Defects, Fault Modeling, and Test Development Framework for RRAMs

Fieback, Moritz; Medeiros, Guilherme Cardoso; Wu, Lizhou; Aziza, Hassen; Bishnoi, Rajendra; Taouil, Mottaqiallah; Hamdioui, Said

DOI
10.1145/3510851

Publication date
2022

Document Version
Final published version

Published in
ACM Journal on Emerging Technologies in Computing Systems

Defects, Fault Modeling, and Test Development Framework for RRAMs

MORITZ FIEBACK, GUILHERME CARDOSO MEDEIROS, and LIZHOU WU,
Delft University of Technology
HASSEN AZIZA, Aix-Marseille University-IM2NP laboratory
RAJENDRA BISHNOI, MOTTAQIALLAH TAOUIL, and SAID HAMDIOUI,
Delft University of Technology

Resistive RAM (RRAM) is a promising technology to replace traditional technologies such as Flash, because of its low energy consumption, CMOS compatibility, and high density. Many companies are prototyping this technology to validate its potential. Bringing this technology to the market requires high-quality tests to ensure customer satisfaction. Hence, it is of great importance to deeply understand manufacturing defects and accurately model them to develop optimal tests. This paper presents a holistic framework for defect and fault modeling that enables the development of optimal tests for RRAMs. An overview and classification of RRAM manufacturing defects are provided. Defects in contacts and interconnects are modeled as resistors. Unique RRAM defects, e.g., forming defects, require Device-Aware defect modeling which incorporates the defect's impact on the device's electric properties by adjusting the affected technology and electrical parameters. Additionally, a systematic approach to define the fault space is presented, followed by a methodology to validate this space. With this methodology, accurate fault modeling for contact, interconnect, and forming defects is performed and tests are developed. The tests are able to detect all faults in a time-efficient manner, thereby proving the effectiveness of the framework. Finally, an outlook on future RRAM testing is presented.

CCS Concepts: • **Hardware → Memory test and repair**; **Defect-based test**; **Fault models and test metrics**; *Test-pattern generation and fault simulation;*

Additional Key Words and Phrases: RRAM, device-aware test, defect modeling, fault modeling, test development

ACM Reference format:
Moritz Fieback, Guilherme Cardoso Medeiros, Lizhou Wu, Hassen Aziza, Rajendra Bishnoi, Mottaqiallah Taouil, and Said Hamdioui. 2022. Defects, Fault Modeling, and Test Development Framework for RRAMs. *J. Emerg. Technol. Comput. Syst.* 18, 3, Article 52 (April 2022), 26 pages.
https://doi.org/10.1145/3510851

This research is supported by EC Horizon 2020 Research and Innovation Program through MNEMOSENE project under Grant 780215.

Authors' addresses: M. Fieback, G. C. Medeiros, L. Wu, R. Bishnoi, M. Taouil, and S. Hamdioui, Delft University of Technology, Mekelweg 4, 2628CD, Delft, The Netherlands; emails: M.C.R.Fieback@tudelft.nl, G.CardosoMedeiros@tudelft.nl, njuwulizhou@gmail.com, R.K.Bishnoi@tudelft.nl, m.taouil@tudelft.nl, s.hamdioui@tudelft.nl; H. Aziza, Aix-Marseille University-IM2NP laboratory, 05, Rue Enrico Fermi - Bât Fermi, Marseille, France; email: hassen.aziza@univ-amu.fr.

1550-4832/2022/04-ART52
https://doi.org/10.1145/3510851

1 INTRODUCTION

Flash and **dynamic random access memory (DRAM)** are reaching their scaling limits due to increased leakage currents, and structural complications [41, 57]. Among emerging memory technologies, **Resistive RAM (RRAM)** is a promising candidate to replace these two memory technologies. RRAMs are based on filamentary memristive devices and therefore are a non-volatile memory [63, 68]. This technology has several advantages: it does not suffer from high leakage power consumption, it can be densely fabricated in a standard CMOS process [63], it has a better cycle endurance and lower access latency than Flash [63, 68], and it allows for in-memory computing [25, 44]. Despite these benefits, this new technology also introduces new problems such as variations in the resistance states [29, 63, 65]. Furthermore, it suffers from new failure mechanisms that are not completely understood yet [9, 18]. Therefore, to enable large-volume production of RRAMs, new tests dedicated to this type of emerging technology need to be developed.

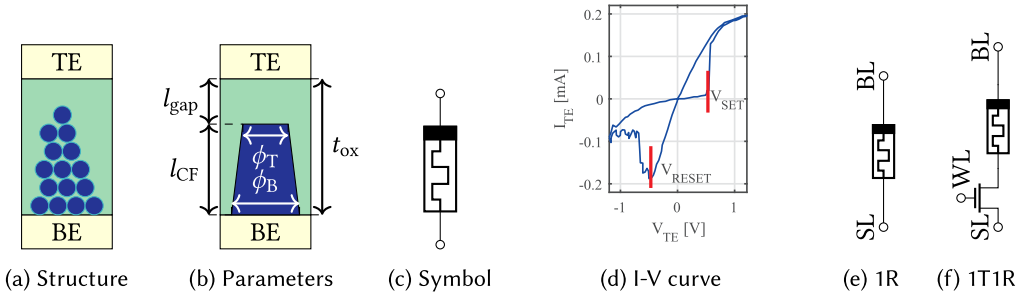
Traditionally, physical defects are modeled as opens or shorts, implemented using linear resistors [28, 34]. These models work well to describe defects in the interconnections of a circuit. The application of these models for a RRAM circuit has led to the identification of some unique faults, e.g., the Undefined Write Fault in which the resistance is set between the high and low states [28], the Undefined Read Fault in which reading a cell in such state results in random read outputs [34], and the Deep Fault in which the resistance is set beyond the high or low limits [34]. Several researchers have published tests and **design-for-test (DfT)** schemes to detect these unique faults, as well as other traditional memory faults [10, 13, 28, 34, 45, 48, 52, 59]. However, these *linear* resistive defect models do not describe the behavior of defects in the *non-linear* RRAM device properly, because they do not represent the actual physical behavior of the defect [19]. Hence, tests based only on these models will lead to escapes and thus low-quality test solutions.

This paper presents a holistic framework for RRAM test development. The first step in this framework is to analyze and model the *physical* behavior of RRAM forming defects using the **Device-Aware Test (DAT)** approach [5, 17, 19, 26], while modeling interconnect and contact defects as linear resistors. This results in defect models that accurately represent the electrical physical behavior. Second, these models are used to perform fault analysis that results in a set of *realistic* faults that may occur in the circuit. Third, a test is developed to detect these faults. This paper thus presents a *unified framework for RRAM test development* that consists of the following contributions:

- An overview of RRAM defects and classification.
- A systematic approach to define the complete fault space.
- A systematic approach to validate the fault space.
- The application of these approaches to perform defect analysis in RRAM that results in fault models and complete tests for interconnection and contact defects based on *traditional* defect modeling approaches.
- The application of these approaches to perform defect analysis in RRAM that results in fault models and complete tests for forming defects based on the *Device-Aware Test* approach.
- A comparison between these two defect modeling approaches.
- An outlook on the future of RRAM testing.

The tests that are developed using this framework are able to detect all faults in a time-efficient manner, thereby showing the effectiveness of the proposed framework.

The remainder of this paper is organized as follows. Section 2 introduces the basic operating principles of RRAM. Section 3 introduces the proposed RRAM test development framework that consists of three steps: defect modeling, fault modeling, and test development. Section 4 presents



(a) Structure (b) Parameters (c) Symbol (d) I-V curve (e) 1R (f) 1T1R

Fig. 1. RRAM device.

an overview of the RRAM manufacturing process and the associated defects. Section 5 describes the modeling of interconnection and contact defects using traditional linear resistor defect models, as well as forming defects using the DAT approach. Section 6 systematically defines the RRAM fault space. Section 7 presents the methodology to validate the fault space. Section 8 applies this methodology and presents the resulting set of realistic faults that need to be detected by a test. Section 9 presents tests to detect these faults. Section 10 discusses the results in the paper and provides an outlook for future research directions. Finally, Section 11 concludes this paper.

2 BACKGROUND

This section provides background information on two topics. First, it explains both the physical and electrical operation principles of the RRAM memristive device. Second, it explains how this memristive device is used in memory circuits.

2.1 RRAM Devices

An RRAM device consists of an oxide stacked between two electrodes, the **top electrode (TE)** and **bottom electrode (BE)** [63, 68], shown in Figure 1(a). In this oxide layer (green), a ***conductive filament*** **(CF)** can exist whose shape determines the resistive properties of the device. By applying voltages to the electrodes, the shape of the CF changes and hence the resistance changes. The CF's shape remains unaffected when the voltage over the device is removed; the memristive device thus retains its resistance and functions as non-volatile memory.

When a positive voltage is applied to the TE with respect to the BE, the bond between some of the metal and oxygen ions in the oxide layer breaks [7, 63]. Under influence of the electric field, the oxygen ions are attracted to the TE thus leaving an oxygen vacancy behind. A chain of these vacancies becomes a CF that is able to carry a current. If a negative voltage is applied to the TE, the oxygen ions move back into the oxide and fill the vacancies. Figure 1(a) illustrates the CF. The blue dots represent the vacancies that conduct a current, while the green background represents the regular oxide. Wider and longer CFs will have a lower resistance, while shorter and thinner ones will have a higher resistance. Figure 1(b) shows an abstracted view of the CF with technology parameters that describe its shape. The parameter explanation can be found in Table 1. Because CFs tend to be formed along defects in the oxide [39], good control of the defect density in the oxide is required. The circuit symbol of the memristive device is shown in Figure 1(c).

The electrical behavior of the RRAM device can be described by a few parameters and a typical I-V hysteresis curve [63], which is shown in Figure 1(d). In the figure, two resistive states can be distinguished: a ***high-resistive state*** **(HRS)** or reset state with resistance R_{HRS} and a ***low-resistive state*** **(LRS)** or set state with resistance R_{LRS}. In this work, we denote R_{HRS} as a logical '0' and R_{LRS} as a logical '1'. The figure shows that the switching between these two states happens at a certain

Table 1. RRAM Key Parameters

Technological Parameters		Electrical Parameters	
t_{ox}	Oxide thickness	V_{reset}	Reset threshold
l_{CF}	CF length	V_{set}	Set threshold
l_{gap}	Gap length	R_{HRS}	Reset resistance
ϕ_T	CF top width	R_{LRS}	Set resistance
ϕ_B	CF bottom width	$t_{H\rightarrow L}$	HRS to LRS delay
		$t_{L\rightarrow H}$	LRS to HRS delay

threshold voltage; at V_{set}=0.5 V the RRAM device switches from HRS to LRS, while at V_{reset}=−0.5 V it switches from LRS to HRS. This switching takes some time, denoted by $t_{H\rightarrow L}$ for the HRS to LRS transition and $t_{L\rightarrow H}$ for the LRS to HRS transition, respectively.

2.2 RRAM Cells

Resistive memories use RRAM devices as a data storage element in the memory cell. There exist several cell designs. The two commonly used ones are **one memristive device (1R)** and **one transistor and one memristive device (1T1R)**, shown in Figures 1(e) and 1(f), respectively. 1R cells can be stacked on top of each other in a crossbar or cross-point structure, thus creating a memory with a high bit density [64]. The parallel nature of this architecture introduces sneak paths [32], where neighboring cells are partially selected as well. Sneak paths can be mitigated by adding a selecting device in series with the memristive device [64]. Typically, a transistor is used as a selector, resulting in a 1T1R cell structure. Although this cell design does not suffer from sneak path issues, it cannot be fabricated as densely as 1R cells due to the transistor connection.

In Figure 1(f), BL and SL denote, respectively, bit line and select line, WL denotes word line. The cell can be set to a logic state by applying appropriate voltages to the BL and SL. The WL in the 1T1R configuration additionally controls the access. The cell can be brought into the set state by applying a voltage higher than V_{set} to the BL with the SL grounded, while it can be brought to the reset state by applying a reversed voltage higher than V_{reset} across the BL and SL. A complete description of an RRAM based on 1T1R cells will be discussed in Section 7.2.

3 OVERVIEW OF RRAM TEST DEVELOPMENT FRAMEWORK

It was shown in [18, 19] that the behavior of a defective RRAM device cannot be properly modeled with traditional linear resistor defect models, because they cannot describe the non-linear behavior of a defective RRAM device. Hence, tests developed based on these models are unable to detect all the *realistic* device failures, leading to test escapes, or they test for *unrealistic* faults, increasing the yield loss. Our proposed complete RRAM test development framework relies on accurate defect and fault modeling in order to develop better test solutions for RRAMs. As shown in Figure 2, appropriate defect and fault modeling leads to the detection of more realistic faults, while decreasing the detection of unrealistic faults, thus increasing test quality. In this section, we describe the proposed framework. It is illustrated in Figure 3 and consists of three steps that are explained next:

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

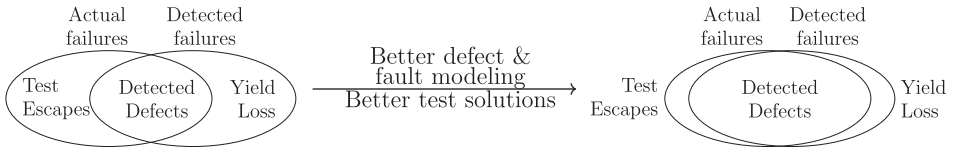


Fig. 2. Aim of the framework.

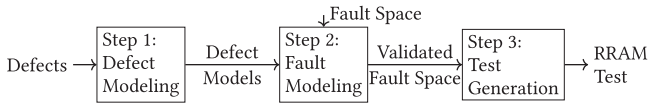


Fig. 3. RRAM test development framework.

Traditionally, defects in RRAMs are modeled as *linear* resistors where the resistance determines the defect strength [28, 34]. However, it has been shown that this defect model is unable to represent the defective behavior of a *non-linear* RRAM device properly [18], leading to low-quality test solutions. The DAT defect modeling approach [19] incorporates the effects of the defect on the physical parameters of a device and thus is able to describe the defective behavior properly. We will apply this approach to model defects in the RRAM device.

(2) **Fault modeling:** In this step, the defect is described as a fault on the functional level of the memory [6]. First, the fault space is classified and defined to describe all theoretically possible faults. Then, the defect models from the first step are incorporated in the netlist and the circuit is simulated. The resulting behavior is analyzed and, if found erroneous, labeled as a fault. This is repeated for all defects and results in the validated fault space. This is a list of faults that can realistically occur in the circuit and for which a test needs to be developed.

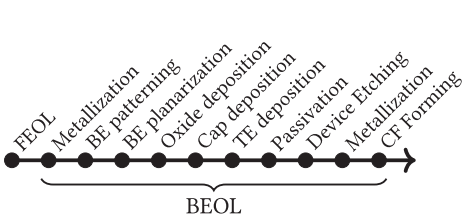
(3) **Test generation:** In this step, test solutions are developed that can detect all faults from the previously validated fault space. The resulting test solutions can be, for example, march algorithms, special DfT schemes, or stress tests.

In the remainder of this paper, we will follow the three steps of the proposed framework. *Defect modeling* is covered by Section 4 where we identify and analyze all RRAM defects, and Section 5 where we model interconnect, contact, and RRAM forming defects. *Fault modeling* is covered by Section 6 where we define the fault space, Section 7 where we present the fault space validation methodology, and Section 8 where we present the results from the fault space validation. Then, *test generation* is addressed by Section 9, where we develop tests to detect the validated faults.

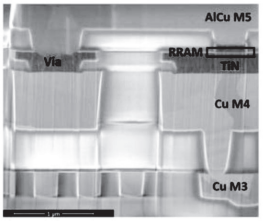
4 RRAM MANUFACTURING PROCESS AND DEFECTS

The RRAM manufacturing process is not ideal and therefore may result in defects. These defects will cause parametric variations or even non-functioning devices. As mentioned before, the detection of defective devices requires appropriate test schemes. In turn, these schemes rely on an accurate understanding of the defect behavior and its extent, i.e., the defect space. We define the defect space by analyzing the RRAM production process from start to finish. For a complete understanding, we study *all* defects that can occur in an RRAM, not only those in the memristive device.

A generalized RRAM production flow can be split into two steps: **front-end-of-line (FEOL)** and **back-end-of-line (BEOL)** [20, 21, 36]. The manufacturing flow is schematically depicted in Figure 4(a) [20, 21, 36]. The process starts with the FEOL phase in which transistors are fabricated on the wafer. This step is the same as the conventional CMOS production flow. In the following



(a) RRAM manufacturing process [20, 21, 36].



(b) RRAM cross section. With permission from [21].

Fig. 4. RRAM production process and cross section.

Table 2. RRAM Defect Classification

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

BEOL phase, the metal layers are fabricated. Somewhere between these layers the memristive devices are placed, e.g., between M4 and M5 [21]. Figure 4(b) shows the cross-section of a resulting RRAM cell [21].

The defects that can occur during the RRAM production can be classified into FEOL and BEOL defects, based on the production phase where they occur. The FEOL defects refer to those that occur in transistor devices. The BEOL defects can be further classified into two categories: Interconnection and RRAM device defects. Table 2 lists the RRAM defects according to this classification. In the following sections, these defects will be discussed in detail.

4.1 FEOL Defects

During the FEOL phase, transistors are fabricated on the wafer. Since transistor defects are well studied, only a brief overview of them is presented here. Kuhn et al. divided transistor defects into two categories: historical and emerging defects [38]. Historical defects include those from *patterning proximity effects*, *line-edge* and *line-width roughness*, *polish variations* for shallow trench isolation, and *variations in the gate dielectric* [37, 38]. Emerging defects are *random dopant* fluctuations, *anneals*, *strains*, and *gate material granularity* [38].

4.2 BEOL Defects

The next production phase is the BEOL where the metal layers as well as the RRAM devices are fabricated. Metallization of the lower layers again is equal to that of a standard production process, and thus the same defects may occur here. For example, misalignment or *small particles* may lead to poor connections that increase the resistance of a wire. Lithographic issues such as *line-edge roughness* may attribute to the formation of *irregular shapes* and affect the wire resistance and capacitance [60], which in turn reduces the RRAM performance [2, 43].

After the lower metal layers are deposited, the RRAM device (see Figure 1(a)) can be constructed. This step starts with the deposition and patterning of the BE that connects to the underlying metal layer. The deposition process leaves a *rough surface* on the BE. Due to this roughness, there

will be many surface defects between the BE and the RRAM oxide that increase the variability as well as the probability of a hard oxide breakdown [40, 53]. Therefore, the BE is planarized by a chemical-mechanical polishing step to reduce the electrode roughness [11, 40]. The polishing step also needs to be well-controlled, in order to prevent the generation of *polish variations*.

After the completion of the BE, the oxide can be deposited on top of it. To minimize device variations, the *thickness of the oxide layer* as well as the *amount of defects in the oxide and at the oxide interface* need to be controlled. For these reasons, the oxide is typically deposited by an atomic layer deposition process rather than physical vapor deposition process, since the former process provides a tighter control [31, 36]. The deposited oxide can have a (poly-) crystalline or an amorphous structure. The edges of a crystalline structure are called **grain boundaries (GBs)** and contain a higher number of defects than the inside of the crystals. Therefore, CFs will tend to form along these edges [30, 39]. Because the dimensions of a crystal can vary, the number of poly-crystals in an oxide will vary, and hence, the crystalline oxides will have a wider resistance distribution than amorphous oxides [22]. However, the amorphous oxides have a smaller HRS/LRS ratio than crystalline structures which makes it more difficult to distinguish the two states [22].

Next, a capping layer can be placed over the oxide. The capping layer acts as a reservoir for the oxygen ions, thus allowing for improved switching performance of the RRAM device [11, 15, 51, 63]. The material in this layer can be specifically selected for its capping capabilities, or it can be the TE material. The TE is deposited in the same fashion as the BE, although the polishing step is not always included. The stack of materials is then etched to separate the individual memristive devices. Lithographic issues may again lead to *variations of the device dimensions*, affecting the performance of the finalized device [56]. Etching may lead to *redeposition of material* along the sidewalls of the RRAM device, thus forming a parasitic leakage path that lowers the resistance and increases the variability [3, 55]. Finally, the memristive device is isolated from its surrounding structures and the production flow continues with the next metal layer, similar to the standard CMOS BEOL.

In the final production step of an RRAM, a CF is formed in the oxide. The forming conditions have a strong impact on the CF shape. Typically, the forming process requires a voltage V_{form} that is higher than V_{set} or V_{reset} [54]. Higher V_{form} will lead to a quicker formation of the CF, but it may also lead to a complete breakdown of the oxide [58]. To prevent this, a forming scheme can be applied that measures the forming of the CF and adapts V_{form} accordingly [23, 46]. Apart from V_{form}, the forming current (I_{form}) flowing through RRAM devices also needs to be taken into account. In general, higher forming currents will result in wider CF structures and hence in a lower overall resistance and less variation, while lower currents will lead to thinner CFs [8, 12, 15, 21, 54]. Furthermore, I_{form} needs to be kept as constant as possible, as I_{form} fluctuation will lead to variations in the device resistance later on [33]. In addition to the forming conditions, the device geometry also plays an important role. Smaller devices will have a higher resistance because the probability of forming a CF is lower, while larger devices have a higher forming probability [8, 20, 58].

Two defects may get introduced by the forming step in the RRAM. It is possible that the CF will barely or not at all form, and thus the memristive device will always remain in HRS. This is referred to as an *under-forming* defect. In contrast, the formation of the CF may also be too strong, for example due to a variable forming current. This will push the resistance of the device below its design specifications and render the device overformed, i.e., an *over-forming* defect [10, 62]. Even stronger formation may lead to a breakdown of the oxide, causing the device to be always stuck at LRS, unable to switch back to HRS [58].

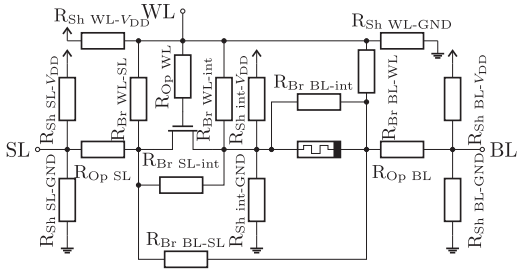


Fig. 5. Open, bridge, and short defect locations.

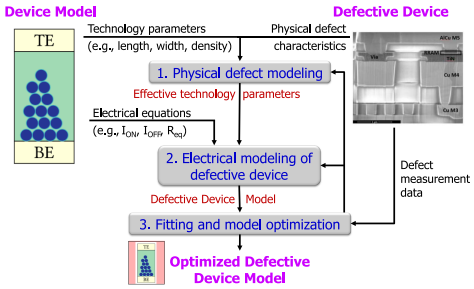


Fig. 6. Device-aware defect modeling.

5 DEFECT MODELING

Defects need to be accurately modeled so that their behavior in a circuit can be studied and optimal test solutions can be developed. In this section, we present defect models for interconnection, contact defect, and forming defects. The former two defect types are modeled using linear resistor defect models, while the latter are modeled using the DAT approach [19].

5.1 Interconnection & Contact Defect Modeling

Defects in the interconnections and contacts are typically modeled as linear resistors [28, 34]. For example, a poorly placed contact can be modeled as an increased series resistance. The linear resistor defect models are classified into three categories: open (Op), short (Sh), and bridge (Br). An open is a broken connection, a short is a short-circuit to either V_{dd} or GND, and a bridge is a short-circuit between two nodes that are not V_{dd} or GND. The resistance of the defect represents the strength of the defect, e.g., a 142 Ω short represents a stronger defect than a short of 10 kΩ [28, 34]. Figure 5 shows the locations and names of all 17 opens, bridges, and shorts considered in this study.

5.2 Device-Aware Defect Modeling Approach

It is of prime importance that a defect model accurately describes the *physical* behavior of a defect. If the model is unable to do so, then it will lead to the sensitization of unrealistic faults, and subsequently result in ineffective tests, as illustrated in Figure 2. Linear resistors surrounding an RRAM device are unable to describe the non-linear behavior *inside* a defective RRAM device properly. Therefore, we apply the device-aware defect modeling approach that is shown in Figure 6 to incorporate the physical behavior of defects in a defect model [19, 66]. It closes the gap between real defects and accurate defect models for test development. The inputs to this approach are a physical

Table 3. Fitting Parameters for the DAT Forming Defect Model

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

device model and measurement data of defective devices. The output is an optimized defect model that takes into account the physics of the defective device. Note that a device can be any device e.g., a transistor, a RRAM device, etc. We demonstrate the approach by modeling the RRAM under and over-forming defects caused by variations in the forming current I_{form}. The approach consists of three steps [19]:

(1) *Physical defect modeling:* The resistance of an RRAM device is mainly affected by the length of the tunneling gap l_{gap} and the width of the top of the filament ϕ_{T} as illustrated in Figure 1(a) [7, 50]. Therefore, the effective values ($l_{\text{gap,eff}}$ and $\phi_{\text{T,eff}}$) for these parameters have to be determined under influence of I_{form}. This is described as:

$$l_{\text{gap,eff}} = \frac{p_1 \cdot R_\mu\left(I_{\text{form}}\right)^2 + p_2 \cdot R_\mu\left(I_{\text{form}}\right) + p_3}{R_\mu\left(I_{\text{form}}\right)^2 + p_4 \cdot R_\mu\left(I_{\text{form}}\right) + p_5} \cdot p_6 + p_7, \tag{1}$$

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

In the above two equations, $R_\mu\left(I_{\text{form}}\right)$ is described by the median resistance in [21] as:

$$R_\mu\left(I_{\text{form}}\right) = r_1 + \frac{r_2}{1 + \left(r_3 \cdot I_{\text{form}}\right)^{r_4}}. \tag{3}$$

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

(2) *Electrical defect modeling:* The RRAM device model in [42] takes l_{gap} and ϕ_{T} as input parameters. Hence, we can include the effective technology parameters directly in the model and observe their effects on the electrical parameters (see Table 1) by simulating it.

(3) *Fitting and model optimization:* In this step, the electrical model is fitted to match the behavior of the real defective devices. We fitted the model to the measurements in [21] by changing the values of parameters $p_{1,2,3,4,5,6,7}$, $q_{1,2,3,4,5,6,7}$, and $r_{1,2,3,4}$ using MathWorks's MATLAB R2019b [49], which resulted in the values listed in Table 3. First, we have fitted Equation (3), then we analyzed how the resistance of the model in [42] is affected by changing the parameters l_{gap} and ϕ_{T}. Subsequently, we fitted Equations (1) and (2) to match to Equation (3). Figure 7 shows the values of $l_{\text{gap,eff}}$ and $\phi_{\text{T,eff}}$, as well as the resistance after forming from [21] and the model prediction. It can be seen that higher forming currents lead to a small decrease in the tunneling gap length ($l_{\text{gap,eff}}$) and a major increase in filament width ($\phi_{\text{T,eff}}$), which is also observed by [7]. Furthermore, it can be seen that the model matches the resistive measurements from [21].

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

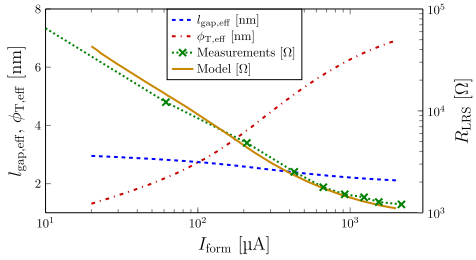


Fig. 7. DAT forming defect model performance compared to measurements in [21].

6 FAULT SPACE CLASSIFICATION AND DEFINITION

Defects that lead to the erroneous behavior in an RRAM can be modelled as a fault [6]. In this section, we first classify faults based on the number of operations and cells involved. Then, we define the complete fault space for single-cell and multi-cell coupling faults. Finally, we unify the notation of other RRAM faults from literature using these definitions.

6.1 Fault Classification

A fault describes the behavior of a defect on the functional level of the memory [6]. That is, it describes the defect's behavior in terms of memory operations that *sensitize* the fault and the logic values that are stored in the cell after sensitization of the fault. Note that multiple operations may be needed to sensitize the fault, or that multiple cells can be affected by the fault. Hence, we can classify the faults based on the number of operations n that are needed to sensitize the fault, and on the number of cells $\#C$ that are involved in the fault [61]. For the number of operations involved, if $n \leq 1$, then the fault is static, if $n > 1$, then the fault is dynamic. For the number of cells involved, if $\#C = 1$, then the fault is a single-cell fault, if $\#C \geq 1$, then the fault is a multi-cell $\#C$-coupling fault. Note that a fault is always classified in both categories, e.g., there can exist static coupling faults, as well as dynamic single-cell faults. In the following sections we present the static and dynamic fault space for single-cell and multi-cell coupling faults.

6.2 Single-Cell Faults

As mentioned in the previous section, a fault is described by its sensitizing sequence and the effect on the data storing cell. A fault is typically described as a **fault primitive (FP)** described by the following notation: ⟨S/F/R⟩ [18, 24] which extends [61]. In this notation:

- S describes the sensitizing operation of the fault. It is denoted as $x_0 O_1 x_1 \cdots O_n x_n$, where $x \in \{0, 1\}$ denotes the value that is stored in the cell, $O \in \{\mathrm{r}, \mathrm{w}\}$ denotes the operation that is performed (i.e., read (r) or write (w)), and n denotes the amount of operations. Examples are: S = 0r0 where a read operation is performed on a cell containing a '0' and the expected output is also '0', and S = 1w0r0 where a value of '0' is written to a cell that initially contains a '1' and is subsequently read out.
- F describes the state that is stored in the cell after the sensitizing operation. For digital memories (e.g., SRAM), it holds that $\mathrm{F} \in \{0, 1\}$. Because the RRAM device is an analog device, more states can be defined than just '1' and '0'. This concept is shown schematically in Figure 8. In this figure, 1 and 0 denote the resistance ranges for a logical '1' and '0', U denotes the undefined state, and L and H denote, respectively, an extremely low and an extremely high conductance state. In the 'L' state, the resistance of the RRAM device is higher than the HRS

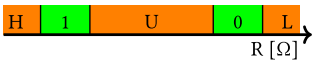


Fig. 8. RRAM resistance range.

range, and, in 'H' the resistance of the cell is lower than the LRS range. Logically, the 'L' and 'H' state behave as a '0' or '1', respectively. The thresholds between these ranges depend on the circuit that is investigated. For RRAMs, F needs to be extended: $F \in \{0, 1, L, H, U\}$.

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

Some faults have a similar behavior, e.g., ⟨1w0/1/−⟩ and ⟨0w1/0/−⟩ both describe a failed transitioning operation. To be able to group these faults we propose to name the faults according to their behavior. If $n = 0$, then the fault is a state fault. These are denoted as FP = S{*ini*}F{*fin*}, where *ini* denotes the initial state of the cell, and *fin* describes the cell contents after the sensitizing operation is completed; it is equal to the F value in the ⟨S/F/R⟩ notation. To illustrate, the ⟨0/U/−⟩ fault is named S0FU. For $n = 1$, the FP are named according to the following scheme: FP = {*out*}{*opn*}{*opd*}{*eff*}F{*fin*}. The fields in this scheme have the following meaning:

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

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

With this information, we can define the complete fault space for single-cell faults. Table 4 lists all static single-cell faults that can occur in an RRAM. It shows that there exist more single-cell static faults in RRAM than in traditional digital memories [61]. This is due to the three additional states ('U', 'L', 'H') that an RRAM device may be in. Dynamic faults require more than one sensitizing operation, i.e., $n \geq 2$. The total number of sensitizing sequences (#S) for a certain n is described by:

$$\#S = \sum_{i=0}^{n} 2 \cdot 3^i. \tag{4}$$

The correctness of this equation can be demonstrated by observing that for $i = 0$, there are only two possible S: S = 0, S = 1. Further, for every $i > 0$, a sensitizing sequence can only be extended by subsequently performing a w0, w1, or a rx with x the same logical value as in the previous

Table 4. Single-Cell Static Fault Primitives

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

operation. Because the total number of FPs grows exponentially with n, we omit listing them as was done for the static ones in Table 4, but the methodology to generate this table is similar.

6.3 Multi-Cell Coupling Faults

Multi-cell faults can also be described with an FP. The FP then needs to describe the fault effect on the victim cell that is caused by $\#C-1$ aggressor cells. In order to do so, the $\langle S/F/R \rangle$ notation scheme has to be extended to $\langle S_{a,1}; \ldots; S_{a,z}; \ldots; S_{a,\#C-1}; S_v/F/R \rangle$ [61]. Here, S_a, z, with $z < \#C$ denotes the sensitizing sequence that is applied to an aggressor cell, S_v denotes the sensitizing sequence that is applied to the victim cell, F and R refer to the victim cell as well. With this extended notation, the complete fault space for static and dynamic multi-cell faults can be defined as well by following the same approach as was done for the single-cell faults. In the remainder of this work, we focus on single-cell faults, because the forming defects do not involve multiple cells.

6.4 Unifying Terminology

In literature, some unique as well as traditional faults have been identified in RRAMs. For clarity, we analyze these faults and define them according to our naming scheme to unify the fault terminology.

- ***SAF*** [28, 35]: **Stuck-at Faults** are defined as a cell always being in a certain state, independent of the sensitizing sequence [61]. The fault can therefore be described by multiple FPs, e.g., a SAF-1 can be represented by a combination of: S0F1, W0TF1, iR0DF1, etc.
- ***Deep 0/1*** [35]: **Deep Faults** are defined as a cell being in the 'L' or 'H' state. Again, multiple FPs can describe this fault: W0DFL, rR1DFH, etc.
- ***SWF*** [35]: **Slow Write Faults** are defined as a cell that fails to transition in the allotted time and is brought into the 'U' state. Therefore, this fault can be described as W1TFU or W0TFU.
- ***URF*** [13, 35]: **Undefined Read Faults** are defined as performing a read operation on a cell that is in a 'U' state, thus resulting in a random read outcome. Because S cannot contain a 'U' state, this FP cannot exist by itself. An appropriate FP for the URF is rR1DFU.
- ***R1D*** [10]: **Read-1 Disturb Faults** are defined as reading operations causing a cell's state to change. Therefore, this fault belongs to the read destructive faults, e.g., iR1DF0, dR1DF0, etc.

7 FAULT SPACE VALIDATION METHODOLOGY

This section presents the fault analysis methodology to validate the fault space to determine which faults are sensitized in the circuit in the presence of a defect. Only for these faults a test needs to be developed. Thereafter, the simulation setup to perform this fault space validation is elaborated.

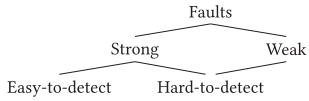


Fig. 9. Detectability of different faults.

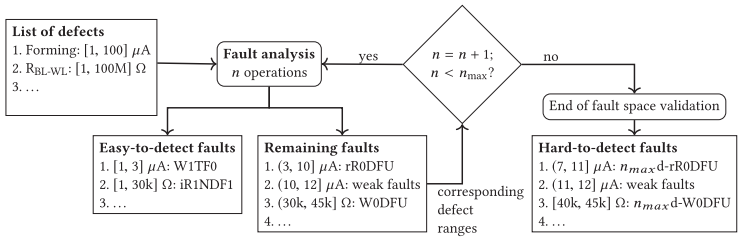


Fig. 10. Fault space validation methodology [19].

7.1 Validation Methodology

The fault space validation methodology aims to determine what faults can realistically occur in a circuit, and to determine which faults can be efficiently detected by a test. The former aim reduces the amount of test escapes, and decreases the yield loss due to wrongly modeled faults, as shown in Figure 2. For the latter aim, the detectability of the faults has to be identified, as shown in Figure 9. A fault can be a strong or a weak fault. A strong fault is *always* sensitized by a certain sequence of operations and it can be described by an FP, e.g., W0DFU, rR1NF1, or dR0DF1. A strong fault will thus always affect the cell's state or the read output, or both. These faults *may* be detected by, for example, a march test that only uses regular memory operations. In contrast, a weak fault does not cause functional errors but parametric deviations instead and cannot be described by an FP, as the cell's contents or read output are unaffected. Examples are, a decrease of bit line swing below the nominal values when reading a certain cell, but not causing a wrong or random read output, or when the switching thresholds shift outside of their nominal ranges, but do switch. Strong faults that are *guaranteed* to be sensitized and detected by regular memory operations are called **Easy-to-Detect (EtD)** faults, e.g., W1TF0 and iR1NF1; both these faults will be detected by performing a regular read operation that always outputs a wrong value. Note that not all strong faults in Table 4 are EtD. For example, it *cannot* be guaranteed that the random read output caused by rR1NF1 is detected by performing a normal read operation, as its output has a *probability* to be either '1' or '0'. Hence, a regular read operation will only in some cases detect these faults. These strong faults together with all weak faults are called **Hard-to-Detect (HtD)** faults. To detect these faults, additional effort is required, e.g., by using DfT schemes. Hence, a strong fault can be further classified as EtD or **strong HtD (sHtD)**, while weak faults are always **weak HtD (wHtD)** faults. Note that these two fault types both are hard faults, as they will always occur given the same circumstances; there is no random component involved in their sensitization.

The fault analysis methodology consists of seven steps and is shown in Figure 10 [19]: (1) circuit generation, (2) defect injection, (3) stimuli generation, (4) circuit simulation, (5) fault analysis, (6) fault primitives identification, (7) defect strength sweeping, and repetition of steps 2 to 7 until all defects and defect strengths are covered. The methodology starts with a list of defects along with their strengths. These defects are injected one by one in the netlist to which stimuli for at most one operation, i.e., $n \leq 1$, are applied. Then, the behavior of the circuit is analysed. If the fault can be described by an FP, then the fault is strong. For these faults the detectability is determined,

i.e., whether it is an EtD or sHtD fault. If the parameters of the circuit are outside of its specifications but are not described by an FP, then the fault is weak, i.e., a wHtD fault. If the parameters of the circuit are within the specification, then the there is no fault present for that value of n.

Defects with certain defect strength ranges that sensitize EtD faults do not require further analysis, as they can already be easily detected. They are listed as EtD faults. Analysing them further allows to better optimize tests, as will be shown in Section 9. The remaining defect strength ranges are added to the remaining faults list. They are simulated again, but now with a longer sensitizing sequence with length $n = n + 1$. This process is repeated until all defects and sizes are in the EtD list, or when the n passes a maximum $n_{\max}$. These remaining HtD faults are dynamic faults that are sensitized when $n = n_{\max}$, e.g., a $n_{\max}$d-rR0DFU fault. The choice of $n_{\max}$ is an economic choice that depends on the allotted total simulation time and the desired fault coverage. Equation (4) showed that the total time increases exponentially with n, hence a higher *guaranteed* fault coverage comes at a higher development cost. Note that $n_{\max}$ is the upper limit and that the test development cost can be reduced e.g., if it becomes clear that no additional faults are sensitized with increasing n, the process can end earlier. Some defect strength ranges do not sensitize any fault when $n = n_{\max}$ and the circuit could be labeled fault-free. However, these defects may form a reliability risk, e.g., electromigration poses a higher risk in thinner wires. Hence, a test developer needs to make a choice between increasing $n_{\max}$ to increase the fault coverage, and to minimize development costs.

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

The total worst-case analysis time ($t_{\text{analysis, w.c.}}$) using this fault analysis methodology is determined by $n_{\max}$ (which affects #S, see Equation (4)), the total number of defects ($\#D_{\text{tot}}$), and the number of defect strengths per defect ($\#D_{\text{strength}}$). It is described by:

$$t_{\text{analysis, w.c.}} = \left(\sum_{i=0}^{n_{\max}} 2 \cdot 3^i \right) \cdot \#D_{\text{tot}} \cdot \#D_{\text{strength}}. \tag{5}$$

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

7.2 Simulation Setup

Next, we describe first the circuit that is used in our experiments, followed by the detailed application of the proposed fault analysis methodology.

7.2.1 Circuit. Figure 11 shows the RRAM architecture that is used in this paper. It consists of a memory cell array and peripheral circuits that drive it. The cell array is divided into nine data words with each consisting of three 1T1R cells (see Figure 1(f)) that store one bit per cell. Three words are placed in a row, sharing the **word (WL)** and **select line (SL)**. Similarly, there are also three words per column that share three **bit lines (BLs)**. The peripheral circuits consist of a WL decoder, SL and BL drivers, **sense amplifiers (SAs)**, and a column address decoder. The WL driver decodes the row address and enables the corresponding row when WL_{EN} is enabled. The column address decoder decodes the column address and generates the **column select (CS)** signal for the corresponding column. The SL driver drives the SL to V_{DD} when a reset operation takes place in the selected cells, otherwise it keeps the SL at GND level. The BL driver drives the BL to V_{DD} when

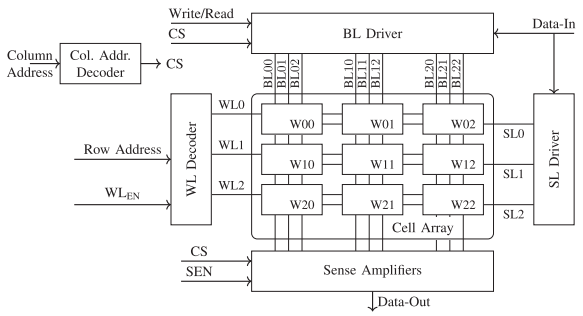


Fig. 11. RRAM simulation circuit architecture.

a set operation is performed, to GND when a reset operation is performed, or it keeps its output in a high impedance state when a read operation is performed. Performing multiple set operations subsequently can lead to over-setting the device, i.e., the filament grows to wide and the resistance becomes too low. This is prevented by always applying a reset operation first, possibly followed by a set operation. The SAs are based on the precharge SA design in [69]. The sensing operation starts when the corresponding column is selected via CS and the SEN signal is enabled. The circuits are implemented using the 130 nm PTM transistor models [1] and RRAM (Pt / HfOx (4 nm)/TiOx (2 nm)/HfOx (4 nm)/TiOx (2 nm) / TiN) compact model from [14, 42] with a nominal forming current of 220 µA.

7.2.2 Application of the Fault Space Validation Methodology. In our simulations we apply the defect models that were discussed in Section 5 in a single memory cell. That is, we take the defect-free memory cell and add one defect with a certain strength at a time. For transistor and interconnect defects we use linear resistor defect models as shown in Figure 5, while for the RRAM device the forming defect model is used. The strength of these transistor and interconnect defects is varied from 1 Ω to 100 MΩ. The forming defect's strength is varied from 20 µA to 2 mA.

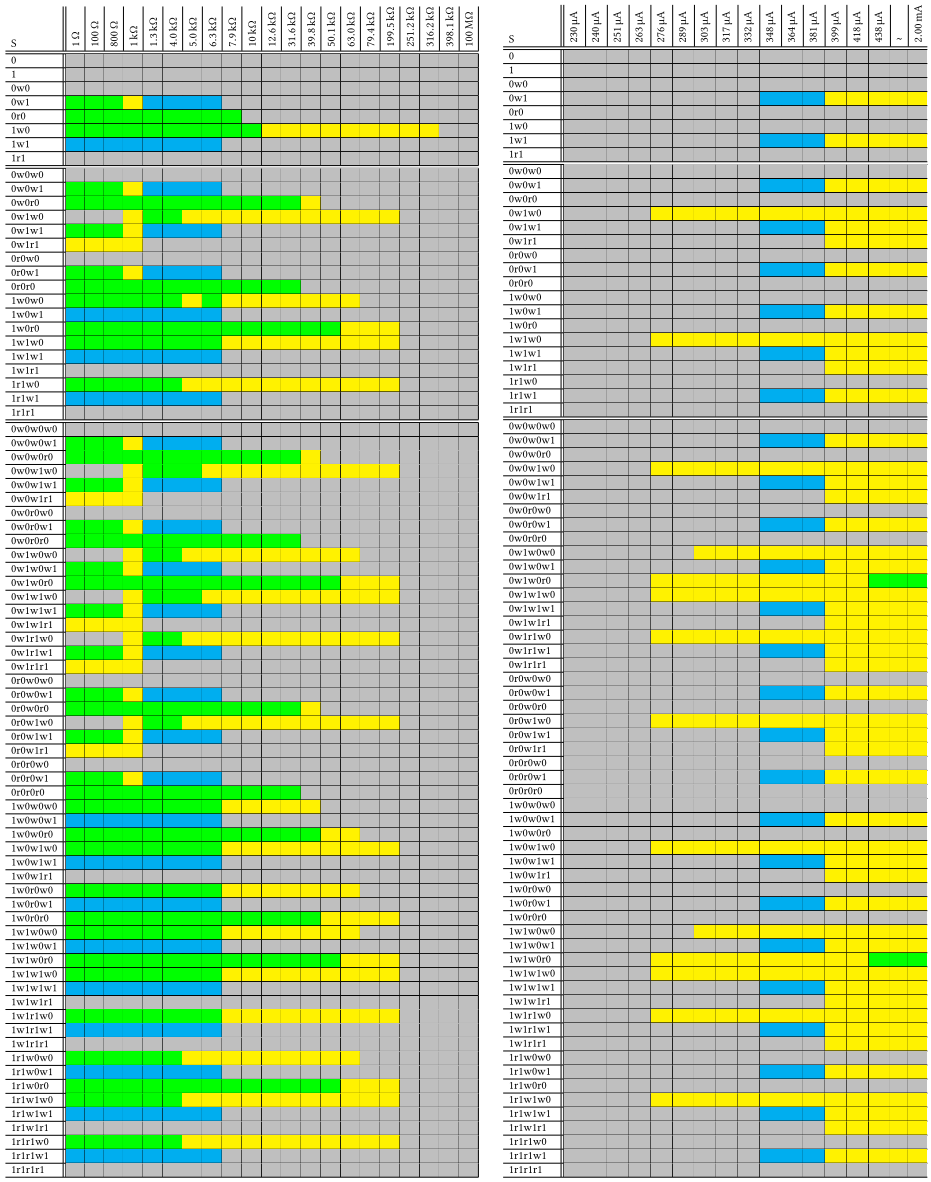
After the defect is injected, we simulate the defective circuit using Cadence's Spectre simulator. We first simulate static faults ($n \leq 1$), then dynamic faults up to $n = n_{\max} = 3$. After the simulation is completed, we identify what faults occur and classify them as EtD or HtD. If no FP is sensitized, then it can still be a weak fault. We define a fault as weak when there is more than 20 % variation in the nodal voltages with respect to the defect-free case, e.g., the voltage on the cell internal node, or on the bit lines. These limits are chosen to accommodate for the naturally occurring cycle-to-cycle variation of the RRAM device [63], while still ensuring reasonable power consumption.

8 FAULT SPACE VALIDATION RESULTS

This section presents the results from the fault analysis. Only for the faults that are validated, a test needs to be developed. We present first the faults that were sensitized by the interconnect and contact defects. Subsequently, we present the faults that were sensitized by the forming defect.

8.1 Interconnect and Contact Defects

We first illustrate in detail the results for one defect, the bridge defect $R_{Br\ BL\text{-}int}$ (see Figure 5). After this, we summarize the results for the remaining defects. The *fault map* in Figure 12(a) shows the classes (i.e., EtD, sHtD, wHtD) of the faults that were sensitized for the $R_{Br\ BL\text{-}int}$ defect with varying defect strength and sensitizing operation. The gray color indicates fault-free behavior, green indicates an EtD fault, yellow an sHtD fault (i.e., it can be described by an FP), and cyan a wHtD fault.



(a) $R_{\text{Br BL-int}}$ defect (b) Over-forming defect

Fig. 12. Fault map for $R_{\text{Br BL-int}}$ and over-forming defect. Fault-free EtD HtD strong HtD weak

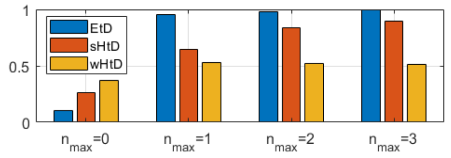
The figure shows that faults are sensitized for lower defect strengths, which is expected for a bridge defect. Furthermore, it can be seen that defects that sensitize EtD faults for lower defect strengths, will sensitize strong HtD faults for increased defect strengths, e.g., see S = 1w0. To illustrate, the defect range $R_{\text{Br BL-int}} \in [1, 10\text{k}]$ sensitizes EtD faults, while the range $R_{\text{Br BL-int}} \in (10\text{k}, 316\text{k}]$ sensitizes sHtD ones. The figure also shows that increasing the length of the sensitizing sequence increases the amount of EtD faults that is sensitized. For example, performing a r0 operation after the previous 1w0 operation (i.e., S = 1w0r0) sensitizes an EtD fault for the defect

Table 5. Sensitized Fault Classes (FCs) Per Defect

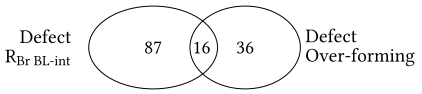
S	$R_{Br\ BL-SL}$	$R_{Br\ BL-WL}$	$R_{Br\ BL-int}$	$R_{Br\ SL-WL}$	$R_{Br\ SL-int}$	$R_{Br\ WL-int}$	$R_{Cp\ BL}$	$R_{Cp\ SL}$	$R_{Cp\ WL}$	$R_{Sh\ BL-GND}$	$R_{Sh\ BL-V_{DD}}$	$R_{Sh\ SL-GND}$	$R_{Sh\ SL-V_{DD}}$	$R_{Sh\ WL-GND}$	$R_{Sh\ WL-V_{DD}}$	$R_{Sh\ int-GND}$	$R_{Sh\ int-V_{DD}}$	Under-forming	Over-forming
0																		$\langle S/L/-\rangle^{1,2}$	
1																		$\langle S/U/-\rangle^{1,2,3}$	
0w0																$\langle S/U/-\rangle^{1}$		$\langle S/L/-\rangle^{1,2}$	
0w1																		$\langle S/U/-\rangle^{1,2,3}$	$\langle S/H/-\rangle^{2}$
0r0																		$\langle S/L/0\rangle^{1,2}$	
1w0	$\langle S/U/-\rangle^{1,2}$		$\langle S/U/-\rangle^{1,2}$				$\langle S/1/-\rangle^{1}$, $\langle S/U/-\rangle^{2}$					$\langle S/U/-\rangle^{1,2}$		$\langle S/1/-\rangle^{1}$, $\langle S/U/-\rangle^{2}$		$\langle S/U/-\rangle^{1}$		$\langle S/L/-\rangle^{1,2}$	
1w1																		$\langle S/U/-\rangle^{1,2,3}$	$\langle S/H/-\rangle^{2}$
1r1		$\langle S/1/0\rangle^{1}$		$\langle S/1/0\rangle^{1}$		$\langle S/1/0\rangle^{1}$	$\langle S/1/0\rangle^{1}$	$\langle S/1/0\rangle^{1}$	$\langle S/1/0\rangle^{1}$		$\langle S/1/0\rangle^{1}$		$\langle S/1/0\rangle^{1}$		$\langle S/L/0\rangle^{1}$		$\langle S/1/0\rangle^{1}$	$\langle S/U/0\rangle^{1,2}$, $\langle S/U/1\rangle^{3}$	
0w0w0																$\langle S/U/-\rangle^{1}$		$\langle S/L/-\rangle^{1,2}$	
0w0w1																		$\langle S/U/-\rangle^{1,2,3}$	$\langle S/H/-\rangle^{2}$
0w0r0	$\langle S/0/1\rangle^{1}$									$\langle S/0/1\rangle^{1}$						$\langle S/U/0\rangle^{1}$		$\langle S/L/0\rangle^{1,2}$	
0w1w0	$\langle S/U/-\rangle^{1}$		$\langle S/U/-\rangle^{1}$				$\langle S/U/-\rangle^{1}$					$\langle S/U/-\rangle^{1}$		$\langle S/U/-\rangle^{1}$		$\langle S/U/-\rangle^{1}$		$\langle S/L/-\rangle^{1,2}$	$\langle S/U/-\rangle^{1,2}$
0w1w1																		$\langle S/U/-\rangle^{1,2,3}$	$\langle S/H/-\rangle^{2}$
0w1r1				$\langle S/1/0\rangle^{1}$		$\langle S/1/0\rangle^{1}$		$\langle S/1/0\rangle^{1}$	$\langle S/1/0\rangle^{1}$		$\langle S/1/0\rangle^{1}$		$\langle S/1/0\rangle^{1}$				$\langle S/1/0\rangle^{1}$	$\langle S/U/1\rangle^{1,2,3}$	$\langle S/H/1\rangle^{2}$
0r0w0																$\langle S/U/-\rangle^{1}$		$\langle S/L/-\rangle^{1,2}$	
0r0w1																		$\langle S/U/-\rangle^{1,2,3}$	$\langle S/H/-\rangle^{2}$
0r0r0	$\langle S/0/1\rangle^{1}$									$\langle S/0/1\rangle^{1}$								$\langle S/L/0\rangle^{1,2}$	
1w0w0			$\langle S/U/-\rangle^{1}$				$\langle S/U/-\rangle^{1}$					$\langle S/U/-\rangle^{1}$		$\langle S/U/-\rangle^{1}$		$\langle S/U/-\rangle^{1}$		$\langle S/L/-\rangle^{1,2}$	
1w0w1																		$\langle S/U/-\rangle^{1,2,3}$	$\langle S/H/-\rangle^{2}$
1w0r0	$\langle S/U/1\rangle^{1}$, $\langle S/U/0\rangle^{2}$		$\langle S/U/1\rangle^{1}$				$\langle S/U/0\rangle^{1,2}$			$\langle S/0/1\rangle^{1}$		$\langle S/U/1\rangle^{1}$		$\langle S/1/1\rangle^{1}$, $\langle S/U/0\rangle^{2}$	$\langle S/L/0\rangle^{1}$	$\langle S/U/0\rangle^{1,2}$		$\langle S/L/-\rangle^{1,2}$	
1w1w0	$\langle S/U/-\rangle^{1}$		$\langle S/U/-\rangle^{1}$				$\langle S/U/-\rangle^{1}$					$\langle S/U/-\rangle^{1}$		$\langle S/U/-\rangle^{1}$		$\langle S/U/-\rangle^{1}$		$\langle S/L/-\rangle^{1,2}$	$\langle S/U/-\rangle^{1,2}$
1w1w1																		$\langle S/U/-\rangle^{1,2,3}$	$\langle S/H/-\rangle^{2}$
1w1r1				$\langle S/1/0\rangle^{1}$		$\langle S/1/0\rangle^{1}$		$\langle S/1/0\rangle^{1}$	$\langle S/1/0\rangle^{1}$		$\langle S/1/0\rangle^{1}$		$\langle S/1/0\rangle^{1}$				$\langle S/1/0\rangle^{1}$	$\langle S/U/1\rangle^{1,2,3}$	$\langle S/H/1\rangle^{2}$
1r1w0	$\langle S/U/-\rangle^{1}$		$\langle S/U/-\rangle^{1}$				$\langle S/U/-\rangle^{1}$					$\langle S/U/-\rangle^{1}$		$\langle S/U/-\rangle^{1}$	$\langle S/L/-\rangle^{1}$	$\langle S/1/-\rangle^{1}$, $\langle S/U/-\rangle^{2}$		$\langle S/L/-\rangle^{1,2}$	
1r1w1																		$\langle S/U/-\rangle^{1,2,3}$	$\langle S/H/-\rangle^{2}$
1r1r1				$\langle S/1/0\rangle^{1}$		$\langle S/1/0\rangle^{1}$		$\langle S/1/0\rangle^{1}$			$\langle S/1/0\rangle^{1}$		$\langle S/1/0\rangle^{1}$		$\langle S/L/0\rangle^{1}$		$\langle S/1/0\rangle^{1}$	$\langle S/U/1\rangle^{1,2,3}$	
0w0w0w0					$\langle S/L/-\rangle^{1}$											$\langle S/U/-\rangle^{1}$		$\langle S/L/-\rangle^{1,2}$	
0w0w0w1																		$\langle S/U/-\rangle^{1,2,3}$	$\langle S/H/-\rangle^{2}$
0w0w0r0	$\langle S/0/1\rangle^{1}$									$\langle S/0/1\rangle^{1}$						$\langle S/U/0\rangle^{1}$		$\langle S/L/0\rangle^{1,2}$	
0w0w1w0	$\langle S/U/-\rangle^{1}$		$\langle S/U/-\rangle^{1}$				$\langle S/U/-\rangle^{1}$									$\langle S/U/-\rangle^{1}$		$\langle S/L/-\rangle^{1,2}$	$\langle S/U/-\rangle^{1,2}$
0w0w1w1																		$\langle S/U/-\rangle^{1,2,3}$	$\langle S/H/-\rangle^{2}$
0w0r1r1				$\langle S/1/0\rangle^{1}$		$\langle S/1/0\rangle^{1}$		$\langle S/1/0\rangle^{1}$	$\langle S/1/0\rangle^{1}$		$\langle S/1/0\rangle^{1}$		$\langle S/1/0\rangle^{1}$				$\langle S/1/0\rangle^{1}$	$\langle S/U/1\rangle^{1,2,3}$	$\langle S/H/1\rangle^{2}$
0r0r0w0																$\langle S/U/-\rangle^{1}$		$\langle S/L/-\rangle^{1,2}$	
0r0r0w1																		$\langle S/U/-\rangle^{1,2,3}$	$\langle S/H/-\rangle^{2}$
0r0r0r0	$\langle S/0/1\rangle^{1}$									$\langle S/0/1\rangle^{1}$						$\langle S/U/0\rangle^{1}$		$\langle S/L/0\rangle^{1,2}$	
1w0w0w0			$\langle S/U/-\rangle^{1}$		$\langle S/L/-\rangle^{1}$		$\langle S/U/-\rangle^{1}$					$\langle S/U/-\rangle^{1}$		$\langle S/U/-\rangle^{1}$		$\langle S/U/-\rangle^{1}$		$\langle S/L/-\rangle^{1,2}$	
1w0w0w1																		$\langle S/U/-\rangle^{1,2,3}$	$\langle S/H/-\rangle^{2}$
1w0w0r0	$\langle S/0/1\rangle^{1}$		$\langle S/U/0\rangle^{1}$				$\langle S/U/0\rangle^{1}$			$\langle S/0/1\rangle^{1}$		$\langle S/U/0\rangle^{1}$		$\langle S/U/1\rangle^{1}$	$\langle S/L/0\rangle^{1}$	$\langle S/U/0\rangle^{1}$		$\langle S/L/0\rangle^{1,2}$	
1w0w1w0	$\langle S/U/-\rangle^{1}$		$\langle S/U/-\rangle^{1}$				$\langle S/U/-\rangle^{1}$					$\langle S/U/-\rangle^{1}$		$\langle S/U/-\rangle^{1}$		$\langle S/U/-\rangle^{1}$		$\langle S/L/-\rangle^{1,2}$	$\langle S/U/-\rangle^{1,2}$
1w0w1w1																		$\langle S/U/-\rangle^{1,2,3}$	$\langle S/H/-\rangle^{2}$
1w0w1r1				$\langle S/1/0\rangle^{1}$		$\langle S/1/0\rangle^{1}$		$\langle S/1/0\rangle^{1}$	$\langle S/1/0\rangle^{1}$		$\langle S/1/0\rangle^{1}$		$\langle S/1/0\rangle^{1}$				$\langle S/1/0\rangle^{1}$	$\langle S/U/1\rangle^{1,2,3}$	$\langle S/H/1\rangle^{2}$
1w0r0w0			$\langle S/U/-\rangle^{1}$				$\langle S/U/-\rangle^{1}$					$\langle S/U/-\rangle^{1}$		$\langle S/U/-\rangle^{1}$	$\langle S/L/-\rangle^{1}$	$\langle S/U/-\rangle^{1}$		$\langle S/L/-\rangle^{1,2}$	
1w0r0w1																		$\langle S/U/-\rangle^{1,2,3}$	
1w0r0r0	$\langle S/U/1\rangle^{1}$, $\langle S/U/0\rangle^{2}$		$\langle S/U/0\rangle^{1}$				$\langle S/U/0\rangle^{1,2}$			$\langle S/0/1\rangle^{1}$		$\langle S/U/1\rangle^{1}$		$\langle S/1/1\rangle^{1}$, $\langle S/U/0\rangle^{2}$	$\langle S/L/0\rangle^{1}$	$\langle S/U/0\rangle^{1,2}$		$\langle S/L/0\rangle^{1,2}$	$\langle S/H/-\rangle^{2}$
1w1w0w0			$\langle S/U/-\rangle^{1}$				$\langle S/U/-\rangle^{1}$					$\langle S/U/-\rangle^{1}$		$\langle S/U/-\rangle^{1}$		$\langle S/U/-\rangle^{1}$		$\langle S/L/-\rangle^{1,2}$	$\langle S/U/-\rangle^{1,2}$
1w1w0w1																		$\langle S/U/-\rangle^{1,2,3}$	$\langle S/H/-\rangle^{2}$
1w1w0r0	$\langle S/U/1\rangle^{1}$		$\langle S/U/1\rangle^{1}$				$\langle S/U/0\rangle^{1}$			$\langle S/0/1\rangle^{1}$		$\langle S/U/?\rangle^{1}$		$\langle S/U/1\rangle^{1}$		$\langle S/U/0\rangle^{1}$		$\langle S/L/0\rangle^{1,2}$	$\langle S/U/0\rangle^{1}$, $\langle S/U/1\rangle^{2}$
1w1w1w0	$\langle S/U/-\rangle^{1}$		$\langle S/U/-\rangle^{1}$				$\langle S/U/-\rangle^{1}$					$\langle S/U/-\rangle^{1}$		$\langle S/U/-\rangle^{1}$		$\langle S/U/-\rangle^{1}$		$\langle S/L/-\rangle^{1,2}$	$\langle S/U/-\rangle^{1,2}$
1w1w1w1																		$\langle S/U/-\rangle^{1,2,3}$	$\langle S/H/-\rangle^{2}$
1w1w1r1				$\langle S/1/0\rangle^{1}$		$\langle S/1/0\rangle^{1}$		$\langle S/1/0\rangle^{1}$	$\langle S/1/0\rangle^{1}$		$\langle S/1/0\rangle^{1}$		$\langle S/1/0\rangle^{1}$				$\langle S/1/0\rangle^{1}$	$\langle S/U/1\rangle^{1,2,3}$	$\langle S/H/1\rangle^{2}$
1w1r1w0	$\langle S/U/-\rangle^{1}$		$\langle S/U/-\rangle^{1}$				$\langle S/U/-\rangle^{1}$					$\langle S/U/-\rangle^{1}$		$\langle S/U/-\rangle^{1}$		$\langle S/U/-\rangle^{1}$		$\langle S/L/-\rangle^{1,2}$	$\langle S/U/-\rangle^{1,2}$
1w1r1w1																		$\langle S/U/-\rangle^{1,2,3}$	$\langle S/H/-\rangle^{2}$
1w1r1r1		$\langle S/1/0\rangle^{1}$		$\langle S/1/0\rangle^{1}$		$\langle S/1/0\rangle^{1}$		$\langle S/1/0\rangle^{1}$			$\langle S/1/0\rangle^{1}$		$\langle S/1/0\rangle^{1}$				$\langle S/1/0\rangle^{1}$	$\langle S/U/0\rangle^{1}$, $\langle S/U/1\rangle^{2,3}$	$\langle S/H/1\rangle^{2}$
1r1w0w0			$\langle S/U/-\rangle^{1}$				$\langle S/U/-\rangle^{1}$					$\langle S/U/-\rangle^{1}$		$\langle S/U/-\rangle^{1}$	$\langle S/L/-\rangle^{1}$	$\langle S/1/-\rangle^{1}$		$\langle S/L/-\rangle^{1,2}$	
1r1w0w1																		$\langle S/U/-\rangle^{1,2,3}$	$\langle S/H/-\rangle^{2}$
1r1w0r0	$\langle S/U/1\rangle^{1}$		$\langle S/U/1\rangle^{1}$				$\langle S/U/0\rangle^{1}$			$\langle S/0/1\rangle^{1}$		$\langle S/U/?\rangle^{1}$		$\langle S/U/?\rangle^{1}$	$\langle S/L/0\rangle^{1}$	$\langle S/1/1\rangle^{1}$, $\langle S/U/0\rangle^{2}$		$\langle S/L/0\rangle^{1,2}$	
1r1w1w0	$\langle S/U/-\rangle^{1}$		$\langle S/U/-\rangle^{1}$				$\langle S/U/-\rangle^{1}$					$\langle S/U/-\rangle^{1}$		$\langle S/U/-\rangle^{1}$		$\langle S/1/-\rangle^{1}$		$\langle S/L/-\rangle^{1,2}$	$\langle S/U/-\rangle^{1,2}$
1r1w1w1																		$\langle S/U/-\rangle^{1,2,3}$	$\langle S/H/-\rangle^{2}$
1r1w1r1				$\langle S/1/0\rangle^{1}$		$\langle S/1/0\rangle^{1}$		$\langle S/1/0\rangle^{1}$	$\langle S/1/0\rangle^{1}$		$\langle S/1/0\rangle^{1}$		$\langle S/1/0\rangle^{1}$				$\langle S/1/0\rangle^{1}$	$\langle S/U/1\rangle^{1,2,3}$	$\langle S/H/1\rangle^{2}$
1r1r1w0	$\langle S/U/-\rangle^{1}$		$\langle S/U/-\rangle^{1}$				$\langle S/U/-\rangle^{1}$					$\langle S/U/-\rangle^{1}$		$\langle S/U/-\rangle^{1}$	$\langle S/L/-\rangle^{1}$	$\langle S/1/-\rangle^{1}$, $\langle S/U/-\rangle^{2}$		$\langle S/L/-\rangle^{1,2}$	
1r1r1w1																		$\langle S/U/-\rangle^{1,2,3}$	$\langle S/H/-\rangle^{2}$
1r1r1r1		$\langle S/1/0\rangle^{1}$		$\langle S/1/0\rangle^{1}$		$\langle S/1/0\rangle^{1}$		$\langle S/1/0\rangle^{1}$			$\langle S/1/0\rangle^{1}$		$\langle S/1/0\rangle^{1}$		$\langle S/L/0\rangle^{1}$		$\langle S/1/0\rangle^{1}$	$\langle S/U/0\rangle^{1}$, $\langle S/U/1\rangle^{2,3}$	$\langle S/H/1\rangle^{2}$

Black FCs are EtD, red FCs are sHtD.

range $R_{\text{Br BL}-\text{int}} \in [1, 501\text{k}]$. Hence, when developing a test, this sequence is to be chosen over S = 1w0, because a wider defect range will sensitize EtD faults. Although extending the length of the sensitizing sequence does increase the amount of sensitized EtD faults, there are still defect strengths that only sensitize strong HtD faults. These faults cannot be guaranteed to be detected by regular memory operations alone. Therefore, special measures have to be taken to detect these faults and guarantee the highest fault coverage possible.



(a) Relative total number of faults for increasing $n_{\max}$.



(b) Number of sensitized faults by $R_{Br\ BL\text{-}int}$ and over-forming defect for $0 \leq n \leq 3$.

Fig. 13. Sensitized faults.

Table 5 summarizes the results for the remaining defects. The table is generated by analyzing the fault maps similar to those in Figure 12 and is used further during test development. The table shows for every defect (column, see Figure 5) the **fault classes (FCs)** (superscripted digits) that are sensitized and the corresponding FPs (in which S denotes the applied sensitizing sequence). For clarity, only the FCs that belong to the largest number of defect strengths are listed. A black digit indicates that the FP in this FC is EtD, while a red digit indicates an sHtD fault. Remember that to detect all defects, at least one FP per FC needs to be detected. To illustrate the construction of the table and the FCs, consider the short defect $R_{Br\ BL\text{-}int}$ in Figure 12(a). The fault map shows that $S \in \{1w0r0, 0w1w0r0, 1w1w0r0, 1r1w0r0\}$ all sensitize EtD faults for the defect strength up to 50.1 kΩ. However, for the same range of defect strengths, there are also sHtD faults sensitized, e.g., by $S \in \{1w0, 1w1w0, \ldots\}$. Because the defect can be detected if any of these faults is detected, these two sets form a single FC, labeled '1' in Table 5. From Figure 12(a) it follows that $S = 1w0$ uniquely sensitizes sHtD faults from 50.1 kΩ up to 316.2 kΩ. Hence, this also forms a single FC, labeled with a red '2' in the table. From the table, we now can see that to detect $R_{Br\ BL\text{-}int}$, only two FCs need to be detected. Further, it tells that FC 1 is EtD and thus can be detected by regular memory operations, while the guaranteed detection of sHtD FC 2 requires additional effort. Note that no wHtD faults are sensitized by this defect without also sensitizing an EtD or sHtD fault. Therefore, these faults are not listed in the table.

8.2 Forming Defect

Figure 12(b) graphically shows the fault types that were sensitized using the over-forming defect model. The figure shows that the defect mainly sensitizes sHtD faults. These are all related to cells being in an illegal state, e.g., W1DFH or 2d-W0TFU. This is to be expected, as the higher forming current will lead to a lower device resistance, hence, the reset state shifts to the 'U' region, and the set state shifts to the 'H' region [21]. The figure further shows that EtD faults are only sensitized when $n \geq 3$, thus again proving that extending the length of S can lead to sensitization of more EtD faults. Table 5 also shows the FCs that were sensitized by the forming defect.

Figure 13(a) shows the relative number of faults per n relative to $n_{\max} = 3$, i.e., the number of EtD faults for $n_{\max} = 3$ equals 100%. The entries are normalized by dividing them by the number of Ss because the absolute number of faults increases exponentially with $n_{\max}$ (see Equation (4)). The figure shows that with increasing $n_{\max}$ more EtD and sHtD faults are sensitized, while the number of wHtD faults decreases when $n_{\max} > 1$. Clearly, increasing $n_{\max}$ leads to the sensitization of faults that are easier to detect.

We can conclude that forming defects cannot be properly modeled with linear resistor defect models by comparing the fault analysis results from defect $R_{Br\ BL\text{-}int}$ and the over-forming defect in Table 5. Since over-forming defects manifest themselves as a lowered device resistance, one could think that this might be modeled using the $R_{Br\ BL\text{-}int}$ defect model. However, from the table it follows that this is not sufficient, as this defect model is unable to show the RRAM cells switch into the 'H' state. This is to be expected, because the defect does not affect the properties of the RRAM

device directly, but rather its surrounding interconnections and contacts. Figure 13(b) shows the number of faults that are sensitized using these two different defect models for $0 \leq n \leq 3$. It becomes clear that only a limited number of faults overlaps. Without using device-aware defect models and only the $R_{\text{Br BL-int}}$ defect model, only 16 *realistic* faults for this defect will be included in the test, while 87 *unrealistic* faults are included as well, leading to a test overhead of 543%. Furthermore, the test will not detect 36 realistic faults, leading to 225% *test escapes*. Note that a similar argumentation can be made for the Op BL and under-forming defect models.

9 TEST GENERATION

In this section, we generate tests that detect the validated faults from the previous section. Since these tests only need to detect the realistic faults, the amount of test escapes due to poor defect and fault modeling is reduced by increasing the fault coverage, while the yield loss is decreased at the same time, as illustrated in Figure 2. In this section, first, we present the test generation method and tests for the interconnect and contact defects. Second, we present tests for the forming defects.

9.1 Interconnect and Contact Defects

Because EtD faults are guaranteed to be detected by regular memory operations, we first present a test for these faults. After this, we present a test that also detects sHtD faults using a DfT scheme.

9.1.1 Test for EtD Faults. All EtD faults can be sensitized and detected by regular memory operations. Therefore, all these faults can be detected by a march algorithm. In order to minimize test time, the length of this algorithm should be minimized, while still covering all sensitized EtD faults. For example, from Table 5 it follows that $S \in \{1w0r0, 0w1w0r0, 1w1w0r0, 1r1w0r0\}$ will sensitize and detect all EtD faults for the $R_{\text{Br BL-int}}$ defect. For the defect Sh SL-GND, it holds that all EtD faults are sensitized when $S \in \{1r1w0, 1r1w0w0, 1r1w0r0, 1r1r1w0\}$. Because 1r1w0r0 sensitizes faults for both defects, this S should be included in the test to maximize test efficiency. In order to obtain an algorithm with minimal test length and maximal fault coverage, this procedure needs to be applied to all defects. The goal is to sensitize all different FCs by using a minimal set of Ss. This can be formulated as an **integer linear programming (ILP)** problem as follows. We illustrate this problem in Table 6, considering only two defects and six sensitizing sequences. Now, consider a binary matrix $\mathbf{A}$ of p columns and q rows. There is one column (index i) per S and one row (index j) per defect strength that sensitizes a fault in the circuit, i.e., a row for defect 1 with strength 1 Ω, a row for defect 1 with strength 10 Ω, etc. Even if only one defect strength for only one S sensitizes a fault, it is added to the matrix. It is set to 1 if S_i sensitizes a fault for the given defect strength, otherwise it is set to 0. In order to detect all defects strengths, a test needs to sensitize all them by applying at least one S. The last column in the table lists the total number of S that can sensitize a defect strength j. Note that this number is always greater than or equal to 1, because the matrix only contains entries that sensitize a fault.

The ILP problem can now be described as minimizing the number of selected Ss in S_{selected}, while still ensuring that every defect is covered by at least one S_{selected} in the selection. Mathematically this can be described as:

$$\min_{S_{\text{selected}}} \sum_{i=1}^{p} c_i \cdot S_{\text{selected},i} \qquad \text{subject to: } \sum_{i=1}^{p} a_{i,j} \cdot S_{\text{selected},i} \geq 1 \text{ for all rows } j. \tag{6}$$

Here, $S_{\text{selected},i} \in \{0, 1\}$ is a binary value that indicates whether the ith sensitizing sequence is selected and c_i is a weight for this sequence normally set to $c_i = 1$. The first statement ensures that the amount of selected Ss is minimized, while the second statement ensures that every defect strength is covered by at least one sensitizing sequence. To illustrate, Table 6 shows that 0r0 needs

Table 6. Example to Illustrate the ILP Problem and its Solution

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

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

We applied MathWorks's MATLAB R2019b [49] to solve this problem for the EtD faults. The solver finds the set $S_{EtD,int/cont} \in \{1r1, 1w0r0, 1r1w0\}$ that sensitizes all EtD faults. Since all faults are EtD, they can be detected by performing a read operation after the sensitization if the last operation is not an incorrect read operation already. For example, S = 1r1w0 can be detected by adding a r0 operation, while S = 1w0r0 sensitizes and detects the faults. All EtD faults sensitized by interconnect and contact defects can be detected by the following march algorithm:

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

Here, all operations between two parentheses form one march element. The operations of a march element are applied directly after each other to a cell before moving to the next address. ⇕ indicates addressing in any order. The test time for this algorithm is $2N_w + 2N_r$, where N_w and N_r denote the time to write and read, respectively, the complete address space once. It can be seen that the algorithm is minimized in number of operations, as 1r1 can be included in 1r1w0, and then only 1w0r0 needs to be added. We developed this algorithm manually, but it is possible to do this automatically as well if the set of required Ss grows larger [4].

From the above, it follows that solving the problem in Equation (6) leads to minimal set of sensitizing sequences that sensitize all EtD faults. Further, it is possible to reformulate the problem in order to optimize the set of sensitizing sequences for different goals. For example, in Equation (6), it is assumed that $N_w = N_r$ as is reported in [47], i.e., writing takes as much time as reading. However, there exist also RRAM designs where $N_w \neq N_r$, as reported in [67]. For the latter design, the test time can be optimized by favouring the operation that is shorter by changing the weights c_i of sensitizing sequences that contain more of the faster operations. To illustrate, when $N_w = 2 \cdot N_r$, $c_w = 2 \cdot c_r$ as well. Hence, when solving the problem in Equation (6), it becomes harder to minimize the first sum when a write operation is selected and thus read operations are favored.

9.1.2 Test for EtD and Strong HtD Faults. To detect all faults in the RRAM, also the sHtD faults need to be detected by a test. There are two ways to implement such a test: (1) a standalone test is developed for the sHtD faults only and performed after the EtD test, or (2) a test is generated

considering both EtD and sHtD faults at the same time. The first option is easier, but results in less optimized tests, while the second option requires slightly more effort, but results in an optimized test. For this latter reason, we develop a test for both EtD and sHtD faults at the same time.

We follow a similar approach as was done for the EtD faults. First, we make the binary defect matrix **A**, now also including the sensitizing sequences that sensitize sHtD faults. Then, we again solve the problem in Equation (6). This results in the following set of sensitizing sequences that will sensitize all EtD and sHtD faults: $S_{EtD-sHtD,int/cont} \in \{1w0, 1r1, 0w0w0w0, 1r1w0r0\}$. The EtD faults can again be detected by performing a read operation after the sensitizing sequence, if required. However, this does not apply for the sHtD faults. For example, when reading a cell that suffers from the W0TFL fault by the under-forming defect, a subsequent read operation will result in '0', thus hiding the faulty state of the cell. Therefore, DfT is required to detect these faults.

Hamdioui et al. have presented a DfT scheme for RRAMs that modifies the write operations in order to detect cells in the 'U' state [27]. The paper proposes to shorten the duration of the write operation, or to decrease the write voltage. These *weak write* operations are denoted as $\hat{w}$. The idea behind the $\hat{w}$ operations is that defect-free cells will be able to switch to the desired state, but that defective cells do not have enough time or driving voltage to switch and thus will remain in a wrong state. As an illustration, consider the W0TFL fault. It can be sensitized and detected by the following sequence: $\Updownarrow (w0, \hat{w}1, r1)$. The $\hat{w}1$ operation is unable to switch the defective cell to '1' and the cell remains in 'L' or '0', while a defect-free cell can be switched to '1'. Hence, the subsequent r1 will output '1' for a defect-free cell and '0' for a defective cell.

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

$$\text{March-EtD-sHtD, int/cont} = \{\Updownarrow (w1) ; \Updownarrow (r1, \hat{w}0, r0) ; \Updownarrow (w0, w0, w0, \hat{w}1) ; \Updownarrow (r1, w0, r0, \hat{w}1, r1)\}.$$

9.2 Forming Defect

For the forming defect test development, we follow the same procedure as for the interconnect and contact defects test development. First, we present a test for only the EtD faults. Second, we present a test for both the EtD and sHtD faults.

9.2.1 Test for EtD Faults. For this test, we again solve the ILP problem stated in Equation (6) for the under-forming and over-forming defect model in Table 5. It follows that the set $S_{EtD,forming} \in \{1r1, 0w1w0r0\}$ is able to sensitize and detect all EtD forming faults. These two can be included in a march algorithm in the following way with a test time of $3N_w + 2N_r$:

$$\text{March-EtD, forming} = \{\Updownarrow (w1, r1) ; \Updownarrow (w1, w0, r0)\}.$$

9.2.2 Test for EtD and Strong HtD Faults. The test development procedure for the EtD and sHtD faults sensitized by forming defects is similar to that of the faults sensitized by interconnection and contact defects. From Table 5 it follows that almost any S is able to sensitize all FCs for the forming defects. For example, the set $S_{EtD-sHtD,forming} \in \{1r1, 1w1, 0w1, 0w1w0\}$ is able to sensitize all forming defect FCs. For their detection, we again make use of the weak write DfT from [27]. After the weak write operation, a read operation needs to be performed in order to detect the defective cell. This results in the following march algorithm with a test time of $4N_w + 2N_r$:

$$\text{March-EtD-sHtd, forming} = \{\Updownarrow (w0, w1, \hat{w}0) ; \Updownarrow (r0) ; \Updownarrow (\hat{w}1, r1)\}.$$

10 DISCUSSION AND FUTURE DIRECTIONS

This paper presented a complete framework for defect and fault modeling, and test development in RRAMs. We can observe the following:

- **Realistic defect and fault models:** We have shown that inaccurate defect modeling leads to the sensitization of non-realistic faults that result in yield loss, as shown in Figure 2. Furthermore, because the defect is not modeled properly, defective chips will not be detected by the test, i.e., it leads to test escapes. We have shown that the usage of the DAT approach to model defects in RRAM devices results in accurate and realistic defect models. Hence, the faults that are validated using these defect models will be realistic as well, and thus improve test quality.
- **Applicability to other RRAM technologies:** The approach is also applicable to multi-level RRAM. For this, the FP notation should be extended so that all valid levels in the RRAM can be included, e.g., in the case of 2 bits per cell, by extending the S and F notation to contain the two additional states, and possible intermediate new 'U' states (see Figure 8). Then, the same approach can be followed to perform fault analysis and subsequent test development.

Based on the observations in this paper, we discuss the following about the future of RRAM test:

- **Modeling of other RRAM defects:** We have shown how the DAT approach can be used to model forming defects in RRAM. In Section 4, we presented a list of RRAM defects for which no suitable defect model is available today. Furthermore, it is expected that with further downscaling of RRAM technology new defects and faults will emerge, e.g., the HtD intermittent undefined state fault [16], or the effects of crystalline variations become more pronounced [31, 36]. Because these types of defects may only subtly affect the performance of the RRAM device and cannot be modeled using linear resistors, the DAT approach must be applied to model them and develop suitable tests.
- **Test solutions:** This work showed that many of the HtD faults that can occur in an RRAM are related to the analog nature of the RRAM device, where it stores a 'L', 'U', or 'H' value. We have shown that these values cannot be guaranteed to be detected by regular march operations alone, hence DfT schemes are required, e.g., the scheme in [27]. Future tests for RRAM should focus on efficiently and reliably detecting these faulty states, even in the presence of severe cycle-to-cycle variations. These variations make the test development more difficult, because they add randomness to the values that are stored in the cell.
- **Reliability:** RRAM devices have a limited cycle endurance. Devices that reach this limit typically show a reduced HRS/LRS window [63], meaning that some cells switch into the 'U' range. These cells need to be detected during the life time of the chip by some kind of built-in-self-test that reuses (parts of) the tests and DfT schemes that were developed for production testing. Future work should focus on developing tests for both fields.

11 CONCLUSION

In this paper we have presented a complete framework for defect and fault modeling, and test generation in RRAMs. We have presented an overview of all RRAM defects and classified them. We also have presented systematic approaches to define and validate the fault space. Then, we applied these approaches to validate the fault space for interconnection and contact defects that were modeled using traditional linear resistor defect modeling, and we applied it to RRAM unique forming defects that were modeled using the DAT approach as well. By comparing the results of these two defect models, it becomes clear that the traditional linear resistor models are unable to describe forming defects properly. Hence, a test based on the traditional models will lead to

test escapes. Next, we presented a test generation approach that is used to develop tests that can detect all validated faults for both defect modeling approaches in an efficient manner. Finally, we presented an outlook on the future of RRAM testing.

REFERENCES

[1] ASU. 2012. Predictive Technology Model (PTM). (2012). http://ptm.asu.edu/.

[2] H. Aziza, P. Canet, J. Postel-Pellerin, M. Moreau, J.-M. Portal, and M. Bocquet. 2017. ReRAM ON/OFF resistance ratio degradation due to line resistance combined with device variability in 28nm FDSOI technology. In *EUROSOI 2017*. IEEE, 35–38. https://doi.org/10.1109/ULIS.2017.7962594

[3] K. Beckmann, J. Holt, W. Olin-Ammentorp, Z. Alamgir, J. Van Nostrand, and N. C. Cady. 2017. The effect of reactive ion etch (RIE) process conditions on ReRAM device performance. *Semicond. Sci. Technol.* 32, 9 (Sept. 2017), 095013. https://doi.org/10.1088/1361-6641/aa7eed

[4] A. Benso, A. Bosio, S. Di Carlo, G. Di Natale, and P. Prinetto. 2005. Automatic march tests generation for static and dynamic faults in SRAMs. In *ETS'05*. 122–127. https://doi.org/10.1109/ETS.2005.8

[5] R. Bishnoi, L. Wu, M. Fieback, C. Münch, S. M. Nair, M. Tahoori, Y. Wang, H. Li, and S. Hamdioui. 2020. Special session–emerging memristor based memory and CIM architecture: Test, repair and yield analysis. In *2020 VTS*. IEEE, 1–10.

[6] Michael L. Bushnell and Vishwani D. Agrawal. 2000. *Essentials of Electronic Testing for Digital Memory & Mixed-Signal VLSI Circuits*. Springer Science+Business Media, New York, New York, USA.

[7] Umberto Celano, Ludovic Goux, Robin Degraeve, Andrea Fantini, Olivier Richard, Hugo Bender, Malgorzata Jurczak, and Wilfried Vandervorst. 2015. Imaging the three-dimensional conductive channel in filamentary-based oxide resistive switching memory. *Nano Lett.* 15, 12 (Dec. 2015), 7970–7975. https://doi.org/10.1021/acs.nanolett.5b03078

[8] Huan-Lin Chang, Hsuan-Chih Li, C. W. Liu, F. Chen, and M.-J. Tsai. 2011. Physical mechanism of HfO2-based bipolar resistive random access memory. In *2011 VTSA*. IEEE, 1–2. https://doi.org/10.1109/VTSA.2011.5872253

[9] Arjun Chaudhuri and Krishnendu Chakrabarty. 2018. Analysis of process variations, defects, and design-induced coupling in memristors. In *ITC 2018*. IEEE, 1–10.

[10] Ching Yi Chen, Hsiu Chuan Shih, Cheng Wen Wu, Chih He Lin, Pi Feng Chiu, Shyh Shyuan Sheu, and Frederick T. Chen. 2015. RRAM defect modeling and failure analysis based on march test and a novel squeeze-search scheme. *IEEE Trans. Comput.* 64, 1 (Jan. 2015), 180–190. https://doi.org/10.1109/TC.2014.12

[11] Pang-Shiu Chen, Yu-Sheng Chen, Heng-Yuan Lee, Tai-Yuan Wu, Pei-Yi Gu, Frederick Chen, and Ming-Jinn Tsai. 2012. Impact of flattened TiN electrode on the memory performance of HfO2 based resistive memory. *Electrochem. Solid-State Lett.* 15, 4 (Jan. 2012), H136. https://doi.org/10.1149/2.001205esl

[12] Yu-Sheng Chen, Wen-Hsing Liu, Heng-Yuan Lee, Pang-Shiu Chen, Sum-Min Wang, Chen-Han Tsai, Yen-Ya Hsu, Pei-Yi Gu, Wei-Su Chen, Frederick Chen, Chen-Hsin Lien, and Ming-Jinn Tsai. 2011. Impact of compliance current overshoot on high resistance state, memory performance, and device yield of HfOx based resistive memory and its solution. In *2011 VTSA*. IEEE, 1–2. https://doi.org/10.1109/VTSA.2011.5872252

[13] Yong-Xiao Chen and Jin-Fu Li. 2015. Fault modeling and testing of 1T1R memristor memories. In *VTS 2015*. IEEE, 1–6. https://doi.org/10.1109/VTS.2015.7116247

[14] Z. Fang, H. Y. Yu, X. Li, N. Singh, G. Q. Lo, and D. L. Kwong. 2011. $HfO_x/TiO_x/HfO_x/TiO_x$ zmultilayer-based forming-free RRAM devices with excellent uniformity. *IEEE Electron Device Letters* 32, 4 (2011), 566–568. https://doi.org/10.1109/LED.2011.2109033

[15] A. Fantini, L. Goux, R. Degraeve, D. J. Wouters, N. Raghavan, G. Kar, A. Belmonte, Y.-Y. Chen, B. Govoreanu, and M. Jurczak. 2013. Intrinsic switching variability in HfO2 RRAM. In *IMW 2013*. IEEE, 30–33. https://doi.org/10.1109/IMW.2013.6582090

[16] Moritz Fieback, Guilherme Cardoso Medeiros, Anteneh Gebregiorgis, Hassen Aziza, Mottaqiallah Taouil, and Said Hamdioui. 2021. Intermittent undefined state fault in RRAMs. In *2021 IEEE European Test Symposium (ETS)*. IEEE, 1–6.

[17] M. Fieback, S. Nagarajan, R. Bishnoi, M. Tahoori, M. Taouil, and S. Hamdioui. 2020. Testing scouting logic-based computation-in-memory architectures. In *2020 ETS*. IEEE, 1–6.

[18] Moritz Fieback, Mottaqiallah Taouil, and Said Hamdioui. 2018. Testing resistive memories: Where are we and what is missing? In *2018 ITC*. IEEE, 1–9. https://doi.org/10.1109/TEST.2018.8624895

[19] M. Fieback, L. Wu, G. C. Medeiros, H. Aziza, S. Rao, E. J. Marinissen, M. Taouil, and S. Hamdioui. 2019. Device-aware test: A new test approach towards DPPB level. In *2019 ITC*. IEEE, 1–10.

[20] B. Govoreanu, G. S. Kar, Y.-Y. Chen, V. Paraschiv, S. Kubicek, A. Fantini, I. P. Radu, L. Goux, S. Clima, R. Degraeve, N. Jossart, O. Richard, T. Vandeweyer, K. Seo, P. Hendrickx, G. Pourtois, H. Bender, L. Altimime, D. J. Wouters, J. A. Kittl, and M. Jurczak. 2011. 10 × 10nm2 Hf/HfOx crossbar resistive RAM with excellent performance, reliability and low-energy operation. In *2011 IEDM*. IEEE, 31.6.1–31.6.4. https://doi.org/10.1109/IEDM.2011.6131652

[21] A. Grossi, E. Nowak, C. Zambelli, C. Pellissier, S. Bernasconi, G. Cibrario, K. El Hajjam, R. Crochemore, J. F. Nodin, P. Olivo, and L. Perniola. 2016. Fundamental variability limits of filament-based RRAM. In *IEDM 2016*. IEEE, 4.7.1–4.7.4. https://doi.org/10.1109/IEDM.2016.7838348

[22] Alessandro Grossi, Eduardo Perez, Cristian Zambelli, Piero Olivo, and Christian Wenger. 2016. Performance and reliability comparison of 1T-1R RRAM arrays with amorphous and polycrystalline HfO2. In *EUROSOI 2016*. IEEE, 80–83. https://doi.org/10.1109/ULIS.2016.7440057

[23] Alessandro Grossi, Cristian Zambelli, Piero Olivo, Enrique Miranda, Valeriy Stikanov, Christian Walczyk, and Christian Wenger. 2016. Electrical characterization and modeling of pulse-based forming techniques in RRAM arrays. *Solid. State. Electron.* 115 (Jan. 2016), 17–25. https://doi.org/10.1016/J.SSE.2015.10.003

[24] Said Hamdioui. 2004. *Testing Static Random Access Memories: Defects, Fault Models and Test Patterns*. Vol. 26. Springer Science & Business Media.

[25] Said Hamdioui, Shahar Kvatinsky, Gert Cauwenberghs, Lei Xie, Nimrod Wald, Siddharth Joshi, Hesham Mostafa Elsayed, Henk Corporaal, and Koen Bertels. 2017. Memristor for computing: Myth or reality? In *DATE*. ACM, 722–731. https://dl.acm.org/citation.cfm?id=3130553.

[26] S. Hamdioui and M. Taouil. 2021. Device aware test for memory units. (2021). Patent No. NL 2023751.

[27] Said Hamdioui, Mottaqiallah Taouil, and Nor Zaidi Haron. 2015. Testing open defects in memristor-based memories. *IEEE Trans. Comput.* 64, 1 (Jan. 2015), 247–259. https://doi.org/10.1109/TC.2013.206

[28] Nor Zaidi Haron and Said Hamdioui. 2011. On defect oriented testing for hybrid CMOS/memristor memory. In *2011 ATS*. IEEE, 353–358. https://doi.org/10.1109/ATS.2011.66

[29] P. Huang, B. Chen, Y. J. Wang, F. F. Zhang, L. Shen, R. Liu, L. Zeng, G. Du, X. Zhang, B. Gao, J. F. Kang, X. Y. Liu, X. P. Wang, B. B. Weng, Y. Z. Tang, G.-Q. Lo, and D.-L. Kwong. 2013. Analytic model of endurance degradation and its practical applications for operation scheme optimization in metal oxide based RRAM. In *IEDM 2013*. IEEE, 22.5.1–22.5.4. https://doi.org/10.1109/IEDM.2013.6724685

[30] V. Iglesias, M. Porti, M. Nafría, X. Aymerich, P. Dudek, and G. Bersuker. 2011. Dielectric breakdown in polycrystalline hafnium oxide gate dielectrics investigated by conductive atomic force microscopy. *J. Vac. Sci. Technol. B, Nanotechnol. Microelectron. Mater. Process. Meas. Phenom.* 29, 1 (Jan. 2011), 01AB02. https://doi.org/10.1116/1.3532945

[31] Zizhen Jiang, Yi Wu, Shimeng Yu, Lin Yang, Kay Song, Zia Karim, and H.-S. Philip Wong. 2016. A compact model for metal-oxide resistive random access memory with experiment verification. *IEEE Trans. Electron Devices* 63, 5 (May 2016), 1884–1892. https://doi.org/10.1109/TED.2016.2545412

[32] Chul-Moon Jung, Jun-Myung Choi, and Kyeong-Sik Min. 2012. Two-step write scheme for reducing sneak-path leakage in complementary memristor array. *IEEE Trans. Nanotechnol.* 11, 3 (May 2012), 611–618. https://doi.org/10.1109/TNANO.2012.2188302

[33] A. Kalantarian, G. Bersuker, D. C. Gilmer, D. Veksler, B. Butcher, A. Padovani, O. Pirrotta, L. Larcher, R. Geer, Y. Nishi, and P. Kirsch. 2012. Controlling uniformity of RRAM characteristics through the forming process. In *IRPS 2012*. IEEE, 6C.4.1–6C.4.5. https://doi.org/10.1109/IRPS.2012.6241874

[34] Sachhidh Kannan, Jeyavijayan Rajendran, Ramesh Karri, and Ozgur Sinanoglu. 2013. Sneak-path testing of crossbar-based nonvolatile random access memories. *IEEE Trans. Nanotechnol.* 12, 3 (May 2013), 413–426. https://doi.org/10.1109/TNANO.2013.2253329

[35] Sachhidh Kannan, Jeyavijayan Rajendran, Ramesh Karri, and Ozgur Sinanoglu. 2013. Sneak-path testing of memristor-based memories. In *2013 ICVD/ICES*. IEEE, 386–391. https://doi.org/10.1109/VLSID.2013.219

[36] G. S. Kar, A. Fantini, Y.-Y. Chen, V. Paraschiv, B. Govoreanu, H. Hody, N. Jossart, H. Tielens, S. Brus, O. Richard, T. Vandeweyer, D. J. Wouters, L. Altimime, and M. Jurczak. 2012. Process-improved RRAM cell performance and reliability and paving the way for manufacturability and scalability for high density memory application. In *VLSIT 2012*. IEEE, 157–158. https://doi.org/10.1109/VLSIT.2012.6242509

[37] Kelin Kuhn, Chris Kenyon, Avner Kornfeld, Mark Liu, Atul Maheshwari, Wei-kai Shih, Sam Sivakumar, Greg Taylor, Peter VanDerVoorn, and Keith Zawadzki. 2008. Managing process variation in Intel's 45nm CMOS technology. *Intel J. Technol.* 12, 45 (2008), 77–85. https://doi.org/10.1535/itj.1201

[38] Kelin J. Kuhn, Martin D. Giles, David Becher, Pramod Kolar, Avner Kornfeld, Roza Kotlyar, Sean T. Ma, Atul Maheshwari, and Sivakumar Mudanai. 2011. Process technology variation. *IEEE Trans. Electron Devices* 58, 8 (Aug. 2011), 2197–2208. https://doi.org/10.1109/TED.2011.2121913

[39] M. Lanza, K. Zhang, M. Porti, M. Nafría, Z. Y. Shen, L. F. Liu, J. F. Kang, D. Gilmer, and G. Bersuker. 2012. Grain boundaries as preferential sites for resistive switching in the HfO2 resistive random access memory structures. *Appl. Phys. Lett.* 100, 12 (March 2012), 123508. https://doi.org/10.1063/1.3697648

[40] H. Y. Lee, Y. S. Chen, P. S. Chen, P. Y. Gu, Y. Y. Hsu, S. M. Wang, W. H. Liu, C. H. Tsai, S. S. Sheu, P. C. Chiang, W. P. Lin, C. H. Lin, W. S. Chen, F. T. Chen, C. H. Lien, and M.-J. Tsai. 2010. Evidence and solution of over-RESET problem for HfOX based resistive memory with sub-ns switching speed and high endurance. In *2010 IEDM*. IEEE, 19.7.1–19.7.4. https://doi.org/10.1109/IEDM.2010.5703395

[41] Seok-Hee Lee. 2016. Technology scaling challenges and opportunities of memory devices. In *IEDM 2016*. IEEE, 1.1.1–1.1.8. https://doi.org/10.1109/IEDM.2016.7838026

[42] Haitong Li, Peng Huang, Bin Gao, Bing Chen, Xiaoyan Liu, and Jinfeng Kang. 2014. A SPICE model of resistive random access memory for large-scale memory array simulation. *IEEE Electron Device Lett.* 35, 2 (Feb. 2014), 211–213. https://doi.org/10.1109/LED.2013.2293354

[43] Jiale Liang, Stanley Yeh, S. Simon Wong, and H.-S. Philip Wong. 2013. Effect of wordline/bitline scaling on the performance, energy consumption, and reliability of cross-point memory array. *ACM J. Emerg. Technol. Comput. Syst.* 9, 1 (Feb. 2013), 1–14. https://doi.org/10.1145/2422094.2422103

[44] E. Linn, R. Rosezin, S. Tappertzhofen, U. Böttger, and R. Waser. 2012. Beyond Von Neumann-logic operations in passive crossbar arrays alongside memory operations. *Nanotechnology* 23 (2012), 305205. https://doi.org/10.1088/0957-4484/23/30/305205

[45] Peng Liu, Jigang Wu, Zhiqiang You, Michael Elimu, Weizheng Wang, and Shuo Cai. 2018. Defect analysis and parallel march test algorithm for 3D hybrid CMOS-memristor memory. In *2018 ATS*. IEEE, 25–29. https://doi.org/10.1109/ATS.2018.00016

[46] Paolo Lorenzi, Rosario Rao, and Fernanda Irrera. 2013. Forming kinetics in HfO2 -based RRAM cells. *IEEE Trans. Electron Devices* 60, 1 (Jan. 2013), 438–443. https://doi.org/10.1109/TED.2012.2227324

[47] Qing Luo, Xiaoxin Xu, Tiancheng Gong, Hangbing Lv, Danian Dong, Haili Ma, Peng Yuan, Jianfeng Gao, Jing Liu, Zhaoan Yu, Junfeng Li, Shibing Long, Qi Liu, and Ming Liu. 2017. 8-Layers 3D vertical RRAM with excellent scalability towards storage class memory applications. In *IEDM 2017*. IEEE, 2.7.1–2.7.4. https://doi.org/10.1109/IEDM.2017.8268315

[48] Yichi Luo, Xiaole Cui, Mengyin Luo, and Qiujun Lin. 2017. A high fault coverage march test for 1T1R memristor array. In *EDSSC 2017*. IEEE, 1–2. https://doi.org/10.1109/EDSSC.2017.8126415

[49] MathWorks. 2019. MATLAB R2019b. (2019). https://mathworks.com/products/matlab.html.

[50] Enrique A. Miranda, Christian Walczyk, Christian Wenger, and Thomas Schroeder. 2010. Model for the resistive switching effect in HfO2 MIM structures based on the transmission properties of narrow constrictions. *IEEE Electron Device Lett.* 31, 6 (June 2010), 609–611. https://doi.org/10.1109/LED.2010.2046310

[51] Baker Mohammad, Maguy Abi Jaoude, Vikas Kumar, Dirar Mohammad Al Homouz, Heba Abu Nahla, Mahmoud Al-Qutayri, and Nicolas Christoforou. 2016. State of the art of metal oxide memristor devices. *Nanotechnol. Rev.* 5, 3 (Jan. 2016), 311–329. https://doi.org/10.1515/ntrev-2015-0029

[52] Seyed Nima Mozaffari, Spyros Tragoudas, and Themistoklis Haniotakis. 2015. Fast march tests for defects in resistive memory. In *2015 ISNA*. IEEE, 88–93. https://doi.org/10.1109/NANOARCH.2015.7180592

[53] Sanjoy Kumar Nandi, Xinjun Liu, Dinesh Kumar Venkatachalam, and Robert Glen Elliman. 2015. Effect of electrode roughness on electroforming in HfO2 and defect-induced moderation of electric-field enhancement. *Phys. Rev. Appl.* 4, 6 (Dec. 2015), 064010. https://doi.org/10.1103/PhysRevApplied.4.064010

[54] Clement Nguyen, Carlo Cagli, Gabriel Molas, Benoit Sklenard, Cecile Nail, Khalil El Hajjam, Jean-Francois Nodin, Christelle Charpin, Sophie Bernasconi, and Gilles Reimbold. 2017. Study of forming impact on 4Kbit RRAM array performances and reliability. In *IMW 2017*. IEEE, 1–4. https://doi.org/10.1109/IMW.2017.7939105

[55] Janice H. Nickel, John Paul Strachan, Matthew D. Pickett, C. Tom Schamp, J. Joshua Yang, John A. Graham, and R. Stanley Williams. 2013. Memristor structures for high scalability: Non-linear and symmetric devices utilizing fabrication friendly materials and processes. *Microelectron. Eng.* 103 (March 2013), 66–69. https://doi.org/10.1016/J.MEE.2012.09.007

[56] Dimin Niu, Yiran Chen, Cong Xu, and Yuan Xie. 2010. Impact of process variations on emerging memristor. In *(DAC'10)*. ACM Press, New York, New York, USA, 877. https://doi.org/10.1145/1837274.1837495

[57] Sung-Kye Park. 2015. Technology scaling challenge and future prospects of DRAM and NAND flash memory. In *IMW 2015*. IEEE, 1–4. https://doi.org/10.1109/IMW.2015.7150307

[58] Nagarajan Raghavan. 2014. Performance and reliability trade-offs for high-κ RRAM. *Microelectron. Reliab.* 54, 9–10 (Sept. 2014), 2253–2257. https://doi.org/10.1016/J.MICROREL.2014.07.135

[59] H. Sribhuvaneshwari and Suthendran Kannan. 2020. Enhanced test algorithm for nanoelectronic resistive random access memory testing using self check write scheme. *Analog Integr. Circuits Signal Process.* (Jan. 2020), 1–11. https://doi.org/10.1007/s10470-019-01576-x

[60] M. Stucchi, M. Bamal, and K. Maex. 2007. Impact of line-edge roughness on resistance and capacitance of scaled interconnects. *Microelectron. Eng.* 84, 11 (Nov. 2007), 2733–2737. https://doi.org/10.1016/J.MEE.2007.05.038

[61] Ad. J. Van de Goor and Z. Al-Ars. 2000. Functional memory faults: A formal notation and a taxonomy. In *VTS*. IEEE Comput. Soc., 281–289. https://doi.org/10.1109/VTEST.2000.843856

[62] Elena Ioana Vatajelu, Paolo Prinetto, Mottaqiallah Taouil, and Said Hamdioui. 2017. Challenges and solutions in emerging memory testing. *IEEE Trans. Emerg. Top. Comput.* (2017), 1–1. https://doi.org/10.1109/TETC.2017.2691263

[63] H.-S. Philip Wong, Heng-Yuan Lee, Shimeng Yu, Yu-Sheng Chen, Yi Wu, Pang-Shiu Chen, Byoungil Lee, Frederick T. Chen, and Ming-Jinn Tsai. 2012. Metal-oxide RRAM. *Proc. IEEE* 100, 6 (June 2012), 1951–1970. https://doi.org/10.1109/JPROC.2012.2190369

[64] Huaqiang Wu, Yan Liao, Bin Gao, Debanjan Jana, and He Qian. 2016. *RRAM Cross-Point Arrays*. Springer Netherlands, Dordrecht, 223–260. https://doi.org/10.1007/978-94-017-7512-0_8

[65] Huaqiang Wu, Xiao Hu Wang, Bin Gao, Ning Deng, Zhichao Lu, Brent Haukness, Gary Bronner, and He Qian. 2017. Resistive random access memory for future information processing system. *Proc. IEEE* 105, 9 (Sept. 2017), 1770–1789. https://doi.org/10.1109/JPROC.2017.2684830

[66] Lizhou Wu, Mottaqiallah Taouil, Siddharth Rao, Erik Jan Marinissen, and Said Hamdioui. 2018. Electrical modeling of STT-MRAM defects. In *2018 ITC*. IEEE, 1–10. https://doi.org/10.1109/TEST.2018.8624749

[67] Tony F. Wu, Binh Q. Le, Robert Radway, Andrew Bartolo, William Hwang, Seungbin Jeong, Haitong Li, Pulkit Tandon, Elisa Vianello, Pascal Vivet, Etienne Nowak, Mary K. Wootters, H. S. Philip Wong, Mohamed M. Sabry Aly, Edith Beigne, and Subhasish Mitra. 2019. 14.3 A 43pJ/cycle non-volatile microcontroller with 4.7μs shutdown/wake-up integrating 2.3-bit/cell resistive RAM and resilience techniques. In *2019 ISSCC*. IEEE, 226–228. https://doi.org/10.1109/ISSCC.2019.8662402

[68] Shimeng Yu and Pai-Yu Chen. 2016. Emerging memory technologies: Recent trends and prospects. *IEEE Solid-State Circuits Mag.* 8, 2 (2016), 43–56. https://doi.org/10.1109/MSSC.2016.2546199

[69] Weisheng Zhao, Mathieu Moreau, Erya Deng, Yue Zhang, Jean-Michel Portal, Jacques-Olivier Klein, Marc Bocquet, Hassen Aziza, Damien Deleruyelle, Christophe Muller, Damien Querlioz, Nesrine Ben Romdhane, Dafine Ravelosona, and Claude Chappert. 2014. Synchronous non-volatile logic gate design based on resistive switching memories. *IEEE Trans. Circuits Syst. I Regul. Pap.* 61, 2 (Feb. 2014), 443–454. https://doi.org/10.1109/TCSI.2013.2278332

Received September 2020; revised April 2021; accepted January 2022