Logic operation-based DFT method and 1R memristive crossbar March-like test algorithm

Peng Liu¹, Zhiqiang You¹, Jishun Kuang^{1a)},
Zhipeng Hu¹, and Weizheng Wang²

¹ College of Computer Science and Electronic Engineering, Hunan University,
China

² College of Computer and Communication Engineering,
Changsha University of Science and Technology, China

a) jshkuang@hotmail.com

Abstract: As an attractive option of future non-volatile memories (NVM), resistive random access memory (RRAM) has attracted more attentions. Due to its high density and low power, one memristor (1R) crossbar is a dominant RRAM structure. In this paper, we propose a logic operation-based design for testability (DFT) architecture for 1R crossbar testing. In this architecture, memristor-aided logic (MAGIC) NOR gates are embedded to check whether all the cells in the crossbar are 0 s or not at a time. A March-like test algorithm is also presented for the proposed architecture, which covers all modeled faults. The test time is reduced drastically with a little area overhead.

Keywords: memristive crossbar, RRAM, DFT, March algorithm, logic operation

Classification: Integrated circuits

References

- [1] D. B. Strukov, G. S. Snider, D. R. Stewart and S. R. Williams: *Nature* **453** (2008) 80. DOI:10.1038/nature06932
- [2] S. Kannan, J. Rajendran, R. Karri and O. Sinaoglu: *IEEE Trans. Nanotechnol.* **12** (2013) 413. DOI:10.1109/TNANO.2013.2253329
- [3] C. Y. Chen, H. C. Shih, C. W. Wu, C. H. Lin, P. F. Chiu, S. S. Sheu and F. T. Chen: *IEEE Trans. Comput.* **64** (2015) 180. DOI:10.1109/TC.2014.12
- [4] O. Ginez, J. M. Portal and C. Muller: *IEEE Eur. Test Symposium* (2009) 61. DOI:10.1109/ETS.2009.23
- [5] Y. X. Chen and J. F. Li: *IEEE VLSI Test Symposium* (2015) 1.
- [6] B. Liu, Z. You, X. Li, J. Kuang and Z. Qin: *IEICE Electron. Express* **10** (2013) 20130369. DOI:10.1587/elex.10.20130369
- [7] S. Kvatinsky, D. Belousov, S. Liman, G. Satat, N. Wald, E. G. Friedman, A. Kolodny and U. C. Weiser: *IEEE Trans. Circuits Syst. II, Exp. Briefs* **61** (2014) 895. DOI:10.1109/TCSII.2014.2357292
- [8] M. Bushnell and V. Agrawal: *Essentials of Electronic Testing for Digital, Memory, and Mixed-Signal VLSI Circuits* (Springer, 2000).

- [9] C. Yakopcic: Ph. D thesis University of Dayton, Dayton (2014).
[10] N. Z. Haron and S. Hamdioui: Design, Automatic Test Europe (2012) 799.

1 Introduction

One memristor (1R) crossbar [1], which is a dominant RRAM structure, is a promising candidate in next-generation high-performance, high-density, emerging non-volatile storage architecture. However, due to its immature manufacture process, memristive crossbar is prone to defects and requires efficient test methods for precise fault models with short test time. A number of test methods and fault models are proposed to test memristive crossbar [2, 3, 4, 5] till now. The modeled faults can be classified as *single-cell* faults and *coupling* faults [2]. A single-cell fault is defined as the fault behavior of the cell only affecting itself, including *Stuck-at 0*, *Stuck-at 1*, *Slow Write 0*, *Slow Write 1*, *Deep-0*, *Deep-1*, *Deep-1/0* [2] and *Disturb Read* [3]. A coupling fault is defined as a transition in a cell (called *aggressor* cell) causing an unexpected change in another cell (called *victim* cell). In 1R crossbar, a coupling fault only occurs for the cells on adjacent wires in the same layer [4]. Kannan et al. modeled the defects in a memristor caused by parametric variations during its fabrication [2]. Then, a testing scheme uses sneak paths inherent in crossbar memories, to test multiple memory elements at the same time, thereby reducing testing time.

Besides RRAM, memristor can also be used for logic operations, including stateful logic [6] and memristor-aided logic (MAGIC) [7].

Fig. 1. (a) Memristor symbol. (b) Schematic of an N -input MAGIC NOR logic [7]. $In_1, In_2 \dots In_N$ are input memristors, Out is a output memristor.

In Fig. 1(a), the resistance of a memristor is changed by the electrical current. When a current flows into the device, the resistance increases. When a current flows out of the device, the resistance decreases. In MAGIC NOR function design, all input memristors are connect each other with the same direction in parallel. The output memristor connects to all input memristors with the opposite direction in series as described in Fig. 1(b).

There are two steps for MAGIC NOR function. The first step initializes the output memristor to low resistance R_{on} . R_{on} and high resistance R_{off} represent logic 1 and 0 in this brief respectively. In the second step, a voltage V_0 is applied to all input

memristors. The voltage across the output memristor depends upon the resistance of the input and output memristors. When the resistances of all input memristors are R_{off} , if the voltage across the output memristor is less than its reset voltage V_{off} , the resistance of output memristor is still in its initial state. If at least one input memristors' resistance is R_{on} , the resistance of output memristor is changed to R_{off} . To meet these situations, V_0 should satisfy the following expression.

$$\frac{V_{off}}{R_{on}} \cdot \left[R_{on} + \left(\frac{R_{off}}{N-1} \right) \parallel R_{on} \right] < V_0 < V_{off} \cdot \left(1 + \frac{R_{off}}{NR_{on}} \right) \quad (1)$$

To keep the resistance of the input memristors, the voltage across the input memristors should be less than their set voltage V_{on} .

$$V_0 < \left(1 + \frac{NR_{on}}{R_{off}} \right) \cdot |V_{on}| \quad (2)$$

Therefore, V_0 should satisfy the following expression to derive the correct logic of MAGIC NOR with keeping the states of input memristors.

$$\frac{V_{off}}{R_{on}} \cdot \left[R_{on} + \left(\frac{R_{off}}{N-1} \right) \parallel R_{on} \right] < V_0 < \min \left[V_{off} \cdot \left(1 + \frac{R_{off}}{NR_{on}} \right), \left(1 + \frac{NR_{on}}{R_{off}} \right) \cdot |V_{on}| \right] \quad (3)$$

In this brief, we propose an efficient logic operation-based DFT architecture for 1R crossbar. Furthermore, a Match-like test algorithm is described to support the proposed DFT architecture.

2 DFT architecture with MAGIC NOR logic

MAGIC NOR operation can be used to check whether all the cells in a crossbar are 0s or not at a time. If there exists one cell with logic 1, the output memristor turns logic 0, which is different to the case with all 0s. We can use this logic operation to replace read 0 operation. The test time is reduced drastically.

Fig. 2 shows the proposed DFT architecture with MAGIC NOR logic for 1R crossbar. There is a memristor on each crosspoint. For MAGIC NOR operation does not change the states of input memristors, and can be placed within memristor crossbar array, we only need to add a row memristors as the output memristors. In our DFT architecture, we add the memristors on the last row as the outputs of MAGIC NOR logic function. *NOR Read/Set Select (NRSS)*, *NOR Write Select (NWS)* and *NOR Read Selects (NRSs)* signals are added to control the realization of MAGIC NOR logic function. There are five operations on test mode, including *Write*, *Read*, *NOR Logic Set*, *NOR Logic Write* and *NOR Logic Read* operations.

During *Write* operation, the *Write Select Rows (WSRs)* and the *Write Select Columns (WSCs)* are valid in the rows and the columns of the target memristors. The cells in the same rows or columns can be written by the same logic value simultaneously. For *Write 1* operation, a voltage $-V_{Cwrite}$ applies to the selected column, while a voltage of $V_{on} - V_{Cwrite}$ applies to a selected rows, where V_{Cwrite} is a positive constant. Consequently, V_{on} applies to target memristors whose states are changed to logic 1. For *Write 0* operation, a voltage $V_{off} - V_{Cwrite}$ applies to the selected column, while a voltage of $-V_{Cwrite}$ applies to a selected rows. Consequently, V_{off} applies to target memristors whose states are changed to logic 0.

Fig. 2. The proposed DFT architecture with MAGIC NOR logic

During *Read* operation, all *Read Selects* (*RSs*) are valid. The read voltage (V_{read}) applies to the column, where the *WSC* is valid. Then, the output current is considered to be the current across R_s on all rows. The states of the selected cells are read out.

There is NOR logic function on each column. The memristors on rows, where $WSRs$ are valid, are the inputs of NOR gate. The memristor on the $n + 1$ row is the output of NOR gate. All or part of NOR logic functions can be performed simultaneously. There are three operations of NOR logic function to realize traditional *Read* operation.

NOR Logic Set operation: The *NRSS* and *WSCs* of the columns, where the memristors need to be initialized, are valid. Other control signals are invalid. The voltage of $V_{on} - V_{Cwrite}$ and $-V_{Cwrite}$ is applied to the last row and the columns, respectively. Consequently V_{on} is applied to target memristors. The states are changed to logic 1.

NOR Logic Write operation: Before this operation, NOR logic set operation usually needs to be performed. After NOR logic set operation, the *WSRs* in the rows of target memristors and *NWS* are valid. Other control signals are invalid. A voltage V_{NW} applies to the rows, where the *WSRs* are valid. Here, V_{NW} satisfies Expression 3. The results of NOR logic in each column are stored to the added memristors $c_{n+1,x}$. The cell $c_{n+1,x}$ keeps its state 1, when all cells are logic 0 in the valid rows and column x . Otherwise, the cell $c_{n+1,x}$ is changed to logic 0. In this operation, all added memristors can be written in parallel.

NOR Logic Read operation: The $NRSS$ and NRS s of each column are valid, other control signals are invalid. A read voltage V_{read} is applied to the last

row. Then, the output current is considered to be the current across R_s on all columns.

3 March-like test algorithm

To detect coupling faults of traditional RAM, March test algorithm applies a sequence of operations to a given memory cell before proceeding to the next cell. Therefore, all the cells should be tested several times one by one [8]. Compared with traditional RAM, 1R crossbar has simpler structure. The coupling faults only occur for the cells on adjacent wires in the same layer. Therefore, an efficient March test algorithm needs to be proposed.

A March-like test algorithm for our DFT architecture is proposed to cover all the modeled faults in a 1R crossbar shown as follows.

$$\left\{ \begin{array}{l} M1 : W_{oo}0; M2 : R_{xx}; M3 : W_{xx}0; M4 : NS_{xx}; M5 : NW_{xx}; \\ M6 : NR_{xx}; M7 : W_{xx}0; M8 : W_{ee}1; M9 : R_{xe}; M10 : NS_{xo}; \\ M11 : NW_{xo}; M12 : NR_{xo}; M13 : W_{xx}1; M14 : R_{xx}; M15 : R_{xx}; \\ M16 : W_{xx}1; M17 : W_{xx}0; M18 : NS_{xx}; M19 : NW_{xx}; M20 : NR_{xx}; \end{array} \right\} \quad (4)$$

where R , W , NS , NW and NR denote *Read*, *Write*, *NOR Logic Set*, *NOR Logic Write* and *NOR Logic Read* operations. The subscripts o , e , x denote *odd*, *even*, both *odd* and *even* of rows or columns respectively. R_{xx} and W_{xx} represent *Read* and *Write* all cells in a crossbar. NS_{xx} , NW_{xx} and NR_{xx} represent to set all the added memristors, to perform NOR operation, and to read out the states of the added memristors. We can see that NS_{xx} , NW_{xx} and NR_{xx} is equivalent to *Read 0* on all cells in a crossbar. In $N \times N$ memristive crossbar, instead of N read cycles, it only needs 1 read and 2 write cycles using our DFT architecture. This March-like test algorithm consists of 20 elements, which are separated by semicolons. Subsequently, we explain which elements of the March-like test algorithm can detect the modeled faults as follows.

- 1) *Stuck-at 1 (SA1)*: These faults can be sensitized by $W0$ and detected by $R0$. The faults of the cells located in the odd rows and the odd columns are sensitized by element $M1$ and detected by element $M2$. The faults of other cells in the crossbar are sensitized and detected by elements $M3$ and $M6$ respectively.
- 2) *Stuck-at 0 (SA0)*: These faults can be sensitized by $W1$ and detected by $R1$. The faults of the cells located in the even rows and the even columns are sensitized by element $M8$ and detected by element $M9$. The faults of other cells in the crossbar are sensitized and detected by elements $M13$ and $M14$ respectively.
- 3) *Slow Write 0 (SW0)*: These faults can be initialized by $W1$, sensitized by $W0$, and detected by $R0$. Therefore, the element $M16$ initializes the fault of all cells in the crossbar, and the element $M17$ sensitizes the fault, then the element $M20$ detects them.
- 4) *Slow Write 1 (SW1)*: These faults can be initialized by $W0$, sensitized by $W1$, and detected by $R1$. Therefore, the element $M7$ initializes the fault of all cells in the crossbar, and the elements $M8$ and $M13$ sensitize the faults of the cells

located in even rows and even columns, and other cells in the crossbar respectively. Then the elements $M9$ and $M14$ detect them.

- 5) *Deep-0*: These faults can be initialized by two $W0$, be sensitized by $W1$ and detected by $R1$. The element $M3$ initializes the faults of all cells in the crossbar as the first $W0$. The element $M7$ initializes the faults of all cells in the crossbar as the second $W0$. The elements $M8$ and $M13$ sensitize the faults of the cells located in even rows and even columns, and other cells in the crossbar respectively. Then, the faults can be detected by elements $M9$ and $M14$ in our March-like test algorithm.
- 6) *Deep-1*: These faults can be initialized by two $W1$, be sensitized by $W0$ and detected by $R0$. The element $M13$ initializes the faults of all cells in the crossbar as the first $W1$. The element $M16$ initializes the faults of all cells in the crossbar as the second $W1$. The element $M17$ sensitizes the faults of all cells in crossbar. Then, the faults can be detected by element $M20$ in our March-like test algorithm.
- 7) *Deep-1/0*: These faults have the characteristics of both *Deep-1* and *Deep-0* faults. Therefore, the test method of either *Deep-1* or *Deep-0* is also effective in detecting the faults.
- 8) *Disturb Read (DR)*: In our DFT, *Read* and *Write 0* operations have the same bias voltage direction. During *Read* operation, the selected cell is flipped if the original data is logic 1, which appears to be *Disturb Read-1* fault. For this fault, it can be sensitized by $R1$ and detected by another succession of $R1$. In our March-like test algorithm, the element $M14$ sensitizes the faults of all cells in crossbar and the element $M15$ detects them.
- 9) *Coupling Fault*: The coupling faults only occurs for the cells on adjacent wires in the same layer. Therefore, all coupling faults can be sensitized by $W0$ on odd rows and odd columns and detected by $R1$ of all cells except the written cells or $W1$ on even rows and even columns and detected by $R0$ of all cells except the written cells. In our March-like test algorithm, the faults can be sensitized by $M1$ or $M8$ and detected by $M2$ or $M9$, $M12$.

Table I summarizes the faults in a 1R crossbar detected by our March-like test algorithm, and the elements that initialize, sensitize and detect the faults.

Table I. Detect faults in our March-like test algorithm

Fault Type	Elements for Initialization and Sensitization	Elements for Detection
<i>SA1</i>	$M1, M3$	$M2, M6$
<i>SA0</i>	$M8, M13$	$M9, M14$
<i>SW0</i>	$M16, M17$	$M20$
<i>SW1</i>	$M7, M8, M13$	$M9, M14$
<i>Deep-0</i>	$M3, M7, M8, M13$	$M9, M14$
<i>Deep-1</i>	$M13, M16, M17$	$M20$
<i>RD</i>	$M14$	$M15$
<i>Coupling</i>	$M1$ or $M8$	$M2$ or $M9, M12$

4 Experimental results

To explaining March-like test algorithm efficiently, an 8×8 1R crossbar structure with our DFT architecture is built. The memristor model reported in [9] is used in our Spice simulation. The parameters of the 1R crossbar are $R_{on} = 125 \text{ k}\Omega$, $R_{off} = 41 \text{ G}\Omega$, $V_{on} = 7.88 \text{ V}$, $V_{off} = 3.798 \text{ V}$, $V_{read} = 1 \text{ V}$, $T_{write} = 10 \text{ ns}$, $T_{read} = 4 \text{ ns}$, $V_{NW} = 6 \text{ V}$, $T_{NW} = 10 \text{ ns}$, $T_{NR} = 4 \text{ ns}$, $T_{NS} = 10 \text{ ns}$, $V_{Cwrite} = 2.26 \text{ V}$.

Fig. 3(a)–(g) show the situations with SA-1, SA-0, SW0, SW1, Deep-0, Deep-1 and DR fault in c_{11} , which can be detected by M_2 , M_{14} , M_{20} , M_{14} , M_{14} , M_{20} and M_{15} , respectively. The coupling faults in c_{11} are detected by M_2 . Fig. 3(h), (i) show short cases between the first column and the second column, and between the first row and the second row. From Fig. 3, we can see all modeled faults can be detected by March-like test algorithm.

Fig. 3. March-like test algorithm can detect (a) *stuck-at 1*, (b) *stuck-at 0*, (c) *Slow Write 0* and (d) *Slow Write 1* (e) *Deep-0*, (f) *Deep-1*, (g) *Disturb Read* faults, *Coupling Fault* caused by shorts between (h) columns and (i) rows. X-axis shows the time slots in test procedure, Y-axis shows the current. The red curves are for the faulty cell cases, while the yellow dotted curves are for the fault-free cell cases.

For $N \times N$ memristor crossbar, the test time of our March-like test algorithm is $7/2N$ read, 7 write and 3 NOR cycles, due to R_{xe} needs $N/2$ read cycles. In our experiment, NOR logic *Write* time and NOR logic *Set* time are the same as *Write* time, and NOR logic *Read* time is the same as *Read* time. Therefore, the total test time is $(7/2N + 3) * Read + 13 * Write$.

As is shown in Table II, our method reduced one order of test time compared with sneak path testing [2] and short write testing [10] methods. The area overhead in our method is N memristors, $2N$ transistors and N resistors for NOR logic read module, 2 transistors for NRSS and NWS signals. If the read module is reused for different blocks, the area overhead is $N * Memristor + 2(N + 1) * Transistor + N * Resistor$.

Table II. Comparison results of test time

	Test Time
Sneak Path Testing [2]	$4/5N^2 * Read + 3N^2 * Write$
Short Write Time [10]	$4N^2 * Read + 7N^2 * Write$
Our Methods	$(7/2N + 3) * Read + 13 * Write$

5 Conclusions

In this brief, a DFT architecture with MAGIC NOR logic and a March-like test algorithm for this architecture have been proposed. The faults of memristive crossbar are simulated by Spice simulation. The March-like test algorithm requires $(7/2N + 3)$ *Read* and 13 *Write* cycles, and a little area overhead for $N \times N$ memristor crossbar. In the future, we will consider to further reduce test time by adding some modules for *Read* 1 operation.

Acknowledgements

This brief was supported by National Natural Science Foundation of China under Grant No. 61472123, 61303042.