Implementation and Characterization of a Memristive Memory System

David Radakovits and Nima TaheriNejad
Institute of Computer Technology
TU Wien, Vienna, Austria

***Abstract*—Memristors are one of the promising emerging technologies to address several challenges faced by the computing system of the day. However, a sizeable portion of the works in the literature are not supported by practical implementations or their details are kept as trade secrets. In this work, we propose and implement a writing and reading circuit for a memristive memory system and present our measurement results. A key feature of the proposed system is that it does not need any read-out compensation and virtually no refreshing (due to read-out). However, we observed that by the passage of the time (and irrespective of not applying any inputs) some information loss happens, which necessitates refreshing and dictates its frequency. We associate this phenomenon, which has not been reported in the literature before, to what we call "leakage current". We anticipate this paper to be a starting point for seeing more implementation-based works in the literature, modeling the leakage current phenomenon, and incorporating such design and consideration into the design process of memristive systems.**

***Keywords*—Memristor, ReRAM, Memory System, Implementation, Characterization, Leakage Current.**

I. Introduction

MEMORY technology plays a major role in the energy budget of modern computer systems [1]–[3]. At the same time, more and more computer systems are powered by batteries, such as mobile devices, wearables, peripheral sensors, and IoT gadgets [4]. Memristors, were first described by L. Chua in 1971 [5] represent the fourth basic circuit element besides resistors, capacitors and inductors. Unlike conventional memory technologies, which mostly use a certain amount of charge to store information, i.e., bits, in a memory cell, memristors store information in their resistance, which only changes if a current (charge) is applied to it. This non-volatile character and its very small size of several nanometers [6] qualifies memristors to make computer memory more energy efficient and dense at the same time. Memristive memory technology also enables a new computing paradigm, which is called In-Memory Computation (IMC) [7]. IMC is a promising candidate to solve the problems caused by the so-called Von-Neumann-bottleneck, which not only limits the performance of modern computer systems, but also causes major energy consumption [8].

Memristive memories have been studied in academia [9]–[12] and industry [13], which includes commercial products [13], [14] and reported integration in some products of some big companies such as Panasonic [14]. However, industrial players do not reveal details of their work. On the other hand, most of academic works are either merely theoretical, or they measure memristive elements as memory candidates in laboratory setups with precise, expensive, and bulky devices. Hence, they little present the characteristics and challenges of memristive memory as a full system in a realistic practical situation [15]. In this paper, we try to bridge this gap by physical implementation of a full memory system and characterize it. While the term "memristor" unites a whole class of partially very different devices, in this work resistive switching devices, or Resistive Random Access Memorys (ReRAMs) are used and meant by the term. Further details about the used memristor technology are given in Section II. An interesting outcome of this work is the "leakage current" characteristic which was observed in our measurements. This characteristic of memristors has not been previously reported in the literature. This is an important contribution since it will attract the attention of researchers to further observe, characterize, and model this phenomenon and incorporate its relevant considerations in their design.

In this work, we propose methods of writing and reading data on memristors in Section II. Experimental evaluations of the proposed memory system is shown in Section III. In Section IV the results are discussed and conclusions are drawn.

II. Methodology

In this section we present our proposed approach and its implementation for writing and reading the circuit. A distinction of this approach compared to many existing approaches [10], [16], [17] is that we use current for reading the data from memristors as opposed to voltage pulses. Reading the data from the memristors using current pulses (and consequently measuring the voltage) was chosen over reading with voltage pulses (and consequently measuring the current), because the change of resistance is a function of charge (current). In memristors current is a non-linear function of voltage across the memristor and hence the effect of voltage reading pulse is widely varied and can hardly be predicted. A current pulse on the other hand, allows for more control changes in the state and thus -if need be- allows a more accurate compensated reading through application of an inverted pulse [11], [18].

A. Writing

The resistance of a memristor can be any resistance between two opposite maxima, which are generally referred to as R_{ON} (the lowest resistance of the memristor) and R_{OFF} (the largest

978-1-7281-0319-8/19/$31.00 ©2019 IEEE

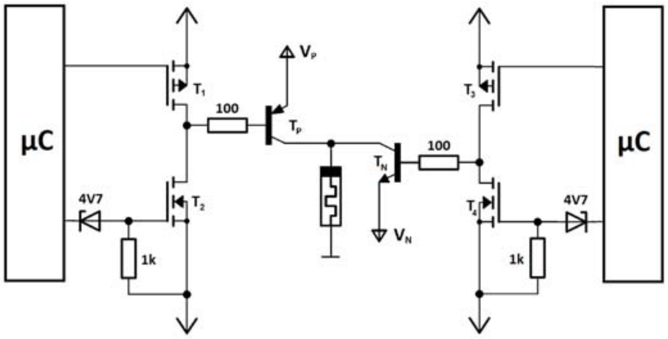


Fig. 1. Driver circuit for the write operation.

resistance value). To store binary information in a memristor, it is beneficial to use those maxima of resistance to represent data, so that the contrast between bit values is maximized. In our work, we chose to map logical ‘1’ to R_{ON} and logical ‘0’ to R_{OFF}. Regardless of its current resistance, to write a logical ‘1’ (or ‘0’) to a memristor, i.e., drive it to R_{ON} (or R_{OFF}), it is sufficient to assume it to be in the opposite maximum and apply a current pulse which is large enough (in both amplitude and duration) to cross the resistance interval completely.

These pulses are generated using transistor drivers, which can be seen in Figure 1. Transistors P and N are driven by two bridges ([T_1, T_2] and [T_3, T_4]), which allow setting the write stage to a high impedance state, which decouples it during read operation. The Zener diodes and resistors at T_2 and T_4 are for level shifting which is necessary to control the bridge correctly, since only positive voltages are available from the microcontroller board. To set the memristor to R_{ON}, T_P is turned on and applies $V_P = 1$V to the memristor. To set it to R_{OFF}, T_N applies $V_N = -400$mV. The duration of the pulse in both cases is 100μs.

B. Reading

To read the data from a memristor, i.e., measure its resistance, we propose the circuit depicted in Figure 2. As we can see there, the memristance is compared with a reference resistor to distinguish the logical value stored in it. The value of the reference memristor R_{ref} must be chosen, such that $R_{on} < R_{ref} < R_{off}$ is fulfilled. To compare the resistances, the same current pulse is applied to both memristor and the reference resistor and the occurring voltage drop is compared using an operational amplifier. We designed and developed this read out circuit based on an initial idea in [18]. It is important to consider that the amplitude of the applied current pulse must be sufficiently small, such that the resistance of the memristor is minimally disturbed. However, some memristors show threshold effects, i.e., the resistance of the memristor does not change if the current through the memristor is smaller than a certain threshold current I_{th}. If the memristor shows such effects, a reading current, I_{read} which satisfies $|I_{read}| < |I_{th}|$ does not affect the resistance at all. However, in the memristor technology which is available to us, similar to many other memristors, the threshold effect does not have a strong presence. In such cases, by minimizing the reading current the consequent drift and the need for drift compensation or refreshing can be minimized. To this end we chose $I_{read} \approx 5\mu$A.

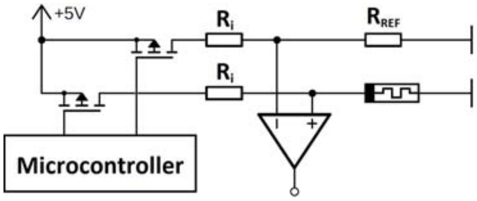


Fig. 2. Operational amplifier circuit for the read operation.

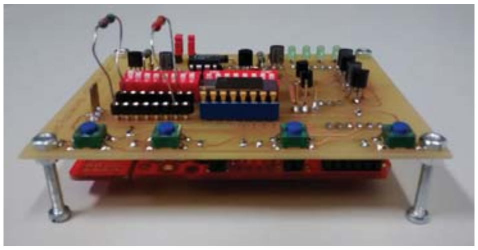


Fig. 3. Discrete implementation of the memory system on PCB using off-the-shelf components. The microcontroller board is attached from the bottom side.

As can be seen in Figure 2, the sources, which provide the reading current pulse, are implemented as quasi current sources. The current sources must provide a sufficiently high source impedance (see R_i in Figure 2), such that the resistance of the memristor doesn’t affect the pulse amplitude. Therefore, R_i was chosen to be 1MΩ, which results in $I_{read} \approx 5\mu$A.

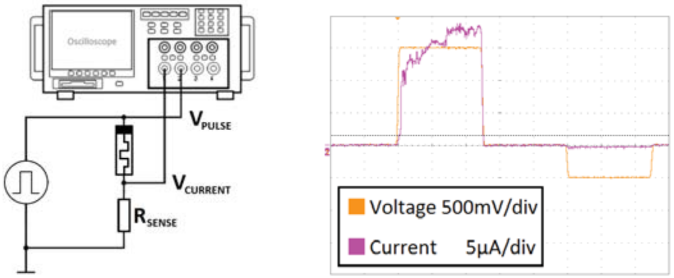
III. Experimental evaluation

A. Setup

The memory system, i.e., write circuit, read circuit and the control circuit, was implemented on a PCB using off-the-shelf component. The end prototype can be seen in Figure 3. For the control circuit a microcontroller, namely an *Infineon XMC1100 boot kit* running at 32 MHz, was used. The microcontroller operated the memory system (writing to it and reading from it) and also conducted the tests and recording of results autonomously. The write transistors, which can be seen in Figure 1, are BC557A (Q_P) and BC547B (Q_N), respectively. The driver transistors are BSP254 (Q_1 and Q_3) and 2N7000 (Q_2 and Q_4). The experiments were undertaken with memristors fabricated by KNOWM [19]. The memristors are tungsten chalcogenide ReRAMs. The data-sheet of the used memristors including more detailed information can be found at [20].

B. Memristor Characterization

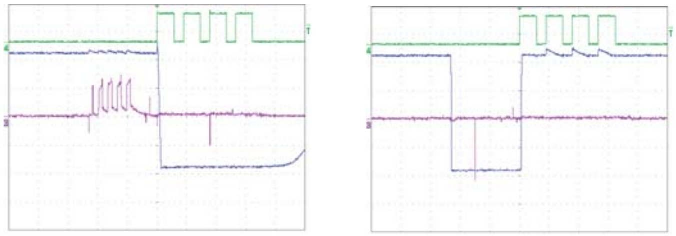
Before the memristors can be used in the memory system, their range of resistance $[R_{ON} : R_{OFF}]$ needs to be determined so that the reference resistor can be chosen accordingly. To measure R_{ON} and R_{OFF} respectively, a series of positive



(a) Characterization setup

(b) Measurement of R_{ON} & R_{OFF}. Time base is 2ms/div.

Fig. 4. Characterization setup and measurement results of resistance in a memristive memory element.



(a) Writing ‘1’

(b) Writing ‘0’

Fig. 5. Writing once and reading four times. The output of the operational amplifier (blue, 2V/div) is inverted. The current through the memristor can be seen in purple (10μA/div). The green line denotes a trigger signal for the read operation.

and negative pulses were applied to the memristor under test. R_{ON} and R_{OFF} values were respectively measured as soon as there was no further change in resistance caused by the applied pulses. 4(a) shows the setup that was used for the measurements. V_{PULSE} and $V_{CURRENT}$ were measured using a LeCroy WaverJet 324A oscilloscope and using those two parameters we calculated the low and high resistance of the memristor. 4(b) is a screen shot from the oscilloscope which shows an example of a resistance measurement. The parameters of the memristor and the performance of the system was tested with can be found in Table I. Next we describe which characteristics of the memristor system were measured and how.

C. Normal Operations

Figure 5 illustrates the memory operation, where a ‘1’ or ‘0’ is written to the memristor, respectively. In 5(a), we see that four consecutive positive pulses were written to the memristor. The magenta line depicts a high current flowing through the memristor (which shows that memristor is in R_{ON} state). In contrast, the magenta line in 5(b) doesn’t show pulses, since the current is very low during negative pulses (memristor being at the R_{OFF} state). In both figures, the blue line indicates the output of the operational amplifier in the read-out stage. Note that this signal is inverted. Therefore, it drops at the first read out (indicated by the green signal) while reading ‘1’ in 5(a) and rises while reading ‘0’ in 5(b).

TABLE I. Memristors system parameters and performance

R_{ON}	R_{OFF}	R_{REF}	R_i	V_P	V_N	${N_{max}}^{\alpha}$	${T_{max}}^{\beta}$
6.8kΩ	>125kΩ	15kΩ	1MΩ	1V	400mV	$> 6000^{\gamma}$	81h
31.7kΩ	>125kΩ	56kΩ	1MΩ	1V	400mV	$> 10'000^{\gamma}$	$>24h^{\gamma}$
16.7kΩ	>125kΩ	47kΩ	1MΩ	1V	400mV	$> 10'000^{\gamma}$	$>24h^{\gamma}$

$^{\alpha}$ N_{max} denotes the maximum number of read cycles.
$^{\beta}$ T_{max} denotes the maximum hold time.
$^{\gamma}$ The “>” denotes, that tests where aborted at this number.

D. Memristor System Characterization

The performance of the memory system was evaluated regarding two main aspects: (i) the *maximum number of read cycles* (N_{max}) which is the maximum number of correct read-backs without refreshing the memory value or compensating the read-out effect, (ii) the *maximum holding time* (T_{max}) which is the maximum period of time for which the stored data can be correctly read without refreshing or compensating the read-out effect. The first set of tests shows the effect and merit of the proposed reading circuit on the stability of memory system and its requirements for being refreshed. The second set of tests reveal the existence and effect of the “leakage current”.

We note that here only the worst achieved test results regarding N_{max} and T_{max} will be discussed. We have performed some additional tests with two other memristors which led to better results. A summary of all performed tests is inserted in Table I.

1) Maximum read cycles (N_{max}): This performance metric was measured by writing a bit to a memristor and reading it every 100ms. For both bits the tests were aborted after 6000 correct read-out cycles[1]. This high number of read cycles shows that the proposed technique has no significant effect on the stored value and virtually no read-out compensation or frequent refreshing is necessary for this memory system. Hence, refreshing frequency of the memory using the proposed read-out circuit is dictated by the maximum holding time and the leakage current.

2) Maximum holding time (T_{max}): This performance metric was measured by writing a bit to the memristor and reading it once every hour. For a written ‘1’, the test was aborted after 12 hours[1] of reading the correct value. Since the read current, due to its direction, drives the memristor deeper into R_{ON} (representing ‘1’), this state is more stable than R_{OFF} (representing ‘0’). Therefore, the experiment for R_{OFF} (representing ‘0’) was extended for a longer time and a maximum holding time of 81 hours was measured for a stored ‘0’. Therefore we expect and consider this as the minimum holding time for ‘1’ and consequently for the memory element in general. Reading only once an hour during the test period resulted in 81 correct readings, i.e., 81 read-outs, however, during maximum read cycle testing a two orders of magnitude higher number of read-outs was achieved. Thus, we conclude that the loss of data doesn’t originate from the reading mechanism itself but has to be traced back to a different effect. This leads to our observation of the “leakage current” which we discuss next.

[1] As it can be seen in Table I, extended tests have been performed, which were aborted at 10’000 cycles and 24 hours of reading time, respectively, without permanent loss of data.

We note that $\leq 5\%$ of the read-out values in all our test were false. However, these read-outs were randomly distributed in the tests. We think external noises or cosmic radiation might be the reason for this random false read-outs.

E. Leakage Current

In theory, the state and consequently the resistance of a memristor should only change if there is an external stimulus applied to the memristor, e.g., a current flowing through it. An implication of our observations in Section III-D2 is that, in contrast to theory, the memristor state can also change in absence of any stimulus. This is shown by the fact that as shown in Section III the number of maximum read cycles -at least in one case- is significantly higher than the number of read cycles that was necessary to reach the maximum holding time. This means that the loss of information is not caused by the read-out effect[2] and rather by another phenomenon happening by the passage of the time and in the absence of inputs. We associate this unintended drift of memristor state to what we call "leakage current", which to the best of our knowledge is not reported or characterized in the literature. We note, that this effect is fundamentally different from the reported "history effect" [21], [22], which describes the independence of a steady state in a memristor from its initial state, after a certain repetitive input pattern is applied. Whereas this "history effect" does not deal with state changes in the absence of inputs [21], we observed and report a state change in the absence of any input stimulus.

IV. Conclusion

We presented the practical implementation and characterization of a ReRAM-based memory system. The system was implemented using off-the-shelf components and achieved a performance of a minimum of a 6000 read cycles and 81 hours of undisturbed storage without refreshing or compensation for the read-out (thanks to the design quality of the proposed read-out mechanism). Moreover, we observed and reported an effect which we call "leakage current". To the best of our knowledge, thus far, this phenomenon is unreported in the literature. This effect causes the memristor to loose the stored data, i.e., to drift in its state, even though no input stimulus were applied to it. Further examination of this effect and its reproduction in simulation models could bring a large benefit to memristor circuit and system design and memristor-related research in general.

References

[1] A. Carroll *et al.*, "An analysis of power consumption in a smartphone." in *USENIX annual technical conference*, vol. 14, 2010, pp. 21–21.

[2] D. Economou, S. Rivoire, C. Kozyrakis, and P. Ranganathan, "Full-system power analysis and modeling for server environments." International Symposium on Computer Architecture-IEEE, 2006.

[3] A. Mahesri and V. Vardhan, "Power consumption breakdown on a modern laptop," in *International Workshop on Power-Aware Computer Systems*. Springer, 2004, pp. 165–180.

[4] R. Iyer and E. Ozer, "Visual IoT: Architectural challenges and opportunities; toward a self-learning and energy-neutral IoT," *IEEE Micro*, vol. 36, no. 6, pp. 45–49, Nov 2016.

[5] L. Chua, "Memristor-the missing circuit element," *IEEE Transactions on circuit theory*, vol. 18, no. 5, pp. 507–519, 1971.

[6] A. S. Oblea, A. Timilsina, D. Moore, and K. A. Campbell, "Silver chalcogenide based memristor devices," in *Neural Networks (IJCNN), The 2010 International Joint Conference on*. IEEE, 2010, pp. 1–3.

[7] P. Trancoso, "Moving to memoryland: in-memory computation for existing applications," in *Proceedings of the 12th ACM International Conference on Computing Frontiers*. ACM, 2015, p. 32.

[8] M. A. Zidan, J. P. Strachan, and W. D. Lu, "The future of electronics based on memristive systems," *Nature electronics*, vol. 1, pp. 22–29, 2018.

[9] H. Kim *et al.*, "Memristor-based multilevel memory," in *12th International Workshop on Cellular Nanoscale Networks and Their Applications (CNNA)*, Feb 2010, pp. 1–6.

[10] M. Zangeneh and A. Joshi, "Design and optimization of nonvolatile multibit 1T1R resistive RAM," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 22, no. 8, pp. 1815–1828, Aug 2014.

[11] N. Taherinejad, P. D. S. Manoj, and A. Jantsch, "Memristors' potential for multi-bit storage and pattern learning," in *2015 IEEE European Modelling Symposium (EMS)*, Oct 2015, pp. 450–455.

[12] K.-H. Kim *et al.*, "A functional hybrid memristor crossbar-array/cmos system for data storage and neuromorphic applications," *Nano letters*, vol. 12, no. 1, pp. 389–395, 2011.

[13] ReRAM. (2017) Crossbar. [Online]. Available: https://www.crossbar-inc.com/en/

[14] G. Hilson, *Imec, Panasonic Push Progress on ReRAM*, eeTimes, 2015. [Online]. Available: https://www.eetimes.com/document.asp?doc_id=1327307

[15] N. Taherinejad and D. Radakovits, "From behavioral design of memristive circuits and systems to physical implementations," *IEEE Circuits and Systems Magazine*, under revision, 2019.

[16] C. E. Merkel *et al.*, "Reconfigurable n-level memristor memory design," in *Neural Networks (IJCNN), The 2011 International Joint Conference on*. IEEE, 2011, pp. 3042–3048.

[17] S. S. Sarwar *et al.*, "Memristor-based nonvolatile random access memory: Hybrid architecture for low power compact memory design," *IEEE Access*, vol. 1, pp. 29–34, 2013.

[18] N. Taherinejad, S. M. PD, M. Rathmair, and A. Jantsch, "Fully digital write-in scheme for multi-bit memristive storage," in *Electrical Engineering, Computing Science and Automatic Control (CCE), 2016 13th International Conference on*. IEEE, 2016, pp. 1–6.

[19] June 2018. [Online]. Available: https://knowm.org/

[20] June 2018. [Online]. Available: https://knowm.org/wp-content/uploads/DM8-16DIP-BS-AF-W.pdf

[21] A. Ascoli *et al.*, "History erase effect in a non-volatile memristor," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 63, no. 3, pp. 389–400, March 2016.

[22] S. Ghedira *et al.*, "Dynamical resistive switching of a generic memristor model: Analysis and simulation," in *2017 International Conference on Engineering MIS (ICEMIS)*, May 2017, pp. 1–5.

[2] Although the read-out potentially has some effect in the loss of information, this effect is minor. This is supported by the fact that in frequent reading number of read-outs which lead to a loss of information is two orders of magnitude larger than the number of infrequent read-outs used to obtain the maximum holding time. Therefore, this two orders of magnitude difference renders the effect of read-out current negligible.