Design Trade-Offs for High Density Cross-Point Resistive Memory

***Abstract*—With conventional memory technologies approaching their scaling limit, the search for a new technology has gained increased attention in the recent years. Resistive RAM (ReRAM), with its superior write latency and energy, small cell size ($4F^2$ for a single level), and support for 3D stacking, has been a promising candidate among emerging memory technologies. A key advantage of ReRAM comes from its non-linear nature, which enables a cross-point array structure without having a dedicated access transistor for each cell.**

While the cross-point structure is effective in improving the memory density, it has inherent disadvantages which introduce extra design challenges. Based on the device characteristics, we perform a comprehensive analysis of issues related to reliability, energy consumption, area overhead, and performance of the cross-point arrays. In addition to the cell-level analysis, we discuss different programming schemes specifically suited for cross-point arrays. We then study the area, energy, and bandwidth of a 256 Mbits ReRAM macro in detail for various write schemes. The simulation results enable designers to identify the most performance/energy/area efficient ReRAM organization and cell parameters that meet specific design goals early in the design stage.

I. Introduction

The scaling of traditional memory technologies, such as DRAM and FLASH, is approaching its physical limit. In the past few years, emerging non-volatile memory technologies (NVM), such as Phase Change RAM (PCRAM), Spin-transfer-torque RAM (STT-RAM), and Resistive RAM (ReRAM) have been widely studied as potential candidates for the next generation memory technologies to meet the requirement of higher density, faster access time, and lower power consumption. Among all of these emerging memory technologies, ReRAM has many unique characteristics, including simple structure, nonlinearity, and high resistance ratio, making itself one of the most promising technologies. Researchers have shown that the state-of-the-art single-level-cell ReRAM can achieve $7.2ns$ random access time for both read and write operations with a resistance ratio larger than 100 [1]. Also, HP labs and Hynix have already announced plans to commercialize memristor-based ReRAM and predicted that ReRAM could eventually replace traditional memory technologies [2].

Unlike other non-volatile memory technologies, ReRAM can be implemented in a cross-point style structure without any access device [3], [4]. Specifically, in a nano cross-point array, each bistable ReRAM cell is sandwiched by two orthogonal nanowires. Thus the area occupied by each cell is $4F^2$ per bit. However, the simplicity of the access-device-free, cross-point structure introduces challenges to the peripheral circuit and memory organization design. While there have been prior studies on cross-point ReRAM arrays [5]–[9], they do not consider the effect of voltage drivers and programming methods on the array. In addition, detailed area, energy, and performance analysis is also absent. In this work, we address the design challenges of cross-point structure based ReRAM. We use a mathematical model to evaluate memory reliability, energy consumption, and area overhead for different designs and cell parameters. The advantages of nonlinearity K_r and write current I_w scaling are all discussed in detail. In addition, the simulation results of area, energy, and write throughput trade-offs are presented. Our study allows for exploring the most energy/area efficient ReRAM design with different design constraints and cell parameters at the very beginning of the design stage. Moreover, system designers can also leverage the proposed model to provide valuable feedback to device researchers who will in turn adjust ReRAM cell design. We believe that this kind of collaboration will be very helpful to shorten the time to market of ReRAM memory.

II. Preliminaries

This section provides background of ReRAM and cross-point architecture, and discusses the modeling of cross-point ReRAM array.

A. Background of ReRAM Technology

As implied by its name, a ReRAM cell uses its resistance to represent the stored information. A ReRAM cell is built on a Metal-Insulator-Metal(MIM) structure and can be switched between a high resistance state (HRS) and a low resistance state (LRS) by applying an external voltage across the cell. The resistance switching behaviors have been observed in many MIM nanodevices with different metal oxide materials. For example, a particular TiO_2 based MIM structure ReRAM, named 'memristor', was developed by HP Labs in 2008 [10]. The proposed memristor-based ReRAM is considered as the first experimental realization and a theoretical model of the fourth fundamental circuit element, which is predicted by Chua [11] about 40 years ago. It has been reported that the memristor-based ReRAM has very small cell size with an access time of less than 50ns [12]. Another H_fO_2-based bipolar ReRAM prototype was fabricated by ITRI with an access time as low as 7.2ns [1].

Although there are several variants of ReRAM cells, all of them can be classified into two broad categories: unipolar ReRAM and bipolar ReRAM. In a unipolar cell, the resistance switching behaviors do not depend on the polarity of the voltage input across the cell and are only related to magnitude and duration of the voltage input. On the other hand, in a bipolar cell, the voltage polarity for ON-to-OFF switching (RESET operation) is different from OFF-to-ON switching (SET operation). The need of different pulse widths for SET and RESET in unipolar ReRAM means that its write latency is determined by the longest pulse. Moreover, the control of SET, RESET, and read operations without any disturbance is another crucial design challenge, especially in high speed ReRAM design. For these reasons, most high performance ReRAM studies are dominated by bipolar ReRAM [1], [4], [13], [14]. In this study, we perform a detailed analysis of the design challenges of bipolar ReRAM cross-point arrays.

B. Cross-Point Architecture

There are two possible memory structures for a bipolar ReRAM array: the traditional MOSFET-accessed structure and the cross-point structure. In the former case, a dedicated MOSFET is used as an access device for each memory cell. As the size of a MOSFET access device is typically much larger than the size of a ReRAM cell, the total area of memory array is primarily dominated by MOSFETs rather than ReRAM cells. Also, in order to provide enough drive current, larger than minimum-sized transistor should be used for write operations. Hence, ReRAM's area advantage goes down significantly because of the access devices. Fortunately, we can exploit the non-linear I-V characteristic of some ReRAM devices to eliminate the access device [12], [15]. The I-V characteristic demonstrated in these fabricated devices shows that the resistance of ReRAM significantly increases as the voltage applied on it decreases. Such observation basically indicates effective cut off of the leakage current from the unselected cells in the sneak paths. Therefore, the area-efficient cross-point ReRAM memory array is enabled by the intrinsic property of the device [16]. A schematic view of a typical cross-point memory array is shown in Figure 1(a). As shown, ReRAM cells are sandwiched between wordlines and bitlines (top electrodes

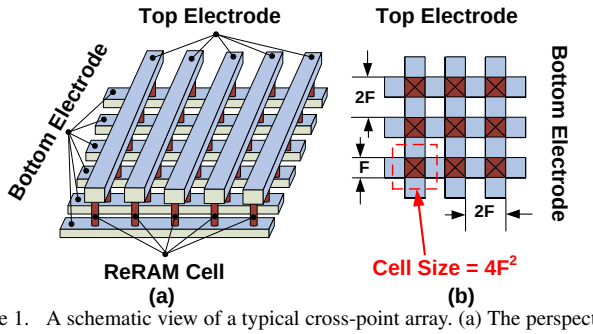

Figure 1. A schematic view of a typical cross-point array. (a) The perspective of the cross-point array. (b) The top view of the array, from which we can clearly see that the size of each cell is $4F^2$.

and bottom electrodes). Figure 1(b) shows a top view of the array, which indicates that each ReRAM cell occupies an area of $4F^2$, the theoretical lower limit for a single layer single level memory cell. In addition, this memory density can be further improved by using a multi-layer multi-level cross-point ReRAM array [3] [17].

Although avoiding access transistor is beneficial from cell area standpoint, it introduces other complexities. Following the traditional writing method in which all the bits activated by a wordline are written at once, now needs two steps to prevent unintentional writing [16]. An alternate way is to write one bit at a time but this requires interleaving data across multiple arrays to reduce write latency. Also, while writing to a cross-point array, the unselected wordlines and bitlines can be either left floating or half-biased. In contrast, while reading a cell, the selected wordline should be biased with a read voltage and all the other wordlines and bitlines in the array are shunted to ground. The current in each bitline is then sensed and compared to a reference current to determine the cell content. However, due to the sneak current existing in the cross-point array, the current in bitlines also varies depending upon the data patterns of unselected cells. This read disturbance restricts the size of a cross-point array, since sneak current increases as the number of cells attached to wordlines and bitlines increases, which makes it difficult to sense the current difference of the selected cell at HRS and LRS. Besides, the existence of the voltage drop along the nanowires also limits the length of wordlines and bitlines. Therefore, a cross-point array should be sized carefully to meet the requirements of the read/write reliability. In general, writes are more problematic than reads. The read disturbance problem can be alleviated by adopting a two-level differential sensing scheme, in which the first level reads the background noise followed by a read to data with the background current. Finally, the differential signal is amplified to get the data. In addition to all of these write/read schemes, different cell parameters will also impact the reliability, energy consumption, bandwidth, and area efficiency of the cross-point ReRAM array. In this case, it is not straightforward for a designer to figure out how to design a workable memory array with the minimum energy consumption and area overheads. Thus, the following sections will propose a worst-case oriented methodology to help designers make decisions early in the design flow.

C. Modeling of the Cross-Point Memory

The basic circuit model of an M by N cross-point ReRAM array is shown in Figure 2. This model is built upon Kirchhoff's Current Law (KCL) and its validity can be guaranteed by deductions from basic circuit theory. The horizontal lines are wordlines and the vertical lines represent bitlines. The ReRAM cells are located at each wordline and bitline cross-point. A detailed cross-point structure is also shown in Figure 2(b). The resistance of the ReRAM cell at the cross-point of i^{th} wordline and j^{th} bitline is represented by $R_{i,j}$. We assume the resistance of the wire connecting two cross-points to be R_{line}.

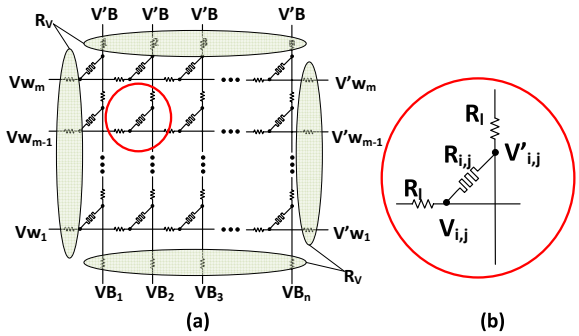

Figure 2. The circuit model of the cross-point array.

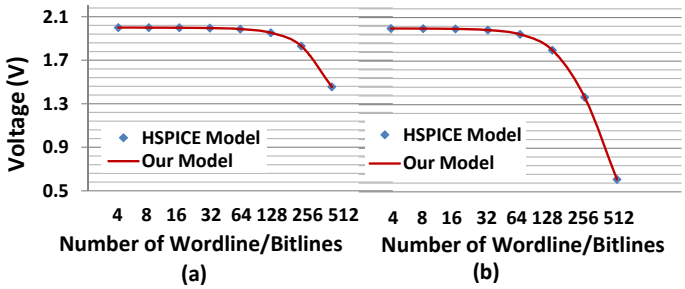

Figure 3. Validation of the analytical model against SPICE simulation. The two figures show the voltage drops obtained from our model and SPICE (a) with a nonlinearity factor of 5 and (b) without nonlinearity.

The input resistance of each wordline or bitline driver is R_v and the resistance of a sense amplifier is R_s. In order to set up the KCL equations, the voltage at each cross-point is indicated as $V_{i,j}$ for the wordline layer and $V'_{i,j}$ for the bitline layer. In addition, the input voltage for the i^{th} wordline is V_{Wi} and for the i^{th} bitline is V_{Bi}. In the case that a wordline is driven from both sides, the voltage at the other end of the i^{th} wordline is represented as V'_{Wi}.

Based on this model, the current equations for each cross-point can be obtained. All of the cross-points have similar structure with no more than three current branches, and therefore it is very easy to set up the KCL equations for each cross-point. Since the cross-points at the edges of the array have different write/read conditions, the KCL equations of these cross-point should be adjusted according to each write/read scheme.All of the KCL equations can be considered as a system of linear equations, which has the following form of $A \cdot V = C$, where A is a $2mn \times 2mn$ coefficient matrix and C is a $2mn \times 1$ vector, containing the constant terms of these equations. Thus, with parameters such as the resistance of ReRAM cells, the resistance of interconnect wires, program voltages, and write/read schemes, voltages at various cross points can be obtained by solving the system of linear equations. With detailed voltage values, $V_{2mn\times 1}$, we can analyze the array at a fine granularity. These values are also critical to evaluate the reliability, energy consumption, drive current density, and area overheads of a cross-point array.

To validate the analytical model, we compare the results with HSPICE [18] simulations using a resistor model in cross-point memory arrays. The results of eight cross-point arrays with different array sizes and specific data patterns are shown in Figure 3, which shows that the voltage drop on the selected cell derived from our analytical model are consistent with the HSPICE simulation results.

III. Analysis of Design Constraints

In this section, we study the effect of various schemes on cross-point ReRAM arrays in detail. Specifically, we evaluate the design constraints on array size, energy consumption and area overhead in worst case scenarios. The results of this study will be useful when designing a cross-point array.

TABLE I
PARAMETERS OF THE BASELINE CROSS-POINT ARRAY

Metric	Description	Typical Values (Range)
A_{cell}	Cell Size	$4F^2$
R_l	Interconnection Resistance	0.65Ω
V_{RESET}	Threshold voltage for RESET	$2.0V$
V_{SET}	Threshold voltage for SET	$-2.0V$
V_{READ}	Read Voltage of Cell	$0.5V$
I_{on}	Write Current for LRS Cell	$40uA$ $(40 \sim 200uA)$
$V_W(R)$	Wordline Voltage during Read	$0.5V$
$V_W(W)$	Wordline Voltage during Write	$\pm 2V$
$V_W(H)$	Half Selected wordline Voltage	$1V$
$V_B(R)$	Bitline Voltage during Read	$0V$
$V_B(W)$	Bitline Voltage during Write	$0V$
$V_B(H)$	Half Selected bitline Voltage	$1V$
K_r	Nonlinearity of ReRAM Cell	20 $(2 \sim 40)$
M, N	Number of wordlines/bitlines	512 $(8 \sim 1024)$

A. Overview

In order to write or read a cross-point array, proper voltages should be applied across the ReRAM cell. Although the goal of a read operation is different from a write operation, both of them are realized by fully biasing the selected wordlines/bitlines and floating (or half biasing) unselected wordlines/bitlines. Thus, the coefficient matrix A and the constant vector C are very similar for both. In addition, their energy consumption and area overhead will also have a similar trend. Therefore, in this section, we first study the write operation comprehensively. After that, for read operation, we mainly focus on the read margin analysis since it is unique for read operations.

Table I shows the circuit parameters of our baseline 50nm design. The data is derived from the recently published studies on ReRAM [16], [19], [20]. The nonlinearity coefficient is defined as

$$K_r(p, V) = p \times R(V/p)/R(V), \tag{1}$$

where $R(V/p)$ and $R(V)$ are the equivalent resistance of the cell biased at V/p and V [16]. Therefore, the resistance of a ReRAM cell with nonlinearity is not constant but varies with the applied voltage. For example, for a ReRAM cell with nonlinearity of 20, the resistance of half biased cell is 10 times larger than resistance of fully biased cell. By using these parameters, we study reliability, energy consumption, and area overheads for four different write schemes, and discuss the sensitivities of these schemes to the data pattern of HRS and LRS ReRAM cells and cell nonlinearity. In this section, the baseline design uses a cell with write current of $40uA$ and nonlinearity $Kr = 20$. A sensitivity study varying the nonlinearity coefficient and the write current is presented in Section IV.

B. Write Operation

To write a ReRAM cell, an external voltage is applied across the cell for a certain duration. Intuitively, there are four possible schemes for the write operation: *FWFB* scheme activates the selected wordline and selected bitline, and leaves all of other lines floating; *FWHB* scheme activates the selected wordline and bitline, leaves all the unselected wordlines floating, and half biases the unselected bitlines; *HWFB* scheme activates the selected wordline and bitline, leaves all the unselected bitlines floating, and half biases the unselected wordlines; *HWHB* scheme activates the selected wordline and bitline, and half biases the unselected wordlines and bitlines. However, the FWFB scheme has an inherent problem that may result in severe write disturbance [8]. Therefore, only the FWHB, HWFB and HWHB schemes are workable for programming a cross-point array. We analyzed these three schemes and found that they all have the same worst case voltage drop. Besides, we found that the HWHB scheme is the most energy/area efficient among these schemes. Therefore, in the following discussion, we only show the simulation results of HWHB schemes. The results for the other two schemes have the similar trend as that of HWHB scheme.

Besides, during the write operation, we can write only one bit per access (single-bit write), write several bits on one wordline at the same time (multi-bit write), or even write all of the cell on one wordline (whole-wordline write). We found that, at the array level, the energy consumption and area overhead increase monotonically with the increase of the number of bits per access. Therefore, in this section, we provide simulation results of two extreme instances: single-bit write and whole-wordline write operation. Detailed analysis of multi-bit write operation is discussed in Section V.

Reliable Write Operations. Write reliability is a serious concern in cross-point arrays. In an ideal condition, the resistance of wires and the sneak currents in unselected cells are negligible. In such a scenario, all the write schemes discussed above can make sure that the write voltage $V_W(W) - V_B(W)$ is fully applied across the specified cell. However, in reality, both wire resistance and sneak current are non-trivial. Hence, the voltage applied across a cross-point varies based on the location of the cell as well as the data pattern stored in all of the ReRAM cells in the array. A write is considered reliable if it modifies the content of the selected cells to the new value without disturbing other unselected cells. Correspondingly, there are two potential problems with writes: *write failure*, an unsuccessful write on selected cells, and *write disturbance*, an undesirable write to unselected cells. It is necessary to ensure that a write scheme guarantees reliable operation even in the worst case (w.r.t the location of cells to written and the data pattern stored in the cross-point array).

Write failure typically results from the voltage drop at the interconnect wires along the wordline and bitline. It has been shown that, for single-bit write operation, the worst case voltage drop occurs when writing the cell at the cross point of the M^{th} wordline and the N^{th} bitline with all of the other cells in the array are in LRS [7]. In order to avoid write failure and successfully program the selected ReRAM cell, the drive voltage should be boosted to a higher level, making sure that the voltage across the cell exceeds the threshold voltage even at the worst case. Figure 4 shows the lower bounds of the drive voltage for different sizes of cross-point array. The minimum wordline/bitline voltage increases from 2.01 V for a 32×32 array to nearly 7 V for a 1024×1024 cross-point array. However, boosting the drive voltage also increases the voltage applied at unselected cells. Therefore, a write disturbance may occur when the voltage applied at an unselected cell exceeds the threshold voltage for SET or RESET operation. According to our analysis, the maximum voltage applied at unselect cells is exactly the same as half of the drive voltage. Thus, only arrays with drive voltage less than 4V are allowable. Otherwise, the array is unreliable because it cannot avoid write failure and write disturbance at the same time. The unreliable array sizes are denoted as red bars in Figure 4. The array size limitation provided by Figure 4 is a hard constraint, and all of the following energy and area trade-offs are bounded by this constraint.

Additionally, the cross-point array can be organized with a different number of wordlines and bitlines. For example, a 256K bit cross-point array can be implemented either by a 512×512 array or by a 64×4096 array. In the latter case, the voltage drops along the wordline will be much worse than along the bitline. Our analysis shows that from a reliability point of view, a cross-point array with the same number of wordlines and bitlines is the best choice. Thus, in the following discuss, we assume the array has the same number of wordline and bitline.

Energy Consumption of Write Operations. The energy consumption of a write operation includes: the energy consumed to change the state of the selected cell, the undesired energy wasted at the half selected cells and unselected cells, and the energy consumed by the interconnect lines. Figure 5(a) shows the decomposed energy

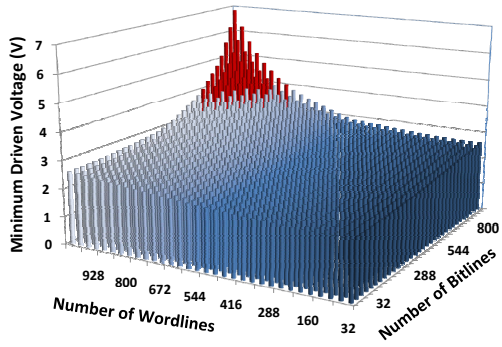

Figure 4. Required write voltages for different cross-point arrays (threshold voltage = 2V.).

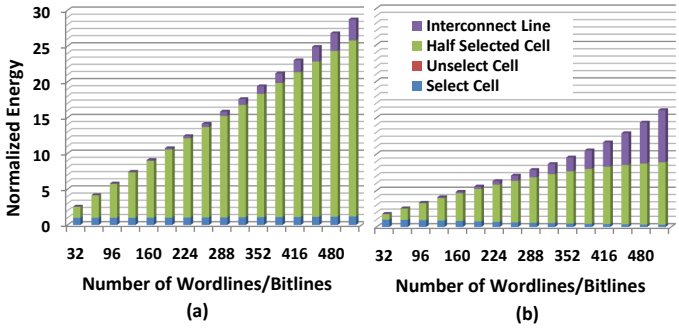

Figure 5. The normalized energy consumption with different array size. (a) Single-bit writing. (b) Whole-wordline writing.

consumption for single-bit write operation. Obviously, the undesired energy consumed by half-selected cells takes a great part of the total energy consumption. Besides, with the increase of array size, the energy dissipated at interconnect lines also becomes significant. Also, this part of the energy wasted during the write operation is a greater part of the total energy for larger array sizes. For example, the undesired energy consumption for writing a 512×512 array is more than 15 times larger than that of a 32×32 array. For whole-wordline write operation, we evaluate the energy consumption of write operations that program the entire wordline at one time. In order to fairly compare the energy consumption, we compare the energy-per-bit instead of the total energy. For example, in order to write a wordline with size of 512 bits, the energy-per-bit can be calculated as: $E_{ave} = E_{total}/512$. Figure 5(b) shows the energy-per-bit of the whole-wordline write operation. Compared with the single-bit write operation, we conclude that for large cross-point array sizes, the whole-wordline write operation is much more energy efficient. This is because the energy wasted at the unselected and half-selected cells are amortized by multiple bits and the average energy for one bit is therefore reduced.

Write Current and Area Overhead of Write Operations. The write operation for a $M \times N$ array requires M wordline voltage drivers and N bitline multiplexors. The drivers and multiplexors should be sized such that they can provide the worst-case current of wordline current and bitline current. The transistor sizing of the wordline/bitline circuitry is achieved using HSPICE simulations. We further calculate the area overhead for the drivers and multiplexors by referring to the CACTI area model. Figure 6(a) shows the maximum write current with different ReRAM array sizes. Not surprisingly, the current requirement increases as the array size increases. Figure 7(a) illustrates the area overhead for the wordline and bitline circuitry. This show that drivers and multiplexors occupy a smaller area than the cross-point array. Only in this case can voltage drivers and multiplexors be implemented beneath the array, resulting an ideal cell size of $4F^2$.

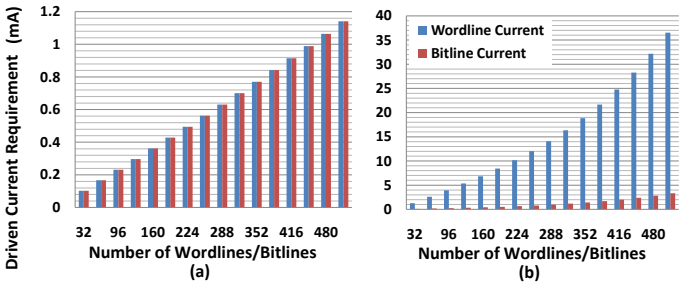

Figure 6. The requirements for wordline and bitline drive currents. (a) One bit per write. (b) One wordline per write.

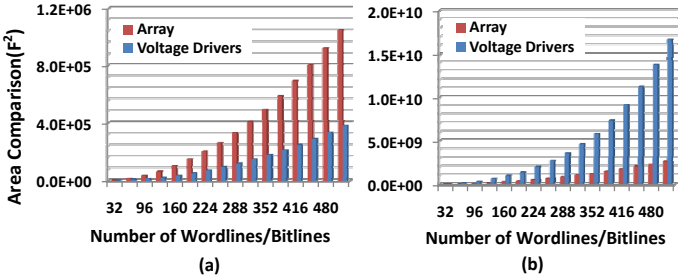

Figure 7. Area overhead comparison. (a) Single-bit write. (b) Whole-wordline write.

Although whole-wordline write operation has the advantage of lower energy consumption, the maximum current requirement for each wordline also increases. As demonstrated in Figure 6(b), although the maximum drive current for each bitline is almost the same as when writing one bit, the driving current requirement for each wordline in a whole-wordline write scheme is > 10 times larger than that of a single-bit write scheme. Since the area of the voltage driver increases proportionally with its driving current, the area overhead for whole-wordline writing is much larger than that of single-bit writing. As shown in Figure 7(b), the peripheral circuitry area is much larger than that of the array. In this case, the total area of the memory array is dominated by the peripheral circuitry rather than the cells. In addition to the extra area overhead, writing multiple bits at one time also worsens the voltage drop along the wordline. Our simulation results show that, in order to program an entire wordline when writing, the maximum reliable array size reduces from 800×800 to 352×352. This is because the current passing through the interconnect wires in the whole-wordline write scheme is much larger than that of the single-bit write scheme, causing more severe voltage drops on the wire resistance.

Therefore, we conclude that although the whole-wordline write operation is more energy efficient, from the standpoint of reliability and area overhead, single-bit write operation is preferred.

Read Operation. In this section we apply a similar sensing scheme as [6] and [7]. In order to read cell $R_{i,j}$, the i^{th} wordline is biased at V_{READ} and all of the other wordlines and bitlines are grounded. Then the state of the selected cell is read out by measuring the voltage across R_s. The energy consumption for a read operation can be analyzed similarly as a write operation. Since the read voltage is much smaller than write voltage, the read energy is expected to be at least one order of magnitude smaller than for a write operation. Considerable sensing margin is achieved by implementing a current-to-voltage converter and sensing the voltage signal using traditional or more recent sense amplifier designs. The input resistance of the current-to-voltage converter is extracted from HSPICE simulation results. Read sensing margin is defined as $\Delta V = \Delta I \times R_{converter}$ where $R_{converter}$ is the input resistance of the converter. The read reliability is determined by the voltage swing for reading HRS and LRS cells. Detailed results will be shown in Section IV.

IV. Nonlinearity and Write Current Scaling

One of the most distinct features of ReRAM is its nonlinearity. Normally, the K_r value for memristor-based ReRAM is larger than 20, meaning that the resistance of a half-biased cell is at least 10 times larger than a full-biased cell. Clearly, ReRAM cells with larger nonlinearity coefficients result in a better memory cell since the sneak current in half selected cells will be significantly reduced. In addition, the increased resistance at half-selected and unselected cells can also mitigate the voltage drop along the activated wordline and bitline. Also, we find that the cross-point array design can benefit from the scaling of the write current. Figure 8 shows the influence of different nonlinearity coefficients and write currents on the array size requirements for a single-bit HWHB writing scheme. This figure shows that the array size limitation is relaxed as the nonlinearity increases or the write current scales. As we can see from the figure, the maximum array size exceeds 1024×1024 when we have a nonlinearity of 30, together with a write current of $40\mu A$.

Moreover, the increase of nonlinearity or scaling of write current can also reduce the energy consumption and area overhead of the cross-point array. As shown in Figure 9(a), for a 512×512 array, the energy consumption for the write operation decreases dramatically with the scaling of nonlinearity coefficient K_r. For example, for a ReRAM cell with write current of 50uA, the write energy is reduced by 98.3% when K_r increases from 2 to 40. The area overhead of the voltage drivers is illustrated in Figure 9(b). As a baseline design ($K_r = 20$ and $I_w = 40\mu A$), the driver area overhead is about 35% of the area of the memory array cells. To design a memory array with an effective cell size close to $4F^2$, we need to make sure the nonlinearity and write current should satisfy certain conditions so that the driver overhead is less than 100% and the wordline drivers can be almost "hidden" underneath the ReRAM cells. As nonlinearity and write current continues to scale, the area overhead can be as low as 10%. In that case, the introduction of 3D stacking of multi-layer cross-point arrays is productive in further reducing the effective cell size to $4/N_l F^2$ where N_l is the number of layers.

Unlike the write operation, the read operation suffers, rather than benefits, from scaling of nonlinearity or write current. This is because the scaling of nonlinearity and write current will reduce read current, degrading the read signal ratio. Figure 10(a) shows the read noise margin with different array sizes for the baseline design in Section III. As can be seen, the read noise margin is reduced for large array sizes. The impact of nonlinearity and write current on read noise margin is illustrated in Figure 10(b). A large K_r value and small write current are harmful to the read noise margin. For example, given a 512×512 array, the read noise margin is less than $10mV$ for $K_r = 40$ and $I_w = 40\mu A$, which makes it very difficult to sense the state of the selected memory cell using traditional sense amplifiers.

Therefore, by knowing the array size and read noise margin constraints, an "optimal cell" with nonlinearity of K_{r_opt} and write current of I_{on_opt} can be determined. For example, when the array size is fixed at 512×512 and the minimum noise margin is $50mV$, a cross-point array with ReRAM cells which have $K_{r\_opt} = 9$ and $I_{on\_opt} = 40mA$ is the most energy and area efficient design.

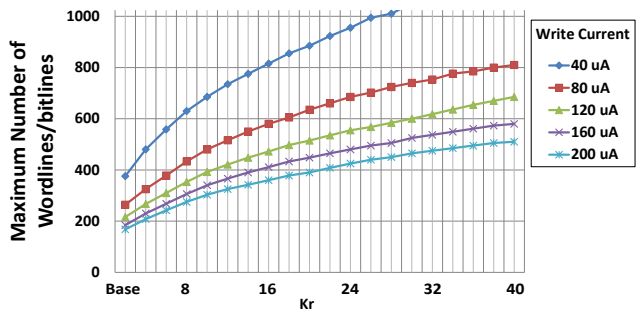

Figure 8. The maximum array size with different nonlinearity coefficients.

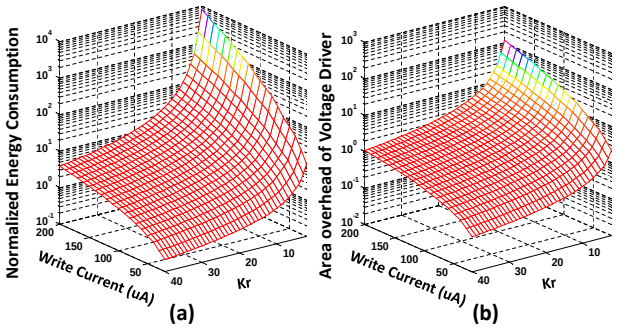

Figure 9. Energy and area overhead comparison. (a) Energy consumption (normalized to baseline). (b) Area overhead of voltage driver (normalized to the area of cross-point array).

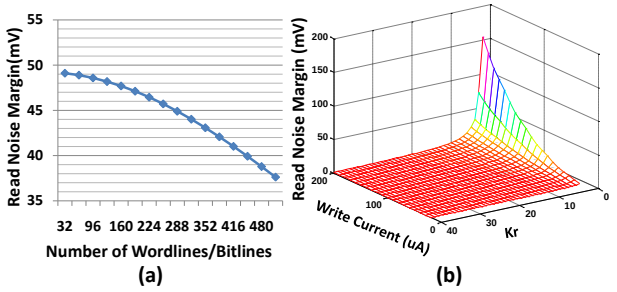

Figure 10. Read noise margin with (a) different array size and (b) scaling of nonlinearity and write current.

V. A Case Study of Cross-point ReRAM Macro Design

Since the array size of a cross-point ReRAM array is limited by the reliability requirements, the design of a ReRAM macro is different from the traditional DRAM design. In this section, we evaluate the area, energy consumption, and bandwidth of a 256 Mbits ReRAM macro. We use an organization similar to Kawahara's design [4], where a 256 Mbits ReRAM macro is divided into eight planes. Each 32 Mbit plane has separate wordline decoder, bitline selectors, sense amplifiers, and write circuity. Due to space constraints, we present results for only four typical cell parameters: ($Kr = 20, I_w = 40uA$), ($Kr = 20, I_w = 200uA$), ($Kr = 40, I_w = 40uA$), and ($Kr = 40, I_w = 200uA$). For each of them, we vary the number of bit per write to investigate the relation among the area, energy consumption, and bandwidth of the ReRAM macro.

Table II shows the total area, energy consumption, and bandwidth of the 256 Mbits ReRAM macro. Consistent with our earlier discussion, as the device nonlinearity improves, both area and energy goes down. The only downside is the noise margin restriction imposed for reads. Similarly, as the drive current increases, the overhead goes up due to large wordline drivers and bitline multiplexors. Hence, bandwidth improvement comes at the cost of area and energy.

To better understand the ideal design choice for a given device parameter, we investigated three metrics: bandwidth per nanojoule (BW/nJ), bandwidth per square millimeter(BW/mm^2), and bandwidth per nanojoule per square millimeter ($BW/(nJ \cdot mm^2)$). Figure **??** (a) shows how BW/mm^2 scale as we increase the number of bits modified per write operation. From the figure, for a given energy budget, writing one bit at a time provides at least 48% better bandwidth compared to the best performing muti-bit writes. Hence, with the right choice of global interconnect, interleaving writes across multiple sub-arrays is an interesting design point. With multi-bit writes, as the number of bits per write increases, the energy efficiency

TABLE II
AREA, ENERGY, AND BANDWIDTH RESULTS OF 256 MBITS RERAM MACRO

Kr	$I_w(uA)$		Number of bit per write at array level							
			1	2	4	8	16	32	64	128
20	40	Area(mm^2)	3.888	3.944	4.056	4.288	4.752	5.688	7.544	11.752
		Energy(nJ)	4.375	12.379	19.860	33.660	59.430	111.404	232.437	576.496
		Bandwidth(MBit/s)	66.687	72.618	144.747	287.534	567.290	1103.991	2089.847	3649.583
20	200	Area(mm^2)	6.512	6.816	7.424	8.640	11.096	16.144	27.288	N/A
		Energy(nJ)	24.584	67.126	106.848	182.409	338.781	716.141	1845.355	N/A
		Bandwidth(MBit/s)	90.038	113.028	217.351	401.199	685.218	1018.788	1213.958	N/A
40	40	Area(mm^2)	3.616	3.672	3.776	3.992	4.752	5.688	7.544	11.432
		Energy(nJ)	2.065	5.550	8.711	14.485	25.601	49.322	107.519	280.121
		Bandwidth(MBit/s)	69.594	74.309	148.111	294.199	580.357	1129.126	2136.164	3777.482
40	200	Area(mm^2)	3.984	4.288	4.888	6.088	8.464	13.328	24.088	N/A
		Energy(nJ)	11.645	29.739	46.939	80.813	155.025	343.658	933.231	N/A
		Bandwidth(MBit/s)	115.781	131.827	253.686	469.243	800.280	1174.237	1362.144	N/A

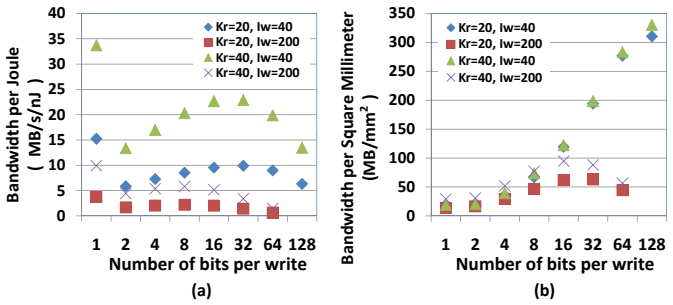

Figure 11. (a) Bandwidth per Joule and (b) bandwidth per square millimeter of 256 Mbits ReRAM macro.

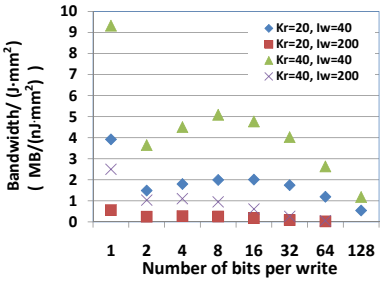

Figure 12. Bandwidth per Joule per square millimeter of 256 Mbits ReRAM macro.

also increases. However, as the word size increases, the voltage drop in the array also increases, which needs to be compensated by increasing the operating voltage of the array (Section III). Beyond 32 bits, this increase in voltage, outweighing the bandwidth improvement, effectively reducing the energy efficiency. Thus from energy standpoint, multi-bit write is optimal when the word size is 8-32 bits, depending upon the nonlinearity and drive current. Figure **??** (b) shows the effect of multi-bit writes on bandwidth per square millimeter. Unlike energy, as long as the drive current is less, it is beneficial to increase the word size as much as possible to improve bandwidth for a given area. Also, writing one bit at a time is the least attractive option for a design primarily constrained by the area. Figure V takes into account both energy and area, and provides a "sweet spot" for multi-bit writes. Thus by understanding the key characteristics of cross-point array, we can identify an optimal configuration that best meets the design constrains.

VI. CONCLUSION

In this paper, we use a mathematical model to study in detail how reliability affects the array organization, size, energy consumption, and area overheads of cross-point arrays. The size of a cross-point is limited by the peripheral circuit overhead and sneak current. Our simulation results show that with best possible device non-linearity and drive current, the maximum array size cannot exceed 1024x1024 without compromising reliability. We also showed that multi-bit writes is more energy efficient than single-bit write, however, the latter significantly reduces the complexity of peripheral circuits and provides better area efficiency. Both high nonlinearity and low write current are key to reduce energy and area of cross-point arrays. Finally, since memory bandwidth is an important design constraint, we studied various designs that maximizes bandwidth for a given area and energy budget. Through our case study, we show that there is an optimal word size for a given device parameter that has the best energy, area, and bandwidth properties.

REFERENCES

[1] S. S. Sheu *et al.*, "A 4Mb embedded SLC resistive-ram macro with 7.2ns read-write random-access time and 160ns MLC-access capability," in *Proc. of ISSCC*, 2011.

[2] "http://www.hpl.hp.com/news/2010/jul-sep/memristorhynix.html."

[3] C. Chevallier *et al.*, "A 0.13 um 64Mb multi-layered conductive metal-oxide memory," in *Proc. of ISSCC*, 2010.

[4] A. Kawahara *et al.*, "An 8Mb Multi-Layered Cross-Point ReRAM Macro with 443MB/s Write Throughput," in *Proc. of ISSCC*, 2012.

[5] M. Ziegler and M. Stan, "Design and analysis of crossbar circuits for molecular nanoelectronics," in *Proc. of IEEE Conf. on Nano.*, 2002.

[6] A. Flocke *et al.*, "A fundamental analysis of nano-crossbars with non-linear switching materials and its impact on TiO2 as a resistive layer," in *Proc. of IEEE Conf. on Nano.*, 2008.

[7] J. Liang and H.-S. Wong, "Cross-point memory array without cell selectors -device characteristics and data storage pattern dependencies," *IEEE Transactions on Electron Devices*, 2010.

[8] M. Ziegler and M. Stan, "CMOS/nano co-design for crossbar-based molecular electronic systems," *IEEE Trans. on Nanotechnology*, 2003.

[9] O. Kavehei *et al.*, "An analytical approach for memristive nanoarchitectures," *IEEE Transactions on Nanotechnology*, Mar 2012.

[10] D. B. Strukov *et al.*, "The missing memristor found," *Nature*, 2008.

[11] L. Chua, "Memristor-the missing circuit element," *IEEE Transactions on Circuit Theory*, Sep 1971.

[12] J. J. Yang *et al.*, "Memristive switching mechanism for metal/oxide/metal nanodevices," in *Nature Nanotechnology*, 2008.

[13] M. Kim *et al.*, "Low power operating bipolar TMO ReRAM for sub 10 nm era," in *Proc. of IEDM*, 2010.

[14] W. Otsuka *et al.*, "A 4Mb conductive-bridge resistive memory with 2.3GB/s read-throughput and 216MB/s program-throughput," in *Proc. of ISSCC, 2011*.

[15] R. Meyer *et al.*, "Oxide dual-layer memory element for scalable non-volatile cross-point memory technology," in *Proc. of NVMTS*, 2008.

[16] C. Xu *et al.*, "Design implications of memristor-based RRAM cross-point structures," in *Proc. of DATE*, 2011.

[17] M.-J. Lee *et al.*, "Stack friendly all-oxide 3D RRAM using GaInZnO peripheral TFT realized over glass substrates," in *Prof. of IEDM*, 2008.

[18] "http://www.synopsys.com/tools/verification/amsverification/circuitsimulation/hspice/pages/default.aspx."

[19] H. Akinaga and H. Shima, "Resistive random access memory (ReRAM) based on metal oxides," *Proceedings of the IEEE*, Dec 2010.

[20] M. Terai, Y. Sakotsubo, Y. Saito, S. Kotsuji, and H. Hada, "Memory-state dependence of random telegraph noise of Ta2O5/TiO2 stack ReRAM," *IEEE Electron Device Letters*, Nov 2010.