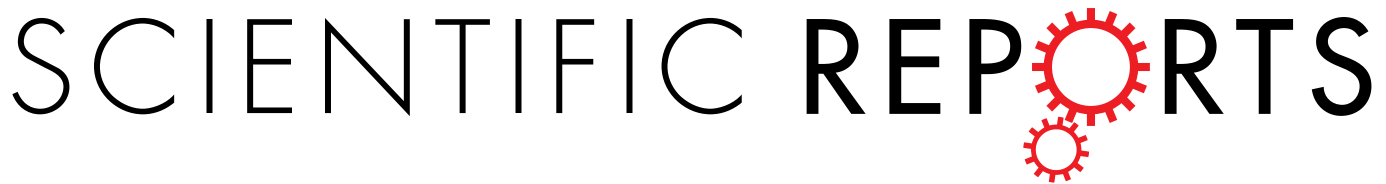

OPEN

Single-Readout High-Density Memristor Crossbar

M. A. Zidan[1], H. Omran[2], R. Naous[2], A. Sultan[2], H. A. H. Fahmy[3], W. D. Lu[1] & K. N. Salama[2]

Received: 30 May 2015
Accepted: 27 November 2015
Published: 07 January 2016

High-density memristor-crossbar architecture is a very promising technology for future computing systems. The simplicity of the gateless-crossbar structure is both its principal advantage and the source of undesired sneak-paths of current. This parasitic current could consume an enormous amount of energy and ruin the readout process. We introduce new adaptive-threshold readout techniques that utilize the locality and hierarchy properties of the computer-memory system to address the sneak-paths problem. The proposed methods require a single memory access per pixel for an array readout. Besides, the memristive crossbar consumes an order of magnitude less power than state-of-the-art readout techniques.

Current processor and memory technologies face design challenges that are related to the continuous scaling down of the minimum feature size anticipated by Moore's Law[1]. Moreover, conventional computing architecture is no longer an effective way to meet the demands of modern applications. An exigent need therefore exists to shift to new technologies at both architectural and device levels. Recently, the high-density, memristor-crossbar architecture has attracted attention in this regard. Memristor based resistive RAM is a promising candidate to replace Hard Disk Drive (HDD), DRAM, and flash memories[1–7]. Moreover, the high-density, memristive-crossbar is also a perfect candidate for neural bio-inspired computing[8–12]. Such applications are driven by recent advances in the fabrication of memristive devices[13–20].

The main advantage of a redox memristive array is its very high density[1,21], which entails that each memory cell occupies few nanometers. The array is simply built as a crossbar structure. Such simple assembly is inherently self-aligned and can be fabricated using only one or two lithography masks[13]. While the simplicity of the structure is its principal advantage, it is also the source of its main problem, namely the sneak-paths problem[21,22]. While accessing the array, current should flow only through the desired cell. However, nothing in the crossbar prevents the current from sneaking through other cells in the array as shown in Fig. 1a,b. This parasitic current ruins the reading and writing operations, and consumes a considerable amount of power.

The direct solution to the sneak-paths problem is to add a selector (gate) to each memory cell such as: MOS transistors[23], threshold devices[24], and complementary memristors[2,25]. In general, this comes at the expense of array density and the complexity of the fabrication process (cost per bit)[21]. Hence, the need arises to address the sneak-paths challenge using the typical gateless crossbar structure but with a quality similar to that of the gated arrays. Several techniques have been proposed for handling such an effect in gateless arrays, including multistage readout[5], multiport readout[3], unfolded arrays[26], engineering device nonlinearity[27], and grounded array[21]. However, these techniques either require extended accessing time, rely on a power-hungry accessing, reduce the density of the array significantly, or are even not a valid solution for practical size arrays[21].

In this work, we introduce single-stage readout techniques for the high-density gateless resistive arrays. The new method reduces the access time to the crossbar array significantly based upon the sneak-paths correlation analysis and the locality property of memory systems. The new readout adopts a very power efficient mode of access to the crossbar, guided by the study of the sneak-paths power consumption presented in this work. In addition, minimal control and sensing circuitry are required. Altogether, compared to previous work presented in the literature, we offer a faster and more power-efficient readout with a simple sensing mechanism.

Sneak-Paths Correlation Analysis

Sneak-paths impact the performance of a crossbar-based system in two ways. First, a considerable amount of undesirable energy is consumed as current sneaks throughout the array cells. Second, the sneak currents cannot be predicted because they are data dependent. Data stored in a memory array is naturally random[28], which leads

[1]Department of Electrical Engineering & Computer Science, University of Michigan, Ann Arbor, MI 48109, USA. [2]Electrical Engineering Program, King Abdullah University of Science and Technology (KAUST), Thuwal 23955, Saudi Arabia. [3]Electronics and Communications Department, Faculty of Engineering, Cairo University, Giza 12316, Egypt. Correspondence and requests for materials should be addressed to M.A.Z. (email: mzidan@umich.edu)

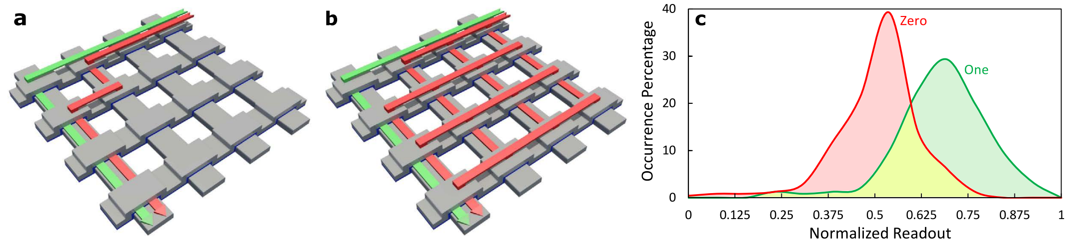

Figure 1. Crossbar with (**a**) a single sneak-path and (**b**) current sneaking throughout the whole array. (**c**) Histogram for the "One" and "Zero" values distributions as a result of the sneak-paths effect for a 256k array filled with NIST RAM images [28].

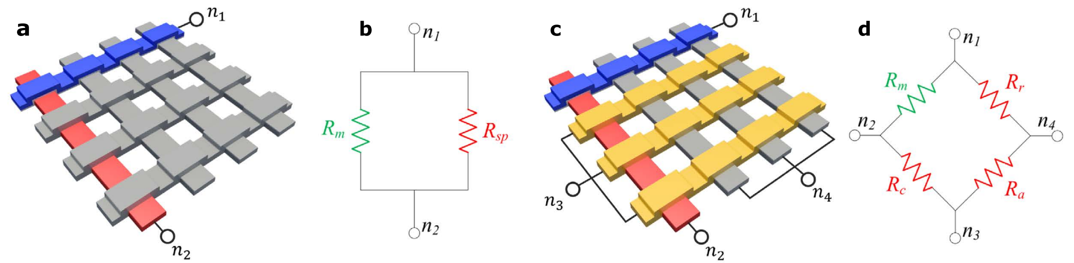

Figure 2. (**a**) Floating terminals accessing mode and (**b**) its equivalent circuit. (**c**) Connected terminals accessing mode and (**d**) its equivalent circuit.

to a random sneak-paths resistance. This is translated into having distributions that represent the "One" and "Zero" values rather than a single value. In addition, the magnitude of the sneak-current is typically higher than the current of the desired memory cell[3]; hence the distributions for the two binary values are highly overlapped, as shown in Fig. 1c. Direct memory readout is therefore not possible; thus, a power efficient sneak-paths immune readout is a necessity for a functional system.

One of the generally utilized properties of the sneak-paths current is its spatial correlation. Knowing the sneak-path noise value at one location of the crossbar helps to estimate the values at other correlated locations. Engineering such properties enables us to propose faster and more power-efficient readout techniques for the resistive crossbar memories. In general, a crossbar can be accessed using two modes: "floating terminals" and "connected terminals", as shown in Fig. 2. In the first approach, the selected array terminals are kept floating. On the other hand, in the "connected terminals" approach the selected rows and columns are connected to two common nodes. The two extra nodes can be used as access terminals to the array[3], or to enforce a bias voltage[29]. This allows for better control of the sneak-paths behavior and yields a more usable equivalent circuit. In such case, the sneak-paths are represented by three lambed resistances ('R_r', 'R_a' and 'R_c') as shown in Fig. 2d. Understanding the correlation of these elements over the crossbar facilitates a better handling of the sneak-paths noise. For instance, 'R_r' is a parallel combination of all the desired row cells apart from the desired one; it is given by, $R_r^{-1} = \sum_{i=1}^{L-1} R_{xi}^{-1}$, where '$R_x$' is the resistance of a one-row cell, and 'L' is the array length. The row cell resistance can be either 'R'_{on}' or 'R'_{off}' which are the ON and OFF resistance of the device under '$V_{n1} - V_{n4}$' voltage drop respectively. The row resistance can be rewritten as,

$$R_r = \frac{R'_{on} R'_{off}}{(L-1)R'_{on} + N_{on}(R'_{off} - R'_{on})} \tag{1}$$

where 'N_{on}' is the number of ON cells within the accessed row not counting the accessed cell itself. The remaining two sneak-path components (R_c and R_a) have similar expressions. Although, in the case of biasing the unused array terminals the sneak-paths component'R_a' is shorted out. It should be noted that although the metal line resistances are not included in the equivalent circuit for sake of simplicity, they have been fully considered in the simulations carried out in this work.

For practical array size, the values of 'R_r' and 'R_c' are almost constant over the same row or column respectively. For instance, the sneak-paths row resistances found at two different locations in the same row have all cells in common except the two cells that are swapped because of the accessed locations. For devices with a large OFF/ON ratio, the relative change in the sneak-paths row resistance is given by,

$$\frac{\Delta R_r}{R_r} \lesssim \left| \frac{\rho}{L + (N_{on})\rho} \right| \tag{2}$$

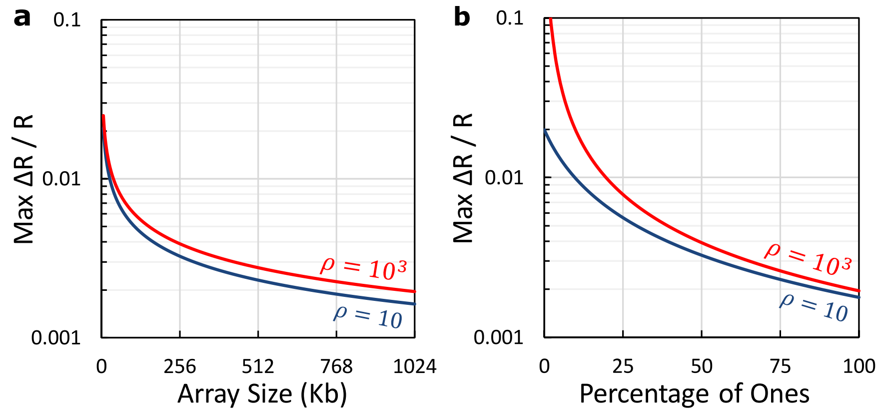

Figure 3. Maximum change in resistance related to its original value versus (**a**) the array size with balanced ones and zeros, and (**b**) percentage of ones for an array of a size of 256 kb. '$\Delta R/R$' represents both the '$\Delta R_r/R_r$' and the '$\Delta R_c/R_c$'.

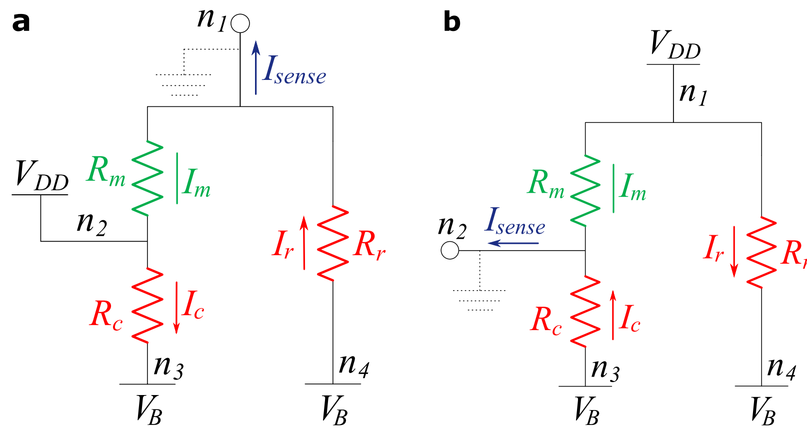

Figure 4. Equivalent circuit for the "connected terminals" accessing mode when the row and column terminals are forced to the same voltage and the sense circuit is connected to either the desired (**a**) row or (**b**) column.

where 'ρ' is the OFF/ON ratio of the used device. The maximum relative change in the row resistance versus the array size for a balanced number of zeros and ones is plotted in Fig. 3a. The figure shows that as the array size increases the effect of a single bit swap diminishes. The other parameter that affects $\Delta R/R$ is the number of ones (per row or column), as given by (2). Figure 3b shows that the maximum relative change of sneak-paths resistance is still small while the percentage of ones per row/column is swept. Hence, 'R_r' is almost constant over a given row and 'R_c' is almost constant over a given column. It should be noted that, given the randomness of the data, 'R_r' and 'R_c' are considered two independent random variables.

Adaptive-Threshold Readout Techniques

The sneak-paths correlation property can be effectively utilized in case of sequential reading for the stored data on a memory array. The good news is that this is the typical memory access scheme in computer systems. Because of the memory-locality property, data is transferred and shared between different memory layers as a block of contiguous bits, rather than in random bits or words[30]. This locality property is of help only if the knowledge gained from reading a single bit can be adopted in reading its neighborhoods. This is true for the "connected terminals" crossbar, where the values of 'R_r' and 'R_c' can be safely shared over the same row or column, respectively, as discussed in the previous sections. This is equivalent to defining an adaptive threshold that changes at each new row readout, which can be achieved with the aid of the "connected terminals" crossbar.

The generic "connected terminals" circuit model shown in Fig. 2d can be simplified for the case of 'V_B' terminals bias. Terminals 'n_3' and 'n_4' are connected to 'V_B', and terminals 'n_1' and 'n_2' are connected to 'V_{DD}' and virtual ground. This can be done with two different implementations as shown in Fig. 4. Using a virtual ground sensing circuit forces all of the array elements to have a defined voltage drop independent of the data stored in the array. The desired cell experiences a full 'V_{DD}' voltage drop, while the sneak-paths components of 'R_r' and 'R_c' have a voltage drop of '$V_{DD} - V_B$'. Because of the device saturation nonlinearity, the full voltage drop on the desired cell makes the magnitude difference between its ON and OFF states much larger than any error introduced by sharing 'R_r' or 'R_c' over a segment. While both of 'R_r' and 'R_c' drain parasitic sneak-current, the current leak through only

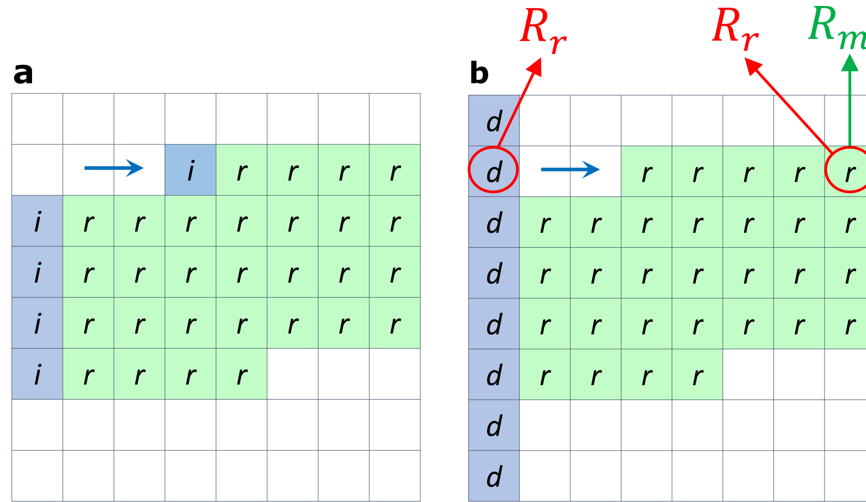

Figure 5. (**a**) Array accessing sequence, where the initial bit per row/column is accessed 'n' times, while the rest of the bits in the same row/column are accessed once. (**b**) The accessing sequence in case of using predefined "dummy bits", where all of the bits of the array are accessed in a single stage fashion. d: dummy bit, i: initial bit, and r: regular bit.

one of them affects the correctness of the readout operation. When the read circuit is connected to node 'n_1', as shown in Fig. 4b, the sense current (I_{sense}) is defined as,

$$I_{sense} = I_m + I_r \tag{3}$$

where 'I_m' is the desired current and 'I_r' is the row sneak current component. Sensing from node 'n_2' swaps the locations and the role of 'R_r' and 'R_c' in the circuit, as shown in Fig. 5a. The sense current is shifted from its desired value by the sneak-current of the row or the column. However this shift is constant within a given row or column, based on the connection orientation.

Initial Bits Readout

Each bit generally has two unknowns: 'R_m' and 'R_r' (or 'R_c'). Without adopting sneak-paths correlation and locality, multiple access stages are needed to estimate the bit value. However, a faster readout can be achieved by categorizing the bits into two types: the "initial bits," which are the first bits accessed in a given column, and "regular bits," which are any other bits in the array. To estimate the value of the "initial bit," two unknowns need to be solved, namely the desired resistance (R_m) and the row sneak resistance (R_r). However, the remaining bits in the row share the same 'R_r' value, and 'I_r' is treated as the significant sneak-path component for a given row. Any of the readout techniques presented in the literature can be used to estimate the "initial bit"[21]. These "initial bits" readout dictates the threshold used for the remaining bits in that row. Figure 5a shows the readout sequence for the array when"initial bits" strategy is adopted. Therefore, the first (initial bit) could be any bit in the array that requires 'n' stages of reading. The rest of the bits in the same row are then accessed in sequence, only one time for each. Reading from the next row requires a new "initial bit", which in this case is the first bit in the row, as shown in Fig. 5a. The same sequence is followed until the fetched data block for the cache is completed, i.e., each row contains one "initial bit", and the rest of the bits are accessed in a single stage fashion. For a contiguous block of data readout using the "initial bits" technique, the proposed readout procedure is given as follows:

Case 1: The first accessed bit in the row 'i' (the initial bits),

- Use a multi stage readout technique to estimate the desired cell current $(I_m^{i,j})$ and the row sneak-current component (I_r^i).

Case 2: Accessing the rest of the bits in the same row,

- Access the desired cell for a single time to estimate its value, where '$I_m^{i,j} = I_{sense}^{i,j} - I_r^i$'.

where 'i' and 'j' are the desired row and column respectively. It should be noted that in the case of sensing from 'n_1' data is accessed in a column-wise rather than row-wise scheme.

The readout circuitry for the "initial bit" is made of two parts: a virtual-ground ADC for the current sensing, and a digital processing circuitry for calculating the "initial bit" parameters and do the threshold comparisons. Typically, a single readout circuitry is needed per memory array. This does not impact the whole memory density as presented in previous works[3].

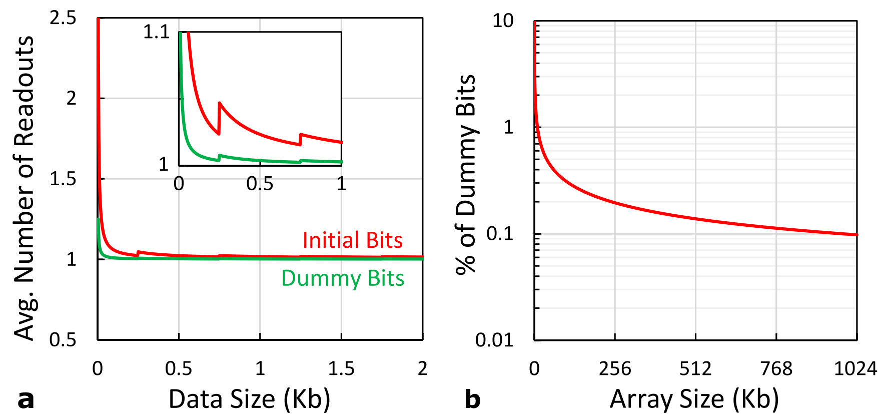

Figure 6. (**a**) The average number of readouts versus the fetched data size for the adaptive threshold techniques, where the first accessed bit is the middle of a row. (**b**) The percentage of "dummy bits" versus the array size.

Predefined Dummy Bits Readout

A more time efficient way to estimate the adaptive threshold is to add "dummy bits" with predefined value to the array. The general concept of adding predefined bits to an array for sneak-paths estimation is presented in[31]. In our case, for a "dummy bit" the value of 'R_m' is known in advance, and a single readout is needed to estimate the value of 'R_r'. This estimated 'R_r' value is reused with the other bits in the same row. A single readout is required in this case to estimate the remaining unknown (R_m). This value is used for the rest of the bits in the same row. The "dummy bit" can be organized in several ways, given that each row contains a single bit. Figure 5b shows a possible organization of dummy bits that is suitable for a row-wise readout analogy. For a contiguous block of data readout using the "dummy bits" technique, the proposed readout procedure is given as follows:

Case 1: Accessing the "dummy bit" of row 'i',

- Estimate the sneak-path row component using a single array access, where '$I_r^i = I_{sense}^{i,0} - I_{dummy}$'.

The current '$I_{sense}^{i,0}$' can be used without any processing, since the values of 'I_{dummy}' is a DC shift that can be compensated in comparison process.

Case 2: Accessing the rest of the bits in the same row,

- Access the desired cell for a single time to estimate its value, where '$I_m^{i,j} = I_{sense}^{i,j} - I_r^i$'.

where 'i' and 'j' are the desired row and column respectively. The dummy current 'I_{dummy}' is known in the design time, where it can be 'I_{on}' or 'I_{off}' depending on which value is used to be stored in the dummy cells. Moreover, a dummy cell could be just a reference static resistor rather than a memristor, since there is no need to write it after the array fabrication.

The "dummy bits" technique adds a small overhead to the readout process, as a "dummy bit" needs to be accessed a single time (in comparison to 'n' times for an "initial bit"). However, for practical size arrays of 256 k size or more, the average number of array accesses per bit that occurs when fetching a block of data from memory is almost one for both methods. Figure 6a shows the average number of readouts per memory bit, where the overhead is shared over "regular bits", versus the fetched data size. It also illustrates how the average number of readouts converges to one very fast. The ripples in the curve occur because that start reading from a new row adds extra overhead of an "initial bit" or a "dummy bit". It should be noted that the typical cache line is 0.5 kb (64 bytes), where multiple lines are fetched from memory in sequence based on the cache policy. This value is much larger in the case of RAM fetching from HDD. While the "dummy bits" technique exhibits a better behavior, it comes at a small cost to the effective area of the array, as "dummy bits" are not used to store real data. This negligible overhead is shown in Fig. 6b.

The readout circuitry for the "dummy bits" technique can be implemented in two ways. The first approach is to use an analog circuit for current sensing and a simple digital circuit for comparisons and estimation, as discussed in the "initial bits" readout. Typically, most of the readout circuit area in this methodology is consumed by the conversion of the data from one domain to the other using ADCs[3]. A more area efficient implementation is to adopt a totally analog compensated readout circuit, as presented in previous work[23]. In this approach, the current of a "dummy cell" is sampled on a first capacitor, and the sensed current from each desired cell is sampled on a second one in sequence. Comparison between the two capacitor voltages leads to estimating the stored data in the desired memory cells[23].

Variability

Variability is a challenge that faces resistive-memory readout techniques. In general, two types of variability issues face memristor-based memory. The first is fabrication variability, in which fabricated cells have a distribution of ON and OFF values, rather than two unique states. The other source of variability is operation variability, in which the device parameters change stochastically or with aging. In the proposed readout, a memristor device variability

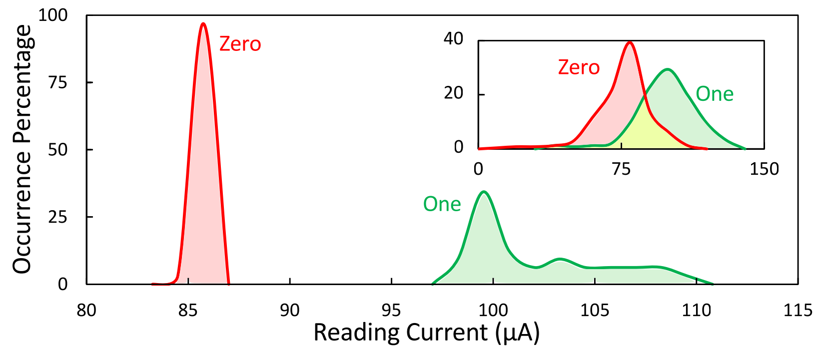

Figure 7. Histogram of the readout current for reading from a single row using the proposed readout technique. Inset: The original histogram of the readout as achieved without applying the adaptive threshold readout.

can affect three types of cells. The first type is the "normal bits", where any change in the cell properties can ruin its readout alone and impact the sneak-paths estimation. This is because the presented sneak-paths estimation techniques do not assume any properties of the "normal" cells, instead a group of parallel resistances whose effect as sneak-paths is probed for each new row readout. However, variability in the "normal bits" has a secondary impact on the read margins of the system, since it widens the distributions representing the possible ON and OFF values. The change in the read margins is defined as,

$$\Delta_{RM} \leq 2\delta_r \tag{4}$$

where δ_r is the maximum absolute shift in a cell value due to variability.

The second type of bit, which are "initial bits" does not suffer from any variability issues, because multi-stage readout techniques used to access such bits typically do not assume any properties of the probed cell. Such methods read and write the cell under probe multiple times during each readout to define a local threshold for the cell. Therefore, adopting multistage readout for accessing the "initial bits" makes the system less vulnerable to variability yet yields a very high total throughput. The last type of bits are "dummy bits". Variability can impact the sneak-paths estimation since the dummy bit is used as a reference for the other cells in the row. Typically, variability in a "dummy bit" results in a threshold shift, which results in a decrease in read margins defined as

$$\Delta_{RM} \leq \delta_r^d \tag{5}$$

where δ_r^d is the maximum absolute shift in a "dummy cell value" due to variability. The variability effect caused by a "dummy cell" can be reduced by storing the less variable memory stage (ON or OFF) in a dummy bit, or by using static resistance as a dummy cell rather than a memristor. Moreover, in the case of devices with high OFF resistance, storing "Zeros" in the dummy cell makes the probed current from its location much closer to the sneak-current rather than the cell current.

Results and Discussions

In order to evaluate the validity and efficiency of crossbar readout techniques, an accurate simulation platform that includes different crossbar non-idealities is needed. To achieve this goal, we employ a Python script that creates SPICE netlists for realistic size arrays and sweeps different parameters and data patterns by calling HSPICE or Cadence APS iteratively. The test array can be filled with any predefined, random, or realistic workload as NIST RAM images[28]. Finally, the Python script braces the SPICE output files to collect the data of interest and tabulate it. We used a crossbar parasitic resistance value of 5 Ω per cell[3] and included the effect of the switching row and column circuitry in all of the simulations in this work. For the memristor device, we adopted a bipolar device model proposed for memory operations[5]. Finally, it should be noted that resistive RAMs are built in the same hierarchy and structure of DRAMs, where subarrays of size up to 256 kb are used to reduce the capacitive loading of the metal lines[32]. Hence, we use an array size up to 256 kb for simulations and comparisons with this work.

Error Free Readout. To verify the proposed concept, we simulated the readout operation at different locations of a 256 kb array of various NIST RAM images. In the first case, the readout locations are distributed over the array while in the second case all the readouts are made for cells in the same column. Figure 7 shows the histogram of the sensed read current for the two cases. The results indicate that the distributions of reading "One" and reading "Zero" are highly overlapped, and that it is not possible to define a threshold to distinguish between the two binary cases, as shown in Fig. 7 (inset). However, for a given row or column, reading from different locations reveals a clear separation between the distribution of ones and zeros, as shown in Fig. 7. This verifies our proposed readout scheme, in which an adaptive threshold is defined for each column (or row) as discussed earlier. The simulation results show that a simple comparator is required to differentiate between "One" and "Zero" states.

Crossbar Power Consumption. Undesirable sneak-paths power consumption is not avoidable in high-density gateless arrays. However, it can be reduced by utilizing devices with nonlinear saturation behavior.

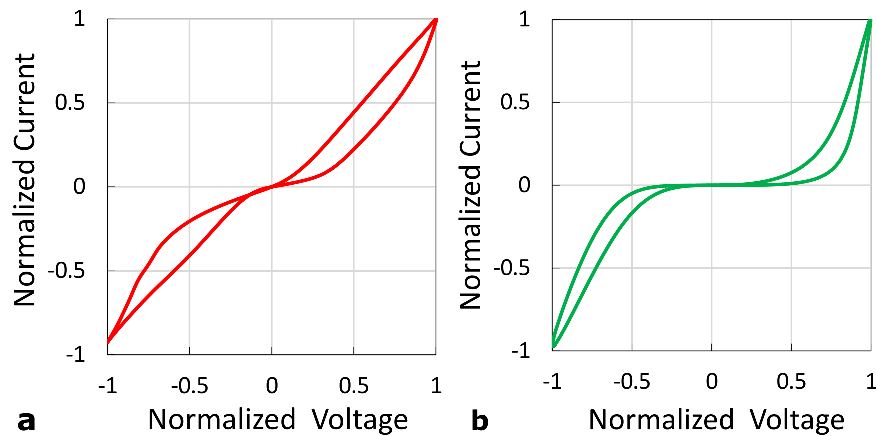

Figure 8. Measured '*i*-*v*' hysteresis of two of our fabricated devices showing (**a**) low and (**b**) high saturation nonlinearity.

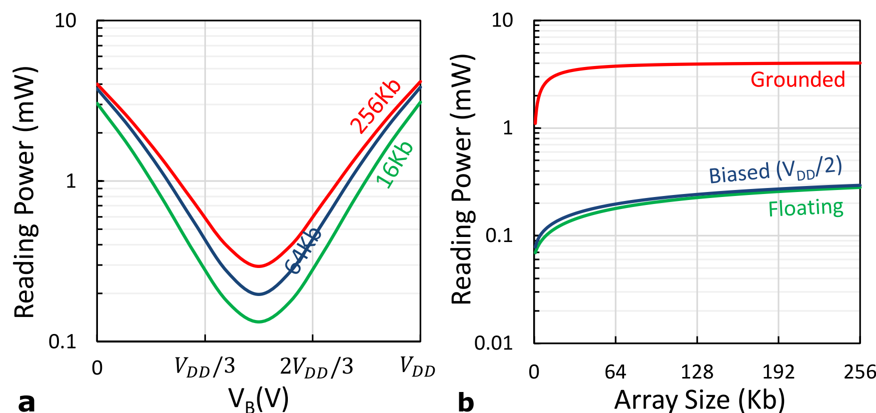

Figure 9. Reading power consumed by a biased-terminals crossbar filled with checkered data pattern versus (**a**) the bias voltage and (**b**) the array size.

	Error Free Readout	# of Reads	# of Writes	Locality Needed	Readout Circuit*	Read Power [mW]	FoM [*Tbit/cm²W*]
Multi-Stage[5]	Yes	3	3	No	ADC + Comp	7	91
Multi-Port[3]	Yes	3	0	No	ADC + Comp	2.1	304
Grounded Rows & Cols[26]	No	1	0	No	VG + Comp	4	160
This Work (initial bits)	Yes	1.05**	< 0.01	Yes	ADC + Comp	**0.293**	**2184**
This Work (dummy bits)	Yes	1.01**	0	Yes	VG + Comp	**0.291**	**2195**

Table 1. Comparison between state-of-the-art gateless readout techniques for a subarray of size 256 kb. *ADC: Analog-to-Digital Converter, Comp: Comparator, and VG: Virtual-Ground. **The number of reads is calculated for the case of 16 bytes being fetched from the array in sequence.

Figure 8 shows the '*i*-*v*' hysteresis of two of our fabricated devices. The second device shows higher saturation nonlinearity than the first one. Reducing the voltage applied to such devices by fifty percent can increase its saturation resistance up to two orders of magnitude[27]. This is a very attractive property since a sneak path is made of a series of memristor devices, where a sub-voltage is dropped on each of them. In the "connected terminals" structure, the device nonlinearity can be enforced by biasing the unused terminals to sub-read voltage. In such case, the very small 'R_a' is shorted out, and the nonlinearity of the other terminals efficiently utilized. Figure 9a shows that the optimal selection is made by biasing the unused terminals voltage to be $V_B = V_{DD}/2$. The power consumption of this method is almost the same as the baseline "floating terminals", as shown in Fig. 9b. The figure also shows the great power-saving of the "connected terminals" while comparing it with the power hungry "grounded

terminals" technique. It should be noted that power consumption saturates for larger array sizes because of the crossbar metal lines[3].

Figure-of-Merit. In general, the presented technique offers a sneak-paths immune readout that is more power efficient and faster than the state-of-the-art crossbar accessing techniques that are presented in the literature. Table 1 shows a detailed comparison of the various gateless techniques that can provide an error-free readout. The different methods are compared based on a figure-of-merit (FoM), which is defined as,

$$FoM = \frac{\text{Array Density}}{\text{Array Read Power}} \tag{6}$$

where the proposed technique shows the best FoM.

Conclusion

Taking advantage of memory locality and the sneak-paths correlation yields a fast and power efficient readout technique. Unlike other techniques, the proposed method achieves the theoretical limit of a single memory access per pixel for an array readout at a fraction of the power of state-of-the-art readout techniques. The presented adaptive-threshold readout is 7 to 24 times better than the other gateless techniques presented in the literature, based on the density-power figure-of-merit. In addition, the new sneak-paths immune technique requires minimal hardware to distinguish between the memory data values.

References

1. International Technology Roadmap for Semiconductors, URL: http://www.itrs.net/. (2013) (Date of access: 11/02/2015).
2. Linn, E., Rosezin, R., Kügeler, C. & Waser, R. Complementary resistive switches for passive nanocrossbar memories. *Nature materials* **9,** 403–406 (2010), doi: 10.1038/nmat2748.
3. Zidan, M. A., Eltawil, A. M., Kurdahi, F., Fahmy, H. A. H. & Salama, K. N. Memristor multi-port readout: A closed-form solution for sneak-paths. *IEEE Transactions on Nanotechnology* **13,** 274–282 (2014), doi: 10.1109/TNANO.2014.2299558.
4. Van Den Hurk, J., Havel, V., Linn, E., Waser, R. & Valov, I. Ag/GeSx/Pt-based complementary resistive switches for hybrid CMOS/nanoelectronic logic and memory architectures. *Scientific Reports* **3,** 2856 (2013), doi: 10.1038/srep02856.
5. Vontobel, P. O. *et al.* Writing to and reading from a nano-scale crossbar memory based on memristors. *Nanotechnology* **20,** 425204 (2009).
6. Eshraghian, K. *et al.* Memristor MOS content addressable memory (MCAM): hybrid architecture for future high performance search engines. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* **19,** 1407–1417 (2011), doi: 10.1109/TVLSI.2010.2049867.
7. Cheng, K.-T. T. & Strukov, D. B. 3D CMOS-memristor hybrid circuits: Devices, integration, architecture, and applications. In *ACM International Symposium on International Symposium on Physical Design (ISPD '12),* 33–40, California, USA (2012, April), doi: 10.1145/2160916.2160925.
8. Park, S. *et al.* Electronic system with memristive synapses for pattern recognition. *Scientific Reports* **5,** 10123 (2015), doi: 10.1038/srep10123.
9. Li, Y. *et al.* Activity-dependent synaptic plasticity of a chalcogenide electronic synapse for neuromorphic systems. *Scientific Reports* **4,** 4906 (2014), doi: 10.1038/srep04906.
10. Prezioso, M. *et al.* Training and operation of an integrated neuromorphic network based on metal-oxide memristors. *Neature* **521,** 61–64 (2015), doi: 10.1038/nature14441.
11. He, W. *et al.* Enabling an integrated rate-temporal learning scheme on memristor. *Scientific Reports* **4,** 4755 (2014), doi: 10.1038/srep04755.
12. Jo, S. H. *et al.* Nanoscale memristor device as synapse in neuromorphic systems. *Nano letters* **10,** 1297–1301 (2010), doi: 10.1021/nl904092h.
13. Xia, Q., Yang, J. J., Wu, W., Li, X. & Williams, R. S. Self-aligned memristor cross-point arrays fabricated with one nanoimprint lithography step. *Nano letters* **10,** 2909–2914 (2010), doi: 10.1021/nl1017157.
14. Kim, K. M., Park, T. H. & Hwang, C. S. Dual conical conducting filament model in resistance switching TiO2 thin films. *Scientific Reports* **5,** 7844 (2015), doi: 10.1038/srep07844.
15. Jo, S. H., Kim, K.-H. & Lu, W. High-density crossbar arrays based on a Si memristive system. *Nano Letters* **9,** 870–874 (2009), doi: 10.1021/nl8037689.
16. Gambardella, A., Prezioso, M. & Cavallini, M. Tunnel conductivity switching in a single nanoparticle-based nano floating gate memory. *Scientific Reports* **4,** 4196 (2014), doi: 10.1038/srep04196.
17. Gergel-Hackett, N., Tedesco, J. L. & Richter, C. A. Memristors with flexible electronic applications. *Proceedings of the IEEE* **100,** 1971–1978 (2012), doi: 10.1109/JPROC.2011.2158284.
18. Yang, J. J. *et al.* High switching endurance in TaOx memristive devices. *Applied Physics Letters* **97,** 232102 (2010), doi: 10.1063/1.3524521.
19. Mikheev, E., Hoskins, B. D., Strukov, D. B. & Stemmer, S. Resistive switching and its suppression in Pt/Nb: SrTiO3 junctions. *Nature Communications* **5,** 3990 (2014), doi: 10.1038/ncomms4990.
20. Li, L. *et al.* Anatomy of vertical heteroepitaxial interfaces reveals the memristive mechanism in Nb2O5-NaNbO3 thin films. *Scientific Reports* **5** (2015), doi: 10.1038/srep09229.
21. Zidan, M. A., Fahmy, H. A. H., Hussain, M. M. & Salama, K. N. Memristor-based memory: The sneak paths problem and solutions. *Microelectronics Journal* **44,** 176–183 (2013), doi: 10.1016/j.mejo.2012.10.001.
22. Ghofrani, A., Lastras-Montano, M. A. & Cheng, K.-T. T. Toward large-scale access-transistor-free memristive crossbars. In *Asia and South Pacific Design Automation Conference (ASP-DAC'15),* vol. 563–568, Chiba/Tokyo, Japan (2015, January), doi: 10.1109/ASPDAC.2015.7059067.
23. Zidan, M. A., Omran, H., Sultan, A., Fahmy, H. A. H. & Salama, K. N. Compensated readout for high density MOS-gated memristor crossbar array. *IEEE Transactions on Nanotechnology* **14,** 3–6 (2015), doi: 10.1109/TNANO.2014.2363352.
24. Zhou, J., Kim, K.-H. & Lu, W. Crossbar RRAM arrays: Selector device requirements during read operation. *IEEE Transactions on Electron Devices* **61,** 1369–1376 (2014), doi: 10.1109/TED.2014.2310200.
25. Yang, Y., Mathew, J., Ottavi, M., Pontarelli, S. & Pradhan, D. Novel complementary resistive switch crossbar memory write and read schemes. *IEEE Transactions on Nanotechnology* **14,** 346–357 (2015), doi: 10.1109/TNANO.2015.2394450.
26. Manem, H., Rose, G., He, X. & Wang, W. Design considerations for variation tolerant multilevel CMOS/nano memristor memory. In *Greatlakes Symposium on VLSI (GLSVLSI'10)*, 287–292, Providence, USA (2010, May), doi: 10.1145/1785481.1785548.
27. Yang, J. J. *et al.* Engineering nonlinearity into memristors for passive crossbar applications. *Applied Physics Letters* **100,** 113501 (2012), doi: 10.1063/1.3693392.

28. NIST Basic Memory Images URL: http://www.cfreds.nist.gov/mem/Basic_Memory_Images.html. (Date of access: 21/12/2014).
29. Liang, J., Yeh, S., Wong, S. S. & Wong, H.-S. P. Effect of wordline/bitline scaling on the performance, energy consumption, and reliability of cross-point memory array. *ACM Journal on Emerging Technologies in Computing Systems (JETC)*. **9,** 9 (2013), doi: 10.1145/2422094.2422103.
30. Jacob, B., Ng, S. & Wang, D. *Memory systems: cache, DRAM, disk* (Morgan Kaufmann Pub, 2007).
31. Naous, R., Zidan, M. A., Sultan-Salem, A. & Salama, K. N. Memristor based crossbar memory array sneak path estimation. In *International Workshop on Cellular Nanoscale Networks and their Applications (CNNA'14),* 1–2, Notre Dame, USA (2014, July), doi: 10.1109/CNNA.2014.6888656.
32. Baker, R. J. *CMOS: Circuit Design, Layout, and Simulation* (Wiley-IEEE Press, 2011).

Author Contributions

M.Z. conceived the idea, introduced the mathematical analysis, built the simulation platform, and analyzed the results. M.Z., A.S, and R.N. introduced the usage of predefined bits. M.Z. and H.O. improved the readout techniques. M.Z. and W.L. Studied the variability effect and the readout circuitry. M.Z., W.L. and K.S. wrote the manuscript. K.S. and H.F. supervised the research. M.Z., H.O., R.N., A.S., H.F., W.L. and K.S. contributed into developing the idea, reviewed the manuscript and discussed the results.

Additional Information

Competing financial interests: The authors declare no competing financial interests.

How to cite this article: Zidan, M. A. *et al.* Single-Readout High-Density Memristor Crossbar. *Sci. Rep.* **6**, 18863; doi: 10.1038/srep18863 (2016).

This work is licensed under a Creative Commons Attribution 4.0 International License. The images or other third party material in this article are included in the article's Creative Commons license, unless indicated otherwise in the credit line; if the material is not included under the Creative Commons license, users will need to obtain permission from the license holder to reproduce the material. To view a copy of this license, visit http://creativecommons.org/licenses/by/4.0/