Boolean Logic Gates from a Single Memristor via Low-Level Sequential Logic

Ella Gale, Ben de Lacy Costello, and Andrew Adamatzky

Unconventional Computing Group, University of the West of England,
Dept. of Applied Sciences & Dept. of Computer Science and Creative Technology,
Frenchay Campus, Coldhabour Lane, Bristol, BS16 5SR, UK
{ella.gale,ben.delacycostello,andrew.adamatzky}@uwe.ac.uk
http://uncomp.uwe.ac.uk

Abstract. By using the memristor's memory to both store a bit and perform an operation with a second input bit, simple Boolean logic gates have been built with a single memristor. The operation makes use of the interaction of current spikes (occasionally called current transients) found in both memristors and other devices. The sequential time-based logic methodology allows two logical input bits to be used on a one-port by sending the bits separated in time. The resulting logic gate is faster than one relying on memristor's state switching, low power and requires only one memristor. We experimentally demonstrate working OR and XOR gates made with a single flexible Titanium dioxide sol-gel memristor.

Keywords: Memristor, sequential logic, ReRAM, OR, XOR, Boolean logic, Time-separated logic.

1 Introduction

The memristor is the recently-discovered [1] fourth fundamental element, joining the set of the resistor, inductor and capacitor. It was predicted to exist based on an expectation of symmetry in electromagnetic phenomena when applied to circuit theory [2], specifically in that it would be passive two-terminal device that would relate the two as-then-unrelated circuit measurables: charge, q, and magnetic flux, φ [1]. From knowledge about its electronic properties, Chua predicted that it would be a non-linear version of a resistor that possesses a memory, hence the name memristor, a contraction of memory-resistor.

Whilst Chua's theoretical contributions were not known to the wider chemical and physics communities, devices highly similar in constitution and operation to Strukov's memristor [1] were created and dubbed ReRAM, for Resistive Random Access Memory, after the use their inventors intended for them. What exactly constitutes a memristor or ReRAM device is a matter of debate, although it

---

[1] The other measurables being current and voltage, and the other five relationships being the definitions of current and voltage and the constitutive relations of the other three fundamental circuit elements.

has been suggested that they may be the same thing [3]. Both memristors and ReRAM have suggested uses as computer memory and both are believed to possess the same physical interactions and thus, in this paper, we shall deal with both under the name of memristors, where it is understood that a large part of the results presented here should be tested on ReRAM devices and are expected to work in the same way.

Both memristors [1] and ReRAM [4] have been suggested as possible low-power next-generation computer memory technology, however the field of ReRAM has been around for 20 years and has not yet produced a commercial product and Hewlett-Packard (the company that discovered the Strukov memristor) has been delaying their computer memory offering based on their memristor.

Chua's theoretical model of the memristor has been used to model neuronal synapses (see for example [5–7]) and to update the Hodgkin-Huxley model of neuronal membranes and axonal transport [8, 9]. It has been shown [10] that the experimental memristor spikes in a similar manner to those seen in axonal transport, where it is understood that neurons demonstrate a voltage spike in response to the current influx, and the spikes shown in [10] are current spikes in response to the voltage change. These current spikes have been seen in other memristor systems to ours and are generally ignored or dismissed as current transients. A view which will be tested with our devices in a forthcoming paper. Regardless of how these spikes arise, they are impossible (as far as we know and the literature states) to remove and thus it is our opinion that future uses of memristor technology will have to involve these spikes. Based on the relation to the brain's operation, we consider that memristor networks will be useful for neuromorphic computing, however in this paper we will demonstrate how a single memristor can be used as a Boolean logic gate by making use of the physical property of these current spikes, which can be done if we take an unconventional approach to logic assignation.

This is not the first paper on how to make logic gates with memristors. Strukov et al [11] resorted to using implication logic to design logic gates which required two memrsitors (IMP-FALSE logic is Turing complete, but somewhat unfamiliar to computer scientists). The most notable Boolean logic gates were simulated by Pershin and di Ventra [12] and required a memcapacitor, three or four memristive systems and a resistor. Before the gate was sent the two bits of data, a set of initialization pulses were required to be sent to put the gate into the correct state to give the correct answer. This system, however, is not true Boolean logic because these initialization pulses were different dependent on what the logic to follow would be. Thus the gate can not be considered to be operating only on the two bits of input data and is not a simple Boolean logic gate (it is a Turing machine doing a computation on several bits of data (Boolean input pulses and initialization pulses) which is capable of modeling a Boolean logic gate). Note also that this scheme was tested with memristor emulators, not real devices. There have been other more complex designs for memristor based Boolean logic gates, the simplest of which requires 11 circuit elements [13]. In this paper, we will demonstrate how to perform Boolean logic with a single memristor.

Although the memristor is credited with being the first computational device to combine memory and processing functionality in one, with the suggestion of an entirely new type of computer, remarkably there have been relatively few papers on how this new computer might work: most people have chosen to focus on stateful memory applications [11].

We will now demonstrate the physical properties of our memristors [14] and validate their reproducibility (section 3), explain the concept of how these spikes can be used to perform Boolean logic (section 4) and, as an example, demonstrate experimentally that a single memristor can act an OR (section 4.1) or and XOR (section 4.2) gate.

2   Methodology

The memristors are flexible TiO_2 sol-gel memristors with aluminium electrodes and were made as in [14] with the sol-gel created as in [15] (the memristor chosen was a curved-type memristor (see [14])). All tests on the memristor were performed with a Keithley 2400 Sourcemeter and data was recorded and analysed using MatLab. Each timestep was 0.02s. The voltages used and voltage waveforms varied as are discussed below.

3   Physical Properties of the Memristor

When there is a change in voltage, ΔV, across a memristor the device exhibits a current spike, the physical cause of which is discussed at length in [10]. This spike is highly reproducible and repeatable and is related to the size of the voltage change (ΔV) [10]. The spike's size (as measured by the first measurement after the Keithley's changed voltage) is highly reproducible, the current then relaxes to a stable long-term value (this value is predictable and reproducible), and it takes approximately 2-3 seconds to get to this value.

This slow relaxation is thought to be the d.c. response of the memristor [10] and if a second voltage change happens within this time frame, its resulting current spike is different to that expected from the ΔV alone. The size and direction of this current spike depends on the direction of ΔV, the magnitude of ΔV and the short-term memory of the memristor.

As an example, consider a memristor pulsed with a positive 1V voltage square wave as in figure 2 (where the pulses are repeated to demonstrate the repeatability) with a timestep of $\approx 0.02s$. The current response is shown in figure 1 and we can see there is a positive current spike associated with the $+\Delta V$ and a perhaps less obvious negative current spike associated with the $-\Delta V$ transition from $+1V \rightarrow 0V$. At approximately 20s, we shortened the square wave to a single time step, and the memory of the system has caused the response spike (responding to the $-\Delta V$ to be smaller (and as it is smaller, it suggests that there is some physical property of the device which has not adjusted to its $+V$
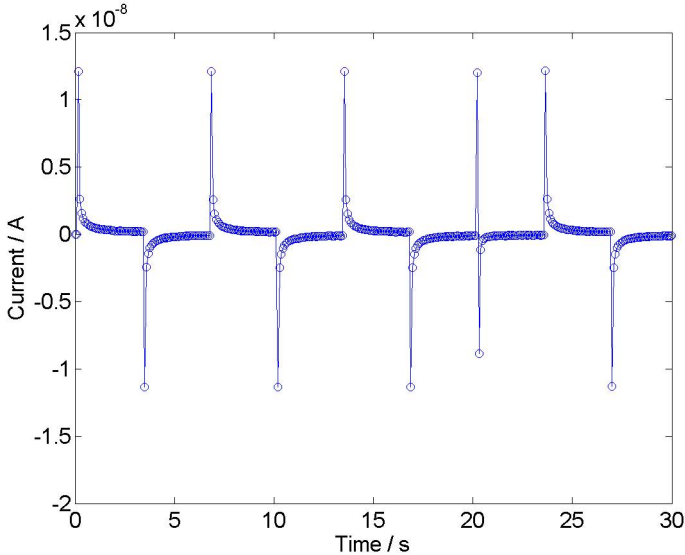
Fig. 1. The effect of adding spikes close in time. The response spikes are the negative current spikes. When a positive spike it included but not allowed to relax the corresponding negative spike is smaller.

value. See [16] for an discussion on why this physical property is predicted to be the oxygen vacancies in the TiO_2.) Thus the response is subtractive in current and additive in resistance state.

To try and understand the subtleties of this apparent 'addition', consider the following system: two voltages are sent to the memristor, one after the other separated by one timestep (i.e. before the memristor has equilibrated), where $V_B > V_A$ and $V_B = 0.12V$, and figure 3 shows the size of the two resulting spikes as a function of increasing V_A. We look at two situations: 1, $V_A(t) \to V_B(t+1)$; 2, $V_B(t) \to V_A(t+1)$. These two situations are drastically different if we look at the transitions, ΔV, as situation 2 has a negative $\Delta V_{B \to A}$, all the other transitions are positive. Situation 1 shows that if the smaller voltage is sent first $(V_A \to V_B)$, the current of the first transition $\Delta i_{0 \to A}$ increases with the size of V_A, and the second transition $\Delta i_{A \to B}$ decreases with the size of ΔV_A, due to the decrease in the effective $\Delta V_{A \to B}$. However, the sum of these two effects is non-linear, so that the total current transferred (approximated as the sum of the spikes here, but actually the area under the two current transients) is not the same as that shown for situation 2 (until $V_B = V_A$). This shows that more current is being transferred and demonstrates that the spikes are dependent on ΔV. Furthermore, it makes it clear that $\Delta i_{0 \to A} + \Delta i_{A \to B} \neq \Delta i_{0 \to B} + \Delta i_{B \to A}$, (except in the trivial case where $V_B = V_A$) and that spike based 'addition' is non-commutative and therefore the order in which the spikes are sent is relevent.
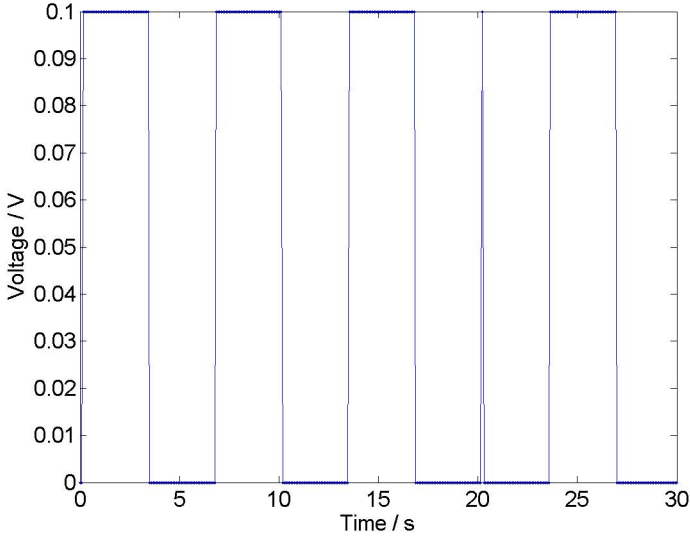
Fig. 2. The input voltage for Test 1

4   Boolean Logic Using Current Spikes in Memristors

We can do Boolean logic with the spike interactions by sending the second bit of information one timestep (0.02s) after the first. We take the input as the current spikes from the voltage level. The output is the response current as measured after the 2nd bit of information. After a logic operation the device is zeroed by being taken to 0V for approximately 4s, and this removes the memristor's memory.

   We have some freedom in how we assign the '1' and '0' states to device properties and these give different logic. The following examples will demonstrate some approaches and build an OR gate or an XOR gate.

4.1   OR Gate

The truth table for an OR gate is given in table 1, essentially, the output should be '1' if either of the inputs was '1'. We take the '0' output as being below a threshold current and the '1' output as being above a threshold. The threshold is set to >18nA with the '0' input being set of 0.01V and the '1' as 0.2V [2], which gives the voltages below:

---

[2] Using '0' as 0V was also tested, it works and is lower power but was not chosen as an example as it is a trivial case.
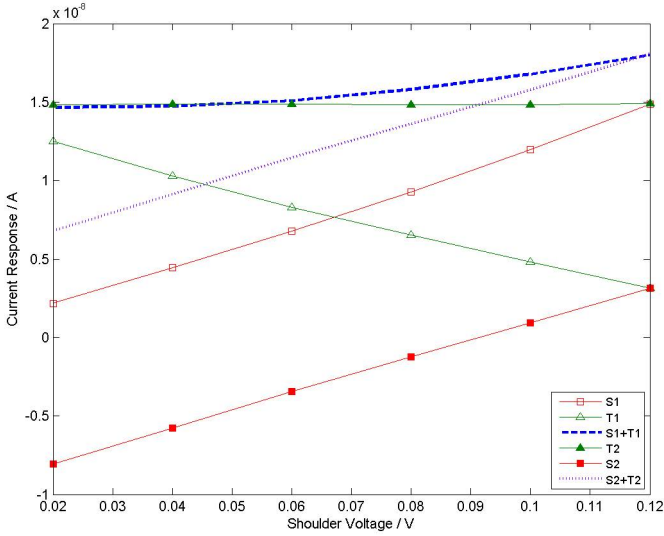
Fig. 3. The effects of the order the spikes are sent in to show that spike addition is non-commutative. S1=$\Delta i_{0\rightarrow A}(t)$, T1=$i_{A\rightarrow B}(t+1)$, T2=$\Delta i_{0\rightarrow B}(t)$ and S2=$\Delta i_{B\rightarrow A}(t+1)$. S1 and T1 refer to the shoulder (S1) and peak (T1) currents resulting switching from $0V \rightarrow V_A \rightarrow V_B$. T2 and S2 refer to the peak (T2) and shoulder (S2) of switching from $0V \rightarrow V_B \rightarrow V_A$. In both cases $V_B > V_A$.

Table 1. OR Truth Table (inclusive OR)

Input 1	Input 2	Output
0	0	0
0	1	1
1	0	1
1	1	1

- 0, 0 = 0.01V, 0.01V
- 0, 1 = 0.01V, 0.2V
- 1, 0 = 0.2V, 0.01V
- 1, 1 = 0.2V, 0.2V.

Figure 4 shows the current data from the voltage inputs above. It can be seen that when a '1' is input, there is a large spike output. To read the logical state of the device, one merely takes the current value as the second bit is read in.

4.2   A Logical System to Create an XOR Gate

The XOR truth table is shown in table 2. If we take logical '1' to be the current resulting from a positive voltage and a logical '0' to be the current resulting from
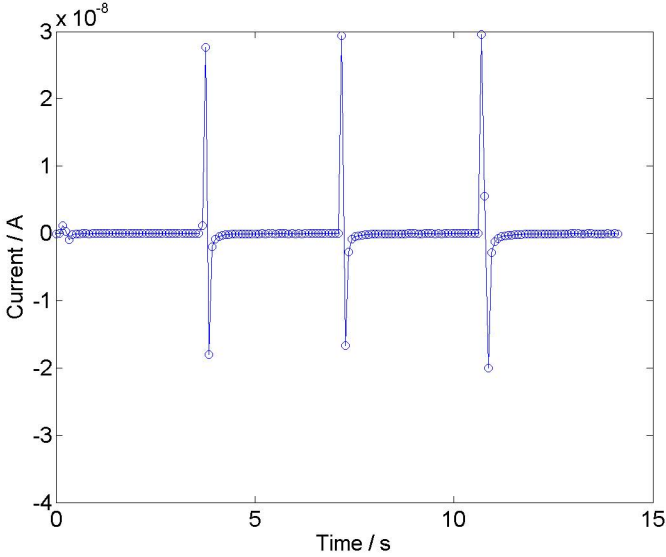
Fig. 4. OR Gate. Using '1' equal to a current spike caused by a voltage change to 0.2V and '0' equal to a current spike caused by a voltage change to 0.01V we can make a serial OR gate (where logical 1 is considered to be a current which is more than 5nA). At 0.04s '0, 0' was input, giving peaks below the threshold i.e. '0' as an output. The three large peaks are '1' outputs resulting from '0,1','1,0' and '1,1' inputs.

a negative voltage, then, the response is the current when the 2nd bit is input (not after, although it could be designed that way but it is slower). We get a high absolute value of current if and only if the two inputs are of different signs, i.e. we have 1 0 or 0 1 which gives us an exclusive OR operation. For this logical system, we used the same voltage level and allowed a change in sign to indicate logical zero or logical one:

− 0, 0 = -0.1V, -0.1V
− 0, 1 = -0.1V, +0.1V
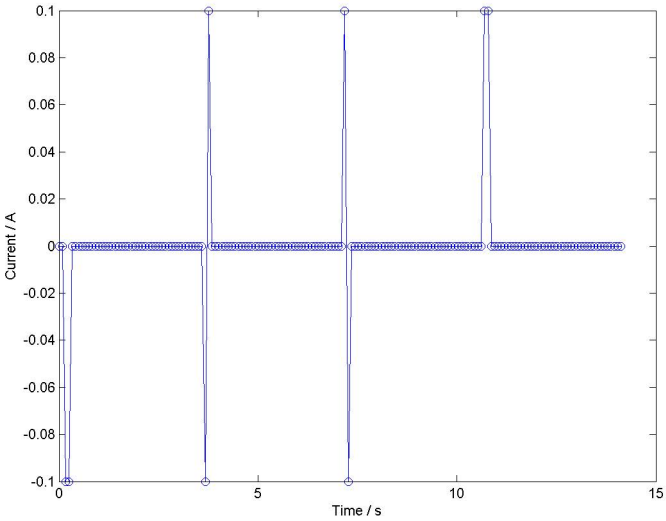− 1, 0 = +0.1V, -0.1V
− 1, 1 = +0.1V, +0.1V.

As an example, the input voltage is shown in figure 6 and the current output is shown in figure 5.

Table 2. XOR Truth Table (exclusive OR)

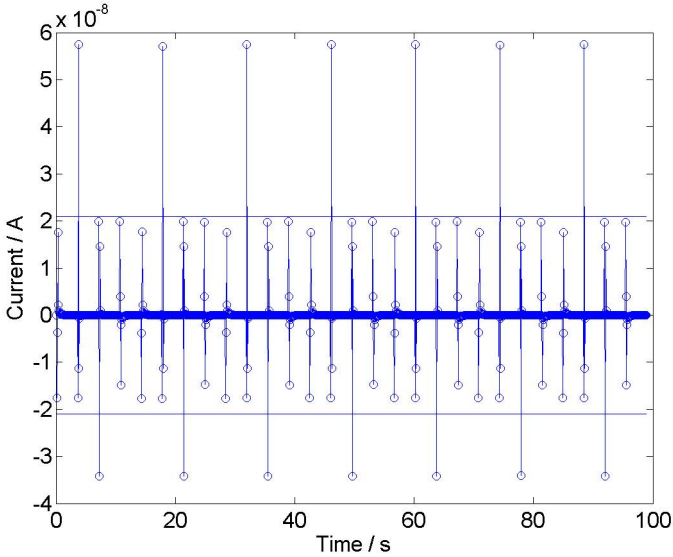| Input 1	Input 2	Output
0	0	0
0	1	1
1	0	1
1	1	0
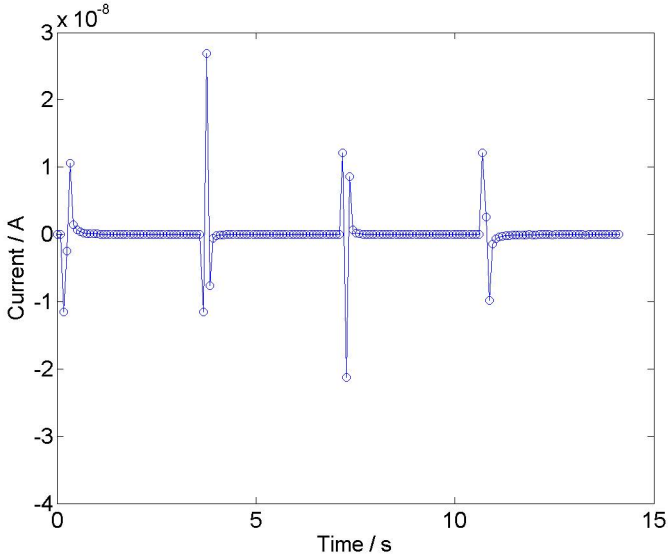
Fig. 5. XOR gate, where a current response over $\pm 1.25 \times 10^{-8}$A is taken as one, as current response under that threshold is taken as zero

Fig. 6. The programming voltage for the XOR gate

Fig. 7. Reproducabiltiy test of XOR function. Here the XOR truth table is run 7 times. The threshold between '1' and '0' is marked as shown.

With a pause between operations to allow the memristor to lose its memory, the XOR operation is reproducible, as shown in figure 7.

As XOR A = NOT A, if we always take the 2nd point after the first (and only bit in this case) as being the response bit (as we did above for the XOR gate), we have a NOT gate.

5   Conclusions

This type of approach is a serial logic gate where the bits are separated in time. This allows us to do logic operations with one memristor at the speed of the spikes (fast) rather than at the speed of equilibration (slow). This approach also allows us to do logic with a two terminal (one-port) device, the extra 'complexity' of the operation is contained within the time domain. Essentially, we use the memristor's short-term memory to hold the first bit and do the calculation. This demonstrates that memristors can act as the processor and memory store in one. It also shows the bizarre property of the memory in a system being used to perform memoryless logic.

The memristor is acting similarly to a sequential logic circuit, where the combinatorial logic is combined with the memory store. Furthermore, the memristor logic gate is asynchronous because there is no need for a clock pulse, but there are issues of race hazard because the second bit must arrive within the time window of the memristor's memory.

The speed of the these operations is not too fast in this proof-of-principle, however, this is because of the speed at which the electrometer can properly measure a current response. Circuit theory suggests that these spikes should exist at shorter times, so we are confident that the devices can be sped up by sending the second spike in faster.

The memristor is very low power, especially if operated at the voltages and currents shown in the paper (it is possible to work at higher voltages if desired).

At the moment the output is a different circuit measurable to the input (i.e. the output is current and the input is voltage), it is necessary to convert from one to the other to enable the creation of logical circuits. However, we expect that a current pulse should propagate through a circuit [17] and cause a change in voltage across the next memristor, which could then be used to do the next operation and thus allow the creation of larger memristor logical circuits. We plan to do further work on testing this and investigating the possibility of using a second memristor as a $V \rightarrow I$ transformer (based on the fact that previous ΔV produced a ΔI).

References

1. Strukov, D.B., Snider, G.S., Stewart, D.R., Williams, R.S.: The Missing Memristor Found. Nature 453, 80–83 (2008)
2. Chua, L.O.: Memristor - The Missing Circuit Element. IEEE Trans. Circuit Theory 18, 507–519 (1971)
3. Chua, L.O.: Resistance Switching Memories are Memristors. Applied Physics A: Materials Science & Processing, 765–782 (2011)
4. Waser, R.: Nanoelectronics and Information Technology. Wiley-VCH Verlag GmbH & Co. KGaA, Weinheim (2003)
5. Cantley, K., Subramaniam, A., Stiegler, H., Chapman, R., Vogel, E.: Hebbian Learning in Spiking Neural Networks with Nano-Crystalline Silicon TFTs and Memristive Synapses. IEEE Tran. Nanotechnology 10, 1066–1073 (2011)
6. Zamarreno-Ramos, C., Carmuñas, L.A., Pérez-Carrasco, J.A., Masquelier, T., Serrano-Gotarredona, T., Linares-Barranco, B.: On Spike-Timing Dependent Plasticity, Memristive Devices and Building a Self-Learning Visual Cortex. Frontiers in Neuromorphic Engineering 5, 26(1)–26(20) (2011)
7. Howard, G.D., Gale, E., Bull, L., de Lacy Costello, B., Adamatzky, A.: Evolution of Plastic Learning in Spiking Networks via Memristive Connection. IEEE Trans. Evolutionary Computation 26, 711–719 (2012)
8. Chua, L., Sbitnev, V., Kim, H.: Hodgkin-Huxley Axon is made of Memristors. Int. J. Bifur. Chaos 22, 1230011(1)–1230011(48) (2012)
9. Chua, L., Sbitnev, V., Kim, H.: Neurons are Poised Near the Edge of Chaos. Int. J. Bifur. Chaos 22, 1250098(1)–1250098(49) (2012)
10. Gale, E.M., de Lacy Costello, B., Adamatzky, A.: Observation and Characterization of Memristor Current Spikes and their Application to Neuromorphic Computation. In: 2012 International Conference on Numerical Analysis and Applied Mathematics, ICNAAM 2012. AIP Conference Proceedings, vol. 1479, pp. 1898–1901. AIP Melvil, New York (2012)

11. Borghetti, J., Snider, G.D., Kuekes, P.J., Yang, J.J., Stewart, D.R., Williams, R.S.: 'Memristive' switches enable 'stateful' logic operations via material implication. Nature 464, 873–876 (2010)
12. Pershin, Y.V., di Ventra, M.: Neuromorphic, Digital and Quantum Computation with Memory Circuit Elements. Proc. IEEE 100, 2071–2080 (2012)
13. Pino, R.E., Bohl, J.W.: Self-Reconfigurable Memristor-Based Analog Resonant Computer, US Patent, US 8,274,312 B2
14. Gale, E., Pearson, D., Kitson, S., Adamatzky, A., de Lacy Costello, B.: Aluminium Electrodes Effect the Operation f Titanium Dioxide Sol-Gel Memristors, arXiv:1106:6293v1 (cond-mat.mtrl-ci),
    `http://arxiv.org/abs/1106.6293`
15. Gale, E., Mayne, R., Adamatzky, A., de Lacy Costello, B.: Drop-coated Titanium Dioxide Memristors, arXiv:1205:2885v2 (cond-mat.mtrl-ci),
    `http://arxiv.org/abs/1205.2885`
16. Gale, E.: The Missing Magnetic Flux in the HP Memristor Found, arXiv:1106:3170v1 (cond-mat.mtrl-ci), `http://arxiv.org/abs/1106.3170`
17. Gale, E., Matthews, O., de Lacy Costello, B., Adamatzky, A.: Beyond Markov Chains, Towards Adaptive Memristor Network-based Music Generation. In: Proceedings of the Annual Convention of Society of the study of Artificial Intelligence and the Simulation of Behaviour, AISB 2013, vol. 8, pp. 28–49 (2013)