Scientific Research Publishing

On the Production Testing of Memristor Ratioed Logic (MRL) Gates

Ahmed Shukry Emara[1], Ahmed Hassan Madian[2], Hassanein Hamed Amer[1], Sherif Hassanein Amer[1], Mohamed Bakr Abdelhalim[3]

[1]Electronics and Communications Engineering, The American University in Cairo, Cairo, Egypt
[2]Radiation Energy Department, Egyptian Atomic Energy Authority, Cairo, Egypt
[3]College of Computing and Information Technology AASTMT, Cairo, Egypt
Email: aemara93@aucegypt.edu, ah_madian@hotmail.com, hamer@aucegypt.edu,
sherif_amer@aucegypt.edu, mbakr@ieee.org

Abstract

This paper focuses on the production testing of Memristor Ratioed Logic (MRL) gates. MRL is a family that uses memristors along with CMOS inverters to design logic gates. Two-input NAND and NOR gates are investigated using the stuck at fault model for the memristors and the five-fault model for the transistors. Test escapes may take place while testing faults in the memristors. Therefore, two solutions are proposed to obtain full coverage for the MRL NAND and NOR gates. The first is to apply scaled input voltages and the second is to change the switching threshold of the CMOS inverter. In addition, it is shown that test speed and order should be taken into consideration. It is proven that three ordered test vectors are needed for full coverage in MRL NAND and NOR gates, which is different from the order required to obtain 100% coverage in the conventional NAND and NOR CMOS designs.

Keywords

1. Introduction

Over the past decades, semiconductor technology has provided enormous enhancements in systems characteristics such as power consumption, speed, reliability and production cost. Such improvements came into practice mainly due to the continuous miniaturization of device dimensions in the fabrication process [1]. This incessant down scaling of devices leveraged the integration of more circuitry on a single chip producing complex hard-

ware systems. However, this down scaling cannot take place forever. There are many factors that limit the down scaling of transistors such as the minimum dimensions that can be fabricated and increase in the off-state power consumption due to high leakage currents [1] [2]. Hence, innovations are required to allow for the continued growth in the complexity of hardware systems. One of these innovations is the memristors and memristive devices [1].

Memristors existences were theoretically predicted in 1971 by Chua [3]. In 2008, Hewlett Packard (HP) physically realized the memristor [4]. A memristor is a resistive switch that produces either a high resistance or a low resistance depending on the polarity of the applied voltage, *i.e.*, the direction of current flow [1] [4]. **Figure 1** shows the symbol and polarity of the memristor.

Memristors are mainly used in memories. In memories, memristors are used to represent logic states, *i.e.*, the resistance of a memristor is used to represent logic 0 or logic 1. Memristors are not only used in memory systems. They are also used to design neuromorphic systems, analog circuits and digital logic circuits [5]. For instance, in [1], memristors are used to design digital logic circuits. In [1], a logic family is described, namely, Memristor Ratioed Logic (MRL). MRL uses memristors that are integrated with CMOS inverters to perform Boolean NAND and NOR functions. One of the main advantages of using memristors in evaluating logic is that it saves physical area and therefore increases logic density.

As of any other device, memristors are prone to defects. Numerous research efforts took place in testing memristor-based memory systems. In [6] [7], different fault models were proposed. [8], proposed two DfT schemes for testing memristors using these fault models and the conventional March test was used, in which a fixed pattern of reads and writes are applied to each memory cell to detect faults in that cell. This method (tests one cell at a time) is time consuming for large memories. Therefore, testing multiple transistors at the same time was needed. This was done by using divide-and-conquer testing technique proposed in [9]. However, this technique does not consider sneak-paths (unwanted current flow) in crossbar memories. In [10], a sneak-path testing scheme was proposed to test multiple memristors simultaneously using sneak-path currents. In [11], a new design was proposed to overcome the issue of sneak path currents in memristor crossbar memories. The design is comprised of one access transistor and one memristor (1T1R). Fault models are proposed in [11], based on electrical defects. A March Test is proposed to cover all the defined faults.

All the aforementioned research efforts focus on testing memory systems designed using memristors. This work is concerned about testing digital logic gates implemented using memristors.

To the best of the authors' knowledge, the production testing of MRL has not been tackled yet in the literature. In this paper, catastrophic faults in memristors and transistors are investigated in the context of production testing. The minimum test set is obtained and then compared to that of the conventional stuck-at fault model. It will be shown that while testing MRL NAND/NOR gates, some test escapes might take place. Hence, two solutions are proposed to face this issue. The first is applying scaled input voltages and the second is to alter the switching threshold of the inverter. Additionally, it will be shown that test sequence and speed should be taken into consideration.

The rest of the paper is organized as follows. Section 2 provides a description of MRL logic family. Section 3 is devoted to the testing of the MRL gates that includes faults in the memristors and transistors. Conclusions are mentioned in Section 4.

2. Memristor Ratioed Logic (MRL) Family

In this section, explanation of the MRL logic family is provided. In [1], MRL is used to design two-input NAND and NOR Boolean functions. The memristors are used to perform the AND and OR functionalities, while a standard CMOS inverter is used to obtain their complements. In [1], the TEAM (ThrEshold Adaptive Memristor) model was used.

Figure 1. Memristor symbol. Thick black line on the left represents the polarity of the device. If current flows into the device, resistance of the memristor decreases and vice versa.

Two-input AND and OR logic gates consists of two memristors connected in series at opposite polarities as shown in **Figure 2(a)** and **Figure 2(b)**, respectively. The memristors are used as computational elements to evaluate logic. On one end of the memristors terminals the inputs A and B are applied, while the common node of the memristors is the output node labeled $V_{out,AND}$ and $V_{out,OR}$. The CMOS inverter is added then for reasons mentioned later in this section.

In the AND logic gate, when the current flows into the memristors, the resistance of the memristors increases and reaches R_{off} eventually. On the other hand, if the current flows out of the memristors, the resistance of the memristors decreases and reaches R_{on} eventually. The OR logic gate has the exact opposite behavior of the AND gate as opposite polarity is used.

In the following explanation, 0 V is used to represent logic "0" and 1V is used to represent logic "1". AND and OR logic gates behave similarly when identical inputs are applied, *i.e.*, AB = 00 or AB = 11. When these inputs are applied there is no current flow through the memristors. Hence, there is no voltage drop between the inputs. Therefore, the output voltage $V_{out,AND}$ and $V_{out,OR}$ are similar to the input voltage. In the case where the inputs are different, *i.e.*, AB = 01 or 10, current flows from the higher input voltage terminal to the lower. This changes the resistance of the two memristive devices.

In the AND logic gate, consider the input vector AB = 10. For this case, the current flows out of the memristor labeled R1 in **Figure 2(a)**. R1 reaches R_{off} by the end of the computational process. Simultaneously, the current flows into the memristor labeled R2 in **Figure 2(a)** and R2 reaches R_{on} towards the end of the logic evaluation. The output voltage $V_{out,AND}$ is a voltage divider between the two memristors, and is therefore

$$V_{out,AND} = \frac{R_{on}}{R_{off} + R_{on}} \times 1\,\text{V} \approx 0\,\text{V} \tag{1}$$

Consider the same scenario AB = 10 for the OR logic gate, where opposite polarity is used. Therefore, the resistance of the memristors behave in the exact opposite way of the AND logic gate and the output voltage $V_{out,OR}$ is therefore

$$V_{out,OR} = \frac{R_{off}}{R_{off} + R_{on}} \times 1\,\text{V} \approx 1\,\text{V} \tag{2}$$

It should be noted that the initial resistance of both memristors does not affect the functionality. However, it affects the delay of computation when both inputs are different [1]. A standard CMOS inverter is added for two main reasons. First, since the AND and OR functions are non-inverting, a complete logic structure is achieved by connecting the output node to a CMOS inverter. In addition, memristive devices lack signal restoration, *i.e.*, the output voltage levels will degrade if these logic gates are cascaded for several levels [1].

3. Production Testing of NAND and NOR

This section investigates production testing for catastrophic faults in the MRL NAND and NOR logic gates shown in **Figure 2(a)** and **Figure 2(b)**. MRL uses memristors and transistors to build logic gates. Therefore, faults that occur in both memristors and transistors are considered.
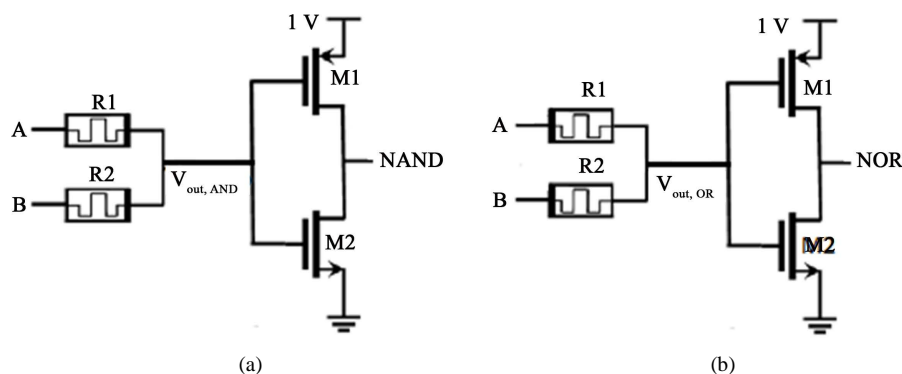
Figure 2. Schematic of (a) two-input MRL NAND (b) two-input MRL NOR.

In this paper the TEAM model is used as this model was used by [1] in the proposed designs. The ELDO simulator from Mentor Graphics is used in this study and the technology is the 45 nm CMOS technology.

The memristor stuck at fault model proposed in [8] is used. This fault model assumes that the resistance of the memristor will remain stuck at either R_{on} or R_{off} irrespective of the applied voltage across its terminals. According to the TEAM model parameters, R_{on} is 100Ω and R_{off} is 200 KΩ. In addition, the transistor five-fault model proposed in [12] is used, as it is one of the most commonly used fault models. The five-fault model consists of five faults per transistor which are: Drain-Source short circuit (DS), Gate-Drain short circuit (GD), Gate-Source short circuit (GS), Open Drain (OD) and Open Source (OS). It was shown in [13] that open circuit faults can be modeled by inserting a 250 MΩ (or more) resistance in the 45 nm technology, while short circuit faults are modeled by inserting a 10 Ω resistance. Faults are injected one at a time as in [14]. For every fault, the circuit output is compared to the fault-free output. A fault is considered detected if the output is different from the fault-free case.

3.1. Memristor Faults

In this section, memristor faults are considered for both the NAND and NOR logic gates. The standard CMOS inverter used in **Figure 2(a)** and **Figure 2(b)** has a switching threshold voltage (V_M) of 0.5 V. $V_{out,AND}$ and $V_{out,OR}$ (input nodes of the inverter) may be affected by noise that is taken as 5% of the supply voltage (1 V), *i.e.*, 0.05 V as in [15]. Hence any input voltage to the inverter that falls between 0.45 V and 0.55 V is considered to be in the undefined region. For the NAND and NOR logic gates, it is observed that, due to faults in the memristors, the output voltage $V_{out,AND}$ and $V_{out,OR}$ falls in the undefined region for some input vectors. Therefore, it is considered here that these input vectors that produce an output in the undefined region, cannot be used as test vectors.

For the NAND logic gate, consider for example, the fault R1 stuck at R_{off}; it is clear from **Figure 3** that all test vectors produce the correct output except the test vector AB = 01. This input vector produces a 0.5 V at the $V_{out,AND}$ node, that falls in the undefined region. A similar argument exists for AB = 10 as the circuit is symmetric.

The explanation of this result is as follows. Applying the test vector "01" forces R1 to switch to R_{on} and R2 should switch to R_{off} by the end of the computation process. However, due to the fault, R1 does not switch to R_{on} and is stuck at R_{off}. Hence the output voltage $V_{out,AND}$ is therefore 0.5 V from (3).
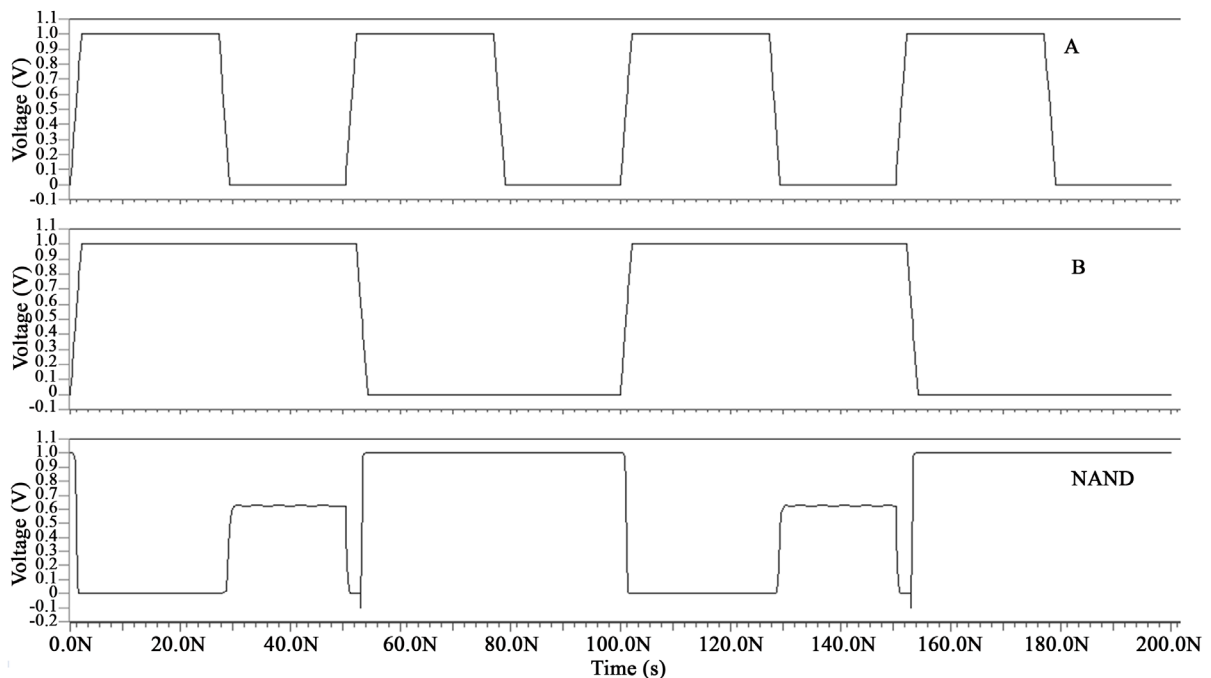
Figure 3. Test results for R1 stuck at R_{off} for the NAND gate.

$$V_{out,AND} = \frac{R_{off}}{R_{off} + R_{off}} \times 1 \text{ V} = 0.5 \text{ V} \qquad (3)$$

Likewise, the same issue of output voltages falling in the undefined region occurs in the NOR logic gate. Therefore, there are two different proposed solutions to face the aforementioned issue as shown in the coming subsections.

3.1.1. Scaled Input Voltages

The first proposed solution is applying scaled input voltages to the inputs of the logic gates to detect all memristor faults. For the NAND logic gate, 0.33 V is used to represent logic low "0" while keeping logic high "1" represented by 1 V. This forces the output voltage $V_{out,AND}$ to be 0.67 V (midpoint between 0.33 V and 1 V) for the same fault (R1 $\rightarrow$ R_{off}), which is interpreted by the CMOS inverter as logic high "1" (0.67 V is not in the undefined region); so the NAND output is logic low "0" and the fault is detected. **Table 1** shows the test results for the NAND gate where logic low "0" is 0.33 V and logic high "1" is 1 V. Note that in **Table 1**, "D" indicates a detected fault while "U" indicates undetected fault. Also R $\rightarrow$ R_{on} indicates that the resistance of the memristor is stuck at R_{on} and R $\rightarrow$ R_{off} indicates that the resistance of the memristor is stuck at R_{off}.

For the NOR gate, the solution is keeping 0 V to represent the logic low "0" while logic high "1" should be represented by 0.67 V. **Table 2** shows the test results for the MRL NOR gate

3.1.2. Changing the Switching Threshold of the Inverter

The second proposed solution is changing V_M of the inverters, by carefully sizing the PMOS and NMOS transistors. For the NAND logic gate, the inverter is designed to have V_M of 0.35 V. Consider the same fault R1 $\rightarrow$ R_{off} for the same input vector AB = 01; this fault causes the input of the inverter $V_{out,AND}$ to be 0.5 V which is interpreted by the inverter as logic high "1" so the NAND output is logic low "0" and the fault is detected. The test results are identical to those shown in **Table 1**. For the NOR logic gate, the inverter is designed to have V_M of 0.65 V. The test results are identical to those shown in **Table 2**. **Table 3** shows a summary of the V_M and the undefined region of the three different CMOS inverters.

Figure 4 shows the test results for the fault previously discussed (R1 stuck at R_{off}). It is clearly shown that the 0.5V value does not appear anymore and the fault is detected using the test vector 01.
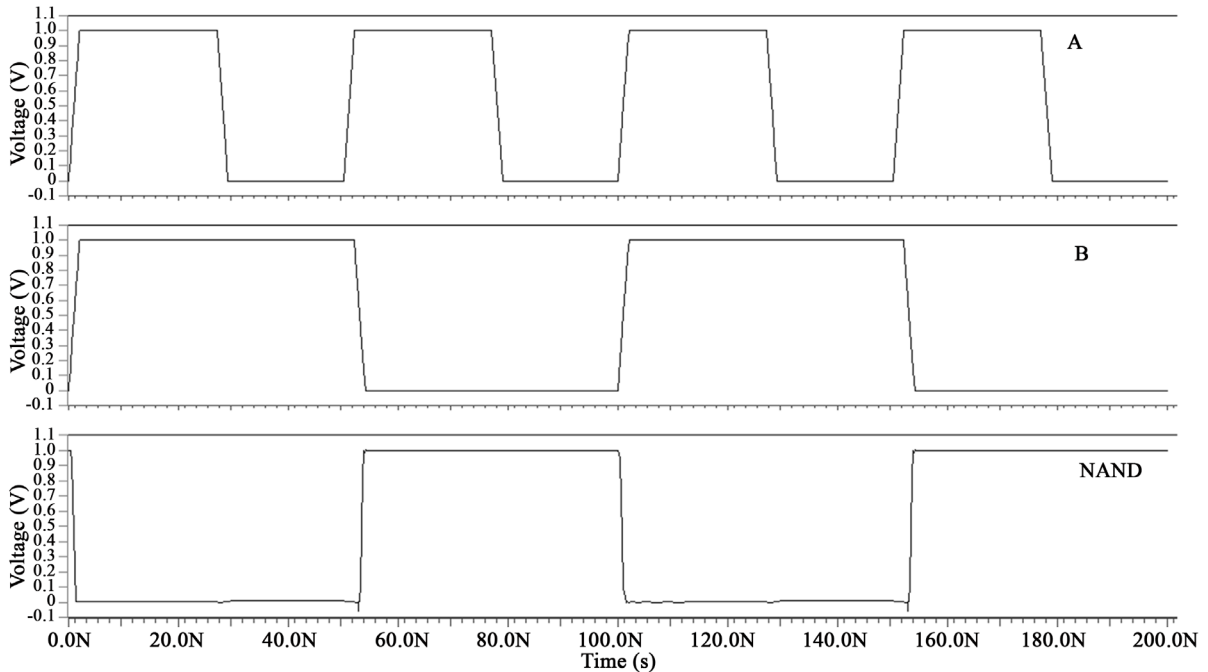
Figure 4. Test results for R1 stuck at R_{off} for the NAND gate using the two proposed solutions.

Table 1. MRL NAND test results.

Input Vector	Faults			
<AB>	R1 → R_{on}	R1 → R_{off}	R2 → R_{on}	R2 → R_{off}
00	U	U	U	U
01	U	**D**	**D**	U
10	**D**	U	U	**D**
11	U	U	U	U

Table 2. MRL NOR test results.

Input Vector	Faults			
<AB>	R1 → R_{on}	R1 → R_{off}	R2 → R_{on}	R2 → R_{off}
00	U	U	U	U
01	**D**	U	U	**D**
10	U	**D**	**D**	U
11	U	U	U	U

Table 3. Summary of VM and the undefined region of the three different inverters.

	Standard Inverter	Low V_M Inverter (NAND)	High V_M Inverter (NOR)
V_M	0.5 V	0.35 V	0.65 V
Undefined Region	0.45 V - 0.55 V	0.3 V - 0.4 V	0.6 V - 0.7 V

3.2. Resistive Open Faults

In this subsection, detection of resistive open faults is shown. It is observed that detecting resistive open faults depend on the speed of test vector application and the order of application of the test vectors. It was shown in [16] that detecting resistive open faults depends on the speed of test vector application. For the NAND logic gate in **Figure 2(a)**, consider, for example, the fault M1 OD, *i.e.*, resistive open in transistor M1. For the input vectors 01 or 10, an RC circuit is established between the supply voltage V_{dd} and the NAND output node. R is R_{M1}, which is the equivalent ON resistance of the PMOS transistor M1 in series with R_{op}, which is the injected fault. C represents the overhead capacitance. In this case the delay of the inverter can be estimated by (4) as in [16]:

$$Delay \cong \left[R_{M1} + R_{op} \right] \cdot C . \tag{4}$$

Therefore, if the test speed is very slow, *i.e.*, enough time is given for logic evaluation and the fault will not be detected. This applies for both NAND and NOR gates. Additionally, it was shown in [16] that testing resistive open faults in the CMOS NAND logic gate depends on the order of test vector application. It is concluded from [16], that although the minimum test set includes only three test vectors, namely 01, 10 and 11, four input vectors have to be applied. For example, a possible test sequence might be 11, 01, 11, 10.

Likewise, detecting resistive open faults in NAND and NOR MRL family depends on the order of test vectors application. The coming two subsections discuss the test sequence needed for full fault coverage in NAND and NOR MRL family.

3.2.1. Detection of Resistive Open Faults for the Scaled Input Voltages Proposal

It is mentioned earlier that input voltages are scaled to detect all memristor faults. It is found that detecting open faults depends on the order of test vector application. For instance, consider the fault M2 OD/OS for the NAND logic gate, *i.e.*, resistive open in transistor M2 in **Figure 2(a)**. This fault isolates the NAND output from the ground voltage. So if the input vector applied is 11 the output node will also not be connected to the supply because this turns transistor M1 off. The output then is floating and retains its previous logic state as in [16]. In order to detect this fault, an initializing vector activating the pull up PMOS transistor M1 must be applied which

is 01 or 10 in this case. Applying these test vectors 11 pulls up the output of the NAND gate to "1". After applying the initializing vector, the test vector 11 is applied. This keeps the output of the NAND gate in the floating state and will retain its previous logic state, which is "1" and hence the fault is detected. **Table 4** shows test results of detecting open faults in transistors M1 and M2 for the MRL NAND gate. Unlike the previous fault, open faults in M1 do not require a specific sequence to be detected and can be detected by either 01 or 10. M1 OD/OS merely cuts the path for the supply voltage and, accordingly, M1 is unable to pull up the output node to "1". When 01 or 10 is applied, bearing in mind that "0" is 0.33 V, 0.33 V (higher than the threshold of the transistor) is transmitted to the input of the inverter, switches M2 ON and the output node is pulled to ground. Hence, the fault is detected.

The same explanation could be given for resistive opens in the NOR MRL logic gates. However, different test vectors are used with specific sequence as shown in **Table 5**.

3.2.2. Detection of Resistive Open Faults for the Different Switching Thresholds of the Inverter Proposal

It is mentioned earlier that changing V_M of the inverter is needed to detect all memristor faults. Order of test vector application is required for full fault coverage in this proposed solution. For instance, consider the fault M1 OD/OS for the NAND logic gate, *i.e.*, resistive open in transistor M1 in **Figure 2(a)**. This fault isolates the NAND output from the supply voltage. So if the input vector applied is 01 or 10 the output node will also not be connected to ground because this turns transistor M2 off. The output then is floating and retains its previous logic state as in [16]. In order to detect this fault, an initializing vector activating the pull down NMOS transistor M1 must be applied which is 11 in this case. Applying the test vector 11 pulls down the output of the NAND gate to "0". After applying the initializing vector, any of the other two test vectors 01 or 10 could be applied. This keeps the output of the NAND gate in the floating state and will retain its previous logic state, which is "0" and hence the fault is detected. The same explanation could be given for other resistive opens in the NAND and NOR MRL logic gates. **Table 6** shows the test sequence required to detect open faults in transistors M1 and M2 for the MRL NAND gate.

The same explanation could be given for resistive opens in the NOR MRL logic gates. However, different test vectors are used with specific sequence as shown in **Table 7**.

Table 4. Test sequence/vectors to detect open faults in MRL NAND using scaled input voltages.

	M1 OD/OS	M2 OD/OS
Initializing Vector <AB>	Not Needed	01/10
Detecting Test Vector <AB>	01/10	11

Table 5. Test sequence/vectors to detect open faults in MRL NOR using scaled input voltages.

	M1 OD/OS	M2 OD/OS
Initializing Vector <AB>	01/10	Not Needed
Detecting Test Vector <AB>	00	01/10

Table 6. Test sequence to detect open faults in MRL NAND using low VM inverter.

	M1 0D/OS	M2 OD/OS
Initializing Vector <AB>	11	01/10
Detecting Test Vector <AB>	01/10	11

**Table 7.** Test sequence to detect open faults in MRL NOR using high VM inverter.

	M1 0D/OS	M2 OD/OS
Initializing Vector <AB>	01/10	00
Detecting Test Vector <AB>	00	01/10

3.3. Resistive Short Faults

Resistive short faults test results for both the two-input NAND and NOR gates are presented in **Table 8** and **Table 9**, respectively. The results are identical for the two proposed solutions.

For the NAND MRL, consider, for instance, the fault M1 DS, this forces the output node to always be logic high "1" as the node is shorted to the supply. Therefore, this fault is only detected by the test vector AB = 11, where the output in the fault free scenario should have been logic low "0". **Figure 5** shows the test result of this fault. Similar analysis could be given for short faults in the NOR MRL gate.

It is concluded from the test results shown above that, for the two proposals, the minimum test set required is identical to that obtained from the conventional single stuck-at fault model. However, it was shown that the order of applying the test vectors is important. A possible test pattern that obtains 100% fault coverage in NAND MRL gate is (10, 11, 01). This is a major difference between MRL NAND and CMOS NAND in that, despite both gates requiring the same three test vectors for full coverage, MRL requires a sequence of three vectors while CMOS requires a sequence of four vectors. A similar argument is valid for the NOR gate. It is also concluded that detecting resistive open faults in MRL NAND/NOR gate depends on the test speed.
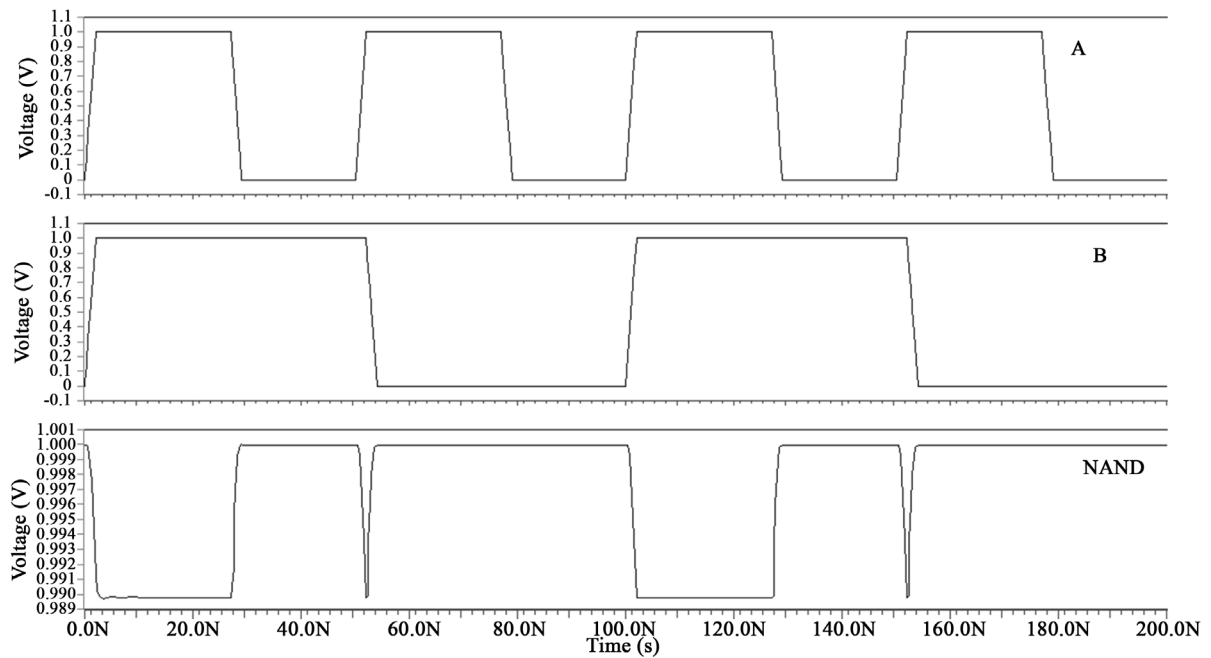
Figure 5. NAND M1 DS fault.

Table 8. Resistive short faults test results in MRL NAND.

Transistor	Faults		
	DS	GD	GS
M1	11	00/01/10	00/01/10
M2	00/01/10	00/01/10	11

Table 9. Resistive short faults test results in MRL NOR.

Transistor	Faults		
	DS	GD	GS
M1	01/10/11	01/10/11	00
M2	00	01/10/11	01/10/11

4. Conclusions

Memristors have been physically characterized in 2008 by HP. One of the main advantages of using memristors in memories, analog circuits, neuromorphic systems and digital circuits is its area occupancy. Memristors and CMOS inverters are integrated with each other to realize logic gates such as NAND and NOR. This design logic family is called MRL. The main advantage of this logic family is that it saves physical area and therefore increases logic density, which allows the increase of system complexity. Hence, it is important to test these gates efficiently.

In this study, the TEAM model and the 45 nm CMOS technology were used. The memristor stuck at fault model and the five-fault model are considered. Faults are injected one at a time. A fault is considered detected if the output is different from the fault-free output scenario.

During the testing of memristor faults, the input of the inverter falls in the undefined region and this can lead to test escapes. Therefore, two solutions were proposed to face this challenge. The first is to apply scaled input voltages and the second is to change the V_M of the inverter. It is shown that the minimum test set obtained in order to obtain full coverage for MRL NAND/NOR gates is identical to that obtained from the conventional single stuck-at fault model. However, the speed of applying the test vectors and the test order should be taken into account. Unlike CMOS NAND/NOR that requires a sequence of four vectors for 100% fault coverage, MRL NAND/NOR requires a sequence of only three test vectors.

References

[1] Kvatinsky, S., Wald, N., Satat, G., Weiser, U.C. and Friedman, E.G. (2012) MRL—Memristor Ratioed Logic. *Proceedings of the International Cellular Nanoscale Networks and their Applications*, Turin, 29-31 August 2012, 1-6.

[2] Haron, N.Z. and Hamdioui. S. (2008) Why Is CMOS Scaling Coming to an End. *Proceedings of the International Design and Test Workshop* (*IDT*), Monastir, 20-22 December 2008, 98-103. http://dx.doi.org/10.1109/idt.2008.4802475

[3] Chua, L.O. and Kang, S.M. (1971) Memristive Devices and Systems. *IEEE Transactions on Circuit Theory*, **18**, 507-519. http://dx.doi.org/10.1109/TCT.1971.1083337

[4] Strukov, D.B., Sinder, G.S., Stewart, D.R. and Williams, S.R. (2008) The Missing Memristor Found. *Nature*, **453**, 80-83. http://dx.doi.org/10.1038/nature06932

[5] Mazumder, P., Kang, S. and Waser, R. (2012) Mmeristors: Devices, Models and Applications. *Proceedings of the IEEE*, **100**, 1911-1919. http://dx.doi.org/10.1109/JPROC.2012.2190812

[6] Wu, J. and Choi, M. (2010) Memristor Lookup Table (MLUT)-Based Asynchronous Nanowire Crossbar Architecture. *IEEE Conference on Nanotechnology*, Seoul, 17-20 August 2010, 1100-1103. http://dx.doi.org/10.1109/nano.2010.5697869

[7] Haron, N.Z. and Hamdioui, S. (2011) On the Defect Oriented Testing for Hybrid CMOS/Memristor Memory. *IEEE Asian Test Symposium*, New Delhi, 20-23 November 2011, 353-358. http://dx.doi.org/10.1109/ats.2011.66

[8] Haron, N.Z. and Hamdioui, S. (2012) DfT Schemes for Resistive Open Defects in RRAMs. *Design*, *Automation and Test in Europe*, Dresden, 12-16 March 2012, 799-804. http://dx.doi.org/10.1109/date.2012.6176603

[9] Hongal, V.A., Kotikalapudi, R., Kim, Y.B. and Choi, M. (2011) A Novel "Divide and Conquer" Testing Technique for Memristor Based Look up Table. *IEEE International Midwest Symposium on Circuits and Systems*, Seoul, 7-10 August 2011, 1-4.

[10] Kannan, S., Rajendran, J., Karri, K. and Sinanoglu, O. (2013) Sneak Path Testing of Crossbar-Based Non-Volatile Random Access Memories. *IEEE Transactions on Nanotechnology*, **12**, 413-426. http://dx.doi.org/10.1109/TNANO.2013.2253329

[11] Chen, Y and Li, J. (2015) Fault Modeling and Testing of 1T1R Memristor Memories. *IEEE VLSI Test Symposium*, Napa, CA, 27-29 April 2015, 1-6. http://dx.doi.org/10.1109/vts.2015.7116247

[12] Olbrich, T., Perez, J., Grout, I.A., Richardson, A.M.D. and Ferrer, C. (1996) Defect-Oriented VS Schematic-Level Based Fault Simulation for Mixed-Signal ICs. P*roceedings of the International Test Conference*, Washington, 20-25 October 1996, 511-520. http://dx.doi.org/10.1109/test.1996.557076

[13] Amer, S.H., Emara, A.S., Mohie El-Din, R., Fouad, M.M., Madian, A.H., Amer, H.H., Abdelhalim, M.B. and Draz. H (2014) Testing Current Mode Two-Input Logic Gates. *Proceedings of the Canadian Conference on Electrical and Computer Engineering*, Toronto, 4-7 May 2014, 1-6. http://dx.doi.org/10.1109/ccece.2014.6901052

[14] Sindiaj, S., Agrawal, V.D. and Singh, V. (2011) Test and Diagnosis of Analog Circuits Using Moment Generating Functions. *Asian Test Symposium*, New Delhi, 20-23 November 2011, 371-376.

[15] Goel, S., Elgamel, M., Bayoumi, M. and Hanafy, Y. (2006) Design Methodologies for High-performance Noise Tolerant XOR-XNOR Circuits. *IEEE Transactions on Circuits and Systems I*, **53**, 867-878.
http://dx.doi.org/10.1109/TCSI.2005.860119

[16] Li, J., Tseng, C.W. and McCluskey, E.J. (2001) Testing for Resistive Opens and Stuck Opens. *Proceedings of the IEEE International Test Conference*, Baltimore, 30 October-1 November 2001, 1049-1058.
http://dx.doi.org/10.1109/test.2001.966731