Architecting Low Power Crossbar-Based Memristive RAM

Miguel Angel Lastras-Montaño Amirali Ghofrani Kwang-Ting Cheng
Department of Electrical and Computer Engineering, University of California, Santa Barbara
{mlastras, ghofrani, timcheng}@ucsb.ece.edu

Crossbar architecture

Crossbar-based memristive arrays are promising candidates for future high-density, low-power memories. Their structural simplicity allows them to be fabricated with pitches as small as 17 nm [6] and with projected reductions, according to the ITRS, to a few nanometers in the next decade [1].

A crossbar is particularly useful if two-terminal switching nano-devices with a nonlinear behavior are placed at each crosspoint and used them as memory and/or computing elements. There have been reported cases of bistable memristive devices [4, 3] that are suitable for such crossbar implementations.

With pitches of $2F_{\text{nano}}$, each nano-device in a crossbar will have a footprint of $4F_{\text{nano}}^2$ which can potentially be reduced to $4F_{\text{nano}}^2/L$ by stacking L crossbar layers. Such footprints translate into memory densities of more than 10^{12}cm^{-2} bits per crossbar layer and with projected densities exceeding 10^{13}cm^{-2} bits per layer in the next decade [1]. This contrasts to the footprint of a typical DRAM cell which falls between 6 to $8F_{\text{cmos}}^2$ (F_{cmos} is the half-pitch of a given CMOS process). This footprint results into memory densities of 10^{12}cm^{-2} without significant projected improvements in the next decade according to the ITRS [1].

Crossbar limitations

Being a passive array, a crossbar structure cannot be used as a stand-alone memory or computing system, it has to be part of a hybrid system in which the crossbar interacts with a CMOS-based subsystem. The mismatch between their feature sizes (F_{nano} and F_{cmos}), represents one limitation that prevents crossbar arrays from scaling to sizes that can be used as alternative memories to DRAM [2]. This limitation is schematically shown in Figure 1(a) and (b) in which the total size of the memory is dominated by the CMOS subsystem for address decoding.

The other limitation comes from the fact that when selecting a particular crosspoint in the array, other crosspoints will be partially selected. These partially selected crosspoints act as leakage points, which restricts the scalability of the crossbar by limiting the maximum number of crosspoints a nanowire can have [2]. There have been several studies [12, 5, 11] in which this limitation is addressed, but it remains an open problem.

CMOL architecture

We believe that the only suitable solution to overcome these two limitations is to adopt the CMOL architecture [7]. In CMOL, instead of having a lateral CMOS/crossbar interface as depicted in Figure 1(a) and (b), an area-distributed interface below a crossbar array is used instead, as shown in Figure 1(c). This interface consists of two rectangular arrays of *pins*, called *blue* and *red* pins hereafter, rotated by an angle α with respect to the direction of the crossbar nanowires [7]. A pair of adjacent blue and red pins, together with their access elements is what defines a CMOS *cell*.

CMOL presents many desired characteristics: it provides very high memory densities and bandwidth, it can be monolithically integrated with the CMOS subsystem, it shows excellent scalability, and it allows stacking multiple crossbar layers [9]. CMOL does not present the limitations of simple crossbars since the rotation between the crossbar and the pin interface eliminates the effect of the F_{nano} and F_{cmos} mismatch and it solves the scaling problem due to leakage currents by dividing the long nanowires into smaller constant-sized nanowire segments [7]. All these nice characteristics, however, come at a price: The CMOL architecture is significantly more complex than traditional memory arrays, which makes it difficult to organize it.

In [8], the authors present a high-level organization of a memory system constructed based on the CMOL architecture, in which a rectangular array of CMOL-based crossbar blocks are connected together along with their decoders. However, details of the actual implementation and various aspects for designing such blocks have not been worked out.

Proposed CMOL organization

Motivated by implementing a scalable 3D memory system based on the CMOL architecture, we identify several important geometric properties of CMOL and present a simple and flexible organization of such crossbar blocks that allows them to be used as stand-alone memories or as memory banks in a multi-bank memory. We find the regularity of a CMOL-based crossbar by introducing the division of a crossbar into equally-sized arrays of CMOS cells that we call *multicells*. This simple division fully exploits the benefits of CMOL and significantly simplifies the lateral decoders. Dividing the crossbar in a CMOL architecture into $P \times Q$ multicells as shown in Figure 2 allows us to read and write arbitrary Q-bit words and to have $L \leq P$ crossbar layers for implementing a multi-crossbar-layer 3D memory system. The total capacity in bits of such system will be:

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

where R is the integer that defines the rotation angle α:

$$\alpha = \arctan(1/R).$$

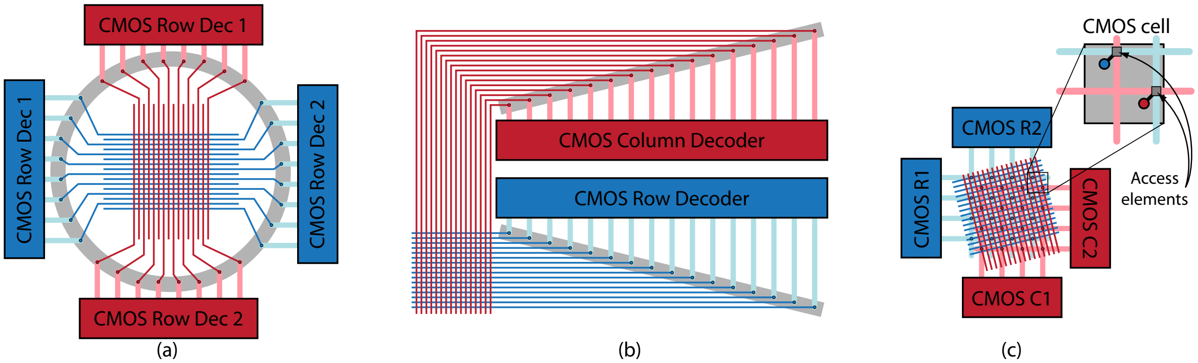

Figure 1: Different interfacing options between CMOS and a crossbar of size 16×16. (a) The typical solution for a simple crossbar; (b) an alternative used in [13]; (c) the CMOL approach.

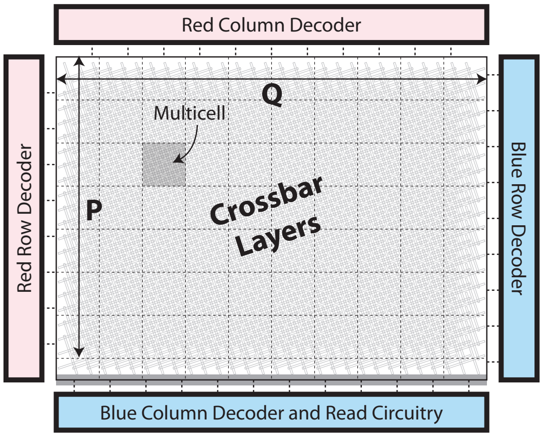

Figure 2: Crossbar division and lateral decoders.

Electrical modeling and validation

To validate the feasibility of a crossbar memory built using the organization proposed in this work, the multiple layers of crossbars were modeled as an RC network that is being driven by the CMOS cells underneath them, which in turn are controlled by the address lines of the lateral decoders.

Figure 3 shows the power and energy consumption when reading and writing a single memory element in a system with $R = 4$, $P = L = Q = 1$ and in Figure 4 we show the energy per bit expended per read and write in arrays with for $1 \leq P \leq 32$ and $1 \leq L \leq 4$. Note that reading requires more energy than writing due to the comparators that are used when reading.

Following the trend of Figure 4, a 1 GB memory can be built by having $P = Q = 1024$ with $R = 4$ and $L = 32$ and it will require ≈ 1.5 pJ/bit. This energy consumption can be reduced by splitting the memory into B banks with block having $(P/B) \times Q$ multicells. The actual reduction depends on the ratio between the energy overhead per multicell rows (P direction) and the energy overhead per banks, but it can be reduced to $\approx 0.4 - 1.0$ pJ/bit, which is significantly lower than the 8-15 pJ/bit that are needed in DRAM for comparable memory capacities [10].

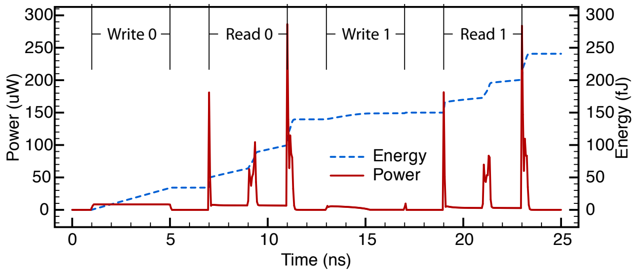

Figure 3: Energy and power for read and write.

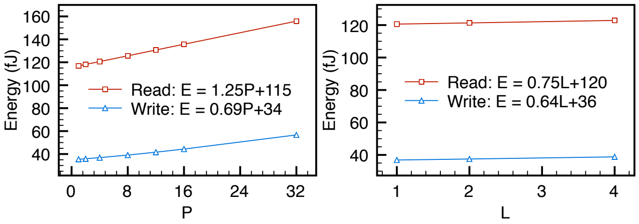

Figure 4: P and L dependency on the energy/bit.

1. REFERENCES

[1] International Technology Roadmap for Semiconductors (ITRS, 2010 Edition). Technical report, 2010.

[2] Amsinck, C.J. *et al.* Scaling constraints in nanoelectronic random-access memories. *Nanotechnology*, 16(10):2251, 2005.

[3] Jo, S.H. *et al.* High-density crossbar arrays based on a Si memristive system. *Nano Lett*, 9(2):870–874, 2009.

[4] Joshua Yang, J. *et al.* Engineering nonlinearity into memristors for passive crossbar applications. *Applied Physics Letters*, 100(11):113501–113501, 2012.

[5] Jung, C.M. *et al.* Two-step write scheme for reducing sneak-path leakage in complementary memristor array. *Nanotechnology, IEEE Transactions on*, 11(3):611–618, 2012.

[6] Jung, G.Y. *et al.* Circuit fabrication at 17 nm half-pitch by nanoimprint lithography. *Nano Letters*, 6(3):351–354, 2006.

[7] Likharev, K. *et al.* CMOL: Devices, circuits, and architectures. *Introducing Molecular Electronics*, pages 447–477, 2005.

[8] Strukov, D.B. *et al.* Prospects for terabit-scale nanoelectronic memories. *Nanotechnology*, 16(1):137, 2004.

[9] Strukov, D.B. *et al.* Four-dimensional address topology for circuits with stacked multilayer crossbar arrays. *Proceedings of the National Academy of Sciences*, 106(48):20155–20158, 2009.

[10] T. Vogelsang. Understanding the energy consumption of dynamic random access memories. In *Proceedings of the 2010 43rd Annual IEEE/ACM International Symposium on Microarchitecture*, pages 363–374. IEEE Computer Society, 2010.

[11] Vontobel, P.O. *et al.* Writing to and reading from a nano-scale crossbar memory based on memristors. *Nanotechnology*, 20(42):425204, 2009.

[12] Zidan, M.A. *et al.* Memristor-based memory: The sneak paths problem and solutions. *Microelectronics Journal*, 2012.

[13] Ziegler, M.M. *et al.* CMOS/nano co-design for crossbar-based molecular electronic systems. *Nanotechnology, IEEE Transactions on*, 2(4):217–230, 2003.