Accelerating RRAM Testing with a Low-cost Computation-in-Memory based DFT

Singh, Abhairaj; Fieback, Moritz; Bishnoi, Rajendra; Bradarić, Filip ; Gebregiorgis, Anteneh; Joshi, Rajiv V.; Hamdioui, Said

DOI
10.1109/ITC50671.2022.00085

Publication date
2022

Document Version
Final published version

Published in
Proceedings - 2022 IEEE International Test Conference, ITC 2022

Citation (APA)
Singh, A., Fieback, M., Bishnoi, R., Bradarić, F., Gebregiorgis, A., Joshi, R. V., & Hamdioui, S. (2022). Accelerating RRAM Testing with a Low-cost Computation-in-Memory based DFT. In C. Ceballos (Ed.), *Proceedings - 2022 IEEE International Test Conference, ITC 2022* (pp. 400-409). (Proceedings - International Test Conference; Vol. 2022-September). IEEE. https://doi.org/10.1109/ITC50671.2022.00085

Accelerating RRAM Testing with a Low-cost Computation-in-Memory based DFT

Abhairaj Singh[1], Moritz Fieback[1], Rajendra Bishnoi[1], Filip Bradarić[1], Anteneh Gebregiorgis[1],
Rajiv V. Joshi[2], Said Hamdioui[1]
[1]Computer Engineering Laboratory, TU Delft, The Netherlands: A.Singh-5@tudelft.nl
[2]IBM Thomas J. Watson Research Centre, USA

2022 IEEE International Test Conference (ITC) | 978-1-6654-6270-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/ITC50671.2022.00085

***Abstract*—Emerging non-volatile resistive RAM (RRAM) device technology has shown great potential to cultivate not only high-density memory storage, but also energy-efficient computing units. However, the unique challenges related to RRAM fabrication process render the traditional memory testing solutions inefficient and inadequate for high product quality. This paper presents low-cost design-for-testability (DFT) solutions that augment the testing process and improve the fault coverage. A computation-in-memory (CIM) based DFT is realized to expedite the detection and diagnosis of faults by developing logic designs involving multi-row activation. A novel addressing scheme is introduced to facilitate the diagnosis of faults. Reconfigurable logic designs are developed to detect unique RRAM faults that offer features such as programmable reference generations, period, and voltage of operation. DFT implementations are validated on a post-layout extracted platform and testing sequences are introduced by incorporating the proposed DFTs. Results show that more than 2.3× speedup and better coverage are achieved with 6× area reduction when compared with state-of-the-art solutions.**

***Index Terms*—Design-for-testability (DFT), Testing RRAM, computation-in-memory (CIM), binary logic, RRAM defects.**

I. Introduction

Resistive random access memory (RRAM) is one of the most promising emerging device technologies that exhibits non-volatility, compatibility to CMOS and high scalability, and can potentially replace conventional memories such as SRAM, DRAM and flash [1]. In addition, these devices enable emerging energy-efficient computation-in-memory (CIM) paradigms that allow computing within the memory units [2, 3]. However, manufacturing defects leading to unique faults have been identified in the RRAM production process when being integrated with CMOS [4]. Therefore, bringing RRAM to the market requires new dedicated test developments.

Traditional and well-established testing schemes for conventional memories are inadequate and cost-inefficient for testing RRAMs [4–6]. Several existing efforts that have focused on RRAM testing can be typically classified into two broad classes: march test algorithms and design-for-testibility (DFT) solutions. Efforts offering modified march test algorithms that involve specific sequences of *sequential* memory (read, write) operations to optimize test time [7, 8] or improve coverage [9–12] fail in the detection of *unique* RRAM faults [5] and typically DFT schemes are introduced to further enable improved fault coverage (FC) and/or optimized test time. However, existing dedicated DFTs are expensive in terms of hardware [13], optimistic regarding variations and lack implementations [14], impractical due to large voltage requirements [6, 15, 16] and exhibit functional issues due to to the reliance on sneak-paths [7, 17]. DFTs that target high FC typically involve slow, probabilistic write operations [6, 18–22]. In summary, there is a conspicuous need for low-cost DFTs and new testing algorithms dedicated to RRAMs that can offer optimal high-quality test solutions.

In this paper, we present low-cost DFT schemes by exploring reconfigurable CIM based logic operations to improve the FC and accelerate the testing process of RRAMs. In our approach, multi-operand NOR logic facilitates the detection and diagnosis of unique RRAM faults with $\mathcal{O}(1)$ time complexity. The contributions of the paper are:

- Proposes a DFT based on simultaneous access (read) of multiple bitcells in a multi-operand NOR logic of varying number of operands to optimize RRAM test time. The scheme includes a customized sequence of address selection patterns that aids the diagnosis of faults.
- Realizes reconfigurable DFTs to improve the FC. We explore programmable reference signals, voltage and duration of operations to develop high sensing margins that guarantee the detection of *unique* RRAM faults [5].
- Validates our compact DFT implementations with circuit-level simulations based on post-layout netlist that facilitates the execution of high quality test algorithms at lost-cost.

A comprehensive simulation platform built on 40*n*m TSMC CMOS technology demonstrates the advantages of our proposed DFT schemes in terms of testing cost and FC. Comparison results with state-of-the-art solutions show that more than 2.3× speed and higher FC is achieved with 6× area reduction.

The organisation of the paper is as follows. Section II introduces RRAM technology and Section III covers the targeted faults. Section IV presents our CIM-based DFT and Section V presents our DFT schemes with additional reconfigurable features. Section VI validates our DFT and develops testing algorithms. Section VII provides a comparison with the prior arts. Section VIII reflects on scalability of our schemes and future directions. Section IX concludes the paper.

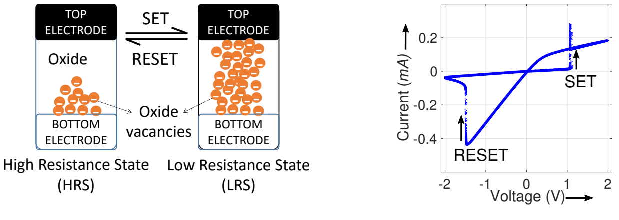
II. Resistive Random Access Memory (RRAM)

This section focuses on RRAM devices with binary storage capability (*bi-stable* element); however, this work can be easily

This work was supported by the EU H2020 grant “DAIS” with funding from the ECSEL Joint Undertaking under grant agreement No 101007273.

2378-2250/22/$31.00 ©2022 IEEE
DOI 10.1109/ITC50671.2022.00085

Regular Paper



(a) (b)

Fig. 1: (a) Typical RRAM device and its (b) I-V characteristics.

extended to multi-level storage elements and other memristor technologies such as phase change memory.

A. RRAM Technology

An RRAM cell structure consists of a metallic oxide that is sandwiched between a Top (TE) and a Bottom Electrode (BE) as described in Fig. 1a [23]. The working principle of RRAM devices is based on the reversible formation of a conductive filament (CF) and the absence and presence of this CF delivers high (HRS) and low resistance states (LRS), respectively. This describes the analog nature of RRAM to realise different states. The switching from HRS to LRS is called 'SET', whereas that from LRS to HRS is called 'RESET'. Fig. 1b shows I-V characteristics during the SET and RESET operations with voltage applied in opposite polarities. The HRS and LRS represent the logic state *1* and *0*, respectively.

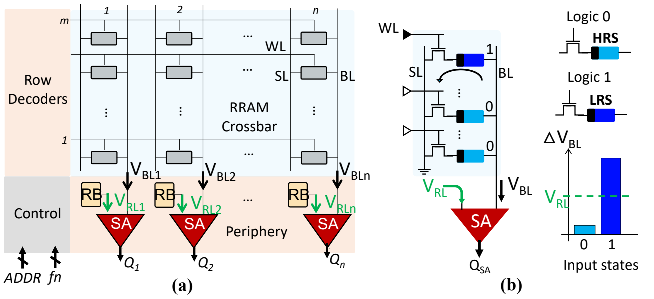
B. RRAM for Memory

A typical m×n RRAM architecture is shown in Fig. 2a. It consists of bitcell array in a crossbar arrangement with periphery blocks such as WL drivers, address decoders, control block, sense amplifier (SA), reference block (RB), etc. It uses address *ADDR* and function *fn* to define the location and type of operation to be performed, respectively. The focus of this work is (but not limited to) one-transistor-one-resistor (1T1R) bitcell configuration, since this is the most extensively explored configuration for high-speed memory [24] and for neural network realizations in a CIM architecture [25].

The 1T1R bitcell configuration is shown in Fig. 2b. The bitcell is accessed via the wordline (WL) that turns on the pass transistor (NMOS), while connecting the select line (SL) to the bitline (BL). In a write operation, BL is supplied with the write voltage and SL is connected to GND for the SET operation and vice-versa for the RESET operation. Fig. 2b shows how a read operation is performed. Before the start of the read cycle, BL is pre-charged to voltage supply (VDD) and SL is connected to ground (GND). WL activation selects a bitcell for reading that enables the BL to discharge through the RRAM device. The developed BL voltage V_{BL} is compared to a reference line voltage V_{RL}. The differential voltage $\Delta V = V_{BL} - V_{RL}$ is sensed using a SA to determine the state of the RRAM device.

III. Targeted Faults

Manufacturing defects that lead to an erroneous behaviour or a deviation from the intended behaviour are modelled as faults. This section summarizes the observed faults in RRAMs [4–6, 8, 9, 26]. These faults can be classified into following two classes: conventional faults and unique faults [5].



(a) (b)

Fig. 2: (a) RRAM-based memory. (b) Working principle of a typical read operation; dark (light) blue represents SET (RESET) RRAM state.

Conventional Faults are faults similar to those observed in traditional memories such as SRAM, DRAMs; they are:

- Stuck-at-faults (SAF) [27]: An RRAM device is stuck in a state and cannot switch.
- Transition faults (TF) [27]: An RRAM device fails to switch properly.
- State coupling faults (CFst) [9, 20]: State of an aggressor RRAM cell alters the state of the victim cell.
- Write disturbance faults (WDF) [9]: Unintentional alteration of the state of an RRAM device during a write operation.
- Incorrect Read faults (IRF) [18]: A read operation generates incorrect outputs whilst state is correct.
- Read-disturb faults (RDF) [8]: A read operation switches the state of the RRAM device, while the read value is correct.

Unique Faults are faults emerging due to the nature of RRAM devices. Besides LRS and HRS, a RRAM device can also occupy an undefined (*U*) state [27]. A *U* can be defined as a state which, when read, can give random outputs because the ΔV developed is less than the minimum sensing margins ΔV_{min} required by the SA to define a deterministic state. Note that the detection of such faults *cannot* be guaranteed with a typical read operation. Other possible states are high (*H*) and low (*L*) states [7]; *i.e.,* deep states that lay beyond the resistance ranges of LRS and HRS. Resistance of *H* is more than maximum HRS and that of *L* is less than minimum LRS. Detecting faults due to such states *cannot* be guaranteed with existing march tests, since these tests only allow fixed, predetermined patterns of *logic 1* or *logic 0* values corresponding to LRS and HRS, respectively. All unique RRAM faults are:

- Deep faults (DF) [7]: RRAM device falls into deep states.
- Undefined write faults (UWF) [27]: A write operation leads to *U* state.
- Unknown read faults (URF) [7]: A read operation switches the state of the RRAM device to *U* or/and produces random read outputs.
- Undefined coupling faults (CFud) [22]: State of an aggressor RRAM cell alters the state of the victim cell to a *U* state.
- Intermittent undefined state faults (IUSF) [4]: RRAM state intermittently changes its switching mechanism from bipolar to complementary, which affects write operations and causes undefined state faults.

Depending on the efforts needed to detect the targeted faults, these can be classified into *easy-to-detect* (ETD) and *hard-to-detect* (HTD) faults [28]. Detection of ETD faults *can* be guaranteed using regular memory operations, whereas detection of HTD faults *cannot* be guaranteed using these operations. Therefore, special DFT schemes are required to *guarantee* the detection of HTD faults.

IV. DFT Schemes Proposed for ETD Faults

This section presents DFT schemes to optimize the detection time of conventional ETD faults. These schemes are also the foundation of the DFT schemes to detect HTD faults.

A. Concept

The DFT is based on performing *multi-operand* NOR logic operations within the memory unit that involves *multi-row* read operations. Such a memory unit that is modified to perform *in-situ* logic operations is referred to as a computation-in-memory (CIM) unit [29, 30]. This allows the selection of multiple cells in parallel, and therefore, the execution of simultaneous read operations with *N* operands. The sensitization and detection of the faults by *N* sequential read operations can be accelerated by a factor of *N* by enabling simultaneous sensitization of the faults during an *N*-operand NOR operation. This logic operation can be represented by NOR^N and the proposed DFT can be referred to as DFT-NOR^N. Note that NOR^N can only replace *N read 0*, and not *N read 1* operations.

The working principle is illustrated using the detection of conventional SAF1 fault. Performing a correct fault-free NOR^N when all the bitcells are initialized to RESET (0) state returns *logic 1* as the output value; this is denoted as 0NOR^{N}1. However, if a faulty RRAM device is stuck at *logic 1*, NOR^N would result in an incorrect *logic 0*. To get a feeling of the potential speedup in RRAM testing, one can apply an appropriate march test algorithm while incorporating these logic operations as a detection sequence. Subsequently, *N* interleaving write and read 0 operations are replaced by *N* write operations and *one* NOR^N.

In addition to detection of a faulty behaviour, faulty cell location can be identified by deploying a binary search algorithm. By performing NOR operations with varying number of operands while selecting certain pattern of rows in a binary search manner, the address selection based on logic outcomes converges to the faulty cell location in $\log_2 N$ (logic) + 1 (read) operations. In summary, the fault detection is accelerated by a factor of *N* for every read 0 operation and faulty-cell identification by a factor of $\frac{N}{log_2 N+1}$.

B. DFT Design and Implementation

The DFT scheme is based on using multiple references for single read and multi-operand NOR logic operations of varying number of operands. In addition, this scheme requires a modified row address decoder to enable simultaneous multiple row activation. In this regard, we present a low-cost scalable multiple reference signals generator and row address decoder to facilitate the selection of multiple addresses.

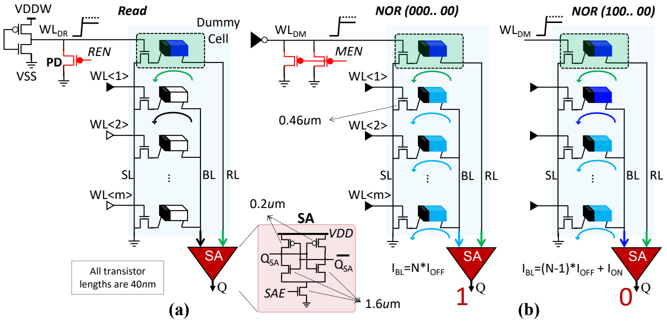


Fig. 3: Proposed dummy wordline lowering technique in reference signal generator implementations for (a) read and (b) NOR logic operation.

1) Reference Signals Generator: The multi-operand logic operation NOR^N creates *N parallel* and *independent* read paths. This implies that the total BL read current in such an operation is the aggregation of *N* bitcell currents. To perform a read or logic operation, the underlying concept is to generate a reference signal I_{ref} that guarantees the correct operation for all possible input states, including the *critical states* which are defined as the states with minimum $\Delta V=\Delta V_{min}$; ΔV_{min} is the minimum ΔV required by the SA. For a read operation, there are only two possible states; *i.e.*, RESET and SET states. Hence, I_{ref} has to be ideally at the middle of these two discharging currents; if we assume SET state current to be I_{ON} and RESET current to be I_{OFF} and $I_{ON} >> I_{OFF}$, then ideally I_{ref} should be about $I_{ON}/2$. On the other hand, for a NOR^N operation, these critical states are: (1) *all* cells are *RESET (0)* denoted by N(0) that results in a BL discharge current of $N*I_{OFF}$, and (2) *one SET (1)* and *N-1 RESET* states denoted by N(1) that results in a BL discharge current of $I_{ON}+(N-1)*I_{OFF}$. Since these two states produce accumulation of several I_{OFF} currents, the assumption of using $I_{ON}/2$ as a reference is not valid as *N* increases. Hence, an appropriate I_{ref} needs to be regulated accordingly.

In our approach, we introduce a row of dummy RRAM bitcells to generate the required I_{ref} by regulating its wordline voltage WL_{DR}, as shown in Fig. 3. The following modifications are required: (1) one dummy row (*i.e.,* one dummy cell per column) to be programmed to SET (or LRS), (2) A bleeder circuit introduced in the wordline driver corresponding to the dummy row; this technique, typically used in a read-assist circuit in SRAMs [31], degrades or lowers the wordline voltage to a pre-defined value. This changes the overall conductance of the bitcell by changing V_{GS} of the pass transistor operating in the linear mode. Therefore, for any operation, we first determine the currents corresponding to critical *logic low* and *logic high* states that are responsible for discharging BL, and configure the dummy wordline signal in such a way that it generates the average of these critical current values.

During a read operation, the pre-charged reference line RL is discharged through the dummy cell selected with the lowered WL_{DR}, as shown in Fig. 3a. The active low signal *REN* is switched to logic 0 (VSS) to enable the bleeder PMOS 'PD'. Note that the strength of the bleeder PMOS 'PD' is such

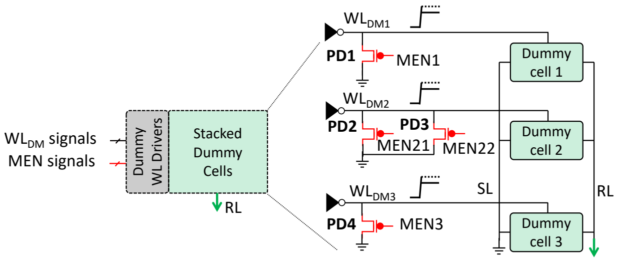


Fig. 4: Proposed configurable reference generator to enable NOR logic of varying number of operands.

Details	Parameters	Possible currents (in I_{OFF})	Number of Operands (N)					
			[1-8]	**16**	**32**	**64**	**128**	**256**
Critical currents (normalized to I_{OFF})	Logic low	-	8	16	32	64	128	256
	Logic high	-	107	115	131	163	227	355
	Ideal Ref.	-	53	66	82	114	178	305
Dummy cell 1	MEN1	50, 100	0	0	0	0	0	1
Dummy cell 2	MEN21/22	16, 32, 64, 100	x	00	01	10	10	11
Dummy cell 3	MEN3	64, 100	x	x	x	x	0	1
Generated Iref (in I_{OFF})		-	50	66	82	114	178	300

Signs	Meaning
-	Not applicable
x	Corresponding WLDM is not selected
MEN* = 0 (or 1)	Enables (or disables) corresponding PMOS bleeder device

TABLE I: Different configurations to generate references for multi-operand NOR with varying number of operands; it is assumed that I_{ON}=100*I_{OFF}.

that a voltage divider between pull-up transistor of the WL_{DR} driver and 'PD' configures WL_{DR} to drive a current of $I_{ON}/2$ (being the ideal I_{ref}) for the dummy cell. A differential BL/RL voltage ($\Delta V=V_{BL}-V_{RL}$) is created which is sensed using a differential SA. We use a cross-coupled SA that involves a positive feedback loop to amplify the input differential voltage, shown in Fig. 3a.

This approach is scaled to perform multi-operand logic operations. Fig. 3b shows a possible design that uses two PMOS bleeder devices to configure the required reference current. However, a single dummy bitcell cannot be configured for a higher number of operands. Assuming an equivalent resistance ratio of the HRS and LRS of 100 (I_{ON}=100*I_{OFF}) [32], it can be seen that it is not possible to generate I_{ref} above 100*I_{OFF} using a single dummy cell in the SET state. For instance, I_{ref} in NOR^{256} requires the average current I_{ref} of the two currents generated by the two critical states *logic low* 256(0) with I_{BL}=256*I_{OFF} and *logic high* 256(1) with I_{BL}=255*I_{OFF}+I_{ON}=355*I_{OFF}; in this case I_{ref} ≈300*I_{OFF} ≈3*I_{ON}, requiring at least three dummy cells. Moreover, additional reconfigurability is needed to generate references for NOR^N with varying N (N=2^z and 1<z<8) that can perform the binary search algorithm. In achieving this, we introduce a total of three dummy cells to perform up to NOR^{256}, as shown in Fig. 4. The size of the PMOS devices 'PD[1-4]' are such that a pre-determined combination of some or all dummy cell currents can generate the required I_{ref} currents. Table I shows the different configurations required to set up appropriate reference signal generations. These configurations ensure correct multi-operand NOR operations. Note that the bitlines are capable of allowing high multi-operand currents, since these wires are conditioned to allow typically large programming currents. Therefore, peak power and current issues are not expected. The top part of the table shows the currents generated by the two critical states *logic low* and *logic high* (normalized to I_{OFF}) and the corresponding ideal I_{ref} currents for different values of operands. The second part of the table lists the different configurations of the three dummy cells leading to appropriate I_{ref} values. Here, column 3 enlists the possible pre-determined currents that different configurations of the corresponding enable signal with prefix *MEN* can provide. As an illustration, lets consider N=64. The two critical states 64(0) and 64(1) result in currents 64*I_{OFF} and 163*I_{OFF}, respectively; the ideal I_{ref} in this case should be the average of the two; *i.e.*, 114*I_{OFF}. Referring to Fig. 4, with *MEN1*=0 (PMOS PD1 is ON), dummy cell 1 delivers 50*I_{OFF}; with *MEN21:MEN22*=10 (PMOS PD2 is ON, PD3 is OFF), dummy cell 2 delivers 64*I_{OFF}; '×' corresponding to *MEN3* implies WL_{DM3} is OFF and dummy cell 3 delivers no current; hence a total of 50*I_{OFF}+64*I_{OFF}=114*I_{OFF} as I_{ref} is generated. Similarly, it can be seen that all required I_{ref} currents can be generated to guarantee correct logic operations.

2) Row Address Decoder: Row address decoder circuit in a 256-row RRAM memory decodes the input address vector AA[7:0] to select one row for memory operations. Typically, AA[7:0] bits are latched and the decoder generates the true and bit-wise complementary form of these inputs, *i.e.*, A[7:0] and A[7:0]' to perform the decoding, as shown in Fig. 5a. Similarly, in test mode, address AA[7:0] is decoded to select a row for testing. To select multiple rows, we propose to set all AA[7:0] bits to 1 and force some or all the complementary bits to 1, thereby, overriding the complementary circuit. This implies that all targeted WLs will be activated simultaneously. For instance, all (256) WLs are activated if all A[7:0] and A[7:0]' are forced to 1. In another instance, if LSB A_0=1 is allowed to have both true and complementary values, all the odd row numbered WLs are activated.

The addressing scheme is described in Fig. 5b. Our scheme is implemented by replacing the inverters with NAND gates that are responsible for generating complementary bits A[7:0]' in the first stage of decoding. The inputs to these NAND gates are the latched AA[7:0] bits and the newly introduced TAA[7:0] control inputs generated by a 3-to-8 Johnson counter (JC) [33]. During a normal memory operation, test enable signal TEN=0 and AA[7:0] propagates normally to perform single row selection (TAA[7:0] are kept as 11111111 by TEN=0). In test mode, TEN=1 enables the JC to generate TAA[7:0] pattern that selects the required set of rows in each cycle (TEN=1 initially resets TAA[7:0] to 00000000). As explained earlier, a binary search engine requires the number of selected addresses in a logarithmic manner in each cycle, *i.e.*, the first cycle requires 256 WL activations, the second cycle requires 128, the third cycle requires 64 and so on. Fig. 5c depicts a possible set of address selections in each cycle. Note that replacing inverters with NAND gates in the decoder circuit has negligible impact ($\sim 10\,ps$) on the critical path delay (i.e., operating cycle of the memory operations).

In summary, we realize the concept and methodology of our DFT approach with significantly low-cost design components; *i.e.*, a reference generator and a row address decoder to accelerate the detection of *ETD faults*. However, a group of

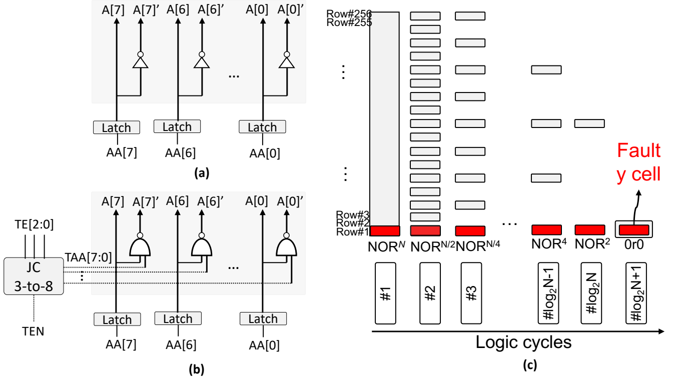

A[7] A[7]' A[6] A[6]' A[0] A[0]'
Latch
AA[7] AA[6] AA[0]
(a)
TE[2:0]
JC 3-to-8
TAA[7:0]
TEN
(b)
Row#256
Row#255
Row#3
Row#2
Row#1
NOR^N NOR$^{N/2}$ NOR$^{N/4}$ NOR^4 NOR^2 0r0
Fault y cell
#1 #2 #3 #log$_2$N-1 #log$_2$N #log$_2$N+1
Logic cycles
(c)

Fig. 5: (a) Typical row address decoder for single row selection (b) Proposed circuit implementation of modified address decoder to realize binary search algorithm, and (c) the address selections for a given column.

fixed reference signals cannot guarantee the detection of *HTD faults* by read or NOR logic operations. In this regard, we propose a reconfigurable DFT which is described next.

V. Reconfigurable DFT Schemes Proposed for HTD Faults

This section presents low-cost reconfigurable logic design-based DFT schemes to target unique HTD RRAM faults.

A. Concept

The idea is to make the DFT design capable of differentiating BL read voltage developed in the faulty case from the fault-free case with high certainty. Fig. 6 briefly illustrates the DFT concept. *Weak 1* and *weak 0* are defined as *faulty U states* that are closer to *logic 1* and *logic 0* states, respectively. These are imaginary states that allows us to discuss worst-case scenarios for detecting faulty *logic 1* and *logic 0* states. The following techniques change the condition of detecting a faulty state from a random read to a deterministic read output and are built on top of DFT-NOR^N. There are *two possible ways* to detect the faulty states, as shown in Fig. 6:

- Enable *extended time (ET) of operation* to give sufficient time for the development of $|\Delta V| > \Delta V_{min}$, and shift the reference signal ΔV_{RL} to the middle of the two ΔV_{BL} developed by critical *logic 1* and *logic 0* states. This will be referred to as DFT-ET-NOR^N. This DFT requires a reconfigurable period of operation and a reconfigurable reference generator.
- In addition to the above technique, enable *high resistance ratio (HR)* of these two critical cases, e.g., one possible way is to decrease the resistance offered by the pass transistor by increasing the WL voltage $V_{WL_{DM^*}}$ (see Fig. 7); this also requires a shift in ΔV_{RL} but since this technique allows early development of the required margins compared to the case when DFT-ET-NOR^N is used, the magnitude of the shift is less. In other words, an increase in resistance ratio can allow fast detection of HTD faults compared to DFT-ET-NOR^N. This will be referred to as DFT-HR-ET-NOR^N. This DFT requires a reconfigurable period of operation, a reconfigurable reference generator and a reconfigurable resistance ratio.

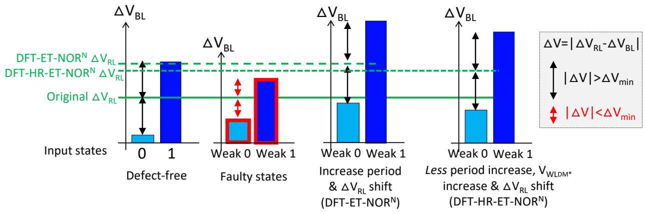

ΔV_{BL}
DFT-ET-NOR^N ΔV_{RL}
DFT-HR-ET-NOR^N ΔV_{RL}
Original ΔV_{RL}
Input states
0 1
Defect-free
Weak 0 Weak 1
Faulty states
Weak 0 Weak 1
Increase period & ΔV_{RL} shift (DFT-ET-NOR^N)
Weak 0 Weak 1
Less period increase, V_{WLDM^*} increase & ΔV_{RL} shift (DFT-HR-ET-NOR^N)
$\Delta V = |\Delta V_{RL} - \Delta V_{BL}|$
$|\Delta V| > \Delta V_{min}$
$|\Delta V| < \Delta V_{min}$

Fig. 6: Concept of detecting faulty RRAM states.

These solutions require three design components that are described in the next subsection. In summary, the duration of the BL discharge is increased to increase the developed ΔV and the reference signal is tuned such that:

- If the cell is fault-free, the shift of reference signal is such that $|\Delta V| > \Delta V_{min}$, but the original polarity of ΔV is maintained to produce the correct read output. Otherwise, the test may declare a fault-free cell as faulty.
- If the cell is faulty (*i.e*, *U* state fault), the reference signal is shifted such that $|\Delta V| > \Delta V_{min}$, and polarity of ΔV is opposite to the one in the fault-free case; this enables the detection of the faulty cell.

B. Design and Implementation

To realize the programmable DFT scheme, three design techniques will be used; they are explained next.

1) Reconfigurable Period of Operation: The (active) period of a read or logic operation, *i.e.*, period of WL activation, must be increased to allow sufficient time for fault detection. For example, the typical technique of using extra margin adjustment (EMA) that enables the reconfigurability of the active period of operation (i.e., duration of WL activation) in SRAMs [34] can be used. Here, the start of the active clock cycle is unchanged but the duration can be regulated by delaying the termination of the active clock with different permutations of the EMA signals. For instance, 3 EMA pins allows 8 different periods of operation.

2) Reconfigurable Reference Generator: The DFT must be capable of shifting the reference signal with programmable magnitudes. To achieve this, we modify the reference generator of Fig. 4 by adding an additional row of dummy cells with some PMOS bleeder devices in the WL driver as shown in Fig. 7. This dummy row can supply additional I_{ref} to shift V_{RL} lower than the normal case. The number of bleeder

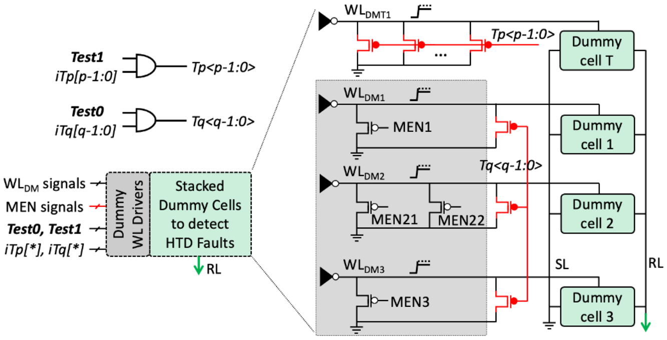


Fig. 7: Proposed reconfigurable reference generator to detect HTD faults.

PMOS devices, say p, in the WL driver of this dummy row can be configured to provide up to 2^p different possible reference signal strengths. On the other hand, additional bleeder PMOS devices can be added to existing dummy WL driver or drivers to reduce the strength of I_{ref} to shift V_{RL} higher than the normal case. In a similar way, additional q PMOS devices can possibly configure up to 2^q different reference signal strengths. The enable signals *Test0* and *Test1* activate the detection of *weak 0* and *weak 1* states, respectively. *Test1* also activates the newly added 'Dummy cell T'. Enable signals Tp[p-1:0] and Tq[q-1:0] regulates the magnitude of I_{ref}.

3) Reconfigurable Resistance Ratio: The effective resistance ratio of the faulty and fault-free states is increased to develop high read margins resulting in the detection of certain *unique* RRAM faults. To achieve this, we increase the global WL voltage VDDW of the WL driver in Fig. 3 (*i.e.*, increase V_{GS} of the NMOS pass transistor) to reduce its resistance and increase the overall resistance ratio. To illustrate this concept, consider the resistance offered by NMOS to be R_N, and resistances of RESET and SET states to be R_R and R_S, respectively ($R_R >> R_S >> R_N$). In the series combination of NMOS and RRAM device in a bitcell, change in R_N will have a much greater impact on the effective resistance of the SET state (R_N+R_S) compared to the RESET state (R_N+$R_R \approx R_R$). This implies that an increase in WL voltage does allow more discharge of V_{BL} in all cases, but this discharge is more severe if the bitcell is in LRS. This increases the chances of detecting a faulty state as well as allows faster detection of the faulty cases. Below is a quantitative illustration to show the increase in resistance ratio (RR) of the faulty and the fault-free state. Let the resistance of the faulty RRAM state be R_F=100KΩ, R_S=8KΩ and R_N at 0.9V and 1.1V be 2KΩ and 1KΩ, respectively. The RRs in these scenarios are;

$$@0.9\text{V: RR} = \frac{R_N+R_F}{R_N+R_S}=\frac{102}{10}\approx 10,\ @1.1\text{V: RR} = \frac{101}{9}\approx 11$$

Here, we see that shift in both RR (from 10 to 11) and LRS (from 10 to 9) are effectively about 10%, whereas the shift in faulty state resistance is only ∼1%. This validates the underlying concept of our DFT scheme. Note that DFT-HR-ET-NOR^N allows faster detection of HTD faults at the expense of increased voltage of operation. Hence, a trade-off between the use of higher voltage and optimization of test time.

In summary, we realize the concept of our DFT approach with low-cost design components to accelerate the detection of ETD and HTD faults and improve the FC.

VI. DFT Validation and Test Development

This section presents validation of our DFT implementations and develops new testing algorithms. We show the potential to provide low-cost and efficient testing methodology.

A. Setup, Fault Modeling and Analysis

1) Simulation Setup: The left part of Table II presents the details of our simulation platform. A 256x256 RRAM memory is built using industry-standard TSMC 40 nm CMOS device technology. The layout of the sense amplifier (SA) described in Fig. 3 is developed and extracted to establish minimum differential sensing margins of 40 mV. The rest of the digital components such as address decoders, drivers, control block etc. are built using standard cell extracted netlists to develop a comprehensive platform for circuit simulations.

Parameters	Specifications	Parameters	Specifications
Simulation Platform		**RRAM Bitcell**	
Simulator	Cadence Spectre	Configuration	1T1R
R/W Voltage	0.9 V/2.5 V ±10 %	RRAM Device	HfO_2/TiO_x [32]
R/W/NOR Time	0.7 ns/2 ns/1.2 ns	HRS/LRS	1 $M\Omega$ / 10 $K\Omega$
CMOS	40 nm TSMC, 3 σ	1T (NMOS: W/L)	460 nm/40 nm
Temperature	-40°C to 125°C	BL (WL) res.	0.2 Ω (0.4 Ω)
Memory Core		BL (WL) cap.	0.3 fF (0.6 fF)
Array	256x256MUX4	n_{min}, n_{max}	0.03, 30
SA	5T voltage-based	l_{ret}	0.05 nm
SA V_{min}	40 mV	r_{ret}	10 nm

TABLE II: Design parameters.

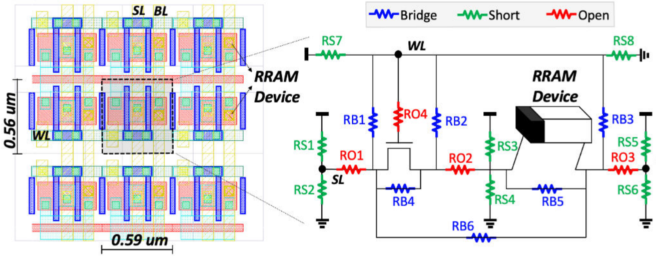


Fig. 8: Layout of a 3x3 RRAM bitcells and derived defect model.

The right half of Table II covers the details of the RRAM device model. A physics-driven HfO_2/TiO_x-based RRAM device (Verilog-A) model is used for our simulations that is based on physical dimensions and the ions concentration of oxide [35]. The cylindrical RRAM device with the oxide concentration has a height l_{det} and radius r_{det}, and the minimum (n_{min}) and maximum (n_{max}) concentration of oxide ions corresponding to the LRS and HRS, respectively, can be set. The internal state parameter n_{real} represents instant oxide ions concentration, thus defining the state of an RRAM device at any given time. A higher (lower) value of n_{real} corresponds to the LRS (HRS).

To accurately capture the wire parasitics of the RRAM memory bitcell array and coupling effects of the neighboring bitcells, the layout of a 3x3 memory matrix is developed and extracted to derive the middle bitcell, as shown in Fig. 8a. Defects in the RRAM device, interconnects and transistors are modelled as linear resistors. All these defects are described as one of the three types of defects, as shown in Fig. 8b; short-circuit to power signals VDD or VSS (resistors are with prefix *RS*), open-circuit or broken connection (*RO*) and a bridge (*RB*) between any two nodes other than VDD or VSS. The resistance of these linear resistors range from 1Ω (2^0Ω) to 134MΩ (2^{27}Ω) swept with a geometric sequence of common ratio 2^3Ω (10 possible values) to capture any possible defects. This also covers the minimum and maximum range of RRAM effective resistance *i.e.,* 10KΩ and 1MΩ, respectively.

2) Fault Modeling and Analysis: To identify faulty behaviour due to possible defects, simulations are conducted with a defect-free netlist and are then compared with those conducting based on defect-injection in the netlist. The aforementioned defects are injected one-at-a-time at the considered locations. The memory and DFT related logic operations considered are:

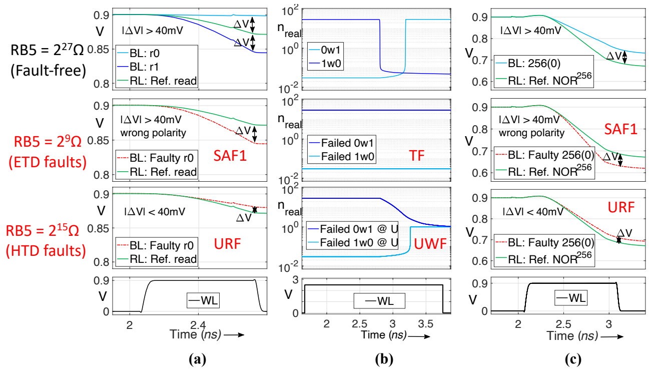


Fig. 9: Timing diagram (from the top): Correct, ETD and HTD faulty behaviours of (a) read, (b) write and (c) $0\text{NOR}^{256}1$ operations. Note that these operations have different WL activation periods.

- *0w0*: write 0 (RESET) to a cell initialized to 0.
- *0w1*: write 1 (SET) to a cell initialized to 0.
- *1w0*: write 0 (RESET) to a cell initialized to 1.
- *1w1*: write 1 (RESET) to a cell initialized to 1.
- *0r0*: read a cell initialized to 0 with expected value 0.
- *1r1*: read a cell initialized to 1 with expected value 1.
- *$0\text{NOR}^N 1$*: N-operand NOR operation with all cells initialized to 0 with expected value 1.

Fig. 9 shows the simulation results for a read, write, and NOR operation for one defect RB5. Based on the strength of the defect, we can distinguish three cases: RB5=$2^{27}\Omega$ fault-free (first row), RB5=$2^{9}\Omega$ ETD fault (second row), $2^{15}\Omega$ HTD fault (third row). The bottom row shows the related WL timing. The voltage and timing specifications associated with these operations are described in Table II. Next, we explain the observations for all three cases. Case fault-free: Fig. 9a shows that when reading a cell ΔV>40mV, and thus the cell is read without any faults. Fig. 9b shows that the cell can transition both from 1→0 and from 0→1. Finally, Fig. 9c shows that for the NOR operation ΔV>40mV as well, and thus that this operation also succeeds. Case ETD (second row): Fig. 9a shows that as the defect RB5=$2^{9}\Omega$ provides a low ohmic read path (SAF1), a higher discharge current flows than expected. This results in ΔV>40mV but with incorrect polarity. Hence, the operation is faulty as the read output is 1 when the expected value is 0. Fig. 9b shows that the low ohmic bridge results in a low voltage across the RRAM device. Hence, the two write operations are faulty (TF). Similar to the faulty read case, Fig. 9c shows a faulty NOR operation due to the low ohmic read path. Case HTD (third row): Fig. 9a shows that ΔV<40mV as the strength of the defect RB5=$2^{15}\Omega$ is in the middle between LRS and HRS resistance values. Hence, the read operation may produce a random read output as the effective RRAM resistance falls in the U state. Fig. 9b shows that the lack of sufficient voltage when writing across the RRAM device causes the RRAM to fall in the U state (UWF). Similar to the faulty read case, Fig. 9c shows a faulty NOR operation as the effective RRAM resistance falls in the U state.

B. DFT Validation

1) Targeting ETD Faults: ETD faults provide sufficient sensing margins (|ΔV|>40mV) with incorrect polarity. Fig. 10 shows the simulation results to illustrate the detection of ETD faults by deploying DFT-NOR^N (where, N=256). We simulated the defects that introduce stuck-at-faults; RB5 is set to $2^{9}\Omega$ (refer to Fig. 8b) to simulate SAF1 and RO2 is set to $2^{18}\Omega$ to simulate SAF0. On the left side of the figure, defect-free and SAF1 cases corresponding to 256(0) state condition are presented, where the faulty case allows a higher I_{BL} that discharges the BL more than expected. This fault can be detected by performing $0\text{NOR}^{256}1$ operation as the faulty case results in output 0 (the expected result is 1). Similarly, on the right side of the figure, defect-free and SAF0 cases corresponding to 256(1) state condition are presented, where the faulty case allows a lower I_{BL} that discharges the BL less than expected. This fault can also be detected by performing $1\text{NOR}^{256}0$ operation as the faulty case results in output 1 (the expected result is 0).

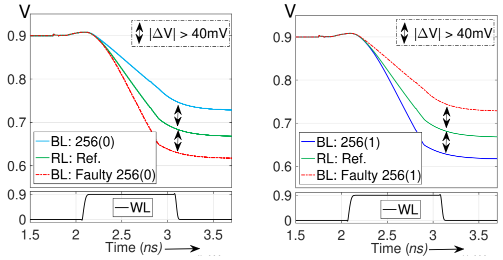


Fig. 10: Accelerating the detection of ETD faulty behaviour using NOR logic.

To validate the binary search technique described in Section IV, we introduce an ETD fault at row address location 215 (binary: 11101011) and then determine its location. RB5 is set to $2^{6}\Omega$ in the bitcell present at this address location to simulate an SAF1. Table III illustrates how the identification of this faulty cell can be achieved after 8 ($\log_2 256$) consecutive NOR operations of varying operand size; *i.e.* by each NOR cycle from NOR^{256} to NOR^{2}, the selected addresses converge to the faulty cell address location. First, we initialize AA[7:0]=11111111 and enable signals TEN=1 (test mode ON) and TEN[2:0]=000 such that TAA[7:0]=00000000. This ensures all 256 address lines are selected in the first cycle. Following the sequence of NOR operations, with every kth NOR operation, the JC increments such that kth LSB of TAA[7:0] is toggled to 1 independent of the outcome. This allows true and complementary form of first up to kth LSB of AA[7:0] to propagate in A[7:0] and A[7:0]' respectively. However, AA[7:0] follows a different sequence depending on the outcome of the operation. Every incorrect kth NOR operation implies that the faulty cell is selected in that cycle. Therefore, AA[7:0] is not changed. However, if the kth NOR operation is correct, then it implies that the faulty cell is not selected; hence for the next cycle, kth LSB of AA[7:0] is reset

Cycle	Outcome	TE[2:0]	TAA[7:0]	AA[7:0]	A[7:0]	A[7:0]'
1 (NOR^{256})	wrong	000	00000000	11111111	11111111	11111111
2 (NOR^{128})	wrong	001	00000001	11111111	11111111	11111110
3 (NOR^{64})	correct	010	00000011	11111111	11111111	11111100
4 (NOR^{32})	wrong	011	00000111	11111011	11111011	11111100
5 (NOR^{16})	correct	100	00001111	11111011	11111011	11110100
6 (NOR^{8})	wrong	101	00011111	11101011	11101011	11110100
7 (NOR^{4})	wrong	110	00111111	11101011	11101011	11010100
8 (NOR^{2})	wrong	111	01111111	11101011	11101011	10010100
9 (Read0)	wrong	-	11111111	11101011	11101011	00010100

TABLE III: Address selections to realize binary search algorithm.

to 0. The illustrative example presented in the table shows that operations of cycles 3 and 5 are correct, the 3rd LSB and 5th LSB of AA[7:0] are toggled to 0, while the others remain at 1. The final cycle is a read operation as ultimately one of the two addresses selected in NOR^2 operation is faulty. In summary, the faulty cell is identified in 9 (log$_2$256+1) cycles.

2) Targeting HTD Faults: Reading a faulty cell suffering from a HTD fault produces random read outputs; this is because the developed ΔV is below ΔV$_{min}$. The DFT techniques that are proposed increase the difference in the developed say ΔV, between the faulty and fault-free state to differentiate one state from the other. We refer to Fig. 11 to validate our DFT schemes. The top sub-figure shows the simulation results without any DFT; the middle sub-figure shows the results when DFT-ET-NOR^N is deployed and the bottom sub-figure the results when DFT-HR-ET-NOR^N is deployed. Here, lets consider a logic NOR operation NOR^{256} where in one of the two critical cases, say 256(0), one device is in the *U* state, as shown in the top part of the figure. This shifts the input signal V$_{BL}$ closer to V$_{RL}$, such that it violates the minimum sensing margin of ΔV$_{min}$=40 *m*V to a value of 20 *m*V. This faulty behaviour can be detected by increasing the time of operation (here, by 4X) such that V$_{BL}$ in the faulty case becomes at least 80 *m*V more than the fault-free case and then configuring V$_{RL}$ >40 *m*V from both faulty and fault-free V$_{BL}$ signals, as shown in the middle part of the figure.

The bottom part of Fig. 11 validates DFT-HR-ET-NOR^N, which can accelerate the detection of HTD faults by improving the read margins. The figure shows that an increase in V$_{WL}$ pushes V$_{BL}$ developed with *weak 0* into a range where, by increasing the time (here, by 3X) and shifting the reference signal, the *weak 0* cell is read as 1. In a similar way, *weak 1*, *L* and *H* states can be detected with pre-determined reference signals associated with each of these states.

Similar to the ETD fault cell identification, 8+1 consecutive NOR+ read cycles converge the address decoder to determine the address with the faulty state bitcell.

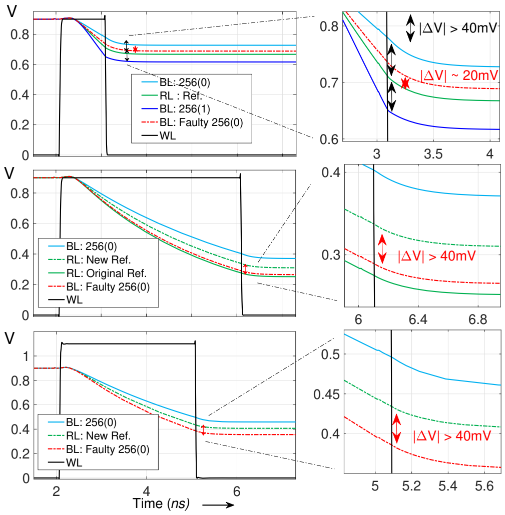


Fig. 11: Increased time of operation and reference shift to detect HTD faults.

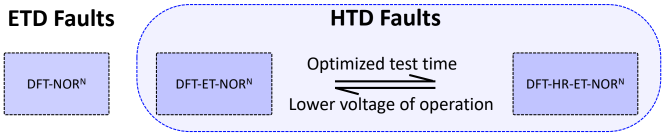


Fig. 12: Summary of the proposed DFT schemes to develop test procedure.

C. Test Procedure

Any memory configuration built on any memory technology that involves read operation based on the difference in BL discharge currents (or developed BL voltages) can benefit from our proposed DFT schemes. Following this, we develop a test procedure for high volume production that deploys our proposed DFT schemes in an efficient manner; i.e., targeting the maximum FC while minimizing test time. Fig. 12 summaries the three proposed DFT schemes. We show that DFT-NOR^N can be used to speed-up the execution of any test algorithm for ETD faults detected by read operation. In addition, DFT-ET-NOR^N and DFT-HR-ET-NOR^N schemes can be used to increase the FC of any test targeting HTD faults as these DFTs enable deterministic read operations rather than random reads for some HTD faults. DFT-HR-ET-NOR^N provides similar FC with faster detection of the target faults compared to DFT-ET-NOR^N at the expense of increased WL voltage. In short, for the most optimized test time with maximum FC, DFT-HR-ET-NOR^N must be used. The idea to develop a testing sequence is to begin with the regular march test i.e., MATS+ algorithm [36] and then modify the algorithm incorporating our DFT schemes. More specifically, replace r0 with NOR^{N}1 for the detection of faults. As a illustration, we present our proposed march test algorithms to detect and determine the location of the URF (notation adapted from [36]). The URF is sensitized by 1w0 operation that can switch the faulty RRAM cell from 1→U state and the following NOR^{N}1 detects this faulty state. Similarly, the 0w1 operation can sensitize this fault by switching the faulty RRAM cell from 0→U state and the following r1 operation detects this state. Additionally, we duplicate every write operation to increase the possibility of capturing IUSF. Following are the march test algorithms to detect and determine the location of all possible faults with test lengths of 6N+8 and 6N+8log$_2$$N$+8, respectively, where N is the RRAM memory size:

$$\text{Fault detection: } \{\Updownarrow_N(\text{w0w0});\ \Uparrow_4(\text{NOR}_x^N 1);\ \Uparrow_N(\text{w1w1}); \\ \ldots\ \Downarrow_N(\text{r}_x 1,\ \text{w0});\ \Uparrow_4(\text{NOR}_x^N 1)\}$$

$$\text{Fault detection: } \{\Updownarrow_N(\text{w0w0});\ \Uparrow_{4log_2N}(\text{NOR}_x^* 1);\ \Uparrow_4(\text{r}_x 0); \\ \text{and location} \quad \ldots\ \Uparrow_N(\text{w1w1});\ \Downarrow_N(\text{r}_x 1,\ \text{w0}); \\ \ldots\ \Downarrow_{4log_2N}(\text{NOR}_x^* 1);\ \Uparrow_4(\text{r}_x 0)\}$$

Here, we define NOR, read 0 and read 1 operations as NOR$_x^N$1, r$_x$0 and r$_x$1 when DFT-HR-ET-NOR^N is deployed. Here, x can be *H*, *weak 0*, *weak 1* or *L* and they correspond to reference signal in the mentioned states), respectively. Note that NOR$_x^N$1 and r$_x$0 operations are performed four times ($\Updownarrow_4$) to capture these four faulty state conditions. NOR$_x^*$1 implies that these NOR operations assume varying number of operands required for binary search algorithm.

VII. Prior Arts and Comparison Results

In this section, we present a qualitative and a quantitative comparison with the state-of-the-art solutions.

A. Qualitative Comparison

Many test solutions for RRAMs have been presented in literature. They are summarized in Table IV. The tests can be classified on their type, i.e., whether they are a march algorithm only, or whether they are a DFT (possibly in combination with a march algorithm). The table lists for every fault if it can be detected by the test. It follows that march algorithms only solutions are unable to achieve a high FC. For e.g., March W-1T1R achieves only 64% FC. This is because RRAM unique faults are HTD, and thus require DFT to be reliably detected. The existing DFTs can be further divided into DFTs that speed up testing [7, 15, 20], and DFTs that increase the FC [13, 18, 22].

1) DFTs to Optimize Test Time: DFTs that speed up the testing process either reduce the duration of the write operation during testing [20], or they read multiple cells at once and thus reduce the number of read operations [7, 15]. In general, these DFT schemes do not result in high FC. In [20], the authors propose to shorten the write *0* operation so that it just barely switches into the HRS range and check it afterwards by using a dedicated reference signal. The assumption is that faulty cells will not switch in time and thus will be detected. However, our DFT schemes do not involve dedicated write operations and, moreover, replace the large number of sequential read operations to a single NOR operation to optimize the test time. In [15], a MAGIC CIM-based NOR operation is used to perform logic in parallel. However, the voltages required to do this are higher than $7\,\mathrm{V}$, which limits the applicability. In contrast, our DFT schemes are based on read or NOR operation that typically operate at low voltages *i.e.,* $0.9\,\mathrm{V}$. In [7], multiple cells are selected and the sneak paths are used to detect faults. However, selecting groups of multiple cells requires an expensive modification of the decoder circuitry and sneak paths may cause functional issues. In contrast, a small number of additional components consisting of a few PMOS transistors and bitcells per column are introduced to implement our DFTs.

2) DFTs for Higher FC: DFTs that increase FC either modify the read or write operation so that HTD faults are sensitized and subsequently detected. In [18], the write operation is weakened so that faulty cells will fail to switch properly. After such an operation, faulty cells will be in a wrong state, while fault-free ones will be in the correct state. The drawback of this DFT is that it needs to be calibrated precisely, to prevent that the weak write operation may also fail on good cells. In our case, the DFTs consist of read operations that do not rely on delicate programming of the RRAM device. In [22], two new references are introduced that can be used to detect the *U* state, i.e., one reference between *1* and *U*, and one between *U* and *0*. The drawback of this is that the sense margin of the SA may cause some faulty cells in *U* to be incorrectly read out as *1* or *0*, while some good cells in *1* or *0* will be read as *U*, resulting in yield loss and test escapes. In contrast, we ensure sufficient read margins by deploying DFT-ET-NOR^N and DFT-HR-ET-NOR^N schemes to have deterministic read and NOR operations. In [13], a sensor measures and compares the internal node of each cell with pre-determined references to detect the state of the cell. It is unclear how this sensor can be adapted to work for multiple cells in an economic way, if every internal node needs to be measured. However, our DFTs do not involve any complicated structures and allow low-cost integration within the memory.

In short, no previous test solutions detect all RRAM faults in a reliable and cost-effective manner. Furthermore, none of the tests are able to guarantee the detection of the IUSF.

B. Quantitative Comparison

The testing cost in terms of area overhead and test length, and the FC are compared with the state-of-the-art solutions in Table IV. In the table, we show that our proposed DFT detects all faults with partial detection of the IUSF. This is because of the probabilistic nature of the occurrence of this fault. However, as several NOR and read operations are performed on the faulty cell ($\log_2$N+1 cycles), we increase the chances of detecting this fault. With better FC than Enhanced March [22], an area-efficiency of 6× (normalized to number of transistors) and 2.3× speed (test length) is achieved, assuming number of rows N_r and columns N_c of the memory are 256 each.

VIII. Discussions and Future Directions

This section highlights the applicability and adaptability of our DFT schemes with different design technologies.

1) Bitcell Configuration: Memory units built on any bitcell configuration that involves read operation based on the difference in BL discharge currents (or developed BL voltages) can benefit from our proposed DFT schemes. BL voltage (or current) in any such configurations can be compared with RL developed by deploying our reference techniques.

2) Multi-bit capacity: Multi-bit capability of memristors have been greatly explored to increase the storage density. In terms of RR, we can assume a reduction in the difference of effective resistance states; this is because more quantized resistance states (>2) have originated from a similar range that is associated with only two states in a 1-bit RRAM. However, this reduction only reduces the minimum possible ΔV_{BL} of the two critical states which can be alleviated using a longer cycle of operation. In short, a reduced maximum number of operands in the NOR operation or increase in cycle time can still offer fast testing and facilitate improved FC.

3) Other Memristor Technologies: The proposed DFT scheme requires high RR to accelerate the detection of faults. It stems from the fact that in a NOR^N operation, the difference in V_{BL} developed with critical states N(0) and N(1) increases as the RR increases and vice-versa (see Eq. 1). Therefore, memory technologies such as phase-change memory with possible high RR can be expected to have low-cost testing with high FC, whereas, spin-torque transfer magnetic random access memory (STT-MRAM) with inherently low RR can have improved FC.

Name	Type	Conventional						Unique					Fault Coverage	Test Length Write, Read	Cost # of transistors
		SAF	TF	WDF	IRF	RDF	CFst	UWF	URF	Deep	IUSF	CFud			
March-MOM [7]	March	Y	Y	P	N	N	P	N	Y	Y	N	N	36%	5N, 4N	-
March-1T1R [9]	March	Y	Y	Y	N	N	Y	N	N	P	N	N	36%	5N^{+}, 4N	-
March C* [8]	March	Y	Y	N	Y	Y	Y	N	N	N	N	N	45%	4N, 6N	-
March C*-1T1R [10]	March	Y	Y	N	Y	Y	Y	N	N	Y	N	N	55%	6N, 6N	-
March-CMOL [12]	March	Y	Y	N	Y	Y	Y	N	N	Y	N	N	55%	-	-
March W-1T1R [11]	March	Y	Y	Y	Y	Y	Y	N	N	Y	N	N	64%	9N, 8N	-
Parallel March [15]	DFT	Y	Y	N	Y	Y	Y	N	N	Y	N	N	55%	4(N+1), 5N+N$_r$	-
Sneak-path [7]	DFT	Y	Y	P	N	N	P	N	Y	Y	N	N	36%	7N, 5N/3	28+26N$_r$
Weak-write [18]	DFT	N	N	N	Y	N	N	Y	Y	Y	N	N	36%	No March	24+18N$_r$
Fast write [20]	DFT	Y	Y	N	N	N	N	Y	Y	Y	N	N	45%	(4T+1+x)N, 6N	50+18N$_r$
On-chip sensor [13]	DFT	Y	Y	Y	Y	Y	N	Y	Y	Y	N	N	73%	No March	20N
Enhanced March [22]	DFT	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	Y	91%	8N, 6N	13(N$_c$+N$_r$+4)
Proposed (DFT-HR-ET-NOR^N)	**DFT**	Y	Y	Y	Y	Y	Y	Y	Y	Y	P	Y	**91+%**	**6N, 8log$_2$N+8**	**2log$_2$N$_r$+96+4N$_c$**

TABLE IV: Comparison of RRAM test solutions. Yes (Y), No (N), Partial (P). N_r and N_c are number of rows and columns of the memory, respectively.

4) Advanced CMOS Technology: With technology scaling, a higher RR can be expected [37]. In Eq. 1, a lower R$_N$ implies higher RR, *i.e.* RR value approaches the ideal value of $\frac{R_{OFF}}{R_{ON}}$. Advanced technology nodes such as FinFET can allow lower variations which improves the overall accuracy of the DFT scheme [38]. In addition, with voltage down-scaling, altering the V$_{WL}$ can have greater impact on the conductance of the pass transistor (V$_{GS}$ is closer to threshold voltage). SRAM designs can also benefit from our proposed scheme, potentially by comparing bitline pair individually (alternatively, one BL or NBL per cycle) with the reference line. In short, this can facilitate a higher impact of our DFT schemes.

IX. Conclusions

In this paper, we present cost-efficient CIM-based DFT schemes that improve the cost of testing and the fault coverage (FC). Our schemes deploy multi-operand NOR logic operations that involve multi-row select read operations to accelerate the testing of ETD faults. In addition, the reconfigurability of the DFT implementation aids the detection of unique RRAM faults that are HTD. The design and implementation of a fast search algorithm facilitates the diagnosis of faults. Our proposed DFT implementations are validated on a post-layout extracted platform and testing sequences are introduced incorporating the proposed DFTs. Results show that more than 2.3× speedup, 6× area reduction, and better FC are achieved compared to the state-of-the-art.

References

[1] S. Yu *et al.*, “Emerging memory technologies: Recent trends and prospects,” *Solid-State Circuits Magazine*, 2016.

[2] W.-H. Chen *et al.*, “CMOS-integrated memristive non-volatile computing-in-memory for AI edge processors,” *Nat. Elect.*, 2019.

[3] F. Cai *et al.*, “A fully integrated reprogrammable memristor–CMOS system for efficient multiply–accumulate operations,” *Nat. Elect.*, 2019.

[4] M. Fieback *et al.*, “Intermittent Undefined State Fault in RRAMs,” in *ETS*, 2021.

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

[6] X. Cui *et al.*, “Design and test of the in-array build-in self-test scheme for the embedded RRAM array,” *TED*, 2019.

[7] S. Kannan *et al.*, “Sneak-path testing of crossbar-based nonvolatile random access memories,” *IEEE Trans. on Nanotechnology*, 2013.

[8] C.-Y. Chen *et al.*, “RRAM defect modeling and failure analysis based on march test and a novel squeeze-search scheme,” *TC*, 2014.

[9] Y.-X. Chen *et al.*, “Fault modeling and testing of 1T1R memristor memories,” in *VTS*, 2015.

[10] P. Liu *et al.*, “Efficient March test algorithm for 1T1R cross-bar with complete fault coverage,” *Elect. Letters*, 2016.

[11] Y. Luo *et al.*, “A high fault coverage march test for 1T1R memristor array,” in *EDSSC*, 2017.

[12] P. Liu *et al.*, “Defect analysis and parallel march test algorithm for 3D hybrid CMOS-memristor memory,” in *ATS*, 2018.

[13] T. Copetti *et al.*, “Validating a DFT Strategy’s Detection Capability regarding Emerging Faults in RRAMs,” in *VLSI-SoC*, 2021.

[14] V. A. Hongal *et al.*, “A novel “divide and conquer” testing technique for memristor based lookup table,” in *MWSCAS*, 2011.

[15] P. Liu *et al.*, “Logic operation-based DFT method and 1R memristive crossbar march-like test algorithm,” *IEICE*, 2015.

[16] P. Liu *et al.*, “Logic operation-based Design for Testability method and parallel test algorithm for 1T1R crossbar,” *Elec. Letters*, 2017.

[17] T. Li *et al.*, “Sneak-path based test and diagnosis for 1R RRAM crossbar using voltage bias technique,” in *DAC*, 2017.

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

[19] S. Hamdioui *et al.*, “Testing open defects in memristor-based memories,” *IEEE Trans. on Computers*, 2013.

[20] S. N. Mozaffari *et al.*, “More efficient testing of metal-oxide memristor–based memory,” *TCAD*, 2016.

[21] M. N. I. Khan *et al.*, “Test challenges and solutions for emerging non-volatile memories,” in *VTS*, 2018.

[22] P. Liu *et al.*, “Fault modeling and efficient testing of memristor-based memory,” *TCASI*, 2021.

[23] H.-S. P. Wong *et al.*, “Metal–oxide RRAM,” *Proceedings IEEE*, 2012.

[24] J. Yang *et al.*, “24.2 A 14nm-FinFET 1Mb Embedded 1T1R RRAM with a 0.022 μ m 2 Cell Size Using Self-Adaptive Delayed Termination and Multi-Cell Reference,” in *ISSCC*, 2021.

[25] E. Esmanhotto *et al.*, “High-density 3D monolithically integrated multiple 1T1R multi-level-cell for neural networks,” in *IEDM*, 2020.

[26] M. Fieback *et al.*, “Defects, Fault Modeling, and Test Development Framework for RRAMs,” *JETC*, 2022.

[27] N. Z. Haron *et al.*, “On defect oriented testing for hybrid CMOS/memristor memory,” in *Asian Test Symposium*, 2011.

[28] M. Fieback *et al.*, “Device-aware test: A new test approach towards DPPB level,” in *ITC*, 2019.

[29] S. Hamdioui *et al.*, “Memristor based computation-in-memory architecture for data-intensive applications,” in *DATE*, 2015.

[30] A. Singh *et al.*, “Referencing-in-array scheme for RRAM-based CIM architecture,” in *DATE*, 2022.

[31] J. Chang *et al.*, “A 20nm 112Mb SRAM in High-K metal-gate with assist circuitry for low-leakage and low-V MIN applications,” in *ISSCC*, 2013.

[32] W. Kim *et al.*, “Multistate memristive tantalum oxide devices for ternary arithmetic,” *Scientific reports, 2016*,

[33] M. R. Stan, “Synchronous up/down counter with clock period independent of counter size,” in *ARITH*, 1997.

[34] R. Aitken *et al.*, “Redundancy, repair, and test features of a 90nm embedded SRAM generator,” in *DFT*, 2003.

[35] F. Cüppers *et al.*, “Exploiting the switching dynamics of HfO2-based ReRAM devices for reliable analog memristive behavior,” *APL materials*, 2019.

[36] A. J. Van De Goor, “Using march tests to test SRAMs,” *DTC*, 1993.

[37] A. Razavieh *et al.*, “Challenges and limitations of CMOS scaling for FinFET and beyond architectures,” *Trans. on Nanotechnology*, 2019.

[38] E. J. Nowak *et al.*, “Turning silicon on its edge [double gate CMOS/FinFET technology],” *Circuits and Devices Magazine*, 2004.