HReRAM: A Hybrid Reconfigurable Resistive Random-Access Memory

Miguel Angel Lastras-Montaño, Amirali Ghofrani, Kwang-Ting Cheng
Department of Electrical and Computer Engineering, University of California, Santa Barbara, CA 93106
{mlastras, ghofrani, timcheng}@ece.ucsb.edu

***Abstract*—Passive crossbar arrays of memristors have been identified as excellent alternatives for future random-access memories. One limitation is their inability of selecting a memory cell without the interference caused by the sneak-path currents from other partially selected cells, as it results not only in unnecessary waste of energy but also in larger current requirements. The complementary resistive switch (CRS), consisting in two anti-serially connected memristors, is considered a potential solution to the sneak-path problem. However, the destructive read operation and reduced endurance of the CRS render it unattractive for the otherwise excellent candidate for next-generation crossbar-based non-volatile memories. In this paper we explore the feasibility and tradeoffs of configuring part of the CRS memory into a memristive mode to mitigate these limitations. The inherent locality of memory accesses for most computer programs offers an opportunity for designing a cache-like adaptive CRS-based crossbar memory with hybrid configurations of CRS and memristive modes, enabling optimization for both endurance and energy consumption. Our simulation results validate that the proposed hybrid system achieves 1.5-7x reduction in energy consumption in comparison with a memristive-only memory system and significantly improves the endurance of the CRS-based memory.**

I. Introduction

Purely memristive non-volatile crossbar arrays are potential candidates as alternatives to existing non-volatile and volatile memories. Due to its unique characteristics, a memristor can provide better-than-NAND Flash storage densities with read and write operation speeds that are comparable to DRAM [17]. A purely memristive crossbar contains no active or select elements in the memory matrix, which makes it very competitive with other emerging memory technologies as well.

Purely memristive crossbar arrays, however, suffer from an inherent *sneak-path problem* due to partially selected devices [21]. Although it is possible to avoid the effect of the sneak-path problem during read operations on memristive-based crossbars [9], [21], the energy consumption and current requirements in the worst-case scenario while reading and writing impose design constrains that end up being the limiting factors for scaling.

The complementary resistive switch or CRS was proposed as a way to mitigate the sneak-path problem [8]. A CRS is formed by two anti-serially connected memristors and stores binary information as an internal configuration rather than as an actual resistance value, as done in a memristor. Both configurations in a CRS have a high resistance, which greatly reduces the data dependencies and parasitic current paths due to partially selected devices.

In contrast to the non-destructive read operation of a single memristor [13], the read operation in a CRS is destructive by nature and must be followed by a restore, i.e., write operation [8]. This type of write amplification negatively affects the endurance of CRS-based memories and results in excessive energy consumption.

Without addressing the destructive behavior of the read operation, a CRS-based memory does not appear to be an attractive alternative to existing memories. On the other hand, the attribute of a CRS to exhibit a high resistance independent of the stored binary value is very desirable as it mitigates the sneak-path problem and potentially reduces the power dissipation in a crossbar array.

A capacitive, non-destructive CRS read operation proposed by S. Tappertzhofen *et al.* [15] takes advantage of the different capacities of the two internal configurations of a carefully designed CRS cell. Besides the geometric memory cell restrictions this method imposes, this technique will not scale well as the capacitive component is expected to decrease with the miniaturization of the device and thus the margin to detect the state of the device.

In this paper we approach the issue with a different perspective. We exploit the unique property that a CRS can behave both as a CRS *and* as a memristor, a behavior experimentally observed in TaO$_x$-based memristors [19] as well as in anti-serially connected memristors [8]. By keeping frequently accessed cells in the *memristive mode* and others in the *CRS mode*, our proposal offers the following advantages: (1) non-destructive read operations for cells in the memristive mode, (2) reduced energy and current consumption by maintaining seldom accessed cells in the CRS mode, and (3) optimized endurance due to significantly fewer accesses to cells kept in the CRS mode. Based on this strategy we propose HReRAM, a hybrid reconfigurable resistive random-access memory, in which we can dynamically change the ratio and locations of cells in the CRS and memristive modes to optimize the energy consumption with respect to a computer program's memory requirements (amount and access patterns).

A. Summary of Contributions

- We propose for the first time to use both the CRS and the memristive behaviors to store binary data.
- We investigate the tradeoffs of configuring cells into one of these two modes in a passive crossbar-based memristive/CRS memory with respect to energy consumption and endurance.
- We present ideas and algorithms for the design of such hybrid memory system which take advantage of the locality of memory accesses for optimizing the overall endurance and energy consumption.

II. Background

A. Memristive devices

A memristor or memristive device is a generic term referring to any two terminal passive device in which the conductivity between its terminals can be reversibly changed between a high resistance (OFF) state and a low resistance (ON) state under the influence of an external electrical bias [12]. Although the actual resistive switching dynamics have been described by several mechanisms, without loss of generality, in this paper we focus on the observed reversible formation of conductive filaments in the thin insulating layer of a metal-insulator-metal (MIM) structure due to the redistribution of ions when a voltage is applied across its terminals [16], [18].

An important property of these MIM and other memristive devices is that the change in resistance is non-volatile and can be maintained for years after the external voltage is removed [17]. Figs. 1(b) and (a) show the typical bipolar I-V linear characteristics and the realization of a MIM memristor, respectively [17].

978-3-9815370-4-8/DATE15/©2015 EDAA

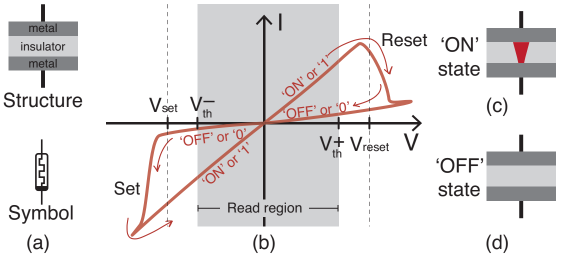

Fig. 1. (a-b) Simple memristor realization with a typical bipolar I-V linear curve. (c-d) The resistance of the device is determined by the presence or absence of a conductive filament (shown in red).

To *program* or *set* the device into a memristive 'ON' or '1' state, a voltage $V_{\rm set}$ below a negative threshold $V_{\rm th}^{-}$ is applied across its terminals until a conductive filament is formed, shorting the device and lowering its resistance (Fig. 1(c)). To *erase* or *reset* a device into a memristive 'OFF' or '0' state, a voltage $V_{\rm reset}$ of opposite polarity and above a positive threshold $V_{\rm th}^{+}$ is applied, dissolving the filament and increasing the resistance of the device (Fig. 1(d)).

Applied voltages between $V_{\rm th}^{-}$ and $V_{\rm th}^{+}$, as shown in grey in Fig. 1(b), will not change the resistance, thus state, of the device since the ionic mobility depends super-exponentially on the applied voltage [13]. This allows us to *read* without disturbing the state of the device by applying a voltage $V_{\rm read}$ within this grey region and measuring the resulting current.

B. Memristive-based random-access memories

A memristive-based *random-access memory* (RAM) is organized in a matrix (crossbar) of memristors to allow short access times that are independent of the location of the data. Such memory matrices can be either *active* or *passive* based on whether or not each memory element contains an active switch (select element), typically a transistor or diode, to isolate each memory cell from the rest.

Purely memristive passive crossbars, known as 0T1R for having 0 transistors per 1 resistive element, are especially interesting as their cell size is not limited by the size of a select element (which is often dominant) and $4F^2$ cells are possible [1], where F is minimum feature size. The downside is that they suffer from the leakage of partially selected devices, which increases the energy consumption and current requirements as a function of the size of the memory, making larger memories impractical beyond a few KB. Active memristive crossbars or 1T1R (for 1 transistor per 1 resistive element) allow the individual isolation of the cells but negatively affect the size of the cells as the select element is predicted to dominate the cell size [1].

A practical alternative is to use arrays of quasi-passive 1TnR crossbars in which 1 transistor is shared by n resistive elements. Such approach shares similar benefits of 0T1R crossbars' high density and, by limiting the number of line-shared devices to n, the energy consumption and current requirements can also be constrained independently of the memory size. The CMOL architecture [7], introduced as a way to interface a standard CMOS die with molecular-sized devices, allows monolithic integration of 1TnR crossbars with a standard CMOS chip to achieve a very high crossbar density. Throughout this paper we assume the proposed ideas to be implemented in CMOL crossbars but for the sake of visual simplicity we present them using arrays of 1TnR crossbars.

C. Sneak-path and parasitic current path problems

As shown in Fig. 2(a), when selecting a target device in a passive (or quasi-passive) crossbar to read it, we also partially select other line-shared devices. Depending on the resistance of these other cells, a sneak current $I_{\rm sneak}$ may interfere with the ability of the external sensing circuitry that is trying to discern $I_{\rm target}$ from $I_{\rm read}$. The sneak-path problem during the read operations can be eliminated by applying the read voltage to one of the lines, grounding the rest, and measuring the currents at the end of the lines [9], [21], as shown in Fig. 2(b). Since for practical applications a read circuit will be shared by several lines (say n of them), the approach in Fig. 2(b) results in unnecessary waste of resources as on average it will consume n times more energy and require n times larger currents than those consumed for reading a single element.

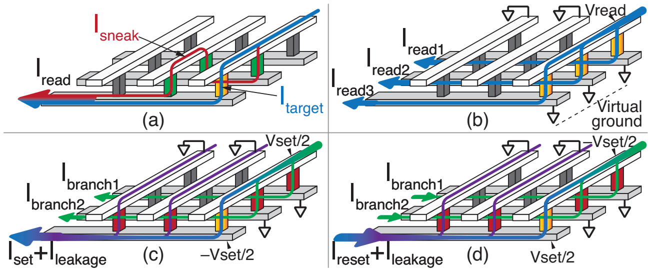

Fig. 2. Sneak-path and branching current problems. (a) Reading with floating terminals, (b) reading with terminals at a known potential, (c) set procedure and (d) reset procedure.

Fig. 2(c) and (d) show the procedure to program (set) and erase (reset) a device using the $V/2$ scheme in which $V/2$ and $-V/2$ voltages are applied on the target memristor's terminals while other lines are grounded. In this case the leakage branching currents during the set and reset operations result in, on average, $n/2$ times more energy and n times larger currents. Again, this scheme is impractical for larger arrays.

D. Complementary resistive switches

The complementary resistive switch or CRS was proposed by E. Linn *et al.* in 2010 as a way to solve the sneak path problem and to minimize the waste of resources due to other parasitic currents in purely resistive crossbars [8]. Conceptually, a CRS consists in two anti-serially connected memristors, as shown in Fig. 3(a), but its behavior has also been observed in TaO$_x$-, TiO$_x$- and HfO$_x$-based systems within a single memory cell [19], [14], [2]. In contrast to a memristor whose resistance could be either low or high depending on the logic value it represents, the resistance of a CRS is high for both logic values as at least one of the memristors in the CRS is kept OFF. As shown in Fig. 3(c), if the top device is OFF and the bottom device is ON, the CRS represents logic '1'. Conversely, if the top device is ON and the bottom device is OFF, the CRS represents logic '0'. If both devices are ON, the CRS is said to be 'ON', indicating a low resistance. As originally conceived, the 'ON' state of a CRS is not used to represent logic values and is present only during the 0 to 1 and 1 to 0 transitions and during the readout procedure [8]. The state of both top and bottom devices being OFF will never be reached during the normal operation of the CRS [8]. This state can only occur in pristine devices right after fabrication.

As schematically shown in the CRS I-V curve in Fig. 3(b), to switch from the '0' to '1' states, a voltage higher than $V_{\rm th,2}^{+}$ should be applied. For the '1' to '0' transition, a negative voltage below $V_{\rm th,2}^{-}$ is applied instead. To read its state, a voltage in the range $[V_{\rm th,2}^{-}, V_{\rm th,1}^{-}]$ is applied (which will force the top memristor in the CRS to be ON) and the resulting current is measured. Within this voltage range, known as the CRS *read window* [11], if a high (low) current is sensed, the CRS was in a '1' ('0') state. Note that the range $[V_{\rm th,1}^{+}, V_{\rm th,2}^{+}]$ could have been used as well, for which a high (low) current represents a '0' ('1'). Also note that a read to a CRS could be a destructive operation (as one of the two logic states will be altered after a

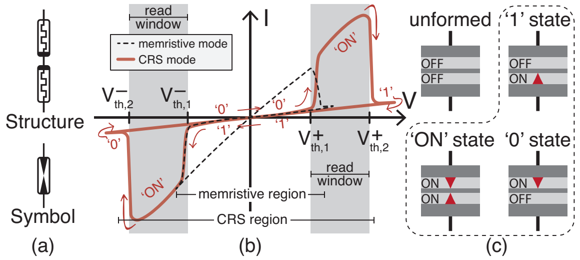

Fig. 3. Complementary resistive switch. (a) its structure and symbol, (b) typical I-V curve and (c) memory states with their corresponding top/bottom resistances.

read operation) and thus must be followed by a restoring write operation, which consumes additional energy and also negatively affects the limited endurance of CRS cells.

E. CRS cell's dual memristive/CRS behavior

Applying a voltage in the left (right) read window to a CRS cell in the '1' ('0') state (shown in gray in Fig. 3(b)) effectively sets the CRS into a low resistance state (i.e., ON state). This set procedure has a similar effect to the set procedure of a single memristor device (shown in Fig. 1(b) and reproduced with a dashed line in Fig. 3(b)). In fact, as experimentally shown by Y. Yang *et al.* for TaO$_x$-based single cell devices, the CRS and memristive behaviors can coexist in the same device by simply choosing different voltage ranges of operation [19]. In their devices, if voltages are applied in the range $[-1.4, 1.0]$ V, a reproducible memristive behavior can be achieved. By applying voltages in the range $[-2.0, 2.0]$ V the CRS behavior can be observed instead. In this way we can conclude that a memristive behavior is embedded in the more general CRS behavior. For the rest of this paper we explore intelligent use of such CRS devices which exhibit both CRS and memristive behaviors as memory elements. We will refer to them as *CRS devices* or simply as *devices*.

III. Motivation and Proposal

From the previous discussion in Section II and the qualitative comparison summarized in Fig. 4, it should be clear that neither a memristive-based nor a CRS-based memory can fulfill all the desired properties for next-generation memories. On one hand, the destructive read operation of CRS memories affects not only the devices' endurance but also degrades their performance and energy consumption due to the need of restoring the data after read as well as the higher voltages required for their memory operations. On the other hand, for memristive-based crossbars, the wide variation in the read and write currents imposed by the partially selected devices limits the crossbar's maximum size and incurs higher energy consumption on the partially selected devices.

With this in mind, we propose HReRAM, a hybrid reconfigurable resistive random-access memory consisting of crossbars of CRS devices and explore the use of both memristive and CRS modes to store information in these devices. The general idea is to keep those frequently accessed memory cells in the *memristive mode*, and the rest in the *CRS mode*. Storing data which require frequent access and modification in devices configured to the memristive mode can avoid the high energy overhead and endurance degradation due to higher operating voltages and the destructive read in the CRS mode. Storing infrequently accessed data as well as unused memory regions in devices under the CRS mode can mitigate the sneak-path problem. In this way we can combine the strengths of both types of devices and avoid their weaknesses (see Fig. 4).

IV. Hybrid Reconfigurable Resistive RAM

HReRAM is formed by several arrays of 1TnR crossbars interfacing with CMOS read/write subsystems using the CMOL interface. The crossbars use CRS devices that exhibit dual memristive/CRS behaviors. Our system could use CRS devices formed by anti-serially connected memristors as well, but the use of single cell CRS devices is preferred as they are structurally simpler [19].

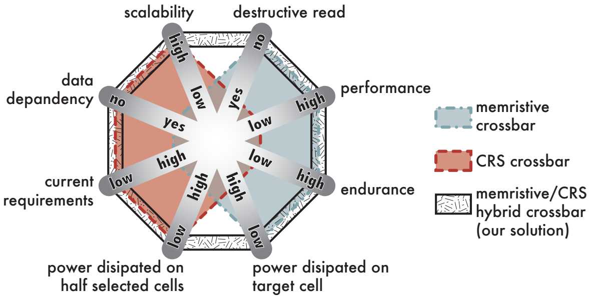

Fig. 4. Qualitative comparison of memristive-based and CRS-based crossbars with respect to the key properties for memory applications. Neither approach fulfills every aspect. Our hybrid solution attempts to combine the better parts of both worlds and avoid their respective drawbacks.

A. General idea

Reducing the maximum number of ON (low resistance) devices per line in $n \times n$ 1TnR crossbars reduces the overall energy consumption and their current requirements which grow linearly with n. In light of this, we define a parameter k as the number of devices in the memristive mode in each line. Thus, $n - k$ is the number of devices in the CRS mode and $m = k/n$ is the *memristive fraction* or *active fraction* of the crossbar. The parameter k could be anything between 0 and n with $k = 0$ corresponding to a fully CRS-based crossbar and $k = n$ a fully memristive crossbar.

As a graphical illustration, Fig. 5 shows the possible crossbar configurations for different values of k in a 1TnR crossbar with $n = 4$. Considering the case of $k = 1$, if an application program only reads and writes devices that fall in the memristive fraction of the crossbar, it corresponds to the best case scenario: (1) read accesses are non-destructive as they are done in memristive cells, (2) write operations are also more efficient and requires a low write voltage, and (3) partially selected devices are all in the CRS mode and thus incur the minimum current and energy.

Note that $k = 1$ limits the *active* memory to only $1/n$ of the total memory capacity. If a program needs a larger memory space, there are two options. First, we stick with the same k and thus the same active memory fraction and swap the fraction of memory that is in the memristive mode with another in the CRS mode. In this way we can address any region of the crossbar while still providing the same benefits for a given k. Note that swapping memory cells between the memristive and CRS modes does not require actual data movement. Instead, it simply requires the change of modes from one to the other while maintaining the original data in the memory cells. Such swapping, however, does incur some performance and energy overheads and thus should be minimized. As the second option, we can increase k, effectively increasing the amount of active memory which in turn could potentially reduce the rate at which we need to swap between modes and thus reduce the swapping overhead. The downside is that the energy and current caused by the partially selected devices during read/write operations will be higher for a larger k. The optimal value of k depends on the memory usage pattern and amount, and can be dynamically changed during the lifetime of an application.

Analogous to the management of the memory system consisting of traditional main memory (DRAM) and secondary disk storage, we can swap information between the memristive and CRS modes in a process that is conceptually similar to a page

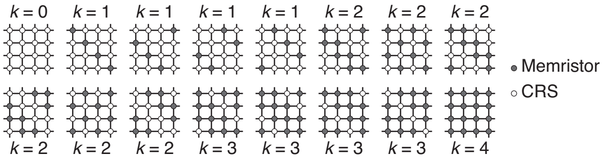

Fig. 5. Possible memristive/CRS configurations for 1TnR crossbars with $n = 4$. The corresponding CMOS read/write subsystems are not shown.

fault mechanism. Upon memory requests, if its page is in the memristive mode, we have a *memristive hit* and could proceed to access the page. If it is a *memristive miss*, the memory cells holding the page are first changed into the memristive mode and then are accessed.

Rather than being a complete replacement to the DRAM/disk systems, we envision HReRAM as a system that fits between a small DRAM buffer and a secondary Flash/Disk storage system [10], however, for more aggressive memristive device characteristics, HReRAM could sit closer to the processing unit and effectively serve as a universal memory.

B. Data Representation

A binary value can be represented in two ways in HReRAM: in a memristive mode or in a CRS mode. We switch between modes using the *activation* and *deactivation* procedures. The activation (deactivation) procedure changes data from the CRS (memristive) to the memristive (CRS) mode.

Based on the internal configuration of the CRS filaments and with a notation of top/bottom elements in a CRS device, the following shows the representation of a logic 1 and a logic 0 in the memristive and CRS modes, as well as the write voltage that needs to be applied to achieve them:

Mode	Logic 1 / Voltage	Logic 0 / Voltage
Memristive	ON/ON $[V_{\text{th},2}^{-}, V_{\text{th},1}^{-}]$	OFF/ON $[V_{\text{th},1}^{+}, V_{\text{th},2}^{+}]$
CRS	OFF/ON $V > V_{\text{th},2}^{+}$	ON/OFF $V < V_{\text{th},2}^{-}$

If a device is in the memristive mode, its elements can only be in the ON/ON or OFF/ON states. Note that we could have also used the ON/OFF state to represent the logic 0 in the memristive mode. With that, we should have also swapped the write voltages for the logic 1 and 0 in the memristive mode.

C. Read, Write and Deactivation Operations

Fig. 6(a) shows our proposal for the read procedure. On a memristive hit, we simply read the memory by applying a read voltage V_{read} in the range $[V_{\text{th},1}^{-}, V_{\text{th},1}^{+}]$ and sensing the current as shown in Fig. 2(c). A high current represents a logic 1 and a low current a logic 0. On a memristive miss, we (1) apply a voltage V_{set} in the range $[V_{\text{th},2}^{-}, V_{\text{th},1}^{-}]$ and (2) apply V_{read} and sense the current. If the data was originally in a CRS logic 1 (OFF/ON) state, step (1) will lower the resistance of the device and set it into a memristive logic 1 (ON/ON) state. If the data was instead a CRS logic 0 (ON/OFF), step (1) will not modify the configuration of the device and the subsequent reading step (2) will sense a high resistance and (correctly) interpret it as a memristive logic 0. Note that although we read a memristive logic 0, the device still has a CRS logic 0 (ON/OFF) and not a memristive logic 0 (OFF/ON). A final write step (3) to CRS logic 1 (OFF/ON) with a voltage $V > V_{\text{th},2}^{+}$ is needed to have a consistent memristive logic 0.

Figs. 6(b) and (c) show the proposed procedures for writing a memristive logic 1 and logic 0, respectively. On a memristive hit we apply V_{set} for logic 1 or V_{reset} in the range $[V_{\text{th},1}^{+}, V_{\text{th},2}^{+}]$ for logic 0. On a memristive miss, if writing a memristive logic 1, we write a CRS logic 1 with a voltage $V > V_{\text{th},2}^{+}$ followed by a memristive logic 1 with V_{set}. To write a memristive logic 0 on a memristive miss, it is sufficient to write a CRS logic 1 since a memristive logic 0 and a CRS logic 1 are represented by the same (OFF/ON) configuration.

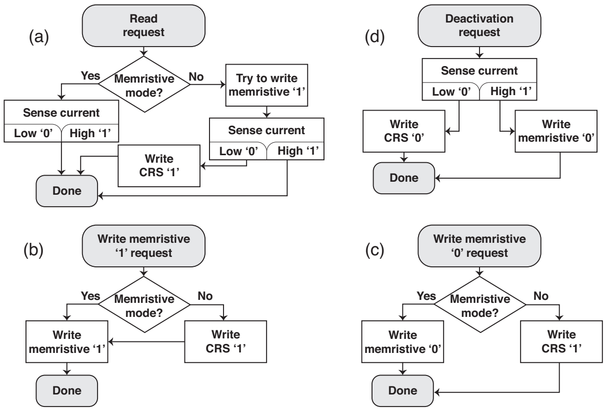

Fig. 6. Read, write and deactivation procedures.

The deactivation procedure to change a page from the memristive to the CRS mode is shown in Fig. 6(d). We (1) sense the current by applying V_{read} and (2) write the value read in step (1) with V_{reset} for logic 1 and $V < V_{\text{th},2}^{-}$ for logic 0.

D. Exploiting the Locality of Memory Accesses

For typical applications, when a process requests M amount of memory, only a fraction of it, denoted as F, will constitute the *working set* of the process [5]. Our goal is to keep the fraction F in the memristive mode and $1 - F$ in the CRS mode. In this way, most of the memory requests will fall within the memristive mode and only a small fraction of the requests in the CRS mode.

Modern operating systems are optimized to minimize page traffic between DRAM and secondary storage by keeping in memory only the working set of the applications. In this case, minimizing the page traffic translates into performance improvement. For HReRAM, the performance penalty of accessing memory cells in the CRS mode instead of the memristive mode is not very significant. The main tradeoffs between these two configurations are on the energy consumption and the endurance of the memory cells.

V. Energy Model

To analyze the energy savings of HReRAM with respect to a memristive-only memory and to compare it between a memristive-only and a CRS-only memory, we first define a set of relevant parameters in Table I. The *hit rate* h quantifies the percentage of memory accesses that address memory cells in the memristive mode. The rest $1 - h$ is for accesses that address cells in the CRS mode. We use m to express the fraction of total memory in the memristive model and p to represent the percentage of data stored in the memory being logic '1'. We define ϵ as the read energy consumed by a single memristor in the ON state. We further express the set and reset energies for a memristor as well as the write energy of a CRS as linear functions of ϵ, with multiples of S, R and C, respectively. Assuming linear devices, given the energy consumption of a device in the ON state E_{on}, the energy in the OFF state E_{off} can be predicted by dividing E_{on} by r. These parameters can be used to express the power dissipation in the crossbar array, not including that of the CMOS subsystem.

The average energy consumed by a 1TnR crossbar when reading a memristor (shown in Fig. 2(b)) can be expressed as:

$$\overline{\mathrm{E_r}} = \epsilon p + \frac{\epsilon}{r}(1-p) + \Big[\epsilon m p + \frac{\epsilon}{r}(1-mp)\Big]n'. \qquad (1)$$

TABLE I. DESCRIPTION OF PARAMETERS FOR POWER ANALYSIS.

Parameter	Description
h (hit rate)	% of accesses falling in memristive mode
$1-h$	% of accesses falling in CRS mode
m	Memory fraction in memristive mode
$1-m$	Memory fraction in CRS mode
p	% of data in the memory being logic 1
$1-p$	% of data in the memory being logic 0
$n \times n$	Size of a 1TnR crossbar
n'	# of partially selected devices: $n-1$
k	# of devices in memristive mode per line
r	$\mathrm{R_{off}/R_{on}}$ ratio of a memristor
ϵ	Read energy of an ON memristor
ϵ/r	Read energy of an OFF memristor
$\epsilon \times S$	Set energy of a memristor (OFF to ON)
$\epsilon \times R$	Reset energy of a memristor (ON to OFF)
$\epsilon \times C$	Write energy of a CRS (1 to 0 or 0 to 1)

For a write operation (as shown in Fig. 2(c-d)), the average energy consumed in a crossbar can be expressed as:

$$\overline{\mathrm{E_w}}(x,y) = \epsilon yx + \frac{\epsilon}{r}(1-y)x + \left[\frac{\epsilon}{2}mp + \frac{\epsilon}{2r}(1-mp)\right]n'x \quad (2)$$

where x is either S, R or C depending on the operation being performed and y is the probability of the target cell having a logic '1'. With (1) and (2), and according to Fig. 6, the deactivation and activation energies can be expressed as:

$$\overline{\mathrm{E_d}} = \overline{\mathrm{E_r}} + p\overline{\mathrm{E_w}}(R,1) + (1-p)\overline{\mathrm{E_w}}(C,0) \quad (3)$$

$$\overline{\mathrm{E_a}} = p\overline{\mathrm{E_w}}(S,p) + \overline{\mathrm{E_r}} + (1-p)\overline{\mathrm{E_w}}(C,0) \quad (4)$$

respectively, and the total average read energy of a memristive crossbar can be expressed as:

$$\overline{\mathrm{E_{Read}}} = h\overline{\mathrm{E_r}} + (1-h)(\overline{\mathrm{E_a}} + \overline{\mathrm{E_d}}) \quad (5)$$

VI. CASE STUDY

To gain better insights to the tradeoffs, we identified values for various parameters for conducting a meaningful case study. Based on the ITRS 2013 tables for Emerging Research Devices [1], we assume V_{read} of 0.1 V, a read time t_{read} of 20 ns, set voltage V_{set} of 1.0 V and set time t_{set} of 2 ns. Since ϵ can be computed as $\epsilon = t_{\mathrm{read}} \cdot V_{\mathrm{read}}^2/\mathrm{R_{on}}$ and the set energy can be approximated as $\epsilon \times S = t_{\mathrm{set}} \cdot V_{\mathrm{set}}^2/\mathrm{R_{on}}$, we found that $S \approx 10$. The ITRS also reports on reset energies that are 8x larger than the set energies, thus $R \approx 80$. We set $C \approx 90$ as it involves a set and reset and further assumed $p = 0.5$, a $r = \mathrm{R_{off}/R_{on}}$ of 100 and 1TnR crossbars of size 100×100, i.e., $n = 100$.

A. Fully CRS vs. Fully Memristive Crossbar

We first compare a CRS-only ($k = 0$) crossbar and a memristive-only ($k = n$) crossbar. In Fig. 7(a) we show the $\overline{\mathrm{E_{Read}}}(\text{memristive})/\overline{\mathrm{E_{Read}}}(\text{CRS})$ energy savings space compared with a fully memristive crossbar as a function of the crossbar size n and the $\mathrm{R_{off}/R_{on}}$ ratio r. The region with red tones (top/right) in Fig. 7(a) contains configurations in which a CRS crossbar is more energy efficient than a memristive crossbar, whereas the region with blue tones (bottom/left) indicate configurations where a memristive crossbar is better. Assuming a maximum current per 1TnR crossbar line of 1 mA and the different $\mathrm{R_{on}}$ and $\mathrm{R_{off}}$ resistances reported in [6] for several memristors, The shaded area in Fig. 7(a) shows the space that can be practically achieved with existing devices. From this analysis, we found that the energy savings of a CRS-only memory can only reach up to ≈1.4x, at the expense of significant reduction in the endurance of the memory elements.

B. Hybrid Memristive/CRS Crossbar

We now explore the energy-saving benefits of HReRAM in which a fraction m of the memory is kept in the memristive mode and the rest $1 - m$ in the CRS mode. Fig. 7(b/top) shows the $\overline{\mathrm{E_{Read}}}(\text{memristive})/\overline{\mathrm{E_{Read}}}(\text{hybrid})$ energy savings space as a function of m and the memristive hit rate h. The curved dashed lines are the energy isolines at 1x down to $1/32$ x. As expected, lower memristive hit rates result in larger energy-wise penalties. For instance, if an application uniformly accesses all the memory space, as shown with a diagonal dashed-dotted line in Fig. 7(b/top), in the worst case (at $m \approx 0.5$), the memory will consume 19x more energy than the memristive baseline. In order to achieve actual energy savings we need to have memory accesses with high memristive hit rates (>90%) at lower memristive fractions (<10%), as detailed in Fig. 7(b/bottom) with energy isolines in the range of $1/2$ x to 16x.

To quantify the expected energy savings of HReRAM during typical workloads, we utilize the power-law cache model [3] in which the cache miss of a cache of size s is given by $as^{-\gamma}$, where a and γ are constants. Ref. [4] extends the power-law to GB-scale caches and finds that for commercial workloads the cache miss is approximately given by $0.9395s^{-0.5966}$ for a cache of size s MB. With this model we computed the cache hit as $1-$the cache miss and mapped the resulting curve into the energy savings space of Fig. 7(b) enforcing the boundary conditions that $h = 0$ if $m = 0$ and $h = 1$ if $m = 1$. The solid lines in color are the curves obtained with caches of size 512 MB to 16 GB.

The energy savings along these curves are shown in Fig. 7(c/top) and in its detail in 7(c/bottom). As expected, memories with greater capacities provide better energy savings that are ranging from 7x for a memory of 16 GB to 1.5x for a memory of 512 MB. In all cases, the maximum energy saving occurs for $m < 0.1$ and at different values of m. In general, different applications will produce different hit rate curves with a maximum energy saving occurring at a different memristive fraction. The optimal memristive fraction could be computed beforehand in a profiling stage, but it can also be dynamically estimated at run-time with low overhead by computing the miss rate curve [20], mapping it to the energy saving space and finding the value of m that produces the minimum energy consumption. Dynamically estimating the optimal m has the advantage that the memory system will be able to reconfigure itself as a response to changes in the memory requirements of an application. For instance, if an application changes from a phase with high locality in memory references to a phase with lower locality, the memory should increase the memristive fraction to increase the memristive hit rate and thus reduce its energy consumption. For the case of memory accesses with uniform distribution, it should be clear that the best configuration will occur at $m = 1$ with no energy savings but also with no penalties. On the other hand, if we restrict the memory accesses to only regions of the memory that are in the memristive mode, i.e., $h = 1$ regardless of the memristive fraction m, we will obtain energy savings of more than 30x at very low memristive fractions, as shown by the 'ideal' case in Fig. 7(c/top) with a dashed line. Although it is hard to imagine that such scenario is even possible, it can be achieved by having a sufficiently large memory and restricting its use to only a small fraction of it. For instance, if we have a HReRAM system with 32 GB of memory and use only $1/16$ of it, we will obtain an energy saving of 11x whereas restricting it to $1/32$ further increases the saving to 16x. In all these cases with an endurance that is virtually the same as the endurance of a memristive-only memory since all the memory accesses are restricted to memory cells that are in the memristive mode.

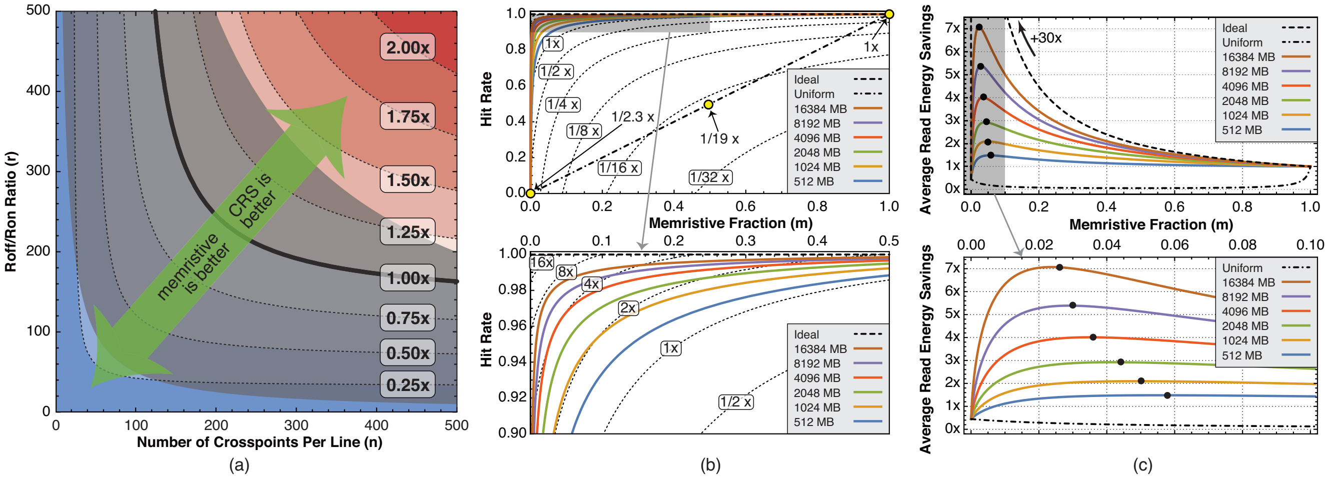

Fig. 7. (a) Normalized (memristive/CRS) average read energy savings as a function of the size of the crossbar n and the R_{off}/R_{on} ratio r. The shaded region covers space that can be achieved with realistic devices [6]. CRS-only offers minimal energy savings at the expense of a poor endurance of the device. (b) Normalized (fully-memristive/hybrid) average read energy savings space as a function of the memristive fraction m and memristive hit rate h showing for different hit rate curves. (c) Average read energy savings for the different curves. The bottom plots in (b) and (c) show the highlighted detail in the top plots.

VII. Conclusion

In this paper, we investigate the benefits of utilizing the CRS's unique capability of behaving both as a CRS and as a memristor to store data in a memory system. We present HReRAM, a hybrid reconfigurable resistive random-access memory that uses CRS crossbars to store binary data in mixed CRS and memristive modes in the same memory system. By keeping frequently used devices in the memristive mode and less frequently used as well as unused devices in the CRS mode, HReRAM exhibits a cache-like structure, saves energy consumption, alleviates the sneak-path problem, and increases device endurance. Not only having the key benefits of a cache-like organization for memory management, HReRAM does not require any physical data movement between the active and inactive parts of the memory, as it only requires to change the modes (memristive or CRS) of the corresponding devices. Furthermore, by dynamically changing the fraction of total active memory in the memristive mode, HReRAM can be tuned to accommodate to the application's changing memory requirements. Our simulation results show that for commercial workloads HReRAM can offer an average energy saving of up to 7x for a 16 GB memory while offering an endurance that is virtually the same as a memristive-only memory.

Acknowledgments

The first author would like to thank Luis Lastras for all the insight and expertise provided while discussing the ideas of this paper and for suggesting references [4] and [10]. This work was supported in part by the Air Force Office of Scientific Research under the MURI grant FA9550-12-1-0038.

References

[1] "The International Technology Roadmap for Semiconductors (ITRS), System Drivers, 2013, http://www.itrs.net/."

[2] Y. C. Bae, A. R. Lee, J. B. Lee, J. H. Koo, K. C. Kwon, J. G. Park, H. S. Im, and J. P. Hong, "Oxygen Ion Drift-Induced Complementary Resistive Switching in Homo TiOx/TiOy/TiOx and Hetero TiOx/TiON/TiOx Triple Multilayer Frameworks," *Advanced Functional Materials*, vol. 22, no. 4, pp. 709–716, 2012.

[3] C. Chow, "Determination of cache's capacity and its matching storage hierarchy," *Computers, IEEE Transactions on*, vol. 100, no. 2, pp. 157–164, 1976.

[4] D. Daly, P. Dube, K. El Maghraoui, D. Poff, and L. Zhang, "A hybrid approach for large cache performance studies," in *Quantitative Evaluation of Systems (QEST), 2011 Eighth International Conference on*. IEEE, 2011, pp. 47–56.

[5] P. J. Denning, "The working set model for program behavior," *Communications of the ACM*, vol. 11, no. 5, pp. 323–333, 1968.

[6] O. Kavehei, E. Linn, L. Nielen, S. Tappertzhofen, E. Skafidas, I. Valov, and R. Waser, "An associative capacitive network based on nanoscale complementary resistive switches for memory-intensive computing," *Nanoscale*, vol. 5, no. 11, pp. 5119–5128, 2013.

[7] K. K. Likharev and D. B. Strukov, "CMOL: Devices, circuits, and architectures," in *Introducing Molecular Electronics*. Springer, 2005, pp. 447–477.

[8] 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.

[9] J. Mustafa and R. Waser, "A novel reference scheme for reading passive resistive crossbar memories," *Nanotechnology, IEEE Transactions on*, vol. 5, no. 6, pp. 687–691, 2006.

[10] M. K. Qureshi, V. Srinivasan, and J. A. Rivers, "Scalable high performance main memory system using phase-change memory technology," *ACM SIGARCH Computer Architecture News*, vol. 37, no. 3, pp. 24–33, 2009.

[11] S. Schmelzer, E. Linn, U. Bottger, and R. Waser, "Uniform complementary resistive switching in tantalum oxide using current sweeps," *Electron Device Letters, IEEE*, vol. 34, no. 1, pp. 114–116, 2013.

[12] D. Strukov and H. Kohlstedt, "Resistive switching phenomena in thin films: Materials, devices, and applications," *MRS bulletin*, vol. 37, no. 02, pp. 108–114, 2012.

[13] D. B. Strukov and R. S. Williams, "Exponential ionic drift: fast switching and low volatility of thin-film memristors," *Applied Physics A*, vol. 94, no. 3, pp. 515–519, 2009.

[14] G. Tang, F. Zeng, C. Chen, H. Liu, S. Gao, C. Song, Y. Lin, G. Chen, and F. Pan, "Programmable complementary resistive switching behaviours of a plasma-oxidised titanium oxide nanolayer," *Nanoscale*, vol. 5, no. 1, pp. 422–428, 2013.

[15] S. Tappertzhofen, E. Linn, L. Nielen, R. Rosezin, F. Lentz, R. Bruchhaus, I. Valov, U. Böttger, and R. Waser, "Capacity based nondestructive readout for complementary resistive switches," *Nanotechnology*, vol. 22, no. 39, p. 395203, 2011.

[16] J. J. Yang, M. D. Pickett, X. Li, D. A. Ohlberg, D. R. Stewart, and R. S. Williams, "Memristive switching mechanism for metal/oxide/metal nanodevices," *Nature nanotechnology*, vol. 3, no. 7, pp. 429–433, 2008.

[17] J. J. Yang, D. B. Strukov, and D. R. Stewart, "Memristive devices for computing," *Nature nanotechnology*, vol. 8, no. 1, pp. 13–24, 2013.

[18] Y. Yang, P. Gao, S. Gaba, T. Chang, X. Pan, and W. Lu, "Observation of conducting filament growth in nanoscale resistive memories," *Nature communications*, vol. 3, p. 732, 2012.

[19] Y. Yang, P. Sheridan, and W. Lu, "Complementary resistive switching in tantalum oxide-based resistive memory devices," *Applied Physics Letters*, vol. 100, no. 20, p. 203112, 2012.

[20] P. Zhou, V. Pandey, J. Sundaresan, A. Raghuraman, Y. Zhou, and S. Kumar, "Dynamic tracking of page miss ratio curve for memory management," in *ACM SIGOPS Operating Systems Review*, vol. 38, no. 5. ACM, 2004, pp. 177–188.

[21] 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.