Memcomputing with membrane memcapacitive systems

Y V Pershin\textsuperscript{1,3}, F L Traversa\textsuperscript{2} and M Di Ventra\textsuperscript{2}

\textsuperscript{1} Department of Physics and Astronomy, University of South Carolina, Columbia, SC 29208, USA
\textsuperscript{2} Department of Physics, University of California, San Diego, La Jolla, CA 92093-0319, USA

E-mail: pershin@physics.sc.edu, ftraversa@physics.ucsd.edu and diventra@physics.ucsd.edu

Received 9 January 2015, revised 3 April 2015
Accepted for publication 9 April 2015
Published 12 May 2015

Abstract

We show theoretically that networks of membrane memcapacitive systems—capacitors with memory made out of membrane materials—can be used to perform a complete set of logic gates in a massively parallel way by simply changing the external input amplitudes, but not the topology of the network. This polymorphism is an important characteristic of memcomputing (computing with memories) that closely reproduces one of the main features of the brain. A practical realization of these membrane memcapacitive systems, using, e.g., graphene or other 2D materials, would be a step forward towards a solid-state realization of memcomputing with passive devices.

Keywords: memcomputing, memory elements, memcapacitors, MEMS, polymorphism

(Some figures may appear in colour only in the online journal)

1. Introduction

Memcomputing, namely computing \textit{with and in} memory, is a novel non-Turing paradigm of computation that employs memory elements to process and store information at the same physical location [1, 2]. Even though this paradigm could be realized with standard complementary metal-oxide-semiconductor (CMOS) technology [3], its main premises rest on the use of passive circuit elements with memory (memelements), namely, memristive [4], memcapacitive and meminductive systems [5]. These memelements can indeed find numerous applications in electronics, including bio-inspired circuits [6, 7], neuromorphic circuits [8–10] and various unconventional computing architectures [1, 3, 11, 12]—just to name a few.

Memcomputing can be employed in both analog and digital mode [1, 2] also combined with standard CMOS technology [9, 13]. The first mode of operation is ideal for the solution of optimization problems, otherwise difficult to solve using standard digital machines [2, 12, 14], as well as for analog computing [15]. Moreover, memristive [3, 16] and memcapacitive [17] neural networks can also be considered as an analog realization of memcomputing. The second—digital—mode combines the strengths of memcomputing (most notably its intrinsic massive parallelism) with standard digital logic functionality.

The possibility of performing logic operations directly in memory with memelements [1, 2, 11, 18, 19] could also solve the long-standing von Neumann bottleneck problem [20] of modern computer architectures. Since usual capacitors have practically very low dissipations, memcapacitive systems are ideal components to perform computation with little energy [21], thus offering a solution to another pressing problem in modern computers: the ever-increasing energy consumption of our digital machines [22].

In previous work we have suggested the use of solid-state memcapacitive systems with diverging and negative capacitance [23] to perform logic operations within an architecture inspired by the dynamic random access memory one. We called this architecture a dynamic computing random access memory (DCRAM) [21]. It is worth noting, however, that although previously used solid-state memcapacitive systems can be used in digital mode, they are intrinsically analog elements. Memcapacitive systems that are fundamentally digital would thus be a better fit for this type of application although it is difficult to achieve 3D integration with membrane memcapacitors.

In this paper we employ the class of membrane memcapacitive systems [24] in the area of binary computing.
Membrane memcapacitive systems (see a schematic in figure 1) fit ideally in this context since stressed membranes have only two stable states unlike analog realizations of memcapacitive systems [23, 25–27]. The energy barrier between these two states plays the role of an intrinsic threshold that automatically assigns a binary value to any intermediate final state of the system. The combination of many of these membrane memcapacitive elements in an architecture like the DCRAM we have previously analyzed [21] would then represent an alternative way of implementing memcomputing in the solid state with passive devices. It is important to notice that such systems can be realized using graphene membranes [28–31] or any other molecular system as the flexible plate, and therefore our predictions are within reach of experimental verification.

There are several potential advantages of logic circuits based on membrane memcapacitive systems compared to traditional logic architectures. As mentioned above, the information processing and storage occur on the same physical platform. This feature reduces the amount of information transfer inside the computing system, bypassing altogether the data transfer between the memory and the central processing unit where information is traditionally processed. In addition, it allows performing logic operations in memory in a massively parallel and polyomorphic way [21]. This latter feature means that the implementation of different logic functions is not based on any specific prewired structure. Rather, the type of logic operation is selected only by the control signal amplitudes. As such, memcapacitive logic circuits require much smaller number of individual components as well as offer a versatility inexistent in traditional logic circuits, even those employing memristive components [11].

2. Membrane memcapacitive system

By definition [5], a voltage-controlled memcapacitive system is given by the equations

\[ q(t) = C(x, V_C) V_C(t), \]
\[ \dot{x} = f(x, V_C, t), \]

where \( q(t) \) is the charge on the capacitor at time \( t \), \( V_C(t) \) is the applied voltage, \( C \) is the memcapacitance, \( x \) is a set of \( n \) state variables describing the internal state of the system, and \( f \) is a continuous \( n \)-dimensional vector function. It is important that the memcapacitance \( C \) depends on the state of the system and can vary in time. Some theoretical and experimental studies of memcapacitive effects can be found in the literature [23–27, 32, 33] (for a recent review, see [34]).

2.1. Membrane memcapacitor model

In the membrane memcapacitive system [24], the mechanism of memory capacitance is geometrical [34]. The capacitor is formed by a strained membrane (upper plate) and a flat fixed lower plate as shown in figure 1. Two equilibrium states of flexible membrane (up-bent and down-bent) are suitable for non-volatile storage of bits of information. When the membrane is in a position closer to the bottom plate, the capacitance of the device is higher—we call this configuration ‘1’. When the membrane is bent up, the system has lower capacitance denoted by ‘0’.

A mathematical model of the membrane memcapacitive system formulated in [24] is based on a double-well potential (figure 1). This model describes the bistable membrane device as a second-order voltage-controlled memcapacitive system [5] in terms of the following equations:

\[ q(t) = C(y) V_C(t), \]
\[ \frac{dy}{d\tau} = y', \]
\[ \frac{dy'}{d\tau} = -4\pi^2 y \left( \frac{y}{y_0} \right)^2 - 1 - G'y - \beta(\tau) \left( \frac{y}{1 + y} \right)^2, \]

where

\[ C(y) = \frac{C_0}{1 + y}, \]

\( y = z/d, z \) is the position of the top membrane with respect to its middle position, \( d \) is the separation between the bottom plate and middle position of the flexible membrane, \( y_0 = z_0/d, G = 2\pi \gamma/\omega_0, \beta(t) = 2\pi/(\omega_0 d), \sqrt{C y/(2 m)} V_C(t) \) and time derivatives are taken with respect to the dimensionless time \( \tau = t \omega_0/(2\pi) \). Here, \( \pm \omega_0 \) are the equilibrium positions of the membrane, \( \gamma \) is the damping constant, \( \omega_0 \) is the natural angular frequency of the system, \( m \) is the mass of the membrane and \( C_0 = \varepsilon_0 S/d \). The membrane displacement \( yd \) and membrane’s velocity \( y' d \) play the role of the internal state variables \( x_1 \) and \( x_2 \) (in equations (1) and (2) \( x = [x_1, x_2] \)).

2.2. Membrane memcapacitor model including membrane resistance

An experimental realization of the bistable membrane memcapacitive system based on graphene was reported in [28]. It has been shown that the membrane dynamics can be described with a high precision by equations similar to
Single device operations were considered in previous work \[ \propto \text{proportional to} \] capacitor. Equation (3) must then be replaced by well as computing should support single device READ/ WRITE operations as they are very small, so we can safely neglect the terms in \[ \text{memcapacitive systems} \]. Charged to the same voltage, the smaller capacitance will result in a weaker current (below the threshold of the differential voltage amplifier) that could be distinguished from the stronger one (above the threshold).

Here, we just briefly review the more involved WRITE process. We note that in the presence of an applied voltage, the capacitive plates experience an attractive force toward each other. Correspondingly, the double well potential from figure 1 becomes asymmetric—its top minimum moves to the right and the bottom one moves to the left. At a certain voltage magnitude, the top minimum disappears and the system, regardless of its initial state, ends up in the bottom minimum. This is the basis to set the system to ‘1’. In order to set the system to ‘0’, a higher voltage is needed. When such voltage is applied and then removed, an accumulated elastic energy becomes sufficiently strong to overcome the potential barrier and set the system to ‘0’ (see [24] for more details).

### 3. Logic gate

We consider first logic operations with the circuit shown in figure 2. This circuit involves three memcapacitive systems and two voltage sources and can be considered as a sub-part of a larger circuit, effectively decoupled from this larger circuit with appropriate switches. Such a larger circuit could be, for example, similar to the DCRAM architecture we have previously introduced [21]. In fact, like in our previous work [21] with solid-state memcapacitive systems [23], we expect more cells will provide a larger set of logic gates at different voltages.

For the sake of simplicity, we do not show any initialization and measurement setup in figure 2 since only the computing stage is of interest. All results reported in this paper were obtained utilizing completely overlapping single square pulses \( V_1(t) \) and \( V_2(t) \), as shown in the top panel of figure 3.
The circuit dynamics is found using Kirchhoff’s circuit laws together with equations (3)–(5) defining the response and dynamics of memcapacitive devices. In particular, one can find that at each moment of time the voltages across the three memcapacitive systems are given by

\[ V_{C_1} = \frac{C_2 V(t) - (C_2 + C_3) V_{t}(t)}{C_1 + C_2 + C_3}, \]  
\[ V_{C_2} = \frac{C_1 V(t) - (C_1 + C_3) V_{t}(t)}{C_1 + C_2 + C_3}, \]  
\[ V_{C_3} = \frac{C_1 V_{t}(t) + C_2 V_{t}(t)}{C_1 + C_2 + C_3}, \]

where the voltages are defined with respect to the terminal denoted by the thick line in the memcapacitive system symbol in figure 2. These instantaneous values of voltages influence the dynamics of the internal state variables through equation (5). In what follows, it is assumed that the input values are stored in \( C_1 \) and \( C_2 \), while \( C_3 \) is reserved for the output value. However, for certain regions of pulse parameters, the final states of \( C_1 \) and \( C_2 \) are different from the input ones. Therefore, these two memcapacitive systems could also be used to store the computing result in some cases and thus few different logic operations could be realized in a single shot (see also section ‘reduced circuit’).

### Table 1. Codes of certain logic operations calculated according to equation (10). These codes are defined with respect to different pairs of input values (\( C_1, C_2 \)). For example, NOT \( C_1 \) is the logical negation on \( C_1 \), copy \( C_2 \) is the copy of the input state of \( C_2 \) into the final state of a given system, IMP1 is the material implication \( C_1 \rightarrow C_2 \), etc. More details are given in the text.

| Operation      | \( C_1 \) | \( C_2 \) |
|----------------|----------|---------|
| Set to 0       | 0        | AND 8   |
| NOR            | 1        | NOT(XOR) 9 |
| NOT(IMP2)      | 2        | Copy \( C_2 \) 10 |
| NOT(C1)        | 3        | IMP1 11 |
| NOT(IMP1)      | 4        | Copy \( C_1 \) 12 |
| NOT(C2)        | 5        | IMP2 13 |
| XOR            | 6        | OR 14   |
| NAND           | 7        | Set to 1 15 |

### 3.1. Material implication

We are now ready to show that these membrane memcapacitive systems are able to perform logic operations. To do this we focus on the logic material implication previously demonstrated with memory resistive devices [11]. Material implication is a very important logic function because it can be used to synthesize the negation (with the help of a false operation), which, together with implication, allows for a functionally complete set of logic gates.
In order to demonstrate the material implication, let us consider the circuit dynamics at specific amplitudes of voltage pulses $\beta_1(\tau)$ and $\beta_2(\tau)$, namely, $\beta_1 = 11$ and $\beta_2 = 42$. Figure 3 shows the dynamics of the internal states of memcapacitive systems (the position of the flexible plate) for four possible initial states of $C_1$ and $C_2$. It is assumed that $C_3$ is in '0' state at $\tau = 0$. Clearly, $C_3$ remains in '0' only at the (1, 0) input combination and its final state is '1' for all other input combinations. This is the material implication. We also note that the final state of $C_2$ is always '1'. The state of $C_1$ remains unchanged during the circuit dynamics operation.

3.2. Maps of logic operations

In order to better understand which logic operations can be implemented with the memcapacitive logic circuit from figure 2, we prescribe a numerical value to operation results as follows. Taking $w_i = 1, 2, 4, 8$ as weights for the input combinations (0, 0), (0, 1), (1, 0) and (1, 1), a numerical code is calculated as a weighted sum of the final state of a particular memcapacitive system, namely

$$
\text{code} = \sum_{i=1}^{4} w_i b_{ij},
$$

where $b_{ij}$ is the final state (0 or 1) of the device of interest $j$ ($C_1$, $C_2$ or $C_3$) for $i$th input combination (0, 0), (0, 1), (1, 0) or (1, 1) that correspond to $i = 1, 2, 3, 4$. For example, for the material implication function shown in figure 3, the code for the final state of $C_3$ is $1 \cdot 1 + 2 \cdot 1 + 4 \cdot 0 + 8 \cdot 1 = 11$. Therefore, 11 is the code for material implication $C_1 \rightarrow C_2$. Similarly, one can find that the material implication $C_2 \rightarrow C_1$ corresponds to the code value 13. Table 1 summarizes the codes for all operations implemented with membrane memcapacitive logic.

Figure 4 is the main result of this work. It identifies the regions of voltage pulse amplitudes $\beta_1$ and $\beta_2$ realizing specific logic functions as the final state of $C_3$. Each point of this plot is calculated similarly to section 'material implication’ calculation assuming that $C_3$ is in 0 at $\tau = 0$. As expected, at smaller values of $\beta_1$ or $\beta_2$ and any input combination, the final state of $C_3$ is 0. Material implication, OR, NAND and some other functions are found at higher values of the applied pulses as shown in figure 4. This calculation demonstrates that the same memcapacitive circuit is capable of realizing different logic functions on demand without any changes in the circuit configuration. The circuit thus performs polymorphic computing in the sense discussed in [21].

The final states of $C_1$ and $C_2$ are shown in figure 5. Although there are large regions where $C_1$ and $C_2$ stay unchanged, one can identify regions of amplitudes implementing the material implication (codes 11 and 13), ‘set to 1’ (code 15) and some other functions. Therefore, a logic function and initialization or two different logic functions can be performed in a single step (intrinsic parallelism [1, 2]) thus further increasing the efficiency of calculations.

Figure 5. Logic operation type as a function of pulse amplitudes $\beta_1$ and $\beta_2$ for the input memcapacitive systems $C_1$, (a), and $C_2$, (b). The data for these plots and figure 4 were obtained within the same calculations.

Figure 6. Magnified region of figure 4 showing non-trivial fine structures.
We also note that there are regions in the input-voltage parameter phase space that show non-trivial features such as those two irregularities observed at approximately \((\beta_1 = 1.5, \beta_2 = 3.5)\) and \((\beta_1 = 3.5, \beta_2 = 1.5)\) in figure 4. Figure 6 shows the feature magnification revealing additional fine structures (straight lines) in those regions. These features originate from the complex dynamics of memcapacitive systems already spotted in a previous work of one of us [24]. We emphasize that such irregularities are observed only in limited intervals of voltage amplitudes, and therefore can be easily avoided in practical realizations of membrane memcapacitive systems. Moreover, by varying geometrical \(y_0\) and physical \(T\) parameters as shown in the Supporting Information, the chaotic behavior of the circuit can be minimized.

3.3. Reduced circuit

The results presented in figure 5 demonstrate that the same memcapacitive device could store both input and output logic values. In order to better understand this capability we consider a reduced circuit consisting of two memcapacitive systems as sketched in figure 7(a). We have performed simulations of the circuit dynamics subjected to the same couple of pulses and simulation parameters as we have discussed above.

The results of these simulations presented in figure 7(b) demonstrate that even such a simple circuit is capable of implementing the OR gate in a significant interval of parameters. However, a single OR is not enough for universal computing. One possibility to attain this goal would be a combination of the OR gate and NOT gates. Considering dynamics of a single membrane memcapacitive system subjected to a voltage pulse, we have indeed found pulse parameters realizing the NOT. Figure 7(c) shows an example of such realization.

In order to better understand the NOT implementation, we plot the difference of final positions of plates for different initial conditions \(y_1(0) = 0.2\) and \(y_2(0) = -0.2\), namely, \(y_1 - y_2\) at \(\tau = 40\). The NOT is realized when \(y_1(40) = -0.2\) and
y_2(40) = 0.2. In other words, when the final y_1 − y_2 = -0.4. Figure 7(d) shows multiple regions of the NOT gate, which could be achieved, for the set of system parameters selected, using a fine pulse width tuning. We emphasize that a further improvement of membrane memcapacitive logic is possible. For example, a larger set of logic operations with two memcapacitive devices could possibly be obtained adding a capacitor to the circuit in figure 7(a). A pulse engineering is an additional opportunity that could lead to improved functionality.

4. Impact of device parameters

In order to better understand the implementation of logic operations with membrane memcapacitive systems, we have performed several additional calculations varying parameters y_0 and \( \Gamma \) of the model. Figure 8 shows results of these calculations for a circuit of three memcapacitive systems (depicted in figure 2) at a fixed value of y_0 = 0.2 and several representative values of \( \Gamma \). This figure demonstrates that at smaller values of \( \Gamma \) (figures 8(a) and (b)) there is a significant region of chaotic-like behavior at larger values of \( \beta_1 \) and \( \beta_2 \) (see the top right parts of these plots). Increasing \( \Gamma \) stabilizes this region (figures 8(c) and (d)). Clearly, already at \( \Gamma = 0.7 \) there are no hints of uncertainty in that large \( \beta_1 \) and \( \beta_2 \) region. Therefore, while the chaotic-like behavior is already not possible at \( \Gamma = 0.7 \), a larger value of \( \Gamma \) could be used in experimental realizations of the circuit to guarantee its operation stability.

Figure 9 shows the results of simulations for the same three-device circuit (figure 2) at a fixed value of \( \Gamma = 0.7 \) and several values of y_0. It follows from figure 9 that the regions of useful logic functions are significantly increased with increasing y_0. There are three potentially interesting regions in figure 9(d) (codes 7, 11, 13) corresponding to NAND and material implications. Each of these regions provides a universal logic capability.

5. Conclusions

We have shown that memcomputing—computing with and in memory—[1] can be implemented with membrane...
memcapacitive systems. This demonstrates that this quite different type of memcapacitive system (compared to the solid-state memcapacitive systems previously studied [21]) is also suitable for massively parallel and polymorphic computing operations directly in memory, thus offering a different realization of the memcomputing concept. Experimentally, our predictions could be verified with membrane memcapacitive systems employing, e.g., a stressed graphene membrane\(^5\), a synthetic (artificial) membrane, or a molecular system as the flexible plate. Graphene membranes, for instance, have been recently demonstrated experimentally as systems for quantum information [31]. Here, instead we suggest their use as semi-classical two-level systems. We thus hope our predictions will motivate further experimental and theoretical work in this direction.

From the fabrication point of view, the membrane memcapacitive memory could be realized in a CMOL-like architecture [39], which is a hybrid architecture combining a semiconductor-transistor (CMOS) layer with a layer of molecular-scale nanodevices formed between two levels of parallel nanowires [39]. In our case, the top layer will be a layer of membrane memcapacitive systems coupled with the bottom CMOS layer using a set of vertical connections. It is anticipated that a single graphene-based memcapacitive system could be scaled down to few 100 s nm\(^2\) area. The amount of memcapacitive systems scales almost linearly with the chip area and the most natural architecture to include circuitry to control computing, reading and writing processes is the well scalable DRAM-like architecture [21]. Furthermore, membrane memcapacitive systems are passive systems (excluding control circuitry) thus the energy per operation can be as small as the energy required for the voltage pulses used to read/write and compute, that is typically of the order of few fJ for standard technologies [21]. Finally, the maximum operation frequency (being passive) is directly related to the membrane damping constant \(\gamma\), thus strongly depending on the materials and technologies used to build the memcapacitors, for example, how the membrane is attached to the substrate strongly affects \(\gamma\). In particular, from [28] and from

---

\(^5\) Graphene is currently used in experimental capacitors [28–30] albeit in a different role.
some rough estimation, a working frequency could be of the order of a 100 MHz. However, in this case very large sheets of graphene have been used and the temperature is very low. We think that at higher temperatures (where the sheet resistance is higher, see section 2.2 for the impact of the resistance on the effective damping parameter) and much smaller membrane sizes, a working frequency of at least few GHz can be reached.

Acknowledgments

This work has been partially supported by the NSF grant ECCS-1202383, and the Center for Magnetic Recording Research at UCSD.

References

[1] di Ventra M and Pershin Y V 2013 The parallel approach Nat. Phys. 9 200
[2] Traversa F L and Di Ventra M 2015 Universal memcomputing machines IEEE Trans. Neural Netw. Learn. Syst. (arXiv: 1405.0931)
[3] Pershin Y V and Di Ventra M 2010 Experimental demonstration of associative memory with memristive neural networks Nat. Nanotech. 5 80
[4] Chu L O and Kang S M 1976 Memristive devices and systems Proc. IEEE 64 209–23
[5] Di Ventra M, Pershin Y V and Chua L O 2009 Circuit models of amoeba learning Phys. Rev. E 80 021926
[6] Traversa F L, Pershin Y V and Di Ventra M 2013 Memory models of adaptive behaviour IEEE Trans. Neural Netw. Learn. Syst. 24 1437–48
[7] Jo S H, Chang T, Ebong I, Bhadviya B B, Mazumder P and Lu W 2010 Nanoscale memristor device as synapse in neuromorphic systems Nano Lett. 10 1297–301
[8] Kim K H, Gaba S, Wheeler D, Cruz-Albrecht J M, Hussain T, Srivinasa N and Lu W 2012 A functional hybrid memristor crossbar-array/cms system for data storage and neuromorphic applications Nano Lett. 12 389–95
[9] Pershin Y V and Di Ventra M 2012 Neuroorphic, digital and quantum computation with memory circuit elements Proc. IEEE 100 2071
[10] Borgiotti J, Snider G S, Kuekes P J, Yang J J, Stewart D R and Williams R S 2010 ‘Memristive’ switches enable ‘stateful’ logic operations via material implication Nature 464 873–8
[11] Pershin Y V and Di Ventra M 2011 Solving mazes with memristors: a massively parallel approach Phys. Rev. E 84 046703
[12] Xia Q et al 2009 Memristor-CMOS hybrid integrated circuits for reconfigurable logic Nano. Lett. 9 3640–5
[13] Pershin Y V and Di Ventra M 2013 Self-organization and solution of shortest-path optimization problems with memristive networks Phys. Rev. E 88 013305
[14] Wright C D, Liu Y, Kohary K I, Aziz M M and Hicken R J 2011 Arithmetic and biologically inspired computing using phase-change materials Adv. Mater. 23 3408–13
[15] Thomas A 2013 Memristor-based neural networks J. Phys. D: Appl. Phys. 46 093001
[16] Pershin Y V and Di Ventra M 2014 Memcapacitive neural networks Electron. Lett. 50 141
[17] Linn E, Rosezin R, Tappertzhofen S, Böttger U and Waser R 2012 Beyond von Neumann-logic operations in passive crossbar arrays alongside memory operations Nanotechnology 23 305205
[18] Ijelvær E V, Jesse S, Morozovska A N, Strelcov E, Eliseev E A, Pershin Y V, Kumar A, Shur V Ya and Kalinin S V 2014 Intermittency, quasiperiodicity and chaos in probe-induced ferroelectric domain switching Nat. Phys. 10 59–66
[19] Backus J 1978 Can programming be liberated from the von Neumann style? A functional style and its algebra of programs. Commun. ACM 21 613–41
[20] Traversa F L, Bonani F, Pershin Y V and Di Ventra M 2014 Dynamic computing random access memory Nanotechnology 25 285201
[21] Kogge P 2011 The tops in flops IEEE Spectr. 48 48–54
[22] M-Rincon J, Di Ventra M and Pershin Y V 2010 Solid-state memcapacitive system with negative and diverging capacitance Phys. Rev. B 81 195430
[23] M-Rincon J and Pershin Y V 2011 Bistable non-volatile elastic membrane memcapacitor exhibiting chaotic behavior IEEE Trans. Electron Devices 58 1809
[24] Liu S, Wu N, Ignatiev A and Li J 2006 Electric-pulse-induced capacitance change effect in perovskite oxide thin films J. Appl. Phys. 100 056101
[25] Lai Q, Zhang L, Li Z, Stickle W F, Williams R S and Chen Y 2009 Analog memory capacitor based on field-configurable ion-doped polymers Appl. Phys. Lett. 95 213503
[26] Flak J, Lehtonen E, Laiho M, Rantala A, Prunnila M and Haatainen T 2014 Solid-state memcapacitive device based on memristive switch Solid. State. Technol. 29 104012
[27] Eichler A, Moser J, Chaste J, Zdrojek M, W-Rae I and Bachtold A 2011 Nonlinear damping in mechanical resonators made from carbon nanotubes and graphene Nanotechnology 6 339–42
[28] Stoller M D, Park S, Zhu Y, An J and Ruoff R S 2008 Graphene-based ultracapacitors Nano Lett. 8 4398–502
[29] El-Kady M F and Kaner R B 2013 Scalable fabrication of high-power graphene micro-supercapacitors for flexible and on-chip energy storage Nat. Commun. 4 1475
[30] Singh V, Bosman S J, Schneider B H, Blanter Y M, Castellanos-Gomez A and Steele G A 2015 Optomechanical coupling between a multilayer graphene mechanical resonator and a superconducting microwave cavity Nat. Nanotechnology 9 820–4
[31] Biolek D, Biolek Z and Biolkova V 2010 SPICE modelling of memcapacitor Electron. Lett. 46 520
[32] Krens M, Pershin Y V and Di Ventra M 2010 Ionic memcapacitive effects in nanotubes Nano Lett. 10 2674
[33] Pershin Y V and Di Ventra M 2011 Memory effects in complex materials and nanoscale systems Adv. Phys. 60 145–227
[34] De S and Coleman J N 2010 Are there fundamental limitations on the sheet resistance and transmittance of thin graphene films? ACS Nano 4 2713–20
[35] Kim K K, Reina A, Shi Y, Park H, Li L-J, Lee Y H and Kong J 2010 Enhancing the conductivity of transparent graphene films via doping Nanotechnology 21 285205
[36] Liu H, Liu Y and Zhu D 2011 Chemical doping of graphene J. Mater. Chem. 21 3335–45
[37] Petkov V P and Boser B E 2004 Capacitive interfaces for mems Advanced Micro and Nanosystems ed H Baltes, O Brand, G K Fedder, C Hierold, J G Korvink and O Tabata (Weinheim: Wiley) pp 49–92
[38] Strukov D B and Likharev K K 2005 CMOL FPGA: a reconfigurable architecture for hybrid digital circuits with two-terminal nanodevices Nanotechnology 16 888–900