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

Fieback, M.; Wu, Lizhou; Cardoso Medeiros, Guilherme; Aziza, Hassen; Rao, S; Marinissen, Erik Jan; Taouil, Mottaqiallah; Hamdioui, Said

DOI
10.1109/ITC44170.2019.9000134

Publication date
2019

Document Version
Accepted author manuscript

Published in
2019 IEEE International Test Conference, ITC 2019

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

Moritz Fieback[1] Lizhou Wu[1] Guilherme Cardoso Medeiros[1] Hassen Aziza[2]
Siddharth Rao[3] Erik Jan Marinissen[3] Mottaqiallah Taouil[1,4] Said Hamdioui[1,4]
[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
[4]CognitiveIC, Van der Burghweg 1, 2628CS, Delft, The Netherlands
Email: S.Hamdioui@tudelft.nl

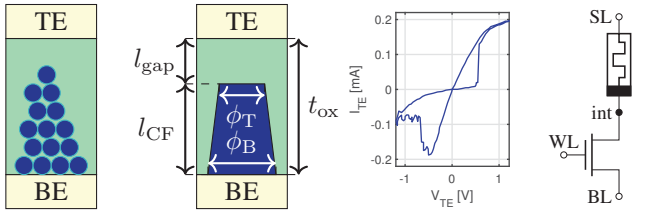
***Abstract*—This paper proposes a new test approach that goes beyond cell-aware test, i.e., device-aware test. The approach consists of three steps: defect modeling, fault modeling, and test/DfT development. The defect modeling does not assume that a defect in a device (or a cell) can be modeled electrically as a linear resistor (as the traditional approach suggests), but it rather incorporates 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 subsequently 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 a defective part per billion (DPPB) level. The new approach clearly sets up a turning point in testing for at least the considered two emerging memory technologies.**

I. Introduction

Technology scaling has driven the phenomenal success of the semiconductor industry in delivering more complex, faster, and cheaper integrated circuits with a high quality of service [1]. Silicon technology has entered the nano-era and transistors with sizes below $5\,\mathrm{nm}$ are being prototyped [2, 3]. 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., in the range of *defective parts per billion* (DPPB) [4]. 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 [5]. This shift in failure mechanisms may impact the way *fault modeling* has to be done in the future. Note that accurate fault models which reflect the realistic 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 *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 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 [6], transition [7, 8], and bridge faults [9, 10]. 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 [11], N-detect [12], embedded multi-detect [13], and layout-aware fault modeling [14]. Moreover, the increasing demand of customers for higher quality has further led to the introduction of cell-aware test [15, 16]; 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 interconnects and terminals of each device within the library cell.

Memory testing went through a quite similar evolution. The early memory tests (before 1980) can be classified as ad-hoc tests due to the absence of formal fault models and proofs [17]; they have a low defect and fault coverage and a very long test time, typically in the order of $\mathcal{O}\left(n^2\right)$ with n the amount of addresses, which made them impractical for larger 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 $\mathcal{O}\left(n\right)$; i.e., linear in the size of the memory. Some important fault models introduced in that time were stuck-at faults and address-decoder faults [18]. These are abstract fault models not based on any actual memory design nor 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 [19, 20], 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 SPICE simulation [21, 22]: read destructive faults, write disturb faults, transition



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

Fig. 1: RRAM device technology.

coupling faults, read destructive coupling faults, etc. Note that the current cell-aware test fault modeling approaches are quite similar to this as they also model defects as linear resistors (opens and shorts) at the terminals and interconnects 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 by linear resistors [23]. It has recently been demonstrated that this assumption is inaccurate for emerging technologies such as resistive random access memory (RRAM) [24] and spin-transfer torque magnetic random access memory (STT-MRAM) [25]; the results showed that the traditional approach may even lead to wrong fault models. Hence, it is incapable of delivering 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 parameter 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 popular emerging memory technologies, namely oxide-based RRAM and STT-MRAM. The main contributions of the paper are as follows.

- Introduction of 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 fault space. This step provides insight not only on the nature of realistic faults, but also on 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.
- Demonstration of DAT on RRAM and STT-MRAM: we apply and demonstrate the superiority of this approach by comparing it to conventional memory test approaches. DAT can model and detect some device defects that cannot be detected by conventional approaches. Hence, it can reduce the amount of test escapes and can better diagnose defects for fast yield learning.

TABLE I: RRAM key parameters.

Technology Parameters		Electrical Parameters	
t_{ox}	Oxide thickness	V_{set}	Set threshold
l_{CF}	CF length	V_{reset}	Reset threshold
l_{gap}	Gap length	R_{LRS}	Set resistance
ϕ_T	CF top width	R_{HRS}	Reset 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 a brief background 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" (a represenatative defect in an RRAM device) and applies the three DAT steps; not only in order to show how the approach works, but also to validate its superiority over conventional approaches. Section V does the same for the "pinhole defect" in STT-MRAMs. Section VI discusses the advantages and limitations of the method and concludes the paper.

II. Technology Background

This section describes the working principles of two memristive technologies: RRAM and STT-MRAM.

A. RRAM Fundamentals

RRAM is an emerging non-volatile memory technology that uses oxide-based (OxRAM) or conductive bridges (CBRAM) memristors to store data [26]. In this work, we will analyze OxRAM devices. The production of the RRAM devices can be integrated in the back-end-of-line (BEOL) of a standard CMOS process [26].

The RRAM device is schematically shown in Fig. 1a. It consists of a top (TE) and a 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 metal and oxygen ions are broken and the oxygen ions are attracted to the TE, leaving behind a chain of oxygen vacancies, referred to as 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 that is lower than the reset threshold (V_{reset}) is applied, then 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 key technology parameters that determine the resistance of the RRAM device. Its resistance has an analog nature, i.e., it can take any value within a certain range. Fig. 1c shows the switching behavior of the device, as

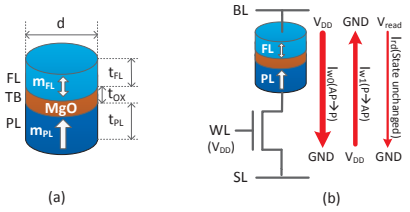


Fig. 2: (a) Simplified MTJ device organization, (b) 1T-1MTJ cell.

well as the 'set' and 'reset' state switching thresholds. From the graph, 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 designs. 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 device. 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 couple multiple cells [27].

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, as shown in Table II. 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 [25]. 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 [28], the MTJ's resistance is low when the magnetization in the FL is *parallel* to that in the PL and 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.

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 1T-1R RRAM, as illustrated in the figure. For STT-MRAMs, R_P, R_{AP}, I_c(AP→P), I_c(P→AP), t_w(AP→P), and t_w(P→AP) are six key electrical parameters determining the electrical behavior of MTJ devices [25], as listed in Table II. Note that P→AP indicates a transition from P state to AP state and AP→P represents the opposite transition.

TABLE II: STT-MRAM key parameters.

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

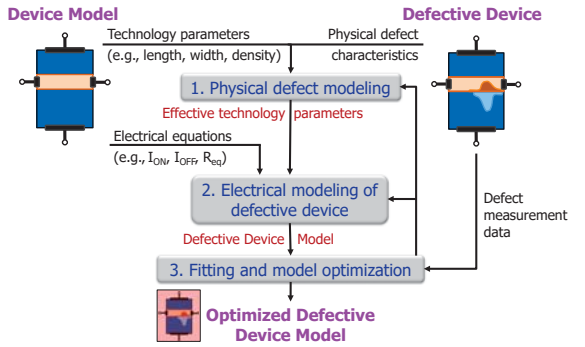


Fig. 3: Generic device defect modeling flow.

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 [24] and STT-MRAM [25], 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 requirements. First, the device defects are physically modeled and their electrical behavior is incorporated into device models. Second, the model is integrated in a memory simulation platform to analyze the impact of the defect on memory behavior; this is done in a systematic manner by validating a pre-defined *complete* fault space using SPICE simulation. The results of this step provide insights on the nature of realistic faults, which are used in order to develop optimal and appropriate test solutions (e.g., March tests, DfT). Next, these three steps are described in detail. These steps will be applied in Section IV and 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 [25]; this is exactly what device defect modeling of DAT 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 a defective device. Note that a device can be a planar or FinFET transistor, an MTJ device, an

RRAM device, a PCM device, etc. The approach consists of the following three steps.

1) Physical defect analysis and modeling. Given a set of physical defects $\mathbf{D} = \{\mathrm{d}_1, \mathrm{d}_2, ..., \mathrm{d}_\mathrm{n}\}$ 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 technology parameters will be modified from their defect-free values (Tp_df), resulting in what we refer to as an *effective technology parameter* Tp_eff. This can be described by the following abstract function:

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

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. This is done by modifying the defect-free device electrical model and converting it into a defect-parameterized model by integrating Equation 1 for each involved technology parameter. This step gives an uncalibrated defective-device model with the effective electrical output parameters.

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

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 fault space are *realistic* for the defect under consideration; i.e., which faults can only 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 memory faults [29]. 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. These faults can be systematically described using the fault

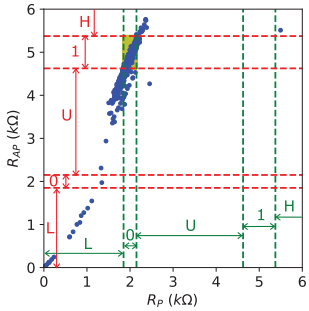


Fig. 4: Measured resistance distribution of R_P and R_AP for $\varnothing 60\,\mathrm{nm}$ MTJ devices, suggesting the existence of states 'L', '0', 'U', '1', and 'H'.

primitive (FP) notation [29]. An FP describes the difference between the observed and expected memory behavior, denoted as a three-tuple $\langle S/F/R \rangle$ where:

- S (sensitization) denotes the operation sequence that sensitizes the fault. A sequence takes the form of $S{=}x_0O_1x_1 \ldots O_ix_i \ldots O_nx_n$, where $x_i{\in}\{0,1\}$ and $O_i{\in}\{\mathrm{r},\mathrm{w}\}$; 0 and 1 denote logical cell values, r and w denote a read and a write operation. If $n \leq 1$, the fault is static, else dynamic.
- F (faulty cell) describes the value that is stored in the cell after S is performed. For traditional charge-based memories, e.g., SRAM, there exist only three states, i.e., $F{\in}\{0,1,\mathrm{U}\}$, where 'U' denotes the undefined state [29]. 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 to define other (faulty) resistance states. Fig. 4 presents the measured resistance distribution of a large number of $\varnothing 60\,\mathrm{nm}$ MTJ devices that were fabricated at IMEC; 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's R_P and R_AP. 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 (R_P='0', R_AP='1') represent good devices in accordance with the above design specifications. However, there is also a large number of devices outside the specification due to 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 of RRAM devices suggest the existence of the five states as well [27, 30].
- R (read output) describes the output of a read operation if the last operation in 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 refer-

TABLE III: Complete single-cell static fault primitives.

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

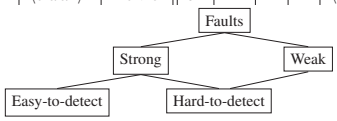


Fig. 5: Fault classification.

ence current), and '$-$' denotes that R is not applicable, i.e., when the last operation in S is a write operation.

The following two examples illustrate the usage of this notation: ⟨0w1/0/-⟩ denotes a w1 operation to a cell that contains a '0' (S=0w1). The write operation to the cells fails, and the cell remains in '0' (F=0). Note that there is no read output (R=$-$). ⟨0r0/H/0⟩ denotes a r0 operation on a cell that holds a '0' (S=0r0). This operation flips the faulty cell's state to 'H' (F=H) and the read output '0' (R=0) is observed.

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

$$\mathrm{FP} = \begin{cases} \mathrm{S}\{ini\}F\{fin\}, & n = 0 \\ [out]\,\{opn\}\{opd\}\{eff\}\mathrm{F}\{fin\}, & n = 1 \\ \{n\mathrm{d}{-}\}\,[out]\,\{opn\}\{opd\}\{eff\}\mathrm{F}\{fin\}, & n > 1 \end{cases}$$

In this scheme, attributes between curly brackets ({ }) are required elements, while elements in regular brackets ([ ]) are only used when a read operation is performed. State faults (n=0) have two attributes, ini, which describes the initial state, and fin which describes the final state, fin is equal to F in the SFR-notation. For $n{\geq}1$, the elements are as follows. opn denotes the operation in S ($opn{\in}\{\mathrm{R},\mathrm{W}\}$). If opn=R, then out is used to define the outcome of the operation. That is, $out{\in}\{\mathrm{i},\mathrm{r},\mathrm{d}\}$, where i denotes an incorrect output, r denotes a random output, and d a deceptive output where a correct output is generated while changing fin. opd is the operand of the operation that is performed, i.e., $opd{\in}\{0,1\}$. eff denotes the effect of the operation, which can be destructive (D), non-destructive (N), or transition (T). Dynamic faults ($n{>}1$) get an additional prefix $n\mathrm{d}{-}$, while the rest of the name is based on the final operation in S. To illustrate this, ⟨0/L/-⟩ is S0FL, ⟨0r0/1/0⟩ is dR0DF1, and ⟨1w0r0/H/?⟩ is 2d-rR0DFH.

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). Note that if the parametric fault is within the process variation specifications, then the deviation is not considered as a weak fault. 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., the use of a special DfT circuitry. 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 rR1NF1 and rR0NF0. 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 read behavior.

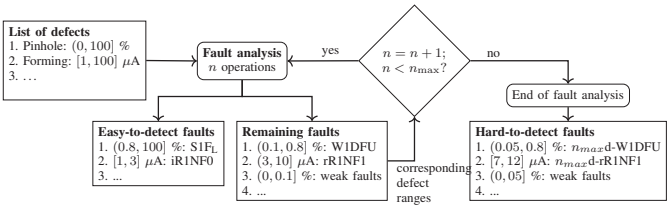


Fig. 6: Fault analysis methodology.

2) Fault Analysis Methodology: Once the defect is modeled and the framework of faults is defined, the verification of the faults can be performed using a systematic simulation-based approach. In this paper we restrict ourselves to single-cell fault analysis because our case studies for RRAM and STT-MRAM involve single-cell defects. Our fault analysis consists of seven steps: 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 until all defect sizes are covered. Note that for the DAT approach, 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. Fig. 6 shows how the fault analysis is applied to a defect. Given a list of defects and ranges of their sizes, the seven steps of the fault analysis are 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 their defect range. In case no easy-to-detect fault is sensitized for the considered defect, the fault is added to a set of remaining faults, i.e., hard-to-detect faults consisting of some FPs or weak faults. Next, all defect ranges that are in the remaining fault set will be further analyzed, but then

using dynamic fault analysis, starting at n=2. Some defects can now trigger easy-to-detect dynamic faults; e.g., S=0w0 causes a weak fault, while S=0w0w0 causes an easy-to-detect fault for the same defect. Once the single-cell dynamic fault analysis for n=2 is completed, we can redo similar analysis for n=3 for defects that are still in the remaining set. The process can be repeated by extending S each time with one operation until the considered $n_{\max}$ is reached. The remaining faults are considered hard-to-detect faults by our analysis. Each step in the process aims to reduce the remaining fault set and increase 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. Overall, the final results are a set of faults that can be easily detected, for example, by the generation of March tests, and another set of faults 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 yet efficient test solutions. All easy-to-detect faults can be detected by applying appropriate test algorithms. To minimize the test cost, the minimal detection conditions for each of the faults are first identified, and thereafter compiled in test algorithms. To further optimize the test time, one can also incorporate DfT; e.g., DfT that enables the test of many faults simultaneously, parallel testing, etc. [27, 31, 32].

Hard-to-detect faults, however, require special attention. Special DfT schemes and tests are required. Examples are: DfT schemes that may directly measure the bit line swing [33], modify the operation conditions such as weak write operations [32], stress tests [34], etc. The aim is to *maximize* the fault coverage for these faults while keeping the test cost affordable.

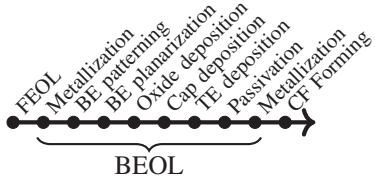
IV. Device-Aware Test for RRAM

In this section we apply the DAT approach on RRAM. However, first we describe RRAM manufacturing defects and select a representative defect.

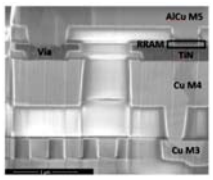
A. RRAM manufacturing defects

The fabrication process flow of an RRAM is depicted in Fig. 7a [24] and their associated defects are listed in Table IV; a more detailed overview can be found in [24]. 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 [35]. 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) Processing flow [24, 35, 38].



(b) Cross-section TEM [35].

Fig. 7: General manufacturing process of RRAM.

TABLE IV: RRAM defect classification [24].

FEOL		BEOL	
Transistor		Interconnection	RRAM Device
Patterning proximity	Shifting of dopants	Opens	Electrode roughness
Line roughness	Random dopants	Shorts	Polish variations
Polish variations	Material impurity	Line roughness	Varying defect density
Anneal	Pinholes in gate oxides	Irregular shapes	Dimensional variations
Strain	Gate granularity	Big bubbles	Material redeposition
Crystal imperfection	Dielectric variations	Small particles	*Overforming*
			Non-forming

a strong impact on the performance and reliability of the device. Few observations on the forming conditions can be made: higher forming currents (I_{form}) result in lower device resistance with less variation [35, 36], and variations in the forming current lead to more resistive variations [37]. Variations in the device geometry and oxide defect density also affect the forming step [38]. A forming defect can result from the forming step; it comes in two variants: overforming, when the CF is too large, and non-forming, when no or only a tiny CF is formed.

B. Forming Defect Modeling

In this section, we model the forming defect using both the DAT and the conventional resistor-based approach.

1) DAT Approach: For the DAT approach, we relate the input parameters of the RRAM device model (such as in [39]) 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 technology parameters shown in Fig. 1b. Note that l_{CF} and ϕ_{T} have the strongest impact on the resistance of the device [36]. 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 defect in the device. The physical defect modeling step can be denoted mathematically as:

$$l_{\mathrm{CF,eff}}\left(I_{\mathrm{form}}\right) = a_1 \exp\left(b_1 \cdot R_\mu\left(I_{\mathrm{form}}\right)\right) + c_1 \exp\left(d_1 \cdot R_\mu\left(I_{\mathrm{form}}\right)\right) \tag{2}$$

$$\phi_{\mathrm{T,eff}}\left(I_{\mathrm{form}}\right) = a_2 \exp\left(b_2 \cdot R_\mu\left(I_{\mathrm{form}}\right)\right) + c_2 \exp\left(d_2 \cdot R_\mu\left(I_{\mathrm{form}}\right)\right) \tag{3}$$

$$\Delta l_{\mathrm{CF,eff}}\left(I_{\mathrm{form}}\right) = a_3 \exp\left(b_3 \cdot R_\sigma\left(R_\mu\right)\right) + c_3 \exp\left(d_3 \cdot R_\sigma\left(R_\mu\right)\right). \tag{4}$$

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

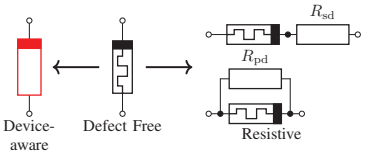


Fig. 8: Device-aware and resistive defective-device models.

TABLE V: FPs for DAT and conventional model.

FP	DAT	Conv.	FP	DAT	Conv.	FP	DAT	Conv.
S1FU	yes	no	W1TFH	yes	yes	W1TF1	no	yes
S1FH	yes	no	W0DFH	yes	yes	W1TF0	no	yes
W1TFH	yes	no				W0TFU	no	yes
W1DFU	yes	no				W1TFU	no	yes
dR1DFU	yes	no				W1TFL	no	yes
dR1DFH	yes	no				W1DFL	no	yes
						iR1NF1	no	yes
						iR0NF0	no	yes
						dR1DFL	no	yes

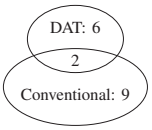


Fig. 9: Static faults.

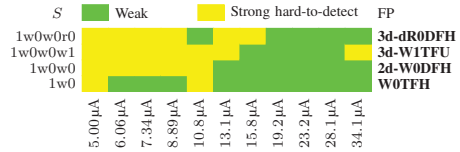


Fig. 10: Forming defect faults based on DAT approach.

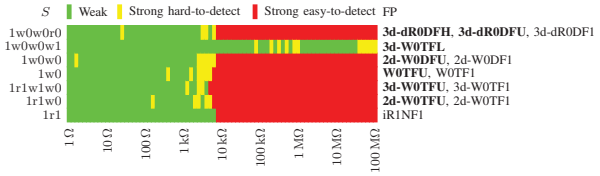


Fig. 11: Forming defect faults based on series resistor model.

interpolation of I_{form} to the median resistance in [35], and R_σ (R_μ) is given by Equation (1) in Ref. [35].

Electrical modeling of the defective device. The RRAM device model in [39] 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, as shown in Table I, can be analyzed.

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 [35]. 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 [35] and thus linked to I_{form}. The effect of Δl_{CF} is similarly analyzed and fitted. We vary I_{form} between $5\,\mu\text{A}$ and $34.1\,\mu\text{A}$ to obtain a wide range of device resistances [35].

2) Conventional Approach: The conventional resistive defect modeling approach models the forming defect as a linear resistor that is either in 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 its resistance value; both R_{pd} and R_{sd} are swept from from $1\,\Omega$ to $100\,\text{M}\Omega$.

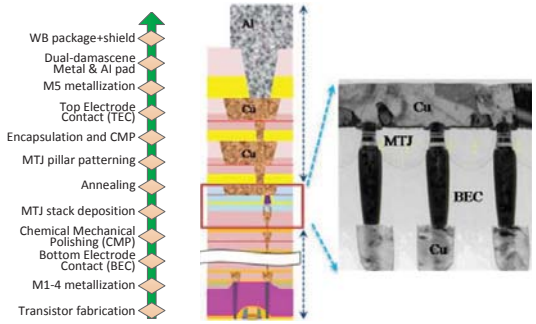
C. Fault Modeling

This step consists of fault analysis based on the use of the electrical models. As a forming defect impacts a single RRAM device (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 perform the fault analysis by injecting defects in a netlist and simulating them using Cadence 's analog simulator Spectre. The netlist contains a 2×2 1T-1R cell array, drivers for the bit and select lines, sense amplifiers, and address decoders. We use the $130\,\text{nm}$ PTM transistor library [40] and the RRAM device model in [39].

We start the fault analysis by first analyzing static faults. Table V lists the static faults (identified in Table III) that were sensitized 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 two 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 when modeling forming defects, hence leading to a waste of test time. Note that only 2 common faults are observed by both approaches.

We continue the fault analysis for n=2 and thereafter for n=3 as shown in Fig. 6, i.e., dynamic fault analysis. Fig. 10 shows the FPs and their Ss for the strong faults that were observed for varying I_{form}. The displayed sequences were chosen to illustrate that more *strong faults* are sensitized with increasing length of S. The longer the sensitizing sequence, the more strong faults and less weak faults are sensitized. Note that the faults are still hard-to-detect faults (name boldfaced in the figure). This can be explained by the fact that a 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 to 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 provide insights on how they should be detected. 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 fault analysis methodology from Fig. 6 to traditional resistive defects is shown in Fig. 11 for R_{sd}. Again, strong hard-to-detect faults are marked boldfaced while easy-to-detect faults are in regular font. Due to space limitations, we omit showing the results for R_{pd}. A clear difference in the sensitized faults by the two models can be seen: the resistive-defect model is unable to switch to the '0' state with increasing resistance (e.g., FP 2d-W0TF1), while the device-aware defect model shows that the device is still switching between the states (e.g., FP W0TFH). Looking at the bottom three sequences in Fig. 11, it can be seen that the fault



(a) Processing flow. (b) Cross-section TEM [42].

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

coverage increases with increasing S. For example, for some defect size where $1\,\mathrm{k}\Omega{<}R{<}10\,\mathrm{k}\Omega$ 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-W0TFU strong hard-to-detect fault, while the second sequence enhances the faulty behavior and causes a strong easy-to-detect 3d-W0TF1 fault.

From the above it follows that the DAT approach and the conventional approach result in the sensitization of different faults. The DAT approach sensitizes many unique faults that are not sensitized by the conventional approach. Therefore, a test based on the conventional approach will result in a low-quality test and therefore in test escapes. Besides that, it follows that the analysis methodology is able to increase the fault coverage by extending the length of S.

D. Test Development

The results from the previous step are used to develop a test solution. In the fault modeling step we have observed that faults caused by the targeted defect (i.e., the forming defect) are related to the memory cell entering a wrong state (i.e., 'U', 'L' or 'H'), causing hard-to-detect faults. Therefore, a DfT scheme is more suited to detect such faults than, for example, a March test. Hamdioui *et al.* in [41] have presented a Short Write Time and Low Write Voltage DfT scheme that can be used to detect faults that cause the cell to be in the 'U' state. Modifications to this scheme 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. iR1NF1, that are not realistic for the forming defect. Although they may be easily detected by the $\Updownarrow(\mathrm{w1,r1})$ element in a March test, testing for them would 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 [42]. Based on the manufacturing phase, STT-MRAM defects can be classified into FEOL and BEOL defects. As MTJs are integrated into metal layers during BEOL processing, BEOL defects can be further categorized into MTJ fabrication defects and interconnection defects. Table VI lists all potential defects.

TABLE VI: STT-MRAM defect classification.

FEOL	BEOL		
Transistor	Interconnection	STT-MRAM Device	
See Table IV	See Table IV	Pinholes in TB	Redepositions on MTJ sidewalls
		Extreme thickness variation of TB	Magnetic layer corrosion
		MgO/CoFeB interface roughness	Magnetic coupling
		Atom inter-diffusion	

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 [34, 43]. A pinhole defect forms due to unoptimized deposition processes [43]. 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 [44]. As a result, it leads to a degradation of both RA and TMR parameters. Moreover, measurement data in [34] 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 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 is the case for the forming defect model in RRAM. Next, we present how pinhole defects are modeled by the DAT approach in the following three steps.

Physical defect analysis and modeling. RA and TMR are the two key technology parameters that are significantly impacted by the presence of a pinhole defect [34, 43]. Thus, we model the effect of a pinhole on these two technology parameters as follows [25].

$$RA_{\mathrm{eff\_ph}}(A_{\mathrm{ph}}) = \frac{A}{\frac{A(1-A_{\mathrm{ph}})}{RA_{\mathrm{df}}} + \frac{A \cdot A_{\mathrm{ph}}}{RA_{\mathrm{bd}}}} \tag{5}$$

$$TMR_{\mathrm{eff\_ph}}(A_{\mathrm{ph}}) = TMR_{\mathrm{df}} \cdot \frac{RA_{\mathrm{eff\_ph}}(A_{\mathrm{ph}}) - RA_{\mathrm{bd}}}{RA_{\mathrm{df}} - RA_{\mathrm{bd}}} \tag{6}$$

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. Next, We integrate Equations (5-6) into our defect-free MTJ compact model which has been calibrated with measurement data of good devices (presented in [34]). In this way, we convert the defect-free MTJ model into a defective-MTJ model which is able to predict the electrical impact of a pinhole defect on the MTJ device. Furthermore, the pinhole size is tunable by changing the input argument A_{ph}.

TABLE VII: Single-cell static fault modeling results for pinhole defects.

	Defect Model	Value	Sensitized Fault Primitive	Detection Condition
DAT	Pinhole area A_{ph}	(0.04, 0.07]%	S1FU, W1DFU, W1TFU, dR1DFU	DfT needed
		(0.07, 0.32]%	S0FL, S1FU, W0DFL, W1DFU, W1TFU, W0TFL, dR0DFL, dR1DFU	
		(0.32, 0.35]%	S0FL, S1FU, W0DFL, W1DFU, W1TFU, W0TFL, dR0DFL, rR1DFU	
		(0.35, 0.61]%	S0FL, S1FU, W0DFL, W1DFU, W1TFU, W0TFL, dR0DFL, **iR1DFU**	⇳ (r1)
		(0.61, 0.78]%	S0FL, **S1F0**, W0DFL, W1DF0, W1TF0, W0TFL, dR0DFL, iR1DF0	⇳ (r1)
		>0.78%	S0FL, **S1FL**, W0DFL, W1DFL, W1TFL, W0TFL, dR0DFL, iR1DFL	⇳ (r1)
Conventional	Series resistor R_{sd}	(310, 3.1k] Ω	**iR0NF0**	⇳ (r0)
		>3.1 kΩ	**iR0NF0**, W1TF0, W0TF1	⇳ (r0)
	Parallel resistor R_{pd}	[0 k, 1.1 k) Ω	**iR1NF1**, W1TF0, W0TF1	⇳ (r1)
		(1.1 k, 3.1 k] Ω	**iR1NF1**, W0TF1	⇳ (r1)

Fitting and model optimization. In this step, we perform electrical characterizations for both good MTJ devices and devices for which we suspect that they contain 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 [34]. It is clear that the simulation results of our proposed defective MTJ model match the measured silicon data in terms of resistance and switching voltage.

C. Fault Modeling

We applied 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. We use MTJ model from [34] and 45 nm PTM transistor models [40]; the peripheral circuits are similar to RRAM.

Table VII shows the results of static fault analysis; it reveals that sufficiently large pinholes (A_{ph}>0.61%) make the MTJ device fall into the resistance range of the '0' state or even the 'L' state; the corresponding fault primitives are listed in the table. As the pinhole gets smaller ($A_{\mathrm{ph}}\in$(0.07%,0.61%]), R_{P} falls into the 'L' state and R_{AP} into the 'U' state. Depending on the exact MTJ resistance in the AP state, the readout value can be one of the following three cases: (1) '0', (2) random ('?'), and (3) '1'. In Case (1), R_{AP} is significantly smaller than the resistance of the reference cell (i.e., $A_{\mathrm{ph}}\in$(0.35%,0.61%]), the readout value of the device in the AP state is '0'. In Case (2), R_{AP} is close to the resistance of the reference cell (i.e., $A_{\mathrm{ph}}\in$(0.32%,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), R_{AP} is much larger than the resistance of the reference cell (i.e., $A_{\mathrm{ph}}\in$(0.07%,0.32%]), the readout is '1'. As the pinhole area becomes even smaller between 0.04% to 0.07%, R_{AP} falls into the 'U' state, while R_{P} remains in the correct range. If the pinhole size is smaller than 0.04%, it leads to a weak fault, while the device still behaves logically correct.

To enable comparison, we also performed fault modeling based on the injection of R_{sd} and R_{pd} resistors into a defect-free netlist; the simulation results are also shown in Table VII. By comparing the derived FPs based on the two defect modeling approaches, we found that there are 17 unique FPs that can only be sensitized with the DAT approach, as shown in Fig. 13. 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'. In contrast, the conventional approach results in 3 FPs which are not applicable to STT-MRAMs (i.e., not found with our approach based on a calibrated model for the pinhole defect). This may lead to tests targeting non-existing faults, meaning a waste of test time and resources. It is worth noting that there is only 1 FP (i.e., W1TF0) that is sensitized by both approaches.

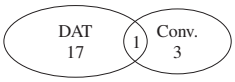


Fig. 13: Sensitized FPs by DAT and conventional (conv.) approaches.

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 0.78%, there are eight sensitized fault primitives. Among these FPs, S1FL (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}>0.78%. The detection conditions for different pinhole sizes are listed in the last column of Table VII.

The fault modeling results based on DAT shown in Table VII clearly suggest that any march tests including the element ⇳(w1,r1) can guarantee the detection of a pinhole defect with A_{ph}>0.35% as an easy-to-detect fault. However, for a smaller pinhole defect, March tests cannot guarantee their detection, because the defect causes hard-to-detect faults. As a small pinhole defect 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 [34]. 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 speed up 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 resistive-based defect modeling, DAT leads to accurate fault models and thereby enables high-quality (towards DPPB-level) test. The DAT approach enables the following.

Test Escape Reduction and Quality Improvement: As we demonstrated for both RRAM and STT-MRAM, the 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, we expect our approach to increase the test quality and reduce the number of test escapes.

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 spent on board testing.

General Applicability: Although it is demonstrated for RRAM and STT-MRAM, the DAT approach can also be applied to any kind of memories including advanced volatile technologies (e.g., SRAM, DRAM) as well as non-volatile ones (e.g., Flash, PCM). Moreover, it can be also applied to logic circuits especially for technology nodes below $10\,\mathrm{nm}$, where it has been shown that many failure mechanisms cannot be modeled with linear resistors [23].

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. M. Y. Sherazi *et al.*, "Standard-cell design architecture options below 5nm node: The ultimate scaling of FinFET and Nanosheet," in *SPIE*, 2019.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

[23] M. Shah *et al.*, "Special Session: A Quality and Reliability Driven DFT and DFR Strategy for Automotive and Industrial Markets," in *2019 IEEE 37th VLSI Test Symposium (VTS)*, Apr. 2019.

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

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

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

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

[28] 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.

[29] S. Hamdioui, *Testing Static Random Access Memories*. Springer, Boston, MA, 2004.

[30] 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.

[31] S. Hamdioui *et al.*, "Test and Reliability of Emerging Non-volatile Memories," in *ATS*, 2017.

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

[33] G. Cardoso Medeiros *et al.*, "DFT Scheme for Hard-to-Detect Faults in FinFET SRAMs," in *ETS*, 2019.

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

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

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

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

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

[39] 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.

[40] Arizona State University. (2012). Predictive Technology Model (PTM), [Online]. Available: http://ptm.asu.edu/.

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

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

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

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