The Missing Applications Found: Robust Design Techniques and Novel Uses of Memristors

Marco Ottavi, Vishal Gupta, Saurabh Khandelwal**
Shahar Kvatinsky***, Jimson Mathew*, Eugenio Martinelli, Abusaleh Jabir**
University of Rome "Tor Vergata", Italy
*Indian Institute of Technology Patna, India
**Oxford Brookes University, Oxford, UK
***Technion - Israel Institute of Technology, Haifa, Israel

***Abstract*—Resistive memory, also known as memristor, is an emerging potential successor to traditional CMOS charge based memories. Memristors have also recently been proposed as a promising candidate for several additional applications such as logic design, sensing, non-volatile storage, neuromorphic computing, Physically Unclonable Functions (PUFs), Content-addressable memory (CAM) and reconfigurable computing. In this paper, we explore three unique applications of memristor technology based implementations, specifically from the perspective of sensing, logic, in-memory computing and their solutions. We review solar cell health monitoring and diagnosis, describe the proposed solutions, and provide directions in memristive gas sensing and in-memory computing. For the gas sensor application, in order to determine the number of memristors to ensure a certain level of accuracy in sensitivity, a technique to optimize the sensor array based on an acceptable sensitivity variation and minimum sensitivity margin is presented. These "out-of-the-box" emerging ideas for applications of memristive devices in enhancing robustness and, at the same time, how the requirements of robust design are enabling unconventional use of the devices. To this end, the papers considers some examples of this mutual interaction.**

***Index Terms*—memristor, logic design, resistive RAM, gas sensor.**

I. Introduction

Memristors (a portmanteau for Memory Resistors) were initially theorized by Leon Chua in 1971 [1] based on symmetry considerations in circuits theory, to link magnetic flux Φ and charge q. In 2009 a first physical implementation based on TiO_2 of the memristor has been proposed by researchers at HP [2]. First fabricated memristive device is mainly composed of TiO_2, and it is divided into two regions: one containing TiO_2 without any doping called the "undoped region", and the other containing O_+^2 vacancies called the "doped region'. The doped region has a width of w (which is considered as the "state variable", as its value determines the device logic state) and has low resistance. The undoped region, on the contrary, has high resistance. As a consequence, the total resistance of the device is the combination of those two resistors. When the doped region width w reaches the full length D of the device, the memristor will have the lowest resistance R_{ON} and this state is called state-1 [2]. On the other hand, the undoped region that has a with of D will lead to the highest resistance R_{OFF} of the device, implying that memristor is in state-0. Figure 1 shows the I-V characteristics memristor. It can be seen that state ON has low resistance state (high slope) and state OFF has high resistance state (low slope). Voltages 'VSET' and 'VRESET' are the threshold voltages to switch memristor to state 'ON' and 'OFF' respectively. For example, if the memristor is initialized as state 'OFF', a voltage 'VSET' can switch it to state 'ON', if memristor is on state 'ON', a voltage 'VRESET' can turn it to state 'OFF'. The device can be in other possible states as well, that is in between 'ON' and 'OFF' based on the amount current that passes through it. While some doubt have been raised about the actual connection between the HP model and Chua's [3] nonetheless memristors (and in general Resistive Devices such as Resistive RAMs) are raising a growing interest in the research community and several applications have been proposed. Clearly the first straightforward application to memristors was to make them the base for new Non-Volatile-Memory devices [2] [4] [5]. Also Neuromorphic Computing has been considered as a memristor change of resistance could be leveraged to implement programmable synaptic weights in an Artificial Neural Network [6].

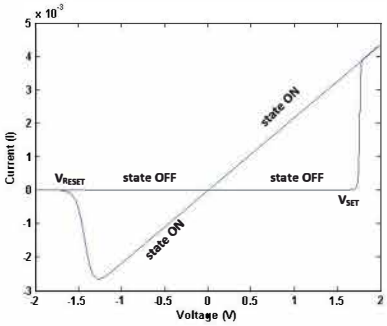

Fig. 1: I-V Characteristic of Memristor

Memristors have, however, unexplored potential to shift the paradigm for several other applications. Recent works from the authors show how memristors could be embedded in a solar cell array [7], furthermore it can be used for sensing in memroy [8], and finally in the case of a new Logic in memory paradigm [9].
This paper is a summary of the research that was carried out and mainly involves looking into three application areas

using this device such as solar cell array health monitoring, memristive gas sensing and in-memory computing and assessing their ability towards doing so. The rest of this paper is organized as follows: Section II reviews methods to detect faults in a solar cells array using memristors, Section III explores how memristive sensor arrays could be used for gas sensing. A design paradigm that allows implementing logic-in-memory with memristors is discussed in Section IV. Finally, Section V discusses an outlook of conclusion and future work.

II. Memristors for Solar Cell Array Health Monitoring

To monitor the status of a PV array we capture the state change behavior of TiO_2 based memristor device. This TiO_2 based memristor could be embedded inside a photovoltaic array (Dye-sensitized solar array [10]) because the fabrication process is identical for both photovoltaic cell and memristor device. This time analysis of memristor device provides valuable information for behavior of the system under measurement both in terms of power generated in normal conditions, and for fault detection. A photovoltaic (PV) cell or solar cell is a device that converts solar energy (photons) in to DC supply. The circuit for an ideal solar cell modeled by a current source with a parallel connection of diode, but for the practical device a shunt resistance and a series resistance component are added. Fig. 2 shows the equivalent circuit for the DSC solar cell.

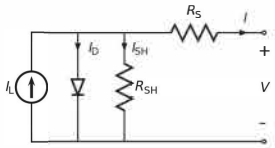

Fig. 2: Equivalent circuit of a PV cell.

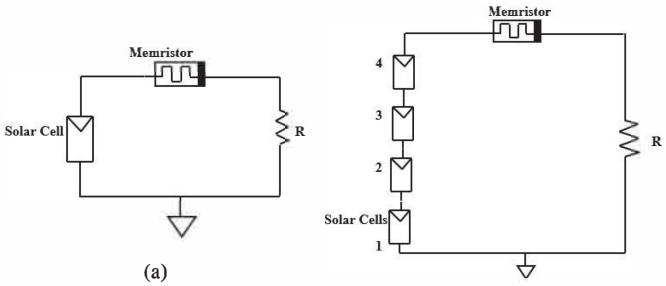

Fig. 3: Schematic of experimental circuit with memristor element: (a) single solar cell setup. (b) series configuration of four solar cells.

As an example of this approach, in the following, we analyze various configuration of solar cell arrangements: single and four series connected solar cells with a memristor element. The configurations are shown in Fig. 3a and 3b respectively where is added a small wire resistance. In our simulations, a healthy dye solar cell has open circuit voltage of 708.67mV and the short circuit current of 15mA. Behaviour of the memristor were characterized by switching the device from high impedance state to low impedance state. Initially, we set our TiO_2 based memristor into the high resistance state, then we triggered this by the dye-sensitized solar cell configuration. In Fig. 4a, curve 1 (single solar cell setup) and curve 2 (configuration of four series connected solar cells) show the variation in current flow through the memristor with respect to the time. This initial variation in current with respect to time represent the switching mechanism for the device from high resistance state to the low resistance state. After certain time interval current goes saturate which shows the minimum current required to switch the memristor from one stable state to the another stable state. The time taken by the memristor device is 1.98ms and 0.29ms for the single solar cell setup and the configuration of four series connected solar cells respectively. The output current of the series connected cell is higher than the single cell configuration, which also shows the faster switching as compared to the single solar cell setup.

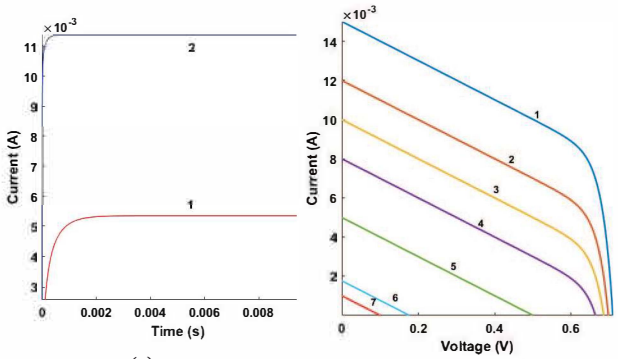

Fig. 4: (a) Simulation results: curve 1 is for single solar cell setup and curve 2 for series configuration of solar cell setup. (b) Effect of solar cell degradation and faulty cell on I-V Curve.

The output power and current of the solar cell is related to the irradiance, (i.e. the amount of solar power per unit are that is shining on the array itself) and this is modeled by the the I_L value in the Fig. 2. Therefore, while in the following we will use this mechanism to detect degradation and faults in the array, it is worth noticing that the memristor switch time could be used to provide an runtime estimation of the power output of the array in nominal operating conditions.

Degradation of solar power plays a crucial role in the PV system. Solar cell experience degradation due to the unavoidable circumstances like UV exposure, thermal cycling, damp heat, humidity freeze and weather cycle. This degradation may cause faults in the system or responsible for the system failure. We consider the family of defects in a cell that results in one of the solar cells is stuck permanently not healthy or partially healthy and full degradation or partial degradation cell voltage/current generated [11]. These faults can be categorized are as ground fault, open-circuit (OC) and short circuit (SC) fault in the solar cell. We also

TABLE I: Time Analysis for the state change behaviour of memristor device for various source of current generators.

Solar cell current generator (mA)	Solar cell power generation (W)	Time (ms)
0	0	Infinite
1	24.975μ	Infinite
1.765	77.800μ	38.76
5	624.375μ	5.97
8	1.598m	2.99
10	2.496m	2.21
12	3.566m	2.15
15 (Healthy Cell)	5.320m	1.98

TABLE II: Time Analysis for the state change behaviour of memristor device for faulty cell in the series combination of the 4 cells.

No of faulty cell	Time (ms) in the series combination of 4 cell configuration
1	0.35
2	0.63
3	1.98

analyzed the state change behaviour of memristor device for various cases of unhealthy and faulty cells. Fig. 4b shows current-voltage characteristics and Table I shows the corresponding numerical data for various current generator and corresponding maximum output power for the solar cell. Curve 1 (solar cell current generator for 15mA) represents the behaviour for healthy cell and have maximum output power is 5.320mW. Curve 2-6 (solar cell current generator for 12mA to 1.765mA) show the degradation in the maximum output power for the solar cells, are considered as a unhealthy solar cell. Curve 7 represents the solar cell current generator for 1mA and switching time for the memristor device is infinite and this cell considered as a faulty solar cell. From these results as shown in Table I, we conclude that switching time for the memristor device increases as the generated power by the solar cell decreases. this shows that switching time for unhealthy cells is more than the healthy cell. If generated maximum power for solar cell is lower than 77.8μW, memristor device is unable to change their state and considered as a faulty cell.

Detection of faults in the solar array is essential to prevent the system failure. We also examined the state change behaviour of memristor for the faulty cells in the series configuration of four cells as shown in Fig. 3b. Table II shows time taken by the memristor to switch their state from high resistance state to the low resistance state for the the number of faulty cells in this configuration. As the number of faulty cells increases the switching time of the device increases. This time analysis of device in the array is helpful for finding the faults. This series configuration of PV cells allow the short circuit (SC) fault and ground fault in the array and can be found by the switching time of the memristor device. We can also find the number of faulty cells in the series configuration. If one of the cell has open circuit (OC) fault in this configuration, the maximum output power will be zero and system will be failed.

III. Reliable Sensing with Process Variation Aware Memristor Array

Recent research demonstrated gas sensing properties of memristive devices [8]. The structure of a typical TiO_2-based memristor is as shown in Fig. 5a [12], [13]. The overall resistance of the memristor is determined by the resistance R_{off} of an undoped region, TiO_2, and R_{on} of a doped region region, TiO_{2-X} (Fig. 5a). A physical gas sensing memristor features a partially covered top terminal as shown in Fig. 5b to allow gas interaction with the semiconductor layer. The degree and direction of resistance change depend on the semiconductor material in the sensor as well as the subject gas.

In this paper we assume that only the R_{on} region of a memristor gas sensor is exposed to gas and changes to R_{OnEff} when exposed to gas, but without any change in the state variable. The region corresponding to R_{off} is not exposed to gas. The relationship between R_{on} and R_{OnEff} for reducing gas is $\frac{R_{\text{OnEff}}}{R_{\text{on}}} = 1 + AC^{\beta}$ and that for oxidising gas is $\frac{R_{\text{OnEff}}}{R_{\text{on}}} = \frac{1}{1+AC^{\beta}}$ [8].

1. Gas Sensor Models and Sensitivity: Let R_{M}^{I} and R_{M}^{F} be the initial and final resistances of a memristor gas sensor after exposure to Cppm of gas respectively. Based on the linear ion drift model, $R_{\text{M}}^{\text{I}} = R_{\text{on}}\frac{w}{D} + R_{\text{off}}\left(1-\frac{w}{D}\right)$ and $R_{\text{M}}^{\text{F}} = R_{\text{OnEff}}\frac{w}{D} + R_{\text{OffEff}}\left(1-\frac{w}{D}\right)$ [8], [14]. Here, D is the physical length of a sensor device and w is a region with a high concentration of dopants. Memristor gas sensors can also be modelled with non-linear memristive behaviour based on [15] as $R_{\text{M}}^{\text{I}} = R_{\text{on}} \cdot e^{\lambda_{\text{I}} \cdot (x - x_{\text{on}})/(x_{\text{off}} - x_{\text{on}})}$ and $R_{\text{M}}^{\text{F}} = R_{\text{OnEff}} \cdot e^{\lambda_{\text{F}} \cdot (x - x_{\text{on}})/(x_{\text{off}} - x_{\text{on}})}$. Here, $\lambda_{\text{I}} = \ln(\frac{R_{\text{off}}}{R_{\text{on}}})$, $\lambda_{\text{F}} = \ln(\frac{R_{\text{off}}}{R_{\text{OnEff}}})$, $x_{\text{on}} \leq x \leq x_{\text{off}}$. In this paper, we assume that $x_{\text{on}} = 0$, $x_{\text{off}} = D$, and $x = D - w$, i.e. these are tied to the physical dimensions of the devices and hence are unaffected by changes in gas concentrations.

We define the relative gas sensitivity, $S = \frac{|R_{\text{M}}^{\text{F}} - R_{\text{M}}^{\text{I}}|}{R_{\text{M}}^{\text{I}}}$. In the proposed optimisation framework, we aim at keeping $S \geq$ a measurable threshold.

2. Gas Sensitivity Under Process Variation and Wire Resistance: In ideal memristor gas sensors we assume that process parameters such as w, D, etc., and R_{on} and R_{off} are constants resulting in accurate and exact readings from one sensor to another under similar conditions. However, owing to process variations in fabricated sensors the sensor-to-sensor accuracy can vary resulting in inaccuracies. To cater for this, readings from an array of sensors is taken to improve accuracy [8]. We show in this paper that as we increase the number of memristors in the array, the effects of process variation decreases. Ideally, we wish to keep as many memristors as possible in the array, however, this reduces sensitivity due to wire resistance as well as increases power consumption. Thus we propose an optimisation framework that will help sensor designers to decide the optimal number of memristors in a sensor array for a more realistic scenario.

The effects of wire resistance is becoming more prominent as the technology nodes shrink [16]. To cater for this, we propose the memristor sensor array model as depicted in Fig. 6. The equivalent resistance is calculated using a fast dynamic programming based iterative algorithm for series-parallel networks.

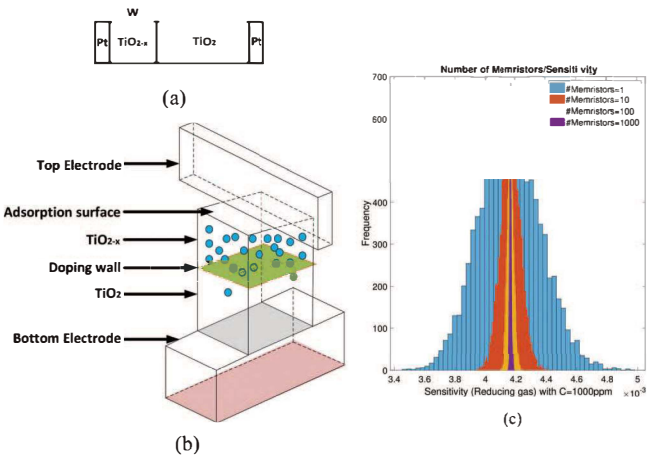

Fig. 5: (a) Structure of the TiO_2-based memristor fabricated by HP Labs; (b) Structure of Memristor for Sensing Applications; (c) Variations in sensitivity with variations in process parameters w and D ($R_w = 1\mu\Omega$, linear model).

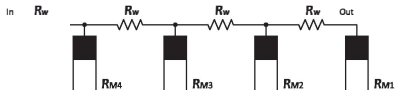

Fig. 6: A memristive sensor array with wire resistance R_w.

Fig. 5c shows the effects of process variation as well as wire resistance on sensitivity based on the linear model. Clearly, as the number of memristors is increased, the variations in sensitivity owing to process variation reduces. Also due to the wire resistance, the bell curves are shifting to the left, i.e. the mean sensitivity is decreasing as we are increasing the number of memristors. The results obtained with a non-linear model show similar trends.

3. An Optimisation Framework: Clearly there are benefits in an array of memristor sensors over a single memristor sensor as demonstrated in Fig. 5c. Hence, we have the following optimisation problem—Given: (i) A maximum number of memristors, nRes; (ii) wire resistance, R_w; (iii) variations in process parameters; (iv) maximum allowed variation in sensitivity; (v) an accuracy margin, ExpAcc; (vi) gas concentration, C; (vii) minimum measurable gas sensitivity; (viii) R_{on} and R_{off}, find out the minimum number of sensors required to ensure that the maximum allowed variation is not exceeded (or at least remains within the accuracy margin, if it is exceeded), while the sensitivity is at least as much as the specified minimum (or does not fall below the minimum by a specified margin). To this end, we present a fast heuristic optimisation algorithm.

a) A Fast Heuristic Optimisation Technique: Let ExpStDev and StDev be the expected maximum variation in sensitivity and the standard deviation of the different sensitivitiesely. A cost function ensures that the sensitivity is not less than ExpStDev$-$ExpAcc. The sensor architecture is considered to be as in Fig. 6. This approach is based on first estimating an upper limit (RightPtr) and lower limit (LeftPtr) by increasing the memristor count, n, exponentially, i.e. for $n = 1,2,4,8,\ldots,m$, where $m \leq \text{nRes}$. The limit is set when ExpStDev lies within the StDevs for LeftPtr=n and that for RightPtr=$2 \times n$ (a solution does not exist if the ExpStDev<the StDev for RightPtr=nRes).

Once the limits are established, an optimisation step narrows the limits by repeatedly finding a median=round((LeftPtr+RightPtr)/2) (i.e. logarithmically). The Monte-Carlo simulation is carried out with each median. If the simulation results in StDev>ExpStDev, then the next median is considered from the upper half with respect to the current median; otherwise it is considered from the lower half. The optimisation step iterates with each median while the following is true:

(LeftPtr<RightPtr)
AND
NOT((LeftPtr+1$\geq$RightPtr) AND (median==RightPtr))

Intermediate solutions are stored in hash tables for further speedup. The algorithm terminates either with a solution or fails if a solution does not exist under the given constraints. Computationally, this algorithm requires approximately $\lceil \log_2(\text{nRes}) \rceil$ simulation runs to establish the limits. Once the limits are established, it requires a further $\lceil \log_2(\text{UpperLimit} - \text{LowerLimit}) \rceil$ simulation runs to find the solution/fail. Hence, the algorithm requires $O(\log_2(\text{nRes}))$ simulation runs.

b) Power Optimisation: The cost function can be modified to ensure that the read power/current does not exceed a threshold, while considering StDev and mean sensitivity. The read power $P_{\text{read}} = V_{\text{read}} \times I_{\text{read}}$ (Watts), where V_{read} is the applied read voltage and $I_{\text{read}} = \frac{V_{\text{read}}}{R_{\text{eff}}}$ is the measurable read current for obtaining the effective array resistance R_{eff}. R_{eff} is computed based on a fast iterative algorithm. Hence, for a given V_{read} we can compute I_{read} and P_{read} and ensure that these do not exceed a certain threshold.

c) Applications in Sensor Array Optimisation: It is known that the sensitivity increases with an increase in gas concentration, C [8]. Our further studies have also shown that the sensors demonstrate higher variations in sensitivity due to process variations at higher Cs. This implies that different Cs will result in different numbers of sensors with this technique. Hence, for practical applications in finding an optimal number of sensors, we propose to set C to a value which demonstrates maximum variation in sensitivity and apply these techniques to obtain the optimal number of memristors under predetermined constraints. This will

ensure that at any other values of C the variations in sensitivities will remain within the specified limits.

4. Experimental Results: The proposed algorithm was coded and tested in MATLAB on a typical quad-core Intel i7 processor based laptop computer with 16GB of RAM. The device models were verified in Spice (LTSpice) simulations to determine the upper and lower limits of parametric variations.

As an example optimisation run, the algorithm was executed with the following parameters: nRes=1,000, number of iterations per Monte-Carlo simulation run=10,000, ExpStDev=0.00002, ExpMean=0.0024, ExpAcc=0.000001, $R_w = 10^{-6}\Omega$, $w/D = 0.5$, maximum variation in w/D i.e. $\Delta w/D = 0.0125$, $R_{\text{on}} = 100\Omega$, $R_{\text{off}} = 10{,}000\Omega$, $A = 4.2 \times 10^{-4}$, $B = 1$, $C = 1{,}000$ppm, GasType=Reducing, model=LinearModel. A simulation run of this algorithm appears in Fig. 5c. It reported a total of 108 sensors at an estimated StDev of 1.99822×10^{-5} and mean sensitivity of 0.00416857, which are all within expected limits. This algorithm required approximately 14 runs of Monte-Carlo simulation (logarithmic) and took only 4.9 seconds to complete. In contrast, an exhaustive technique where we incrementally tried out all memristors required 38.6 seconds to complete and reported the same result. We assumed the "Three-Sigma-Rule" for estimating limits on process variations. Table III shows the relative accuracy of the sensitivity boundaries (corners) as calculated vs the figures observed during simulations.

TABLE III: Sensitivity Boundaries: Calculated vs Simulated.

Memristors	Mean	Calculated	Simulated	%Error
1	0.00416271	0.00333415	0.00341963	2.49966
		0.00499128	0.00493576	-1.12493
10	0.00416828	0.00390692	0.00390795	0.0264562
		0.0044296	0.00443419	0.102377
100	0.00416863	0.00408471	0.00410142	0.407361
		0.00425255	0.00424209	-0.24648
1000	0.00416823	0.00414182	0.00410142	-0.0265055
		0.00419464	0.00424209	-0.0892682

IV. Logic-in-memory with Memristors

Another interesting application of memristors is logic. Memristors can construct a logic gate with or without additional circuit devices. When memristors are used as switches, similarly to CMOS transistors, the operation is based on switching the resistance of the memristors according to the input voltages and by that changing the voltage at the output. In this type of circuits, the states of the memristors are complementary, *i.e.*, one memristor is in R_{on} and the other in R_{off}. This logic technique, called *Memristor Ratioed Logic* (MRL) [17] is useful when memristors are integrated with CMOS logic, and offer a three dimensional extension to the logic circuits. Nevertheless, since the switching time of memristors is substantially slower than the delay of CMOS logic gates (nanoseconds versus picoseconds), MRL gates are much slower than CMOS logic.

A different approach for memristive logic is to use the resistive state of the memristors as the input and output of the gate. This approach is called *stateful logic* [18], and is more suitable for in-memory computation. In stateful logic techniques, such as IMPLY [19] and MAGIC [9], the stored data in the memristive memory cells serve as the inputs and the resistance at the end of the computation is the output. The schematic of a MAGIC gate within a memristive crossbar array is illustrated in Fig. 7. This approach eliminate the external data movement, which is the primary bottleneck in modern computing systems, and by that substantially improve the performance and energy efficiency [20].

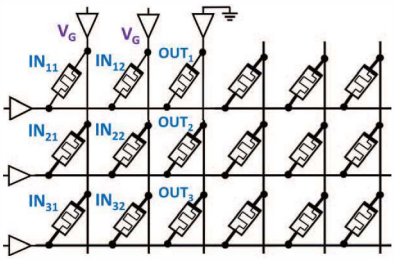

Fig. 7: Schematic of three MAGIC NOR gates within a memristive crossbar array. Each row performs an independent gate, where the bitlines connected to V_G serve as the inputs of the gate, and the bitline connected to ground is the output of the gate.

Using stateful logic, memristive crossbar arrays can be used not only as memory, but also as processing unit, enabling a unit called *memristive Memory Processing Unit* (mMPU) [21]. This unit is backward compatible with conventional von Neumann architecture and can therefore be used as standard memory. In addition to that, mMPU can be used to process data, and a controller decides whether to compute or perform standard memory read/write operations. Since stateful logic operations can be performed concurrently on numerous rows (or columns) within a single memristive memory array [22], the mMPU may exhibit high throughput for selected applications, such as image processing, where mMPU outperforms different accelerators by 30× to 300× [21]. The system view of the mMPU is shown in Fig. 8.

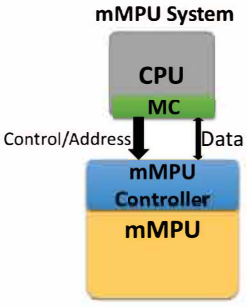

Fig. 8: System view of the memristive memory processing unit (mMPU) in a non-von Neumann architecture. The mMPU gets commands from the CPU through the memory controller (MC) and its controller transfer it to in-memory operations (*e.g.*, MAGIC NOR) performed within the memristive memory arrays.

While single gate and small scale stateful logic have been demonstrated [23], [24], a large scale computation is

still an open issue. Furthermore, the algorithms to support such computations, including how to efficiently parallelize the operations and map them into the memory, have been demonstrated only for specific tasks [22], [25], [26] or for limited parallelism [27]–[29], and a more holistic approach is still required. Another open issue is the programming model since such a non-von Neumann architecture requires a new model that can support both the von Neumann compatibility, as well as the processing in memory.

V. Conclusions

In this paper, we have elaborated three different use cases of memristive technology. Firstly, we presented a solar cell array health monitoring approach for potential fault detection solution for efficient testing of cells in the presence of faults. In this approach, we divide the array into segments such that any faults is detectable and the method has been validated through SPICE simulation. Secondly, using our memristor gas sensor model, the initial and final resistances of each sensor under a gas concentration C is computed in the presence of process variation and wire resistance. In order to determine the number of memristors to ensure a certain level of accuracy in sensitivity, a technique to optimise the sensor array based on an acceptable sensitivity variation and minimum sensitivity margin is presented. Also presented the accuracy of the corners within reasonable levels of observed values. Lastly, it is argued that memristors can construct a logic gate with or without additional circuit devices. Also reviewed an approach that eliminate the external data movement, which is the primary bottleneck in modern computing systems, and this would substantially improve the performance and energy efficiency.

References

[1] Leon Chua. Memristor-the missing circuit element. *IEEE Transactions on circuit theory*, 18(5):507–519, 1971.

[2] Dmitri B Strukov, Gregory S Snider, Duncan R Stewart, and R Stanley Williams. The missing memristor found. *nature*, 453(7191):80, 2008.

[3] Sascha Vongehr and Xiangkang Meng. The missing memristor has not been found. *Scientific reports*, 5:11657, 2015.

[4] Yenpo Ho, Garng M Huang, and Peng Li. Nonvolatile memristor memory: device characteristics and design implications. In *Proceedings of the 2009 International Conference on Computer-Aided Design*, pages 485–490. ACM, 2009.

[5] Yenpo Ho, Garng M Huang, and Peng Li. Dynamical properties and design analysis for nonvolatile memristor memories. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 58(4):724–736, 2011.

[6] Kuk-Hwan Kim, Siddharth Gaba, Dana Wheeler, Jose M Cruz-Albrecht, Tahir Hussain, Narayan Srinivasa, and Wei Lu. A functional hybrid memristor crossbar-array/cmos system for data storage and neuromorphic applications. *Nano letters*, 12(1):389–395, 2011.

[7] Jimson Mathew, Yuamfam Yang, M Ottavi, T Browna, A Zampetti, A Di Carlo, AM Jabir, and Dhiraj K Pradhan. Fault detection and repair of dsc arrays through memristor sensing. In *2015 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFTS)*, pages 7–12. IEEE, 2015.

[8] A. Adedotun, J. Mathew, A. Jabir, C. D. Natale, E. Martinelli, and M. Ottavi. Efficient sensing approaches for high-density memristor sensor array. *Journal of Computational Electronics*, 17(3):1285–1296, 2018.

[9] S. Kvatinsky, D. Belousov, S. Liman, G. Satat, N. Wald, E. G. Friedman, A. Kolodny, and U. C. Weiser. Magic—memristor-aided logic. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 61(11):895–899, 2014.

[10] R Sastrawan, J Renz, C Prahl, J Beier, A Hinsch, and R Kern. Interconnecting dye solar cells in modules—i–v characteristics under reverse bias. *Journal of Photochemistry and Photobiology A: Chemistry*, 178(1):33–40, 2006.

[11] Jimson Mathew, Marco Ottavi, Yunfan Yang, and Dhiraj K Pradhan. Using memristor state change behavior to identify faults in photovoltaic arrays. In *2014 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT)*, pages 86–91. IEEE, 2014.

[12] J. J. Yang, M. D. Pickett, X. Li, D. A. Ohlberg, D. R. Stewart, and R. S. Williams. Memristive switching mechanism for metal/oxide/metal nanodevices. *Nature Nanotechnology*, 3(7):429–433, 2008.

[13] R. S. Williams. How we found the missing memristor. *IEEE Spectrum*, 45(12):28–35, 2008.

[14] Strukov Dmitri B, Gregory S Snider, Duncan R Stewart, and R Stanley Williams. The missing memristor found. *nature*, 453(7191):80, 2008.

[15] S. Kvatinsky, E. G. Friedman, A. Kolodny, and U. C. Weiser. Team: Threshold adaptive memristor model. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 60(1):211–221, 2013.

[16] A. Adeyemo, J. Mathew, and A. Jabir. Minimising impact of wire resistance in low-power crossbar array write scheme. *Journal of Low Power Electronics*, 13(4):649–660, 2017.

[17] S. Kvatinsky, N. Wald, G. Satat, A. Kolodny, U. C. Weiser, and E. G. Friedman. Mrl — memristor ratioed logic. In *2012 13th International Workshop on Cellular Nanoscale Networks and their Applications*, pages 1–6, Aug 2012.

[18] J. Reuben, R. Ben-Hur, N. Wald, N. Talati, A. Haj Ali, P.-E. Gaillardon, and S. Kvatinsky. Memristive logic: A framework for evaluation and comparison. In *2017 27th International Symposium on Power and Timing Modeling, Optimization and Simulation (PATMOS)*, pages 1–8, Sep. 2017.

[19] S. Kvatinsky, G. Satat, N. Wald, E. G. Friedman, A. Kolodny, and U. C. Weiser. Memristor-based material implication (imply) logic: Design principles and methodologies. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 22(10):2054–2066, Oct 2014.

[20] R. Ben Hur and S. Kvatinsky. Memory processing unit for in-memory processing. In *2016 IEEE/ACM International Symposium on Nanoscale Architectures (NANOARCH)*, pages 171–172, July 2016.

[21] A. Haj-Ali, R. Ben-Hur, N. Wald, R. Ronen, and S. Kvatinsky. Not in name alone: A memristive memory processing unit for real in-memory processing. *IEEE Micro*, 38(5):13–21, Sep. 2018.

[22] N. Talati, S. Gupta, P. Mane, and S. Kvatinsky. Logic design within memristive memories using memristor-aided logic (magic). *IEEE Transactions on Nanotechnology*, 15(4):635–650, July 2016.

[23] J. Borghetti, G. S. Snider, P. J. Kuekes, J. J. Yang, D. Stewart, and R. S. Williams. Memristive switches enable stateful logic operations via material implication. *Nature*, 464:873–6, 04 2010.

[24] H. Bae, B. Jang, H. Park, S.-H. Jung, H. M. Lee, J.-Y. Park, S.-B. Jeon, G. Son, I.-W. Tcho, K. Yu, S. Im, S.-Y. Choi, and Yang-Kyu Y.K. Choi. Functional circuitry on commercial fabric via textile-compatible nanoscale film coating process for fibertronics. *Nano Letters*, 17, 09 2017.

[25] A. Haj-Ali, R. Ben-Hur, N. Wald, and S. Kvatinsky. Efficient algorithms for in-memory fixed point multiplication using magic. In *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–5, May 2018.

[26] A. Haj-Ali, R. Ben-Hur, N. Wald, R. Ronen, and S. Kvatinsky. Imaging: In-memory algorithms for image processing. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 65(12):4258–4271, Dec 2018.

[27] N. Talati, A. Haj Ali, R. Ben Hur, N. Wald, R. Ronen, P.-E. Gaillardon, and S. Kvatinsky. Practical challenges in delivering the promises of real processing-in-memory machines. In *2018 Design, Automation Test in Europe Conference Exhibition (DATE)*, pages 1628–1633, March 2018.

[28] R. Ben Hur, N. Wald, N. Talati, and S. Kvatinsky. Simple magic: Synthesis and in-memory mapping of logic execution for memristor-aided logic. In *2017 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 225–232, Nov 2017.

[29] P. L. Thangkhiew, R. Gharpinde, and K. Datta. Efficient mapping of boolean functions to memristor crossbar using magic nor gates. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 65(8):2466–2476, Aug 2018.