

# Architecting Energy Efficient Crossbar-Based Memristive Random-Access Memories

Miguel Angel Lastras-Montañó  
Department of Electrical and  
Computer Engineering  
University of California, Santa Barbara  
Santa Barbara, California  
Email: mlastras@ece.ucsb.edu

Amirali Ghofrani  
Department of Electrical and  
Computer Engineering  
University of California, Santa Barbara  
Santa Barbara, California  
Email: ghofrani@ece.ucsb.edu

Kwang-Ting Cheng  
Department of Electrical and  
Computer Engineering  
University of California, Santa Barbara  
Santa Barbara, California  
Email: timcheng@ece.ucsb.edu

**Abstract**—Memristive devices are promising candidates for future high-density, power-efficient memories. The sneak path problem of purely-resistive crossbars and the inherent nanowire voltage drop, however, prevent the use of memristors in large-scale memory systems. In this paper we provide a simple yet flexible 3D memory organization and decoding scheme for memristive crossbars that exploits the benefits of the CMOL interface and avoid the limitations of conventional resistive crossbars. We propose an electrical model of the system to simulate and estimate its delay and energy consumption and show that such memories provide high read/write concurrency with power consumption per read/write operation that is significantly lower than that of DRAM.

## I. INTRODUCTION

The ever-increasing need for larger, faster, and lower power memories has been addressed so far by aggressive technology scaling and operating voltage reduction. DRAM, as the current prominent technology for main memory, has been following this trend but the increasing power consumption due to high leakage currents in their access transistors and the necessity of refreshing the memory has rendered DRAM inadequate in the long term [1].

Among the alternatives to DRAM, several non-volatile random access memories (NVRAM) have been identified as candidates. Non-volatile memories are specially useful since they retain their data even when the power is interrupted. Flash memory is the most common form of NVRAM but its slow writing speed and low endurance hinder its use as main memory. Emerging technologies such as ferroelectric RAM, magnetoresistive RAM and phase-change RAM are also alternatives that address some of these issues but due to their need for an access element (e.g. a transistor) per memory cell, they suffer from the same scaling limitations as DRAM.

Resistive RAM (RRAM or ReRAM) is another emerging non-volatile technology that offers the possibility of eliminating the memory access elements [2] due to the unique I-V and dynamic characteristics of their memory elements [3], while offering fast read/write operations [4] and high endurance [5]. Such memories have two-terminal *memristive devices* (memristors) as memory elements in which information is stored as a resistance across the two terminals [6]. There are several realizations for memristive devices, each described by a different switching phenomena [7], but the general structure of a memristor consists of a thin film of a switching material that is “sandwiched” between two metallic electrodes. The relative simplicity of these devices and the option of not requiring an access element offer better scaling properties compared to other technologies.

To form a high-density memory using memristive devices, the crossbar structure is a natural option due to its simplicity and regularity [8], [9]. A memristive crossbar array is formed by two perpendicular layers of parallel nanowires, at the crosspoints of which a memristor is formed. At the maximum density, the footprint of a memory cell is only  $4F_{\text{NANO}}^2$ , where  $F_{\text{NANO}}$  is the nanowire’s width,

and it can be further reduced to  $4F_{\text{NANO}}^2/L$  by stacking  $L$  crossbar layers.

While the current memory density of DRAM is  $\sim 10^{10}$  bits/cm<sup>2</sup> with no significant projected improvement [1], several recent studies have demonstrated functional memristive crossbars with much higher densities [9]–[12]. In the most promising results [10], one layer of memristive crossbar with  $F_{\text{NANO}} = 9$  nm was fabricated providing memory densities of  $10^{11}$  bits/cm<sup>2</sup>. In [13] the authors found that the minimum  $F_{\text{NANO}}$  for a memristor is 4 nm, which would further improve the density to  $10^{12}$  bits/cm<sup>2</sup> per crossbar layer. It is shown in [14] that current state of the art nanoimprint lithography techniques can already fabricate structures of this minimum size.

The crossbar architecture, however, has scaling limitations that prevents its practical implementation in larger arrays: With no access element per memory cell, selecting a particular crosspoint of two nanowires partially selects other crosspoints on the same nanowires. Leakage currents at these partially selected crosspoints both increases the power consumption and reduces the noise margin of the read circuitry, thus limiting the maximum number of crosspoints a nanowire can have [15]. Moreover, the high resistivity of nano-scale wires results in significant voltage drop along the line which limits the length of the nanowires [4]. While several methods are proposed to improve such limits [16]–[18], none can support arbitrarily large arrays without per cell access elements. Another downside of the crossbar architecture is that the feature size mismatch between the memristive layer and the CMOS layer (used to control and decode), makes the CMOS/crossbar interface more challenging and area-consuming. This limitation is schematically shown in Fig. 1(a).

The CMOL architecture [19] is a suitable solution that overcomes these limitations. In CMOL, instead of having a lateral CMOS/crossbar interface as depicted in Fig. 1(a), an area-distributed interface below a rotated crossbar array is used (Fig. 1(b)) and each nanowire is divided into segments of predefined length. This architecture has several advantages: it eliminates the area for pitch reduction, provides very high memory densities, offers excellent scalability by limiting the size of the nanowires, can be monolithically integrated with a CMOS subsystem [20]–[22] and allows stacking of multiple crossbar layers to form a 3D memory [23].

While CMOL offers all these advantages, there are several issues that need to be worked out before the actual implementation of a CMOL-based memory system: (1) In order to demonstrate the advantages of a memory system as a whole, an electrical model of the system is necessary to estimate its total power consumption and access speed. To the best of our knowledge, there is no work addressing that. (2) Without the tangible regularity of crossbars, the CMOL architecture is structurally more complex than traditional memory arrays, which complicates the memory organization and address

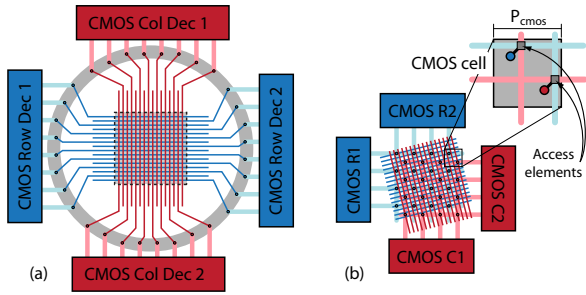


Fig. 1. Interfacing options between CMOS and a nanowire crossbar. (a) The typical solution; (b) the CMOL approach with lateral decoders and distributed interface. The nanowire segmentation in (b) is not shown.

decoding. Without a proper organization, the full potential of CMOL cannot be exploited. In [24], the authors present a high-level CMOL memory organization in which a matrix of CMOL-based crossbar blocks are connected together along with their decoders. Details of the actual implementation and various aspects for designing such blocks, however, have not been worked out. It is worth mentioning that without the details of the memory organization, the accurate electrical model of the system cannot be attained.

In this paper, we propose a memory organization and decoding scheme for CMOL-based crossbar memories that facilitates the implementation of scalable 3D memory systems. The proposed organization unveils the regularity of CMOL by introducing the division of the crossbar and underlying CMOS circuitry into *multicells*. Our organization allows the usage of such crossbars as standalone memories or as memory banks in a multi-bank memory. An electrical model is developed based on the physical properties of the nanowires, CMOS/nanowire interface, dynamic behavior of the memristive devices, and transistor-level implementation of the CMOS circuitry. This model is then used to validate the memory organization and evaluate its competitiveness in terms of delay and energy consumption.

The next section covers the necessary backgrounds on memristive devices and elaborates on CMOL. Sections III and IV describe the proposed organization and the read/write operations, respectively. The electrical model of the system is explained in Section V. Section VI shows the simulation results and a discussion of the effect of memory banking on energy consumption is presented. Concluding remarks are given in Section VII.

## II. BACKGROUND

A simplified I-V curve of a memristor is shown in Fig. 2(a). Such behavior can be obtained by highly nonlinear memristors [3], and can be also achieved using complementary resistive switches [25], [26]. Applying voltages below a threshold voltage  $V_{th}$  neither generates a significant current nor changes the device resistance.

To read the device, a read voltage  $V_r$  is applied across the memristor and the resulting current is measured to determine its resistance. To write a low resistance state (LRS), a write voltage

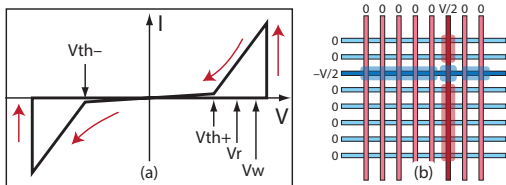


Fig. 2. (a) Assumed memristive I-V curve. (b) V/2 scheme.

$V_w > V_r$  is applied for a period of time  $t_{write}$ . To write a high resistance state (HRS),  $-V_w$  is used instead.

To access a memristor in a crossbar, the “V/2 scheme”, shown in Fig. 2(b), can be used. A voltage of  $\pm V/2$  and  $\mp V/2$  are applied to a horizontal and a vertical nanowire, respectively, while grounding the others. This results in a voltage of  $\pm V$  at the target memristor and a voltage across other memristors of either zero or  $\pm V/2$ . If  $V < 2V_{th}$ , no significant leakage currents result on these partially selected memristors, which allows us to eliminate the access elements.

A major problem of the simple nanowire crossbar architecture (Fig. 3(a)) is that larger memory capacities result in longer nanowires. The natural solution is to use shorter nanowires instead. This can be achieved by dividing the long nanowires into smaller segments, as shown in Fig. 3(b). A consequence of such division is that the nanowire segments that are located in the middle, shown in gray, are no longer accessible by the lateral decoders. To have unique access to the entire “sea” of nanowires, a *distributed interface* can be used instead to connect to the nanowires, as shown in Fig. 3(c).

The CMOL architecture [19] is an implementation of these ideas. In CMOL, the distributed interface consists of two rectangular arrays of *pins*, called *blue* and *red* pins hereafter, rotated by an angle  $\alpha$  with respect to the direction of the crossbar nanowires [19]. The vertical (red) nanowires interface with the array of red pins whereas the horizontal (blue) nanowires interface with the array of blue pins. Each pin is activated by means of an *access element*, e.g., a transistor or transmission gate, which in turn is enabled using two lateral CMOS decoders (two decoders for each array of pins). A pair of adjacent blue and red pins, together with their access elements, define a *CMOS cell*, as shown in Fig. 1(b). The pitch of a CMOS cell is  $P_{CMOS}$  and depends on the complexity of the access elements [27] as well as the CMOS technology pitch size. The lateral decoders together with the access elements constitutes the CMOS subsystem of the memory.

By rotating the crossbar by an angle  $\alpha$ , the pitch size of the nanowires and that of the CMOS subsystem can be decoupled. For any  $P_{CMOS}$  and  $F_{NANO}$ ,  $\alpha$  can be found so that each pin, and thus each nanowire, can be uniquely accessed by the lateral CMOS decoders [19]. The angle  $\alpha$  determines the length of the segments and therefore the number of crosspoints on each segment, and is defined by  $\alpha = \arctan(1/R)$  in which  $R$  is an integer greater than 1. An analysis of the rotated crossbar reveals that for an even  $R$ , the number of crosspoints per segment is  $R^2$  whereas for an odd  $R$ , the number is  $R^2 - 1$ . In both cases, for a given  $\alpha$ , the number of crosspoints per segment will be constant and independent of the overall size of the crossbar array.

To access a crosspoint in the crossbar we first need to access one segment, say a red segment using a red pin, and then one blue segment that falls within the *connectivity domain* of the accessed red segment. The connectivity domain of a red (blue) segment is formed by all the  $R^2$  or  $R^2 - 1$  blue (red) segments it is directly connected to. Since every segment is uniquely connected to a single pin, it is

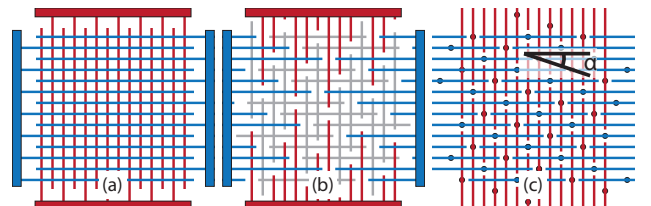


Fig. 3. (a) A simple crossbar; (b) wire segmentation; (c) distributed interface.

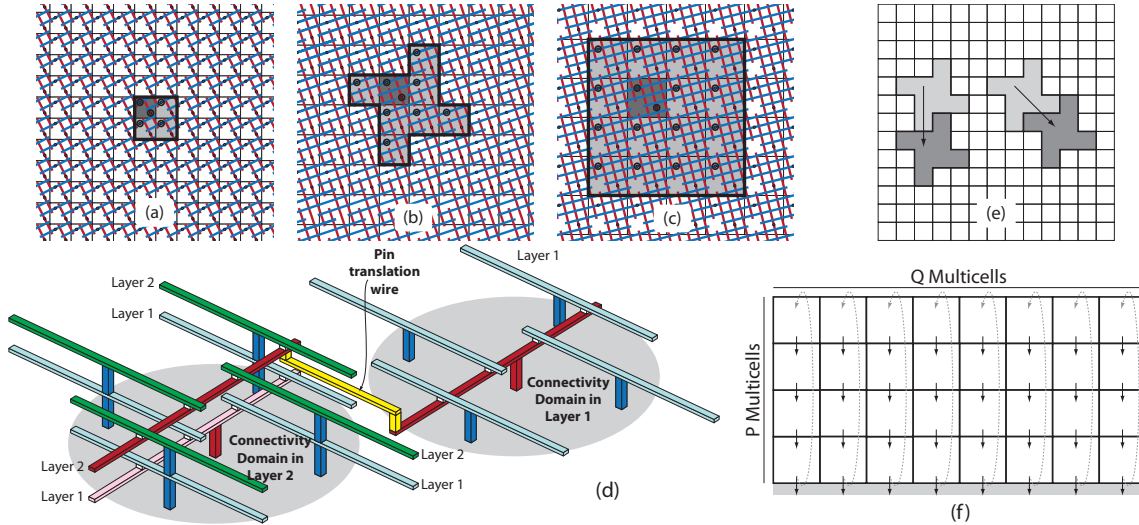


Fig. 4. (a) - (c): Crossbar structures and their connectivity domains for  $R$  equal to 2, 3, 4, respectively; (d) diagram showing a possible mapping between two crossbar layers; (e) two mapping directions; (f) crossbar and CMOS cell division into multicells, as well connectivity domain mappings.

easier to visualize the shape of a connectivity domain by highlighting the pins and the CMOS cells that drive them, instead of the actual segments. Figs. 4(a-c) show the shape of the connectivity domain of a red wire for  $R$  equals to 2, 3, and 4 respectively. A grid in these figures corresponds to a CMOS cell. Note that for an even  $R$  the connectivity domains are squares of size  $R^2$  (i.e. consisting of  $R \times R$  CMOS cells), whereas for an odd  $R$  they have a star-like shape. Also note that a larger  $R$  results in a larger CMOS cell for a given  $F_{\text{NANO}}$ . That explains why the grid size in Fig. 4(c) is larger than that in Fig. 4(b), which in turn is larger than that in Fig. 4(a).

A consequence of having limited connectivities per segment is that not every pair of red/blue pin will have a corresponding crosspoint, resulting in the underuse of the address space provided by the lateral CMOS decoders. To address this problem, multiple layers of crossbars can be stacked to form a 3D memory, where the connectivity domain of a nanowire segment in one crossbar layer can be extended to a different non-overlapping region in another crossbar layer [23]. This is implemented by means of an extra layer of pin translation wires between the crossbar layers, as shown in Fig. 4(d). This will enhance the address space and the effective memory density. The mapping used in [23] (Fig. 4(e) right), however, significantly increases the complexity of locating the additionally reachable crosspoints in other crossbar layers.

### III. MEMORY ORGANIZATION

The limited connectivity of the nanowire segments in a CMOL-based crossbar, i.e., not every pair of red/blue pins results in a direct connection, makes it difficult to design a modular organization that allows reading/writing multiple bits concurrently while maintaining the benefits of CMOL, such as having several crossbar layers.

Motivated by designing such an organization, we propose a simple yet powerful solution that divides the crossbar and array of CMOS cells into  $P \times Q$  equally-sized subarrays, called *multicells*, each of which consisting of an array of  $R \times R$  CMOS cells, regardless of  $R$  being odd or even. Partitioning the crossbar into multicells of width  $R$  CMOS cells is particularly useful, as the crossbar regions reachable by the CMOS cells with that minimum distance can be accessed concurrently without electrically interfering with each other.

With this division, the mapping of connectivity domains from one crossbar layer to other crossbar layers can now be done at the multicell level, i.e., each CMOS cell in a multicell in one crossbar layer will be mapped to its same position in the multicell that is one multicell-row below it (refer to Figs. 4(f) and (e) left). With this mapping, only the CMOS cells that are in the last multicell-row cannot be mapped, which is an improvement compared to the  $45^\circ$  mapping scheme used in [23] (Fig. 4(e) right) in which the loss occurs at both the bottom and left borders. One option to further improve the address space is to map the last multicell-row to the top multicell-row as indicated by the dashed lines in Fig. 4(f).

This division permits the stacking of up to  $P$  crossbar layers for implementing a multi-crossbar-layer 3D memory system. It also allows reading or writing  $Q$  bits (one bit per multicell-column) concurrently using simple hierarchical lateral decoders. Furthermore, this division also facilitates the addition of extra crossbar layers by structurally simplifying the pin translation wires between the crossbar layers by having straight translation wires, rather than zigzag structures used in [23].

Given a configuration of  $P$ ,  $Q$ ,  $R$  and  $L$  (where  $L$  denotes the number of crossbar layers), a memory built using this organization will consist of an array of  $(P \times R) \times (Q \times R)$  CMOS cells. Assuming an even  $R$ , each CMOS cell can access  $R^2$  elements yielding a maximum memory capacity of  $PQR^4$  crosspoints per crossbar layer. If the bottom multicell-row can be mapped to the top multicell-row (please refer to Fig. 4(f) where the dashed line illustrates this mapping), the total capacity and the maximum capacity (occurring when  $L = P$ ) of the 3D memory will be:

$$\uparrow C_{\text{tot}} = LPQR^4 \quad \uparrow C_{\text{max}} = P^2QR^4 \quad (1)$$

If such mapping is not possible, the capacities are:

$$\downarrow C_{\text{tot}} = (2P + 1 - L)LQR^4/2 \quad (2)$$

$$\downarrow C_{\text{max}} = P(P + 1)QR^4/2 \quad (3)$$

#### A. Double Decoding Scheme

To access a particular crosspoint in the crossbar we need to use two types of lateral decoders, one for each set of pins (red and blue). The “red” decoder selects one red pin/segment whereas the “blue”

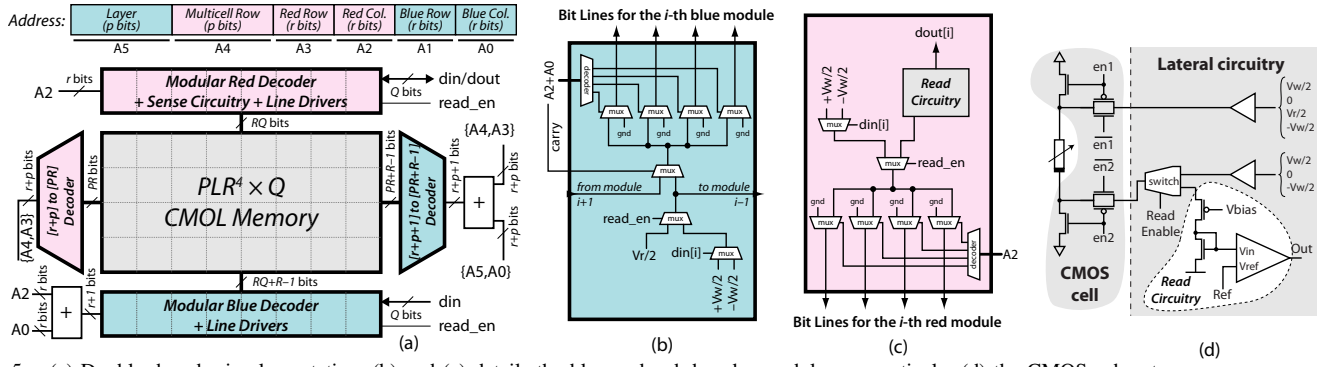


Fig. 5. (a) Double decoder implementation; (b) and (c) details the blue and red decoder modules, respectively; (d) the CMOS subsystem.

decoder is used to select one of the  $LR^2$  blue pins that fall within the extended connectivity domain of the red pin. To avoid the special cases that occurs at the periphery of the array (the connectivity domain of those pins is not complete), a constant number ( $\sim R$ ) of rows and columns of blue pins and blue segments are added to complete the missing crosspoints. In this way, all the red segments will have the same number of crosspoints.

Once a particular crosspoint is selected in one of the multicell-columns, which represents one bit, the corresponding crosspoints in the same row of all other  $Q - 1$  multicell-columns can be accessed concurrently, forming a  $Q$ -bit word. Assuming for simplicity that  $P = 2^p$  and  $R = 2^r$  where  $p$  and  $r$  are integers, Fig. 5(a) shows the memory implementation including modular double decoders and the number of address bits needed in each part. The red pin selection needs  $2r + p$  bits and it is independent of the blue decoder. The blue pin selection on the other hand needs to be aware of the red pin selection. In order to resolve this dependency, the red address is considered as the base address for the blue part, and the given blue address is offset to that base. In total,  $4r + 2p$  address bits are needed to locate a  $Q$ -bit word. Figs. 5(b) and (c) show our implementation of the decoder modules for blue and red parts respectively for the case of  $R = 4$ .

#### IV. MEMORY OPERATIONS

In Fig. 6 we show a simplified switch-level view of the circuit used to implement the read and write operations. For the sake of illustrating the concept, the diagram shows a crossbar constructed with  $R = 2$  ( $2^2$  crosspoints per nanowire segment), however, the same concept works for any value of  $R$ . The buffers, which form part of the lateral circuitry, can drive the red/blue bit lines to  $\pm V_w/2$ ,  $\pm V_r/2$  or Ground. The access elements, on the other hand, should

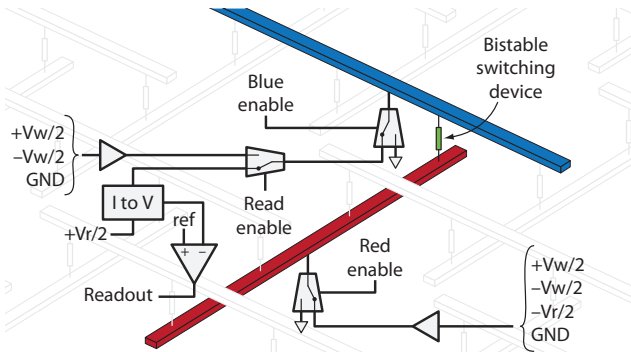


Fig. 6. Electrical diagram for  $R = 2$ .

select between grounding the pin or connecting it to the bit-lines based on the enable signals produced by the lateral decoders. An extra switch in the lateral circuitry is used to select between the read and write circuitries. The read circuitry consists of a current-to-voltage converter, a comparator and its voltage reference.

With this supporting circuitry, a crosspoint can be set into the LRS or HRS by applying  $\pm V_w/2$  to a blue segment and  $\mp V_w/2$  to the corresponding red segment to create a voltage of  $\pm V_w$  across the device. For the read operation, a voltage  $V_r/2$  ( $-V_r/2$ ) is applied to a red (blue) segment. When not reading or writing, the segments are grounded.

Applying symmetric voltages across the devices allows us to read and write multiple bits in different multicell-columns concurrently. Fig. 7 illustrates this during a write. Only the selected crosspoints have a voltage of  $\pm V_w$  across them which effectively writes a LRS or a HRS. The rest of the crosspoints have a voltage of either 0 or  $\pm V_w/2$  across them, which is too low to modify the device's content.

Fig. 5(d) shows a CMOS implementation of the circuit in Fig. 6. The access elements consist of a transmission gate and a pull-down transistor. The read circuitry uses a diode-connected transistor and a source follower to convert the current into a voltage which is used by the comparator to produce the read data.

#### V. ELECTRICAL MODELING

We modeled the crossbar layers as an RC network connected to the CMOS cells underneath them. The CMOS cells are in turn controlled by the word and bit lines from the lateral decoders. Fig. 8(a) shows the partial structure of the components involved. The nanowire separation is  $a \times F_{\text{NANO}}$  where  $a = 2$  gives the highest crosspoint density and  $t \times F_{\text{NANO}}$  is the thickness of the nanowires. Pin translation wires and the pins (the CMOS/crossbar interface) are modeled as a cylindrical structure with diameter  $F_{\text{NANO}}$  and height  $h$ .

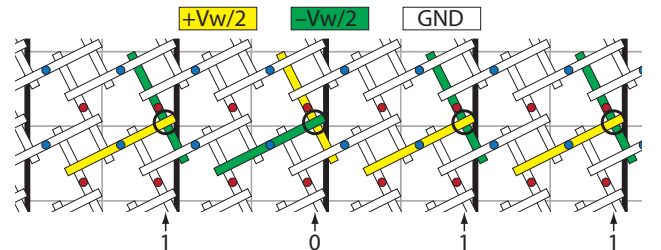


Fig. 7. Arbitrary patterns can be written on different multicells by applying  $\pm V_w/2$  on one of the top segments and  $\mp V_w/2$  on the bottom ones while grounding the rest. Here  $R = 2$ . Read operations employ the same method.



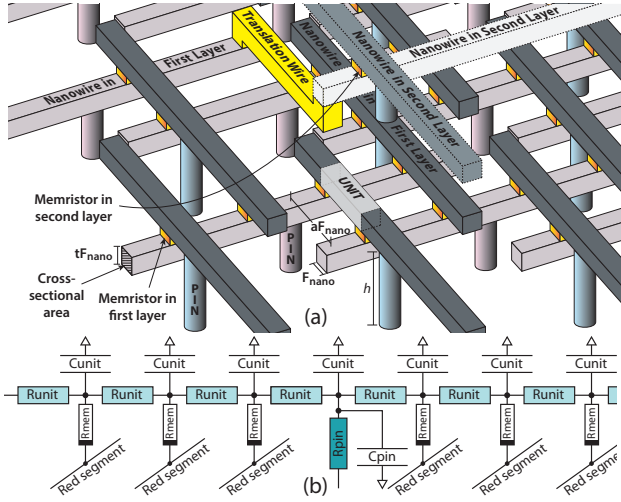


Fig. 8. (a) Partial structure of a crossbar using CMOL interface; (b) RC network for a single crossbar layer.

The nanowires and translation wires are partitioned into nanowire *units* of length  $aF_{\text{NANO}}$  and a resistor and a capacitor are associated with each unit. The resistance per unit  $R_{\text{unit}}$  can be extracted using the cross-sectional area and the resistivity  $\rho$  of the material:

$$R_{\text{unit}} = \rho \frac{aF_{\text{NANO}}}{tF_{\text{NANO}}^2} = \rho \frac{a}{tF_{\text{NANO}}} \quad (4)$$

For nanometric scales, the electrical resistivity of a material increases as the mean free path of the electrons in the bulk material becomes comparable to the dimensions of the structure. In this paper, the increment in the resistivity expected by the ITRS [1] is considered and used in Equation (4) to estimate  $R_{\text{unit}}$ .

For the capacitance per nanowire unit,  $C_{\text{unit}}$ , we use the results obtained in [24] in which it can be approximated as:

$$C_{\text{unit}} \approx (0.48 \times 10^{-10}) \varepsilon a F_{\text{NANO}} \quad (5)$$

where  $\varepsilon$  is the relative dielectric constant of the insulating material. For  $\text{SiO}_2$ ,  $\varepsilon = 3.9$ . The resistance and capacitance of the pins can be calculated as a function of its height  $h$  and its diameter  $F_{\text{NANO}}$  using:

$$R_{\text{pin}} = \rho_{\text{pin}} h / \pi F_{\text{NANO}}^2 \quad (6)$$

$$C_{\text{pin}} = 2\pi \varepsilon_{\text{ox}} h / \log(1 + t_{\text{ox}} / F_{\text{NANO}}) \quad (7)$$

where  $t_{\text{ox}}$  and  $\varepsilon_{\text{ox}}$  are the thickness and permittivity of the oxide surrounding the pin.

For a given feature size  $F_{\text{NANO}}$ , pitch  $aF_{\text{NANO}}$ , relative wire thickness  $t$  and geometrical parameter  $R$ , we can extract  $R_{\text{unit}}$ ,  $C_{\text{unit}}$ ,  $R_{\text{pin}}$  and  $C_{\text{pin}}$  and construct the RC network shown in Fig. 8(b). For simplicity we show a single crossbar. Multiple crossbar are connected adding an extra  $R_{\text{pin}}$  and  $C_{\text{pin}}$  for each blue nanowire and  $R \times R_{\text{unit}}$  and  $R \times C_{\text{unit}}$  for the pin translation wires connecting the red nanowires. The memristive devices were modeled based on the dynamic model proposed in [28]. For the sense circuitry, the latch-based comparator proposed in [29] was used.

## VI. EXPERIMENTAL RESULTS

For our simulations we assumed a maximum density crossbar with  $F_{\text{NANO}} = 25 \text{ nm}$  and relative thickness  $t = 3$ , which results in  $R_{\text{pin}} \approx 5 \Omega$ ,  $R_{\text{unit}} \approx 1 \Omega$ ,  $C_{\text{pin}} \approx 80 \text{ aF}$  and  $C_{\text{unit}} \approx 60 \text{ aF}$ . Memristors have HRS of  $2 \text{ M}\Omega$  and LRS of  $20 \text{ k}\Omega$ , and are accessed using  $V_w = 1 \text{ V}$ ,  $V_r = 0.8 \text{ V}$ , and  $t_{\text{write}} = 4 \text{ ns}$ . The non-linear characteristics are

TABLE I. DECODER DELAY AND POWER CONSUMPTION WITH  $R = 4$  AT SWITCHING ACTIVITIES OF 0.1 AND 1.0 AT 1 GHz.

$P = Q$	Delay	Power @ 0.1	Power @ 1.0
128	0.60 ns	1.9 mW	16 mW
256	0.72 ns	3.8 mW	32 mW
512	0.76 ns	7.7 mW	65 mW

based on the device reported in [3] in which  $I(V/2) \approx I(V)/100$  for  $V \approx 1 \text{ volt}$ .

1) *Crossbar Simulation*: In Fig. 9(a) we show the power and energy consumption while reading and writing a single memory element in a system with  $P = L = Q = 1$ . The experiment consists in (1) writing a LRS state (representing logic '1'), (2) a read operation, (3) writing a HRS (logic '0') and (4) another read operation. Since the energy consumed per operation depends on the initial state of the memory element, writing a LRS on a cell already in LRS consumes considerably more power than writing a HRS on a cell in HRS. In our analysis we consider the former to estimate the worst case scenario.

To evaluate the overall performance and power consumption of the memory system, the double decoder needs to be considered, in addition to the read and write circuitries, wires, and devices shown in Fig. 6. The double decoder was synthesized using Design Compiler with the 45 nm Nangate Open Cell Library [30]. Table I shows the delay and power dissipation of the decoder designed for different memory capacities running at 1 GHz with a switching activity of 1.0 and 0.1 for random inputs with equal 0 and 1 data probabilities. Figs. 9(b) and (c) shows the total energy per bit expended by the crossbar, decoder and the sense circuitry for larger arrays with  $1 \leq P \leq 32$  and  $1 \leq L \leq 4$ .

2) *Memory Banking*: Splitting the memory into memory banks may reduce the total energy per bit if the overhead of banking is not too high. An interesting case occurs when a memory of size  $P \times Q$  is partitioned into  $B$  banks each of size  $P/B \times Q$ . If we consider that the CMOS cells in the last multicell-row cannot be mapped, splitting the memory into  $B$  banks will result into  $B \times$  more cells that cannot be mapped, which reduces the total memory capacity. Using Equation (2) we can see that the normalized multicell capacity of a  $Q$ -bit word memory with  $P$  multicell-rows,  $L$  layers and  $B$  banks is:

$$\frac{\downarrow C_{\text{tot}}(P, B)}{R^4} = \frac{\downarrow C_{\text{tot}}(P/B, B)B}{R^4} = PLQ - \frac{B(L^2 - L)Q}{2} \quad (8)$$

where  $B(L^2 - L)Q/2$  is the loss in multicell memory capacity due to the banking. If we want the capacity to remain intact we could increase the original number of row multicells  $P$  to  $P' = P + \delta P$ .

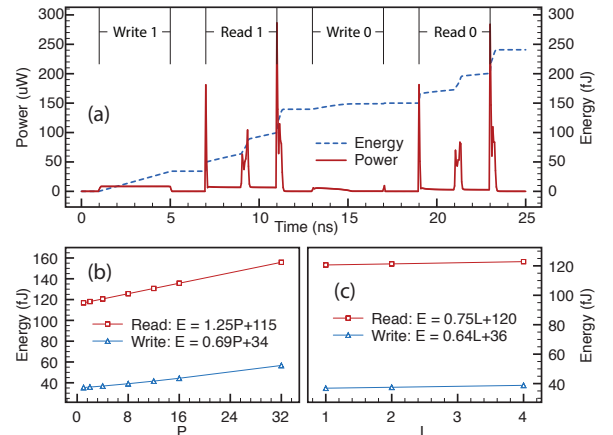


Fig. 9. (a) Energy and power for read and write. (b) and (c) energy/bit dependency as a function of  $P$  and  $L$ , respectively.

Using Equations (8) and (2) we can see that  $\delta P = (L - 1)(B - 1)/2$ . The energy per bit for a memory with constant capacity with  $P$  multicell-rows and  $B$  banks can be expressed in general as:

$$\downarrow E_{\text{bit}}(P', B) = \mu \frac{P + \delta P}{B} + \gamma B + \kappa. \quad (9)$$

where the constants  $\mu$  and  $\gamma$  represent the energy overhead per row and bank, respectively, whereas  $\kappa$  is the part of the energy that is independent of the size and number of banks. Based on Equation (9), the optimal number of banks is:

$$\downarrow B_{\text{opt}} = \sqrt{\mu(2P + 1 - L)/2\gamma} \quad (10)$$

As a practical example of Equation (10), let us find the optimal number of banks that minimizes the energy per bit in a memory of 1 GB. Assuming that the fabrication process allows us to use up to 32 crossbar layers with  $R = 4$  and due to area overhead limitations we can use up to 8 banks, we can build a 1 GB memory making  $P = Q = 1024$ . With this configuration, if  $B = 1$ , the memory will require  $\approx 1.5$  pJ/bit. Depending on the ratio  $\mu/\gamma$ , i.e., the ratio between the energy overhead of having an extra row and extra bank, the energy consumption can be reduced to  $\approx 0.4 - 1.0$  pJ/bit. It can be seen that power consumption of these memristive crossbar arrays is significantly lower than that of DRAM, that is reported to be 8-15 pJ/bit for comparable memory capacities [31].

## VII. CONCLUSION

In this paper we present a simple and flexible memory organization for memristive crossbar arrays based on the CMOL concept. The organization simplifies the decoding scheme and it facilitates the implementation of multi-crossbar-layer 3D memories. Moreover, an electrical model of a CMOL-based memory system is implemented and utilized to estimate its power consumption. By considering memory banking, our simulation results demonstrate the potential of CMOL-based crossbar arrays for future memory by showing one order of magnitude reduction in power consumption compared to DRAM.

## ACKNOWLEDGMENTS

This work was supported in part by the Air Force Office of Scientific Research under the MURI grant FA9550-12-1-0038.

## REFERENCES

- [1] "International Technology Roadmap for Semiconductors (ITRS, 2012 Updated Edition)," Tech. Rep., 2012. [Online]. Available: <http://public.itrs.net/>
- [2] A. Ghofrani, M. Lastras-Montaño, and K.-T. Cheng, "Toward large-scale access-transistor-free memristive crossbars," in *Asia and South Pacific Design Automation Conference*, Jan 2015, pp. 563–568.
- [3] Joshua Yang, J. et al., "Engineering nonlinearity into memristors for passive crossbar applications," *Applied Physics Letters*, vol. 100, no. 11, pp. 113 501–113 501, 2012.
- [4] D. Niu, C. Xu, N. Muralimanohar, N. P. Jouppi, and Y. Xie, "Design trade-offs for high density cross-point resistive memory," in *Proceedings of the 2012 ACM/IEEE international symposium on Low power electronics and design*. ACM, 2012, pp. 209–214.
- [5] M.-J. Lee, C. B. Lee, D. Lee, S. R. Lee, M. Chang, Hur et al., "A fast, high-endurance and scalable non-volatile memory device made from asymmetric Ta<sub>2</sub>O<sub>5-x</sub>/TaO<sub>2-x</sub> bilayer structures," *Nature materials*, vol. 10, no. 8, pp. 625–630, 2011.
- [6] D. B. Strukov, G. S. Snider, D. R. Stewart, and R. S. Williams, "The missing memristor found," *Nature*, vol. 453, no. 7191, pp. 80–83, 2008.
- [7] D. B. Strukov and H. Kohlstedt, "Resistive switching phenomena in thin films: Materials, devices, and applications," *MRS bulletin*, vol. 37, no. 02, 2012.
- [8] A. Ghofrani, M. Lastras-Montaño, and K.-T. Cheng, "Towards data reliable crossbar-based memristive memories," in *IEEE International Test Conference*, Sept 2013, pp. 1–10.
- [9] K.-H. Kim, S. Gaba, D. Wheeler, J. M. Cruz-Albrecht, T. Hussain, N. Srinivasa, and W. Lu, "A functional hybrid memristor crossbar-array/CMOS system for data storage and neuromorphic applications," *Nano letters*, vol. 12, no. 1, pp. 389–395, 2011.
- [10] C. Ho, C.-L. Hsu, C.-C. Chen, J.-T. Liu, C.-S. Wu, C.-C. Huang, C. Hu, and F.-L. Yang, "9 nm half-pitch functional resistive memory cell with < 1μA programming current using thermally oxidized sub-stoichiometric WO<sub>x</sub> film," in *Electron Devices Meeting (IEDM), 2010 IEEE International*. IEEE, 2010, pp. 19–1.
- [11] J. E. Green, J. W. Choi, A. Boukai, Y. Bunimovich, E. Johnston-Halperin, E. DeIonno, Y. Luo, B. A. Sherif, K. Xu, Y. S. Shin et al., "A 160-kilobit molecular electronic memory patterned at 10<sup>11</sup> bits per square centimetre," *Nature*, vol. 445, no. 7126, pp. 414–417, 2007.
- [12] S. Pi, P. Lin, and Q. Xia, "Memristor crossbar arrays with junction areas towards sub-10×10 nm<sup>2</sup>," in *Cellular Nanoscale Networks and Their Applications (CNNA), 2012 13th International Workshop on*. IEEE, 2012, pp. 1–2.
- [13] V. V. Zhirmov, R. Meade, R. K. Cavin, and G. Sandhu, "Scaling limits of resistive memories," *Nanotechnology*, vol. 22, no. 25, p. 254027, 2011.
- [14] W.-D. Li, W. Wu, and R. Stanley Williams, "Combined helium ion beam and nanoimprint lithography attains 4 nm half-pitch dense patterns," *Journal of Vacuum Science & Technology B: Microelectronics and Nanometer Structures*, vol. 30, no. 6, pp. 06F304–06F304, 2012.
- [15] C. J. Amsinck, N. H. Di Spigna, D. P. Nackashi, and P. D. Franzon, "Scaling constraints in nanoelectronic random-access memories," *Nanotechnology*, vol. 16, no. 10, p. 2251, 2005.
- [16] M. A. Zidan, H. A. H. Fahmy, M. M. Hussain, and K. N. Salama, "Memristor-based memory: The sneak paths problem and solutions," *Microelectronics Journal*, 2012.
- [17] C.-M. Jung, J.-M. Choi, and K.-S. Min, "Two-step write scheme for reducing sneak-path leakage in complementary memristor array," *Nanotechnology, IEEE Trans. on*, vol. 11, no. 3, pp. 611–618, 2012.
- [18] P. O. Vontobel, W. Robinett, P. J. Kuekes, D. R. Stewart, J. Straznicky, and R. S. Williams, "Writing to and reading from a nano-scale crossbar memory based on memristors," *Nanotechnology*, vol. 20, no. 42, p. 425204, 2009.
- [19] K. K. Likharev and D. B. Strukov, "CMOL: Devices, circuits, and architectures," in *Introducing Molecular Electronics*. Springer, 2005.
- [20] Q. Xia, W. Robinett, M. W. Cumbie, N. Banerjee, T. J. Cardinali, J. J. Yang, W. Wu, X. Li, W. M. Tong, D. B. Strukov et al., "Memristor-CMOS hybrid integrated circuits for reconfigurable logic," *Nano letters*, vol. 9, no. 10, pp. 3640–3645, 2009.
- [21] M. Payvand, A. Madhavan, M. A. Lastras-Montaño, A. Ghofrani, J. Rofeh, K.-T. Cheng, D. B. Strukov, and L. Theogarajan, "A Configurable CMOS Memory Platform for 3D Integrated Memristors," in *IEEE International Symposium on Circuits and Systems*. IEEE, 2015.
- [22] J. Rofeh, A. Sodhi, M. Payvand, M. A. Lastras-Montaño, A. Ghofrani, A. Madhavan, S. Yemencioğlu, K.-T. Cheng, and L. Theogarajan, "Vertical Integration of Memristors onto Foundry CMOS Dies using Wafer-Scale Integration," in *IEEE Electronic Components and Technology Conference*, 2015.
- [23] D. B. Strukov and R. S. Williams, "Four-dimensional address topology for circuits with stacked multilayer crossbar arrays," *Proceedings of the National Academy of Sciences*, vol. 106, no. 48, 2009.
- [24] D. B. Strukov and K. K. Likharev, "Prospects for terabit-scale nanoelectronic memories," *Nanotechnology*, vol. 16, no. 1, p. 137, 2005.
- [25] 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.
- [26] M. A. Lastras-Montaño, A. Ghofrani, and K.-T. Cheng, "HReRAM: A Hybrid Reconfigurable Resistive Random-Access Memory," *Proceedings Design, Automation, and Test in Europe (DATE), IEEE*, 2015.
- [27] D. B. Strukov and K. K. Likharev, "CMOL FPGA: a reconfigurable architecture for hybrid digital circuits with two-terminal nanodevices," *Nanotechnology*, vol. 16, no. 6, p. 888, 2005.
- [28] S. Shin, K. Kim, and S.-M. Kang, "Compact models for memristors based on charge-flux constitutive relationships," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 29, no. 4, pp. 590–598, 2010.
- [29] H. Jeon and Y.-B. Kim, "A CMOS low-power low-offset and high-speed fully dynamic latched comparator," in *SOC Conference (SOCC), 2010 IEEE International*. IEEE, 2010, pp. 285–288.
- [30] "Nangate open cell library," [Online] <http://www.si2.org/openeda.si2.org/projects/nangatelib>.
- [31] T. Vogelsang, "Understanding the energy consumption of dynamic random access memories," in *Proceedings of the 2010 43rd Annual IEEE/ACM International Symposium on Microarchitecture*. IEEE Computer Society, 2010, pp. 363–374.