Statistical Lifetime Analysis of Memristive Crossbar Matrix

Peyman Pouyan, *Student Member IEEE*, Esteve Amat, Antonio Rubio, *Senior Member IEEE*
Department of Electronic Engineering, UPC, Barcelona Spain
{peyman.pouyan, esteve.amat, antonio.rubio}@upc.edu

***Abstract*— Memristors are considered one of the most favorable emerging device alternatives for future memory technologies. They are attracting great attention recently, due to their high scalability and compatibility with CMOS fabrication process. Alongside their benefits, they also face reliability concerns (e.g. manufacturing variability). In this sense our work analyzes key sources of uncertainties in the operation of the memristive memory and we present an analytic approach to predict the expected lifetime distribution of a memristive crossbar.**

***Index Terms*—Memristor; uncertainty; crossbar; endurance; process variability; RRAM; emerging device**

I. Introduction

Memristors are one of the new emerging device alternatives that receive significant attention as a promising candidate for future nano-scale memory technologies [1]. They can scale down to few nanometers, and also have high switching speed, long retention time, low programming power and non-volatile characteristics [2]. Memristor theory analysis has been worked since many years ago [3], however it has not been until recently that the first physical devices were manufactured successfully [4]. Their device structure is an oxide sandwiched between two electrodes that can switch between two resistance states, high resistance state (HRS) and low resistance state (LRS); therefore they can store data in the state of the resistance value.

Memristors can be differentiated into various types based on their switching mechanism and constructing materials. For instance, one type of device is based on the HP memristor model [4], and another type is based on the physics-based model given in [5]. In this paper we focus on one of the most frequent and important memristor types, the binary metal oxide resistive switching random access memory (RRAM). The main theory that describes the switching mechanism in this memristor type is based on conductive filamentary (CF) switching [6]. This mechanism describes that the electroforming process in fresh samples forms an initial filament, between up and bottom electrodes, without connecting them to each other. At this state, the device is at its HRS (high resistive) mode. Applying a positive voltage will extend and complete the filament, until it makes a conduction path between the two electrodes and putting the device in LRS (low resistive) state, this is called the set operation. In the reset phase applying a negative voltage will rupture the filament and disconnect the conduction path, putting the device in HRS mode.

Generally, memristors are organized in a matrix-like structure called crossbar architecture to construct a 2D memory system. The storage cell in the crossbar can be built with only a single memristor (1R) or with one selector such as one transistor and one memristor (1T1R) [7]. Independently from the architecture and the cell structure the desired memory performance depends on the reliable operation of the memristor. However, manufacturing the memristors at nano-scale sizes make them susceptible to various kinds of reliability concerns. For instance, two of the most important sources of uncertainty in the design of memristive memories are process variability and endurance. Process variability causes deviations in the nominal high and low resistance values of the memristive device, while the endurance effect induces dynamic variations to the resistance values due to aging at each cycle, and also limits the write cycles. These two concerns are even more important in multi level memory cell (MLC), which is a desirable capability for future high-density memories [2].

In this sense, our paper is organized as following: Section II reviews the main sources of unreliability in memristors; Section III presents the impact of these unreliability factors in the read cycle certainty of a memristive memory; Section IV describes an analytical model of memristive crossbar lifetime; Section V proposes reconfiguration as an improving-reliability solution in redundant memristive crossbars; and finally Section VI concludes the paper.

II. Uncertainty in Memristors

A. Process Variability

Statistical variation in the high and low resistance values of the memristor is a major barrier in reliable operation of the devices. There are two types of resistance fluctuation in memristive devices: device-to-device and cycle-to-cycle variability [2], in this work we consider the first for circuit and system analysis. In the manufacturing of memristors at nano-scale sizes oxide thickness and doping concentration of oxygen vacancies uncertainties are the origins of the poor device-to-device uniformity. Moreover, the random electroforming process applied to fresh devices can impose different-size initial conductive filament inside the memristor, also resulting to deviations in high and low resistance values. To model all these variability sources in memristive devices, we consider two normal distributions for the HRS and LRS values in the set of manufactured samples [8] with a defined mean and standard deviation value (Section III).

B. Endurance

The second critical reliability concern in memristors in our study is the endurance degradation. This mechanism reduces HRS distribution while increases LRS, reducing consequently the distance window between HRS and LRS values causing additional variations from the nominal expected values (Fig. 1). Generally endurance phenomenon depends on different operation factors such as the environment temperature, device switching speed and also the material characteristics. Regarding this, three types of endurance failure behavior are reported in RRAM devices [9]. In the first mechanism the HRS resistance tends to decrease while the LRS resistance increases during cycles. Fig.1 depicts this degradation mechanism and defines the final failure state of the device as the point that the HRS to LRS resistance ratio decrease reaches an arbitrary given ratio *K*. The second and third endurance mechanisms are due to the abrupt or gradual (respectively) drop of HRS value during the cycles and getting stuck at LRS value [9].

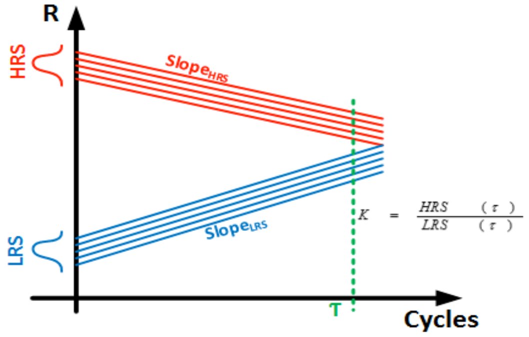

Fig. 1. The LRS and HRS degradation during the write cycles, *k* is the ratio between them and *τ*=lifetime in terms of endurance cycles

In this work, we focus on the first mechanism as it is considered the worst case [9, 10]. Therefore, we analyze its impact in the reliability of a memristive crossbar memory.

III. Uncertainty Impact in Memristor Read Cycle

There are different approaches to read the memristance state of a component in a crossbar memory [11]. The main technique is based on the comparison of the selected memristive cell's current (or voltage) in the crossbar with a reference cell one [11]. The resistance variations mentioned previously can reduce the sensing window or even cause a fault in the read process (for instance sensing LRS instead of HRS or vice versa). In order to investigate the reliability of the read process in the memristive crossbar we analyze the probability of the error using Matlab simulation while performing the reading operation. Let's consider in this work LRS and HRS resistance random values defined by normal distributions, with characteristics obtained from experimental results in the literature [9,10]: *μ(LRS)=1KΩ*, *μ(HRS)=100KΩ* and *σ(LRS)=σ(HRS)=20% of the mean value* corresponding to their mean and standard deviation values.

Fig. 2.a shows the two truncated LRS and HRS probability distributions of fresh devices between *-3σ* to *+3σ*. Endurance will impose the *μ(*HRS) and *μ(*LRS) values to get closer to each other as we have mentioned before. Regarding to this, Fig. 2.b shows the probability of the reading error (*Pe*), as a factor that determines the possibility of an incorrect read operation in respect to a given reference resistance value (R_{ref}). The *Pe* graph is plotted while sweeping the reference resistance value (in which below R_{ref} is expected to be in LRS state and higher than R_{ref} the HRS state) along the two LRS and HRS distributions. It is observed that as the *μ(*HRS) and *μ(*LRS) values get closer to each other (getting their ratio smaller), the *Pe* in the read operation of a memristive cell becomes higher. Therefore, Fig 2.b points out the relevance of considering the simultaneous impact of variability and endurance for robust and reliable memristive memory design.

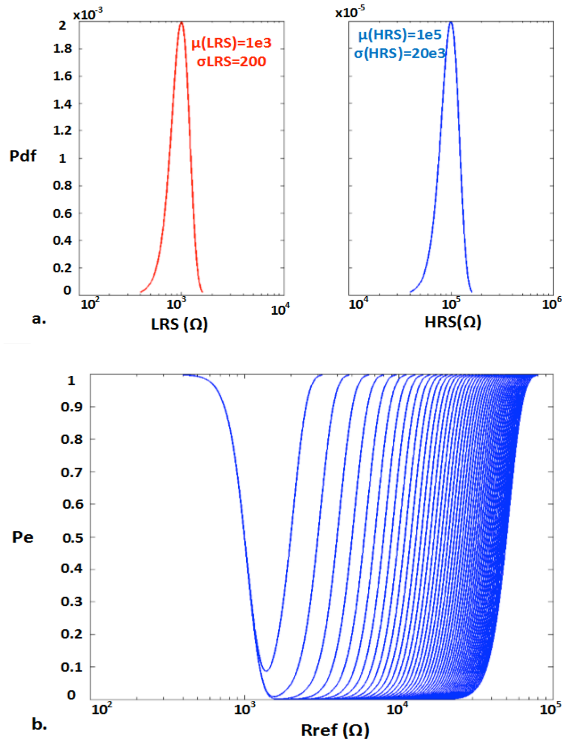

Fig. 2. a) The LRS and HRS probability distributions following a normal distribution b) Pe in respect to different values for Rref.

IV. Crossbar Lifetime Analysis

In order to estimate the probability distribution of a memristive crossbar lifetime, we first evaluate the probability distribution of a single memristor's lifetime, starting from cycle zero and assuming a degrading phenomenon due to endurance as shown in [9,10]. We can estimate the lifetime of a memristor (*τ*, a random variable) by assuming a linear approximation for the degradation slopes of LRS and HRS with the number of cycles following the concept shown in Fig. 1. The values for these slopes, as well as *HRS(0)* and *LRS(0)* (which are the initial resistance values at cycle zero) are all taken from experimental measurements (Fig. 5 [9]). Then by defining the point of failure for a memristor as the point where the *HRS(τ)/LRS(τ)* ratio becomes equal to a given *K* value (as shown in Fig. 1), we derive the point of failure as in equation (1):

$$K = \frac{HRS(0) - SlopeHRS \times \tau}{LRS(0) + SlopeLRS \times \tau} \qquad (1)$$

Solving (1) would result in obtaining the memristor lifetime (τ) expression, measured in number of endurance cycles (2).

$$\tau = \alpha \times HRS(0) - \beta \times LRS(0) \qquad (2)$$

Where α and β are coefficients that depend only on the degradation of slopes ($slope_{HRS}$ and $slope_{LRS}$) and the selected K parameter. Next, from the principle of sum of normal distributions, the mean and variance values for the τ variable are calculated as in (3) and (4):

$$\mu(\tau) = \alpha \times \mu(HRS(0)) - \beta \times \mu(LRS(0)) \qquad (3)$$

$$\sigma^2(\tau) = \alpha \times \sigma^2(HRS(0)) + \beta \times \sigma^2(LRS(0)) \qquad (4)$$

So, under the previous assumptions (μ and σ of HRS and LRS, same as section III and taking now K=5 [10]), the lifetime of a single memristor (starting from cycle zero) follows a normal distribution, as it is shown in Fig. 3.a (distribution τ).

Regarding this information, in the next step we analytically obtain the probability distribution of the number of cycles up to the first failure in a multiple-component crossbar matrix with n memristors (let's consider n=16 as a matter of example). In other words, we will find the probability distribution of the number of cycles for the memristor (*i-th*) of the crossbar, which first reaches the critical ratio of K, considering process variability and independent variables for each memristor of the matrix. This can be calculated as the probability distribution of the minimum of the individual independent random variables (τ_i) in each cell for the complete n-component crossbar. Assuming a set of τ_i values with a normal distribution where $1 \leq i \leq n$, we are intended to calculate the probability density function (*PDF*) of the random variable g, where $g = Min\ (\tau_1, \tau_2, \ldots, \tau_n)$. The *CDF* of g is found as in (5):

$$CDF(g) = 1 - (1 - CDF(\tau))^n \qquad (5)$$

From (5) we calculate the *PDF(g)* as in (6) and (7):

$$PDF(g) = \frac{\partial CDF(g)}{\partial \tau} \qquad (6)$$

$$PDF(g) = n \times (1 - \int_{-\infty}^{\tau} \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(z-\mu)^2}{2\sigma^2}} dz)^{n-1} \times \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(\tau-\mu)^2}{2\sigma^2}} \qquad (7)$$

Fig. 3.a also shows the probability distribution of the crossbar lifetime up to the first failure *(PDF(g))*. We have determined the mean and standard deviation of the g by numerical calculation verifying the analytical result. We have verified the correctness of the *PDF(g)* with performing 10,000 Monte-Carlo experiments. In each experiment we generate n random numbers (τ_1, τ_2, …. , τ_n), samples of a normal distribution with a known $\mu(\tau)$ and $\sigma(\tau)$, each one representing the lifetime of a single memristor and then we find the minimum value among them. As shown in figures the *PDF(g)* from the Monte Carlo analysis in Fig 3.b perfectly matches with our analytic approach.

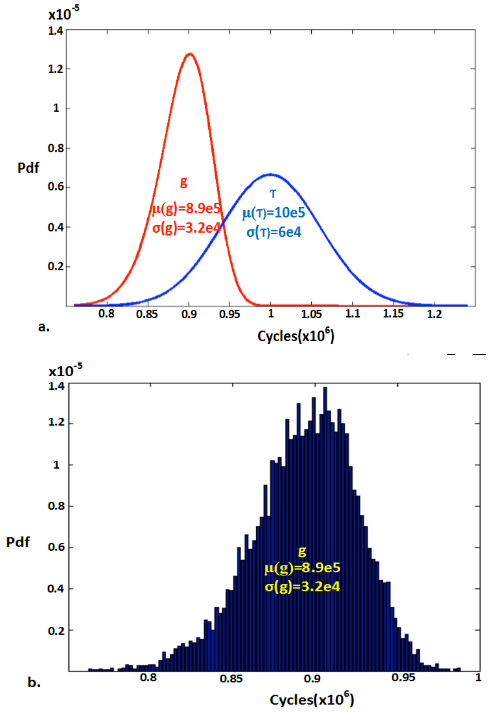

Fig. 3 a) The τ (fresh memristor lifetime) and g (crossbar lifetime up to first fail) probability distributions b) The g probability distribution verified with Monte-Carlo simulation

In the following we proceed to find the probability distribution of cycles up to the second failure. By having the number of cycles at the beginning (τ) and at the point of first failure (g), we define another random variable (h), which is lifetime at cycle zero minus the time of the first failure as in (8):

$$h = \tau - g \qquad (8)$$

Then equation (9) presents the mean value for the h and (10) presents the standard deviation. Note that since τ and g are not independent random variables the $\sigma(h)$ is calculated by considering the correlation factor (ρ) as in 10:

$$\mu(h) = \mu(\tau) - \mu(g) \qquad (9)$$

$$\sigma(h) = \sqrt{\sigma^2(\tau) + \sigma^2(g) + 2 \times \rho \times \sigma^2(\tau) \times \sigma^2(g)} \qquad (10)$$

Next, to obtain the crossbar lifetime up to the second failure, similar to the first failure, we need to find the *PDF(y)*, when $y = Min(h_1, h_2, \ldots, h_{n-1})$. It is now *n-1* because one memristor is not considered after the first fail. In this sense Fig. 4.a shows the probability distribution of lifetime for a crossbar up to the second failure (which is the number of cycles for the interval between first and second fail). We also verify this lifetime distribution by using Monte-Carlo simulation in Fig. 4.b. The results of our lifetime analysis in Fig. 3 and Fig. 4 demonstrate that the memristive crossbar lifetime is highly reduced for posterior fails after the first and mainly second failure. This implies the need to establish efficient reconfiguration mechanisms to achieve reliable memristive crossbar applications. In the next section we analyze the crossbar lifetime using our reconfiguration approach.

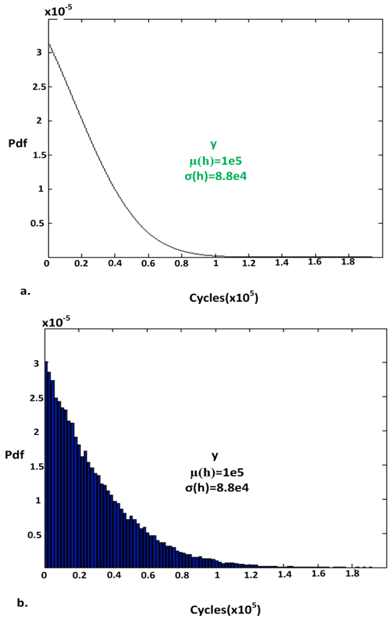

Fig. 4. a) The y (crossbar lifetime between first to second fail) probability distributions b) The y probability distribution verified with Monte-Carlo simulation

V. RECONFIGURATION IN MEMRISTIVE CROSSBAR

The reconfiguration techniques can be differentiated into two main categories: 1-Reactive, 2-Proactive [12, 13]. The former can be based on conventional repair techniques such as row/column replacement of faulty one with a spare one [12]. In such an approach the use of spare units is limited only to the time that a fail occurs. Another approach is based on the utilization of spare and operational units together with the highest simultaneity and is called the proactive reconfiguration [13]. This presents a relevant enhancement of the system performance and for this we have chosen as a baseline configuration. Therefore, in our work we define two types of proactive techniques named as non-adaptive proactive and adaptive proactive approach.

A. *Non-Adaptive Proactive Approach*

Let's assume a memristive crossbar of size NxN, where only mxm of its units are in active mode to perform a given function. Both NxN and mxm crossbars are square sized and united (squared shape). In this non-adaptive proactive approach we consider that the reconfiguration mechanism skips the whole original mxm crossbar, even with most healthy memristors once a memristor in the original mxm reaches its lifetime limit. Every time a near-failing device is detected we skip to the next mxm crossbar (see Fig. 5). So then, in this technique the memristive crossbar lifetime would be solely extended if we can allocate as many mxm unique crossbar structures, inside the NxN crossbar. This means that when an mxm crossbar reaches the end of its lifetime, the whole mxm structure shifts inside the NxN crossbar. We consider that an operative matrix ends its life when the first memristor of the matrix reaches the end of its lifetime. Consequently, the lifetime extension is in proportion of number of mxm crossbars that can fit inside the NxN. If N is equal to α times of m (N= α x m) then α^2 times unique mxm crossbars can be allocated inside a NxN crossbar and the lifetime extension would be equal to α^2 times of a single mxm crossbar lifetime.

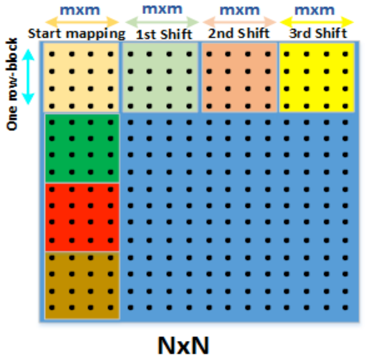

Fig. 5. Non-adaptive proactive approach of a mxm=4x4 in NxN=16x16 memristive crossbar, where each shift is a complete mxm jump

The non-adaptive technique can extend the crossbar lifetime by getting benefit from high redundancy inside a crossbar. However, the mxm lifetime in each sector arrangement is limited to the weakest unit in that structure, and this limits the efficient utilization of the resources both in the local mxm and global NxN crossbar. Therefore, in order to optimize the lifetime extension to its maximum level we would need to use another proactive approach (adaptive proactive), which can perform the shifts more intelligently. This technique is based on a dynamic redundancy allocation strategy that we propose in the next section.

B. *Adaptive Proactive Approach*

This technique provides the possibility to use the crossbar resources more uniformly, in order to extend its lifetime efficiently. In this approach, the mxm structure can shift inside the NxN crossbar, but this time, the shift would be a given number of columns corresponding to the place of the weakest unit, where the weakest unit is located. The skip is not fixed now but sensitive to the location of the near-failing device (see Fig. 6). Note that in the previous approach the shifting step was the complete mxm frame in comparison with now that the new mxm structure can have some nodes in common with the previous mxm structure. For example, if we assume that NxN=16x16 and mxm=4x4 is allocated in the left corner of NxN crossbar, if there is a weak unit (i.e. unit with the lowest endurance, or with the HRS/LRS value closer to K) in column 3 of mxm crossbar which is reaching its lifetime limit, then the mxm crossbar will only shift 3 columns to the left and still utilize one column of the previous mxm structure, what involves a benefit in a more optimum use of the system resources.

There could be different strategies for dynamic shift of mxm inside NxN crossbar, for instance the mxm shift can be in the x-axis or y-axis or even in diagonal direction. In addition, the first mapping of the mxm inside NxN can be in different locations such as in the center or in the left/right corner and this would influence the shifting strategy. In this work we assume that the first region for mxm mapping is in the left corner of NxN crossbar, and the mxm shifts in respect to place of the weakest element in the x-axis direction (canonical strategy). When the mxm matrix reaches to the right corner of NxN where it cannot be shifted by maintaining its original size anymore, the mxm crossbar would jump to the first left corner of the NxN crossbar (a complete shift of mxm structure in y-axis direction). In the next phases the shifting would be similar until most of resources in NxN crossbar are utilized. Fig. 6 describes this procedure for an example (m=4 and N=16).

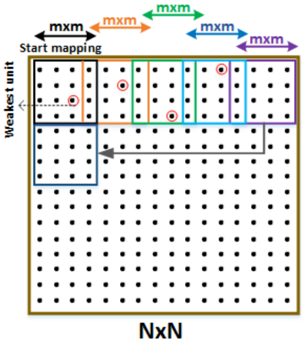

Fig. 6. Adaptive Proactive approach of an mxm=4x4 in NxN=16x16 memristive crossbar, where each shift is in respect to location of weakest unit

In order to estimate the crossbar lifetime in the adaptive proactive reconfiguration approach, first we determine the expected number of shifts that an mxm crossbar can make inside an NxN crossbar. Each shifting step can vary from 1 to m depending to the place of weakest element inside mxm. By solving this problem and knowing the total crossbar lifetime with adaptive proactive reconfiguration we can have an approximation of average lifetime at each mxm shift inside the NxN crossbar. If T=Total crossbar lifetime, t_1= mxm lifetime at first configuration, t_j=lifetime of mxm at each shift, and E=expected number of mxm shifts inside NxN, then we can write the total lifetime of mxm in NxN crossbar as (11):

$$T = t_1 + \sum_{j=1}^{E} t_j \qquad (11)$$

We can solve the expected number of mxm shifts in NxN by using different approaches, while considering two different assumptions. The first assumption is based on the fact that each shift has an equal probability, as it can be a random number between 1 to m from uniform distribution and its probability is equal to 1/m. Considering this, we could solve the problem mathematically and also by Monte-Carlo simulation, as it follows in sub-sections 1 and 2.

1) Analytic Approach in uniform assumption

The expected number of shifts can be analytically solved by applying the mathematics in [14] to our case. In this sense, the number of shifts can be derived when i=0 in the (12):

$$E(i) = 1 + (1/m)\{E(i+1) + E(i+2) + \ldots.. + E(i+N)\} \qquad (12)$$

Where i ranges from N-1 to 0 and E(0) is equal to average number of shifts. Fig. 7 depicts the average number of shifts for different values of m and N.

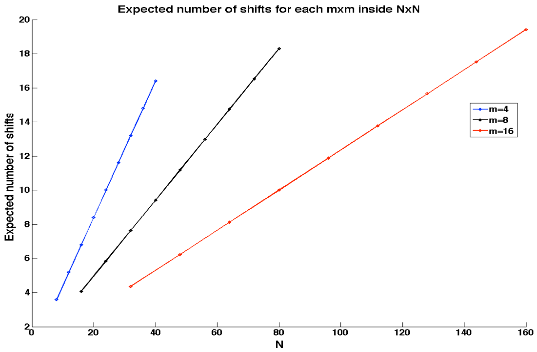

Fig. 7.Average number of shifts in analytic approach, for mxm=4x4, 8x8, 16x16 and NxN ranging from 8x8 to 160x160

For example, it is observed in Fig. 7 that if mxm=4x4 and NxN=40x40 then the average number of shifts for mxm in first row block of NxN would be equal to 17. This means that from first until the last valid shift it would take 17 steps that the mxm structure can shift inside one row block of NxN.

2) Monte-Carlo Approach in uniform assumption

Next, we verify our mathematical result with Monte-Carlo simulation. To do so, we repeatedly generate random numbers (representing the possible shifts) between 1 and m from the uniform distribution, and sum up them together. Once the sum of generated random numbers equals or overflows the value N, the number of random generations is our expected value for possible shifts of mxm inside a row block of NxN. We repeat this procedure 10000 times, and at the end calculate the average of required shifts for a corresponding m and N. Fig. 8 presents our result for the Monte-Carlo simulation, the average number of shifts and the corresponding standard deviation; it verifies with our mathematical result.

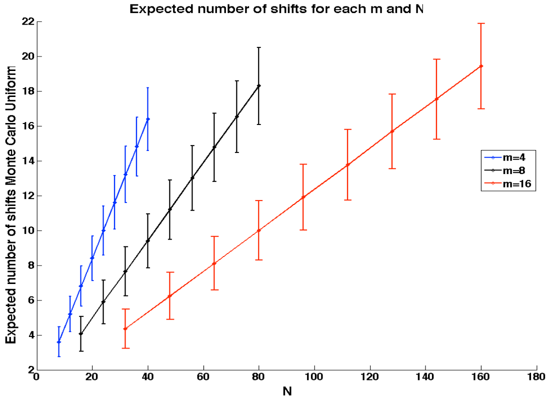

Fig. 8.Average number of shifts in Monte-Carlo, for mxm=4x4, 8x8, 16x16 and NxN ranging from 8x8 to 160x160

Our second assumption is based on the fact that each shift does not have an equal probability, for instance because of variability or endurance (each shift can be a random number between 1 to m and its probability now, is not equal to 1/m and can be a random number between 0 and 1). This time we solve the problem by a semi-analytic approach as described in sub-section 3.

3) Mathematical Semi-Analytic Approach in non-uniform assumption

By performing mathematical analysis now in the semi-analytic approach the (12) can be written as (13):

$$E(i) = 1 + r_1 \times E(i+1) + r_2 \times E(i+2) + \ldots.. + r_{N-1} \times E(i+N) \quad (13)$$

Where i ranges from N-1 to 0 and E(0) gives the average number of required shifts. The r_1 to r_{N-1} are random numbers between 0 and 1. We create these random numbers as following:

1-Create m random numbers that sum up 1
2-Repeat step 1 α times, where α =N/m and make a set of random numbers (r_1 to r_N)
3-Use N-1 terms of the above set (r_1 to r_{N-1}) for the calculation in (13)

Next, we calculate E(0) mathematically from (13) and repeat the above process (steps 1-3) 10000 times, each time with a different set of random portions (r_1-r_2,….,r_{N-1}). At the end we average all E(0) values from each iteration and obtain the expected number of shifts in this scenario. Fig 9 depicts the expected number of shifts in the semi-analytic approach and compares it with the analytic approach.

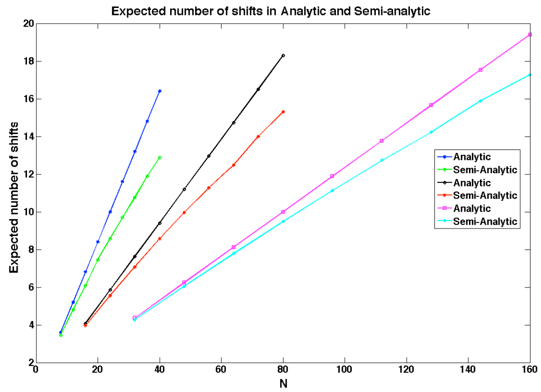

Fig. 9.Average number of shifts in semi-analytic, for mxm=4x4, 8x8, 16x16 and NxN ranging from 8x8 to 160x160

It is observed that the analytic approach would give us the upper bound for the average number of shifts. As an example if mxm=4x4 and NxN=40x40, then the average number of shifts for mxm in one row block of NxN would be equal to 13 in the semi-analytic approach. Thus, we have an approximation of the number of shifts that an mxm structure can make inside an NxN crossbar. Thus, Table I presents the approximations of the average mxm lifetime in the crossbar for each shift. It is observed that, as the mxm structure gets bigger the lifetime average per shift gets smaller (due to larger number of shifts for bigger mxm structures).

Table I. Average lifetime per shift in mxm structures.

	mxm=4x4	mxm=8x8	mxm=16x16
Average lifetime per shift	6.23 x 10^5	5 x10^5	4.2 x 10^5

Next, we perform Monte-Carlo simulations on crossbar matrixes constructed of memristors with random lifetime, and compare the total crossbar lifetime in the adaptive and non-adaptive approach. So then, Table II presents some lifetime extension results as a matter of example, highlighting the benefit of the adaptive proactive approach.

Table II. Lifetime extension adaptive versus non-adaptive.

	mxm=4x4	mxm=8x8	mxm=16x16
Lifetime extension	30%	45%	65%

VI. Conclusions

This work studies two major sources of unreliability in memristive crossbar-based memories: process variability and endurance; and analyzes their impact in the crossbar lifetime. An analytical procedure has been presented to evaluate the lifetime of a multiple-element crossbar for the first and second failure, which has also been verified by a Monte-Carlo approach. Additionally two innovative proactive reconfigurations have been proposed and the gain in crossbar lifetime has been evaluated utilizing for both of these techniques.

Acknowledgment

This work has been funded by Spanish MINECO and ERDF TEC2013-45638-C3-2-R.

References

[1] www.ITRS.net

[2] H.-S. P. Wong, *et. al*, “Metal–Oxide RRAM,” *Proc. IEEE*, vol. 100, no. 6, pp. 1951–1970, 2012.

[3] L.O. Chua, “Memristor-The missing circuit element,” *IEEE Transaction on Circuit Theory*, vol. 18, pp. 507-519,1971.

[4] D. B. Strukov, G. S. Snider, D. R. Stewart, and R. S. Williams, “The missing memristor found.,” *Nature*, vol. 453, no. 7191, pp. 80–3, 2008.

[5] T M. D. Pickett, *et. al*, “Switching dynamics in titanium dioxide memristive devices,” *J. Appl. Phys.*, vol. 106, no. 7, p. 074508, 2009

[6] B. R. Field-, D. Ielmini, “Modeling the Universal Set / Reset Characteristics of Filament Growth,” *IEEE TED*, vol. 58, no. 12, pp. 4309–4317, 2011.

[7] M. Zangeneh,A. Joshi, “Design and Optimization of Nonvolatile Multibit 1T1R Resistive RAM,” *IEEE TVLSI*, pp. 1–14, 2013.

[8] S. Deora, *et. al*, “Ac Variability and Endurance Measurement for Resistive Switching Memories,” *IEEE TDMR*, vol. 14, no. 1, pp. 300–303, 2013.

[9] B. Chen, *et. al*, “Physical mechanisms of endurance degradation in TMO-RRAM,” *Proc IEDM*, pp. 12.3.1–12.3.4, 2011.

[10] A. Benoist, *et. al*, “28nm Advanced CMOS Resistive RAM Solution as Embedded Non-Volatile Memory,” *Proc IEEE IRPS*, pp. 5–9, 2014.

[11] R. Schemes, *et. al*, “A High-Speed 7 . 2-ns Read-Write Random Access 4-Mb Embedded Resistive RAM (ReRAM) Macro,” *IEEE JSSC*, vol. 48, no. 3, pp. 878–891, 2013.

[12] C.Jeffery, *et. al*, “Hierarchical Fault Tolerance for Nanoscale Memories,” *IEEE TNANO*, vol. 5, no. 4, pp. 407–414, 2006.

[13] P.Pouyan, E.Amat, A.Rubio; ”Process-Variability Aware Proactive Reconfiguration Technique for Mitigating Aging Effects in Nano-scale SRAM Lifetime” *Proc VTS*, pp 240-245, 2012.

[14] M.Conroy, “A Collection of Dice Problems”, pp 33-34, 2015.