is that, since the memristor is so far into the deep 1 state, the reset operation is not strong enough to get it out of the deep 1 state. With this simulation, the detection of the deep 1 state during normal operation is confirmed.

Finally, to test the detection of the deep 0 state, the same sensitisation sequence $S = 0r0w1r1w0r0$ is used. However, this time V_{dd} is increased from 1.5 V to 2 V. In this way, the reset operation is sped up, causing the memristor to go into the deep 0 state. The results of this simulation can be seen in Figure 6.7. At $t = 28$ µs, the reset operation is performed using a higher voltage. After that, at $t = 36$ µs, all the output signals are low, indicating that the memristor is indeed in the deep 0 state. With this simulation, the detection of the deep 0 state during normal operation is confirmed.

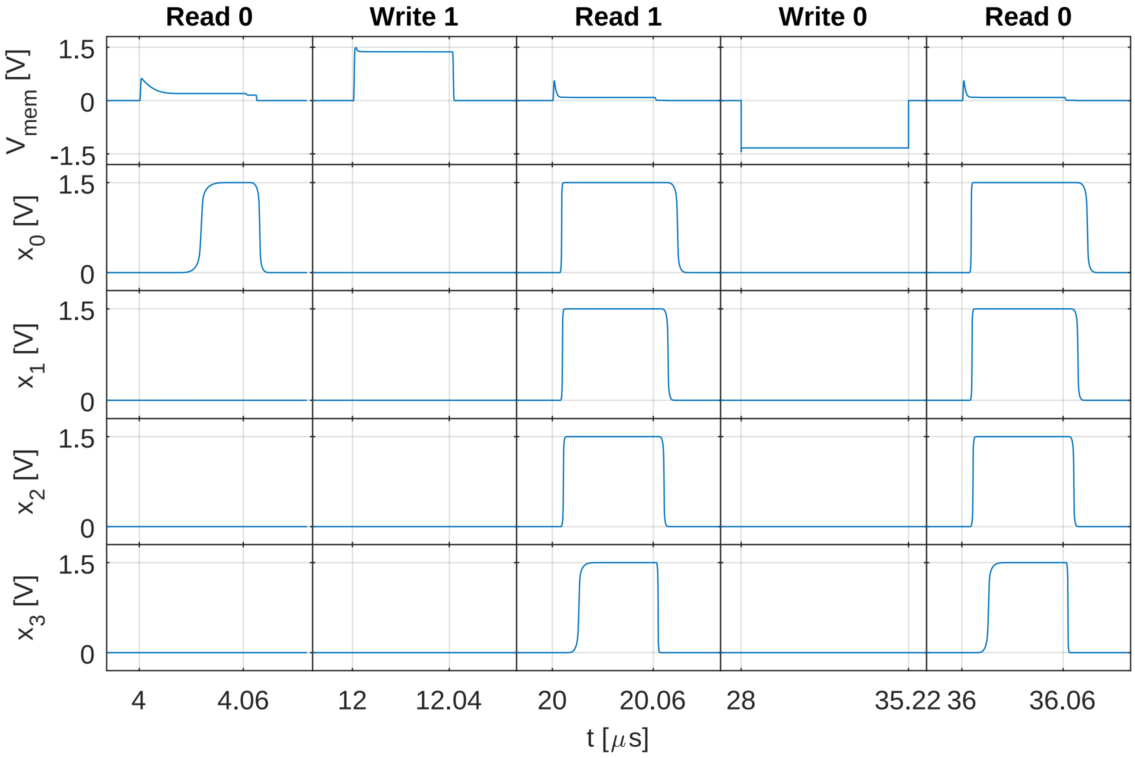


Figure 6.6: Forcing the memristor into the deep 1 state.

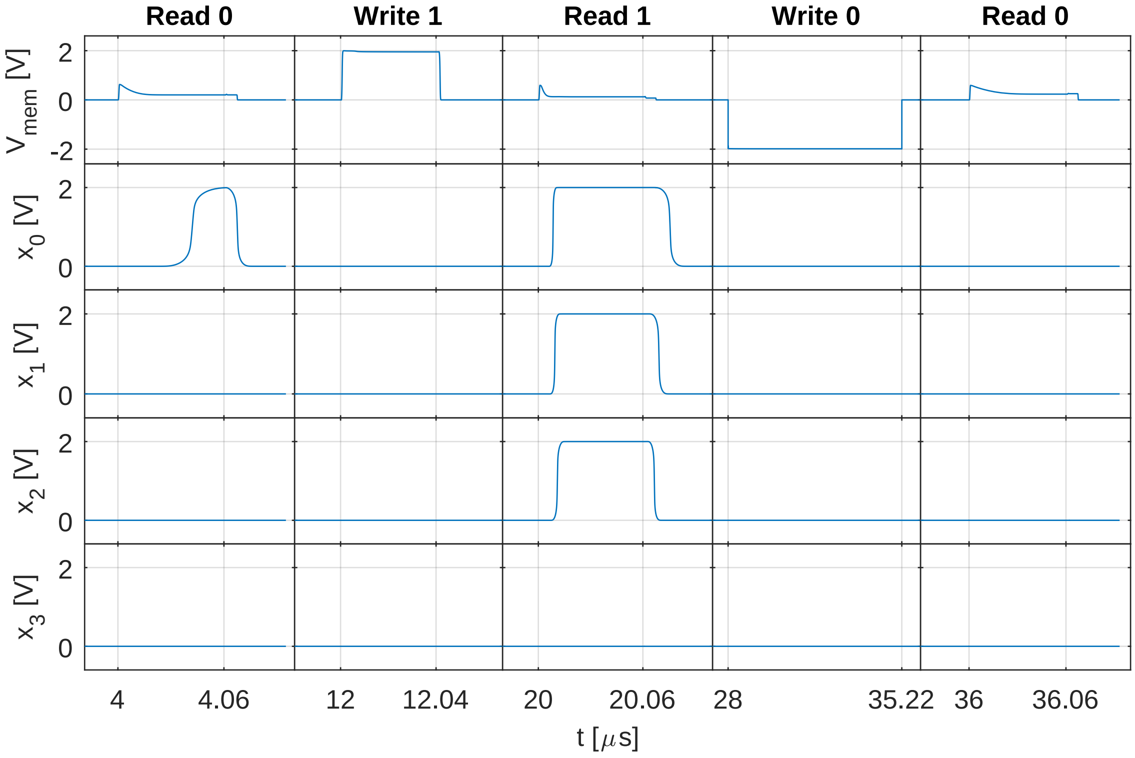


Figure 6.7: Forcing the memristor into the deep 0 state.

Even though the previous figures clearly show that the PRR DFT provides the correct output, it is difficult to compare the output signals to each other. In Figure 6.8, the output signals of all the previous figures can be seen. Keep in mind that the output signals for deep 0 rise to 2 V instead of 1.5 V. As mentioned previously, this is because the V_{dd} voltage is increased to cause the deep 0 state during the reset operation.

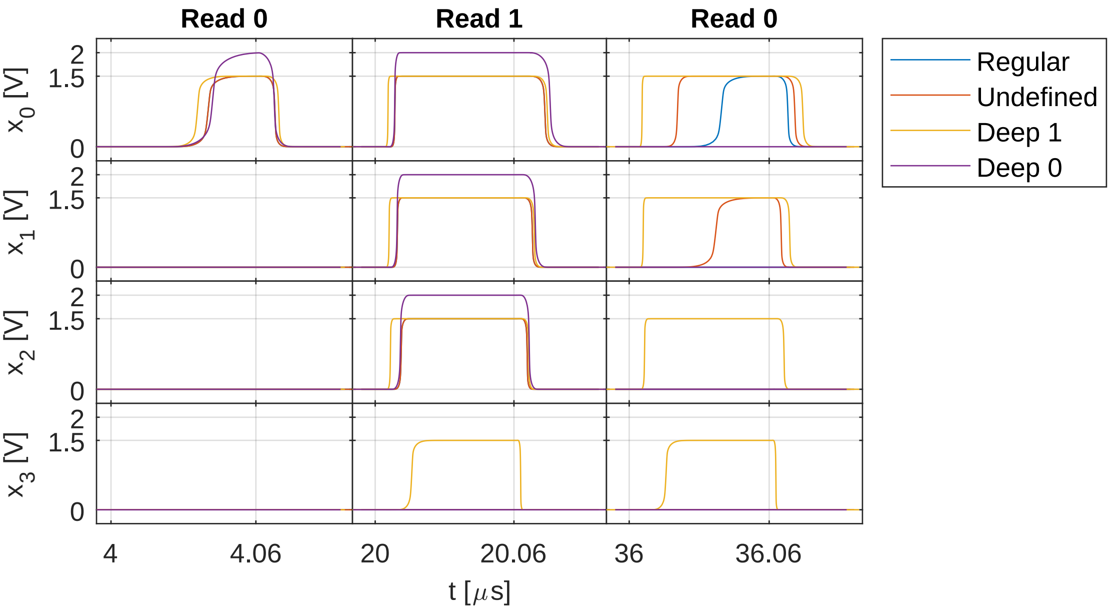


Figure 6.8: Comparison of the output signals.

From the comparison of the output signals, it can be concluded that the delay time is larger for the read 0 operation than for the read 1 operation. The reason behind this is that the current is lower during a read 0 operation, resulting in slower charging/discharging of the capacitances present in the circuit. In general, the delay time of the output signal is dependent on the magnitude of the current going through the memristor during a read operation.

6.3 Process variations analysis

To further validate the implementability of the PRR DFT, the impact of process variations on the correctness of the circuit is explored. This is achieved by replacing every transistor with a transistor that includes statistical mismatch from the TSMC model library.

The Monte Carlo analysis is conducted for 1000 iterations, in which one read 0 operation and one read 1 operation is performed per iteration. It should be noted that, even though both read operations are performed in the same iteration, the read operations do not affect each other since the memristor is always placed in the correct state before every read operation. The summarised results of the Monte Carlo analysis can be seen in Table 6.2.

Table 6.2: Results of the Monte Carlo analysis.

	r0	r1	Total
Amount	1000	1000	2000
Correct	924	994	1918
Incorrect	76	6	82
Pass rate	92.40%	99.40%	95.90%

From Table 6.2, it can be concluded that 95.90% of the 2000 read operations result in the correct output. What stands out is that there are more incorrect outputs for the read 0 operation than there are for the read 1 operation. This discrepancy is caused by two aspects: the distance between reference currents and the chosen read time.

The reference currents at the boundaries of HRS are much closer to each other than the reference currents at the boundaries of LRS, as can be seen in Figure 6.2. Small deviations in the sizing of the transistors will result in the output current of the current mirrors deviating from the input current. This small deviation between the input and output current does not present a substantial impact on the output of the PRR DFT when the reference currents are further away from the nominal current of the state, as is the case with LRS. However, when the reference currents are closer to the nominal current of the state, as is the case with HRS, then the impact on the output of the PRR DFT is more substantial.

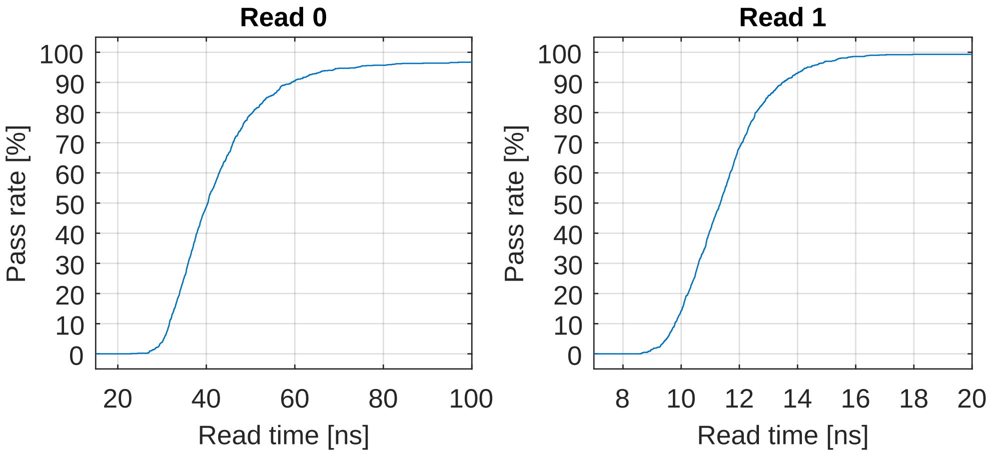


Figure 6.9: Impact of the read time on the pass rate under process variations.

On top of the aforementioned aspect, the read time also affects the pass rate. Under process variations, the switching times of the transistors will deviate from the nominal value. This results in an incorrect output, although the current is within the correct boundaries, since the output signal did not have enough time to rise to its final value. In Figure 6.9, the impact of the read time on the pass rate under process variations can be seen. From this figure, it can be concluded that the read 0 operation requires a longer read time than the read 1 operation for the same pass rate. This is attributed to the fact that the current during a read 0 operation is approximately three times lower than the current during a read 1 operation. The lower current will results in slower charging/discharging of the capacitances that are present in the circuit. The conclusion is that, by changing the read time, the pass rate can also be changed. For example, if the read time is increased from 60 ns to 100 ns, the overall pass rate would increase from 95.90% to 98.05%. All in all, a 95+% pass rate can be seen as an indication that the PRR DFT is decently resilient against process variations.

6.4 Defect detection

The next step is to validate the resistive defect detection capability of the PRR DFT. For this purpose, the normal 1T1R cell is replaced with a defective 1T1R cell which contains 17 resistive defects. The defective 1T1R cell can be seen in Figure 6.10. It should be noted that the defects are not active at once but are considered one at a time. Since this experiment contains large amount of data, the complete results of this experiment and their in-depth elaboration are presented in Appendix A. However, the summarised results and main findings are presented in this section.

In total, 17 resistive defects are considered. These defects are logarithmically swept from 100 Ω to 100 MΩ using 101 steps. Moreover, eight different sensitisation sequences per sweep are used. This results in a total of 13,736 measurements performed. In order to measure the improvement in detection capability of the PRR DFT, a regular read circuit with one reference in the middle of the undefined state is simulated as well. To be more precise, one reference at $I_{ref} = 6$ µA is utilised and the provided output signal is either logic 0 or logic 1. The summarised result of this simulation is shown in Table 6.3. From the results of Table 6.3, it can be concluded that the PRR DFT is able to detect the defect in 43.05% of

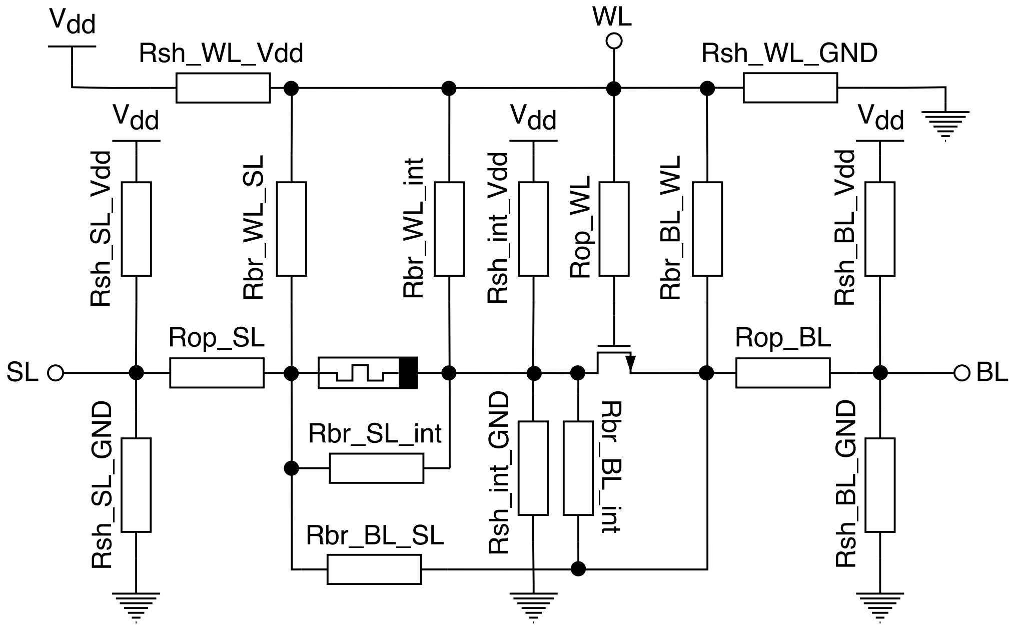


Figure 6.10: 1T1R resistive defect injection. Adapted from [27]

13,736 measurements, while the regular read circuit sits at only 20.79%. That is an improvement of 107.07%. It should be noted that not obtaining 100% defection detection for this experiment is not only expected but also a positive outcome. In the case that 100% defect detection is obtained, that implies that even the smallest defect strength results in an incorrect output and that the DFT would not be able to handle process variations.

Table 6.3: Comparison between a regular read circuit and the PRR DFT in terms of resistive defect detection.

	Regular	PRR DFT	Improvement
Measurements	13,736	13,736	–
Detectable	2,856 (20.79%)	5,914 (43.05%)	107.07%
Undetectable	10,880 (79.21%)	7,822 (56.95%)	–

To get a clearer picture, the sensitisation sequences per defect are merged and only the number of detectable defect strengths per defect are considered. The result of this change can be seen in Table 6.4. From Table 6.4, it can be concluded that the detection capability is improved with respect to a regular read circuit. For some defects, the detection capability is substantially improved (Rsh_BL_GND), while for others it is marginally improved (Rop_BL) or stays the same (Rop_SL). Overall, an improvement of 14.79% is achieved when eight sensitisation sequences are considered.

Furthermore, from Table 6.4, it can be concluded that the relative improvement per defect between a regular read circuit and the PRR DFT is larger when only the $0r0$ and $1r1$ sensitisation sequences are considered than when all eight sensitisation sequences are considered. Moreover, an overall improvement of 17.86% is obtained in that case. This is contributed to the fact that the PRR DFT can detect five states instead of just two, as is the case with a regular read circuit. The ability to detect all the five states allows for higher defect detection capability even when only a read operation is performed. Furthermore, the total amount of detectable defect strengths for the PRR DFT when using only the sensitisation sequences $0r0$ and $1r1$ is slightly higher than the total amount of detectable defect strengths for the regular read circuit using eight sensitisation sequences. This further proves the effectiveness of the PRR DFT. However, in terms of absolute defect detection, using more sensitisation sequences will result in the detection of more defect strengths, as can be seen from the “Total” row of Table 6.4.

Table 6.4: Detectable defect strengths (out of 101) per defect.

	8 sensitisation sequences			Only $0r0$ and $1r1$		
	Regular	PRR DFT	Improvement	Regular	PRR DFT	Improvement
Rop_BL	83	84	1.20%	62	68	9.68%
Rop_SL	82	82	0.00%	62	68	9.68%
Rop_WL	53	54	1.89%	43	50	16.28%
Rbr_BL_int	0	36	∞%	0	25	∞%
Rbr_BL_SL	48	51	6.25%	48	51	6.25%
Rbr_BL_WL	35	38	8.57%	24	27	12.50%
Rbr_SL_int	47	51	8.51%	45	49	8.89%
Rbr_WL_int	50	53	6.00%	46	51	10.87%
Rbr_WL_SL	62	65	4.84%	62	65	4.84%
Rsh_BL_GND	25	42	68.00%	25	37	48.00%
Rsh_BL_Vdd	33	35	6.06%	30	34	13.33%
Rsh_int_GND	47	57	21.28%	41	52	26.83%
Rsh_int_Vdd	47	51	8.51%	45	50	11.11%
Rsh_SL_GND	45	70	55.56%	45	65	44.44%
Rsh_SL_Vdd	60	63	5.00%	60	63	5.00%
Rsh_WL_GND	45	46	2.22%	34	37	8.82%
Rsh_WL_Vdd	29	30	3.45%	0	0	0.00%
Total	791	908	14.79%	672	792	17.86%

6.5 March test development

A March test is devised that covers all the identified fault models, as presented in Subsection 3.3.3. This newly devised March test, called PRR March, is shown in Equation 6.1. The PRR March test was developed by first considering the necessary operations to detect and sensitise every fault separately before merging those operations into one March test. In this way, the coverage of all the considered faults is ensured while keeping the amount of memory operations at a minimum.

$$\text{PRR March: } \{\ \text{M1: } \Uparrow (r1, w0);\ \ \text{M2: } \Uparrow (r0, r0, w1);\ \ \text{M3: } \Downarrow (r1, w0);\ \ \text{M4: } \Downarrow (r0, w1); \} \tag{6.1}$$

The explanation of how the PRR March test can detect the identified fault models follows:

- SAF: this fault can be detected by performing a $r0$ and $r1$ operation, since the faulty cell is always in state 0 or state 1. In other words, any time $r0$ is performed, the cell is checked for SA1 while, any time $r1$ is performed, the cell is checked for SA0. In the case of the PRR March, the $r0$ operation in M2 or M4 detects SA1, while the $r1$ operation in M1 or M3 detects SA0.
- TF: this fault can be detected by performing a write operation followed by a read operation. To be more precise, $0w1r1$ and $1w0r0$ are sensitisation sequences that detect TF. In the case of the PRR March, TF0 is sensitised by $w0$ in M3 and detected by $r0$ in M4. On the other hand, TF1 is sensitised by $w1$ in M2 and detected by $r1$ in M3.
- IRF: since IRF results in an incorrect read output, it can be detected in the same way as SAF.
- RDF: this fault is sensitised by performing one read operation and detected by performing another read operation. Since the read operation is in the set direction, the cell can only change if it is initially in HRS. To be more precise, the $0r0r0$ sensitisation sequence should be used. In the case of the PRR March, RDF is sensitised by $w0$ in M1 and the first $r0$ in M2, while it is detected by the second $r0$ in M2.
- CFst: in total, CFst has four variants. First the coupling can be between state 1 and state 1 or between state 0 and state 0 of the aggressor and victim cell, respectively. Furthermore, the address of the victim cell can either be higher or lower than the address of the aggressor cell. These four variants can be detected by $\{w0; \Uparrow r0w1\}$, $\{w0; \Downarrow r0w1\}$, $\{w1; \Uparrow r1w0\}$ and $\{w1; \Downarrow r1w0\}$.

$\{w0; \Uparrow r0w1\}$ is implemented by $w0$ in M1 and $r0w1$ in M2. $\{w0; \Downarrow r0w1\}$ is implemented by $w0$ in M3 and $r0w1$ in M4. $\{w1; \Uparrow r1w0\}$ is implemented by $w1$ during forming and $r1w0$ in M1. Finally, $\{w1; \Downarrow r1w0\}$ is implemented by $w1$ in M2 and $r1w0$ in M3.

- WDF: since WDF has the same effect as CFst, it can be detected in the same way as CFst.
- UWF: since the PRR DFT can detect the undefined state during a read operation, UWF can be detected in the same way as TF.
- URF: since the PRR DFT can detect the undefined state during a read operation, URF can be detected in the same way as SAF.
- Deep: since the PRR DFT can detect the deep states during a read operation, Deep can be detected in the same way as SAF or TF.
- CFud: since the PRR DFT can detect the undefined state during a read operation, CFud can be detected in the same way as CFst.
- IUSF: the intermittent nature of IUSF results in the inability of detection during testing. However, since the PRR DFT is a replacement of the read circuit, IUSF can be detected during every read operation that is performed in-field.

7. Discussion

In this chapter, the results of this thesis are discussed. First, the PRR DFT is compared to the state of the art. After that, the additional benefits and usages of the PRR DFT are provided. Finally, the potential future steps are given.

7.1 Comparison

The comparison between regular March tests, DFTs and the PRR DFT can be seen in Table 7.1. In this table, under the listed faults, "Y" denotes that the fault is covered, "N" denotes that the fault is not covered and "P" denotes that the fault is partially covered. For the test time, "N" denotes the total number of cells in the RRAM array, "T" the tile size, "x" the ratio between $w0$ and $w1$ duration, while "a" and "b" denote the number of consecutive $w1$ and $w0$ operations to detect dWDF, respectively. From this comparison, it can be concluded that the PRR DFT offers complete fault coverage while also keeping the test time at a minimum.

Table 7.1: Full march test, DFT and PRR comparison.

		Conventional						Unique						Test Time	
Year		SAF	TF	WDF	IRF	RDF	CFst	UWF	URF	Deep	IUSF	CFud	Coverage	Write	Read
1993	March C– [46]	Y	Y	N	Y	N	Y	N	N	N	N	N	36%	5N	5N
2013	March–MOM [10]	Y	Y	P	N	N	P	N	Y	Y	N	N	36%	5N	4N
2015	March–1T1R [48]	Y	Y	Y	N	N	Y	N	N	P	N	N	36%	(1+2a+2b)N	5N
2015	March C* [47]	Y	Y	N	Y	Y	Y	N	N	N	N	N	45%	4N	6N
2016	March C*–1T1R [49]	Y	Y	N	Y	Y	Y	N	N	Y	N	N	55%	6N	6N
2018	March–CMOL [39]	Y	Y	N	Y	Y	Y	N	N	Y	N	N	55%	N.A.	N.A.
2017	March W–1T1R [50]	Y	Y	Y	Y	Y	Y	N	N	Y	N	N	64%	9N	8N
2017	Parallel March [9]	Y	Y	N	Y	Y	Y	N	N	Y	N	N	55%	4(N+1)	5N+N$_r$
2013	Sneak-path Testing [10]	Y	Y	P	N	N	P	N	Y	Y	N	N	36%	7N	5/3N
2012	Weak-Write [12], [13]	N	N	N	Y	N	N	Y	Y	Y	N	N	36%	No March	
2017	Fast-Write [14]	Y	Y	N	N	N	N	Y	Y	Y	N	N	45%	(4+1/T+x/T)N	6N
2021	On-chip Sensor [15]	Y	Y	Y	Y	Y	N	Y	Y	Y*	N	N	73%	No March	
2021	Enhanced March [16]	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	Y	91%	8N	7N
2021	March RC [16]	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	Y	91%	4N+2	6N
2022	PRR	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	100%	4N	5N

The PRR DFT is capable of delivering complete fault coverage, while keeping the test time low due to its ability of detecting all five memristor states with a single read operation. Other DFTs that employ multiple reference currents [14], [16] need to select which reference to use in a sequential manner, while the PRR DFT uses all four reference currents in parallel. Furthermore, since the PRR DFT is a replacement of the regular read circuit, intermittent faults such as IUSF are detectable. Regarding the test time, the bulk of the read and write operations in the PRR March test is required in order to detect coupling faults. If only single-cell faults would be considered, then the test time would be even lower.

Next to fault coverage and test time, another important metric is area consumption. In Table 7.2, the comparison between state-of-the-art DFTs and the PRR DFT in terms of area overhead is presented. In this table, N represents the total number of memristors, N_r the number of rows and N_c the number of columns in the RRAM array. The PRR DFT consist of 26 transistors in total and it acts as a replacement of the regular read circuit. Since the regular read circuit consists of approximately 12 transistors [57], the additional area overhead of the PRR DFT is 14 transistors per column.

From Table 7.2, it can be concluded that the PRR DFT sits on the lower side of the spectrum in terms of area consumption when compared to state-of-the-art DFTs. This is attributed to the fact that the PRR DFT acts as a replacement of the regular read circuit and to the fact that the design is not over-engineered. By acting as a replacement, the number of transistors saved by removing the original read circuit can be subtracted from the amount of transistors used in the PRR DFT. Furthermore, by keeping the design of the PRR DFT as simple as possible, the area overhead due to unnecessary complexity is reduced. For example, simple current mirrors are used instead of more complex versions such as Widlar, Cascode or Wilson [55]. The more complex versions double the area of the current mirror while only marginally improving the tracking accuracy.

Table 7.2: Comparison between state-of-the-art DFTs and PRR based on the area overhead.

Year	DFT	Area overhead (# of transistors)
2015	MAGIC NOR 1R [8]	$13(2N_c + N_r)$ [16]
2017	MAGIC NOR 1T1R [9]	$13(N_c + 3N_r)$ [16]
2013	Sneak-path Testing [10]	$28 + 26N_r$
2017	Sneak-path Testing using Voltage Bias [11]	$28 + 26N_r$
2012	Weak-Write – SWT [12], [13]	$24 + 18N_r$ [10]
2017	Fast-Write [14]	$24 + 18N_r + 26$
2021	On-chip Sensor [15]	$20N$
2021	Enhanced March [16]	26
2021	March RC [16]	$13(N_c + N_r + 4)$
2022	PRR	$14N_c$

Since it is difficult to compare DFTs solely based on equations, a visual example is shown in Figure 7.1. In this figure, four setups are considered based on different row-to-column ratios of the RRAM array, while the memory size is swept from 1 kb to 1 Gb. The y-axis represents the area overhead, in terms of transistors, relative to the amount of transistors present on a RRAM chip, which is approximated by $N_{RRAM} = N + 45N_c + 12N_r$. From Figure 7.1, it can be concluded that the area (transistor) overhead of the PRR DFT is lower compared to the state-of-the-art DFTs, except for Enhanced March [16]. However, it should be noted that the area overhead reported by the authors of Enhanced March [16] is not realistic. Liu *et al.* assume that only one additional multiplexer is required for the Enhanced March DFT to work. However, it is unfeasible to provide the reference current to all the CSAs in a large memory array using only one multiplexer.

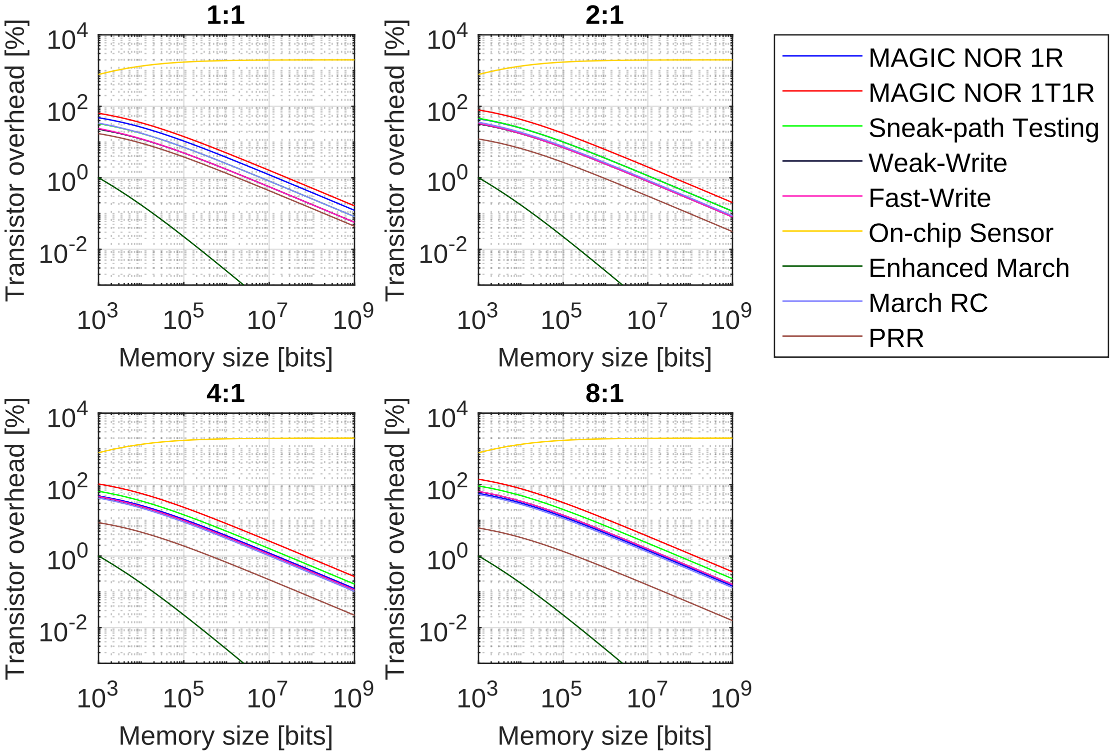


Figure 7.1: Visualisation of the area (transistor) overhead for the DFTs.

Figure 7.1 also shows the impact of the row-to-column ratio of the RRAM array on the area overhead of the PRR DFT. As the row-to-column ratio increases, the difference between the PRR DFT and other DFTs increases as well. This is attributed to the fact that the PRR DFT's area overhead is only dependent on the number of columns and not on the number of rows, resulting in a lower area overhead for larger row-to-column ratios. Keep in mind that the y-axis in Figure 7.1 does not represent the actual area overhead but a transistor overhead relative to an approximation of the number of transistor in a regular RRAM chip. To be more precise, the most important part of this figure is the relative difference between the DFTs and the trend for larger memory sizes.

7.2 Additional benefits & usages

What sets the PRR DFT apart from other DFTs is that it acts as a replacement for the regular read circuit of RRAM. This provides some additional benefits which are not available for other DFTs. First, the PRR DFT is constantly used for every read operation. Other DFTs consume additional area on the chip and they are only used once after fabrication to test the chip for defects. After this post-fabrication test, the consumed area on the chip is basically dead-silicon since the DFT is never used again.

The fact that the PRR DFT is constantly active during every read operation also means that in-field testing is performed. Every time a read operation is performed, the PRR DFT will return the actual state of the memristor instead of just the logic state based on a single reference. This means that, in the case the memristor is in the undefined state or the deep states, the faulty cell can be registered and further steps can be taken. With a regular read circuit, this would not be the case and only a logic 0 or logic 1 will be presented at the output.

The aforementioned in-field testing ability is also the reason why the PRR DFT is able to detect IUSF. Due to the intermittent nature of IUSF, one-off testing after fabrication does not guarantee the detection of this fault. On the other hand, by being able to detect the state of the cell during every read operation, the detection of IUSF is guaranteed.

In terms of in-field testing, the PRR DFT offers a lot of flexibility by being able to individually turn the reference currents on and off. In the case that in-field testing is not required, only one reference current can be turned on. In this way, the PRR DFT behaves like a regular read circuit with one output signal representing state 0 or state 1. On the other hand, if only the detection of the undefined state is desired, then two reference currents can be turned on. In this case, there will be two output signals that together form the thermometer code representing state 0, undefined state and state 1.

If the area usage is highly restricted, the PRR DFT can be changed from parallel reference comparison to sequential reference comparison. Instead of using four comparators, only one comparator can be used with four different reference currents. This can be achieved by connecting four differently sized transistors in parallel that each generate a different reference current when V_{dd} is applied to their gate. The desired reference current for a comparison is then selected by only turning on the transistor that generates that reference current.

Another additional usage of the PRR DFT is diagnosis of customer returns. By simply performing read operations, the actual state of every cell can be determined. The data obtained from reading all the cells can be used to further diagnose the returned RRAM chip.

Furthermore, the PRR DFT can be adapted to work with Multi-Level Cell (MLC). Instead of having five states, of which only two are actual logic state, all the five states can be used as logic states. In this way, multiple bits can be stored inside a single cell. The desired number of bits per cell can be obtained by increasing or decreasing the number of parallel current comparators in the PRR DFT.

In this thesis, the PRR DFT is specifically developed for 1T1R RRAM. However, the PRR DFT can also be used for other architectures, such as 1R and 1D1R. Furthermore, it can also be used for other emerging memory technologies, e.g. STT-MRAM and PCRAM.

Finally, the CMOS design is completely documented in this thesis, together with the validation of the PRR DFT. Moreover, all the source code for the design and validation is provided publicly [56]. By doing so, additional transparency is provided that is missing with the current state of the art.

7.3 Future work

Due to the limited duration of this project, there are still some issues left unsolved. Firstly, a high-level overview and working principle of the CLW DFT is provided. However, the low-level design is not further developed. The voltage drop over the diode-connected transistor made the implementability and scalability of the design difficult since high write voltage and large transistors widths are required. To make the CLW DFT feasible, a better way of current mirroring should be employed.

Secondly, the voltage drop over the diode-connected transistor is also causing difficulties in the PRR DFT design. To combat this, the transistor width of the current mirrors is increased, which results in a reduced influence of the diode-connected transistor's voltage drop on the voltage over the memristor during a read operation. However, this increase in width results in slower switching speed, causing a longer read time. Currently, the PRR DFT has a 60 ns read time, which could further be reduced if a better way of current mirroring is used.

A possible solution for both issues would be to replace the gate-driven current mirrors with bulk-driven current mirrors [58], [59]. With bulk-driven current mirrors, the bulks of the transistors are connected to each other and the bulk of the transistor on the input side is connected to its drain, while the gates of the transistors are biased at a fixed voltage. By connecting the drain of the transistor to the bulk instead of the gate, the voltage drop on the input side of the current mirror is no longer equal to the gate-source voltage, which is dependent on the drain current and always larger than V_{th}. However, bulk-driven current mirrors do not completely solve the dependence of the voltage drop over the transistor on the current going through it, but they do reduce its magnitude. Moreover, bulk-driven current mirrors have a lower output resistance and lower current driving capability than gate-driven current mirrors [59]. All things considered, the best way of implementing the CLW DFT and improving the read time of the PRR DFT remain open topics.

8. Conclusion

The defect-prone nature of RRAM, and the unique faults that come with it, increased the demand for DFTs that can detect said faults. However, the current state of the art is still lacking in a few regions. Most importantly, it is not capable of detecting all unique faults. Moreover, it omits low-level CMOS implementations details, accurate validations, process variations and is lacking the ability to detect intermittent faults. The aim of this work was to fill the gaps in the state of the art.

First, the background information about emerging memory technologies, working principle of RRAM and the fabrication process was provided. It was necessary to include all this information in order to make a self-contained thesis. Furthermore, possible defects during the fabrication process are considered followed by the fault models that encompass those defects.

After providing the necessary background information, a deep-dive into the state-of-the-art testing techniques was performed. The state-of-the-art DFTs were individually explained, their disadvantages were discussed and they were compared to get a clearer picture of the current state of the art. This was an important step to lay the foundation which the work of this thesis can build upon.

From this foundation, the analysis of fault detection capability through read and write operations led to the proposal of the PRR DFT and the CLW DFT. For the CLW DFT, only a high-level overview and working principle were given, while the PRR DFT was selected for further low-level development. The reason for this decision was the influence of the voltage drop over the diode-connected transistor, which lies in series with the memristor, on the voltage over the memristor. Larger current and voltages during a write operation made the implementability of the CLW DFT challenging and scalability difficult.

Nevertheless, the PRR DFT's low-level design in CMOS was provided and the design decisions explained. On top of that, the design of the testing circuit, used in order to validate the PRR DFT, was documented as well. Furthermore, the influence of the voltage drop over the diode-connected transistor on the current going through the memristor was presented. In this way, the impact on the output of the PRR DFT was determined and appropriate measures were taken to mitigate it. The measures include the increase of the transistor width for the current mirrors and selecting the reference currents appropriately.

After the PRR DFT design was completed, it was time to validate its functionality. First, the ability to detect all five memristor states was confirmed. This was checked in two ways. By performing a sweep of the memristance and by using (modified) write operations in order to force the memristor in all the five states. The conclusion from this validation step is that the PRR DFT is capable of detecting all five memristor states. Next, Monte Carlo analysis was performed to measure the impact of process variations on the correctness of the PRR DFT's output during a read operation. In total, 2000 read operations were performed, from which 95.90% were correct. This result indicates that the PRR DFT is decently resilient against process variations. As the final validation step, the ability of detecting resistive defects in the 1T1R cell was charted. For this purpose, 17 resistive defects were injected into the 1T1R cell and the impact on the cell state and the output of the PRR DFT for different defect strengths was measured. Moreover, the measurements were compared to the measurements of a regular read circuit which uses only one reference in the middle of the undefined state. From this comparison, it was determined that the PRR DFT offers better detection of resistive defects than a regular read circuit due to its ability to detect all five memristor states instead of just two. To be more precise, an overall improvement of 14.79% was achieved when eight sensitisation sequences, consisting of at most two operations ending on a read, were used and merged.

Finally, the comparison between the PRR DFT and the state-of-the-art DFTs based on fault coverage, test time and area overhead, was presented. The area overhead of the PRR DFT, in terms of transistor count, was found to be $14N_c$, where N_c is the number of columns in the RRAM array. In order to perform the comparison in terms of fault coverage and test time, first a March test was devised. Based on this March test, it was found that the PRR DFT offers 100% identified fault coverage while only requiring $4N$ write operations and $5N$ read operations, where N is the total amount of memristors (cells) in the RRAM array. All in all, it can safely be concluded that this work achieved its initial goal of filling the aforementioned gaps in the state of the art.

[This Page Intentionally Left Blank]

Acronyms

1D1R One-Diode-One-Resistor 10, 11, 69

1R One-Resistor 10, 11, 21, 24, 69

1T1R One-Transistor-One-Resistor 10, 11, 21, 22, 24, 45, 48, 50, 57, 63, 64, 69, 71

AP Anti-Parallel 4

BBL Blue Bit Line 11

BE Bottom Electrode 4, 5, 7, 11, 12, 14, 45

BEOL Back-End-Of-Line V, 1, 11–13

BISR Built-In Self-Repair 17

BIST Built-In Self-Test 17

BL Bit Line 10, 11, 24, 25, 38, 39, 42, 47, 50, 51, 57, 82

BP Blue Point 11

BWL Blue Word Line 11

CBRAM Conductive Bridge Random-Access Memory 5

CF Conductive Filament 1, 5, 7, 11, 12, 14

CFst State Coupling Fault 16–19, 31, 36, 38, 39, 65–67

CFud Undefined Coupling Fault 17, 19, 31, 32, 36, 39, 66, 67

CLM Channel Length Modulation 48, 54

CLW Closed-Loop Write V, 2, 41–44, 70, 71

CMOL CMOS Molecular 10, 11, 18

CMOS Complementary Metal-Oxide-Semiconductor V, 1, 2, 11–13, 44, 57, 69, 71, 73

CPU Central Processing Unit 1

CSA Current-Sense Amplifier 22, 24, 45, 68

DA Device-Aware 14

DAC Digital-to-Analogue Converter 25

dCFud Dynamic Undefined Coupling Fault 31, 32

Deep Deep State Fault 17–19, 30, 31, 36, 39, 40, 66, 67

DFT Design-For-Testability V, 1, 2, 17, 19–38, 41–45, 52, 54, 56–59, 62–71, 81, 82, 84–92

DRAM Dynamic Random-Access Memory V, 1, 3

dWDF Dynamic Write Disturbance Fault 16, 18, 35, 67

FEOL Front-End-Of-Line 11, 13, 14

FeRAM Ferroelectric Random-Access Memory 1, 3, 4

FL Free Layer 4

FMT Fast March Test 28

FP Fault Primitive 16, 17

GST GeSbTe 4, 5

HRR High Resistance Reference 29, 30

HRS High Resistance State 1, 5, 7, 16, 17, 21, 22, 42, 45–47, 53, 58–60, 63, 65

IRF Incorrect Read Fault 16, 19, 36, 38, 65, 67

IUSF Intermittent Undefined State Fault 17, 19, 35, 36, 40, 66, 67, 69

LRR Low Resistance Reference 29, 30

LRS Low Resistance State 1, 5, 7, 16, 17, 20–22, 42, 45–47, 51, 53, 58, 59, 63, 82

LWV Low Write Voltage 25, 26

MAGIC Memristor-Aided loGIC 21, 35, 68

MFM Metal-Ferroelectric-Metal 4

MIM Metal-Insulator-Metal 5

MLC Multi-Level Cell 69

MOSFET Metal–Oxide–Semiconductor Field-Effect Transistor 3

MTJ Magnetic Tunnel Junction 4

NMOS Negative-Channel Metal-Oxide-Semiconductor 11, 43, 47–49, 51, 52, 54, 55

NVM Non-Volatile Memory V, 1, 4, 5

OxRAM Oxide Random-Access Memory 5, 47

P Parallel 4

PCRAM Phase-Change Random-Access Memory 1, 3–5, 69

PL Pinned Layer 4

PLWV Programmable Low Write Voltage 26

PMOS Positive-Channel Metal-Oxide-Semiconductor 47, 51, 52, 54, 55

PRR Parallel-Reference Read V, 2, 41–45, 52, 54, 56–59, 62–71, 81, 82, 84–92

PSWT Programmable Short Write Time 26

RBL Red Bit Line 11

RD Resistive Defect 14

RDF Read Disturb Fault 16, 18, 19, 36, 38, 65, 67

RL Reference Layer 4

RoD Region-of-Detection 22, 24, 25

RP Red Point 11

RRAM Resistive Random-Access Memory V, 1–14, 16–22, 27, 31, 35, 37, 40, 41, 44, 45, 52, 57, 67–69, 71

RWL Red Word Line 11

SA0 Stuck-at-0 16, 20, 65

SA1 Stuck-at-1 16, 20, 24, 65

SAF Stuck-at-Fault 16, 18, 19, 36–39, 42, 65–67

SL Source Line 11, 38, 39, 42, 47, 50–52, 57, 58, 82

SRAM Static Random-Access Memory 1, 3

STT-MRAM Spin-Transfer Torque Magnetic Random-Access Memory 1, 3, 4, 69

SWF Slow Write Fault 16, 18, 38

SWT Short Write Time 25, 26, 35, 68

TB Tunnel Barrier 4

TE Top Electrode 4, 5, 7, 11, 12, 14, 45, 46, 82, 83

TF Transition Fault 16, 18, 19, 36, 38, 65–67

TMR Tunnelling MagnetoResistance 4

TSMC Taiwan Semiconductor Manufacturing Company 45, 48, 50, 57, 62

URF Undefined Read Fault 17, 19, 36, 39, 66, 67

UWF Undefined Write Fault 17, 19, 25, 31, 36, 39, 40, 66, 67

WDF Write Disturbance Fault 16, 18, 19, 36, 38, 66, 67

WL Word Line 10, 11, 24, 25, 55, 57, 60, 82, 83

[This Page Intentionally Left Blank]

Bibliography

[1] Y. Hayakawa, A. Himeno, R. Yasuhara, *et al.*, "Highly reliable TaOx ReRAM with centralized filament for 28-nm embedded application," *Digest of Technical Papers - Symposium on VLSI Technology*, vol. 2015-August, T14–T15, Aug. 2015, ISSN: 07431562. DOI: 10.1109/VLSIT.2015.7223684.

[2] S. Yu and P. Y. Chen, "Emerging Memory Technologies: Recent Trends and Prospects," *IEEE Solid-State Circuits Magazine*, vol. 8, no. 2, pp. 43–56, Mar. 2016, ISSN: 19430582. DOI: 10.1109/MSSC.2016.2546199.

[3] T. Mikolajick, M. Salinga, M. Kund, and T. Kever, "Nonvolatile Memory Concepts Based on Resistive Switching in Inorganic Materials," *Advanced Engineering Materials*, vol. 11, no. 4, pp. 235–240, Apr. 2009, ISSN: 1527-2648. DOI: 10.1002/ADEM.200800294.

[4] N. Aswathy and N. Sivamangai, "Future Nonvolatile Memory Technologies: Challenges and Applications," pp. 308–312, Oct. 2021. DOI: 10.1109/ACCESS51619.2021.9563288.

[5] H. S. Wong, H. Y. Lee, S. Yu, *et al.*, "Metal-Oxide RRAM," *Proceedings of the IEEE*, vol. 100, no. 6, pp. 1951–1970, 2012, ISSN: 00189219. DOI: 10.1109/JPROC.2012.2190369.

[6] M. Fieback, M. Taouil, and S. Hamdioui, "Testing Resistive Memories: Where are We and What is Missing?" *Proceedings - International Test Conference*, vol. 2018-October, Jan. 2019, ISSN: 10893539. DOI: 10.1109/TEST.2018.8624895.

[7] V. A. Hongal, R. Kotikalapudi, Y. B. Kim, and M. Choi, "A Novel "Divide and Conquer" Testing Technique for Memristor based Lookup Table," *Midwest Symposium on Circuits and Systems*, 2011, ISSN: 15483746. DOI: 10.1109/MWSCAS.2011.6026406.

[8] P. Liu, Z. You, J. Kuang, Z. Hu, and W. Wang, "Logic operation-based DFT method and 1R memristive crossbar March-like test Algorithm," *IEICE Electronics Express*, vol. 12, no. 23, Nov. 2015, ISSN: 13492543. DOI: 10.1587/ELEX.12.20150839.

[9] 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," *Electronics Letters*, vol. 53, no. 25, pp. 1631–1632, Dec. 2017, ISSN: 1350-911X. DOI: 10.1049/EL.2017.2424.

[10] S. Kannan, J. Rajendran, R. Karri, and O. Sinanoglu, "Sneak-Path Testing of Crossbar-Based Nonvolatile Random Access Memories," *IEEE Transactions on Nanotechnology*, vol. 12, no. 3, pp. 413–426, 2013, ISSN: 1536125X. DOI: 10.1109/TNANO.2013.2253329.

[11] T. Li, X. Bi, N. Jing, X. Liang, and L. Jiang, "Sneak-Path Based Test and Diagnosis for 1R RRAM Crossbar Using Voltage Bias Technique," *Proceedings - Design Automation Conference*, vol. Part 128280, Jun. 2017, ISSN: 0738100X. DOI: 10.1145/3061639.3062318.

[12] S. Hamdioui, M. Taouil, and N. Z. Haron, "Testing Open Defects in Memristor-Based Memories," *IEEE Transactions on Computers*, vol. 64, no. 1, pp. 247–259, Jan. 2015, ISSN: 00189340. DOI: 10.1109/TC.2013.206.

[13] N. Z. Haron and S. Hamdioui, "DfT Schemes for Resistive Open Defects in RRAMs," *Proceedings -Design, Automation and Test in Europe, DATE*, pp. 799–804, 2012, ISSN: 15301591. DOI: 10.1109/DATE.2012.6176603.

[14] S. N. Mozaffari, S. Tragoudas, and T. Haniotakis, "More Efficient Testing of Metal-Oxide Memristor-Based Memory," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 36, no. 6, pp. 1018–1029, Jun. 2017, ISSN: 02780070. DOI: 10.1109/TCAD.2016.2608863.

[15] T. S. Copetti, T. Gemmeke, and L. M. Bolzani Poehls, "Validating a DFT Strategy's Detection Capability regarding Emerging Faults in RRAMs," in *2021 IFIP/IEEE 29th International Conference on Very Large Scale Integration (VLSI-SoC)*, IEEE, Oct. 2021, pp. 1–6, ISBN: 978-1-6654-2614-5. DOI: 10.1109/VLSI-SoC53125.2021.9606993.

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

[17] M. Fieback, G. C. Medeiros, A. Gebregiorgis, H. Aziza, M. Taouil, and S. Hamdioui, "Intermittent Undefined State Fault in RRAMs," *Proceedings of the European Test Workshop*, vol. 2021-May, May 2021, ISSN: 15581780. DOI: 10.1109/ETS50041.2021.9465401.

[18] L. O. Chua, "Memristor—The Missing Circuit Element," *IEEE Transactions on Circuit Theory*, vol. 18, no. 5, pp. 507–519, 1971, ISSN: 00189324. DOI: 10.1109/TCT.1971.1083337.

[19] D. B. Strukov, G. S. Snider, D. R. Stewart, and R. S. Williams, "The missing memristor found," *Nature 2008 453:7191*, vol. 453, no. 7191, pp. 80–83, May 2008, ISSN: 1476-4687. DOI: 10.1038/NATURE06932.

[20] H. Ishiwara, "Ferroelectric Random Access Memories," *Journal of Nanoscience and Nanotechnology*, vol. 12, no. 10, pp. 7619–7627, Oct. 2012, ISSN: 1533-4880. DOI: 10.1166/jnn.2012.6651.

[21] D. Takashima, "Overview of FeRAMs: Trends and Perspectives," *2011 11th Annual Non-Volatile Memory Technology Symposium, NVMTS 2011*, pp. 36–41, 2011. DOI: 10.1109/NVMTS.2011.6137107.

[22] A. V. Khvalkovskiy, D. Apalkov, S. Watts, *et al.*, "Basic principles of STT-MRAM cell operation in memory arrays," *Journal of Physics D: Applied Physics*, vol. 46, no. 7, p. 074 001, Jan. 2013, ISSN: 0022-3727. DOI: 10.1088/0022-3727/46/7/074001.

[23] H. S. Wong, S. Raoux, S. Kim, *et al.*, "Phase Change Memory," *Proceedings of the IEEE*, vol. 98, no. 12, pp. 2201–2227, 2010, ISSN: 00189219. DOI: 10.1109/JPROC.2010.2070050.

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

[25] F. Zahoor, T. Z. Azni Zulkifli, and F. A. Khanday, "Resistive Random Access Memory (RRAM): an Overview of Materials, Switching Mechanism, Performance, Multilevel Cell (mlc) Storage, Modeling, and Applications," *Nanoscale Research Letters 2020 15:1*, vol. 15, no. 1, pp. 1–26, Apr. 2020, ISSN: 1556-276X. DOI: 10.1186/S11671-020-03299-9.

[26] F. Nardi, S. Balatti, S. Larentis, D. C. Gilmer, and D. Ielmini, "Complementary Switching in Oxide-Based Bipolar Resistive-Switching Random Memory," *IEEE Transactions on Electron Devices*, vol. 60, no. 1, pp. 70–77, 2013, ISSN: 00189383. DOI: 10.1109/TED.2012.2226728.

[27] M. Fieback, G. Cardoso Medeiros, L. Wu, *et al.*, "Defects, Fault Modeling, and Test Development Framework for RRAMs," *ACM Journal on Emerging Technologies in Computing Systems*, vol. 18, no. 3, p. 52, Jul. 2022, ISSN: 1550-4832. DOI: 10.1145/3510851.

[28] O. Ginez, J. M. Portal, and C. Muller, "Design and Test Challenges in Resistive Switching RAM (ReRAM): An Electrical Model for Defect Injections," *Proceedings of the 14th IEEE European Test Symposium, ETS 2009*, pp. 61–66, 2009. DOI: 10.1109/ETS.2009.23.

[29] D. B. Strukov and R. S. Williams, "Exponential ionic drift: fast switching and low volatility of thin-film memristors," *Applied Physics A 2008 94:3*, vol. 94, no. 3, pp. 515–519, Nov. 2008, ISSN: 1432-0630. DOI: 10.1007/S00339-008-4975-3.

[30] M. D. Pickett, D. B. Strukov, J. L. Borghetti, *et al.*, "Switching dynamics in titanium dioxide memristive devices," *Journal of Applied Physics*, vol. 106, no. 7, p. 074 508, Oct. 2009, ISSN: 0021-8979. DOI: 10.1063/1.3236506.

[31] S. Kvatinsky, E. G. Friedman, A. Kolodny, and U. C. Weiser, "TEAM: ThrEshold Adaptive Memristor Model," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 60, no. 1, pp. 211–221, 2013, ISSN: 15498328. DOI: 10.1109/TCSI.2012.2215714.

[32] S. Kvatinsky, M. Ramadan, E. G. Friedman, and A. Kolodny, "VTEAM: A General Model for Voltage-Controlled Memristors," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 62, no. 8, pp. 786–790, Aug. 2015, ISSN: 15497747. DOI: 10.1109/TCSII.2015.2433536.

[33] Z. Jiang and H.-S. P. Wong, "Stanford University Resistive-Switching Random Access Memory (RRAM) Verilog-A Model," Oct. 2014. DOI: 10.4231/D37H1DN48.

[34] Y. V. Pershin and M. Di Ventra, "SPICE Model of Memristive Devices with Threshold," *Radioengineering*, vol. 22, no. 2, pp. 485–489, Apr. 2012, ISSN: 12102512.

[35] M. Bocquet, H. Aziza, W. Zhao, *et al.*, "Compact Modeling Solutions for Oxide-Based Resistive Switching Memories (OxRAM)," *Journal of Low Power Electronics and Applications*, vol. 4, no. 1, pp. 1–14, Jan. 2014, ISSN: 20799268. DOI: 10.3390/JLPEA4010001.

[36] F. Cüppers, S. Menzel, C. Bengel, *et al.*, "Exploiting the switching dynamics of HfO2-based ReRAM devices for reliable analog memristive behavior," *APL Materials*, vol. 7, no. 9, p. 091 105, Sep. 2019, ISSN: 2166532X. DOI: 10.1063/1.5108654.

[37] 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.

[38] K. K. Likharev and D. B. Strukov, "CMOL: Devices, Circuits, and Architectures," *Lecture Notes in Physics*, vol. 680, pp. 447–477, 2006, ISSN: 00758450. DOI: 10.1007/3-540-31514-4{_}17.

[39] P. Liu, J. Wu, Z. You, M. Elimu, W. Wang, and S. Cai, "Defect Analysis and Parallel March Test Algorithm for 3D Hybrid CMOS-Memristor Memory," *Proceedings of the Asian Test Symposium*, vol. 2018-October, pp. 25–29, Dec. 2018, ISSN: 10817735. DOI: 10.1109/ATS.2018.00016.

[40] C. Xu, X. Dong, N. P. Jouppi, and Y. Xie, "Design Implications of Memristor-Based RRAM Cross-Point Structures," *Proceedings -Design, Automation and Test in Europe, DATE*, pp. 734–739, 2011, ISSN: 15301591. DOI: 10.1109/DATE.2011.5763125.

[41] R. J. Baker, *CMOS: Circuit Design, Layout, and Simulation*, Third Edition. Hoboken, NJ, USA: John Wiley & Sons, Inc., Aug. 2010, ISBN: 9780470891179. DOI: 10.1002/9780470891179.

[42] S. Hamdioui, *Testing Static Random Access Memories*, ser. Frontiers in Electronic Testing. Boston, MA: Springer US, 2004, vol. 26, ISBN: 978-1-4419-5430-5. DOI: 10.1007/978-1-4757-6706-3.

[43] L. M. Poehls, M. C. Fieback, S. Hoffmann-Eifert, *et al.*, "Review of Manufacturing Process Defects and Their Effects on Memristive Devices," *Journal of Electronic Testing: Theory and Applications (JETTA)*, vol. 37, no. 4, pp. 427–437, Aug. 2021, ISSN: 15730727. DOI: 10.1007/S10836-021-05968-8.

[44] K. J. Kuhn, M. D. Giles, D. Becher, *et al.*, "Process Technology Variation," *IEEE Transactions on Electron Devices*, vol. 58, no. 8, pp. 2197–2208, Aug. 2011, ISSN: 00189383. DOI: 10.1109/TED.2011.2121913.

[45] M. Fieback, L. Wu, G. C. Medeiros, *et al.*, "Device-Aware Test: A New Test Approach Towards DPPB Level," *Proceedings - International Test Conference*, vol. 2019-November, Nov. 2019, ISSN: 10893539. DOI: 10.1109/ITC44170.2019.9000134.

[46] A. J. van de Goor, "Using March Tests to Test SRAMs," *IEEE Design and Test of Computers*, vol. 10, no. 1, pp. 8–14, 1993, ISSN: 07407475. DOI: 10.1109/54.199799.

[47] C. Y. Chen, H. C. Shih, C. W. Wu, *et al.*, "RRAM Defect Modeling and Failure Analysis Based on March Test and a Novel Squeeze-Search Scheme," *IEEE Transactions on Computers*, vol. 64, no. 1, pp. 180–190, Jan. 2015, ISSN: 00189340. DOI: 10.1109/TC.2014.12.

[48] Y. X. Chen and J. F. Li, "Fault modeling and testing of 1T1R memristor memories," *Proceedings of the IEEE VLSI Test Symposium*, vol. 2015-January, Jun. 2015. DOI: 10.1109/VTS.2015.7116247.

[49] 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," *Electronics Letters*, vol. 52, no. 18, pp. 1520–1522, Sep. 2016, ISSN: 00135194. DOI: 10.1049/EL.2016.1693.

[50] Y. Luo, X. Cui, M. Luo, and Q. Lin, "A High Fault Coverage March Test for 1T1R Memristor Array," *EDSSC 2017 - 13th IEEE International Conference on Electron Devices and Solid-State Circuits*, vol. 2017-January, pp. 1–2, Dec. 2017. DOI: 10.1109/EDSSC.2017.8126415.

[51] S. Kvatinsky, D. Belousov, S. Liman, *et al.*, "MAGIC - Memristor-Aided Logic," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 61, no. 11, pp. 895–899, Nov. 2014, ISSN: 15497747. DOI: 10.1109/TCSII.2014.2357292.

[52] A. F. Gomez, F. Forero, K. Roy, and V. Champac, "Robust Detection of Bridge Defects in STT-MRAM Cells under Process Variations," *IEEE/IFIP International Conference on VLSI and System-on-Chip, VLSI-SoC*, vol. 2018-October, pp. 65–70, Feb. 2019, ISSN: 23248440. DOI: 10.1109/VLSI-SOC.2018.8645022.

[53] H. Aziza, M. Moreau, M. Fieback, M. Taouil, and S. Hamdioui, "An Energy-Efficient Current-Controlled Write and Read Scheme for Resistive RAMs (RRAMs)," *IEEE Access*, vol. 8, pp. 137 263–137 274, 2020, ISSN: 21693536. DOI: 10.1109/ACCESS.2020.3011647.

[54] D. A. Neamen, *Semiconductor Physics and Devices: Basic Principles*, 4th ed. Mcgraw Hill Higher Education, Jan. 2011, ISBN: 9780071089029.

[55] P. R. Gray, P. J. Hurst, S. H. Lewis, and R. G. Meyer, *Analysis and Design of Analog Intergrated Circuits*, 5th ed. New York: Wiley, Jan. 2009, ISBN: 978-0-470-24599-6.

[56] F. Bradarić, *PRR DFT GitLab Repository.* [Online]. Available: `https://gitlab.tudelft.nl/mcrfieback/dft-for-rram`.

[57] I. P. Tolić, J. Mikulić, G. Schatzberger, and A. Barić, "Design of Sense Amplifiers for Non-Volatile Memory," pp. 59–64, Jul. 2019. DOI: `10.23919/MIPRO.2019.8757026`.

[58] B. J. Blalock and P. E. Allen, "Low-voltage, bulk-driven MOSFET current mirror for CMOS technology," *Proceedings - IEEE International Symposium on Circuits and Systems*, vol. 3, pp. 1972–1975, 1995, ISSN: 02714310. DOI: `10.1109/ISCAS.1995.523807`.

[59] M. Rakus, V. Stopjakova, and D. Arbet, "Comparison of gate-driven and bulk-driven current mirror topologies," *Formal Proceedings of the 2016 IEEE 19th International Symposium on Design and Diagnostics of Electronic Circuits and Systems, DDECS 2016*, May 2016. DOI: `10.1109/DDECS.2016.7482457`.

A. Defect detection results

In this appendix, Table A.1–A.17 contain detailed results of the defect detection validation. In total, there are 17 tables corresponding to 17 resistive defects. In every table, the results for a regular read circuit and the PRR DFT can be seen and compared. Although the validation is performed with 101 different defect strengths, the tables only contain at most two identical columns. For example, if for five consecutive defect strengths the columns are the same, only the first and last column are kept while the three columns in the middle are removed. In this way, the sizes of the tables are reduced drastically while not losing any information.

For every defect strength, a total of eight sensitisation sequences S are performed. To be more precise, the following sensitisation sequences are considered: $0r0$, $1r1$, $0w0r0$, $1w1r1$, $0w1r1$, $1w0r0$, $0r0r0$ and $1r1r1$. The output consists of two values F, which is the state of the cell, and R, which is the read output. In the tables, the following colour encoding is used:

- Green - state F correct and output R correct
- Yellow - state F correct while output R incorrect
- Blue - state F incorrect while output R correct
- Grey - state F incorrect and output R incorrect

When the cell is yellow or grey, the corresponding defect strength can be detected since the output R is incorrect. However, when the cell is green or blue, the corresponding defect strength cannot be detected since the output R is correct. More elaborate explanation of the results follows.

Rop_BL

Table A.1 shows the defect detection results when the Rop_BL defect is considered. Since this is an open defect, low resistance values will not results in an incorrect state nor output. However, this is the desired behaviour which confirms that the PRR DFT can handle process variations. Furthermore, by comparing the results of the regular read circuit and the PRR DFT, it can be concluded that the PRR DFT can detect more defect strengths. The higher detection capability is attributed to the ability of the PRR DFT to detect all five states instead of just the two logic states. For example, if the sensitisation sequence $0r0$ is considered, the regular read circuit is unable to detect the defect since the higher resistance is still classified as the logic 0 state. On the other hand, the PRR DFT detects the higher resistance and provides a deep 0 (L) at the output. A similar effect can be seen when the sensitisation sequence $1r1$ is considered. However, this time the regular read circuit will detect the defect at higher resistance values since it will result in the output changing from state 1 to state 0. Even though the regular read circuit can detect it, the PRR DFT offers detection at lower defect resistances since it can detect the defect as soon as it reaches the undefined state.

An interesting effect is observed when the sensitisation sequence $1w0r0$ is considered. At first, for the defect strength of 1.0 kΩ, the write 0 operation fails which results in the detection of the defect. However, as the defect resistance further increases, the additional resistance in series with the memristor compensates for the failed write 0 operation. This results in the cancellation of the two effects and the output being correct for a defect strength between 25.1 kΩ and 43.7 kΩ. As the defect strength further increases, the deep 0 (L) state will be provided at the output which can again be detected.

Rop_SL

Rop_SL has a similar effect as Rop_BL since it is also in series with the memristor. For this reason, Table A.2 shows a similar trend as Table A.1. A difference can be seen when looking at the sensitisation sequence $0r0$, in which the state of the cell turns into L for high defect resistance while only a read operation is performed. This is caused by the approach that is used to determine the state of the cell. The state of the cell is determined by taking the voltage over the memristor and dividing it with the current going through the memristor. However, for low voltages over the memristor, this results in less accurate state estimation since the memristor does not act as a linear element. Fortunately, the state estimation has no impact on the output R which is used to determine if the defect can be detected or not.

Rop_WL

Even though Rop_WL is not in series with the memristor, such as Rop_BL and Rop_SL, Table A.3 still shows the same trend as Table A.1 and Table A.2. This is because a resistive defect on the WL will cause the access transistor to conduct less, effectively acting as a resistor in series with the memristor.

Rbr_BL_int

In Table A.4, the effect of the Rbr_BL_int resistive defect can be seen. The regular read circuit is incapable of detecting this defect since it effectively shortens the drain and source of the access transistor for low resistance values. However, when the memristor is in the LRS (state 1), shorting the access transistor reduces the voltage drop over it resulting in a higher voltage drop over the memristor. This higher voltage drop will result in a higher current which is detected as the deep 1 state (H) by the PRR DFT.

Rbr_BL_SL

The effect of Rbr_BL_SL is similar to the effect of Rbr_BL_int. However, the defect now shorts the BL to the SL instead of only shorting the access transistor. In Table A.5, the effect of Rbr_BL_SL can be seen. For large resistance values, the defect acts as an open circuit and it does not affect the output. However, for small resistance values, it reduces the equivalent resistance value, forcing the output to a lower states.

Rbr_BL_WL

Since the WL voltage is higher than V_{dd}, introducing a resistive defect between the WL and BL will result in a higher voltage at the BL. If Rbr_BL_WL has a low resistance value, a higher current will flow through the memristor during a read operation, forcing the output to a lower state, as can be seen in Table A.6.

Rbr_SL_int

The effect of Rbr_SL_int is similar to that of Rbr_BL_SL. The difference is that now only the memristor is being shorted for low resistance values. The result of this simulation can be seen in Table A.7.

Rbr_WL_int

In Table A.8, the effect of the Rbr_WL_int resistive defect can be seen. The same reasoning behind the results of Rbr_BL_WL apply to the results of Rbr_WL_int. However, the higher voltage is now directly applied to the TE of the memristor, resulting in a more profound effect.

Rbr_WL_SL

The effect of Rbr_WL_SL is similar to that of Rbr_WL_int and Rbr_BL_WL. The difference is that the higher voltage of the WL is now applied directly to the input of the read circuit, resulting in an even more profound effect than for Rbr_WL_int. The result of this simulation can be seen in Table A.9.

Rsh_BL_GND

Shorting the BL to GND will result in less current going through the memristor for low resistance values. This can be seen by looking at the results of Table A.10 in which the output is forced into lower states when the resistance of the defect decreases.

Rsh_BL_Vdd

Rsh_BL_Vdd has the inverse effect of Rsh_BL_GND, as can be seen in Table A.11. By shorting the BL to V_{dd}, more current goes through the memristor during a read operation which forces the output into higher states when the resistance of the defect decreases. Moreover, it will also result in a failed write 0 operation since the voltage over the memristor will be decreased.

Rsh_int_GND

In Table A.12, the effect of the Rsh_int_GND resistive defect can be seen. The effect of Rsh_int_GND is similar to the effect of Rsh_BL_GND. However, it is more profound since it is closer to the input of the read circuit.

Rsh_int_Vdd

Since Rsh_int_Vdd is providing V_{dd} to the TE of the memristor, it has a similar but larger effect than Rsh_BL_Vdd. The result of this simulation can be seen in Table A.13.

Rsh_SL_GND

Rsh_SL_GND has an even more profound effect than Rsh_int_GND since the defect is directly connected to the input of the read circuit, as can be seen in Table A.14.

Rsh_SL_Vdd

In Table A.15, the effect of the Rsh_SL_Vdd resistive defect can be seen. The effect of this defect is similar to Rsh_int_Vdd but more profound since it is directly connected to the input of the read circuit.

Rsh_WL_GND

Connecting the WL to GND will reduce the voltage at the WL and result in the reduced conduction of the access transistor. The effect is similar to having a resistor in series with the memristor, as can be concluded from Table A.16.

Rsh_WL_Vdd

In Table A.17, the effect of Rsh_WL_Vdd can be seen. By connecting the WL to V_{dd}, the voltage of the WL is reduced, similar to Rsh_WL_GND. However, the reduction in voltage is less severe since it is limited to V_{dd} instead of GND. This is the reason why this defect is only detectable when a write operation is performed that tries to flip the state of the memristor.

Table A.1: Defect detection comparison between a regular read circuit and the PRR DFT for Rop_BL.

Strength [Ω]			100		912		1.0 k		1.2 k		8.3 k		9.5 k		11.0 k		12.6 k		14.5 k		16.6 k		19.1 k		21.9 k		25.1 k		43.7 k		50.1 k		100.0 M	
		S	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R
Regular	Rop_BL	0r0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		1r1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	0	1	0	1	0	1	0
		0w0r0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		1w1r1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	0	1	0	1	0	1	0
		0w1r1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	0	U	0	U	0	0	0	0	0	0	0	0	0
		1w0r0	0	0	0	0	U	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	0	1	0	1	0	1	0
		0r0r0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		1r1r1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	0	1	0	1	0	1	0
PRR DFT	Rop_BL	0r0	0	0	0	0	0	0	0	0	0	0	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L
		1r1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	U	1	U	1	U	1	U	1	0	1	0	1	L	1	L
		0w0r0	0	0	0	0	0	0	0	0	0	0	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L
		1w1r1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	U	1	U	1	U	1	U	1	0	1	0	1	L	1	L
		0w1r1	1	1	1	1	1	1	1	1	1	1	1	1	1	U	1	U	1	U	1	0	U	0	U	0	0	L	0	L	0	L	0	L
		1w0r0	0	0	0	0	U	U	1	1	1	1	1	1	1	1	1	1	1	U	1	U	1	U	1	U	1	0	1	0	1	L	1	L
		0r0r0	0	0	0	0	0	0	0	0	0	0	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L
		1r1r1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	U	1	U	1	U	1	U	1	0	1	0	1	L	1	L

Table A.2: Defect detection comparison between a regular read circuit and the PRR DFT for Rop_SL.

Strength [Ω]			100		1.0 k		1.2 k		1.4 k		1.6 k		2.8 k		3.2 k		3.6 k		4.2 k		4.8 k		8.3 k	
		S	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R
Regular	Rop_SL	0r0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		1r1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
		0w0r0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		1w1r1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
		0w1r1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	U	1	U	0	0	0	0	0
		1w0r0	0	0	0	0	U	0	U	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
		0r0r0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		1r1r1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
PRR DFT	Rop_SL	0r0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		1r1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
		0w0r0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		1w1r1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
		0w1r1	1	1	1	1	1	1	1	1	1	1	1	1	1	U	U	U	U	0	0	0	0	0
		1w0r0	0	0	0	0	U	0	U	U	1	1	1	1	1	1	1	1	1	1	1	1	1	1
		0r0r0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		1r1r1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Strength [Ω]			9.5 k		11.0 k		12.6 k		14.5 k		19.1 k		21.9 k		25.1 k		28.8 k		50.1 k		57.5 k		66.1 k		100.0 M	
		S	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R
Regular	Rop_SL	0r0	0	0	0	0	0	0	0	0	0	0	0	0	L	0	L	0	L	0	L	0	L	0	L	0
		1r1	1	1	1	1	1	1	U	1	U	1	U	0	U	0	0	0	0	0	0	0	L	0	L	0
		0w0r0	0	0	0	0	0	0	0	0	0	0	0	0	L	0	L	0	L	0	L	0	L	0	L	0
		1w1r1	1	1	1	1	1	1	U	1	U	1	U	0	U	0	0	0	0	0	0	0	L	0	L	0
		0w1r1	0	0	0	0	0	0	0	0	0	0	0	0	L	0	L	0	L	0	L	0	L	0	L	0
		1w0r0	1	1	1	1	1	1	U	1	U	1	U	0	U	0	0	0	0	0	0	0	L	0	L	0
		0r0r0	0	0	0	0	0	0	0	0	0	0	0	0	L	0	L	0	L	0	L	0	L	0	L	0
		1r1r1	1	1	1	1	1	1	U	1	U	1	U	0	U	0	0	0	0	0	0	0	L	0	L	0
PRR DFT	Rop_SL	0r0	0	L	0	L	0	L	0	L	0	L	0	L	L	L	L	L	L	L	L	L	L	L	L	L
		1r1	1	1	1	1	1	1	U	U	U	U	U	U	U	0	0	0	0	0	0	L	L	L	L	L
		0w0r0	0	L	0	L	0	L	0	L	0	L	0	L	L	L	L	L	L	L	L	L	L	L	L	L
		1w1r1	1	1	1	1	1	1	U	U	U	U	U	U	U	0	0	0	0	0	0	L	L	L	L	L
		0w1r1	0	0	0	L	0	L	0	L	0	L	0	L	L	L	L	L	L	L	L	L	L	L	L	L
		1w0r0	1	1	1	1	1	1	U	U	U	U	U	U	U	0	0	0	0	0	0	L	L	L	L	L
		0r0r0	0	L	0	L	0	L	0	L	0	L	0	L	L	L	L	L	L	L	L	L	L	L	L	L
		1r1r1	1	1	1	1	1	1	U	U	U	U	U	U	U	U	0	0	0	0	0	L	L	L	L	L

Table A.3: Defect detection comparison between a regular read circuit and the PRR DFT for Rop_WL.

Strength [Ω]			100		57.5 k		66.1 k		75.9 k		100.0 k		114.8 k		199.5 k		229.1 k		263.0 k		302.0 k		8.3 M		9.5 M	
		S	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R
Regular	Rop_WL	0r0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		1r1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	0	1	0
		0w0r0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		1w1r1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	0	1	0
		0w1r1	1	1	1	1	U	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		1w0r0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		0r0r0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		1r1r1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	0	1	0
PRR DFT	Rop_WL	0r0	0	0	0	0	0	0	0	0	0	0	0	L	0	L	0	L	0	L	0	L	0	L	0	L
		1r1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	U	1	0	1	L	1	L	1	L
		0w0r0	0	0	0	0	0	0	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L
		1w1r1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	U	1	0	1	L	1	L	1	L
		0w1r1	1	1	1	1	U	U	0	0	0	0	0	L	0	L	0	L	0	L	0	L	0	L	0	L
		1w0r0	0	0	0	0	0	0	0	0	0	0	0	L	0	L	0	L	0	L	0	L	0	L	0	0
		0r0r0	0	0	0	0	0	0	0	0	0	0	0	L	0	L	0	L	0	L	0	L	0	L	0	L
		1r1r1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	U	1	0	1	L	1	L	1	L

Strength [Ω]			11.0 M		19.1 M		21.9 M		25.1 M		43.7 M		50.1 M		57.5 M		66.1 M		75.9 M		87.1 M		100.0 M	
		S	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R
Regular	Rop_WL	0r0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		1r1	U	0	U	0	U	0	U	0	U	0	U	0	U	0	U	0	U	0	0	0	0	0
		0w0r0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		1w1r1	U	0	U	0	U	0	U	0	U	0	U	0	U	0	U	0	U	0	0	0	0	0
		0w1r1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		1w0r0	0	0	0	0	U	0	1	1	1	1	1	1	1	1	1	0	1	0	1	0	1	0
		0r0r0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		1r1r1	U	0	U	0	U	0	U	0	U	0	U	0	U	0	U	0	U	0	U	0	0	0
PRR DFT	Rop_WL	0r0	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L
		1r1	U	L	U	L	U	L	U	L	U	L	U	L	U	L	U	L	U	L	0	L	0	L
		0w0r0	0	0	0	0	0	0	0	0	0	0	0	L	0	L	0	L	0	L	0	L	0	L
		1w1r1	U	L	U	L	U	L	U	L	U	L	U	L	U	L	U	L	U	L	0	L	0	L
		0w1r1	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L
		1w0r0	0	0	0	0	U	0	1	1	1	1	1	1	1	U	1	0	1	L	1	L	1	L
		0r0r0	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L
		1r1r1	U	L	U	L	U	L	U	L	U	L	U	L	U	L	U	L	U	L	U	L	0	L

Table A.4: Defect detection comparison between a regular read circuit and the PRR DFT for Rbr_BL_int.

Strength [Ω]			100		2.8 k		3.2 k		12.6 k		14.5 k		100.0 M	
		S	F	R	F	R	F	R	F	R	F	R	F	R
Regular	Rbr_BL_int	0r0	0	0	0	0	0	0	0	0	0	0	0	0
		1r1	1	1	1	1	1	1	1	1	1	1	1	1
		0w0r0	0	0	0	0	0	0	0	0	0	0	0	0
		1w1r1	1	1	1	1	1	1	1	1	1	1	1	1
		0w1r1	1	1	1	1	1	1	1	1	1	1	1	1
		1w0r0	0	0	0	0	0	0	0	0	0	0	0	0
		0r0r0	0	0	0	0	0	0	0	0	0	0	0	0
		1r1r1	1	1	1	1	1	1	1	1	1	1	1	1
PRR DFT	Rbr_BL_int	0r0	0	0	0	0	0	0	0	0	0	0	0	0
		1r1	1	H	1	H	1	1	1	1	1	1	1	1
		0w0r0	0	0	0	0	0	0	0	0	0	0	0	0
		1w1r1	1	H	1	H	1	H	1	H	1	1	1	1
		0w1r1	1	H	1	H	1	H	1	H	1	1	1	1
		1w0r0	0	0	0	0	0	0	0	0	0	0	0	0
		0r0r0	0	0	0	0	0	0	0	0	0	0	0	0
		1r1r1	1	H	1	H	1	1	1	1	1	1	1	1

Table A.5: Defect detection comparison between a regular read circuit and the PRR DFT for Rbr_BL_SL.

Strength [Ω]		100		4.2 k		4.8 k		5.5 k		6.3 k		7.2 k		8.3 k		14.5 k		16.6 k		19.1 k		38.0 k		43.7 k		57.5 k		66.1 k		75.9 k		87.1 k		100.0 k		114.8 k		100.0 M	
	S	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R
Regular Rbr_BL_SL	0r0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0
	1r1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	0w0r0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0
	1w1r1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	0w1r1	0	1	0	1	U	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	1w0r0	1	1	1	1	U	1	U	1	U	1	U	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0
	0r0r0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0
	1r1r1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
PRR DFT Rbr_BL_SL	0r0	0	H	0	H	0	H	0	H	0	H	0	1	0	1	0	1	0	1	0	1	0	1	0	U	0	U	0	U	0	U	0	U	0	U	0	0	0	0
	1r1	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	0w0r0	0	H	0	H	0	H	0	H	0	H	0	1	0	1	0	1	0	1	0	1	0	1	0	U	0	U	0	U	0	U	0	U	0	0	0	0	0	0
	1w1r1	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	0w1r1	0	H	0	H	U	H	1	H	1	H	1	H	1	H	1	H	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	1w0r0	1	H	1	H	U	H	U	H	U	H	U	H	0	1	0	1	0	1	0	1	0	1	0	U	0	U	0	U	0	U	0	U	0	U	0	0	0	0
	0r0r0	0	H	0	H	0	H	0	H	0	H	0	1	0	1	0	1	0	1	0	1	0	1	0	U	0	U	0	U	0	U	0	U	0	U	0	0	0	0
	1r1r1	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Table A.6: Defect detection comparison between a regular read circuit and the PRR DFT for Rbr_BL_WL.

Strength [Ω]		100		457.1		524.8		602.6		2.1 k		2.4 k		2.8 k		3.2 k		3.6 k		4.2 k		4.8 k		5.5 k		6.3 k		11.0 k		12.6 k		14.5 k		16.6 k		100.0 M	
	S	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R
Regular Rbr_BL_WL	0r0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	1r1	1	0	1	0	1	0	1	0	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	0w0r0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	1w1r1	1	0	1	0	1	0	1	0	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	0w1r1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	U	0	U	0	U	1	1	1	1	1	1	1	1	1	1	1	1	1
	1w0r0	1	0	1	0	1	0	1	0	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	U	1	U	0	0	0	0	0
	0r0r0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	1r1r1	1	0	1	0	1	0	1	0	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
PRR DFT Rbr_BL_WL	0r0	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	1r1	1	L	1	L	1	L	1	0	1	0	1	U	1	U	1	U	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	0w0r0	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	1w1r1	1	L	1	L	1	L	1	0	1	0	1	U	1	U	1	U	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	0w1r1	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	0	U	0	U	0	U	U	1	1	1	1	1	1	1	1	1	1	1	1
	1w0r0	1	L	1	L	1	L	1	0	1	0	1	U	1	U	1	U	1	1	1	1	1	1	1	1	1	1	1	1	U	U	U	U	0	0	0	0
	0r0r0	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	1r1r1	1	L	1	L	1	0	1	0	1	0	1	U	1	U	1	U	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Table A.7: Defect detection comparison between a regular read circuit and the PRR DFT for Rbr_SL_int.

Strength [Ω]			100		1.2 k		1.4 k		1.6 k		1.8 k		16.6 k		19.1 k		25.1 k		28.8 k		33.1 k		38.0 k		43.7 k		50.1 k		57.5 k		66.1 k		75.9 k		87.1 k		100.0 k		114.8 k		100.0 M	
		S	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R
Regular	Rbr_SL_int	0r0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		1r1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
		0w0r0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		1w1r1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
		0w1r1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	U	1	1	1	1	1	1	1	1	1	1	1	1	1
		1w0r0	1	1	1	1	1	1	1	1	1	1	1	1	U	1	U	1	U	1	U	1	0	1	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0
		0r0r0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		1r1r1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
PRR DFT	Rbr_SL_int	0r0	0	H	0	H	0	1	0	1	0	1	0	1	0	1	0	1	0	U	0	U	0	U	0	U	0	U	0	U	0	U	0	U	0	0	0	0	0	0	0	0
		1r1	1	H	1	H	1	H	1	H	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
		0w0r0	0	H	0	H	0	1	0	1	0	1	0	1	0	1	0	1	0	U	0	U	0	U	0	U	0	U	0	U	0	U	0	0	0	0	0	0	0	0	0	0
		1w1r1	1	H	1	H	1	H	1	H	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
		0w1r1	0	H	0	H	0	1	0	1	0	1	0	1	0	1	0	1	0	U	0	U	0	U	0	U	0	U	U	U	1	1	1	1	1	1	1	1	1	1	1	1
		1w0r0	1	H	1	H	1	H	1	H	1	1	1	1	U	1	U	1	U	1	U	1	0	U	0	U	0	U	0	U	0	U	0	U	0	U	0	U	0	0	0	0
		0r0r0	0	H	0	H	0	1	0	1	0	1	0	1	0	1	0	1	0	U	0	U	0	U	0	U	0	U	0	U	0	U	0	U	0	0	0	0	0	0	0	0
		1r1r1	1	H	1	H	1	H	1	H	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Table A.8: Defect detection comparison between a regular read circuit and the PRR DFT for Rbr_WL_int.

Strength [Ω]			100		9.5 k		11.0 k		12.6 k		14.5 k		33.1 k		38.0 k		43.7 k		50.1 k		57.5 k		66.1 k		75.9 k		87.1 k		100.0 k		114.8 k		131.8 k		151.4 k		100.0 M	
		S	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R
Regular	Rbr_WL_int	0r0	1	1	1	1	U	1	U	1	0	1	0	1	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		1r1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
		0w0r0	1	1	1	1	U	1	U	1	0	1	0	1	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		1w1r1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
		0w1r1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
		1w0r0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	U	1	U	1	U	1	U	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0
		0r0r0	1	1	1	1	1	1	U	1	0	1	0	1	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		1r1r1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
PRR DFT	Rbr_WL_int	0r0	1	H	1	H	U	H	U	H	0	1	0	1	0	U	0	U	0	U	0	U	0	U	0	U	0	0	0	0	0	0	0	0	0	0	0	0
		1r1	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	1	1	1	1	1	1	1
		0w0r0	1	H	1	H	U	H	U	H	0	1	0	1	0	U	0	U	0	U	0	U	0	U	0	U	0	0	0	0	0	0	0	0	0	0	0	0
		1w1r1	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	1	1	1
		0w1r1	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	1	1	1	1	1
		1w0r0	1	H	1	H	1	H	1	H	1	H	1	H	1	1	U	1	U	1	U	1	U	U	0	U	0	U	0	U	0	U	0	0	0	0	0	0
		0r0r0	1	H	1	H	1	H	U	H	0	1	0	1	0	U	0	U	0	U	0	U	0	U	0	U	0	0	0	0	0	0	0	0	0	0	0	0
		1r1r1	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	1	1	1	1	1	1	1

Table A.9: Defect detection comparison between a regular read circuit and the PRR DFT for Rbr_WL_SL.

Strength [Ω]			100		1.6 k		1.8 k		2.1 k		43.7 k		50.1 k		57.5 k		66.1 k		75.9 k		87.1 k		173.8 k	
		S	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R
Regular	Rbr_WL_SL	0r0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
		1r1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
		0w0r0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
		1w1r1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
		0w1r1	0	1	0	1	0	1	0	1	0	1	U	1	U	1	1	1	1	1	1	1	1	1
		1w0r0	1	1	1	1	U	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
		0r0r0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
		1r1r1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
PRR DFT	Rbr_WL_SL	0r0	0	H	0	H	0	H	0	H	0	H	0	H	0	H	0	H	0	H	0	1	0	1
		1r1	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	H
		0w0r0	0	H	0	H	0	H	0	H	0	H	0	H	0	H	0	H	0	H	0	1	0	1
		1w1r1	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	H
		0w1r1	0	H	0	H	0	H	0	H	0	H	U	H	U	H	1	H	1	H	1	H	1	H
		1w0r0	1	H	1	H	U	H	0	H	0	H	0	H	0	H	0	H	0	H	0	1	0	1
		0r0r0	0	H	0	H	0	H	0	H	0	H	0	H	0	H	0	H	0	H	0	1	0	1
		1r1r1	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	H

Strength [Ω]			199.5 k		229.1 k		263.0 k		302.0 k		398.1 k		457.1 k		524.8 k		602.6 k		691.8 k		794.3 k		100.0 M	
		S	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R
Regular	Rbr_WL_SL	0r0	0	1	0	1	0	1	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0
		1r1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
		0w0r0	0	1	0	1	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0
		1w1r1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
		0w1r1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
		1w0r0	0	1	0	1	0	1	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0
		0r0r0	0	1	0	1	0	1	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0
		1r1r1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
PRR DFT	Rbr_WL_SL	0r0	0	1	0	1	0	1	0	U	0	U	0	U	0	U	0	U	0	U	0	0	0	0
		1r1	1	H	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
		0w0r0	0	1	0	1	0	1	0	U	0	U	0	U	0	U	0	0	0	0	0	0	0	0
		1w1r1	1	H	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
		0w1r1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
		1w0r0	0	1	0	1	0	1	0	U	0	U	0	U	0	U	0	U	0	U	0	0	0	0
		0r0r0	0	1	0	1	0	1	0	U	0	U	0	U	0	U	0	U	0	U	0	0	0	0
		1r1r1	1	H	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Table A.10: Defect detection comparison between a regular read circuit and the PRR DFT for Rsh_BL_GND.

Strength [Ω]			100		794.3		912		1.4 k		1.6 k		1.8 k		2.1 k		2.4 k		2.8 k		3.2 k		3.6 k		4.2 k		14.5 k		16.6 k		28.8 k		33.1 k		100.0 M	
		S	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R
Regular	Rsh_BL_GND	0r0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		1r1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
		0w0r0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		1w1r1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
		0w1r1	0	0	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
		1w0r0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		0r0r0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		1r1r1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
PRR DFT	Rsh_BL_GND	0r0	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	0	0	0	0	0	0	0
		1r1	1	L	1	L	1	L	1	L	1	L	1	0	1	0	1	0	1	U	1	U	1	U	1	1	1	1	1	1	1	1	1	1	1	1
		0w0r0	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	0	0	0
		1w1r1	1	L	1	L	1	L	1	L	1	0	1	0	1	0	1	U	1	U	1	U	1	U	1	1	1	1	1	1	1	1	1	1	1	1
		0w1r1	0	L	0	L	1	L	1	L	1	L	1	0	1	0	1	0	1	U	1	U	1	U	1	1	1	1	1	1	1	1	1	1	1	1
		1w0r0	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	0	0	0	0	0	0	0
		0r0r0	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	0	0	0	0	0	0	0
		1r1r1	1	L	1	L	1	L	1	L	1	L	1	0	1	0	1	0	1	U	1	U	1	U	1	1	1	1	1	1	1	1	1	1	1	1

Table A.11: Defect detection comparison between a regular read circuit and the PRR DFT for Rsh_BL_Vdd.

Strength [Ω]			100		2.4 k		2.8 k		3.2 k		3.6 k		4.2 k		5.5 k		6.3 k		7.2 k		8.3 k		9.5 k		11.0 k		12.6 k		100.0 M	
		S	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R
Regular	Rsh_BL_Vdd	0r0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		1r1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
		0w0r0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		1w1r1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
		0w1r1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
		1w0r0	1	1	1	1	1	1	U	1	U	1	U	1	U	1	0	1	0	1	0	1	0	0	0	0	0	0	0	0
		0r0r0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		1r1r1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
PRR DFT	Rsh_BL_Vdd	0r0	0	1	0	1	0	1	0	1	0	1	0	U	0	U	0	U	0	U	0	0	0	0	0	0	0	0	0	0
		1r1	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	1	1	1	1	1
		0w0r0	0	1	0	1	0	1	0	1	0	1	0	U	0	U	0	U	0	U	0	0	0	0	0	0	0	0	0	0
		1w1r1	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	1	1	1
		0w1r1	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	1	1	1	1	1
		1w0r0	1	H	1	H	1	1	U	1	U	1	U	1	U	1	0	U	0	U	0	U	0	U	0	U	0	0	0	0
		0r0r0	0	1	0	1	0	1	0	1	0	1	0	U	0	U	0	U	0	U	0	0	0	0	0	0	0	0	0	0
		1r1r1	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	1	1	1	1	1

Table A.12: Defect detection comparison between a regular read circuit and the PRR DFT for Rsh_int_GND.

Strength [Ω]			100		346.7		398.1		457.1		1.0 k		1.2 k		2.1 k		2.4 k		4.2 k		4.8 k		14.5 k		16.6 k		21.9 k		25.1 k		28.8 k		33.1 k		38.0 k		50.1 k		57.5 k		66.1 k		75.9 k		114.8 k		131.8 k		229.1 k		263.0 k		100.0 M	
		S	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R
Regular	Rsh_int_GND	0r0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		1r1	U	0	U	0	U	0	U	0	U	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
		0w0r0	0	0	0	0	0	0	L	0	L	0	L	0	L	0	L	0	L	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		1w1r1	U	0	U	0	U	0	U	0	U	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
		0w1r1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	U	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
		1w0r0	0	0	0	0	L	0	L	0	L	0	L	0	L	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		0r0r0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		1r1r1	U	0	U	0	U	0	U	0	U	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
PRR DFT	Rsh_int_GND	0r0	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	0	0	0	0	0	0	0
		1r1	U	L	U	L	U	L	U	L	U	L	1	L	1	L	1	L	1	L	1	L	1	L	1	0	1	0	1	U	1	U	1	U	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
		0w0r0	0	L	0	L	0	L	L	L	L	L	L	L	L	L	L	L	L	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	0	0	0
		1w1r1	U	L	U	L	U	L	U	L	U	L	1	L	1	L	1	L	1	L	1	L	1	L	1	0	1	0	1	U	1	U	1	U	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
		0w1r1	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	U	0	1	U	1	1	1	1	1	1	1	1	1	1	1	1
		1w0r0	0	L	0	L	L	L	L	L	L	L	L	L	L	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	0	0	0	0	0	0	0
		0r0r0	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	0	0	0	0	0	0	0
		1r1r1	U	L	U	L	U	L	U	L	U	L	1	L	1	L	1	L	1	L	1	L	1	L	1	0	1	0	1	U	1	U	1	U	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Table A.13: Defect detection comparison between a regular read circuit and the PRR DFT for Rsh_int_Vdd.

Strength [Ω]			100		21.9 k		25.1 k		28.8 k		33.1 k		38.0 k		43.7 k		50.1 k		57.5 k		66.1 k		75.9 k		87.1 k		100.0 k		114.8 k		100.0 M	
		S	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R
Regular	Rsh_int_Vdd	0r0	1	1	1	1	1	1	1	1	1	1	U	1	U	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		1r1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
		0w0r0	1	1	1	1	1	1	1	1	U	1	U	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		1w1r1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
		0w1r1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
		1w0r0	1	1	1	1	1	1	1	1	U	1	U	1	U	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0
		0r0r0	1	1	1	1	1	1	1	1	1	1	U	1	U	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		1r1r1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
PRR DFT	Rsh_int_Vdd	0r0	1	H	1	H	1	H	1	1	1	1	U	1	U	U	0	U	0	U	0	0	0	0	0	0	0	0	0	0	0	0
		1r1	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	1	1	1	1	1
		0w0r0	1	H	1	H	1	1	1	1	U	1	U	1	0	U	0	U	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		1w1r1	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	1	1	1
		0w1r1	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	1	1	1
		1w0r0	1	H	1	H	1	1	1	1	U	1	U	1	U	U	0	U	0	U	0	U	0	U	0	0	0	0	0	0	0	0
		0r0r0	1	H	1	H	1	H	1	1	1	1	U	1	U	1	0	U	0	U	0	0	0	0	0	0	0	0	0	0	0	0
		1r1r1	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	1	1	1	1	1

Table A.14: Defect detection comparison between a regular read circuit and the PRR DFT for Rsh_SL_GND.

Strength [Ω]			100		2.4 k		2.8 k		6.3 k		7.2 k		25.1 k		28.8 k		33.1 k		38.0 k		43.7 k		50.1 k		57.5 k		66.1 k		691.8 k		794.3 k		1.4 M		1.6 M		100.0 M	
		S	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R
Regular	Rsh_SL_GND	0r0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		1r1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
		0w0r0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		1w1r1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
		0w1r1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
		1w0r0	1	0	1	0	U	0	U	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		0r0r0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		1r1r1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
PRR DFT	Rsh_SL_GND	0r0	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	0	0	0	0	0	0	0
		1r1	1	L	1	L	1	L	1	L	1	L	1	L	1	L	1	0	1	0	1	U	1	U	1	U	1	1	1	1	1	1	1	1	1	1	1	1
		0w0r0	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	0	0	0
		1w1r1	1	L	1	L	1	L	1	L	1	L	1	L	1	0	1	0	1	0	1	U	1	U	1	1	1	1	1	1	1	1	1	1	1	1	1	1
		0w1r1	1	L	1	L	1	L	1	L	1	L	1	L	1	L	1	0	1	0	1	U	1	U	1	U	1	1	1	1	1	1	1	1	1	1	1	1
		1w0r0	1	L	1	L	U	L	U	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	0	0	0	0	0	0	0
		0r0r0	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	0	0	0	0	0	0	0
		1r1r1	1	L	1	L	1	L	1	L	1	L	1	L	1	L	1	0	1	0	1	U	1	U	1	U	1	1	1	1	1	1	1	1	1	1	1	1

Table A.15: Defect detection comparison between a regular read circuit and the PRR DFT for Rsh_SL_Vdd.

Strength [Ω]			100		2.1 k		2.4 k		4.8 k		5.5 k		57.5 k		66.1 k		151.4 k		173.8 k		199.5 k		229.1 k		302.0 k		346.7 k		398.1 k		457.1 k		524.8 k		602.6 k		100.0 M	
		S	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R
Regular	Rsh_SL_Vdd	0r0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0
		1r1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
		0w0r0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0
		1w1r1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
		0w1r1	0	1	0	1	U	1	U	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
		1w0r0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0
		0r0r0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0
		1r1r1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
PRR DFT	Rsh_SL_Vdd	0r0	0	H	0	H	0	H	0	H	0	H	0	H	0	1	0	1	0	1	0	1	0	U	0	U	0	U	0	U	0	U	0	U	0	0	0	0
		1r1	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
		0w0r0	0	H	0	H	0	H	0	H	0	H	0	H	0	1	0	1	0	1	0	1	0	U	0	U	0	U	0	U	0	U	0	0	0	0	0	0
		1w1r1	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
		0w1r1	0	H	0	H	U	H	U	H	1	H	1	H	1	H	1	H	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
		1w0r0	0	H	0	H	0	H	0	H	0	H	0	H	0	1	0	1	0	1	0	1	0	U	0	U	0	U	0	U	0	U	0	U	0	0	0	0
		0r0r0	0	H	0	H	0	H	0	H	0	H	0	H	0	1	0	1	0	1	0	1	0	U	0	U	0	U	0	U	0	U	0	U	0	0	0	0
		1r1r1	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	H	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Table A.16: Defect detection comparison between a regular read circuit and the PRR DFT for Rsh_WL_GND.

Strength [Ω]			100		457.1		524.8		794.3		912		7.2 k		8.3 k		9.5 k		11.0 k		12.6 k		14.5 k		16.6 k		19.1 k		21.9 k		38.0 k		43.7 k		50.1 k		57.5 k		100.0 M	
		S	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R
Regular	Rsh_WL_GND	0r0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		1r1	0	0	0	0	U	0	U	0	1	0	1	0	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
		0w0r0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		1w1r1	0	0	0	0	U	0	U	0	1	0	1	0	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
		0w1r1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	U	0	U	1	1	1	1	1
		1w0r0	0	0	0	0	U	0	U	0	1	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		0r0r0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		1r1r1	0	0	0	0	U	0	U	0	1	0	1	0	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
PRR DFT	Rsh_WL_GND	0r0	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		1r1	0	L	0	L	U	L	U	L	1	L	1	L	1	0	1	0	1	U	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
		0w0r0	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	0	0	0	0	0	0	0	0	0	0	0
		1w1r1	0	L	0	L	U	L	U	L	1	L	1	L	1	0	1	0	1	U	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
		0w1r1	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	0	0	0	0	0	0	0	U	0	U	U	1	1	1	1
		1w0r0	0	L	0	L	U	L	U	L	1	L	1	L	1	0	1	0	0	L	0	L	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		0r0r0	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	L	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		1r1r1	0	L	0	L	U	L	U	L	1	L	1	L	1	0	1	0	1	U	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Table A.17: Defect detection comparison between a regular read circuit and the PRR DFT for Rsh_WL_Vdd.

Strength [Ω]			100		2.1 k		2.4 k		4.2 k		4.8 k		5.5 k		6.3 k		100.0 M	
		S	F	R	F	R	F	R	F	R	F	R	F	R	F	R	F	R
Regular	Rsh_WL_Vdd	0r0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		1r1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
		0w0r0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		1w1r1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
		0w1r1	0	0	0	0	0	0	0	0	U	0	U	1	1	1	1	1
		1w0r0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
		0r0r0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		1r1r1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
PRR DFT	Rsh_WL_Vdd	0r0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		1r1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
		0w0r0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		1w1r1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
		0w1r1	0	0	0	0	0	0	0	0	U	0	U	U	1	1	1	1
		1w0r0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
		0r0r0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		1r1r1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1