CACF: A Novel Circuit Architecture Co-optimization Framework for Improving Performance, Reliability and Energy of ReRAM-based Main Memory System

YANG ZHANG, DAN FENG, WEI TONG, YU HUA, JINGNING LIU, ZHIPENG TAN, CHENGNING WANG, BING WU, ZHENG LI, and GAOXIANG XU, Wuhan National Laboratory for Optoelectronics, Key Laboratory of Information Storage System (School of Computer Science and Technology, Huazhong University of Science and Technology), Ministry of Education of China

Emerging Resistive Random Access Memory (ReRAM) is a promising candidate as the replacement for DRAM due to its low standby power, high density, high scalability, and nonvolatility. By employing the unique crossbar structure, ReRAM can be constructed with extremely high density. However, the crossbar ReRAM faces some serious challenges in terms of performance, reliability, and energy consumption. First, ReRAM's crossbar structure causes an IR drop problem due to wire resistance and sneak currents, which results in nonuniform access latency in ReRAM banks and reduces its reliability. Second, without access transistors in the crossbar structure, write disturbance results in serious data reliability problem. Third, the access latency, reliability, and energy use of ReRAM arrays are significantly influenced by the data patterns involved in a write operation.

To overcome the challenges of the crossbar ReRAM, we propose a novel circuit architecture co-optimization framework for improving the performance, reliability, and energy use of ReRAM-based main memory system, called CACF. The proposed CACF consists of three levels, including the circuit level, circuit architecture level, and architecture level. At the circuit level, to reduce the IR drops along bitlines, we propose a double-sided write driver design by applying write drivers along both sides of bitlines and selectively activating the write drivers. At the circuit architecture level, to address the write disturbance with low overheads, we propose a RESET disturbance detection scheme by adding disturbance reference cells and conditionally performing refresh operations. At the architecture level, a region partition with address remapping method is proposed to leverage the nonuniform access latency in ReRAM banks, and two flip schemes are proposed in different regions to optimize the data patterns involved in a write operation. The experimental results show that CACF improves system performance by 26.1%, decreases memory access latency by 22.4%, shortens running time by 20.1%, and reduces energy consumption by 21.6% on average over an aggressive baseline. Meanwhile, CACF significantly improves the reliability of ReRAM-based memory systems.

This work was supported by the National High Technology Research and Development Program (863 Program) No.2015AA015301; National Natural Science Foundation of China (NSFC), No.61772222, No.61772212, No.U1705261, No.61472153 and No.61502191; State Key Laboratory of Computer Architecture, No.CARCH201505. This work was also supported by Engineering Research Center of Data Storage Systems and Technology, Ministry of Education, China.

Extension of Conference Paper: This article is extended from "A Novel ReRAM-based Main Memory Structure for Optimizing Access Latency and Reliability" published in the Proceedings of the 54th Design Automation Conference, 2017 (DAC'2017).

Authors' addresses: Y. Zhang, D. Feng (corresponding author), W. Tong, Y. Hua, J. Liu (corresponding author), Z. Tan, C. Wang, B. Wu, Z. Li, and G. Xu are with the Wuhan National Laboratory for Optoelectronics, School of Computer Science and Technology, Huazhong University of Science and Technology, Division of Data Storage System, Wuhan 430074, China; emails: {youngzhang, dfeng, tongwei, csyhua, jnliu, tanzhipeng, chengningwang, wubin200, lizheng, gxxu}@hust.edu.cn.

https://doi.org/10.1145/3195799

CCS Concepts: • **Computer systems organization** → **Architectures**; • **Hardware** → **Emerging technologies**; • **Information systems** → **Information storage systems**;

Additional Key Words and Phrases: ReRAM, crossbar, IR drop, nonuniform access latency, write disturbance, data patterns

ACM Reference format:
Yang Zhang, Dan Feng, Wei Tong, Yu Hua, Jingning Liu, Zhipeng Tan, Chengning Wang, Bing Wu, Zheng Li, and Gaoxiang Xu. 2018. CACF: A Novel Circuit Architecture Co-optimization Framework for Improving Performance, Reliability and Energy of ReRAM-based Main Memory System. *ACM Trans. Archit. Code Optim.* 15, 2, Article 22 (May 2018), 26 pages.
https://doi.org/10.1145/3195799

1 INTRODUCTION

DRAM has been applied to main memory for more than four decades. However, the scalability of DRAM reaches its bottleneck due to increasing power consumption, degraded reliability, and the difficulty of constructing high aspect ratio capacitors [35, 39, 40, 52, 60, 66]. As ITRS indicates, the scaling path of DRAM beyond 16nm is not clear [57]. Consequently, emerging Nonvolatile Memories (NVMs) such as Phase Change Memory (PCM), Spin-Transfer Torque RAM (STT-RAM), and Resistive RAM (ReRAM) are actively explored as replacements for DRAM due to their better scalability with low power consumption and high density [25, 27, 44, 54, 65, 67, 70]. Among these candidates, ReRAM is the most promising due to its lower power consumption and higher density.

The history of ReRAM goes back to 1971, when Leon Chua predicted the existence of this device [7]. However, ReRAM was not experimentally created until 2008 by the HP Labs [51, 56]. ReRAM's resistance switch depends on the time integral of current across its terminals. Therefore, ReRAM can keep its resistance without electrical current, which makes it suitable as nonvolatile memory. ReRAM cells can be built into a crossbar structure without access transistors to reach extremely high density due to their nonlinearity. Moreover, multiple layers of crossbars can be stacked together through 3D integration technology to achieve even higher density [22, 58]. Recently, Micron and HP have developed crossbar ReRAM prototypes for main memory systems [9, 16]. Intel and Micron have also announced that 3D crossbar technology has been applied to 3D XPoint memory [18].

Although the crossbar structure is effective building high-density ReRAM arrays, it also faces some serious challenges in terms of performance, reliability, and energy consumption. First, the crossbar structure causes an IR drop problem due to wire resistance and sneak currents. As the size of arrays becomes larger, the magnitude of IR drops increases obviously. Unfortunately, when the IR drop reaches a certain value, the voltage applied on the ReRAM cells will be too small to perform a reliable write or read. Moreover, the switching time of an ReRAM cell is exponentially inversely proportional to the voltage applied on the cell. So the IR drop problem significantly increases the switching time of ReRAM cells. Furthermore, the IR drops of ReRAM cells are various in a crossbar array, which causes nonuniform access latency in ReRAM memory banks. Second, without access transistors in the crossbar structure, writing an ReRAM cell may affect the resistances of other cells sharing the same wordline and bitline. The effect is accumulative for a series of write operations and may eventually result in data corruption, which greatly reduces the reliability of ReRAM arrays [11, 24]. This effect is referred to as *write disturbance*. Third, the access latency, reliability, and energy use of a crossbar array are significantly influenced by the data patterns during a write operation, especially for a multiple-bits writing crossbar array. Due to the nonlinearity of ReRAM cells, writing more 0s into a crossbar array increases the access latency but also benefits reliability and energy use. Writing more 1s into a crossbar array reduces the access latency but harms

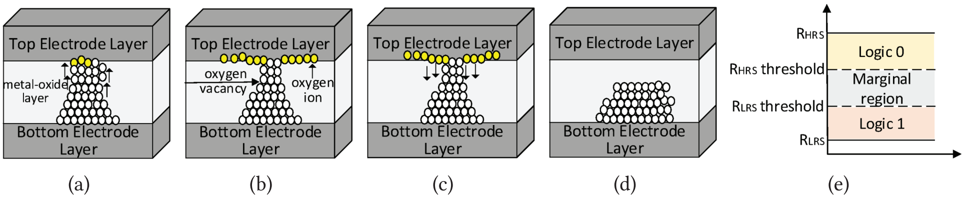


Fig. 1. ReRAM cell structure. (a) SET operation, (b) LRS, (c) RESET operation, (d) HRS, (e) resistance regions.

reliability and energy use. To make ReRAM more suitable for the main memory system, we have to mitigate the IR drop problem, address the write disturbance, leverage the nonuniform access latency in ReRAM memory banks, and optimize the data patterns involved in a write operation. In this article, we propose a novel **C**ircuit-**A**rchitecture **C**o-optimization **F**ramework **(CACF)** for improving the performance, reliability, and energy consumption of a ReRAM-based main memory system. The contributions of this article include:

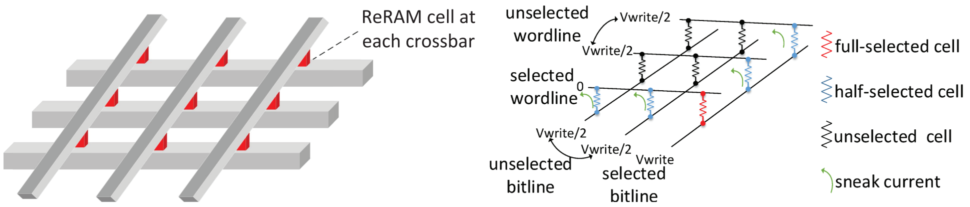
- At the circuit level, we propose a Double-Sided Write Driver (DSWD) design to reduce the IR drops along bitlines by applying write drivers at both sides of bitlines and selectively activating the write drivers.
- At the circuit architecture level, we analyze the differences between the RESET disturbance and SET disturbance. The analysis results show that it's nearly impossible for a SET operation to result in data corruption, and we only need to address the RESET disturbance to achieve high reliability and low overheads. Then we propose a RESET Disturbance Detection Scheme (RDDS) to address the write disturbance and build a compact probability model to show the effect of process variation on RESET disturbance.
- At the architecture level, we divide each 8-bit writing crossbar array into multiple regions according to the nonuniform access latency in ReRAM banks and then remap the hot data to fast regions and remap the cold data to slow regions, which efficiently reduces the access latency. Then we propose a 1-dominated flip scheme in the fast regions to further reduce the access latency and propose a 0-dominated flip scheme in the slow regions to improve the reliability and energy consumption of ReRAM arrays.

The remainder of this article is organized as follows. Section 2 describes the background of ReRAM and motivations of our design. Section 3 presents the DSWD circuit design. Section 4 describes the RDDS scheme at the circuit architecture level. Section 5 describes the region partition with an address remapping method and two flip schemes at the architecture level. Section 6 presents the CACF design. Section 7 shows the experimental results and hardware overheads. Section 8 describes the related work, and Section 9 concludes our work.

2 PRELIMINARIES AND MOTIVATION

2.1 ReRAM Cell Structure

A ReRAM cell has a very simple structure consisting of a metal-oxide layer sandwiched between a top metal electrode and a bottom metal electrode, as shown in Figure 1. The state of a ReRAM cell is represented by the resistance value of the cell. The resistance range of a ReRAM cell is divided into three regions, as shown in Figure 1(e). The High-Resistance State (HRS) and Low-Resistance State (LRS) are used to denote logic 0 and logic 1, respectively. Any resistance that falls in the marginal region is considered unreliable in distinguishing logic 0 and logic 1. By applying an external voltage with specified polarity, magnitude, and duration to ReRAM cells, the resistance



(a) (b)

Fig. 2. (a) Crossbar structure; (b) RESET operation in a crossbar array.

states of ReRAM cells can switch between HRS and LRS. For ReRAM cells, switching from LRS to HRS is defined as a RESET operation, and switching from HRS to LRS is defined as a SET operation. According to the switching behaviors, ReRAM can be classified into unipolar and bipolar ReRAMs. For a unipolar ReRAM cell, resistance switching is only related to the magnitude and duration of the external voltage applied to the cell, resulting in difficulty in controlling SET, RESET, and read operations. On the other hand, for a bipolar ReRAM cell, RESET and SET operations require different voltage polarities. In this work, we focus on the bipolar ReRAM cell because it has a better operating margin and it is more commonly used in crossbar memory. For the bipolar ReRAM cell, the material of the top electrode layer usually has the active characteristic, such as Ti, whereas the material of the bottom electrode layer is usually inactive, such as Pt. During a SET operation (Figure 1(a)), with positive voltage (V_{set}) applied on the top electrode layer, the oxygen ions drift to the anode layer and a lot of oxygen vacancies are left in the metal-oxide layer. Then the Conductive Filaments (CFs) are generated and the state of the ReRAM cell switches to LRS, which is shown in Figure 1(b). During a RESET operation (Figure 1(c)), with the negative voltage ($-V_{reset}$) applied on the top electrode layer, the oxygen ions are forced back to the metal-oxide layer and they combine with oxygen vacancies. Then the CFs are cut off and the state of the ReRAM cell switches to HRS, as shown in Figure 1(d).

2.2 ReRAM Array Structure

Generally, a ReRAM array structure can be classified into two types: a 1T1R grid structure and a crossbar structure. In the 1T1R grid structure, each cell has a dedicated MOSFET transistor and can be accessed independently without disturbance. However, since the size of a MOSFET transistor is typically much larger than that of an ReRAM cell, the total area of ReRAM arrays is primarily dominated by MOSFET transistors rather than ReRAM cells. Hence, these transistors tremendously increase the area and cost. For example, a 1T1R ReRAM prototype has been constructed with a cell size of $15F^2$ [45] and another prototype of HfOx-based ReRAM, which is also built with 1T1R structure, has a cell size of $9.5F^2$ [48]. In the crossbar structure (Figure 2(a)), all cells are interconnected to each other without transistors, and a cell only occupies an area of $4F^2$, which is the smallest theoretical size for a single-layer memory structure. Therefore, in a highly cost-conscious memory market, the crossbar structure is more suitable for ReRAM-based main memory.

To write (RESET and SET) a ReRAM cell in the crossbar array, the wordline and bitline connected to the cell should be selected with the proper potential. In addition, the unselected wordlines and bitlines are set to a certain voltage to avoid disturbing other cells in the array. A Half-Wordline Half-Bitline (HWHB) write scheme is widely adopted for the crossbar array [59, 60, 67]. With HWHB, when a RESET operation is performed, the selected bitline is applied with a full voltage V_{write}, the selected wordline is grounded, and all of the unselected wordlines and bitlines are half biased at $V_{write}/2$. As shown in Figure 2(b), the cell applied with full voltage is called a *full-selected* *cell*, while the other cells on the selected wordline or bitline are called *half-selected cells*. The cells on the unselected wordlines or bitlines are called *unselected cells*. Since there are no access transistors to isolate ReRAM cells in the crossbar structure, activating the selected wordline and bitline(s) will result in current flowing across half-selected cells. These currents are commonly referred to as *sneak currents*. The sneak currents aggravate the IR drop problem, which increases the access latency and reduces the reliability of ReRAM arrays. To decrease the sneak currents, many recent ReRAM prototypes employ a dedicated selector in each cell to improve the nonlinearity of ReRAM cells [23, 36]. In our work, we also model the crossbar ReRAM with a selector in each cell (the structure is also called 1S1R). When a SET operation is performed, the selected bitline is grounded, the selected wordline is biased at V_{write}, and all of the unselected wordlines and bitlines are half-biased at $V_{write}/2$. However, even with the HWHB write scheme, the sneak currents still exist due to the $V_{write}/2$ voltage across these half-selected cells. To read a cell in the crossbar array, the selected wordline is biased at V_{read} and all the other wordlines and bitlines are grounded. Then the current in the selected bitline is sent to the sense amplifiers to determine the value of the stored bit.

2.3 Write Disturbance

Actually, there are many fault types in crossbar ReRAM, such as retention failure, stuck-at-fault, write disturbance, and the like. The retention failure of ReRAM is an abrupt resistance drop of the HRS cell (HRS failure) or a sudden resistance increase of the LRS cell (LRS failure), which results from the random generation of oxygen vacancies and the recombination of oxygen vacancies with oxygen ions. The stuck-at-fault ReRAM is caused by excessive voltage applied on the selected cells, which results in endurance degradation. The retention failure and stuck-at-fault also appear in other memories, and many techniques have been proposed to solve these problems, such as Hamming code [13], BCH code [20], SEC-DED code [17], Error-Correcting-Pointer [46], Dynamically Replicated Memory [19], SAFER [47], and more. These techniques can potentially be applied in ReRAM to address retention failure and stuck-at-fault. However, the write disturbance in a crossbar ReRAM is much more serious than that in other memories because there are no access transistors to isolate ReRAM cells in the crossbar structure. In addition, the process variation of ReRAM is more complicated than that of other memories [69], which significantly alters the disturbing effect among ReRAM cells. Therefore, in this article, we focus on the write disturbance in crossbar ReRAM.

As described in Section 2.2, even with an HWHB write scheme, the half-selected cells are still biased at $V_{write}/2$, which can affect their resistances. Unfortunately, the effect is accumulative for a series of write operations and may eventually result in data corruption. This effect is referred to as *write disturbance*. Note that the effect can be disturbing or healing based on the written logic and the logic stored in the half-selected cells [11]. If a ReRAM cell R stores a logic 1, writing a logic 0 (logic 1) into one of the cells sharing the same wordline or bitline shifts R's resistance toward logic 0 (logic 1), weakening (strengthening) the stored logic of R. If R stores a logic 0, writing a logic 0 (logic 1) into one of the cells sharing the same wordline or bitline shifts R's resistance toward logic 0 (logic 1), strengthening (weakening) the stored logic of R. In other words, performing a RESET operation only disturbs the half-selected cells with logic 1 (called *RESET disturbance*), and performing a SET operation only disturbs the half-selected cells with logic 0 (called *SET disturbance*).

Figure 3 shows the differences between the RESET disturbance for the half-selected cells with logic 1 and the SET disturbance for the half-selected cells with logic 0. Note that the half-selected cells with logic 0 have extremely small currents due to their high resistances. Moreover, the SET latency is very short. The resistance switch of a ReRAM cell depends on the time integral of current across its terminals. Therefore, during a SET operation, the resistance switches of the half-selected cells with logic 0 are negligible. In other words, it's nearly impossible for a SET operation to result

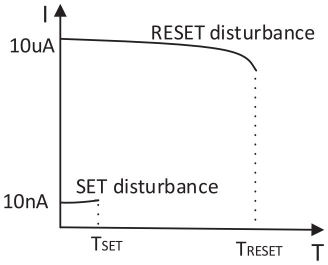


Fig. 3. RESET disturbance and SET disturbance.

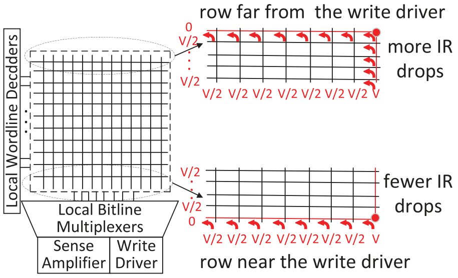

row far from the write driver
0
V/2
V/2
V/2 V/2 V/2 V/2 V/2 V/2 V/2 V
more IR drops
Local Wordline Decoders
V/2
V/2
0
V/2 V/2 V/2 V/2 V/2 V/2 V/2 V
fewer IR drops
Local Bitline Multiplexers
Sense Amplifier
Write Driver
row near the write driver

Fig. 4. The relationship of rows and IR drops in a crossbar array.

in data corruption. On the other hand, during a RESET operation, the currents of the half-selected cells with logic 1 are much larger and the RESET latency is also much longer. The RESET disturbance significantly switches the resistances of the half-selected cells with logic 1. Therefore, for high reliability and low overheads, we only need to address the RESET disturbance.

2.4 Motivation

ReRAM is an asymmetric memory whose write latency is much longer than read latency and RESET latency is much longer than SET latency. So the RESET operation becomes the performance bottleneck. In addition, the RESET latency of a ReRAM cell is exponentially inversely proportional to the voltage applied on the cell. The relationship between RESET latency t and the voltage drop across the cell V_d can be expressed as an equation: $t \times e^{kV_d} = C$, where k and C are fitting constants from experiments [29]. Therefore, the reduction of V_d resulting from the IR drop problem significantly increases RESET latency. More seriously, if the V_d is too small, the state of the ReRAM cell may fail to change, and it may cause a write failure. Increasing the output voltage of the write driver directly doesn't work because the $V_{write}/2$ applied on half-selected cells increases with the larger V_{write}, and the half-selected cells may suffer from more serious write disturbance. To construct an efficient and reliable ReRAM-based main memory system, we should keep the V_d large enough without corrupting the data stored in other cells.

The Double-Sided Ground Biasing (DSGB) design [60] is a state-of-the-art circuit design to decrease IR drops in crossbar ReRAM. DSGB applies another ground on the other side of the selected wordline, reducing the IR drops along wordlines. However, the IR drops along bitlines during the RESET operation are always ignored. In fact, the IR drops along bitlines account for a large proportion as the size of ReRAM arrays becomes larger. Therefore, it's urgent to design a ReRAM circuit to reduce IR drops along bitlines.

Moreover, the write disturbance always exists in ReRAM arrays and significantly reduces their reliability. However, in previous works, the write disturbance is not discussed or excessively considered. In these researches [54, 60, 65, 67], the write disturbance in ReRAM arrays results in suboptimal reliability. On the other hand, in Ghofrani et al. [11], the RESET disturbance and SET disturbance are equally treated. Additional storage and time overheads are required to address the SET disturbance. In this situation, the write disturbance is excessively considered, resulting in extremely high overheads. In fact, it's nearly impossible for a SET operation to result in data corruption, and we only need to address the RESET disturbance for the high reliability and low overheads.

Furthermore, in a crossbar array, rows near the write driver have lower wire resistances along bitlines than rows far from the write driver. This means that rows near the write driver have

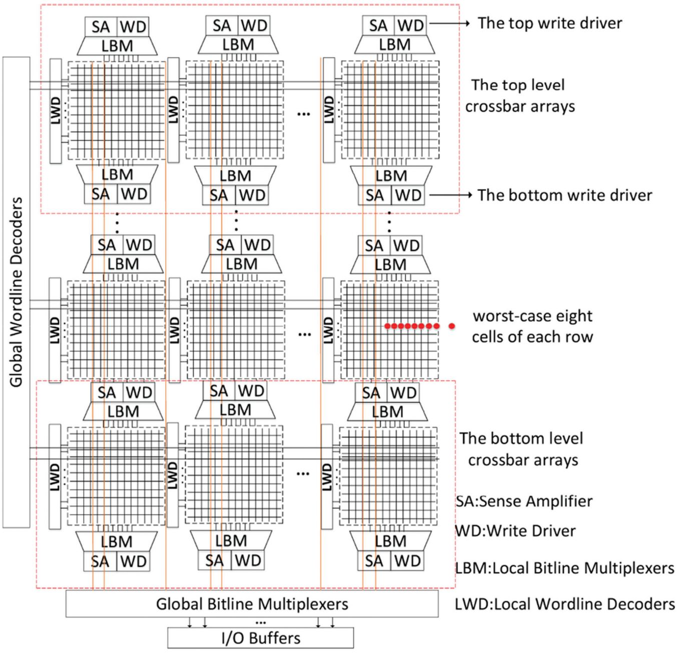

SA
WD
LBM
LWD
The top write driver
The top level crossbar arrays
The bottom write driver
Global Wordline Decoders
worst-case eight cells of each row
The bottom level crossbar arrays
SA:Sense Amplifier
WD:Write Driver
LBM:Local Bitline Multiplexers
LWD:Local Wordline Decoders
Global Bitline Multiplexers
I/O Buffers

Fig. 5. Schematic view of a ReRAM bank with DSWD scheme.

fewer IR drops than rows far from the write driver, as shown in Figure 4. In other words, rows near the write driver have shorter write latency, and rows far from the write driver have longer write latency, which causes nonuniform access latency in ReRAM memory banks. However, in most ReRAM designs, the write latency is pessimistically referred to the worst-case latency of the farthest cell from the write driver, which seriously decreases performance.

Finally, data patterns during a write operation have a significant influence on the access latency, reliability, and energy use of a crossbar array, especially for a multiple-bits writing crossbar array. Due to the nonlinearity of ReRAM cells, writing more 0s into the crossbar array significantly increases the access latency. For the sake of performance, we should ensure that fewer 0s are written into the crossbar array during a write operation. On the other hand, when the crossbar array writes more 0s, more ReRAM cells are in high resistance states and the sneak currents will be significantly reduced according to Ohm's Law, which improves the reliability and energy use of the crossbar array. However, previous designs have never systematically considered the data patterns for the access latency, reliability, and energy use of ReRAM arrays.

3 THE CIRCUIT LEVEL OPTIMIZATION

To reduce IR drops along bitlines, we propose a Double-Sided Write Driver (DSWD) circuit design for crossbar arrays. Different from the conventional design, DSWD applies additional write drivers and sense amplifiers for the top and bottom level crossbar arrays in ReRAM banks, as shown in Figure 5. DSWD guarantees that each crossbar array has the same number of write drivers at both sides of bitlines. These write drivers are called the top write drivers and the bottom write drivers according to their sides. During a RESET operation, if the selected wordline is in the upper half of the crossbar array, we enable the top write drivers. Otherwise, we enable the bottom write drivers. The options of write drivers can be implemented through a simple selection circuit. In the DSWD

Table 1. Parameters in the Crossbar Array Model

Metric	Description	Value
A	Mat size: A *wordlines*× A *bitlines*	1024
n	Number of bits to write in a mat	8
I_{on}	Cell current of a LRS ReRAM during RESET	20μA
R_{wire}	Wire resistance between adjacent cells	2.82Ω
K_r	Nonlinearity of the selector	3000
V_W	Full selected voltage during write	3.2V
V_R	Read voltage	1.6V
R_{RESET}	Resistance of a HRS cell	160$M\Omega$
R_{SET}	Resistance of a LRS cell	160$K\Omega$

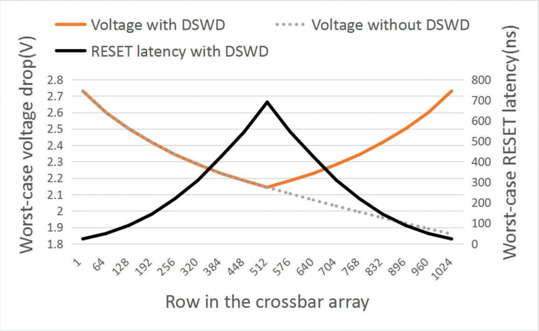


Fig. 6. The relationship of worst-case voltage drop and RESET latency at each row in a crossbar array with DSWD.

design, write drivers and sense amplifiers between adjacent crossbar arrays are shared, as in the conventional design. Therefore, just the top and bottom level crossbar arrays need additional write drivers and sense amplifiers in the DSWD design. In addition, data parallelism has been maintained through alternately activating different level crossbar arrays.

To quantitatively show the advantages brought by DSWD, we build a detailed circuit model for the 1024 × 1024 crossbar array following Kirchhoff's Current Law [41, 54, 59], and we model a 1Gb ×8 ReRAM chip architecture with a DDR3-compatible interface. A rank is composed of 8 banks, and each bank has 1024 mats (a mat is a 1024 × 1024 crossbar array), where the 1024 mats form a 32 × 32 matrix. In our work, we take the 8-bit writing scheme as an example rather than a single-bit writing scheme because the 8-bit writing scheme is more energy efficient [60]. Table 1 shows the key parameters obtained from HfOx-based cells [29] and IBM's MIEC device [2]. The latency of the worst-case eight cells which are furthest from the row decoder is measured as the RESET latency of each row. The worst-case eight cells of each row are shown in Figure 5. The RESET latency of the worst-case eight cells can be classified into eight categories according to the number of 0s written into the row. When all the worst-case eight cells write 0s, the RESET latency is the worst. To verify the DSWD design, we perform a RESET operation for the worst-case eight cells of each row in our model. The relationship of voltage drops and the worst-case RESET latency of each row in the crossbar array is shown in Figure 6. The results show that rows over 512 in the DSWD design have much larger voltage drops than the conventional design (without DSWD). In the conventional design, the worst-case row is row 1024 and the worst-case voltage drop is too small to perform a reliable write operation for a 1024 × 1024 crossbar array. However, with the

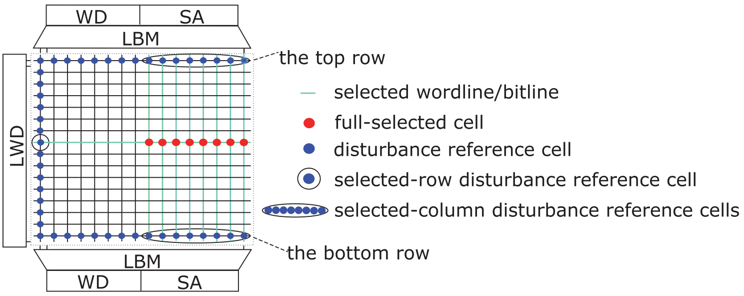

WD
SA
LBM
LWD
the top row
the bottom row
selected wordline/bitline
full-selected cell
disturbance reference cell
selected-row disturbance reference cell
selected-column disturbance reference cells

Fig. 7. Disturbance reference cells in RDDS scheme.

DSWD design, the worst-case row is row 512 and the length of the IR drop path along bitlines is significantly reduced for the rows over 512. Therefore, DSWD provides larger voltage drops for cells over row 512, improving the access latency and reliability of these cells.

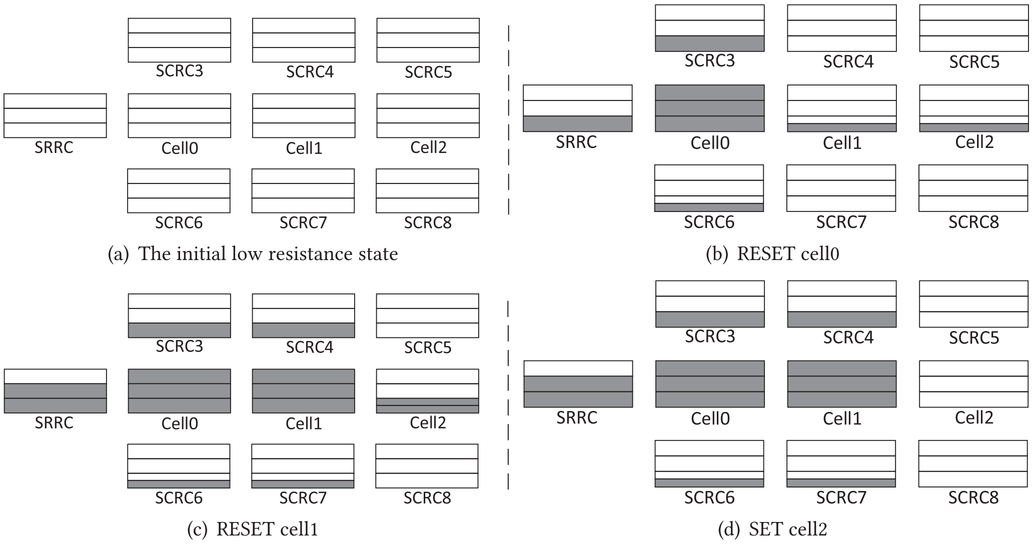
However, DSWD also brings hardware overheads because it requires additional write drivers and sense amplifiers for the top and bottom level crossbar arrays, thus reducing the area efficiency. Fortunately, only the write drivers at one side of bitlines are enabled at the same time in the DSWD design. DSWD doesn't cause additional energy consumption compared with the conventional design. Considering the great reduction in RESET latency, the additional overhead is completely acceptable.

4 THE CIRCUIT-ARCHITECTURE LEVEL OPTIMIZATION

4.1 RESET Disturbance Detection Scheme

As analyzed in Section 2.3, it's nearly impossible for a SET operation to result in data corruption. In order to improve reliability with low overheads, we only need to address the RESET disturbance, and we propose a RESET disturbance detection scheme called RDDS. In RDDS, we add the disturbance reference cells to detect the RESET disturbance. In a crossbar array, we select the cells on the first column and the cells on the top and bottom rows as the disturbance reference cells. During a RESET operation, the disturbance reference cell on the selected row is called a *Selected-Row Disturbance Reference Cell* (SRRC), and the disturbance reference cells on the selected columns are called *Selected-Column Disturbance Reference Cells* (SCRCs), as shown in Figure 7. Based on the DSWD design, if the selected cells are in the upper half of the crossbar array, the SCRCs on the top row suffer the worst-case sneak currents and write disturbance on these selected columns. Otherwise, the SCRCs on the bottom row suffer the worst-case sneak currents and write disturbance on these selected columns. That's why we select the cells on both top and bottom rows as the disturbance reference cells in our design.

The disturbance reference cells are ordinary ReRAM cells, and the only difference is that the user can't write these cells. In the RDDS scheme, the disturbance reference cells are initially set to logic 1, because the RESET operation only disturbs those cells with logic 1. The disturbance reference cells have two features. First, as their correct binary data are always logic 1, detecting the RESET disturbance for them becomes feasible. Second, a RESET operation disturbs them in the same way as it disturbs other half-selected cells with logic 1. It's worth noting that the SRRC and the SCRCs suffer the worst-case RESET disturbance during a RESET operation because they suffer the largest sneak currents and they are never written into logic 1 to recover to their initial resistance states. Unlike them, other half-selected cells with logic 1 may have been written into logic 1 through SET operations, which offsets the accumulated RESET disturbance. Therefore, as



(a) The initial low resistance state

(b) RESET cell0

(c) RESET cell1

(d) SET cell2

Fig. 8. The principle of the RESET disturbance detection. (a) The initial resistance states of ReRAM cells with logic 1 (the white color represents low resistance). (b) RESET cell0 (suppose that cell0, cell1, and cell2 are in the upper half of a crossbar array and on the same row). The gray area represents the resistance increment. Cell0's resistance increases to R_{HRS} after the RESET operation. The SRRC and SCRC3 suffer the worst-case RESET disturbance. Cell1, cell2, and SCRC6 suffer relatively small RESET disturbance. (c) RESET cell1. Cell1's resistance increases to R_{HRS} after the RESET operation. As cell0's resistance is already R_{HRS}, the resistance state of cell0 is not changed. The SRRC and SCRC4 suffer the worst-case RESET disturbance. Cell2 and the SCRC7 suffer relatively small RESET disturbance. The RESET disturbance for cell2 and the SRRC is accumulative. (d) SET cell2. Cell2's resistance recovers to R_{LRS} after the SET operation. We can observe that the SRRC always suffers the worst-case RESET disturbance on the selected row and the SCRCs always suffer the worst-case RESET disturbance on the selected columns.

long as the resistances of the SRRC and SCRCs stay in the correct range, the validity of the data stored in other half-selected cells with logic 1 can be guaranteed. The idea is illustrated in Figure 8.

To avoid data corruption caused by the RESET disturbance, RDDS reads out the resistances of the SRRC and the SCRCs when a RESET operation is done. We use the high resolution comparators to read out the resistances of the SRRC and the SCRCs rather than their binary data, as shown in Figure 9. RDDS compares the voltage of sense amplifiers (the voltage is inversely proportional to the resistance of the SRRC or SCRC, called V_{in}) with the reference voltage (V_{ref}) which corresponds to the R_{LRS} threshold. The comparison process can be presented as an expression:

$$readout = \begin{cases} V_{in} - V_{ref} > \lambda & \text{no refresh} \\ 0 < V_{in} - V_{ref} <= \lambda & \text{refresh,} \end{cases}$$

where λ is a constant no smaller than the precision of the comparator.

In the RDDS scheme, when the SRRC is read and the readout of the comparator is no larger than λ, it means that the resistance state of the SRRC will be switched and the stored logic may be corrupted. To guarantee the validity of the data stored in half-selected cells, RDDS refreshes the whole LRS cells on the selected row through SET operations. Similarly, when the SCRCs are read and the readout of the comparator corresponding to a column is no larger than λ, all LRS cells on this column will be refreshed through SET operations. Therefore, the data corruption resulting from the RESET disturbance can be avoided through conditional refresh operations.

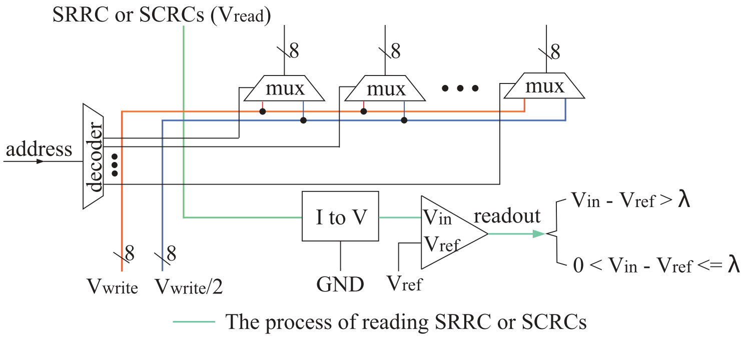


Fig. 9. The circuitry to read out the resistances of the SRRC and the SCRCs.

Although RDDS addresses the RESET disturbance and improves the reliability of ReRAM arrays, it also causes additional read and refresh operations. Fortunately, in the RDDS scheme, the SRRC can be read through the half-biased voltage ($V_{write}/2$) during the RESET operation, saving read time and energy. Only two read operations are required to read the SCRCs on the top and bottom rows. In addition, the refresh operation in RDDS scheme is actually the SET operation and the SET operation is very fast, which causes low time overhead. In addition, the disturbance reference cells only cause 0.3% storage overhead. Compared with the conventional design, RDDS doesn't waste storage and time overheads addressing the SET disturbance. Therefore, RDDS addresses the write disturbance with extremely low overheads.

4.2 The Effect of Process Variation on RESET Disturbance

In our design, the number of consecutive RESET operations before corrupting the data of the half-selected cells is defined as the *RESET Disturbance Tolerance* (RDT). The refresh frequency in the RDDS scheme is determined by the smallest RDT and λ, and λ corresponds to the smallest RDT. If λ is too small, the refresh operation may not be triggered. If λ is too large, the refresh operation may be triggered too early, which increases the overheads of refresh operations. Note that without the disturbing effect of process variation, the cell on column 1 and row 512 suffers the worst-case RESET disturbance because its biased voltage is close to $V_{write}/2$ and the RESET latency of this row is the longest. Thus, the RDT of the cell is the smallest. To derive the smallest RDT, we continuously perform RESET operations on row 512 in our ReRAM circuit model. Then we compare the resistance of the cell on column 1 and row 512 with the R_{LRS} threshold and acquire the smallest RDT. The results from our ReRAM circuit model show that without the disturbing effect of process variation, the smallest RDT is 245 and λ should be 77μV in the RDDS scheme.

However, due to process variation, the disturbing effect may differ among ReRAM cells, and the cell suffering the worst-case RESET disturbance may vary. In other words, considering process variation, the data of other half-selected cells may be corrupted before the data corruption of the disturbance reference cells. Our solution is to properly enlarge λ to trigger the refresh operation earlier in the RDDS scheme, which covers the disturbing effect of process variation. To show the effect of process variation on RESET disturbance, we build a compact probability model based on the cell-array interaction. We adopt the gap distance (l) to represent the state of a ReRAM cell and choose the increment of the gap distance (Δl) as a random variable in our model, which is similar to other works [12, 31]. For a RESET operation, Δl obeys lognormal distribution [31, 34] ($\ln(\Delta l) \sim N(\mu, \sigma^2)$). Then we get the expression $\Delta l = Ca^X$, where X follows the standard normal distribution, $C = e^\mu$ and $a = e^\sigma$. Moreover, C depends on the amplitude (A) and width (Δt) of the

Table 2. Parameters of Monte Carlo Simulations

Parameter	Typical Value
C_1	$7.62e-7$
k_1	0.71
k_2	0.2
b	1
Δt	$690ns$
α	0.9987

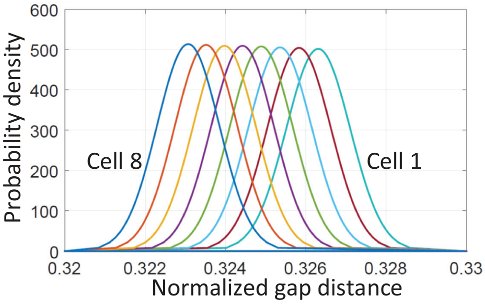
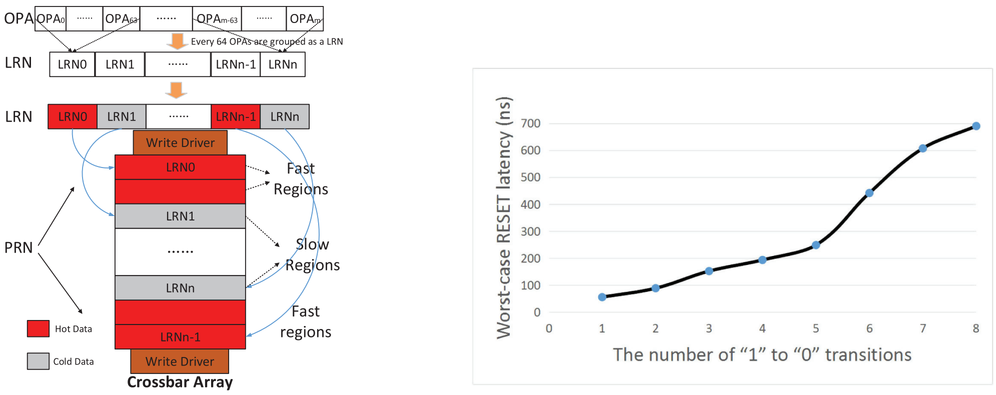


Fig. 10. Accumulative RESET disturbance distributions of the eight cells near the row decoder (Cell1 to Cell8) on row 512 under the process variation.

voltage pulse applied on the ReRAM cell [32, 63], and a is proportional to the gap distance (l) [10]. Therefore, we can also get two expressions: $C = C_1 e^{k_1 A} \Delta t$ and $a = k_2 l + b$, where C_1, k_1, k_2, and b are fitting constants. Furthermore, the accumulative RESET disturbance of the cell j (Δl_{sum}^j) can be expressed as $\Delta l_{sum}^j = \sum_{i=1}^n \Delta l_i^j$, where i and n represent the RESET disturbance cycles. To simplify the calculations in our model, we assume that the waveform of the voltage applied on a given ReRAM cell is the same between cycles. We use Monte Carlo simulations to estimate the accumulative RESET disturbance distributions.

Each half-selected cell has an accumulative RESET disturbance distribution. We assume M_j is the α-quantile of the cell j, and the relation between M_j and α is shown in the expression $P(\Delta l_{sum}^j < M_j) = \alpha$, where P represents the probability that the accumulative RESET disturbance of the cell j is smaller than M_j. If α is very close to 100%, it can be assumed that the accumulative RESET disturbance will never exceed M_j. Given α, the α-quantile of each half-selected cell can be estimated. We run Monte Carlo simulations with the key parameters in Table 2. We take the cells on row 512 as the target because the RESET latency of this row is the longest and the disturbing effect of process variation on this row is the most serious. According to the distance to the row decoder, the cells on row 512 are respectively called Cell1, Cell2, Cell3, and so on. Figure 10 shows the accumulative RESET disturbance distributions of the eight cells near the row decoder (Cell1 to Cell8) on row 512 under the process variation. We argue that only these eight cells may suffer the worst-case RESET disturbance due to the extremely low probability of other cells having a larger accumulative RESET disturbance than Cell 1, such as 0.04% for Cell 9. The experimental results from Monte Carlo simulations show that, within 228 RESET disturbance cycles, the probability for Cell 1 to have the largest accumulative RESET disturbance is very close to 100%. In other words, considering process variation, the smallest RDT in the RDDS scheme decreases to 228. Then we



(a) (b)

Fig. 11. (a) Region partition and address remapping in a crossbar array with DSWD (b)The relationship of worst-case RESET latency and the number of "1" to "0" transitions in a 1024 × 1024 crossbar array with DSWD.

calculate λ in our ReRAM circuit model. The results show that λ should be enlarged from 77μV to 82.7μV in the RDDS scheme to cover the effect of process variation on RESET disturbance.

5 THE ARCHITECTURE-LEVEL OPTIMIZATION

5.1 Region Partition and Address Remapping

To leverage the nonuniform access latency in ReRAM banks, we divide a crossbar array into multiple regions. We gather every 32 contiguous rows as a region, and each crossbar array is divided into 32 regions. Although it's technically possible to assign a different latency to each region, we adopt two types of regions to simplify our design: *fast regions* and *slow regions,* according to their access latencies. We gather the rows near the write driver as the fast regions and the rows far from the write driver as the slow regions, as shown in Figure 11(a). The fast regions have shorter write latency, while the slow regions have longer write latency. The ratio of fast regions to slow regions is 1:3 in our design.

Considering the access hotspots in memory banks, we propose an efficient address remapping method to match the access hotspots with the nonuniform access latency in ReRAM banks. The address remapping method is implemented in the memory controller and it contains three stages: Original Physical Address (OPA) to Logic Region Number (LRN) to hot data identification and LRN to Physical Region Number (PRN), where OPA is the original physical address of a memory request before remapping. Figure 11(a) shows the detailed address remapping scheme.

1) OPA to LRN: We gather all 64 contiguous OPAs as a LRN. The data of a memory request are 64B, and the data of 64 OPAs are exactly 4KB. The data of a LRN are 4KB, and the data of a region in our design are also 4KB.

2) Hot data identification: We allocate a counter for each LRN to record the temperature according to its access frequency, where the read and write requests have the same weight. We calculate the temperature per time stamp, and the temperature of each LRN is divided by 2 after a time stamp. When the temperature of a LRN reaches the threshold, the data of the LRN are identified as hot data. Otherwise, the data of the LRN are identified as cold data.

3) LRN to PRN: We remap the LRN with hot data to the fast regions and remap the LRN with cold data to the slow regions. When the fast regions are full, we first migrate the data of fast regions

with the lowest temperature to slow regions in the same crossbar array and modify the address remapping table. After data migration in the same crossbar array, if the highest temperature of slow regions in a crossbar array is higher than the lowest temperature of fast regions in another crossbar array, the data migration between fast regions and slow regions needs to move bulk data in multiple crossbar arrays, and the address remapping table also needs to be modified. To support direct bulk data movements between crossbar arrays, our proposed ReRAM architecture links neighboring crossbar arrays' bitlines together with isolation transistors, similar to that in Change et al. [3].

5.2 1-Dominated Flip Scheme

Due to the nonlinearity of ReRAM cells, writing more 0s into a multiple-bits writing crossbar array significantly increases access latency. The number of "1" to "0" transitions (N_i) has an important influence on RESET latency, as shown in Figure 11(b). So it's necessary to reduce the N_i for better performance during a write operation. Flip schemes are commonly used in NVMs to decrease the write cost [38, 60]. In the basic flip scheme [60], the new data are flipped if the number of 0s in new data is more than half, which reduces the N_i for each write operation. However, in this scheme, the old data are completely ignored and the N_i is not minimum, resulting in suboptimal performance. CAFO [38] is a state-of-the-art flip scheme to reduce the cost of servicing write requests. Differing from row-only flip schemes, CAFO flips all the rows and columns that incur a positive gain from the cost model, which can minimize the N_i in a cache line. However, CAFO is not efficient in reducing the write latency of ReRAM arrays because all bits in a cache line are written into multiple crossbar arrays simultaneously, and the write latency is decided by the slowest crossbar array. Although CAFO can achieve the minimum N_i in multiple crossbar arrays, it can't guarantee the minimum N_i in each crossbar array. In addition, the storage overhead of CAFO is relatively high. Therefore, we abandon the previous flip schemes and propose a 1-dominated flip scheme (1-DFS) to reduce the N_i in this work. 1-DFS contains three phases: read, analysis, and write. The algorithm of 1-DFS scheme is similar to our previous work [67].

1) Read phase: 1-DFS leverages a *read-before-write* scheme to reduce the N_i. 1-DFS first reads out the old data and the flip flag bit $\{D', F'\}$.

2) Analysis phase: 1-DFS compares the new data and the default flip flag bit $\{D, F\}$ with $\{D', F'\}$ and then records the N_i and assigns the number to A. In addition, 1-DFS flips the new data and the default flip flag, and gets the flipped data and the new flip flag bit $\{D'', F''\}$. Then 1-DFS compares $\{D'', F''\}$ with $\{D', F',\}$ records the N_i, and assigns the number to B.

3) Write phase: If A is bigger than B, write $\{D'', F''\}$ into the crossbar array. Otherwise, write $\{D, F\}$ into the crossbar array.

5.3 0-Dominated Flip Scheme

The reliability and energy of a crossbar array are also significantly influenced by the data patterns involved in a write operation. When a crossbar array writes more 0s, more ReRAM cells are in high resistance states and fewer sneak currents are produced, according to Ohm's Law. The IR drop problem in the crossbar array will be also mitigated, and write/read failure may be avoided. Moreover, as previous research indicates, in a crossbar array of size 100×100, only about 1% of the total energy is consumed by the full-selected ReRAM cell being accessed and about 97% of the energy is dissipated by the LRS half-selected cells [26]. In other words, when the crossbar array has more HRS cells, energy use will be significantly reduced. Therefore, writing more 0s into the crossbar array can efficiently improve reliability and energy consumption.

CAFO [38] can also be applied in ReRAM to increase the number of 0s written into the crossbar array. There is no doubt that CAFO can achieve the most 0s in multiple crossbar arrays where a cache line is written. However, CAFO may decrease the number of 0s in a certain crossbar array

Table 3. Parameters of Simulation

Parameter	Value
CPU	4-Core, out of order, 3GHz, 192-entry recoder buffer, 8 issue width
L1 Cache	Private, 16KB I-cache, 16KB D-cache, 2-way assoc
L2 Cache	Shared, 16-way assoc, 4MB, 64B cache line, 20-cycle latency
main memory	4GB, DDR3-1333, 4channel, 1rank/channel, 8banks/rank
ReRAM Timing(ns)	tRCD(18), tCL(15), tCWD(13), tFAW(30), tWTR(7.5), tWR(-), tSET(10), $tRESET_{fast}(14.4, 18.7, 26.3, 31, 36.7, 57.6, 76.4, 90.6)$, $tRESET_{slow}(55.2, 88.3, 151.8, 193.2, 248.4, 441.6, 607.2, 690)$
DRAM Timing(ns)	tRCD(15), tCL(15), tCWD(13), tFAW(30), tWTR(7.5), tWR(15)
PCM Timing(ns)	tRCD(48), tCL(15), tCWD(13), tFAW(50), tWTR(7.5), tWR(300)

through row and column flips. In this case, CAFO may worsen the reliability and energy use of ReRAM arrays due to the nonlinearity of ReRAM cells. Considering the high overheads of CAFO, we abandon the CAFO flip scheme. In this work, we propose a 0-dominated flip scheme (0-DFS) to ensure more 0s are written into the crossbar array. 0-DFS only records the number of 0s in the new data. If the number of 0s in the new data is smaller than N/2, where N is the number of bits to write in a crossbar array, the new data will be flipped and written into the crossbar array with a flip flag bit "0". Otherwise, the new data will be directly written into the crossbar array with a flip flag bit "1". The algorithm of the 0-DFS scheme is similar to our previous work [67].

6 CACF DESIGN

As Section 5.1 explains, the hot data are written into the fast regions and the cold data are written into the slow regions for the sake of performance. To achieve better performance, higher reliability, and lower energy consumption, we propose a novel circuit architecture co-optimization framework for ReRAM-based main memory system, called CACF. CACF not only mitigates the IR drop problem by the DSWD circuit design, but also addresses the write disturbance through the RDDS scheme at the circuit architecture level. At the architecture level, CACF leverages the nonuniform access latency in ReRAM memory banks through the region partition with a address remapping method. Then, CACF optimizes the data patterns involved in a write operation to simultaneously improve the access latency, reliability, and energy of ReRAM arrays. CACF actually implements a 1-DFS scheme in the fast regions and implements a 0-DFS scheme in the slow regions. CACF further reduces the access latency of fast regions through the 1-DFS scheme and improves the reliability and energy use of crossbar arrays through the 0-DFS scheme in the slow regions. Considering that hot data in the fast regions are sensitive to access latency, CACF can clearly optimize access latency and maximally improve performance for ReRAM-based memory system. In addition, the cold data in the slow regions are not sensitive to access latency, and it is suitable to implement the 0-DFS scheme in the slow regions. Therefore, CACF can efficiently improve the performance, reliability, and energy use of ReRAM-based main memory system.

7 EXPERIMENTAL RESULTS

7.1 Experiment Setup

To evaluate our work, we use GEM5 [1] simulator with NVMain [43] as our simulation platform. Table 3 shows the detailed configurations of our simulator. Most ReRAM-related memory timing parameters are derived from previous work [60]. The RESET latencies of fast and slow regions are respectively classified into eight categories according to the number of "1" to "0" transitions. We

Table 4. Benchmark Characteristics of SPEC CPU2006

Benchmark	Description	RPKI	WPKI
zeusmp	Four copies of zeusmp	12.18	6.86
sjeng	Four copies of sjeng	8.61	8.50
mcf	Four copies of mcf	2.32	1.27
lbm	Four copies of lbm	9.80	9.27
libquantum	Four copies of libquantum	7.48	7.86
leslie3d	Four copies of leslie3d	5.82	8.85
bzip2	Four copies of bzip2	2.35	1.25
bwaves	Four copies of bwaves	12.15	11.5
astar	Four copies of astar	2.16	1.12
gobmk	Four copies of gobmk	1.76	1.52

use 10 workloads from the multiprogrammed SPEC CPU2006 benchmark suite [15] with different memory Read Per Kilo Instructions (RPKI) and memory Write Per Kilo Instructions (WPKI) rates, as shown in Table 4. We run all the benchmarks for 500 million instructions to warm up caches and then run 1 billion instructions for our proposed design. We select the DSGB design [60] as the aggressive baseline. To quantitatively compare our flip schemes with CAFO [38] and the basic flip scheme [60], we implement CAFO and the basic flip scheme in the fast regions to reduce the number of "1" to "0" transitions and also perform the two flip schemes in slow regions to increase the number of 0s written into crossbar arrays. The comparison configurations are as follows: *Baseline*: 8-bit writing crossbar arrays with the worst-case RESET latency under DSGB. *RPAR*: Apply the region partition with address remapping method under DSWD. *1-DFS*: Apply the 1-DFS scheme in the whole crossbar array based on *RPAR*. *LRR*: Apply the 1-DFS scheme in the fast regions and apply the 0-DFS scheme in the slow regions based on *RPAR*. *Basic flip*: Apply the basic flip scheme in the fast regions and slow regions based on *RPAR*. *CAFO*: Apply CAFO in the fast regions and slow regions based on *RPAR*. *CACF*: Apply the RDDS scheme based on *LRR*. To simulate the RDDS scheme, we add two read operations for each RESET operation and also add a refresh operation in our simulator when the number of RESET operations for a row or a column in a crossbar array reaches 228. To show the effects of different ratios of fast to slow regions on the same workload, we set different ratios of fast to slow regions for the same workload. The ratio of fast regions to all regions in a crossbar array is recorded as R in our experiments.

7.2 The Effect of R

To analyze the effects of different R values on the same workload, we set $R = 1/8$, $R = 1/4$, $R = 1/2$, and $R = 3/4$. Figures 12, 13, 14, and 15 show the average memory access latency, IPC speedup, running time, and energy consumption with different R values for the zeusmp benchmark, respectively. For the zeusmp benchmark, when R is 1/4, our methods achieve the best performance and the lowest energy consumption. When R is 1/8, the size of fast regions is too small to match the hot data for the zeusmp benchmark, which results in suboptimal performance. As R becomes larger, the size and access latency of fast regions become larger. When R is 1/2 or 3/4, the size of fast regions is too large to match the hot data for the zeusmp benchmark, which increases the access latency. In addition, the results show that the performance of the *LRR* is worse than the *RPAR* when R is 1/8 because the size of slow regions is too large, and the 0-DFS scheme in slow regions causes a great performance loss for the *LRR*. But as R becomes larger, the performance loss becomes smaller.

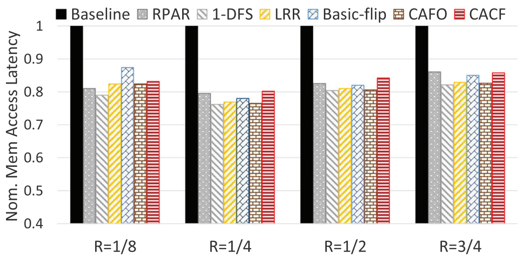


Fig. 12. The memory access latency with different R values for zeusmp benchmark.

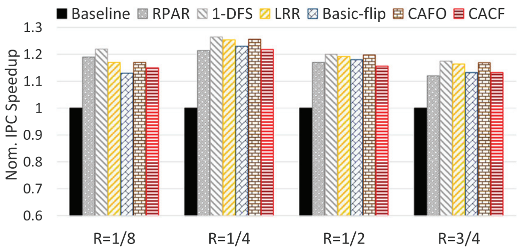


Fig. 13. The IPC speedup with different R values for zeusmp benchmark.

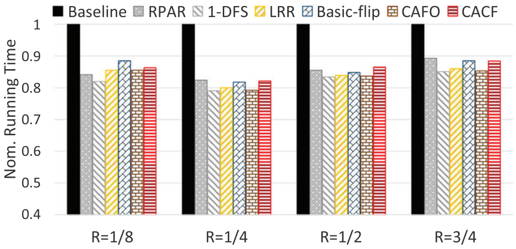


Fig. 14. The running time with different R values for zeusmp benchmark.

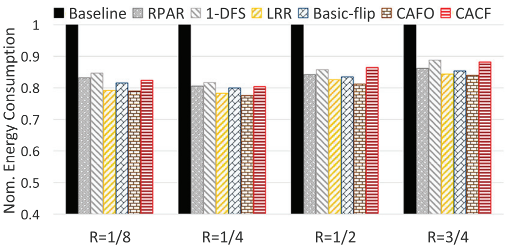


Fig. 15. The energy consumption with different R values for zeusmp benchmark.

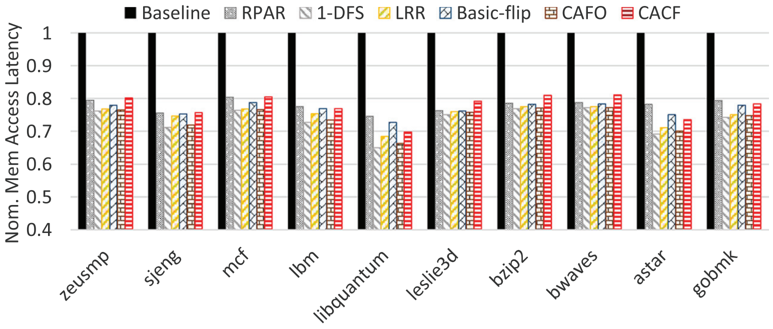


Fig. 16. The average memory access latency.

7.3 Memory Access Latency

To show system performance on nondeterministic workloads, we fix the value of R at 1/4 in our experiments. Figure 16 shows the average memory access latency for different workloads, and the results are normalized to the baseline. As shown in Figure 16, the proposed techniques can efficiently reduce memory access latency. *CACF* reduces access latency by 22.4% compared with the baseline. *1-DFS* achieves the lowest access latency because the whole crossbar array is optimal for it. The *LRR* has a 1.6% higher access latency than the *1-DFS* because the 0-DFS scheme in slow regions causes more “1” to “0” transitions. But the *LRR* makes the crossbar array more reliable. The *Basic flip* gets a 1.8% higher access latency than the *LRR* because our 1-DFS scheme achieves fewer “1” to “0” transitions compared with the basic flip scheme. Accordingly, our flip schemes can achieve lower access latency than the basic flip scheme with the same storage overhead (12.5%). Although *CAFO* shows 1% more access latency reduction compared with the *LRR*, CAFO doubles

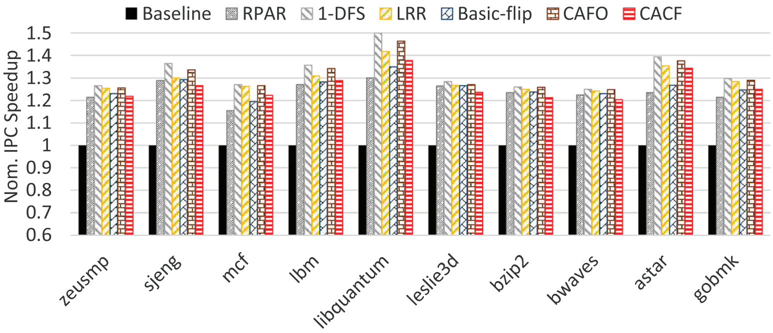


Fig. 17. The average IPC speedup.

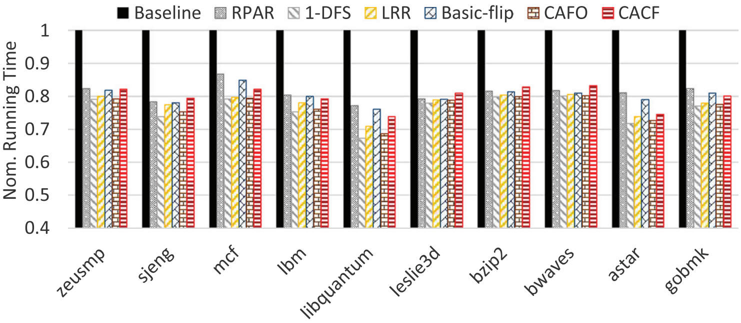


Fig. 18. The average running time.

the storage overhead (25%). Therefore, our flip schemes are more suitable for the ReRAM design. *LRR* gets 2.7% more access latency reduction than *CACF* because the RDDS scheme in *CACF* causes additional read and refresh operations. However, the *CACF* addresses the write disturbance and achieves higher reliability.

7.4 IPC Speedup

We use the Instructions Per Cycle (IPC) to evaluate system performance. Figure 17 shows the average IPC speedup for different design configurations. The results show that *CACF* achieves a 26.1% IPC improvement compared with the baseline. *1-DFS* achieves the highest IPC because the whole crossbar array is optimal for performance. *LRR* gets a 3.5% higher IPC improvement than the *Basic flip*. *CAFO* only achieves a 1.7% higher IPC improvement compared with *LRR*. But *CAFO* doubles the storage overhead. Although *CACF* achieves 3.3% lower IPC improvement compared with *LRR*, *CACF* achieves higher reliability by applying the RDDS scheme.

7.5 Running Time

The running time of workloads is one of the key metrics to measure whole-system performance. As our design can efficiently reduce access latency, the running time of workloads can be also shortened. Figure 18 shows the average running time for different design configurations. The results show that *CACF* reduces the running time by 20.1% compared with the baseline. *1-DFS* achieves a 1.6% greater running time reduction than *LRR* because the 0-DFS scheme in *LRR* increases access latency. But the 0-DFS scheme in *LRR* improves the reliability of ReRAM arrays. *Basic flip* and *CAFO* show 19.8% and 23.4% running time reductions, respectively, while *LRR* gets a 22.2% run-

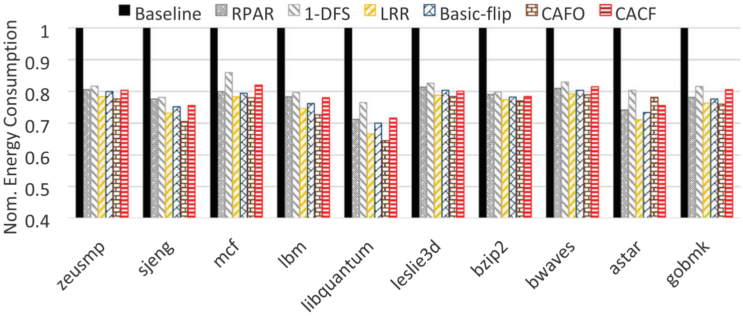


Fig. 19. The average energy consumption.

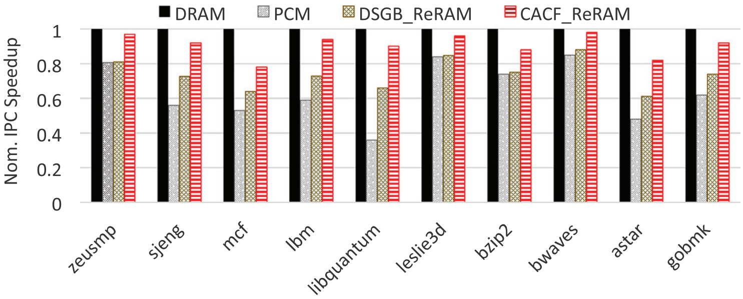


Fig. 20. Performance comparison among different memory technologies.

ning time reduction. *LRR* achieves a 2.1% greater running time reduction than *CACF* because the RDDS scheme in *CACF* increases access latency. But *LRR* has lower reliability without addressing the write disturbance.

7.6 Energy Consumption

The 0-DFS scheme ensures more 0s written into the crossbar array, which efficiently decreases sneak currents and energy consumption. In addition, reducing the access latency helps to decrease energy consumption. Figure 19 shows the average energy consumption for different design configurations. The results show that *CACF* reduces energy consumption by 21.6% compared with the baseline. *1-DFS* has the highest energy consumption because the 1-DFS scheme in the whole crossbar array causes more 1s to be written into the crossbar array and significantly increases the sneak currents. *Basic flip*, *LRR*, and *CAFO* achieve relatively low energy consumption because the flip schemes in slow regions increase the number of 0s and reduce sneak currents. The *Basic flip* achieves a 1.7% higher energy consumption compared with *LRR*, and *CAFO* only achieves a 0.2% lower energy consumption than *LRR*. Although *CACF* shows a 3% higher energy consumption than *LRR*, *CACF* is more reliable by addressing the write disturbance.

7.7 Comparison with DRAM and PCM

We compare the CACF ReRAM design (CACF_ReRAM) with the DDR3 DRAM and PCM designs in terms of performance. We also model a baseline ReRAM design (DSGB_ReRAM) [60] for comparison. The DRAM- and PCM-related memory timing parameters are derived from Micron's DDR3 technical specifications [8] and Lee's work [30], respectively. The detailed parameters of DRAM and PCM are shown in Table 3. Figure 20 shows the performance comparison among different memory technologies, and the results are normalized to the DRAM design. The results show that the average performance degradation for the CACF_ReRAM is only 9.3% compared with DRAM. However, PCM is much slower than DRAM, with an average performance degradation of 36.2%, which results from the long-latency SET operations of PCM. Both the DSGB_ReRAM and CACF_ReRAM outperform the PCM design. Therefore, relative to PCM, ReRAM can achieve much better performance, whereas, compared with DRAM, ReRAM can significantly benefit from its large capacity, low cost, and nonvolatility with an insignificant performance penalty.

7.8 Hardware Overhead

In the DSWD design, write drivers and sense amplifiers between adjacent crossbar arrays are shared, as shown in Figure 5. Only the top and bottom level crossbar arrays need additional write drivers and sense amplifiers. In other words, with the DSWD design, the area remains the same at the nanocrossbar layers and shared peripheral circuitry but differs at the top CMOS layers of the top-level crossbar arrays and the bottom CMOS layers of the bottom-level crossbar arrays in a ReRAM bank. To clearly show the area overhead of DSWD, we take the number of transistors as the unit of measure. In our design, a cross-coupled inverter latch-sense amplifier requires seven CMOS transistors per nanowire, and the voltage buffer needs four CMOS transistors per nanowire. In the conventional $M \times N$ crossbar ReRAM bank, the CMOS peripheral circuitry on bitlines costs $(7 + 4) \times N \times K \times L$ transistors, and the CMOS peripheral circuitry on wordlines costs $4 \times M \times P$ transistors, where K denotes the number of crossbar arrays in each level, L denotes the number of levels on which the write drivers and sense amplifiers are applied, and P represents the number of crossbar arrays in the ReRAM bank. In our ReRAM design, each bank has 1024 crossbar arrays with an array size of 1024×1024, and the 1024 crossbar arrays form a 32×32 matrix in a ReRAM bank. Therefore, without the DSWD design, the number of transistors utilized in the CMOS peripheral circuitry is $11 \times 1024 \times 32 \times 31 + 4 \times 1024 \times 1024$, whereas, in the DSWD design, $11 \times 1024 \times 32 \times 33 + 4 \times 1024 \times 1024$ transistors are required in the CMOS peripheral circuitry because DSWD only adds additional write drivers and sense amplifiers at two levels along bitlines. Therefore, DSWD only causes a 4.7% peripheral circuitry overhead. Furthermore, the number of ReRAM cells in a ReRAM bank is $1024 \times 1024 \times 1024$. Since the area of a CMOS transistor is twice that of a HfOx-based ReRAM cell, the area of ReRAM cells in a ReRAM bank is equal to the area of $1024 \times 1024 \times 1024 \times 1/2$ transistors, which is much larger than the area of the CMOS peripheral circuitry caused by DSWD. The area overhead of DSWD in a ReRAM bank is smaller by 0.14%, which is insignificant.

In the RDDS scheme, $1024 \times 3 - 2$ disturbance reference cells are needed, and the total number of ReRAM cells in a crossbar array is 1026×1025. Therefore, the disturbance reference cells cause a 0.3% storage overhead. In addition, the comparators used to read out the resistances of the disturbance reference cells cause some hardware overheads. In our design, a comparator detects at least 228 levels, and an 8-bit resolution comparator is required. The 8-bit resolution comparator costs 168 transistors [61], and 8 comparators are needed in each crossbar array. In a ReRAM bank, $168 \times 8 \times 1024$ transistors are required for the comparators in the RDDS scheme. The number of transistors in the other CMOS peripheral circuitry in a ReRAM bank is $11 \times 1024 \times 32 \times 33 + 4 \times 1024 \times 1024$. Therefore, the comparators in the RDDS scheme cause a 8.6% peripheral circuitry overhead.

In the region partition with address remapping method, the address remapping table causes storage overhead. In our design, a rank is 1GB and a PRN is 4KB. So there are 256K PRNs in a rank, and $log_2 256K \times 256K$ bits are required. An additional 14 bits are required to count the temperature for each LRN, and $14 \times 256K$ bits are needed. Therefore, for a 1GB memory rank, the region partition with address remapping method incurs 1MB storage overhead. In addition, to support direct bulk data movements between crossbar arrays in a ReRAM bank, $1024 \times 32 \times 33$ isolation transistors are required to link neighboring crossbar arrays' bitlines together, which causes a 6.7% peripheral circuitry overhead. In the 1-DFS and 0-DFS schemes, every 8 bits require a flip flag bit, causing a 12.5% storage overhead.

8 RELATED WORK

8.1 Related Circuit-Level Optimization in Memory Technologies

Circuit level optimization usually brings great benefits for memory technologies, and numerous works focus on circuit-level optimization. Talati et al. [53] proposed a novel ReRAM design called *transpose memory* to improve the logic functionality of crossbars by applying sense amplifiers on both wordlines and bitlines. Different from the transpose memory, DSWD is used to reduce the IR drops along bitlines by adding additional write drivers and sense amplifiers on the other side of bitlines. Therefore, both the structures and functions of transpose memory and DSWD are different. In addition, in transpose memory, to access the array from different directions simultaneously, all the peripheral circuitry of the selected crossbar array needs to be activated, which increases energy consumption compared with the conventional design. In DSWD, only the write drivers on one side of the bitlines of the selected crossbar array are activated, and no additional energy consumption is caused compared with the conventional design. Xu et al. [60] proposed the DSGB ReRAM design by applying another ground on the other side of the selected wordline. DSGB reduces the IR drops along the selected wordline and improves the worst-case RESET latency. This is different from DSWD, which reduces the IR drops along bitlines. Transpose memory and DSGB can be well applied in ReRAM in conjunction with DSWD. Zhao et al. [68] mitigated the IR drop problem in the 1TnR structure by reorganizing the peripheral circuitry and changing the directions of wordlines and bitlines. Shevgoor et al. [49] optimized the read operation in crossbar arrays through a novel sample and hold circuit. Different from these two works, our proposed techniques aim to optimize the write operation in the crossbar structure.

8.2 Leveraging the Asymmetric Access Characteristics in Memory Technologies

The region partition with address remapping method is motivated by the asymmetric access characteristics in ReRAM banks. By remapping hot data to fast regions, our method leverages the nonuniform access latency in ReRAM banks and improves the performance of ReRAM-based memory systems. Based on the observation that programming different states into an MLC ReRAM cell costs different latency and energy, Niu et al. [42] proposed Incomplete Data Mapping (IDM). By eliminating certain high-latency and high-energy states, IDM improves the performance and energy use of MLC ReRAM. But this mechanism can't be applied in our design because our proposed techniques are based on the SLC ReRAM.

In other memory technologies, there are many works improving performance by leveraging the asymmetric access characteristics. The asymmetric access DRAM has been proposed by various works [3, 28, 37, 50]. CHARM [50] introduced the asymmetric-subarray DRAM by designing some fast banks with shorter bitlines for faster data sensing and closer placement to the chip I/O for faster data transfers. Based on the asymmetric-subarray DRAM, Lu et al. [37] achieved lightweight row migration by exploiting shared sense amplifiers between neighboring subarrays. Lee et al. [28] proposed the asymmetric access TL-DRAM by splitting each long bitline into fast and slow segments through an isolation transistor. LISA [3] links neighboring subarrays' bitlines together with isolation transistors to make data movement across subarrays fast and energy-efficient. In these works, by allocating hot data to fast subarrays/rows, the asymmetric access DRAM significantly improves performance. Although the asymmetric access DRAM resembles our techniques, the two are fundamentally different. The asymmetric access DRAM requires additional manufacturing processes, while the asymmetric access characteristics in ReRAM are inherent due to the IR drops. In addition, the asymmetric access DRAM maps frequently accessed pages to the fast subarrays/rows using the OS or uses the fast subarrays/rows as a cache, which can't adapt to the dynamic changes in the hot dataset or loses the total capacity. Different from these works, our techniques introduce heterogeneous timing parameters into the memory controller and offer better adaptivity to dynamic changes in the hot dataset. Chang et al. [4] exploited the latency variation of DRAM caused by irregularity in the manufacturing process. Yoon et al. [62] proposed an asymmetric page mapping method for MLC PCM to map read-intensive pages to a fast-read/slow-write (FR) region and write-intensive pages to a slow-read/fast-write (FW) region. Different from these studies, our proposed techniques focus on the latency variation caused by the IR drop problem in the SLC ReRAM.

8.3 Flip Schemes in Memory Technologies

Flip schemes are commonly used in memory technologies to reduce write costs. Xu et al. [60] proposed a basic flip scheme for the multiple-bits writing crossbar array to reduce the number of "1" to "0" transitions during a write operation. In this scheme, the new data are flipped if the number of 0s in new data is more than half. However, the old data are ignored in this scheme and the number of "1" to "0" transitions is not minimum, resulting in suboptimal performance. Different from the basic flip scheme, 1-DFS takes the old data into account and achieves fewer "1" to "0" transitions. Maddah et al. proposed [38] a state-of-the-art flip scheme (CAFO) to minimize the write cost by flipping all the rows and columns that incur a positive gain from the cost model. Different from CAFO, 1-DFS is a row-only flip scheme, and the storage overhead of 1-DFS is half that of CAFO. In addition, CAFO is not much more effective than our 1-DFS scheme in reducing the write latency of ReRAM arrays because all bits in a cache line are written into multiple crossbar arrays simultaneously, and the write latency of a cache line is determined by the slowest crossbar array. Although CAFO can achieve the minimum "1" to "0" transitions in multiple crossbar arrays, it can't guarantee the minimum "1" to "0" transitions in each crossbar array.

Other flip schemes have been proposed to reduce the write cost of PCM. Flip-N-Write [6] flips the new data if the number of different bits is more than half compared with the old data. Two-stage-write [64] takes the PCM write properties into account based on Flip-N-Write. Three-stage-write [33] combines Flip-N-Write with two-stage-write to further reduce PCM write latency. Flip-Min [21] encodes each possible input data vector into 256 different vectors through an inverted (72, 64) Reed Muller code and selects the vector with the least number of bit flips to write. Han et al. [14] introduced Flip-N-Write, Flip-Min, and CAFO flip schemes and compared the performance of these flip schemes in terms of bit flip-reduction rate and lifetime improvement. All these works only use a kind of flip scheme in the whole memory array to reduce the write cost. Different from these works, our techniques apply different flip schemes in suitable regions of the memory array to optimize the data patterns, which simultaneously improves the performance, reliability, and energy use of ReRAM arrays.

8.4 Related Reliability Improvement in Memory Technologies

The memory reliability is very important, and many works focus on reliability improvement in memory technologies. Ghofrani et al. [11] proposed a solution to the write disturbance of ReRAM by applying asymmetric voltages for disturbance confinement and restoring weakened data on wordlines. However, the solution causes extremely high overheads because applying asymmetric voltages for crossbar arrays is not energy efficient, and solving the SET disturbance causes unnecessary storage and time overheads. Different from this solution, RDDS only addresses the RESET disturbance in crossbar arrays with symmetric voltages, which achieves high reliability and low overheads. In addition, RDDS detects the write disturbance on both wordlines and bitlines, providing higher reliability for ReRAM arrays.

Other works have introduced ways to address other fault types of ReRAM. Chen et al. [5] identified the key parameters controlling retention in HfO_2 ReRAM cells and achieved significant improvement in retention by applying a thermal budget to process flow. Xu et al. [59] proposed an error-resilient architecture by using ECC to correct retention failure and applying hard-error tolerating technique (such as ECP [46] or DRM [19]) to address stuck-at-fault. Zheng et al. [69] proposed a multiple-program-and-verify method to detect the pseudo-hard error and then recovered the error by increasing the voltage amplitude or the write pulse width. These mechanisms address other fault types of ReRAM and can be applied in conjunction with our techniques.

Other memory technologies also suffer from reliability problems, and many techniques have been proposed to improve reliability. Wen et al. [55] proposed a content-dependent ECC (CD-ECC) in STT-RAM to achieve balanced error correction in both bit-flipping directions. Seong et al. [47] proposed SAFER to recover a multi-bit stuck-at-fault error. By dynamically partitioning a data block and ensuring at most one failed bit in each partition, SAFER recovers multi-bit stuck-at-fault errors with single bit error correction techniques. These works and others [19, 46] primarily focus on addressing retention failure and stuck-at-fault in PCM and STT-RAM. Both can potentially be used in ReRAM in conjunction with our techniques.

9 CONCLUSION

In this article, we propose a novel circuit architecture co-optimization framework (CACF) for improving the performance, reliability, and energy consumption of ReRAM-based main memory systems. At the circuit level, the double-sided write driver design is proposed to mitigate the IR drop problem of the crossbar structure. At the circuit architecture level, a RESET disturbance detection scheme is presented to address the write disturbance for the high reliability and low overheads. At the architecture level, the region partition with address remapping method is proposed to leverage the nonuniform access latency in ReRAM banks. We also propose a 1-dominated flip scheme in the fast regions to reduce access latency and propose a 0-dominated flip scheme in the slow regions to improve reliability and energy use of ReRAM arrays. The experimental results show that CACF achieves 26.1% system performance improvement, 22.4% memory access latency reduction, 20.1% running time decrease, and 21.6% energy consumption reduction on average over an aggressive baseline. Meanwhile, CACF significantly improves the reliability of ReRAM-based memory system.

ACKNOWLEDGMENTS

We thank the anonymous reviewers and editors for their constructive comments and suggestions. We also thank anyone who helped us improve this work.

REFERENCES

[1] Nathan Binkert, Bradford Beckmann, Gabriel Black, Steven K. Reinhardt, Ali Saidi, Arkaprava Basu, Joel Hestness, Derek R. Hower, Tushar Krishna, and Somayeh Sardashti. 2011. The gem5 simulator. *ACM SIGARCH Comput. Archit. News* 39, 2 (2011), 1–7.

[2] G. W. Burr, K. Virwani, R. S. Shenoy, A. Padilla, M. BrightSky, E. A. Joseph, M. Lofaro, A. J. Kellock, R. S. King, and K. Nguyen. 2012. Large-scale (512kbit) integration of multilayer-ready access-devices based on mixed-ionic-electronic-conduction (MIEC) at 100% yield. In *IEEE Symposium on VLSI Technology (VLSIT)*. 41–42.

[3] Kevin K. Chang, Prashant J. Nair, Donghyuk Lee, Saugata Ghose, Moinuddin K. Qureshi, and Onur Mutlu. 2016. Low-cost inter-linked subarrays (LISA): Enabling fast inter-subarray data movement in DRAM. In *IEEE International Symposium on High Performance Computer Architecture (HPCA)*. 568–580.

[4] Kevin K. Chang, Abhijith Kashyap, Hasan Hassan, Saugata Ghose, Kevin Hsieh, Donghyuk Lee, Tianshi Li, Gennady Pekhimenko, Samira Khan, and Onur Mutlu. 2016. Understanding latency variation in modern DRAM chips: Experimental characterization, analysis, and optimization. *ACM SIGMETRICS Performance Evaluation Review* 44, 1 (2016), 323–336.

[5] Yang Yin Chen, Masanori Komura, Robin Degraeve, Bogdan Govoreanu, Ludovic Goux, Andrea Fantini, Naga Raghavan, Sergiu Clima, Leqi Zhang, and Attilio Belmonte. 2013. Improvement of data retention in HfO_2/Hf 1T1R RRAM cell under low operating current. In *IEEE International Electron Devices Meeting (IEDM)*. 1–10.

[6] Sangyeun Cho and Hyunjin Lee. 2009. Flip-N-Write: A simple deterministic technique to improve PRAM write performance, energy and endurance. In *Proceedings of the 42nd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*. 347–357.

[7] Leon Chua. 1971. Memristor-the missing circuit element. *IEEE Transactions on Circuit Theory* 18, 5 (1971), 507–519.

[8] Micron Corp. 2017. Micron DDR3 SDRAM Data Sheet. Retrieved from http://www.micron.com/products/dram/ddr3-sdram.

[9] Richard Fackenthal, Makoto Kitagawa, Wataru Otsuka, Kirk Prall, Duane Mills, Keiichi Tsutsui, Jahanshir Javanifard, Kerry Tedrow, Tomohito Tsushima, and Yoshiyuki Shibahara. 2014. A 16Gb ReRAM with 200MB/s write and 1GB/s read in 27nm technology. In *IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*. 338–339.

[10] Bin Gao, Yingjie Bi, Hong-Yu Chen, Rui Liu, Peng Huang, Bing Chen, Lifeng Liu, Xiaoyan Liu, Shimeng Yu, and H-S Philip Wong. 2014. Ultra-low-energy three-dimensional oxide-based electronic synapses for implementation of robust high-accuracy neuromorphic computation systems. *ACS Nano* 8, 7 (2014), 6998–7004.

[11] Amirali Ghofrani, Miguel Angel Lastras-Montaño, and Kwang-Ting Cheng. 2013. Towards data reliable crossbar-based memristive memories. In *IEEE International Test Conference (ITC)*. 1–10.

[12] Ximeng Guan, Shimeng Yu, and H-S Philip Wong. 2012. A SPICE compact model of metal oxide resistive switching memory with variations. *IEEE Electron Device Lett.* 33, 10 (2012), 1405–1407.

[13] Richard W. Hamming. 1950. Error detecting and error correcting codes. *Bell Labs Tech. J.* (1950), 147–160.

[14] Miseon Han and Youngsun Han. 2016. Bit flip reduction schemes for improving PCM lifetime: A Survey. *IEIE Transactions on Smart Processing and Computing* 5, 5 (2016), 337–345.

[15] John L. Henning. 2007. Performance counters and development of SPEC CPU2006. *ACM SIGARCH Comput. Archit. News* 35, 1 (2007), 118–121.

[16] HP and SanDisk. 2014. The Memristor Project. Retrieved from http://www.businessweek.com/articles/2014-06-11/with-the-machine-hp-may-have-invented-a-new-kind-of-computer.

[17] Mu-Yue Hsiao. 1970. A class of optimal minimum odd-weight-column SEC-DED codes. *IBM J. Res. Dev.* 14, 4 (1970), 395–401.

[18] Intel. 2016. Intel 3D XPoint Unveiled-The Next Breakthrough in Memory Technology. Retrieved from http://www.intel.com/content/www/us/en/architecture-and-technology/3d-xpoint-unveiled-video.html.

[19] Engin Ipek, Thomas Moscibroda, and others. 2010. Dynamically replicated memory: Building reliable systems from nanoscale resistive memories. *ACM SIGARCH Computer Architecture News*, Vol. 38. ACM, 3–14.

[20] Todd K. Moon. 2005. Error correction coding: Mathematical methods and algorithms. *Ltd* 750, 2 (2005), 750.

[21] Adam N. Jacobvitz, Robert Calderbank, and Daniel J. Sorin. 2013. Coset coding to extend the lifetime of memory. In *19th IEEE International Symposium on High Performance Computer Architecture (HPCA)*. 222–233.

[22] Myoungsoo Jung, John Shalf, and Mahmut Kandemir. 2013. Design of a large-scale storage-class RRAM system. In *Proceedings of the 27th ACM International Conference on Supercomputing (ICS)*. 103–114.

[23] Akifumi Kawahara, Ryotaro Azuma, Yuuichirou Ikeda, Ken Kawai, Yoshikazu Katoh, Yukio Hayakawa, Kiyotaka Tsuji, Shinichi Yoneda, Atsushi Himeno, and Kazuhiko Shimakawa. 2012. An 8 Mb multi-layered cross-point ReRAM macro with 443 MB/s write throughput. *IEEE Journal of Solid-State Circuits* 48, 1 (2013), 178–185.

[24] Kuk-Hwan Kim, Siddharth Gaba, Dana Wheeler, JoseMCruz-Albrecht, Tahir Hussain, Narayan Srinivasa, and Wei Lu. 2011. A functional hybrid memristor crossbar-array/CMOS system for data storage and neuromorphic applications. *Nano Lett.* 12, 1 (2011), 389–395.

[25] Emre Kültürsay, Mahmut Kandemir, Anand Sivasubramaniam, and Onur Mutlu. 2013. Evaluating STT-RAM as an energy-efficient main memory alternative. In *IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*. 256–267.

[26] Miguel Angel Lastras-Montaño, Amirali Ghofrani, and Kwang-Ting Cheng. 2016. A low-power hybrid reconfigurable architecture for resistive random-access memories. In *IEEE International Symposium on High Performance Computer Architecture (HPCA)*. 102–113.

[27] Benjamin C. Lee, Engin Ipek, Onur Mutlu, and Doug Burger. 2009. Architecting phase change memory as a scalable dram alternative. *ACM SIGARCH Comput. Archit. News*, Vol. 37. ACM, 2–13.

[28] Donghyuk Lee, Yoongu Kim, Vivek Seshadri, Jamie Liu, Lavanya Subramanian, and Onur Mutlu. 2013. Tiered-latency DRAM: A low latency and low cost DRAM architecture. In *19th IEEE International Symposium on High Performance Computer Architecture (HPCA)*. 615–626.

[29] H. Y. Lee, Y. S. Chen, P. S. Chen, P. Y. Gu, Y. Y. Hsu, S. M. Wang, W. H. Liu, C. H. Tsai, S. S. Sheu, and P. C. Chiang. 2010. Evidence and solution of over-RESET problem for HfOx based resistive memory with sub-ns switching speed and high endurance. In *IEEE International Electron Devices Meeting (IEDM)*. 19–7.

[30] Kwang-Jin Lee, Beak-Hyung Cho, Woo-Yeong Cho, Sangbeom Kang, Byung-Gil Choi, Hyung-Rok Oh, Chang-Soo Lee, Hye-Jin Kim, Joon-Min Park, and Qi Wang. 2008. A 90 nm 1.8 V 512 Mb diode-switch PRAM with 266 MB/s read throughput. *IEEE J. Solid-State Circuits* 43, 1 (2008), 150–162.

[31] Boxun Li, Peng Gu, Yi Shan, Yu Wang, Yiran Chen, and Huazhong Yang. 2015. RRAM-based analog approximate computing. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 34, 12 (2015), 1905–1917.

[32] Haitong Li, Tony FWu, Subhasish Mitra, and H-S PhilipWong. 2017. Resistive RAM-centric computing: Design and modeling methodology. *IEEE Transactions on Circuits and Systems I: Regular Papers* 64, 9 (Sept 2017), 2263–2273. DOI: http://dx.doi.org/10.1109/TCSI.2017.2709812

[33] Yanbin Li, Xin Li, Lei Ju, and Zhiping Jia. 2015. A three-stage-write scheme with flip-bit for PCM main memory. In *ASP-DAC*. IEEE, 328–333.

[34] Beiye Liu, Hai Li, Yiran Chen, Xin Li, Qing Wu, and Tingwen Huang. 2015. Vortex: Variation-aware training for memristor x-bar. In *52nd ACM/EDAC/IEEE Design Automation Conference (DAC)*. 1–6.

[35] Jamie Liu, Ben Jaiyen, Richard Veras, and Onur Mutlu. 2012. RAIDR: Retention-aware intelligent DRAM refresh. In *ACM SIGARCH Computer Architecture News*, Vol. 40. IEEE Computer Society, 1–12.

[36] Tz Yi Liu, Tian Hong Yan, Roy Scheuerlein, Yingchang Chen, Jeffrey KoonYee Lee, Gopinath Balakrishnan, Gordon Yee, Henry Zhang, Alex Yap, and Jingwen Ouyang. 2013. A 130.7mm2 2-layer 32Gb ReRAM memory device in 24nm technology. *IEEE Journal of Solid-State Circuits* 49, 1 (2014), 140–153.

[37] Shih-Lien Lu, Ying-Chen Lin, and Chia-Lin Yang. 2015. Improving DRAM latency with dynamic asymmetric subarray. In *MICRO*. IEEE, 255–266.

[38] Rakan Maddah, Seyed Mohammad Seyedzadeh, and Rami Melhem. 2015. CAFO: Cost aware flip optimization for asymmetric memories. In *21st IEEE International Symposium on High Performance Computer Architecture (HPCA)*. 320–330.

[39] Onur Mutlu. 2016. Rethinking memory system design. In *Mobile System Technologies Workshop (MST)*. IEEE, 1–3.

[40] Onur Mutlu. 2017. The rowhammer problem and other issues we may face as memory becomes denser. In *DATE*. IEEE, 1116–1121.

[41] Dimin Niu, Cong Xu, Naveen Muralimanohar, Norman P. Jouppi, and Yuan Xie. 2012. Design trade-offs for high density cross-point resistive memory. In *Proceedings of the 2012 ACM/IEEE International Symposium on Low Power Electronics and Design (ISLPED)*. 209–214.

[42] Dimin Niu, Qiaosha Zou, Cong Xu, and Yuan Xie. 2013. Low power multi-level-cell resistive memory design with incomplete data mapping. In *31st IEEE International Conference on Computer Design (ICCD)*. 131–137.

[43] Matt Poremba and Yuan Xie. 2012. Nvmain: An architectural-level main memory simulator for emerging non-volatile memories. In *ISVLSI*. IEEE, 392–397.

[44] Moinuddin K. Qureshi, Vijayalakshmi Srinivasan, and Jude A. Rivers. 2009. Scalable high performance main memory system using phase-change memory technology. *ACM SIGARCH Comput. Archit. News* 37, 3 (2009), 24–33.

[45] Yoshihiro Sato, Koji Tsunoda, Kentaro Kinoshita, Hideyuki Noshiro, Masaki Aoki, and Yoshihiro Sugiyama. 2008. Sub-100-μA reset current of nickel oxide resistive memory through control of filamentary conductance by current limit of MOSFET. *IEEE Trans. Electron Devices* 55, 5 (2008), 1185–1191.

[46] Stuart Schechter, Gabriel H. Loh, Karin Strauss, and Doug Burger. 2010. Use ECP, not ECC, for hard failures in resistive memories. *ACM SIGARCH Comput. Archit. News* 38, 141–152.

[47] Nak Hee Seong, Dong Hyuk Woo, Vijayalakshmi Srinivasan, Jude A. Rivers, and Hsien-Hsin S. Lee. 2010. SAFER: Stuck-at-fault error recovery for memories. In *Proceedings of the 43rd Annual IEEE/ACM International Symposium on Microarchitecture*. IEEE Computer Society, 115–124.

[48] Shyh-Shyuan Sheu, Meng-Fan Chang, Ku-Feng Lin, Che-Wei Wu, Yu-Sheng Chen, Pi-Feng Chiu, Chia-Chen Kuo, Yih-Shan Yang, Pei-Chia Chiang, and Wen-Pin Lin. 2011. A 4Mb embedded SLC resistive-RAM macro with 7.2 ns read-write random-access time and 160ns MLC-access capability. In *IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*. 200–202.

[49] Manjunath Shevgoor, Naveen Muralimanohar, Rajeev Balasubramonian, and Yoocharn Jeon. 2015. Improving memristor memory with sneak current sharing. In *33rd IEEE International Conference on Computer Design (ICCD)*. 549–556.

[50] Young Hoon Son, O. Seongil, Yuhwan Ro, Jae W. Lee, and Jung Ho Ahn. 2013. Reducing memory access latency with asymmetric DRAM bank organizations. *ACM SIGARCH Comput. Archit. News* 41, 380–391.

[51] Dmitri B. Strukov, Gregory S. Snider, Duncan R. Stewart, and R. Stanley Williams. 2008. The missing memristor found. *Nature* 453, 7191 (2008), 80–83.

[52] Jeffrey Stuecheli, Dimitris Kaseridis, Hillery C. Hunter, and Lizy K. John. 2010. Elastic refresh: Techniques to mitigate refresh penalties in high density memory. In *43rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*. 375–384.

[53] N. Talati, S. Gupta, P. Mane, and S. Kvatinsky. 2016. Logic design within memristive memories using memristor-aided loGIC (MAGIC). *IEEE Transactions on Nanotechnology* 15, 4 (July 2016), 635–650.

[54] Chengning Wang, Dan Feng, Jingning Liu, Wei Tong, Bing Wu, and Yang Zhang. 2017. DAWS: Exploiting crossbar characteristics for improving write performance of high density resistive memory. In *ICCD*. IEEE, 281–288.

[55] Wujie Wen, Mengjie Mao, Xiaochun Zhu, Seung H Kang, Danghui Wang, and Yiran Chen. 2013. CD-ECC: Content-dependent error correction codes for combating asymmetric nonvolatile memory operation errors. In *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. IEEE Press, 1–8.

[56] R. Stanley Williams. 2008. How we found the missing memristor. *IEEE Spectrum* 45, 12 (2008), 28–35.

[57] Linda Wilson. 2013. International technology roadmap for semiconductors (ITRS). *Semiconductor Industry Association* (2013).

[58] Cong Xu, Pai-Yu Chen, Dimin Niu, Yang Zheng, Shimeng Yu, and Yuan Xie. 2014. Architecting 3D vertical resistive memory for next-generation storage systems. In *Proceedings of the 2014 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. IEEE Press, 55–62.

[59] Cong Xu, Dimin Niu, Yang Zheng, Shimeng Yu, and Yuan Xie. 2015. Impact of cell failure on reliable cross-point resistive memory design. *ACM Transactions on Design Automation of Electronic Systems (TODAES)* 20, 4 (2015), 63.

[60] Cong Xu, Dimin Niu, Naveen Muralimanohar, Rajeev Balasubramonian, Tao Zhang, Shimeng Yu, and Yuan Xie. 2015. Overcoming the challenges of crossbar resistive memory architectures. In *21st IEEE International Symposium on High Performance Computer Architecture (HPCA)*. 476–488.

[61] G. M. Yin, F. Opt Eynde, and Willy Sansen. 1992. A high-speed CMOS comparator with 8-b resolution. *IEEE J. Solid-State Circuits* 27, 2 (1992), 208–211.

[62] Hanbin Yoon, Justin Meza, Naveen Muralimanohar, Norman P. Jouppi, and Onur Mutlu. 2015. Efficient data mapping and buffering techniques for multilevel cell phase-change memories. *ACM Trans. Archit. Code Optim.* 11, 4 (2015), 40.

[63] Shimeng Yu, Yi Wu, and H-S Philip Wong. 2011. Investigating the switching dynamics and multilevel capability of bipolar metal oxide resistive switching memory. *Appl. Phys. Lett.* 98, 10 (2011), 103514.

[64] Jianhui Yue and Yifeng Zhu. 2013. Accelerating write by exploiting PCM asymmetries. In *HPCA*. IEEE, 282–293.

[65] Hang Zhang, Nong Xiao, Fang Liu, and Zhiguang Chen. 2016. Leader: Accelerating ReRAM-based main memory by leveraging access latency discrepancy in crossbar arrays. In *Proceedings of the 2016 Conference on Design, Automation & Test in Europe (DATE)*. EDA Consortium, 756–761.

[66] Tao Zhang, Matt Poremba, Cong Xu, Guangyu Sun, and Yuan Xie. 2014. CREAM: A concurrent-refresh-aware DRAM memory architecture. In *20th IEEE International Symposium on High Performance Computer Architecture (HPCA)*. IEEE, 368–379.

[67] Yang Zhang, Dan Feng, Jingning Liu, Wei Tong, Bing Wu, and Caihua Fang. 2017. A novel ReRAM-based main memory structure for optimizing access latency and reliability. In *54th ACM/EDAC/IEEE Design Automation Conference (DAC)*. 1–6.

[68] Lei Zhao, Lei Jiang, Youtao Zhang, Nong Xiao, and Jun Yang. 2017. Constructing fast and energy efficient 1TnR based ReRAM crossbar memory. In *18th IEEE International Symposium on Quality Electronic Design (ISQED)*. 58–64.

[69] Yang Zheng, Cong Xu, and Yuan Xie. 2015. Modeling framework for cross-point resistive memory design emphasizing reliability and variability issues. In *ASP-DAC*. IEEE, 112–117.

[70] Ping Zhou, Bo Zhao, Jun Yang, and Youtao Zhang. 2009. A durable and energy efficient main memory using phase change memory technology. *ACM SIGARCH Comput. Archit. News* 37, 14–23.

Received July 2017; revised February 2018; accepted March 2018