Constructing Fast and Energy Efficient 1TnR based ReRAM Crossbar Memory

Lei Zhao[1], Lei Jiang[2], Youtao Zhang[1], Nong Xiao[3], and Jun Yang[1]
[1]University of Pittsburgh
[2]Indiana University Bloomington
[3]National University of Defense Technology
lez21@pitt.edu, jiang60@iu.edu, zhangyt@cs.pitt.edu, nongxiao@nudt.edu.cn, juy9@pitt.edu

Abstract— ReRAM (Resistive Random Access Memory) is an emerging non-volatile memory technology that exhibits high cell density and low standby power. ReRAM crossbars, while having the smallest $4F^2$ cell size, suffer from large sneak leakage, which not only wastes dynamic energy but also degrades system performance significantly.

In this paper, we propose V-ReRAM, a novel ReRAM crossbar design based on 1TnR cell structure. By reorganizing the peripheral circuit, V-ReRAM greatly reduces the number of half-selected cells and thus the sneak leakage. V-ReRAM further improves RESET performance by exploiting the RESET latency difference among memory cells in ReRAM crossbars. Our experimental results show that, on average, V-ReRAM improves the system performance by 7.3% and reduces memory energy consumption by 72%, comparing to the baseline 1T4R based ReRAM crossbar.

Keywords— ReRAM, sneak current, RESET, 1TnR, crossbar

I. Introduction

Modern computer systems exhibit increasing demands for large capacity memory, due to the wide adoption of chip multiprocessors as well as the proliferation of data-intensive applications. Unfortunately, traditional DRAM faces significant power, leakage, and process variation challenges. Memory systems may consume more than 25% of system power [17]. A more serious drawback of DRAM is its scalability. The recent ITRS report [8] indicates that there is no path forward to scale DRAM below 16nm. These challenges jeopardize the applicability of DRAM in future systems, which inspires the search for non-volatile memory alternatives.

ReRAM (Resistive Random Access Memory) is an emerging non-volatile memory technology that has many advantages. It exploits the resistance of Metal-Oxide-Metal structure to represent stored information. ReRAM has fast read and write speeds, low energy consumption. ReRAM can achieve $4F^2$ cell size (the smallest 2D cell size) by adopting crossbar structure, or even smaller per bit die area by 3D stacking. As such, in addition to the schemes that adopt ReRAM to replace NAND flash SSDs [16], recent efforts from both industry [4], [7] and academia [23] architected ReRAM as DRAM complement for future main memory systems. ReRAM crossbar based main memory can significantly enlarge memory capacity to meet the increasing memory demand in modern data intensive applications.

Unfortunately, ReRAM crossbars suffer from large sneak path leakage. This is because an ReRAM crossbar cannot completely isolate the to-be-accessed cells from other cells in one subarray, resulting in large *sneak currents* on the selected wordline and bitlines. Sneak current not only wastes dynamic energy but also leads to large voltage drop and degraded memory performance [23]. Optimizing ReRAM crossbar has been a major focus in recent studies, e.g., identifying better ReRAM devices [16], [4], designing better cell structures [24] and better architectural support [23]. However, it remains challenging to achieve good performance and reduce energy consumption in ReRAM crossbars.

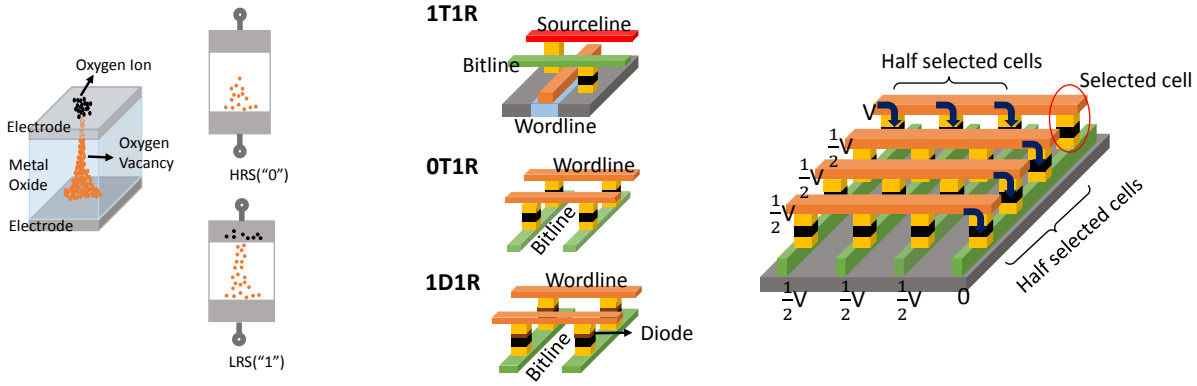
In this paper, we propose a 1TnR-based ReRAM crossbar structure to achieve performance improvement and energy efficiency. We summarize our contributions as follows.

- We propose V-ReRAM to reduce the number of half-selected cells in each memory access, which leads to large reduction of voltage drop during the access. Given that RESET latency reduces exponentially with smaller voltage drop, V-ReRAM achieves significant performance and energy consumption improvements.
- We propose to exploit the latency difference when RESETting the cells stored at different places in cell subarray, which enables location-aware RESET strategy to further improve the write performance of ReRAM crossbar.
- We evaluate V-ReRAM with comparison to the state-of-the-art. Our experimental results show that on average, V-ReRAM improves system performance by 7.3% and reduces memory energy consumption by 72%.

In the rest of the paper. We briefly discuss ReRAM basics in Section 2. We motivate and elaborate the designs in Section 3. We summarize the experiment methodology in Section 4 and analyze the results in Section 5. We discuss more related work in Section 6 and conclude the paper in Section 7.

II. ReRAM Basics

ReRAM [1], [21], [22] is an emerging non-volatile memory technology that uses resistance to represent stored information. By adopting simple metal-Oxide-metal (MOM) stack, i.e., sandwiching metal oxide material between metal electrodes (as shown in Figure 1(a)), an ReRAM cell can switch between high resistance state (HRS) and low resistance state (LRS), representing logic '0' and '1', respectively. The operations that switch ReRAM cell between



(a) ReRAM cell structure (b) Three ReRAM cell array types (c) ReRAM has large sneak leakage

Fig. 1: ReRAM cell and three different types of cell array structures.

HRS and LRS are referred to as RESET and SET operations, respectively. Recent studies showed that many materials, such as TaO_2 [10] and HfO_2 [14], exhibit resistance switching effect.

Based on if accessing an ReRAM cell needs an access selector and what selector it has, there are three types of ReRAM cell array structures. As shown in Figure 1(b), the 0T1R cell array represents the high density ReRAM crossbar that does not use selector. It was adopted in many of recent prototypes [21], [6]. A major concern of ReRAM crossbar is that, when accessing one or multiple cells in the crossbar, there exists significant large current, referred to as *sneak current*, flowing through other cells on the selected bitline and wordline, referred to as *sneak paths*. To mitigate sneak current in ReRAM crossbar, when performing a write operation, e.g., a RESET operation, the driver sets the selected wordline to V_{RESET}, the selected bitline to 0, and all other bitlines and wordline to $V_{RESET}/2$, as shown in Figure 1(c). Performing a SET operation uses opposite current direction and V_{SET} voltage. The cells to be written are referred to as *fully-selected* cells because they are under full voltage stress. Other cells on the selected bitlines and wordlines are referred to as *half-selected* cells. The rest of the cells are *not-selected* cells.

Alternatively, it is possible to minimize sneak current by adopting traditional 1T1R cell, i.e., each cell consists of one access transistor and one ReRAM device (Figure 1(b)). Such an organization leads to much larger cell size, e.g., 20 F^2 per cell [22], which reduces memory density and increases per bit fabrication cost. The 1D1R cell array (Figure 1(b)) is a compromise that integrates one bipolar diode selector on top of the ReRAM device in the crossbar. Comparing to 0T1R, 1D1R structure enlarges non-linearity κ, the ratio between the current of fully-selected cell and the current of half-selected cell. κ is a key parameter that determines how severe the sneak leakage is and how large an ReRAM crossbar may be fabricated [18].

A. 1TnR ReRAM structure

We next discuss the 1TnR structure [24] on which our V-ReRAM crossbar is based. To simplify the discussion, we elaborate the design using 1T4R while the devised schemes are applicable to , e.g., 1T8R, structures as well. As shown in Figure 2(a), a 1T4R cell array is organized as a crossbar to achieve $4F^2$ cell size — there is one ReRAM cell at each crosspoint of bitline and (local) wordline. Each local wordline connects four ReRAM cells to its wordline selection transistor. These four cells are referred to as one LCG (local cell group) in this paper. In addition, two neighboring LCGs connect to one global wordline, and one wordline selection signal controls half of wordline selection transistors in one column, e.g., `WSL0` selects the first and the third LCGs simultaneously.

To RESET one or multiple cells in one LCG, e.g., ReRAM cell `C0` in Figure 2(c), `GWL0` connects to the RESET voltage source V_{RESET}, bitline `BL0` connects to `GND`, and `WSL0` is enabled, such that V_{RESET} is applied to `C0`. To minimize sneak leakage, `BL1` to `BL3` and `GWL1` connect to $V_{RESET}/2$ voltage source while WSL_1 is disabled. The not-to-be-written cells in this LCG, i.e., `C1`, `C2`, and `C3`, are half-selected. In addition, `C8` is also half-selected as `BL0` and `GWL1` have 0 and $V_{RESET}/2$ voltage, respectively. `WSL1` enables not only the first LCG but also the third LCG. This cell structure can achieve $4F^2$ cell size because wordline selection and global wordline wires are laid out in vertical layers (Figure 2(b)).

In the figure, when a wordline selection transistor is OFF, its corresponding LCG cells are disconnected from the global wordline, resulting in negligible sneak leakage. The wordline selection signal drives a group of interleaving LCGs along the bitline direction. When one LCG is selected, other LCGs from this group are also selected, resulting in many half-selected cells.

III. The Design Details

A. Motivation

In this paper, we assume a memory rank consists of eight memory chips and one ECC chip. Each chip contains data from eight banks. Each bank is partitioned to a set of subarrays. Given one logical memory line, its data bits are saved in multiple subarrays across all chips. By default, we use 512bit x 512bit subarrays and each subarray provides 16 bits for one memory access. Accessing one 64B or 512-bit memory line needs to activate 32 subarrays in total,

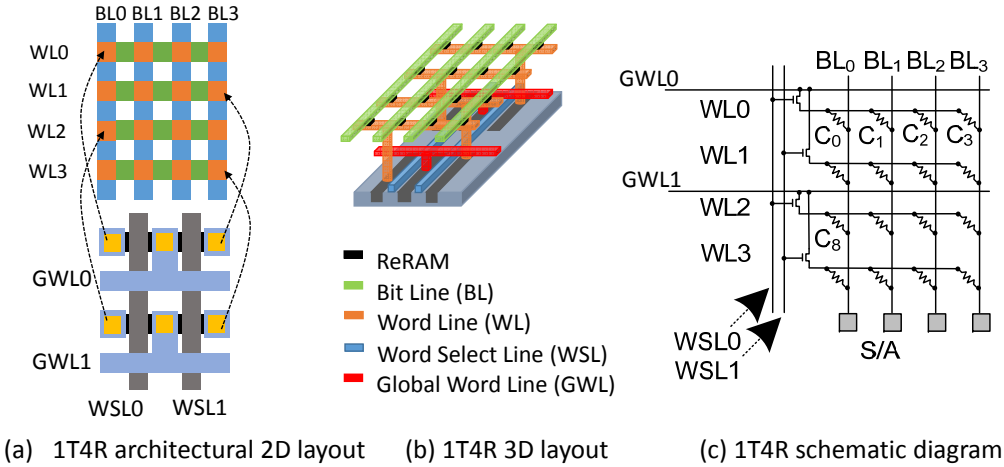


Fig. 2: A 1T4R based ReRAM crossbar has $4F^2$ cell size.

i.e., four subarrays in each chip.

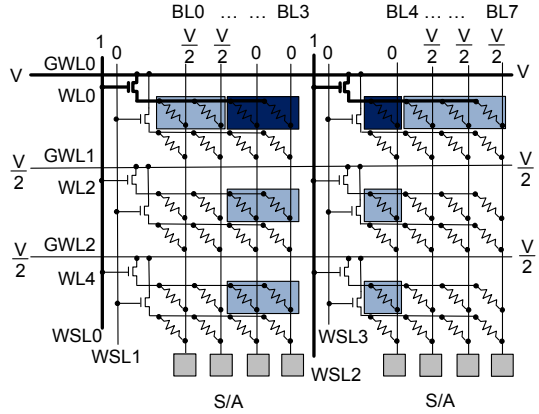


Fig. 3: A 1T4R crossbar activates many HS cells (Dark blue shaded boxes indicate fully-selected cells. Light blue shaded boxes indicate half-selected cells).

In general, accessing multiple bits from a 1T4R based crossbar subarray needs to activate a number of LCGs (local cell groups, i.e., the cells controlled by one wordline selection signal on one row)[1]. For the example shown in Figure 3, we need to RESET three cells in an `N×N` subarray. The selected global wordline `GWL0` connects to the voltage driver that provides the write voltage `V`. The selected bitlines connect to the ground `GND`. Assume the three cells are from two LCGs, we need to activate `WSL0` and `WSL2`. Thus, in addition to the three fully-selected (FS) cells, we need to activate many half-selected (HS) cells — (i) `WSL0` activates local wordlines interleavingly, that is, `WL0`, `WL2`, etc. It enables one LCG from each activated `WL`. (ii) in the LCGs that have FS cells, other cells are also half-selected, in this example, there are 2+3=5 such cells. To summarize, this three-bit RESET operation needs to activate three FS cells and `3×(N/2-1)+(2+3)` HS cells, or 770 HS cells when N=512.

Both FS and HS cells lead to voltage drop along the selected wordline and bitlines. Recent studies [6], [23] revealed the relationship between the voltage difference across the target cell and switching time is shown in Equation 1:

$$t \times e^{k \cdot V_d} = C \tag{1}$$

where t is the switching time; V_d denotes the voltage difference across the cell; C and k are experimental constants. In particular, the cell RESET time is sensitive to voltage drop. A 0.4V more voltage drop could lead to 10× RESET latency increase [6].

An ReRAM subarray integrates local row decoders and sense amplifiers, which have large area overhead. It is highly preferable to fabricate large subarrays in order to improve die area efficiency. However, the super non-linearity relationship between RESET latency and voltage drop puts a tight restriction on (i) subarray size; and (ii) the number of cells that can be RESET simultaneously. Larger subarray and larger number of concurrent RESET can significantly degrade RESET speed and thus the ReRAM memory performance.

In this paper, we focus on designs that can effectively reduce RESET time. The baseline 1T4R crossbar has adopted recent latency optimization schemes, i.e., double-sided voltage driver and RESET batching from [23].

B. The V-ReRAM Design Details

To mitigate the voltage drop in 1T4R crossbar, we propose V-ReRAM organization as shown in Figure 4. Intuitively, it exchanges bitlines and wordlines, and relocates sensing circuits to minimize HS cells in each memory access. It lays out the cells from one memory line along vertical bitlines, referred to as new wordlines (`NWLs`), and relocates sense amplifiers to the right of the subarray. For discussion purpose, the original `GWLs` are referred to `NBLs` (new bitlines).

Assume the cells accessed in Figure 3 are now stored in `NWL0`, as shown in Figure 4. To write the three to-be-RESET cells, we only need to enable `WSL0`. All other `WSLs` are disabled. V-ReRAM applies write voltage `V` on both sides of `NWL0`, and connects three `NBLs` (i.e., `NBL2`, `NBL3`, and `NBL4`) to `GND` to enable writing the three cross-point

[1]We assume the bits in one LCG are from the same memory line as otherwise one memory access needs to activate more LCGs.

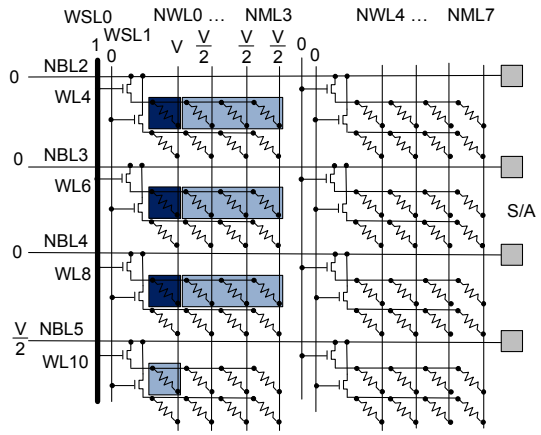


Fig. 4: V-ReRAM activates fewer HS cells.

cells. In the new subarray configuration, `WSL0` enables half of `NBL`s. Except the three fully selected to-be-RESET cells, other cells on `NWL0` are half selected as well. For each of `NBL2`, `NBL3`, and `NBL4`, there are three half-selected cells, i.e., those in the cell group with a fully selected cell. At the bank level, S/A connects to new global bitlines that send out the sensed data, or receive the data to be written to the subarray. There is one thing need to be noticed here, since we changed the directions of wordlines and bitlines, the number of cells on one wordline is half of that in baseline (recall the interleaved-activated LCGs). In order to make the comparation more fair, we adopt the subarray size of `(2N)`×`(N/2)` in V-ReRAM for all our evaluations.

To summarize, V-ReRAM activates three FS cells, and `N+6` (=N-3+3×3), or 518 HS cells in the RESET operation, representing a large improvement of the baseline structure. Reducing HS cells results in reduced voltage drop and thus improved access latency. Based on our experiments, SET operations are much faster and less sensitive to voltage drop than RESET operations, thus we mainly focus on RESET operations in our study.

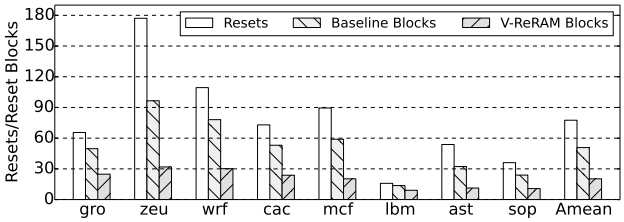


Fig. 5: Comparing the number of activated cell groups.

Figure 5 compares the average number of activated cell groups in the baseline and V-ReRAM. Section 4 lists the setting details. From the figure, on average, one write operation needs to RESET 77 cells that spread in 50 groups in different subarrays. Some subarrays need to activate two or more groups. The baseline needs to activate 19,758 HS cells during the RESET phase of the write operation; V-ReRAM needs to enable 20 WSL signals on average, resulting in 10,394 HS cells. By greatly reducing the number of HS cells, V-ReRAM is expected to gain both performance and energy consumption benefits.

Sensing circuit. In V-ReRAM, S/As are relocated to the right of the subarray, which greatly reduces the number of HS cells on each bitline. Given that the performance of S/A depends on the load on each bitline, V-ReRAM has the potential of achieving better read performance. This paper focuses on RESET performance improvement and thus does not exploit this potential.

Because each subarray provides 16 bits for one memory line, V-ReRAM integrates 16 S/As, the same number of S/As in the baseline. While supporting bursty read mode needs more S/As, the number is kept be the same in either the baseline or the V-ReRAM design.

C. Location-aware RESET scheduling

The ReRAM cells in V-ReRAM, depending on their locations, have different voltage drops. In particular, cells that are closer to the voltage drivers suffer from smaller voltage drops. Given that there are two write drivers at the both sides of selected `NWL` [23], the cells located in the middle of each `NWL` suffer from the largest drop. Due to super non-linear relationship between a cell's RESET switching time and the applied voltage on one cell, these cells have much larger RESET latency.

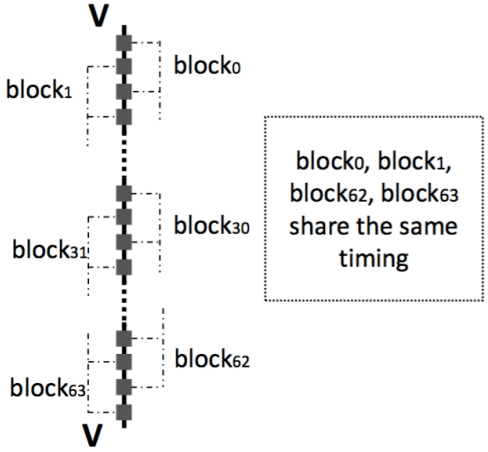


Fig. 6: Variable RESET latency in V-ReRAM.

In Figure 6, the 1024 cells on one `NWL` are partitioned to 64 blocks; each block saves 16 cells from one memory line; the cells are saved interleavingly. The latency of RESET operations in may blocks, e.g., $block_0$ to $block_{12}$, are much faster than the worst case, i.e., $block_{31}$. Figure 7 compares the RESET latencies when resetting different blocks. We show 16 groups because (i) $group_{2i}$ and $group_{2i+1}$ ($0 \le i \le 31$) are interleaved together and thus have similar latencies. (ii) due to using two voltage drivers at both ends of `NWL`, $group_{2i}$ and $group_{2j}$ ($0 \le i, j \le 31$, $i+j=31$) share similar latencies.

Based on the above observation, we propose a location-aware, concurrent cell RESET-aware programming strategy as follows. V-ReRAM adopts flip-n-write [2] such that only changed cells need to be written. One write operation is divided into RESET phase and SET phase, which write 0s and 1s, respectively. Since previous designs do not explore the latency differencies along the `NWL`, they use the

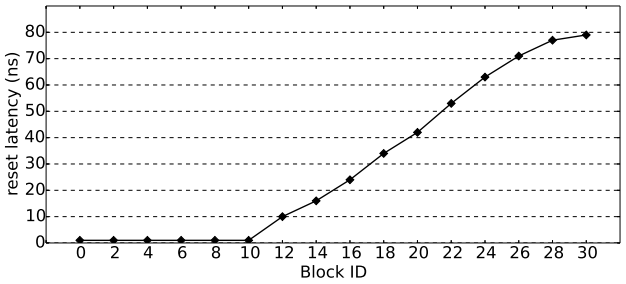


Fig. 7: RESET latency depends on the location along `NWL`.

worst case RESET latency ($block_{31}$) for every RESETs. In contrary, V-ReRAM can terminate the RESET phase of writing a subarray based on a table T[i] ($0 \leq i \leq 15$) that records the corresponding latency of resetting block `2i` (or `2i+1`, or `64-2i`, or `64-(2i+1)`) (as shown in Figure 7). Other timing parameters are shown in Section 4, similar to those in [23].

IV. Experimental Methodology

To evaluate the proposed designs, we use the Verilog-A model [9] of ReRAM to build a whole subarray in HSpice. The latency of RESET and SET operations are simulated in the spice model. The ReRAM parameters used in the spice model is shown in Table I. We use NVsim[3] to get the area and peripheral circuit's latency and power parameters. We then plug these parameters in our architectural evaluation. We assumed 32nm technology node in both spice and NVSim simulation.

We used an in-house simulator to evaluate the performance and energy consumption with different designs. The simulator models a full memory hierarchy including L1/L2 caches and ReRAM based main memory. The detailed configuration of the simulated system is listed in Table I. We performed our evaluation on a subset of SPEC2006 benchmarks that are classified into *write sensitive* and *write insensitive* categories, according to the number of read/write operations per thousand instructions (RPKI/WPKI), as shown in Table II.

In the paper, we implemented and compared three schemes as follows:

— `Baseline`. It is the scheme that uses the 1T4R crossbar ReRAM array in [24]. The baseline adopts *flip-n-write* and *double-sided voltage driver* [23] for latency reduction.

— `V-ReRAM`. It represents the basic V-ReRAM subarray organization. The multiple cells from one memory line are laid out in cell groups controlled by one WSL with one cell from each group.

— `V-ReRAM-Var`. It is built on top of `V-ReRAM` and exploits the latency difference based on its data location in the cell subarray.

V. Experimental Results

A. Hardware Overhead

We estimated the area overhead of V-ReRAM designs in NVsim [3]. The results showed that the area increase due to structure reorganization is less than 1%.

TABLE I: System Configuration

	Parameters
Processor	8 cores single issue in-order CMP; 4GHz
L1 I/D-cache	Private; 32KB; 2-way; 64-byte block size; 2 cycle latency
L2 cache	Private; 2MB; 4-way; 64-byte block size; 10 cycle read latency
Main memory	8GB; 1 channel; 2 ranks; 8 banks per rank; 24-entry write queue per bank
Timing Parameters [23]	tRCD=18; tCL=15; tCWD=13; tFAW=30; tWTR=7.5
ReRAM Parameters	MatSize=512x512 or 1024x256; K_r=14; R_{on}=50K; R_{off}=2500K; V_{reset}=3V; V_{set}=3V

TABLE II: Benchmark Classification

Write sensitive	WPKI	RPKI	Write non-sensitive	WPKI	RPKI
mcf	19.89	47.32	gromacs	0.18	0.55
lbm	6.83	31.78	zeusmp	0.19	4.01
astar	4.43	9.02	wrf	0.01	0.02
soplex	2.72	16.79	cactusADM	0.19	4.57

The latency table has 16 entries, or 32B if each timing value needs 2B. The table is saved in the bridge chip, which is responsible for fine-grained timing control. The bridge chip is also adopted in the baseline, in order to exploit the latency difference when RESETting different numbers of cells [23]. Recent studies showed that it is important to integrate bridge chip on memory DIMM [5] to accommodate the significant access-to-access latency difference in emerging non-volatile memories.

B. Average RESET Latency

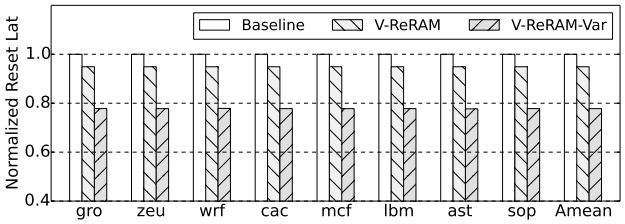


Fig. 8: Comparing average RESET latencies under different schemes (normalized to `baseline`).

Figure 8 compares the average RESET latencies in different schemes, with the results normalized to `Baseline`. On average, `V-ReRAM` and `V-ReRAM-Var` reduce 6% and 23% RESET latencies, respectively. The large reduction is mainly due to the smaller number of HS cells and thus smaller voltage drop in the subarray.

C. Performance

Figure 9 compares the performance of different schemes, with the results normalized to `baseline`. While `V-ReRAM` reduces the average RESET latency, the performance improvement depends on the intensity of write operations. Write-intensive benchmarks, e.g., `mcf` and `lbm`, tend to achieve larger performance improvements. On average, `V-ReRAM` achieves 1.5% improvement over `baseline`.

V-ReRAM-Var achieves more performance improvement by exploring the latency difference among different blocks. Comparing to V-ReRAM, it achieves up to 9% more improvement for write-intensive benchmarks, and, on average, 7.2% more improvement over baseline.

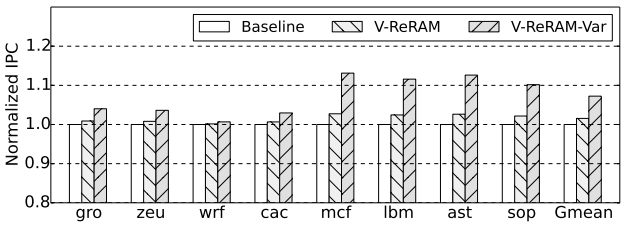


Fig. 9: The normalized IPCs of different schemes.

D. Dynamic Energy Consumption

Figure 10 summarizes the dynamic memory energy consumption in different schemes. The results are normalized to those of Baseline. From the figure, V-ReRAM consumes 28% of the dynamic energy in Baseline. This is because V-ReRAM reduces not only the number of half-selected cells in the crossbar, but also the number of activated wordlines. As a result, V-ReRAM not only reduces the energy consumption of RESET operations but also the READ and SET operations. V-ReRAM-Var achieves negligible energy benefits over V-ReRAM.

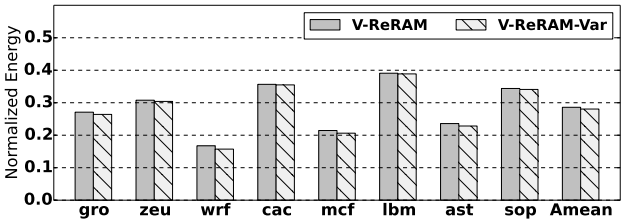


Fig. 10: The normalized dynamic memory energy consumption of different schemes.

E. Sensitivity Study

We then studied the proposed designs with different subarray sizes and different numbers of cells from one subarray.

Different subarray sizes. Figure 11 compares the IPC and dynamic energy consumption with different array sizes. The results are normalized to those of Baseline. Each subarray still provides 16 bits for each memory line. The results are normalized to baseline at each size. V-ReRAM achieves larger improvement with increasing subarray size — the performance improvement increases to 5.3 times when the subarray size doubles, and decreases to 0.1% when the size halves. This is because a small subarray has small voltage drop in the baseline such that V-ReRAM has less performance impact. Due to the reduction of half-selected cells and activated wordlines, V-ReRAM effectively reduces the energy consumption even with smaller subarray sizes (Figure 11b).

Different cells per subarray. Figure 12 compares the IPC and dynamic energy consumption when each subarray provides different numbers of cells, referred to as *Cnum*. The results are normalized to those of Baseline. We fixed

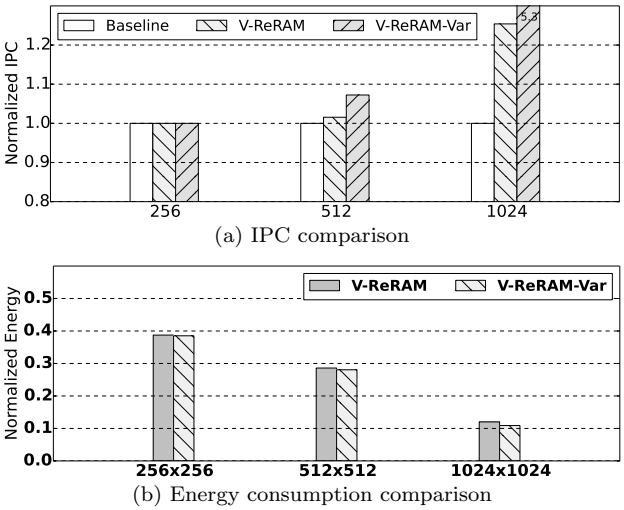


Fig. 11: The comparison of different subarray sizes.

the subarray size to be 512x512 in Baseline and 1024x256 in V-ReRAM and V-ReRAM-Var. By adopting flip-n-write, up to half of these cells may be RESET simultaneously. Given a larger *Cnum*, baseline needs to activate more cell groups in one subarray, resulting in more sneaky leakage and degraded performance.

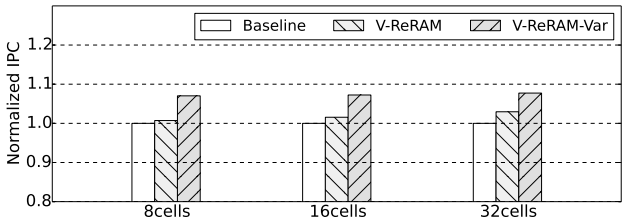
From the figure, we observed stable performance improvements with different *Cnum* values. V-ReRAM activates a smaller number of half-selected cells with a larger *Cnum* and thus achieves larger energy consumption reduction.

VI. Related Work

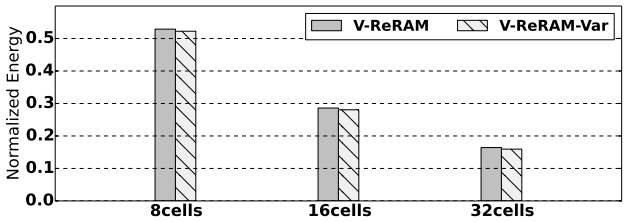
Since the sneak path leakage has a large impact on performance and energy efficiency of crossbar ReRAM structures, there have been many recent studies addressing this issue. [15] proposes weighted sensing scheme to suppress the sneak-path leakage of unselected cells. [13] studies the impact of sneak current on read failure. [23] proposed architectural innovations, including double-sided voltage drivers to reduce the voltage drop along wordline on the worst case cell, flip-n-write to reduce the to-be-accessed cells, and RESET grouping to split RESET into multiple rounds. They combine all these techniques to reduce the impact of sneak current on RESET latency. [19] addresses the read latency degradation caused by sneak current. They proposed a method to reuse the measured background sneak current to help improve the subsequent read operations. [11] explores the complementary resistive switch (CRS) characteristics by stacking two opposite ReRAM cells together to construct a reconfigurable memory architecture that can switch between CRS mode and normal mode. Similar to our exploration of variant latencies along the long wordline in crossbar ReRAM architectures, [12], [20] exploits the latency difference in long bitlines for DRAM array structures.

VII. Conclusion

ReRAM is an emerging memory technology with great potential to be architected as main memory. ReRAM cross-



(a) IPC comparison



(b) Energy consumption comparison

Fig. 12: The comparison of different cells per subarray.

bars, while achieving high density, face significant sneak path leakage challenges. In this paper, we devised V-ReRAM crossbar design to reduce the number of HS cells, so that the performance and energy efficiency both improves upon the baseline. We also explored the latency difference of cells along the long wordline. Based on this observation, our location aware scheme further improves system performance and memory energy consumption.

References

[1] L.O. Chua, "Memristor - The Missing Circuit Element," In *IEEE Trans. of Circuit Theory*, 18, 1971.

[2] S. Cho, "Flip-N-Write: A Simple Deterministic Technique to Improve PRAM Write Performance, Energy and Endurance,", In *MICRO*, 2009.

[3] X. Dong, *et al.*, "NVSim: A Circuit-Level Performance, Energy, and Area Model for Emerging Non-Volatile Memory", In *TCAD*, 2012.

[4] R. Fackenthal, *et al.*, "A 16Gb ReRAM with 200MB/s Write and 1GB/s Read in 27nm Technology," In *ISSCC*, 2014.

[5] K. Fang, *et al.*, "Memory Architecture for Integrating Emerging Memory Technologies," In *PACT*, 2011.

[6] B. Govoreanu, *et al.*, "10x10nm^2 Hf/HfO$_x$ Crossbar Resistive RAM with Excellent Performance, Reliability and Low-Energy Operation," In *IEDM*, 2011.

[7] HP & Sandisk, "The Memristor Project," announced Oct. 2014.

[8] ITRS, "The International Technology Roadmap for Semiconductors Report," http://www.itrs.net.

[9] Zizhen Jiang, *et al.*, "Stanford University Resistive-Switching Random Access Memory (RRAM) Verilog-A Model," *https://nanohub.org/publications/19/1*, 2014.

[10] Y.-B. Kim, *et al.*, "Bi-layered RRAM with Unlimited Endurance and Extremely Uniform Switching," In *IEEE Symposium on VLSI Technology*, 2011.

[11] M.A. Lastras-Montano, *et al.*, "A Low-Power Hybrid Reconfigurable Architecture For Resistive Random-Access Memories," In *HPCA*, 2016.

[12] D. Lee, *et al.*, "Tiered-Latency DRAM: A Low Latency and Low Cost DRAM Architecture," In *HPCA*, 2013.

[13] Y. Li, *et al.*, "Understanding the Impact of Diode Parameters on Sneak Current in 1Diode 1ReRAM Crossbar Architectures," In *NANOARCH*, 2013.

[14] H. Lee, *et al.*, "Low Power and High Speed Bipolar Switching with A Thin Reactive Ti Buffer Layer in Robust HfO2 Based RRAM," In *IEDM*, 2008.

[15] C. Liu, *et al.*, "A Weighted Sensing Scheme for ReRAM-Based Cross-Point Memory Array," In *ISVLSI*, 2014.

[16] T.-Y. Liu, *et al.*, "A 130.7mm^2 2-layer 32Gb ReRAM Memory Device in 24nm Technology," In *ISSCC*, 2013.

[17] D. Meisner, *et al.*, "PowerNap: Eliminating Server Idle Power," In *ASPLOS*, 2009.

[18] D. Niu, "Design Trade-offs for High Density Cross-point Resistive Memory", In *ISLPED*, 2012.

[19] M. Shevgoor, *al.*, "Improving Memristor Memory with Sneak Current Sharing," In *ICCD*, 2015.

[20] Y.H. Son, *al.*, "Reducing Memory Access Latency with Asymmetric DRAM Bank Organizations," In *ISCA*, 2013.

[21] D.B. Strukov, *et al.*, "The Missing Memristor Found," In *Nature*, 2008.

[22] H.-S. Wong, *et al.*, "Metal Oxide RRAM," In *Proceedings of IEEE*, 2012.

[23] C. Xu, *et al.*, "Overcoming the Challenges of Crossbar Resistive Memory Architectures," In *HPCA*, 2015.

[24] C.W. Yeh, *et al.*, "Compact One-Transistor-N-RRAM Array Architecture for Advanced CMOS Technologoy," In *JSSC*, 50(5), 2015.