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

***Abstract*—With conventional memory technologies approaching their scaling limit, emerging non-volatile memory technologies have attracted increasingattention because of their non-volatility, high access speed, low power consumption, and good scalability. Resistive RAM (ReRAM), with its simple structure, small cell size ($4F^2$), and the 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 cross-point RAM array structures without having a dedicated access transistor for each cell. While cross-point design is effective in improving the memory density, it has inherent disadvantages which introduce extra design challenges. Based on the device characteristics, we propose a mathematical model to perform a comprehensive analysis of issues of reliability, energy consumption, and area overhead for the cross-point array structure. In addition to the cell-level analysis, different programming schemes are also discussed in this paper. The proposed model enables designers to identify the most energy/area efficient ReRAM organization and cell parameters that meet specific design goals during the early 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, non-linearity, 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 [**?**]. Also, HP labs and Hynix have already announced plans to commercialize memristor-based ReRAM and predicted that ReRAM could eventually replace traditional memory technologies [**?**].

Unlike other non-volatile memory technologies, ReRAM can be implemented in a cross-point style structure without any access device. 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 literally the area underneath the intersection of wires, which is $4F^2$ per bit. However, the simplicity of 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 [**?**], [**?**], [**?**], [**?**], they do not consider the effect of voltage drivers and programming methods on the array. In addition, detailed area and energy analysis is also absent. In this work, we address the design challenges of cross-point structure based ReRAM. We build an accurate 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. 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.

The rest of this paper is organized as follows. In Section II, an overview of ReRAM technology and cross-point architectures is given. Section A discusses the proposed mathematical model for the cross-point structure ReRAM and the edge conditions for different write and read schemes. Section IV analyzes different design constraints of write and read operations on cross-point based ReRAM arrays. The energy consumption and area overheads are also analyzed in this section. Then in Section V, the effect of nonlinearity and write current on the design constraints is evaluated. Finally, the conclusion is presented in Section VI.

II. Preliminaries

This section provides background of ReRAM and cross-point architecture, and discusses their advantages and limitations.

A. Background of ReRAM Technology

As implied by its name, a ReRAM cell uses its resistance to represent the stored information. A ReRAM cell can be switched between a high resistance state (HRS) and a low resistance state (LRS) by applying an external voltage across the cell. In general, a ReRAM cell is built on a Metal-Insulator-Metal (MIM) structure. 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 [**?**]. 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 [**?**] 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. Another H_fO_2-based bipolar ReRAM prototype was fabricated by ITRI this year with an access time as low as 7.2ns [**?**].

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 for 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 [**?**], [**?**], [**?**]. 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 implementation: a traditional MOSFET-accessed structure and a cross-point structure. In the MOSFET-accessed memory array, a 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 driven current, larger than minimum-sized

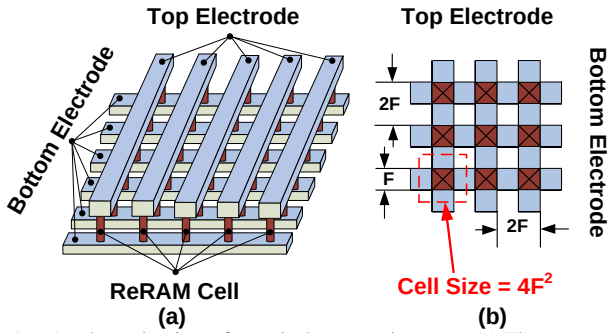

Fig. 1. A schematic view of a typical cross-point array. (a) The perspective of the cross-point array. (b) The top view of the array, from which we can clearly see that the size of each cell is $4F^2$.

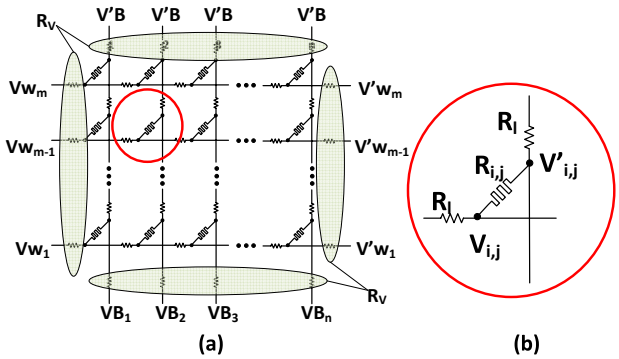

Fig. 2. The circuit model of the cross-point array.

transistor should be used for write operations. Hence, ReRAM's area advantage gets lost because of the access devices. Fortunately, the access device can be eliminated due to the large current-voltage (I-V) nonlinearity of some ReRAM devices [**?**], [**?**]. The I-V characteristic demonstrated in these fabricated devices shows that the resistance of ReRAM significantly increases as the voltage applied on it increases. 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. 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 and bottom electrodes). Figure 1(b) shows a top view of the array, which shows that each ReRAM cell occupies an area of $4F^2$, the theoretical lower limit for a single layer single level memory cell. This memory density can be further improved by using a multi-layer multi-level cross-point ReRAM array [**?**] [**?**].

There are several write/read schemes for cross-point ReRAM arrays. For example, the write operation can write either a single-bit per access or several bits attached to the same wordline at the same time. Although the second scheme has higher bandwidth, it requires a two-step write operation to prevent unintentional writing [**?**], which significantly increases the write latency. Furthermore, 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. Therefore, a cross-point array should be sized such that the current difference of the selected cell at HRS and LRS is large enough for reliable sensing. In addition to all of these write/read schemes, different cell parameters will also impact the reliability, energy consumption, 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.

III. Modeling of the Cross-Point Memory

In this section, we present a brief introduction of mathematical model of the cross-point array. More details can be found in the Appendix.

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}. 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.

However, it is important to note that KCL equations for cross-points at the edges of the array vary with different write/read schemes. For example, the unselected wordline for write operation can be either half biased or left floating. Thus, the edge conditions should be adjusted according to each write/read scheme. In particular, according to their locations and write/read schemes, all of the cross-points in an array can be classified into three major categories: *normal point*, *activated point* and *floating point*. The normal points are located inside the memory array. The activated point and floating point represent the nodes at the edge of cross-point array with different conditions: an edge point, which is directly connected to the voltage input or to the ground, can be considered as an activated point. Otherwise, it is a floating point. The detailed models for these points can be found in the Appendix. All of the KCL equations can be considered as a system of linear equations, which has the following form

$$A \cdot V = C, \tag{1}$$

where A is a $2mn \times 2mn$ coefficient matrix and C is a $2mn \times 1$ vector, containing the constant terms of these equations. We demonstrate that all of the KCL equations have simple and similar structures. Therefore, the linear equation system has a relatively fixed format and simple structure, making it easy to establish and adjust the coefficients and constants according to different design schemes. Besides, due to the simplicity of the KCL equation, A is populated primarily with zeros and can be saved as a sparse matrix, which will further reduce the storage cost during the computation.

To validate our analytical model, we compare the results with

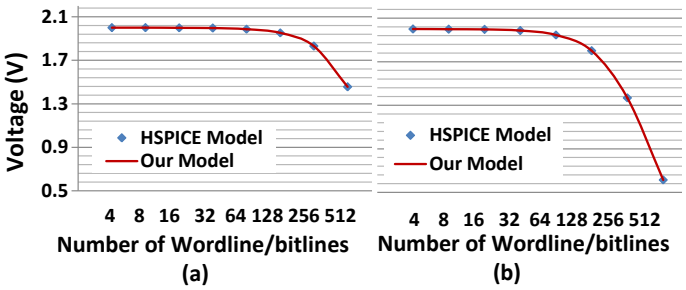

Fig. 3. Validation of the analytical model against SPICE simulation. The two figures show the voltage drops obtained from our model and SPICE (a) with a nonlinearity factor of 5 and (b) without nonlinearity.

HSPICE simulations using a resistor model in cross-point memory arrays. DC analysis was performed by HSPICE which solved the voltage of every node in the array. 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. 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, $V_{2mn\times1}$, we can analyze the array at a fine granularity. These values are also critical to evaluate the reliability, energy consumption, driven current density, and area overheads of a cross-point array.

IV. Analysis of Design Constraints - A Case Study

In this section, we study the effect of various schemes on cross-point ReRAM arrays in detail by using our model. 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.

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 [**?**], [**?**], [**?**]. The nonlinearity coefficient is defined as

$$K_r(p,V) = p \times R(V/p)/R(V), \tag{2}$$

where $R(V/p)$ and $R(V)$ are the equivalent resistance of the cell biased at V/p and V [**?**]. Therefore, the resistance of a ReRAM cell with nonlinearity is not constant but varies with the applied voltage. 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 V.

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:

1) According to the location of a selected cell, activate one wordline and one bitline and leave all of other lines floating (FWFB scheme).
2) Activate the selected wordline and bitline. Leave all the unselected wordlines floating and half bias the unselected bitlines (FWHB scheme).
3) In contrast to the scheme 2), activate the selected wordline and bitline. Leave all the unselected bitlines floating and half bias the unselected wordlines (HWFB scheme).
4) Activate the selected wordline and bitline. Then half bias the unselected wordlines and bitlines (HWHB scheme).

Unfortunately, the FWFB scheme has an inherent problem that may result in severe write disturbance [**?**]. Therefore, in the following discussion, we only compare the results of FWHB, HWFB and HWHB schemes. For each of these three schemes, we can either write several cells on one wordline at the same time or write only one bit per access and distribute the write operation to several arrays. In the following discussion, we start from single-bit per access write operation, and then the results of multi-bit per access method are discussed.

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 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 cell, and *write disturbance*, an undesirable write to an unselected cell. 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 [**?**]. In order to avoid write failure and successfully program the selected ReRAM cell, the driven voltage should be boosted to a higher level, making sure that the voltage across the cell exceeds the threshold voltage

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	$40uA$ $(20 \sim 200uA)$
$V_W(R)$	Wordline Voltage during Read	$0.5V$
$V_W(W)$	Wordline Voltage during Write	$\pm2V$
$V_W(H)$	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 $(8 \sim 1024)$

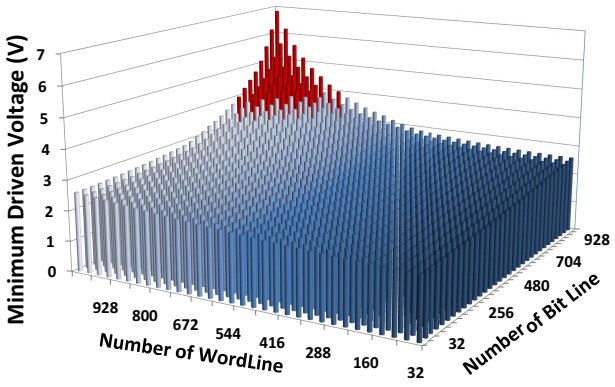

Fig. 4. Required write voltages for different cross-point arrays (threshold voltage = 2V.).

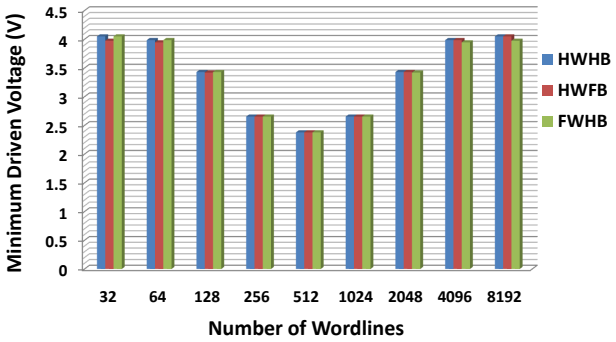

Fig. 5. Required write voltages with different memory shapes (array capacity = 256Kbits, threshold voltage = 2V).

even at the worst case. Figure 4 shows the lower bounds of the driven 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. Additionally, the cross-point array can be organized with different number of wordlines and bitlines. For example, a 256K bits 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. Figure 5 examines the voltage requirements for different array organizations with different write schemes. This result shows that from a reliability point of view, a cross-point array with the same number of wordlines and bitlines is the best choice. Furthermore, we also notice that when the array has the same number of wordlines and bitlines, FWFB, HWFB and FWHB schemes have the same minimum driven voltage.

However, boosting the driven voltage also introduces other potential problems for the array design. In particular, increasing the driven voltage will increase 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. Specifically, the maximum voltage applied at unselect cells is exactly the same as half of the driven voltage. Thus, only arrays with driven 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 tradeoffs are bounded by this constraint.

Energy Consumption of Write Operations. The energy consumption of a write operation includes: the energy consumed to change the state of the selected cell (denoted as E_{select}), the undesired energy

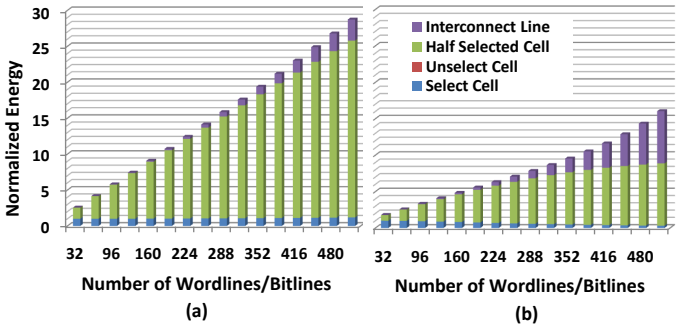

Fig. 6. The normalized energy consumption with different array size. (a) Single-bit writing. (b) Multi-bit writing.

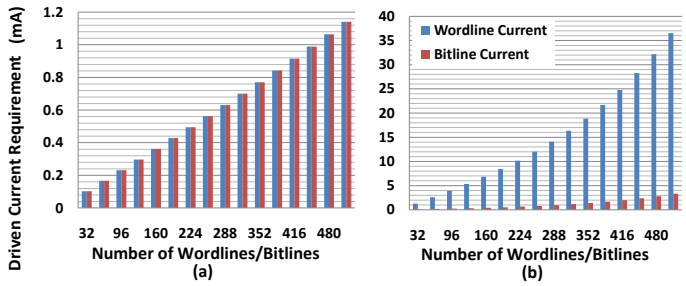

Fig. 7. The requirements for wordline and bitline driven currents. (a) One bit per write. (b) One wordline per write.

wasted at the half selected cells ($E_{halfselect}$) and unselected cells ($E_{unselect}$), as well as the energy consumed by the interconnect lines (E_{line}). Notice that since the impact of sneak paths for floating schemes (FWHB and HWFB) is more serious, the energy consumed at unselected cells for floating schemes is larger than the half-biased scheme. However, we find that compared to the total energy consumption, the energy consumed by unselected cells are negligible. Therefore, the total energy consumptions for FWHB and HWFB schemes are almost the same as that of HWHB scheme. In the following discussion, we focus on the HWHB scheme. Figure 6(a) shows the decomposed energy consumption for the cross-point array. Obviously, for a cross-point ReRAM array, the undesired E_{line} and $E_{halfselect}$ take a large amount of the total energy consumption. 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.

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 multiplexes by referring to the CACTI area model. Figure 7(a) shows the maximum write current with different ReRAM array sizes. Not surprisingly, the current requirement increases as the array size increases. Figure 8(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 $4F^2$.

Discussion on Multi-Bits Write Operation. So far, we have only discussed the write operation with one bit per access. In this section,

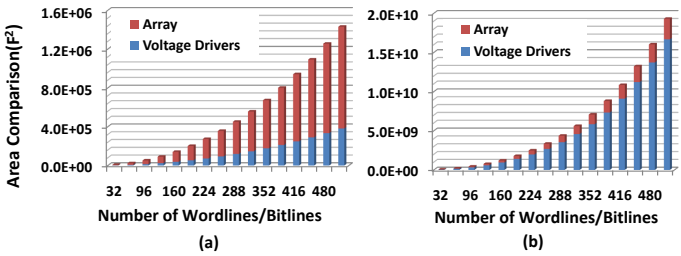

Fig. 8. Area overhead comparison. (a) One bit per write (B) One wordline per write.

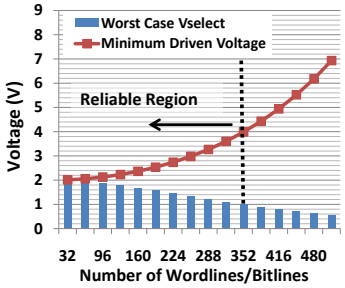

Fig. 9. Worst case select voltage and write voltage requirements for multi-bit writing (one wordline per write).

we compare the difference between single-bit per access and multi-bit per access write operations.

First of all, 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 6(b) shows the energy-per-bit of the multi-bit write operation. Compared with the single-bit write operation as shown in Figure 6(a), we conclude that for large cross point array sizes, the multi-bit 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. However, although multi-bit write operation has the advantage of lower energy consumption, the maximum current requirement for each wordline also increases. As demonstrated in Figure 7(b), although the maximum driven current for each bitline is almost the same as when writing one bit, the driving current requirement for each wordline in a multi-bit 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 multi-bit writing is much larger than that of single-bit writing. As shown in Figure 8(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. As shown in Figure 9, in order to write 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 multi-bit 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 multi-bit 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 [**?**] and [**?**]. 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 V.

V. Nonlinearity and Write Current Scaling

One of the most distinct features of ReRAM is its nonlinearity. Normally, the $K_r(p, V)$ 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 10 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 11(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 1 to 40. The area overhead of the voltage drivers is illustrated in Figure 11(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 reduce the effective cell

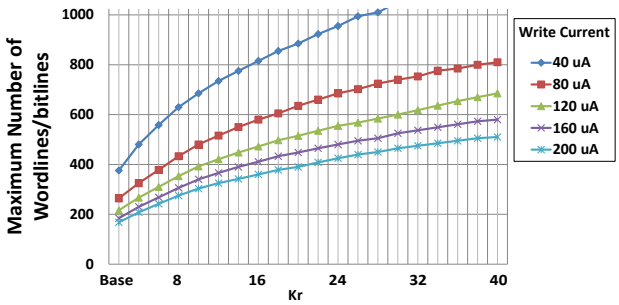

Fig. 10. The maximum array size with different nonlinearity coefficients.

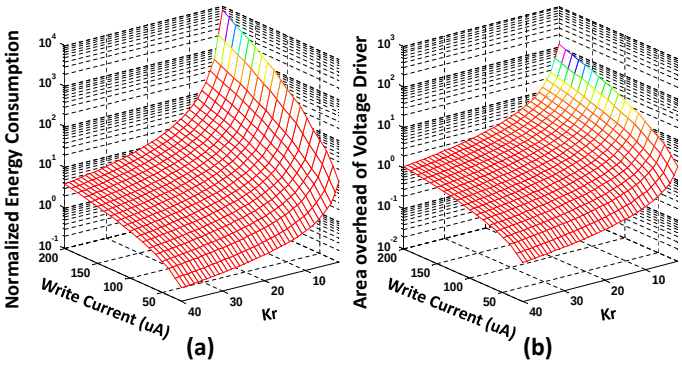

Fig. 11. Energy and area overhead comparison. (a) Energy consumption (normalized to baseline). (b) Area overhead of voltage driver (normalized to the area of cross-point array).

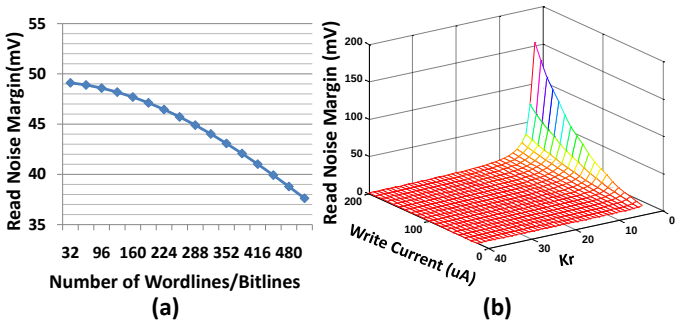

Fig. 12. Read noise margin with (a) different array size and (b) scaling of nonlinearity and write current.

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 12(a) shows the read noise margin with different array sizes for the baseline design in Section IV. 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 12(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 given 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.

VI. Conclusion

ReRAM is a promising candidate for next-generation non-volatile memory technology. The area efficient cross-point structure is the most attractive memory organization for ReRAM memories. However, problems inherent in the cross-point structure, such as the existence of sneak current and voltage drops along the wires introduce challenges to the design of reliable ReRAM cross-point memory arrays. In this paper, we first develop a mathematical model for cross-point arrays. We show that the proposed model has a simple structure and is flexible enough to evaluate different write/read schemes. By using this model, we study in detail how reliability affects the array organization, size, energy consumption, and area overheads of cross-point arrays. The simulation results show that multi-bit write operation is more energy efficient than single-bit write operation and therefore is more suitable for energy-constrained design. However, for an area-constrained design, single-bit write operation is better. Finally, we point out that both increasing nonlinearity and scaling of write current of the ReRAM cell can reduce the energy consumption and area overhead significantly, and it is favorable for large, energy efficient ReRAM design.

Appendix
Details of ReRAM Cross-Point Modeling

As mentioned in Section , the model is built upon KCL, therefore, the current equations for each cross-point can be set following

$$\Sigma_{I=1}^{k} I_k = 0. \tag{3}$$

For the sake of brevity, we assume that the wordline voltage drivers only are located at the edge of $V_{W1} \sim V_{Wm}$ and bitline multiplexes or are locate at the edge of $V_{B1} \sim V_{Bn}$. Points locate at the other two edges are left floating.

First of all, for normal points which are located inside the memory array, the KCL equations take the form of

$$R_l^{-1}V_{i,j-1} - (2R_l^{-1} + R_{i,j}^{-1})V_{i,j} + R_l^{-1}V_{i,j+1} + R_{i,j}^{-1}V'_{i,j} = 0, \tag{4}$$

for the node at wordline layer and

$$R_l^{-1}V'_{i-1,j} - (2R_l^{-1} + R_{i,j}^{-1})V'_{i,j} + R_l^{-1}V'_{i+1,j} + R_{i,j}^{-1}V_{i,j} = 0, \tag{5}$$

for all of the nodes with $1 < i < m$ and $1 < j < n$ in a $m \times n$ array.

For all of the points $V_{i,1}$ $(1 \le i \le m)$ according to different write schemes, they can be connected to the voltage driver V_{Wi} (as activated points) or left floating (as floating points). For activated points, we have

$$-(R_v^{-1}+R_l^{-1}+R_{i,1}^{-1})V_{i,1}+R_l^{-1}V_{i,2}+R_{i,1}^{-1}V'_{i,1} = -R_v^{-1}V_{Wi}, \tag{6}$$

and for floating points, we have

$$-(R_l^{-1} + R_{i,1}^{-1})V_{i,1} + R_l^{-1}V_{i,2} + R_{i,j}^{-1}V'_{i,1} = 0. \tag{7}$$

Similarly, for the points of $V'_{1,j}$ $(1 \le j \le n)$, the KCL equations take the form of

$$-(R_s^{-1}+R_l^{-1}+R_{1,j}^{-1})V'_{1,j}+R_l^{-1}V'_{2,j}+R_{1,j}^{-1}V_{1,j} = -R_s^{-1}V_{Bj}, \tag{8}$$

for activated points and

$$-(R_l^{-1} + R_{1,j}^{-1})V'_{1,j} + R_l^{-1}V'_{2,j} + R_{1,j}^{-1}V_{1,j} = 0. \tag{9}$$

for floating points.

Finally, all of the other points at $V_{i,n}$ and $V'_{m,j}$ $(1 \le i \le m, 1 \le j \le n)$ are floating points and have the form of

$$-(R_l^{-1} + R_{i,n}^{-1})V_{i,n} + R_l^{-1}V_{i,n-1} + R_{i,n}^{-1}V'_{i,n} = 0, \tag{10}$$

$$-(R_l^{-1} + R_{m,j}^{-1})V'_{m,j} + R_l^{-1}V'_{m-1,j} + R_{m,j}^{-1}V_{m,j} = 0. \tag{11}$$

Then, for clarity, a $2mn \times 1$ vector V is defined to represent all of the variables in the KCL equations:

$$V = [V_1{}^T, V_2{}^T ... V_m{}^T, V_1'^T, V_2'^T ... V_m'^T]^T, \tag{12}$$

where,

$$V_i = [V_{i,1}, V_{i,2} ... V_{i,n}]^T, \quad V'_i = [V'_{i,1}, V'_{i,2} ... V'_{i,n}]^T, \tag{13}$$

for $i = 1, 2...m$. Then all of the KCL equations can be considered as a system of linear equations, which has the form

$$A \cdot V = C. \tag{14}$$

A is a $2mn \times 2mn$ coefficient matrix, which is determined by Equations(4)-(11). C is a $2mn \times 1$ vector, containing the constant terms of these equations. Obviously, the KCL equation for each point has a relatively simple structure and they are similar to each other. Thus, the linear equation system has a fixed format and simple structure, which is easy to establish and adjust according to different design schemes and cell parameters. Moreover, matrix A is populated primarily with zeros and can be saved as a sparse matrix, which will further reduce the storage cost during the computation.

The characteristics of the linear system can be summarized as:

1) As shown in Equation (15), the coefficient matrix A can be further partitioned into four subblocks :

$$\mathbf{A} = \begin{bmatrix} A1 & A2 \\ A3 & A4 \end{bmatrix}. \tag{15}$$

All of these subblocks have the same size of $mn \times mn$. Subblock $A2$ and $A3$ are diagonal matrixes and have the value of: $A2_{i,i} = A3_{i,i} = R_{i,i}^{-1}$.

$$A2 = A3 = \begin{bmatrix} R_{1,1}^{-1} & 0 & \dots & 0 \\ 0 & R_{2,2}^{-1} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & R_{mn,mn}^{-1} \end{bmatrix}. \tag{16}$$

$A2$ and $A3$ do not change their values with different schemes. However, $A1$ and $A4$ are a little more complex than $A2$ and $A3$. $A1$ is a tridiagonal matrix and has nonzero elements only located in the main diagonal, and the first line below and above the diagonal. Similarly, A_4 is a special tridiagonal matrix, which has nonzero elements in the main diagonal, and the n^{th} line below and above the diagonal, where n is the number of bitline in the cross-point model. The value of the elements in $A1$ and $A4$ can be easily derived from Equation (4) and (5). However, the edge conditions vary with different program schemes. Therefore, the coefficients related to the edge conditions should be set according to the program schemes.
Clearly, the four edges shown in Figure 2 correspond to different coefficients in $A1$ and $A4$. Due to the space limitations, we consider the nodes at the left edge of the array as an example. A similar procedure can be followed to initiate the coefficients of other edges. The coefficients of nodes at the left edge of the array ($V_{i,1}$) can be set as:

$$A1(k,k) = \begin{cases} -(R_l^{-1} + R_{i,1}^{-1}) & \text{if } floating \\ -(R_v^{-1} + R_l^{-1} + R_{i,1}^{-1}) & \text{if } activated \end{cases} \tag{17}$$

where $k = (n-1)i + 1$ for $i = 1, 2...m$.

2) The constant terms C are $2mn \times 1$ vector. Equation(4)-(11) show that only KCL equations of the activated points have constant terms. Therefore, only the following elements in C may have non-zero value: $C((i-1)n+1)$, $C(in)$, $C(mn+i)$ and $C((2m-1)n+i)$ for $i = 1, 2...m$, corresponding to the nodes at the four edges respectively. Likewise, as an example, we consider nodes $V_{i,1}$. The constant corresponding to these nodes can be defined as:

$$C((i-1)n+1) = \begin{cases} 0 & \text{if } floating \\ -R_v^{-1} V_{Wi} & \text{if } activated \end{cases} \tag{18}$$