Memristor-based Synaptic Networks and Logical Operations Using In-Situ Computing

Omid Kavehei [1], Said Al-Sarawi [1], Kyoung-Rok Cho [2], Nicolangelo Iannella [1], Sung-Jin Kim [2],
Kamran Eshraghian [2], Derek Abbott [1]

[1] *School of Electrical & Electronic Engineering, The University of Adelaide, SA 5005, Australia*
[2] *College of Electronics & Information Engineering (WCU Program), Chungbuk National University, South Korea*
omid@eleceng.adelaide.edu.au

***Abstract*—We present new computational building blocks based on memristive devices. These blocks, can be used to implement either supervised or unsupervised learning modules. This is achieved using a crosspoint architecture which is an efficient array implementation for nanoscale two-terminal memristive devices. Based on these blocks and an experimentally verified SPICE macromodel for the memristor, we demonstrate that firstly, the Spike-Timing-Dependent Plasticity (STDP) can be implemented by a single memristor device and secondly, a memristor-based competitive Hebbian learning through STDP using a 1×1000 synaptic network. This is achieved by adjusting the memristor's conductance values (weights) as a function of the timing difference between presynaptic and postsynaptic spikes. These implementations have a number of shortcomings due to the memristor's characteristics such as memory decay, highly nonlinear switching behaviour as a function of applied voltage/current, and functional uniformity. These shortcomings can be addressed by utilising a mixed gates that can be used in conjunction with the analogue behaviour for biomimetic computation. The digital implementations in this paper use in-situ computational capability of the memristor.**

I. Introduction

The classical von Neumann machine suffers from a large sequential (fetch-execute-store cycle) processing overload due to the existence of the data bus between memory and logic. Neuromorphic engineering introduces a more efficient (event driven) implementation but not necessarily low-power. Software techniques are power hungry and there traditionally has been was no low-power hardware device (switch) to provide tighter coupling between memory and logic, as in biological systems. The memristor is an emerging technology that combines (non-volatile) memory and in-situ computational characteristics in one device in the way that promises an entirely new computer architecture.

The mathematical foundation of the memristor, as the fourth fundamental passive element, has been expounded by Leon Chua [1] and later extended to a broader class, known as memristive devices and systems [2]. This broad classification today includes all resistance switching memory devices [3]. Realisation of a solid-state memristor in 2008 [4] has generated a new wave of research in realization of both large memory arrays as well as new thinking in the neuromorphic engineering domain. Memristors (the term *memristor* is a portmanteau of *memory* and *resistor*) are capable of encoding information in two or more stable levels each with relatively long decay times. The decay can be long in human terms (e.g. days and weeks), which is a practical implementation of a non-volatile memory [2], [4]–[7]. It has also been experimentally proven -in small scale- that these two-terminal memristive devices are able to carry out logic operations [8]. Therefore, memristor is a possible option for implementing a tighter coupling between memory and logic technologies.

There are many memristor-based applications. The obvious application of such a nanometer scale device is in implementing non-volatile, low-power, and dense memory arrays. Owing to the multi-stable state property and the relatively long term decay, memristors are also able to encode synaptic weights [9]. Furthermore, several possibilities for neuromorphic engineering domain and learning have been also studied [6], [10]–[12]. In this paper we demonstrate very basic analogue and digital circuits that are implemented in memristor technology.

Contributions that this paper provides can be categorised as follow:

- Brief characterisation of memristor for neuromorphic purposes.
- Experimental results demonstrating the multi-stable state of a silver/titanium dioxide/indium tin oxide (Ag/TiO$_2$/ITO).
- Demonstration of the use of memristor as a synaptic connection that mimics the Spike-Timing Dependent Plasticity (STDP) rule.
- Show a memristor-based competitive Hebbian learning through STDP.
- Circuit for analogue multiplication and accumulation using fixed weights pattern.
- Experimental show that a sharp switching behaviour in a fabricated Ag/TiO$_2$/ITO and Pt/TiO$_2$/Pt (Pt: Platinum) memristors as well as state decay. Demonstrating a memristive-based analogue computing.
- CRS-based logic gates through material implication and PLA implementations.

Note that memristor and memristive device characteristics, modelling, materials, and underlying physics are not within the scope of this paper. The interested reader can find further details in [1], [5], [10], [13], [14] for further details. The simulations carried out in this work using SPICE macro-model implementation of presented model in [13].

978-1-4577-0674-5/11/$26.00 © 2011 IEEE

II. Memristor model

Memristor device characteristics can be defined using a system of two equations,

$$\begin{cases} I = g(w, V) \cdot V \\ \frac{\mathrm{d}w}{\mathrm{d}t} = f(w, V) \ , \end{cases} \tag{1}$$

where w is a physical variable indicating the internal memristor state that in theory is such that $0 < w < L$, L is the thickness of a thin-film metal-oxide (memristive) material sandwiched between two metallic electrodes, and I and V represent current and voltage, respectively. The $g(\cdot)$ function represents the memristor's conductance. The state variable can be expressed using a normalised form $x = 1 - w/L$. In this case, $w \to 0$ or moving towards higher conductances can be expressed as $x \to 1$ and $w \to L$ or moving towards lower conductances can be shown as $x \to 0$. In this paper, R_{HRS} represents high resistance state and R_{LRS} shows low resistance state. Eq. (1) shows that the output of the system (here I), at a given time, depends on w and V. State transition conditions are also explained by the function $f(\cdot)$. To measure this function several time-domain experiments for I and V are required. According to our measurements, a $\sinh(\cdot)$ like behaviour can explain dynamics of the device while an additional term is needed. The $\sinh(\cdot)$ term defines the dependency of velocity, dw/dt, to the effective applied electric field that has been described as an ionic crystal behaviour in an external electric field [15]. The additional term highlights the dependency of conductance, G_t, to the previous conductance, G_{t-1}. Intuitively, we use an exponential form function $h(w)$ to define dw/dt as a function of w based on Fig. 3 in [16]. The $h(w)$ function then should be multiplied by the $\sinh(\cdot)$. The conductance behaviour as a function of w is also shown in Fig 2 of [14]. Due to the asymmetric behaviour of $w \to 0$ and $w \to L$ [16], we have used two different $h(w)$ definition to address a more accurate switching properties [14], [16].

The state variable equation then can be defined as

$$\frac{dw}{dt} = h(w) \sum_i \upsilon_i V^i + d(w), \tag{2}$$

where υ_i are coefficients for low and high electric fields. The index, i, is an positive odd integer so it is the expansion of $\sinh(\cdot)$. This demonstration help to easily extract linear approximation of the memristor model in [4] and also combine effects of Joule heating and $L - w$ (the effective electric field) in the coefficients [13]. The function $d(w)$ represents the decay term which can be weeks, months, or more. The decay term appears to be very similar to synaptic weight update (learning) rule [6], [11]. The first term of Eq. (2), represents a voltage dependent, highly nonlinear which makes high-speed digital computing possible. This property originated from the fact that resistance modulation inside the metal-oxide occurs via electron-ion interactions. This term creates a significant problem for learning applications in the current form.

To compensate this problem we have to take advantage of its high nonlinearity. This nonlinear behaviour produces a threshold-like region that voltages below that threshold does not change the conductance. Considering the fact that, memristor's conductance, G, can be tuned by a series of voltage pulses with appropriate pulse widths and a voltage around the threshold, obviously, pulse time is the other parameter involved. Applying a voltage around the threshold slightly changes x (or w) if it is maintained for a few μs. It is observed that such voltage cannot change the state if the duration is around a few ns. However, slightly increase in the applied voltage increases the speed by several orders of magnitude, which makes nanosecond (digital) switching possible. Therefore, a series of few μs pulses with an appropriate pulse shape can be used to mimic learning rule [12].

III. Analogue Memory and Computing

A. Muti-stable state memory

Here we demonstrate such behaviour in Ag/TiO$_2$/ITO experiment, which is an identification for existence of an ionic drift. Fig. 1 illustrates the existence of the multi-stable memory levels. The experiments carried out using a Keithley 4200-SCS. Triangular input voltage was swept from 0 V to -0.9 V and vice versa. Current compliance of 500 μA was applied to avoid any damage to the device. At the end of each cycle device was disconnected from inputs.

The most critical limitation of analogue memristor is its state decay. Although many stable state can be observed, our measurements for five conductance levels showed decay distribution ranging from a few hours up to a few days. More measurements were not possible with our limited time.

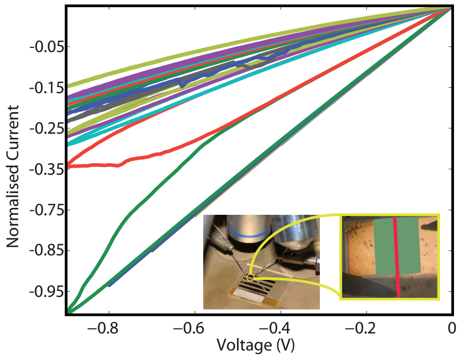


Fig. 1. Memristor analogue behaviour. Experimental result from Ag/TiO$_2$/ITO memristor. Current values are normalised to their maximum value (35 μA). Inset shows a Device Under Test (DUT). The red and the green areas highlight a memristor device.

B. Memristive, plasticity, and learning

The connection can be drawn between memristive devices and biological synaptic update rule, known as STDP, that has been observed in the brain [9]. This can be achieved by collecting data from a memristive device based on the time difference, Δt, between two signals, so called pre- and post-synaptic signals. The results are shown in Fig. 2 (a), which shows how the Device Under Test (DUT) weight (resistance) changes as a function of Δt. The intermediate states vanish after a certain decay duration whereas a significantly higher potentiation ($x \rightarrow 1$) will be kept as a long term memory. So, the existence of intermediate states decay helps in mimicking the long-term potentiation and short-term plasticity (LTP and LTP) behaviour [17].

The collected information is then used as stimuli for a network of 1×1000 memristors are connected to one neuron being implemented and pre- and post-synaptic spikes shape is the same as [12], then this network implements the competitive Hebbian learning [18]. Initial states have been shown in Fig. 2 (c) in red. Intentionally, a Gaussian distribution has been employed for the memristors' initial state values. After running the simulation for 35 minutes, the network results in a population distribution of weights similar to a previously published competitive Hebbian learning rules [18]. The additive and multiplicative features of a memristive network strictly depends on the device and its nonlinearity parameters. Fig. 2 (b) demonstrates a Poissonian ISI distribution.

C. Programmable analogue circuits

Although plasticity plays an important role for adaptation and development, networks with fixed synaptic weight pattern should be also studied. Therefore, one of the challenges for this emerging technology is to integrate learning and unlearning hardware as part of a neural computational platform. Since memristors possess a threshold-like behaviour, usually low- or very low-voltage operations do not change the memristor's initial state. This fact helps developing programmable analogue computing circuits [19]. There is also a similar design in [20]. The is no simulation or experimental result.

Here, we introduce the use of a memristive array for implementing a multiplication of inputs and the memristor's internal state, w, which represents the memristor's conductance. Fig. 3 (a) illustrates a single row of the array and Fig. 3 (b) shows its simulation results for two elements, M1 and M2, connected to two inputs, In_1 and In_2. In this case, we first applied a voltage pulse to M1 to read its conductance, then a pulse to M2 for the same reason. When two voltage pulses are simultaneously applied to M1 and M2, accumulation operation can be clearly observed.

D. Existence of a threshold-like switching

In this part, we show the existence of a switching threshold in a TiO_2-based memristor. According to Chua's definition [1], memristor links electrical charge to flux, φ, and $\varphi = \int V dt$. Therefore, the amount of flux passing through the device can be controlled by V and/or time. So, low pulse widths should not change the conductance if the voltage is lower than a certain value and small voltages similarly do not change the conductance if the applied pulse width is not sufficient. The analysis started from the amorphous (RESET) state and a crystallisation window created above 0.8 V and 100 μs. Fig. 4 illustrates the results from a Pt/TiO_2/Pt memristor. It is observed that the area of crystallisation window decreases as R_{HRS} increases in different devices [14].

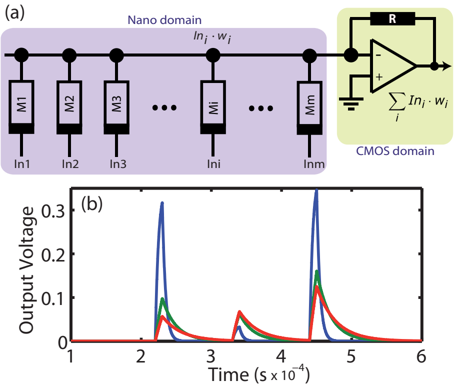


Fig. 3. Multiply-accumulate module. (a) Shows a single row of multiply elements (memristors), $\mathrm{In}_i \cdot w_i$. (b) Demonstrates simulation results for two memristors, M1 and M2. In this simulation, memristor M2 programmed at $x = 0.5$, which is equivalent to $(R_{\mathrm{HRS}} + R_{\mathrm{LRS}})/2$. Then memristor M1 changes its resistance from R_{LRS} to R_{HRS} in three steps. Each step is a simulation that is shown with different colours. Blue for $R_{\mathrm{M1}} = R_{\mathrm{LRS}}$, green for R_{M1} close to $(R_{\mathrm{HRS}} + R_{\mathrm{LRS}})/2$, and red for R_{M1} close to R_{HRS}. The summing amplifier can be replaced by any thresholding module for different applications.

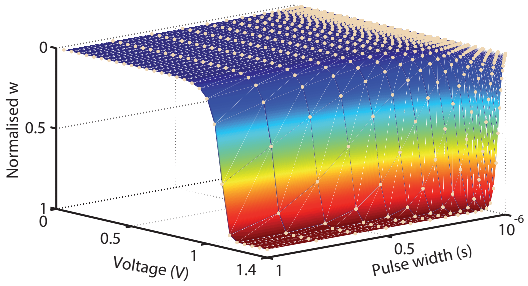


Fig. 4. Existance of a switching threshold in the memristor material. The pulse widths are from 10 μs to 1 s.

IV. Digital In-Situ Computing

The existence of the sharp switching threshold, functional uniformity, intermediate state initialisation, and most importantly state decay creates several problems that can be eliminated or compensated for by using the memristor device as a binary switch.

A. Complementary Resistive Switch (CRS)

Although the memristor has introduced new possibilities and it is very well adapted in a crossbar architecture, the inherent interfering current paths between neighbouring cells of an addressed cell impose limitations on the array scalability [13], [21]. A possible solution is to build a diode or a transistor in series with a memristor. Using transistors adds other scalability issues due to the fact that transistors are not very well stackable and the application of diodes imposes a high drive current limiting the use of such array in an ultra-low-power applications.

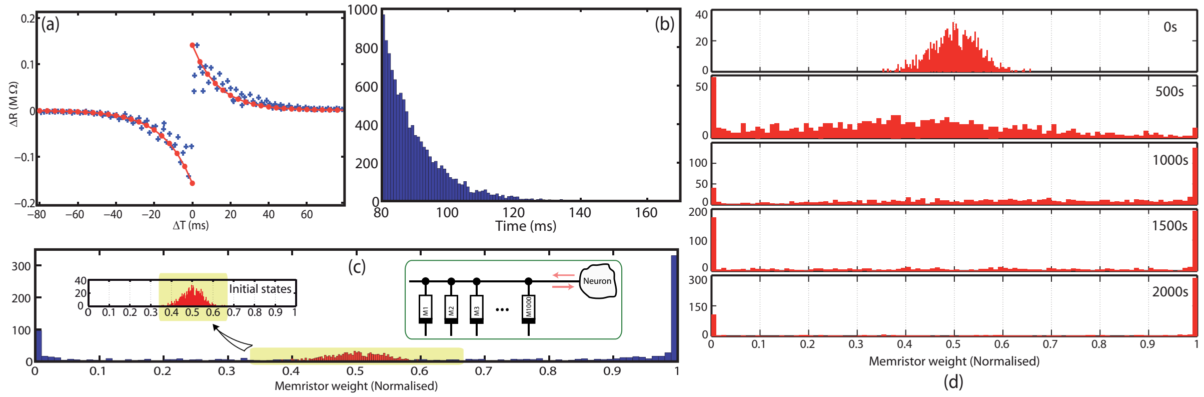


Fig. 2. Memristor, plasticity, and competitive learning. (a) Dots illustrate experimental data and the solid (red) line shows the fitting STDP rule. We exclude devices that reach lowest and highest conductances in depression and potentiation processes, respectively, because they add no extra information to our analysis. (b) A Poisson inter-spike interval (ISI) distribution for 1×1000 memristors (synapse) connected to one neuron, inset in (c). (c) Illustrates simulation results of such network. It is clear that it follow the competitive learning behaviour reported in [18]. (d) Evolution of synaptic strength from 0 s to 2000 s.

Linn *et al.* [21] introduced a new paradigm by exploiting two anti-serially (with opposite polarities) connected memristors. The structure is similar to a *memistor* (note the missing "r") [22]–[24]. A (digital) CRS uses a combination of a High Resistance State (HRS) and a Low Resistance State (LRS) to encode logic "0" and logic "1". Consequently, the overall resistance of such device is always around HRS, resulting in significant reduction in the parasitic current paths through neighbouring devices. Fig. 5 (a) summaries the CRS states. If p and q indicate resistances of the memristors M1 and M2, respectively, four different states can be observed. For example, $p/q \leftarrow$L/H indicates that LRS is written in p (memristor M1) and HRS in q (memristor M2). Combinations L/H and H/L for p and q represent logic "1" and logic "0", respectively. Note that the H/H state only occurs once in a "fresh" device. According to Fig. 5 (c) any transition between the states occurs if the applied voltage exceed the SET thresholds, $V_{\mathrm{th,S1}}$ or $V_{\mathrm{th,S2}}$ and the device's initial state supports the transition. Possible state transitions are shown in Table I, where p'/q' shows the next state, p/q illustrates the initial state, and output is a current pulse or spike. These outputs enable us to have two different read-out mechanisms, logic→ON or logic→logic.

The transitions in Table I can be defined using *material implication* logic [8], [25]. It has been proven that implication and FALSE operation are a complete set for logical operations [26]. This logical operation results in change of q depending on the state of p (or vise versa), known as p IMP q, 'p implies q' or 'if p then q'. Therefore, if we represent p NIMP q it means 'p not implies q', Table I (i), for example, represents $q \leftarrow$H and we say the conditions (initial p/q and ΔV) not implies q.

The destructive read-out should not be a problem for two reasons: (1) refreshing a digital memory is a normal task depends on the decay term and (2) there are no alternative available to combine the CRS properties and a non-destructive read-out.

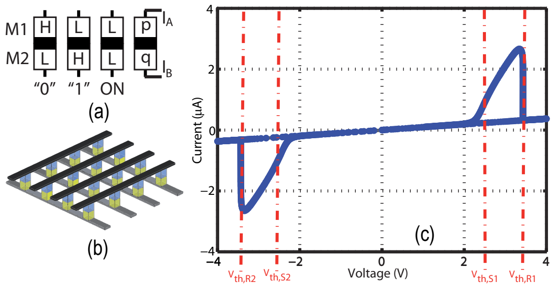


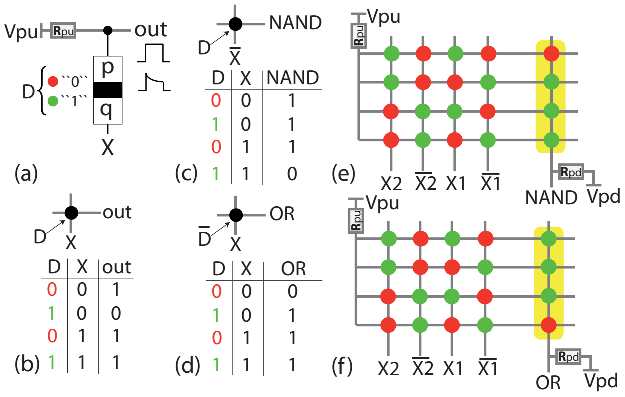
Fig. 5. CRS device structure and logical definition of each combination. (a) demonstrates all operational states, (b) illustrates the crossbar view, and (c) shows CRS functionality.

TABLE I
STATE TRANSITIONS IN A CRS

	p/q	$\Delta V = V_{\mathrm{I_A}} - V_{\mathrm{I_B}}$	p'/q'	Output
i)	"1"	$V_{\mathrm{th,S1}} < \Delta V < V_{\mathrm{th,R1}}$	ON	pulse
ii)	"1"	$V_{\mathrm{th,R1}} < \Delta V$	"0"	spike
iii)	"0"	$V_{\mathrm{th,R2}} < \Delta V < V_{\mathrm{th,S2}}$	ON	pulse
iv)	"0"	$\Delta V < V_{\mathrm{th,R2}}$	"1"	spike
v)	ON	$V_{\mathrm{th,R1}} < \Delta V$	"0"	–
vi)	ON	$\Delta V < V_{\mathrm{th,R2}}$	"1"	–

B. CRS-based logical operations

Here, we introduce CRS-based logical operation and PLA (programmable logic array) that works with the two transitions, logic→logic and logic→ON, but we only present it with the later transition. The idea is to charge a bit-line in a crossbar array, and applying inputs to its word lines. The inherent implication property of the device causes a change under certain conditions that we have already discussed. In [25], AND and NOR operations are proposed using the logic→logic transition and current spike read-out process. This method is very dependent on the current spike which occurs by a transient ON state between two logic states. In their implementation, two combinations have been evaluated out of two possible combinations for two CRS devices. Assume voltage, ΔV, is applied across a CRS device that is exceeded its RESET threshold, in this situation this device changes its stored logic, D, if D is a certain logic depends on the signature of ΔV. Furthermore, if two CRS devices are connected together, that intermediate point can be connected to either ground or power supply to generate NOR/AND gate. That is the reason that no more possible state can be assumed using such approach.



D	X	out
0	0	1
1	0	0
0	1	1
1	1	1

D	X	NAND
0	0	1
1	0	1
0	1	1
1	1	0

D	X	OR
0	0	0
1	0	1
0	1	1
1	1	1

Fig. 6. CRS-based logic gate structures. (a) D represents stored data, X is an input, and R_{pu} is pull-up resistor. The output is initially charged and it is discharged depends on D and In. (b) Shows how a not implication, NIMP, can be implemented. Here $q' \leftarrow D$ NIMP X. (c) Two inputs NAND gate is implemented by storing one input as device state and another one as an actual input. Here complementary of signal X is applied to the device. (d) Similar to NAND but complementary of D stored in the CRS and X is applied as an input. Therefore, an OR operation implemented, simply by a single CRS device. Obviously, operations are sequential and they requires one (or several) initialisation but this is a drawback for all of the available Boolean logic operations reported in [8], [25]. Pull-up (charge) voltage is enough to push a device to its ON state and not writing a logic, $V_{\mathrm{th,S}} < V_{\mathrm{pu}} < V_{\mathrm{th,R}}$. NOT function can be also implemented using a single CRS if D stores (the data) A and $X = 0$, $F = \overline{A}$. (e) and (d) are PLA implementation of the two logic gates. Here we remove the outputs' complementary signals, AND and NOR. The yellow highlights show the OR-plane and the rest are in the AND-plane.

Here two comprehensive forms of building logical gates are introduced. The first form, allows storing one or more inputs as device state and the second method does not. Fig. 6 illustrates how CRS works as an implementation of a not implication, NIMP, operation and how NAND and OR operations can be implemented using a single CRS device. Fig. 6 (a)-(d) are well explained in the figure's caption and their operations is also described. Fig. 6 (e) and (f) follow similar phenomenon but in a form of a PLA. The idea is to have a logic→ON transition in the OR-plane whenever an output product term is addressed. From the NIMP operation, we know that if the applied inputs are part of the output product terms, that bit-line does not discharged so there will be enough voltage across the output CRS device with stored logic "1" (greens) to turn to ON and conduct significantly more current to charge the output signal load.

In the case of using differential voltage pairs, $V_{\mathrm{pu}} = -V_{\mathrm{pd}} = 1.4$ V was selected as 2.8 V is the READ voltage (in Fig. 5), where V_{pu} and V_{pd} are pull-up and pull-down voltages. Here we applied $V_{\mathrm{pu}} = 2.8$ V and $V_{\mathrm{pd}} = 0$ V, so we used 0.25 μm CMOS transistors in our CMOS domain. Therefore, equivalent input voltage for logics "1" is 2.8 V and for logics "0" is 0 V. The pull-up and pull-down resistors, R_{pu} and R_{pd}, are both equal to $R_{\mathrm{LRS}}\sqrt{2(r+1)}$, where $r = R_{\mathrm{HRS}}/R_{LRS}$ [13]. The used peripheral CMOS circuitries can be found in [27]. The sense amplifier was designed for voltage sensitivity more than 100 mV.

TABLE II
CRS-BASED LOGIC IMPLEMENTATION WITH TWO INOUTS AND TWO CRSS, $F = \overline{D_1} \cdot \overline{D_2} + \overline{D_1} \cdot X_2 + \overline{D_2} \cdot X_1 + X_1 \cdot X_2$.

D_1	D_2	X_1	X_2	Function
$\overline{A}$	A	0	B	$A \cdot B$ (AND)
A	$\overline{A}$	0	$\overline{B}$	$\overline{A + B}$ (NOR)
A	$\overline{A}$	$\overline{B}$	B	$A \oplus B$ (XOR)
$\overline{A}$	A	$\overline{B}$	B	$A \odot B$ (XNOR)

Assuming we have two inputs, X_1 and X_2, and two CRS devices, D_1 and D_2, connected to these inputs and a charged bit-line. A number of functions can be implemented by writing $\overline{F} = D_1 \cdot \overline{X_1} + D_2 \cdot \overline{X_2}$, hence, $F = \overline{D_1} \cdot \overline{D_2} + \overline{D_1} \cdot X_2 + \overline{D_2} \cdot X_1 + X_1 \cdot X_2$. The first term, $\overline{D_1 D_2}$, indicates that if both CRSs store "0" TRUE ($F = 1$) is implemented. Some other function that is implemented using this configuration are shown in Table II. In [28] we demonstrates a CRS-based content addressable memory based on the XOR/XNOR function. Fig. 7 (a) illustrates simulation of a two input NAND function. The most significant advantage of this method is that the initialisation step (step 1) which is writing data into CRS arrays and not a simple refreshing cycle. While in a PLA structure, Fig. 7 (b), the initialisation is a refreshing cycle. Furthermore, in computer arithmetic operations signals arrive with relative delays, like SUM results and CARRY output, that can be used in parallel with the programming of CRS arrays.

V. Conclusion

This paper introduced basic functional blocks for analogue and digital computation based on memristive devices. It is difficult to have a fair comparison between emerging and the conventional devices as the emerging technologies are at their early stages. Moreover, architectural aspects for future computers seems to be dependent to the concept of universal memory and computational capability of one individual device or nano-system that is entirely different with the classical von Neumann computational framework. Therefore, introducing more compatible circuits and algorithms with these futuristic technologies could play an important role. This work presented the existence of ionic drift in the fabricated memristors. We have also illustrated how the memristor can be used to implement competitive Hebbian learning (additive STDP). An analogue multiply-accumulation circuit was introduced that is able to implement a low precision multiplication and addition. This circuit combines inherent non-volatile memory

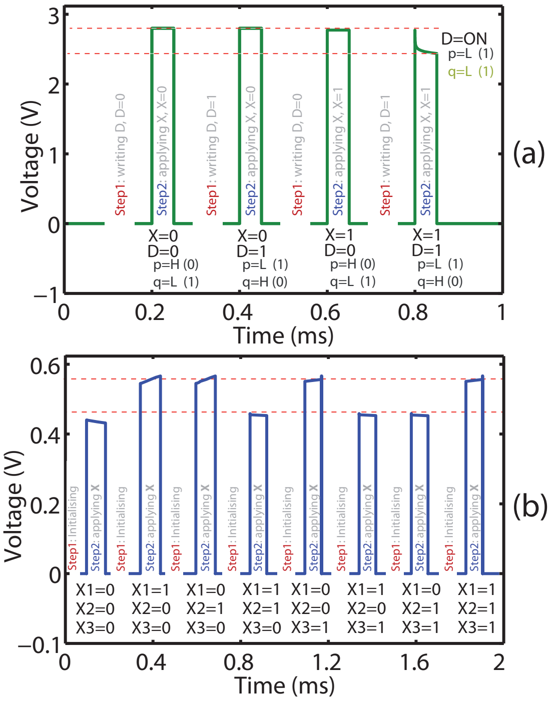


Fig. 7. CRS-based logic gate simulations. (a) A 2-input NAND gate (Fig. 6 (c)) simulation. In this style, we are allowed to store one input as the CRS state. (b) A 3-input XOR (SUM) function, implemented in a PLA structure. In both cases, (a) and (b), dashed red line show worst-case low and high output voltages that are sent to sense amplifiers. Due to limited space, complementary output, XNOR, is not shown. Initialisation in (b) means, the array should be initialised before the next logical operation and this is the main reason that the first approach (in (a)) is a far more efficient implementation in terms of both hardware and number of steps. No initialisation is required in (a), because 'writing *D*' effectively means writing one of the input's logic into the device.

and dynamics of a memristor as a synapse. The problem of state decay then results in developing a digital version of such learning system which is out of the scope for this paper. However, the idea of digital computing using a more robust memristive device, CRS, was explained and two methods for implementing logical blocks were introduced.

Acknowledgement

This work was supported by Australian Research Council (ARC) and grant No.R 33-2008-000-1040-0 from the World Class University project of MEST and KOSEF through Chungbuk National University.

References

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

[2] L. O. Chua and S. M. Kang, "Memristive devices and systems," *Proceedings of the IEEE*, vol. 64, no. 2, pp. 209–223, 1976.

[3] L. Chua, "Resistance switching memories are memristors," *Applied Physics A: Materials Science & Processing*, vol. 102, pp. 765–783, 2011.

[4] D. B. Strukov, G. S. Snider, D. R. Stewart, and R. S. Williams, "The missing memristor found," *Nature*, vol. 453, pp. 80–83, 2008.

[5] O. Kavehei, A. Iqbal, Y. Kim, K. Eshraghian, S. Al-Sarawi, and D. Abbott, "The fourth element: characteristics, modelling and electromagnetic theory of the memristor," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Science*, vol. 466, no. 2120, p. 2175, 2010.

[6] G. S. Snider, "Instar and outstar learning with memristive nanodevices," *Nanotechnology*, vol. 22, art no. 015201, 2011.

[7] T. Hasegawa, T. Ohno, K. Terabe, T. Tsuruoka, T. Nakayama, J. Gimzewski, and M. Aono, "Learning abilities achieved by a single solid-state atomic switch," *Advanced Materials*, vol. 22, no. 16, pp. 1831–1834, 2010.

[8] J. Borghetti, G. Snider, P. Kuekes, J. Yang, D. Stewart, and R. Williams, "'Memristive' switches enable 'stateful' logic operations via material implication," *Nature*, vol. 464, no. 7290, pp. 873–876, 2010.

[9] S. Jo, T. Chang, I. Ebong, B. Bhadviya, P. Mazumder, and W. Lu, "Nanoscale memristor device as synapse in neuromorphic systems," *Nano Letters*, vol. 10, no. 4, pp. 1297–1301, 2010.

[10] Y. Pershin and M. Di Ventra, "Memory effects in complex materials and nanoscale systems," *Advances in Physics*, vol. 60, no. 2, pp. 145–227, 2011.

[11] G. S. Snider, "Cortical computing with memristive nanodevices," *Sci-DAC Review*, vol. 10, pp. 58–65, 2008.

[12] C. Zamarreño-Ramos, L. Camuñas-Mesa, J. Pérez-Carrasco, T. Masquelier, T. Serrano-Gotarredona, and B. Linares-Barranco, "On spike-timing-dependent-plasticity, memristive devices, and building a self-learning visual cortex," *Frontiers in neuroscience*, vol. 5, 2011.

[13] O. Kavehei, S. Al-Sarawi, K. Cho, K. Eshraghian, and D. Abbott, "An analytical approach for memristive nanoarchitectures," *arXiv preprint:1106.2927*, 2011.

[14] O. Kavehei, K. Cho, S. Lee, S. Kim, S. Al-Sarawi, D. Abbott, and K. Eshraghian, "Fabrication and modeling of Ag/$\mathrm{TiO_2}$/ITO memristor," in *54th IEEE International Midwest Symposium on Circuits and Systems (MWSCAS)*, 2011.

[15] N. Mott and R. Gurney, *Electronic processes in ionic crystals*. Dover, 1964, ch. 2.

[16] M. D. Pickett, D. B. Strukov, J. L. Borghetti, J. J. Yang, G. S. Snider, D. R. Stewart, and R. S. Williams, "Switching dynamics in titanium dioxide memristive devices," *Journal of Applied Physics*, vol. 106, no. 7, art no. 074508, 2009.

[17] T. Ohno, T. Hasegawa, T. Tsuruoka, K. Terabe, J. K. Gimzewski, and M. Aono, "Short-term plasticity and long-term potentiation mimicked in single inorganic synapses," *Nature Materials*, vol. 10, no. 8, p. 591595, 2011.

[18] S. Song, K. Miller, and L. Abbott, "Competitive Hebbian learning through spike-timing-dependent synaptic plasticity," *Nature Neuroscience*, vol. 3, pp. 919–926, 2000.

[19] Y. Pershin and M. Di Ventra, "Practical approach to programmable analog circuits with memristors," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 57, no. 8, pp. 1857–1864, 2010.

[20] B. Mouttet, "Proposal for memristors in signal processing," *Nano-Net*, pp. 11–13, 2009.

[21] E. Linn, R. Rosezin, C. Kügeler, and R. Waser, "Complementary resistive switches for passive nanocrossbar memories," *Nature Materials*, vol. 9, no. 5, pp. 403–406, 2010.

[22] B. Widrow, "An adaptive 'ADALINE' neuron using chemical "memistors"," Stanford Electronics Laboratories Technical Report, Tech. Rep. TR-1553-2 23, Oct. 1960.

[23] S. Thakoor, A. Moopenn, T. Daud, and A. Thakoor, "Solid-state thin-film memistor for electronic neural networks," *Journal of Applied Physics*, vol. 67, no. 6, pp. 3132–3135, 1990.

[24] Q. Xia, M. D. Pickett, J. J. Yang, X. Li, W. Wu, G. Medeiros-Ribeiro, and R. S. Williams, "Two-and three-terminal resistive switches: Nanometer-scale memristors and memistors," *Advanced Functional Materials*, 2011 (in press).

[25] R. Rosezin, E. Linn, C. Kügeler, R. Bruchhaus, and R. Waser, "Crossbar logic using bipolar and complementary resistive switches," *IEEE Electron Device Letters*, vol. 32, no. 6, pp. 710–712, 2011.

[26] A. Whitehead and B. Russell, *Principia mathematica*, 1912, vol. 2.

[27] M. S. Qureshi, M. Pickett, F. Miao, and J. P. Strachan, "CMOS interface circuits for reading and writing memristor crossbar array," in *IEEE International Symposium on Circuits and Systems (ISCAS)*, 2011, pp. 2954–2957.

[28] O. Kavehei, S. Al-Sarawi, S. Sriram, M. Bhaskaran, and D. Abbott, "High density and non-volatile CRS-based CAM," *Arxiv preprint arXiv:1108.3716v1*, 2011.