Detection Limit for Intermediate Faults in Memristor Circuits

Rasika Joshi and John M Acken
Department of Electrical and Computer Engineering,
Portland State University,
Portland, OR, USA
joshiras@pdx.edu and acken@pdx.edu

Abstract— Memristor crossbar structures are widely used in logic, memory, security, and neuromorphic applications. It becomes necessary to test these devices for faults since they are prone to high defect densities. In this paper, we introduce a new terminology “intermediate faults” in memristor circuits. Intermediate faults are faults whose memristor resistance values lie between low resistance state (LRS) and High resistance state (HRS) values. This paper extends the fault detection method for HRS/LRS stuck-at faults to detecting intermediate faults using sneak paths in memristor circuits. We describe the importance of setting the detection limit for testing intermediate faults. Our simulation results present the detection limit value for intermediate resistances using five long and three long sneak paths in a 3x3 crossbar array. Our fault detection scheme can be used for detecting intermediate faults along with stuck-at low resistance and stuck-at high resistance faults.

Keywords— Memristors, sneak paths, sneak path current, intermediate faults, fault detection

I. INTRODUCTION

In 1971, Leon Chua theorized the existence of memristor [1]. Memristor was called the fourth element among the other fundamental elements, namely the resistor, capacitor, and inductor. R.S. William’s team came up with a physical model of the memristor in 2008 that served as a foundation for understanding the hysteresis behavior of the memristor [2]. Since then, there have been several efforts in memristor design, architecture, memory and test for various applications. Memristor crossbars are a widely used architecture that offers high density and low energy consumption to outperform conventional memories. Crossbar structures are used in several applications such as logic, memory, stochastic computation, security and neuromorphic systems [3-7].

Nanoscale memristor devices are prone to defects due to the nondeterministic nature of nanoscale fabrication. This necessitates testing memristor devices to detect memristor faults and to diagnose the location of such faults. Providing high quality and efficient test solutions are of great importance in order to enable the commercialization of memristor devices.

The unique properties of memristor crossbar arrays: sneak paths and sneak path currents are used in our paper for testing memristor circuits. Sneak paths are current paths parallel to the target memristor path. Characterization of sneak paths in memristor circuits has been published in [8], which characterized the sneak path current as a function of the size of the array, resistance programming, input voltage and I/O switch-vectors in this study. This work provides boundary conditions for applications that use memristor crossbar arrays and provides insights to memristor crossbar testing. Previous work in [9] describes a testing technique for memristor High resistance state (HRS) and Low resistance state (LRS) stuck-at fault detection and fault diagnosis using sneak paths. The work focused on an improved test-time fault detection technique for stuck-at low resistance and stuck-at high resistance faults.

We are extending the fault detection method to test for intermediate faults in memristor circuits in this paper. We propose a detection limit method for intermediate faults using a 3x3 crossbar array as an example.

The paper is divided into sections to discuss the methodology and an example of fault detection of intermediate faults. Section 2 discusses the background for memristors, sneak paths, and characterization of sneak paths. Section 3 discusses intermediate faults. We also discuss the fault detection method in brief. Section 4 presents the simulation results and observations on the detection limit for a 3x3 crossbar example. Section 5 discusses the conclusion and current limitations of the research.

II. BACKGROUND

Memristor is a two-terminal passive device whose resistance varies based on the input voltages applied to it. In simple words, if a positive voltage is applied to this two-terminal passive device the resistance decreases, and negative voltage applied increases the resistance. HP labs developed memristors that consisted of 50nm wide Titanium oxide ($\text{TiO}_2$) film sandwiched between two platinum electrodes. The film consists of un-doped low conductivity zone with approximately 2:1 ratio of oxygen to titanium and a high-conductivity zone with oxygen deficient TiO$_{2-x}$. The internal state variable of memristor denoted by “$x$” is equivalent to ratio of the length of the doped region “$w$” to the total length of the TiO$_2$ film “$D$”. When a positive voltage ($v(t) > 0$) is applied across the memristor, the resistance of the memristor decreases due to the drifting of the oxygen vacancies into the un-doped region. Similarly, when a negative voltage ($v(t) < 0$) is applied, the overall resistance of the memristor increases since the oxygen vacancies drift in the opposite direction. Low resistance state
(LRS) $R_{on}$ occurs when $x=1$ and high resistance state (HRS) $R_{off}$ occurs when $x=0$. $R_{on}$ and $R_{off}$ are the limit values of the memristor resistance when $w=D$ and $w=0$ [10] respectively. The total memristance $M(x)$ of the memristor is expressed in (1).

$$M(x) = xR_{on} + (1-x)R_{off}$$

(1)

where $x = (w/D) \in (0,1)$

The LRS and HRS values of the memristor represent logic values of 1 and 0, respectively. The range of published memristor values for LRS and HRS are described in [8]. In that work the representative values of LRS=$10\,\Omega$ and HRS=$1\,\Omega$. The operations of reading and writing to a memristor are described in previous references and will not be discussed in detail in our paper. We have used a simple resistive model for crossbar circuits for our fault testing and sneak path current analysis.

Sneak paths are current paths parallel to the intended current paths occurring in memristor crossbar architectures. Sneak path currents may impact the performance of the crossbar array causing incorrect read and writes in memory arrays. Fig. 1 shows an example sneak path current in a 3x3 crossbar circuit. Here, the current path highlighted in bold is the intended current path through the selected cell at the intersection between the column and row of interest. The intended current path is also called the primary current path. The current path highlighted in dashed is the sneak path current flowing parallel to the intended current path. The IO test vector set for a memristor crossbar array comprises of the IO switch-vector settings for the wordlines and bitlines [8]. The wordlines are the horizontal connections and the bitlines are the vertical connections. Let us consider a crossbar array of size $m \times n$ where $m$ is the number of rows or wordlines and $n$ is the number of columns or bitlines. $m_{open}$ is the number of wordlines open, and $m_{closed}$ is the number of wordlines closed. If the input voltage source is connected to the wordline, the wordline is closed and it’s open if it’s not connected to the voltage source. A wordline closed is called a selected wordline. $X_i$ is the switch state for the $i^{th}$ row or the input wordline, where “1” is closed and “0” is open. Let us define $n_{open}$ as the number of bitlines open, and $n_{closed}$ as the number of bitlines closed. The grounded current sensor on that column output is connected to the bitline for a closed bitline. An open bitline is not connected to a grounded output current sensor. A bitline closed is also called as selected bitline. $Y_j$ is the switch state for the $j^{th}$ column, or the output bitline where “1” is closed and “0” is open. The input state of $X_1X_2...X_m$ is combined with the output state of $Y_1Y_2...Y_n$ to define the I/O switch-vector of $X_1X_2...X_mY_1Y_2...Y_n$ [8].

The length of sneak paths and number of sneak paths were characterized as a function of input conditions, array size and memristor resistances [8]. Models were developed for sneak path length and sneak path current calculation for any given crossbar array size. This sneak path current analysis forms the basis for an improved method for detecting and diagnosing memristor faults.

III. INTERMEDIATE FAULTS

There have been several published fault models for memristor circuits, a sample of which are shown in Table I. Different types of physical defects such as variation in length, area and doping give rise to memristor faults. The fault detection method in [9] was used for detecting stuck-at LRS faults and stuck-at HRS faults. Stuck-at LRS faults are caused due to excessive doping. Hence, it will be stuck-at logic 1 irrespective of the voltage applied to it. Similarly, lack of doping could cause a memristor to be in a stuck-at HRS state. In this case, logic 1 is the expected output of a fault-free memristor, while logic 0 is the output in the presence of a SA0 fault. Table I has a column describing the value of the memristor due to defect. The question marks in the table represent intermediate faults in memristor circuits.

Intermediate faults are those type of faults where the memristor resistance state lies between LRS and HRS. SW (slow-to-write) and Deep faults are discussed in [11,13,15,16]. An intermediate fault could be SW1/SW0 (Slow-to-write 1) fault where the memristor state might be either in undefined state or could be logic 0/logic 1 respectively as discussed in [11,13,15,16]. It could also be deep fault where the memristor state could have elevated $M_{on}$ and $M_{off}$ resistance values for a Deep-1 fault and lower $M_{on}$ and $M_{off}$ resistance values for a Deep-0 fault. It could be an undefined state fault where the logical state of the device is unknown and can lie between logic 0 and logic 1. It becomes important to have a proper detection limit to detect such faults since the resistance state of these faults could be interpreted either at logic 0 or logic 1.

![Fig. 1. Primary current through memristor cell and sneak path current in a crossbar array [8].](image-url)
Let us consider a 3x3 memristor array with IO switch-vector =100 100 as shown in Fig. 2 showing the sneak path of M1c-M2c-M2b-M3b-M3a. It’s possible to create a long length sneak path when the memristors are at different resistance value. In this example, we can have four five long sneak paths namely: M1c-M2c-M2b-M3b-M3a, M1c-M3c-M3b-M2b-M2a, M1b-M2b-M2c-M3c-M3a, and M1b-M3b-M3c-M2c-M2a. The five long sneak path M1b-M3b-M3c-M2c-M2a can be achieved by keeping memristors M1c, M2b and M3a programmed to HRS while other memristors at LRS.

### A. Fault Detection Method

The fault detection methodology in [9] targets single stuck-at-LRS and single stuck-at-HRS faults in memristor crossbar arrays. The condition to detect a fault is based on the difference between the device current ($I_{\text{cut}}$) and the reference current ($I_{\text{reference}}$). The stuck-at LRS fault is detected for a given IO switch-vector if the $I_{\text{cut}} - I_{\text{reference}} > $ Detection limit, similarly, if the $I_{\text{reference}} - I_{\text{cut}} > $ Detection limit, the stuck-at HRS fault is detected for a given IO switch-vector. We are setting the detection limit for detecting stuck-at LRS and stuck-at HRS faults using a 3x3 crossbar array as an example. The fault detection method can be extended to intermediate faults as well. However, the detection limit needs to be properly defined to detect such faults based on the memristor state. In the next section, we will show simulation results and the proposed detection limits to detect intermediate faults in a 3x3 crossbar array.

### IV. Fault Detection Example Using 3x3 Crossbar

Long length sneak paths are those sneak paths whose length is more than three memristors. Fault detection uses five memristor long sneak paths is described in the following example. A faulty memristor along the long sneak path is detected using the long length sneak paths. The sneak paths are calculated using our python-based tool [8]. The tool generates an LTspice compatible crossbar circuit for a given size of array and resistance programming. The sneak path current simulations are performed using LTspice.

---

**TABLE I Memristor Faults**

<table>
<thead>
<tr>
<th>Fault</th>
<th>Cause of Defect</th>
<th>References</th>
<th>Memristor state</th>
</tr>
</thead>
<tbody>
<tr>
<td>SA0 or SA open</td>
<td>Under-doped/open defect</td>
<td>[11][12][13][15][16]</td>
<td>HRS</td>
</tr>
<tr>
<td>SAL – stuck at logic level</td>
<td>Open defect</td>
<td>[11][15]</td>
<td>LRS</td>
</tr>
<tr>
<td>SA1 or SA short</td>
<td>Fully doped/short to VDD</td>
<td>[11][12][16]</td>
<td>HRS</td>
</tr>
<tr>
<td>SW0</td>
<td>Under-doped/Open defect</td>
<td>[11][13][15][16]</td>
<td>?</td>
</tr>
<tr>
<td>SW1</td>
<td>Excessively doped/open defect</td>
<td>[11][13][15][16]</td>
<td>?</td>
</tr>
<tr>
<td>Deep 0</td>
<td>Increase in Length or Decrease in Area</td>
<td>[11][13][15][16]</td>
<td>?</td>
</tr>
<tr>
<td>Deep 1</td>
<td>Decrease in Length or Increase in Area</td>
<td>[11][16]</td>
<td>?</td>
</tr>
<tr>
<td>UR</td>
<td>Excessively doped</td>
<td>[11][16]</td>
<td>?</td>
</tr>
<tr>
<td>Undefined state faults</td>
<td>Undefined logical state due to defect</td>
<td>[17]</td>
<td>?</td>
</tr>
<tr>
<td>Unknown state faults</td>
<td>Open defects</td>
<td>[18]</td>
<td>?</td>
</tr>
</tbody>
</table>
through $M1a$ is 100µA which is much larger than the sneak path current. We will look at single stuck-at faults in our simulations. For the path $M1c-M2c-M2b-M3b-M3a$, if there is a stuck-at HRS fault on either $M1c$ or $M3a$, the new sneak path current would be 1.91µA. Similarly, if there is a stuck-at HRS fault on either $M2c$ or $M3b$, the faulty sneak path current would be 2.83µA. The sneak path reduces to 3.75µA if there is a fault on $M2b$. Let us look at the sneak path current in the presence of different intermediate faults. Table II captures the different output sneak path current in presence of intermediate resistance faults between 10KΩ and 1MΩ. We have analyzed the sneak path current for $R_{\text{intermediate}}$ = 500KΩ, 200KΩ, 100KΩ, 50KΩ and 20KΩ resistance values. $I_{\text{Reference}}$ = 21.1µA and $I_{\text{CUT}}$ = $I_{\text{Faultycurrent}}$ for each of the memristors along the five long sneak path, as shown in Table II. Noise margins are technology dependent. Any fabrication process is going to have different amounts of variations which requires setting a tolerance or noise margin. For our example, we are considering the tolerance margin for the sneak path current values at ±10%. This tolerance margin is commonly used, for example in [19] and [20]. In [19], the authors reported a variation of 5%, 10% and 20% for considering variations in the circuit parameters. Similarly, in [20], the authors consider a 10% tolerance margin for measuring voltage thresholds. The difference between $I_{\text{Reference}}$ and $I_{\text{CUT}}$ needs to be greater than the detection limit to detect the HRS fault. Fig. 3 shows the difference between the fault free current with noise variation added and the 50KΩ faulty sneak path current with noise variation added is ~8µA. We choose half of this value which is ~4µA as the detection limit to help detect memristor faults with resistance values closer to HRS. The Detection limit is represented in Fig. 3 by a black box. From Table II, the detection limit of 4µA can help to detect all the $R_{\text{intermediate}}$ faults > 20KΩ. However, we cannot detect the 20KΩ intermediate fault since the difference between $I_{\text{Reference}}$ and the $I_{\text{CUT}}$ is ~3µA. With $R_{\text{LRS}}$=10KΩ, the 20KΩ intermediate resistance is closer to the LRS value and can be detected as a stuck-at LRS fault. We cannot detect such faults using this method and need to use another approach described below to detect stuck-at LRS faults.

For a single stuck-at LRS fault, three memristor long sneak paths are used to detect the fault. For the Fig. 4 example of 3x3 memristor array with IO switch-vector =100100 with all memristors in HRS, there are four possible sneak paths namely $M1b-M2b-M2a$, $M1b-M3b-M3a$, $M1c-M2c-M2a$, and $M1c-M3c-M3a$. The total simulated fault-free sneak path current is 0.8µA when the array is programmed with HRS =1MΩ. The fault-free sneak path current of 0.8µA increases to 1.1355µA if either $M1b$, $M1c$, $M2a$, and $M3a$ have a stuck-at LRS fault. The difference between the faulty current and the fault-free sneak current helps to detect the LRS fault. For this example, when $R_{\text{intermediate}}$ = 20KΩ, the faulty sneak path increases to 1.128µA. As described previously, we are picking the detection limit as 0.16µA of the worst-case difference between the fault free sneak current and the faulty sneak path current. As mentioned before, if the $I_{\text{CUT}}$ = $I_{\text{Reference}}$ > Detection limit, the stuck-at LRS fault is detected for a given IO switch-vector. For $R_{\text{intermediate}}$ = 20KΩ, the difference between the reference current and the faultly sneak path current is ~0.32µA and the stuck-at LRS fault can be detected. The original sneak path current of 0.8µA increases to 0.8736µA if the remaining memristors $M2b$, $M2c$, $M3b$ and $M3c$ have a stuck-at LRS fault. These faults may not be detected since the difference between the faulty and fault free sneak path currents is less than 0.16µA. Different IO switch-vectors such as 010100 and 001100 need to be used to detect LRS faults in $M2b$, $M2c$ and $M3b$, $M3c$ respectively for a complete fault coverage.

As the array size scales up, the detection process becomes more efficient since more faults can be detected using long length sneak paths. Our contribution [8] discusses characterization on sneak path lengths and sneak path currents for different array sizes, resistance programming and input voltage. For the fault detection analysis, we are not considering primary current since it stays constant even with the increase in the size of the array for a given resistance programming and input voltage.

V. SUMMARY AND CONCLUSION

Just as memristors have variations in resistance values, so do faulty memristors have variations in values. The term for the faulty memristor variations is intermediate faults. In this paper, we have shown a testing solution to detect intermediate faults in memristor circuits based on a published fault detection method using sneak paths. A different method needs to be used for intermediate memristors closer to HRS and another approach is used for detecting intermediate faults closer to LRS. We propose a method to set detection limits for intermediate fault detection, as demonstrated using 3x3 crossbar array simulations.

<table>
<thead>
<tr>
<th>$R_{\text{intermediate}}$</th>
<th>$I_{\text{Faultycurrent}}$</th>
<th>$M1c$</th>
<th>$I_{\text{Faultycurrent}}$</th>
<th>$M2b$</th>
<th>$I_{\text{Faultycurrent}}$</th>
<th>$M2c$</th>
<th>$I_{\text{Faultycurrent}}$</th>
<th>$M3a$</th>
<th>$I_{\text{Faultycurrent}}$</th>
<th>$M3b$</th>
</tr>
</thead>
<tbody>
<tr>
<td>20kΩ</td>
<td>17.6µA</td>
<td>17.98µA</td>
<td>17.79µA</td>
<td>17.6µA</td>
<td>17.79µA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>50kΩ</td>
<td>11.9µA</td>
<td>12.8µA</td>
<td>12.5µA</td>
<td>11.9µA</td>
<td>12.5µA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100kΩ</td>
<td>7.961µA</td>
<td>9.293µA</td>
<td>8.63µA</td>
<td>7.961µA</td>
<td>8.63µA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>200kΩ</td>
<td>5.027µA</td>
<td>6.618µA</td>
<td>5.824µA</td>
<td>5.027µA</td>
<td>5.824µA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>500kΩ</td>
<td>2.770µA</td>
<td>4.549µA</td>
<td>3.661µA</td>
<td>2.770µA</td>
<td>3.661µA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
REFERENCES


