Device-Aware Test: A New Test Approach Towards DPPB Level

Moritz Fieback, Lizhou Wu, Guilherme Cardoso Medeiros, Hassen Aziza, Siddharth Rao, Erik Jan Marinissen, Mottaqiallah Taouil, Said Hamdioui

Device-Aware Test: A New Test Approach Towards DPPB

Moritz Fieback[1] Lizhou Wu[1] Guilherme Cardoso Medeiros[1] Hassen Aziza[2]
Siddharth Rao[3] Erik Jan Marinissen[3] Mottaqiallah Taouil[1] Said Hamdioui[1]
[1]Computer Engineering Laboratory, Delft University of Technology, Mekelweg 4, 2628CD, Delft, The Netherlands
[2]IM2NP, UMR CNRS 7334, Aix-Marseille Université, 38 rue Joliot Curie, F-13451, Marseille, France
[3]IMEC, Kapeldreef 75, B-3001, Leuven, Belgium
Email: {M.C.R.Fieback, Lizhou.Wu, G.CardosoMedeiros, M.Taouil, S.Hamdioui}@tudelft.nl,
hassen.aziza@univ-amu.fr, {Siddarth.Rao, Erik.Jan.Marinissen}@imec.be

***Abstract*—This paper proposes a new test approach that goes beyond cell-aware test: device-aware test. The approach consists of three steps: defect modeling, fault modeling, and test/DfT development. The defect modeling does not assume that a defect in a device (or a cell) can be modeled electrically as a linear resistor (as the traditional approach suggests), but it rather incorporates and captures the impact of the physical defect on the technology parameters of the device and thereafter on its electrical parameters. Once the defective electrical model is defined, a systematic fault analysis (based on fault simulation) is performed to derive appropriate fault models and thereafter test solutions. The approach is demonstrated using two memory technologies: resistive random access memory (RRAM) and spin-transfer torque magnetic random access memory (STT-MRAM). The results show that the proposed approach is able to sensitize faults for defects that are not detected with the traditional approach, meaning that the latter cannot lead to high-quality test solutions as required for defective part per billion (DPPB) level. The new approach clearly sets up a turning point in testing at least for the considered two emerging memory technologies.**

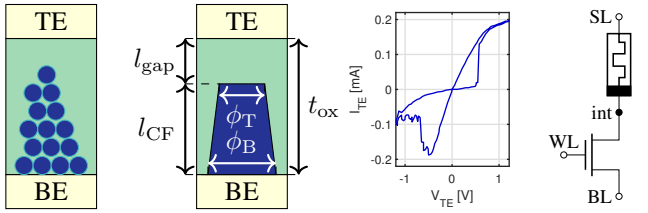
I. Introduction

Technology scaling has driven the phenomenal success of the semiconductor industry in delivering larger, faster, and cheaper integrated circuits with a high quality of service [1]. Silicon technology has entered the nano-era and $5\,\mathrm{nm}$ transistors are being prototyped [2]. However, it is widely recognized that defects and variability in device characteristics during the fabrication process, and their impact on the overall quality and reliability of the system represent major challenges, especially when considering high-quality levels, e.g., *defective part per billion* (DPPB) level [3]. Moreover, newly-emerging failure mechanisms in the nano-era are causing the fault mode of chips to be dominated by transient, intermittent, and weak faults rather than hard and permanent faults [4]. This shift in failure mechanisms may impact the way *fault modeling* has been done. Note that accurate fault models which reflect the real defects of new technologies are a must for developing high defect coverage test solutions. High-quality testing is a very critical step in the whole design and manufacturing chain responsible for screening out all the *defective* chips before they are sold, as it is the last chance to deliver the required quality and reliability to the end customer. All of these indicate the necessity and the importance of high-quality test solutions.

Testing defects in logic and memory chips underwent a long evolution process. For logic, early test methods were mainly functional and did not use any fault models. However, the increasing cost of such test approaches has led to the development of fault models (and hence structural testing) starting from the late 1970s. The most well-known fault models include stuck-at fault [5], transition fault [6, 7], and bridge fault [8, 9]. Despite the great success of these fault models, there was a clear need from the industry for new approaches and fault models (starting from late 1990s onwards) in order to reduce the increasing number of test escapes that customers were reporting. This led to the introduction of additional high-quality approaches and models such as stuck-short and stuck-open transistor models [10], N-detect [11], embedded multi-detect [12], and layout-aware fault modeling [13]. Moreover, the increasing demand of customers for higher quality has further led to the introduction of cell-aware test [14, 15]; it assumes that many escapes during testing are due to defects within a standard library cell, and therefore models defects as linear resistors (opens, shorts) at or between the terminals of each device within the library cell.

Testing memories went also through a quite similar revolution. The early memory tests (typically before 1980) can be classified as ad-hoc tests due to the absence of formal fault models and proofs [16]; they have the property of having a low defect/fault coverage and a very long test time, typically in the order of $O\left(n^2\right)$, which made them unpractical for increasing memory sizes. During the early 1980s, many memory fault models have been introduced, allowing the fault coverage of a certain test to be provable while the test time is usually in order $O\left(n\right)$; i.e., linear with the size of the memory. Some important fault models introduced in that time were stuck-at faults and address-decoder faults [17]. These are abstract fault models not based on any actual memory design or real defects. In the late 1990s, experimental results based on DPPM screening of a large number of tests applied to a large number of memory chips indicated that many detected faults cannot be explained with the well-known fault models [18, 19], which suggested the existence of additional faults. This stimulated the introduction of new fault models (both static and dynamic) based on linear resistor defect injection and

978-1-5386-3413-4/17/$31.00



(a) Ox. vacancies (b) Tech. parameters (c) I-V curve (d) 1T-1R cell

Fig. 1: RRAM device technology.

SPICE simulation [20, 21]: read destructive fault, write disturb fault, transition coupling fault, read destructive coupling fault, etc. Note that the cell-aware test fault modeling approach is quite similar to this as it also models defects as linear resistors (opens and shorts) at the terminals of devices in each memory cell. The above clearly shows that testing of both logic and memory assumes that physical defects in devices can be be modeled as linear resistors. Although it can be convincing for modeling opens and shorts in interconnects, this assumption has never been validated for devices. In addition, it is well known that scaling below $10\,\mathrm{nm}$ is giving rise to many device failure mechanisms that cannot be modeled with linear resistors. Moreover, it has been shown recently that this assumption is inaccurate for emerging technologies such as resistive random access memory (RRAM) [22] and spin-transfer torque magnetic random access memory (STT-MRAM) [23], and may lead to wrong fault models. Hence, it cannot lead to high-quality test solutions. This has inspired us to develop a new *device-aware test* (DAT) approach which is the topic of this paper.

This paper introduces device-aware testing which takes cell-aware testing one step further. Instead of using a fault model derived from injecting linear resistors in transistor-level netlists, DAT first changes the electrical model of the defective device (e.g., transistor) by incorporating the impact of the defect in the device's electrical parameters (model); these are then used to perform circuit simulation to derive the fault models and thereafter test solutions. In this paper, we introduce and demonstrate DAT for two emerging memory technologies, namely RRAM and STT-MRAM. The main contributions of the paper are as follows.

- We introduce the three-step DAT approach: defect modeling, fault modeling, and test development. One of the key differentiators is the defect modeling step which takes the physical defects into consideration and captures their impact on the electrical parameters, hence enabling accurate fault modeling. The latter systematically defines the complete (theoretical) memory fault space and thereafter systematically performs the fault analysis (using defect modeling of the first step and circuit simulation) to validate the space. This step provides insight information not only on the nature of realistic faults, but also about the best way to test them, which is used in the third step of DAT (test development). As an example, a fault resulting in a wrong read value can be easily detected with a March test as it is able to sensitize the fault, while a fault resulting in a random read value needs special design-for-testability (DfT) to guarantee its detection.
- We apply DAT on RRAM and STT-MRAM and demonstrate the superiority of this approach as compared to conventional memory test approaches. DAT can model and detect some of the device defects that cannot be detected by the conventional approach. Hence, it can further reduce the amount of test escapes and can even better diagnose defects for fast yield learning.

TABLE I: RRAM key parameters.

Technology 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	$t_{H\to L}$	HRS to LRS switching delay
		$t_{L\to H}$	LRS to HRS switching delay

The rest of the paper is organized as follows. Section II provides brief background information on the operating principles of RRAM and STT-MRAM, respectively, as they will be used for the validation of DAT approach. Section III gives a complete view of the DAT methodology; each of the three steps is described in detail. Section IV selects the "forming defect" (representing a defect in an RRAM device) and applies the three steps of DAT approach; not only in order to show how the approach works, but also to validate its superiority. Section V does the same by then by selecting "pinhole defect" for STT-MRAM. Section VI discusses the advantages and limitations of the method and concludes the paper.

II. Technology Background

This section provides the technical working principles of RRAM and STT-MRAM, respectively.

A. RRAM Fundamentals

Resistive random access memory (RRAM) is an emerging non-volatile memory technology that uses oxide-based memristors to store data [24]. The production of the RRAM devices can be integrated in the back-end-of-line (BEOL) of a standard CMOS process [24].

The RRAM device is schematically shown in Fig. 1a. It consists of two electrodes (top (TE) and bottom electrode (BE)) and a metallic-oxide between them. By applying a positive voltage to the TE that is higher than the set threshold (V_{set}), bonds between the metal and oxygen ions are broken and the oxygen ions are attracted to the TE, leaving behind a chain of oxygen vacancies, a conductive filament (CF). The device is now in its low resistive state R_{LRS} (i.e., 'set' representing logic '1') . If a negative voltage is applied that is lower than the reset threshold (V_{reset}), the ions move back to fill the vacancies, bringing the device in its high resistive state R_{HRS} (i.e., 'reset' representing logic '0'). The size of the CF determines the resistance of the device; wider CFs result in lower resistance and longer CFs result in higher resistance. Fig. 1b and Table I show the technology parameters that determine the resistance of the RRAM device. Its resistance has an analogue nature, i.e., it can take any value within a certain range. Fig. 1c shows the switching behavior of the device, both the difference in conduction between the 'set'

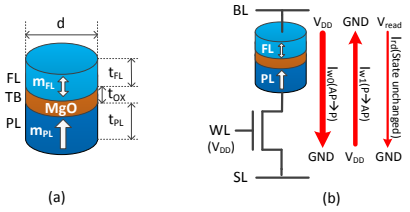


Fig. 2: (a) Simplified MTJ device organization, (b) 1T-1MTJ cell.

and 'reset' state, as well as the switching thresholds between them. From the plot, it becomes clear that the RRAM device is a non-linear device due to its hysteresis.

Multiple cell designs exist for RRAMs, the most common of them are the 1T-1R and 1R design. The 1T-1R cell is depicted in Fig. 1d. By applying appropriate voltages to the bit line (BL), word line (WL), and select line (SL), the state of the device can be changed. The transistor controls the current flow through the RRAM. A 1R design does not have an access transistor, which has the benefit of smaller cell designs, but also the drawback that sneak-paths exist that unwantedly couple multiple cells [25].

B. STT-MRAM Fundamentals

The *magnetic tunnel junction* (MTJ) is the core of STT-MRAM, as it is the data-storing element. As shown in Fig. 2a, an MTJ device is composed of two ferromagnetic layers sandwiching an ultra-thin insulating MgO layer called *tunnel barrier* (TB). The top ferromagnetic layer is called *free layer* (FL), its magnetization can be switched by a spin-polarized current flowing through it. There are several key technology parameters that significantly impact the STT-induced switching behavior for the magnetization in the FL. They are the *saturation magnetization* M_s and the *magnetic anisotropy field* H_k of the FL, and the *potential barrier height* $\bar{\varphi}$ of the TB [23]. In contrast, the magnetization in the bottom ferromagnetic layer is pinned to a certain direction. Therefore, the bottom layer is usually referred to as *pinned layer* (PL). Due to the tunneling magneto-resistance (TMR) effect [26], the MTJ's resistance is low when the magnetization in the FL is parallel to that in the PL. The resistance is high when in anti-parallel configuration. The TMR ratio is defined by: $TMR{=}(R_{AP}{-}R_P)/R_P$, where R_{AP} and R_P are the resistances in the anti-parallel and parallel states, respectively. To evaluate the resistivity of MTJ devices, the *resistance-area* (RA) product is commonly used in the MRAM community, as it is independent of the device size. In summary, RA, TMR, $\bar{\varphi}$, M_s, and H_k are critical technology parameters of the MTJ device, as listed in Table II.

Fig. 2b shows the most widely-adopted STT-MRAM cell design, namely the bottom-pinned 1T-1MTJ cell, and its corresponding control voltages during write and read operations. The cell includes an MTJ device and an NMOS selector; it has three terminals similar to RRAM, as illustrated in the figure. For STT-MRAMs, R_P, R_{AP}, I_c, and t_w are four key electrical parameters determining the electrical behavior of MTJ devices [23]. I_c is the critical switching current, and t_w is the average switching time.

TABLE II: STT-MRAM key parameters.

Technology Parameters		Electrical Parameters	
RA	Resistance-area product	R_P	Resistance in P state
TMR	Tunneling magneto-resistance ratio	R_{AP}	Resistance in AP state
H_k	Magnetic anisotropy field of the FL	I_c	Critical switching current
M_s	Saturation magnetization of the FL	t_w	Average switching time
$\bar{\varphi}$	potential barrier height of the TB		

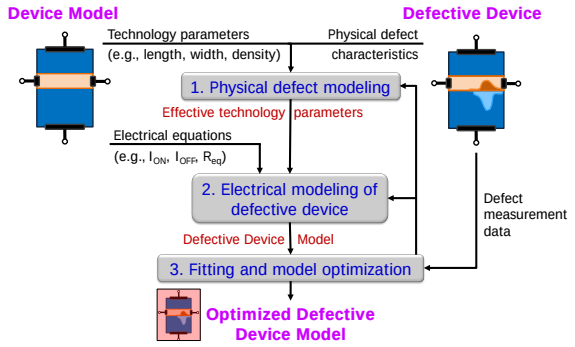


Fig. 3: Generic device defect modeling flow.

III. Device-Aware Test

Traditional memory testing assumes that a device defect can be modeled as a linear resistor in series or in parallel with the device. However, it has been shown that this approach is not accurate at least for emerging memory technologies such as RRAM [22] and STT-MRAM [23], resulting in incomplete or inaccurate fault modeling; hence escapes. Device-Aware Test (DAT) aims at solving this problem, and setting up a step toward meeting DPPB-level requirement. First, the device defects are physically modeled and their electrical behaviors are incorporated into device models. Second, the model is integrated in a memory simulation platform in order to analyze the impact of the defect on memory behavior; this is done in a systematic manner by validating a pre-defined fault framework/space using SPICE Simulation. The results of this step provide insights on the nature of realistic faults, which is used in order to develop optimal and appropriate test solutions (e.g., March test, DfT). Next, these three steps are described in detail. Note that these steps will be applied in Section IV and Section V to RRAM and STT-MRAM, respectively.

A. Device Defect Modeling

Inaccurate defect modeling may result in poor fault models, thereby limiting the effectiveness of proposed test solutions and DfT designs, not only in terms of defect coverage but also in terms of test time. For example, a test targeting a fault model that does not represent any real defect will not increase the defect coverage while still consuming test time. To accurately model physical defects, the device model should incorporate the way the defect impacts the technology parameters (e.g., length, width, density) and thereafter the electrical parameters (e.g., the critical switching current) of the device [23]; this is exactly what device defect modeling does. Fig. 3 shows the flow of such modeling approach; its inputs are 1) the electrical model of a device, and 2) the defect under investigation. The output is an optimized (parameterized) model of the defective device. Note that in general a device can be a FinFET

transistor, an STT-MRAM device, an RRAM device, a PCM device, etc. The approach consists of three steps:

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

$$Tp_\mathrm{eff}(\boldsymbol{S_i}) = f_i(Tp_\mathrm{df}, \boldsymbol{S_i}) \quad (1)$$

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

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

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

B. Fault Modeling

The second DAT step is fault modeling. In this step, the defect models from the previous step are used to analyze the behavior of a memory in the presence of defects. The results from this analysis are used to develop a high-quality test. First, we define the *fault space* that describes *all possible* faults, and classify them. Second, we present the *fault analysis methodology* that determines which faults from the space are *realistic* for the defect under consideration; i.e., which faults can be sensitized in the presence of such a defect.

1) Fault Space and Classification: In this work, we limit the analysis to static and dynamic single-cell faults [27]. A static fault is defined as a fault that can be sensitized by performing at most one operation, while a dynamic fault is sensitized by more than one operation. If more than one cell is involved in the fault, the fault is called a coupling fault. A strong fault can be systematically described using the fault

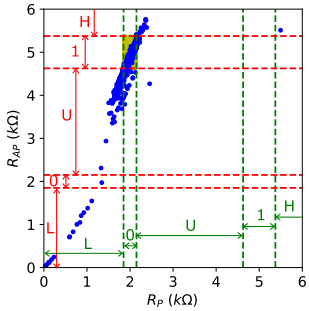


Fig. 4: Measured resistance distribution of R_P and R_AP for $\varnothing 60\,\mathrm{nm}$ MTJ devices, suggesting the existence of states 'L', '0', 'U', '1', and 'H'.

primitive (FP) notation [27]. An FP describes the difference between the observed and expected memory behavior, denoted as a three-tuple $\langle S/F/R \rangle$ where:

- S denotes the operation sequence that sensitizes the fault. A sequence takes the form of $S{=}x_0 O_1 x_1 \ldots O_n x_n$, where $x{\in}\{0,1\}$ and $O{\in}\{\mathrm{r}, \mathrm{w}\}$; 0 and 1 denote logical cell values, r and w denote a reading and a writing operation. If $n \leq 1$, the fault is static, else dynamic.
- F describes the value that is stored in the cell after S is performed. For traditional charge-based memories, e.g., SRAM, there exist only two digital states, i.e., $F{\in}\{0,1\}$ [27]. However, emerging memory technologies like RRAM and STT-MRAM use a resistive storage element; pre-defined resistance ranges determine the logic state of the cell. Due to defects or extreme process variations, the state of such devices can be be outside these ranges, hence the need of the definition of other (faulty) resistance states. Fig. 4 presents the measured resistance distribution of a large number of $\varnothing 60\,\mathrm{nm}$ MTJ devices; it shows that $F{\in}\{0, 1, \mathrm{U}, \mathrm{L}, \mathrm{H}\}$, as will be explained next. Each point in the figure represents a device whose R_P is shown on the x-axis and R_AP on the y-axis. From a design perspective, the nominal R_P is $2\,\mathrm{k}\Omega$ and the nominal R_AP is $5\,\mathrm{k}\Omega$; this assures a good read reliability with $TMR = 150\%$. A 3σ of the nominal values is used to define the resistance ranges of the two states 0 and 1. As shown in the figure, the points inside the shaded box represent good devices in accordance with the above design specifications. However, there is also a large number of devices outside the specification due to some defects or extreme process variations. These are: (1) extreme low resistance state 'L', 2) extreme high resistance state 'H', and 3) undefined state 'U'. Note that the definitions of states '0' and '1' for STT-MRAM differ from RRAM, where state '0' stands for high resistance while '1' for low resistance. Measurement data suggesting the existence of the five states for RRAM can be found in [25, 28, 29].
- R describes the output of a read operation if S is a read operation. $R \in \{0, 1, ?, -\}$ where $?$ denotes a random read value (e.g., the sensing current is very close to sense amplifier reference current), and '$-$' denotes that R is not applicable, i.e., when S is a write operation.

TABLE III: Single-cell static fault primitives.

#	S	F	R	FP notation	Name	#	S	F	R	FP notation	Name
1	1	0	-	⟨1/0/-⟩	$SF1_0$	27	1r1	1	?	⟨1r1/1/?⟩	$RRF1_1$
2	0	1	-	⟨0/1/-⟩	$SF0_1$	28	0r0	0	?	⟨0r0/0/?⟩	$RRF0_0$
3	1	U	-	⟨1/U/-⟩	$SF1_U$	29	1r1	0	0	⟨1r1/0/0⟩	$IRDF1_0$
4	0	U	-	⟨0/U/-⟩	$SF0_U$	30	0r0	1	1	⟨0r0/1/1⟩	$IRDF0_1$
5	1	L	-	⟨1/L/-⟩	$SF1_L$	31	1r1	U	0	⟨1r1/U/0⟩	$IRDF1_U$
6	0	L	-	⟨0/L/-⟩	$SF0_L$	32	0r0	U	1	⟨0r0/U/1⟩	$IRDF0_U$
7	1	H	-	⟨1/H/-⟩	$SF1_H$	33	1r1	L	0	⟨1r1/L/0⟩	$IRDF1_L$
8	0	H	-	⟨0/H/-⟩	$SF1_H$	34	0r0	L	1	⟨0r0/L/1⟩	$IRDF0_L$
9	1w0	1	-	⟨1w0/1/-⟩	$WTF1_1$	35	1r1	H	0	⟨1r1/H/0⟩	$IRDF1_H$
10	0w1	0	-	⟨0w1/0/-⟩	$WTF0_1$	36	0r0	H	1	⟨0r0/H/1⟩	$IRDF0_H$
11	1w0	U	-	⟨1w0/U/-⟩	$WTF1_U$	37	1r1	0	1	⟨1r1/0/1⟩	$RDF1_0$
12	0w1	U	-	⟨0w1/U/-⟩	$WTF0_U$	38	0r0	1	0	⟨0r0/1/0⟩	$RDF0_1$
13	1w0	L	-	⟨1w0/L/-⟩	$WTF1_L$	39	1r1	U	1	⟨1r1/U/1⟩	$RDF1_U$
14	0w1	L	-	⟨0w1/L/-⟩	$WTF0_L$	40	0r0	U	0	⟨0r0/U/0⟩	$RDF0_U$
15	1w0	H	-	⟨1w0/H/-⟩	$WTF1_H$	41	1r1	L	1	⟨1r1/L/1⟩	$RDF1_L$
16	0w1	H	-	⟨0w1/H/-⟩	$WTF0_H$	42	0r0	L	0	⟨0r0/L/0⟩	$RDF0_L$
17	1w1	0	-	⟨1w1/0/-⟩	$WDF1_0$	43	1r1	H	1	⟨1r1/H/1⟩	$RDF1_H$
18	0w0	1	-	⟨0w0/1/-⟩	$WDF0_1$	44	0r0	H	0	⟨0r0/H/0⟩	$RDF1_H$
19	1w1	U	-	⟨1w1/U/-⟩	$WDF1_U$	45	1r1	0	?	⟨1r1/0/?⟩	$RRDF1_0$
20	0w0	U	-	⟨0w0/U/-⟩	$WDF0_U$	46	0r0	1	?	⟨0r0/1/?⟩	$RRDF0_1$
21	1w1	L	-	⟨1w1/L/-⟩	$WDF1_L$	47	1r1	U	?	⟨1r1/U/?⟩	$RRDF1_U$
22	0w0	L	-	⟨0w0/L/-⟩	$WDF0_L$	48	0r0	U	?	⟨0r0/U/?⟩	$RRDF0_U$
23	1w1	H	-	⟨1w1/H/-⟩	$WDF1_H$	49	1r1	L	?	⟨1r1/L/?⟩	$RRDF1_L$
24	0w0	H	-	⟨0w0/H/-⟩	$WDF0_H$	50	0r0	L	?	⟨0r0/L/?⟩	$RRDF0_L$
25	1r1	1	0	⟨1r1/1/0⟩	$IRF1_1$	51	1r1	H	?	⟨1r1/H/?⟩	$RRDF1_H$
26	0r0	0	1	⟨0r0/0/1⟩	$IRF0_0$	52	0r0	H	?	⟨0r0/H/?⟩	$RRDF0_H$

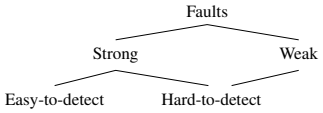


Fig. 5: Fault classification.

Table III lists all single-cell static FPs and their names. The naming of the FPs follows the following scheme:

$$\text{FP} = \{\textit{read impact}\}\,\{\textit{behavior}\}\,\{\textit{initial}\}_{\{F\}} \qquad (2)$$

Here, *read impact* is applicable only if a read sensitizing operation results in a faulty read: incorrect (I), or random (R) read values. *behavior* describes the behavior of the faulty cell: it specifies the nature of the operation (read (R) or write (W)) and the resulting fault effect (destructive (D), transition (T), or nothing). For example, 'WDF' means write destructive fault. *initial* describes the initial state of the cell and F is the value that is stored in the cell after S is performed. For example, $RRDF0_1$=⟨0r0/1/?⟩ is the Random Read Destructive Fault that puts the cell into '1' during a read '0', and returns a random read value at the output. State faults are an exception to this scheme because no sensitizing operation is performed. Their names follow this scheme: $\text{FP} = \text{SF}\,\{\textit{initial}\}_{\{F\}}$. For dynamic faults, the name of an FP gets the prefix $nd-$ where n denotes the number of operations in S and the FP is based on the last operation in S, e.g., ⟨1r1w0/L/-⟩ is 2d-WTF1_L.

Memory faults can be classified, as shown in Fig. 5, into two types: *strong* and *weak* faults. Strong faults are functional faults that can always be sensitized (and may be detected) by applying a sequence of operations and can cause functional errors; e.g., all FPs of Table III are strong faults. In contrast, weak faults do not result in FPs, but they cause parametric faults, e.g., a reduction in bit line current during a read operation. Note that these faults cannot be detected with any sequence of operations as they do not cause any functional errors. Obviously, these faults need to be also detected as they cause reliability problems (e.g., shorter lifetime, higher in-field failure rate). Depending on the effort needed to detect them, faults can be further divided into easy-to-detect and hard-to-detect faults. The detection of easy-to-detect faults can be simply *guaranteed* by applying write and read operations, e.g., by using a March test. The detection of hard-to-detect faults, however, *cannot* be guaranteed by just March tests and their detection requires additional effort; e.g., use of a special circuity such as DfT. Note that strong faults consist of easy-to-detect and hard-to-detect faults, while weak faults are all hard-to-detect. Examples of strong hard-to-detect faults are random read faults such as $RRF1_1$ and $RRF0_0$. For example, in an STT-MRAM with a small defect, the bit-line current during a read may be very close to the reference current of the sense amplifier causing random behaviour between devices.

2) Fault Analysis Methodology: Once the defect is modeled and the framework of faults is defined, the validation of the faults can be performed using a systematic circuit simulation approach. In this paper we will restrict ourselves to single-cell fault analysis as the case studies we will show for RRAM and STT-MRAM involve single-cell defects. Our fault analysis consists of seven steps [27]: 1) circuit generation, 2) defect injection, 3) stimuli generation, 4) circuit simulation, 5) fault analysis, 6) fault primitives identification, and 7) defect size sweeping and repetition of steps 2 to 6 till all sizes are covered. Note that in our case, defect injection means changing the electrical model of the device (e.g., RRAM or STT-MRAM) with the defective device model obtained in step 1 of DAT, while defect size sweeping means changing the size of the defect which also modifies the electrical parameters of the defective device model. Fig. 6 shows the methodology of fault analysis that enables us to get more insight on the nature of realistic faults and the way to test them. Given a list of defects and ranges of their sizes, the seven steps of the fault analysis should be first performed for the validation of *static* single-cell FPs of Table III (i.e., $n{\leq}1$). The result will be a set of FPs classified into easy-to-detect faults and hard-to-detect faults, associated with the size/range of the defect/parameters. In case that no FP is sensitized in the presence of a defect, the fault is considered to be weak and it is added to the list of hard-to-detect faults. Next, all defects that resulted in hard-to-detect faults will be further analyzed, but then using dynamic fault analysis, starting at $n{=}2$. Some defects leading to hard-to-detect faults can trigger now easy-to-detect faults; e.g., S=0w0 causes a weak fault, while S=0w0w0 causes an easy-to-detect strong fault. Once 2 operation single-cell dynamic fault analysis is done, we can redo similar analysis for $n{=}3$ for defects that resulted in hard-to-detect faults. The process can be repeated by extending S each time with one operation till the considered $n_{\max}$ is reached. Each step in the process aims at reducing the hard-to-detect fault set and increasing the easy-to-detect fault set; this is an important step towards not only optimizing test cost but also towards improving the overall product quality. The final results will be a set of faults that can be easily detected by the generation of March tests, and another set that needs special attention in order to guarantee their detection (e.g., DfT, special tests, etc.).

C. Test Development

The results of the fault analysis facilitate the development of high-quality and efficient test solutions. All easy-to-detect faults can be detected by applying appropriate test algorithms.

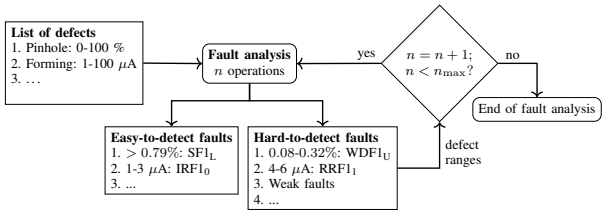


Fig. 6: Fault analysis methodology.

Developing optimized algorithms starts by identifying the minimal detection conditions for each of the faults and thereafter compiling them in test algorithms. This will guarantee 100% fault coverage of easy-to-detect faults at minimal test cost. One can also incorporate a DfT scheme in order to further optimize the test time; e.g., a DfT that enables the test of many faults simultaneously, parallel testing, etc.

Hard-to-detect faults, however, require special attention. Special DfT schemes and tests may be required. Examples are: DfT schemes that may directly measure the bit line swing, modify the operation conditions such as weak write operations [28], stress tests, etc. The aim is to *maximize* the fault coverage for these faults while keeping the test cost economically affordable.

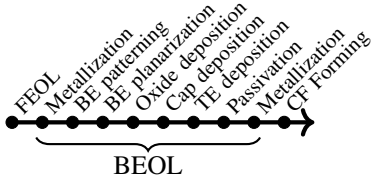
IV. Device-Aware Test for RRAM

In this section we apply the DAT approach on RRAM by following the three steps discussed in the previous section. However, first we describe RRAM manufacturing defects and select a representative defect.

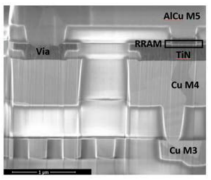
A. RRAM manufacturing defects

The fabrication process flow of an RRAM is depicted in Fig. 7a [22] and their associated defects are listed in Table IV; a more detailed overview can be found in [22]. The process starts with manufacturing transistors on the wafer in the front-end-of-line (FEOL) production phase. Then, the lower metal interconnection layers are deposited in the BEOL phase. RRAM devices are typically constructed between two metal layers (e.g., M4 and M5) as depicted in Fig. 7b [30]. After the devices are fabricated, the remaining metal layers are deposited. The devices do not have a conductive filament (CF) yet, therefore an initial CF forming step needs to be performed in order to achieve a functional device. In this paper we focus on defects that result from this step.

During the forming step, an initial CF is generated in the RRAM device's oxide. The conditions of this step have a strong impact on the performance of the device and therefore this step may result in defects. A few observations of the forming conditions can be made: higher forming currents (I_{form}) result in lower device resistance with less variation [30, 31], and variations in the forming current lead to more resistive variations [32]. Variations in the device geometry and oxide defect density also affect the forming step [33]. Two defects can result from the forming step: overforming, when the CF is too large, and non-forming, when no or only a tiny CF is formed.



(a) Processing flow [22, 30, 33].



(b) Cross-section TEM [30].

Fig. 7: General manufacturing process of RRAM.

TABLE IV: RRAM defect classification [22].

FEOL	BEOL	
Transistor	Interconnection	Memristor
patterning proximity	opens	electrode roughness
line roughness	shorts	polish variations
polish variations	line roughness	varying defect density
dielectric variations		dimensional variations
random dopants		material redeposition
anneal		*overforming*
strain		*non-forming*
gate granularity		

B. Forming Defect Modeling

In this section, we model the forming defect using DAT approach and show how the conventional approach use linear resistor to model the defect.

1) DAT Approach: For the DAT approach, we relate the input parameters of the RRAM device model (such as in [34]) to the forming current, thus incorporating the physics of the forming step, that could result in overforming or non-forming, into the electrical model. The model can be included in a netlist to observe its electrical effects.

Physical defect analysis and modeling. The forming current is directly related to the shape of the CF, i.e., it affects the key parameters shown in Fig. 1b. It is shown that l_{CF} and ϕ_{T} have the strongest impact on the resistance of the device [31]. Therefore, these parameters are used to model the forming effects of the device. To include the stochastic variation of the l_{CF}, an additional parameter Δl_{CF} (that sets the strength of this variation) is included. These parameters are used to model the forming effects of the device. The physical defect modeling step can be denoted mathematically as:

$$l_{\mathrm{CF,eff}}\left(I_{\mathrm{form}}\right) = a_1 \exp\left(b_1 \cdot R_\mu\left(I_{\mathrm{form}}\right)\right) + c_1 \exp\left(d_1 \cdot R_\mu\left(I_{\mathrm{form}}\right)\right), \tag{3}$$

$$\phi_{\mathrm{T,eff}}\left(I_{\mathrm{form}}\right) = a_2 \exp\left(b_2 \cdot R_\mu\left(I_{\mathrm{form}}\right)\right) + c_2 \exp\left(d_2 \cdot R_\mu\left(I_{\mathrm{form}}\right)\right), \tag{4}$$

$$\Delta l_{\mathrm{CF,eff}}\left(I_{\mathrm{form}}\right) = a_3 \exp\left(b_3 \cdot R_\sigma\left(R_\mu\right)\right) + c_3 \exp\left(d_3 \cdot R_\sigma\left(R_\mu\right)\right). \tag{5}$$

Here, a_k, b_k, c_k and d_k ($k \in \{1, 2, 3\}$) are fitting parameters. $R_\mu\left(I_{\mathrm{form}}\right) = f\left(I_{\mathrm{form}}\right)$, where $f\left(I_{\mathrm{form}}\right)$ is a cubic Hermite interpolation of I_{form} to the median resistance in [30], and $R_\sigma\left(R_\mu\right)$ is given by Equation (1) in [30].

Electrical modeling of the defective device. The RRAM device model in [34] takes l_{CF}, ϕ_{T}, and Δl_{CF} as input parameters. These three parameters dictate the switching behavior and the resistance of the RRAM device, and thus are well suited to model the effects of forming on the device's electrical behavior. When the resulting model is simulated in a netlist, the effects on the electrical parameters, e.g., resistance, switching speed and thresholds, can be analyzed.

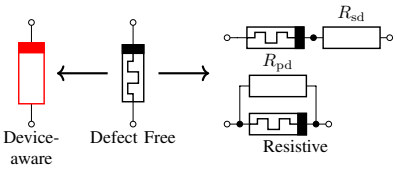


Fig. 8: Device-aware and resistive defective device models.

TABLE V: Faults sensitization using device-aware vs resistive defect model.

FP #	DAT	Conv.	FP #	DAT	Conv..	FP #	DAT	Conv.	FP #	DAT	Conv.
1	no	no	14	no	yes	27	no	no	40	no	no
2	no	no	15	yes	no	28	no	no	41	no	yes
3	yes	no	16	yes	yes	29	no	no	42	no	no
4	no	no	17	no	no	30	no	no	43	no	no
5	no	no	18	no	no	31	no	no	44	yes	no
6	no	no	19	yes	no	32	no	no	45	no	no
7	no	no	20	no	no	33	no	no	46	no	no
8	yes	no	21	no	yes	34	no	no	47	no	no
9	no	yes	22	no	no	35	no	no	48	no	no
10	no	yes	23	no	no	36	no	no	49	no	no
11	no	yes	24	yes	yes	37	no	no	50	no	no
12	no	yes	25	no	yes	38	no	no	51	no	no
13	no	no	26	no	yes	39	yes	no	52	no	no

Fitting and model optimization. In this step, the three alterable parameters are calibrated so that the defective behavior of the RRAM device corresponds with measurements of real devices such as in [30]. To realize this, we first analyze the influence of l_{CF} and ϕ_{T} on the mean resistance. These parameters are then fitted against the measurements in [30] and thus are linked to I_{form}. The effect of Δl_{CF} is analyzed and fitted in a similar fashion. We vary I_{form} between $5\,\mu\mathrm{A}$ and $34.1\,\mu\mathrm{A}$.

2) Conventional Approach: The conventional resistive defect modeling approach models the forming defect as a resistor that is either parallel (R_{pd}) or in series (R_{sd}) with a defect-free RRAM device. The difference with the device-aware defect models is shown in Fig. 8. The strength of a resistive defect is represented by the resistance value; we sweep the resistance of both R_{pd} and R_{sd} from $1\,\Omega$ and $100\,\mathrm{M}\Omega$ in our simulations.

C. Fault Modeling

This step consists of fault analysis based on the use of the electrical models generated in the first step. We start with the static fault analysis for both defect models. Next, we analyze the dynamic faults for the DAT and traditional approach.

As a forming defect impacts a single RRAM device (cell, see Fig. 1d), we only analyze single-cell faults. The possible single-cell static faults are those listed in Table III; the dynamic fault space can be constructed by following the definitions in Section III-B.

We start the fault analysis by analyzing static faults. Table V lists the static faults that were sensitized both with the DAT approach as well as the conventional (conv.) approach for all I_{form}, R_{pd}, and R_{sd}. Fig. 9 summarizes the unique faults that are sensitized by both approaches and their overlapping faults. The figure clearly shows the difference between the approaches. The unique DAT faults (6 out of 8 of the realistic faults which corresponds to 75%) may lead to test escapes in case tests are used based on the conventional defect model. On top of that, the conventional defect model approach triggers 9 faults which are not realistic, hence leading to a waste of test time. Note that only 2 common faults are observed between both approaches.

Fig. 9: Static fault sensitization overlap.

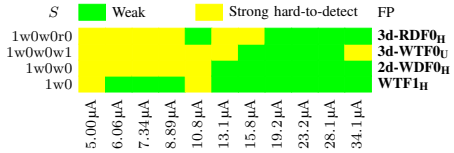


Fig. 10: RRAM forming defect faults.

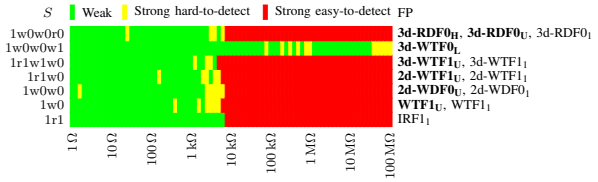
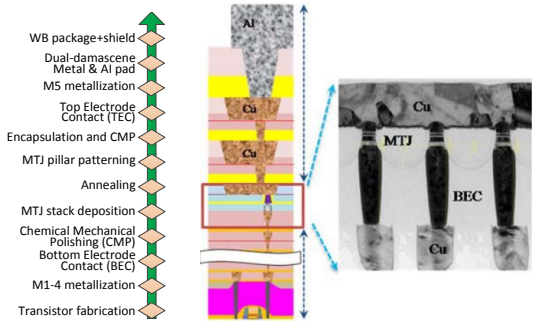


Fig. 11: RRAM series resistor faults.

We continue the fault analysis with two case studies in which we increase the length of S, i.e., we sensitize dynamic faults. Fig. 10 shows the fault class and FP for the strong faults that were observed for varying I_{form} on the same row of their sensitizing operation. The sequences were chosen to illustrate that more strong faults are sensitized with increasing length of S. The longer the sensitizing sequence, the stronger the fault becomes. Note that the faults are still hard-to-detect faults (name bold faced in the figure). This can be explained by the fact that lower I_{form} results in increased RRAM device resistance (both R_{LRS} and R_{HRS}), or even non-forming defects. Due to this increase, the cells are unable to switch in the valid '1' region and instead switch into the 'U' region, while cells that have to switch into the '0' region end up in the 'H' region, as illustrated by the FPs. Note that despite the faults being strong hard-to-detect, they can still be caught easier than weak faults. The figure shows further that the ranges of fault types are interrupted. This is caused by the stochastic behavior of the filament growth and rupture, sometimes bringing the cell in an unpredicted state.

The application of the methodology to traditional resistive defects is shown in Fig. 11 for R_{sd}. Again, strong hard-to-detect faults are marked bold faced while easy-to-detect faults in regular font. Due to the space limitations, we omit showing the results for R_{pd}. The figure also shows that the fault coverage increases when the length of S is increased. For example, for a defect size of $R = 5.01\,\mathrm{k}\Omega$ both strong hard-to-detect faults (for the sequence S=1r1w0) as well as strong easy-to-detect faults (for the sequence S=1r1w1w0) can be observed. The first sequence leads to a 2d-WTF1$_{\mathrm{U}}$ strong hard-to-detect fault, while the second sequence enhances the faulty behavior and causes a strong easy-to-detect 3d-WTF1$_1$ fault. For comparison, Fig. 11 also shows the same sequences that were shown in Fig. 10. A difference can be seen here which is that the resistive defect model is unable to switch to the '0' state with increasing resistance, while the device-aware defect model shows that the device is still switching between the states. This difference is caused by the fact that the series resistor reduces the voltage over the RRAM device, and hence,



(a) Processing flow. (b) Cross-section TEM [36].

Fig. 12: General manufacturing process of STT-MRAM.

the switching threshold is never reached.

From the above, it follows that the DAT approach and the conventional approach result in the sensitization of different faults. The device-aware model is always able to show the switching of the cell, while the resistive defect model only shows switching behavior for a limited defect range, thus incorrectly modeling this defect. Therefore, using an inappropriate defect model would lead to low-quality tests that detect non-existing faults and miss existing ones. Further, it follows that the analysis methodology is able to increase the fault coverage by extending the length of S. This extension may have defects turn the fault behaviour from weak hard-to-detect faults into strong hard-to-detect ones, and strong hard-to-detect faults into strong easy-to-detect ones, thus increasing their detection probability.

D. Test Development

The results from the previous step are used to develop an RRAM test. In the fault modeling step we have observed that faults caused by these defects are related to the RRAM device being in a wrong state (i.e., 'U', 'L' or 'H'), causing hard-to-detect faults. Therefore, a DfT scheme is required that focuses on detecting if a cell is in one of these states. Hamdioui *et al.* in [35] have presented a Short Write Time and Low Write Voltage DfT scheme that can be used to detect if the resistance of a cell is in the 'U' state. Note that modifications to this scheme would allow the detection of cells in the 'L' and 'H' state as well. In contrast, the R_{sd} defect model sensitizes many strong easy-to-detect faults, e.g. $\mathrm{IRF1}_1$ and $\mathrm{WTF1}_1$, that are not realistic. Although they may be easily detected by the $\Updownarrow(\mathrm{w1,r1})$ element in a March test, testing for them would still increase test cost unnecessary. Note that the faults sensitized by the R_{sd} may still be applicable to model resistive open defects.

V. Device-Aware Test for STT-MRAMs

In this section, we first describe STT-MRAM manufacturing defects with a particular emphasis on pinhole defects. Thereafter, we apply the DAT methodology to pinhole defects.

A. Manufacturing Defects

The STT-MRAM manufacturing process mainly consists of the standard CMOS fabrication steps and the integration of MTJ devices into metal layers. Fig. 12a shows the bottom-up manufacturing flow and Fig. 12b the vertical multi-layer structure of STT-MRAM cells [36]. Based on the manufacturing phase, STT-MRAM defects can be classified into front-end-of-line (FEOL) and back-end-of-line (BEOL) defects. As MTJs are integrated into metal layers during BEOL processing, BEOL defects can be further categorized into MTJ fabrication defects and metalization defects. Table VI list all potential defects.

TABLE VI: STT-MRAM defect classification.

FEOL	BEOL	
Transistor fabrication	MTJ fabrication	Metalization
Material impurity	Pinholes in TB	Open vias/contacts
Crystal imperfection	Extreme thickness variation of TB	Irregular shapes
Pinholes in gate oxides	MgO/CoFeB interface roughness	Big bubbles
Shifting of dopants	Atom inter-diffusion	Small particles
	Redepositions on MTJ sidewalls	
	Magnetic layer corrosion	
	Magnetic coupling	

Among these defects, pinhole defects in the MgO tunnel barrier are seen as one of the most important defects that may occur in STT-MRAMs [37, 38]. A pinhole defect forms due to unoptimized deposition processes [37]. This causes the formation of metallic shorts in the MgO tunnel barrier, probably due to diffusion of Boron into the MgO barrier or other metallic impurities [39]. As a result, it leads to a degradation of both RA and TMR parameters. Moreover, measurement data in [38] also suggests that a small pinhole grows in area over time because of Joule heating and an electric field across the pinhole circumference. Therefore, if even small pinhole defects are not detected during manufacturing tests, they might cause an early breakdown in the field.

B. Pinhole Defect Modeling

For the conventional resistor-based defect modeling approach, a pinhole defect is modeled as a series resistor R_{sd} or a parallel resistor R_{pd}, as it is the case for modeling the forming defect in RRAM. Next, we present how pinhole defects are modeled by the DAT approach in the following three steps.

Physical defect analysis and modeling. With comprehensive theoretical studies and experimental characterizations for pinhole defects in various MTJ devices as well as our above analysis, it is clear that RA and TMR are the two key technology parameters that are significantly impacted by a pinhole defect [37, 38]. Thus, we model the effect of pinholes on these two technology parameters as follows [23].

$$RA_{\mathrm{eff\_ph}}(A_{\mathrm{ph}}) = \frac{A}{\frac{A(1-A_{\mathrm{ph}})}{RA_{\mathrm{df}}} + \frac{A \cdot A_{\mathrm{ph}}}{RA_{\mathrm{bd}}}} \quad (6)$$

$$TMR_{\mathrm{eff\_ph}}(A_{\mathrm{ph}}) = TMR_{\mathrm{df}} \cdot \frac{RA_{\mathrm{eff\_ph}}(A_{\mathrm{ph}}) - RA_{\mathrm{bd}}}{RA_{\mathrm{df}} - RA_{\mathrm{bd}}} \quad (7)$$

where $A_{\mathrm{ph}} \in [0,1]$ is the normalized pinhole area with respect to the cross-sectional area A of the MTJ device. RA_{df} and TMR_{df} are the defect-free MTJ's RA and TMR parameters (i.e., when $A_{\mathrm{ph}}=0$), respectively. RA_{bd} is the resultant RA after breakdown.

Electrical modeling of the defective device. We then integrate Equations (6-7) into our defect-free MTJ compact model which has been calibrated by the measurement data of good devices (presented in [38]). In this way, we convert

the defect-free MTJ model into a defective MTJ model which is able to predict the electrical impact of pinhole defects on the MTJ device. Furthermore, the pinhole size is tunable by changing the input argument A_{ph}.

Fitting and model optimization. In this step, we perform electrical characterizations for both good MTJ devices and devices with suspected pinhole defects. By fitting to the measured silicon data, we can further optimize our pinhole-parameterized MTJ compact model. By stressing a device with a suspected pinhole defect and curve fitting method, we obtained RA_{bd}=0.41 $\Omega \cdot \mu\mathrm{m}^2$ for our devices. The fitting and model optimization results are presented in [38]. It is clear that the simulation results with our proposed defective MTJ model match the measured silicon data in terms of resistance and switching voltage.

C. Fault Modeling

In this section, we apply the proposed fault modeling methodology to pinhole defects. Similar to what we did for RRAM previously, we first performed fault analysis with the DAT approach. Thereafter, we used the conventional approach to do fault analysis and compared both approaches.

Table VII shows that sufficiently large pinholes (A_{ph}>0.62%) make the MTJ device fall into the resistance range of '0' state or even 'L' state; the corresponding fault primitives are listed in the table. As the pinhole gets smaller ($A_{\mathrm{ph}}\in$[0.08%,0.61%]), it transforms R_{P} into 'L' state and R_{AP} into 'U' state. Depending on the exact MTJ resistance in the AP state, the readout value can be in three cases: 1) '0', 2) random ('?'), and 3) '1'. In Case 1) where R_{AP} is significantly larger than the resistance of the reference cell (i.e., $A_{\mathrm{ph}}\in$[0.36%,0.61%]), the readout value of the device in AP state is '0'. In Case 2) where R_{AP} is close to the resistance of the reference cell (i.e., $A_{\mathrm{ph}}\in$[0.033%,0.35%]), the readout value can be random. In other words, the read operation is unstable, and therefore both '0' and '1' are possible readout values. In Case 3) where R_{AP} is much smaller than the resistance of the reference cell (i.e., $A_{\mathrm{ph}}\in$[0.08%,0.32%]), the readout is '1'. As the pinhole area becomes smaller between 0.05% to 0.07%, the expected '1' state transforms to a 'U' state, while the expected '0' state remains correct. If the pinhole size is smaller than 0.04%, the device behaves normally, leading to no deterministic faults.

In contrast, we also performed fault modeling based on the injection of the R_{sd} and R_{pd} into a defect-free netlist; the simulation results are also shown in Table VII. Comparing the fault modeling results based on the two defect modeling approaches reveals the following.

- The faulty behavior of the memory due to a pinhole defect *cannot* be caught with the conventional resistor-based defect modeling approach. It is clear in the table that the FPs sensitized by our proposed pinhole defect model are not observed in the simulation results with the conventional approach. This is because the MTJ device is considered as a **black box** for the conventional approach. Thus, only '0' and '1' states are seen in the simulations. However, our simulations and measurement data clearly show that pinhole defects can lead the device to states 'U' and 'L'. This means that relying on the traditional approach for fault modeling and test development may result in low quality test solutions, meaning higher number of escapes.
- The conventional approach results in some fault primitives which are not applicable to STT-MRAMs (i.e., not found with our approach based on a calibrated model for the pinhole defect). For example, using a series resistor R_{sd} results in IRF0_0, while using a parallel resistor R_{pd} results in WTF0_0. This may lead to tests targeting non-existing faults, meaning a waste of test time.

TABLE VII: Single-cell static fault modeling results.

Defect Model	Value	Sensitized Fault Primitive	Detection Condition
Pinhole area A_{ph}	0-0.04%	Fault-free	DfT needed
	0.05-0.07%	$\mathrm{SF1_U}$, $\mathrm{WDF1_U}$, $\mathrm{WTF0_U}$, $\mathrm{RDF1_U}$	
	0.08-0.32%	$\mathrm{SF0_L}$, $\mathrm{SF1_U}$, $\mathrm{WDF0_L}$, $\mathrm{WDF1_U}$, $\mathrm{WTF0_U}$, $\mathrm{WTF1_L}$, $\mathrm{RDF0_L}$, $\mathrm{RDF1_U}$	
	0.33-0.35%	$\mathrm{SF0_L}$, $\mathrm{SF1_U}$, $\mathrm{WDF0_L}$, $\mathrm{WDF1_U}$, $\mathrm{WTF0_U}$, $\mathrm{WTF1_L}$, $\mathrm{RDF0_L}$, $\mathrm{RRDF1_U}$	
	0.36-0.61%	$\mathrm{SF0_L}$, $\mathrm{SF1_U}$, $\mathrm{WDF0_L}$, $\mathrm{WDF1_U}$, $\mathrm{WTF0_U}$, $\mathrm{WTF1_L}$, $\mathrm{RDF0_L}$, **$\mathrm{IRDF1_U}$**	⇳ (r1)
	0.62-0.78%	$\mathrm{SF0_L}$, **$\mathrm{SF1_0}$**, $\mathrm{WDF0_L}$, $\mathrm{WDF1_0}$, $\mathrm{WTF0_0}$, $\mathrm{WTF1_L}$, $\mathrm{RDF0_L}$, $\mathrm{IRDF1_0}$	⇳ (r1)
	>0.79%	$\mathrm{SF0_L}$, **$\mathrm{SF1_L}$**, $\mathrm{WDF0_L}$, $\mathrm{WDF1_L}$, $\mathrm{WTF0_L}$, $\mathrm{WTF1_L}$, $\mathrm{RDF0_L}$, $\mathrm{IRDF1_L}$	⇳ (r1)
Series resistor R_{sd}	0−310 Ω	Fault-free	DfT needed
	310−3.1 kΩ	**$\mathrm{IRF0_0}$**	⇳ (r0)
	3.1 k−∞ Ω	**$\mathrm{IRF0_0}$**, $\mathrm{WTF0_0}$, $\mathrm{WTF1_1}$	⇳ (r0)
Parallel resistor R_{pd}	0−1.1 kΩ	**$\mathrm{IRF1_1}$**, $\mathrm{WTF0_0}$, $\mathrm{WTF1_1}$	⇳ (r1)
	1.1 k−3.1 kΩ	**$\mathrm{IRF1_1}$**, $\mathrm{WTF1_1}$	⇳ (r1)
	3.1 k−∞ Ω	Fault-free	DfT needed

D. Test Development

Based on our simulation results with the calibrated pinhole defect model, it is clear that the larger the pinhole, the larger its fault effect, and hence the easier it is to detect it. As shown in Table VII, a pinhole defect with a specific range of defect sizes can cause multiple faults. However, any test that is able to detect one of these faults can guarantee the detection of this specific pinhole defect. For example, when the pinhole area A_{ph} is larger than 0.79%, there are eight sensitized fault primitives. Among these FPs, $\mathrm{SF1_L}$ (marked with bold font in the table) can simply be detected by a read '1' operation, because they are strong easy-to-detect faults. Thus, ⇳(r1) is the detection condition in a March algorithm for a pinhole with A_{ph}>0.79%. The detection conditions for different pinhole sizes are listed in the last column of Table VII.

Combining the last three rows in Table VII, it is clear that any march tests including the element ⇳(w1,r1) can guarantee the detection of a pinhole defect with A_{ph}>0.36% as an easy-to-detect fault. However, for smaller pinhole defects, March tests cannot guarantee their detection, because the defect causes hard-to-detect faults. As small pinhole defects grow in area over time due to the accumulated Joule heating, they would cause an early breakdown in the field if not detected during manufacturing tests [38]. This calls for DfT designs or stress tests dedicated to detecting a tiny pinhole defect. One possible solution is to subject the STT-MRAM to a hammering write '1' operation sequence with elevated voltage or prolonged pulse width to deliberately speedup

the growth of pinhole defects, thereby causing easy-to-detect faults. However, this approach is prohibitively expensive for high-volume testing. In addition, the amplitude and duration of the hammering write pulse need to be carefully tuned to avoid any inadvertent destruction of good devices while maintaining an acceptable test effectiveness and efficiency.

VI. Discussion and Conclusion

In this paper we have presented the device-aware test approach which consists of three steps: defect modeling, fault modeling, and test development. In contrast to conventional based resistive testing, DAT leads to accurate fault models and thereby enables high-quality (towards DPPB-level) test. Based on the observations, we conclude the following.

Test Escape Reduction and Quality Improvement: As we demonstrated for both RRAM and STT-MRAM previously, our proposed DAT approach results in more accurate fault models which reflect the physical defects. Many faults sensitized using our approach are unique and not observed by the conventional resistor-based defect modeling approach. Hence, our approach clearly reduces the number of test escapes and increases the test quality.

Efficient Yield Learning: Modeling the defects accurately and creating a fault dictionary for them may speed up the yield learning process significantly. As each defect can be modeled separately using device-aware testing, instead of using resistive defect models for all defects, unique fault signatures can be created for each defect. This improves the yield learning curve, as the defects can be more accurately diagnosed based on their fault signatures.

Test Time Optimization: Nowadays, companies are spending a lot of time on functional test (or system test) to compensate for the fault coverage due to the limitations of traditional fault modeling and testing. The DAT approach allows for the development of appropriate and efficient structural tests, which can be applied at manufacturing stage; hence, significantly reducing the expensive test time spend on board testing.

General Applicability: DAT is not limited to emerging memories; the approach can also be applied in the test generation for other circuits, e.g. SRAMs and logic, as well as for other kinds of devices, such as FinFETs and PCM devices. Future work should focus on applying the DAT approach there as well.

References

[1] S. Borkar, "Design challenges of technology scaling," *IEEE Micro*, vol. 19, no. 4, 1999.

[2] N. Loubet *et al.*, "Stacked nanosheet gate-all-around transistor to enable scaling beyond FinFET," in *IEEE VLSI*, 2017.

[3] S. Borkar, "Microarchitecture and Design Challenges for Gigascale Integration," in *37th Int. Symp. Microarchitecture*, 2005.

[4] A. N. Bhoj *et al.*, "Fault models for logic circuits in the multigate era," *IEEE Trans. Nanotech.*, vol. 11, no. 1, Jan. 2012.

[5] K. Mei, "Bridging and stuck-at faults," *IEEE Trans. Comput.*, vol. C-23, no. 7, 1974.

[6] J. Waicukauski *et al.*, "Transition fault simulation," *IEEE Des. Test Comput.*, vol. 4, no. 2, 1987.

[7] H. Cox *et al.*, "Stuck-open and transition fault testing in CMOS complex gates," in *IEEE ITC*, 1998.

[8] F. Ferguson *et al.*, "Test pattern generation for realistic bridge faults in CMOS ICs," in *ITC*, 1991.

[9] J. Rearick *et al.*, "Fast and accurate CMOS bridging fault simulation," in *ITC*, 1993.

[10] P. Dahlgren *et al.*, "A fault model for switch-level simulation of gate-to-drain shorts," in *VLSI Test*, 1996.

[11] I. Pomeranz *et al.*, "On n-detection test sets and variable n-detection test sets for transition faults," *VLSI Test*, 1999.

[12] J. Geuzebroek *et al.*, "Embedded multi-detect ATPG and its effect on the detection of unmodeled defects," in *ITC*, 2007.

[13] S. K. Goel *et al.*, "Circuit topology-based test pattern generation for small-delay defects," in *ATS*, 2010.

[14] F. Hapke *et al.*, "Cell-aware test," *IEEE Trans. Comput. Des. Integr. Circuits Syst.*, vol. 33, no. 9, 2014.

[15] Z. Gao *et al.*, "Defect-location identification for cell-aware test," in *LATS*, 2019.

[16] M. A. Breuer *et al.*, *Diagnosis & reliable design of digital systems*. Computer Science Press, 1976.

[17] A. J. van de. Goor, *Testing semiconductor memories: theory and practice*. Gouda, Netherlands: ComTex Publishing, 1998.

[18] I. Schanstra *et al.*, "Industrial evaluation of stress combinations for march tests applied to SRAMs," in *ITC*, 1999.

[19] A. J. van de Goor *et al.*, "Industrial evaluation of DRAM tests," in *DATE*, 1999.

[20] S. Hamdioui *et al.*, "An experimental analysis of spot defects in SRAMs: realistic fault models and tests," in *ATS*, 2000.

[21] E. I. Vatajelu *et al.*, "Analyzing resistive-open defects in SRAM core-cell under the effect of process variability," in *ETS*, 2013.

[22] M. Fieback *et al.*, "Testing resistive memories: Where are we and what is missing?" In *ITC*, 2018.

[23] L. Wu *et al.*, "Electrical modeling of STT-MRAM defects," in *ITC*, 2018.

[24] H.-S. P. Wong *et al.*, "Metal-oxide RRAM," *Proceedings of the IEEE*, vol. 100, no. 6, 2012.

[25] S. Kannan *et al.*, "Sneak-path testing of memristor-based memories," in *Int. Conf. VLSI Design*, 2013.

[26] A. V. Khvalkovskiy *et al.*, "Erratum: Basic principles of STT-MRAM cell operation in memory arrays," *J. Phys. D: Appl. Phys*, vol. 46, no. 13, 2013.

[27] A. van de Goor *et al.*, "Functional memory faults: a formal notation and a taxonomy," in *VLSI Test*, 2000.

[28] N. Z. Haron *et al.*, "DfT schemes for resistive open defects in RRAMs," in *DATE*, 2012.

[29] C. Y. Chen *et al.*, "RRAM defect modeling and failure analysis based on march test and a novel squeeze-search scheme," *Transactions on Computers*, vol. 64, no. 1, 2015.

[30] A. Grossi *et al.*, "Fundamental variability limits of filament-based RRAM," in *IEDM*, 2016.

[31] N. Raghavan, "Performance and reliability trade-offs for high-κ RRAM," *Microelectronics Reliability*, vol. 54, no. 9-10, 2014.

[32] A. Kalantarian *et al.*, "Controlling uniformity of RRAM characteristics through the forming process," in *IRPS*, 2012.

[33] B. Govoreanu *et al.*, "10x10nm2 Hf/HfOx crossbar resistive RAM with excellent performance, reliability and low-energy operation," in *IEDM*, 2011.

[34] H. Li *et al.*, "A SPICE model of resistive random access memory for large-scale memory array simulation," *Electron Device Letters*, vol. 35, no. 2, Feb. 2014.

[35] S. Hamdioui *et al.*, "Testing open defects in memristor-based memories," *Transactions on Computers*, vol. 64, no. 1, Jan. 2015.

[36] Y. J. Song *et al.*, "Highly functional and reliable 8Mb STT-MRAM embedded in 28nm logic," in *IEDM*, 2016.

[37] W. Zhao *et al.*, "Failure analysis in magnetic tunnel junction nanopillar with interfacial perpendicular magnetic anisotropy," *Materials*, vol. 9, no. 1, 2016.

[38] L. Wu *et al.*, "Pinhole defect characterization and fault modeling for STT-MRAM testing," in *ETS*, May 2019.

[39] S. Mukherjee *et al.*, "Role of boron diffusion in CoFeB/MgO magnetic tunnel junctions," *Physical Review B*, vol. 91, no. 8, 2015.