A Novel ReRAM-based Main Memory Structure for Optimizing Access Latency and Reliability

Yang Zhang, Dan Feng*, Jingning Liu, Wei Tong*, Bing Wu, Caihua Fang
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, Wuhan, China
{u201014528, dfeng, jnliu, tongwei, wubin200, caihuafang}@hust.edu.cn
*Co-corresponding authors

ABSTRACT

Emerging Resistive Memory (ReRAM) is a promising candidate as the replacement for DRAM because of its low power consumption, high density and high endurance. Due to the unique crossbar structure, ReRAM can be constructed with a very high density. However, ReRAM's crossbar structure causes an IR drop problem which results in non-uniform access latency in ReRAM banks and reduces its reliability. Besides, the access latency and reliability of ReRAM arrays are greatly influenced by the data patterns involved in a write operation. In this paper, we propose a performance and reliability efficient ReRAM-based main memory structure. At the circuit level, we propose a double-sided write driver design to reduce the IR drops along bitlines. At the architecture level, a region partition with address remapping method and two flip schemes are proposed to reduce the access latency and improve the reliability of ReRAM arrays. The experimental results show that the proposed design can improve the system performance by 30.3% on average and reduce the memory access latency by 25.9% on average over an aggressive baseline, meanwhile the design improves the reliability of ReRAM-based memory system.

Keywords

ReRAM, crossbar, IR drop, non-uniform access latency, data patterns

1. INTRODUCTION

Non-Volatile Memories (NVMs) such as Phase Change Memory (PCM), Spin-Transfer Torque RAM (STT-RAM) and Resistive RAM (ReRAM) have better scalability with low power consumption and high density while the scalability of DRAM reaches its bottleneck. As ITRS indicates, the scaling path of DRAM beyond 16nm is not clear[1]. Therefore, NVMs with excellent characteristics are actively explored as replacement for DRAM. Among these candidates, ReRAM is the most promising because of its lower power consumption, higher density and higher endurance[2][3].

ReRAM cells can be built into a crossbar structure without access transistors to reach an extremely high density because of their nonlinearity. Moreover, multiple layers of crossbar can be stacked through 3D integration technology which can achieve a higher density[4][5]. However, the crossbar structure faces many challenges. 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 full-selected ReRAM cells will be too small to perform a reliable write or read. Moreover, the switching time of ReRAM cells is exponentially inversely proportional to the voltage applied on ReRAM cells. So the IR drop problem increases the switching time of ReRAM cells and causes the non-uniform access latency in ReRAM memory banks. Furthermore, the access latency and reliability of ReRAM arrays are greatly influenced by the data patterns during a multiple-bits write operation. For a multiple-bits writing crossbar, writing more "0" exponentially increases the access latency but it benefits the reliability. While writing more "1" reduces the access latency but it's harmful to the reliability. To make ReRAM more suitable for main memory system, we have to mitigate the IR drop problem and leverage the non-uniform access latency in ReRAM memory banks and optimize the data patterns to reduce the access latency and improve the reliability of ReRAM arrays. In this paper, we propose a novel ReRAM-based main memory structure for optimizing access latency and reliability. The contributions of this paper include:

1. At the circuit level, we propose a novel circuit design called double-sided write driver (DSWD) to reduce the IR drops along bitlines.

2. At the architecture level, we divide a 8-bits writing crossbar array into multiple logical regions according to the non-uniform access latency in ReRAM banks, then remap the hot data to fast regions and remap the cold data to slow regions, which reduces the access latency efficiently.

3. We propose a latency-based flip scheme in the fast regions to reduce the access latency and propose a reliability-based flip scheme in the slow regions to improve the reliability of ReRAM arrays.

DAC '17, June 18-22, 2017, Austin, TX, USA

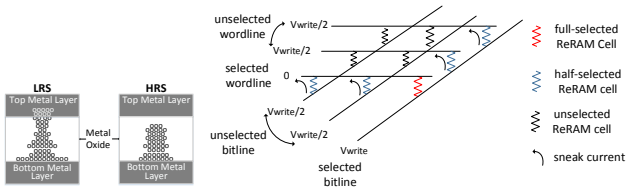
DOI: http://dx.doi.org/10.1145/3061639.3062191

4. The proposed design can improve the system performance by 30.3% on average and reduce the memory access latency by 25.9% on average over an aggressive baseline, meanwhile the design improves the reliability of ReRAM-based memory system.

2. PRELIMINARIES AND MOTIVATION

2.1 ReRAM Basics

ReRAM cell has a very simple structure, consisting of a metal-oxide layer sandwiched between a top metal layer and a bottom metal layer, as shown in Figure 1(a). By applying an external voltage to ReRAM cells, the resistance of ReRAM cells can change between high resistance state (HRS) and low resistance state (LRS). HRS is referred to the logic "0" and LRS is referred to the logic "1". For ReRAM cells, the switching from LRS to HRS is defined as a RESET operation and the switching from HRS to LRS is defined as a SET operation.



(a) (b)

Figure 1: (a)ReRAM cell structure (b) crossbar structure and RESET operation

Generally, ReRAM array structure can be classified into two types: 1T1R grid structure and crossbar structure. In a 1T1R grid structure, each cell has a dedicated MOSFET transistor and each cell can be accessed independently without disturb. However, these transistors tremendously increase the area and the cost. While in a crossbar structure, 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. So the crossbar structure is more suitable for ReRAM-based main memory. However, crossbar structure brings other challenges for ReRAM-based main memory.

The crossbar structure causes an IR drop problem due to wire resistance and sneak currents, which is harmful to the access latency and reliability of ReRAM arrays. To reduce the sneak currents, Half-Wordline Half-Bitline (HWHB) write scheme is widely adopted. With HWHB, when a RESET operation is performed, the selected bitline is set to V_{write} and the selected wordline is set to 0, and all of the unselected wordlines and bitlines are half biased at $V_{write}/2$, as shown in Figure 1(b). HWHB makes the voltage drop of half-selected ReRAM cells biased at $V_{write}/2$ and the voltage drop of unselected cells biased at 0. However, even with HWHB write scheme, the sneak currents still exist due to the $V_{write}/2$ voltage across these half-selected ReRAM cells.

2.2 Motivation

ReRAM is an asymmetric memory whose write latency is much lager than read latency and RESET latency is much larger than SET latency. So the RESET operation becomes the performance bottleneck[2][3]. Besides, the switching time of ReRAM is exponentially inversely proportional to the voltage applied on the cell. The relationship between switching time 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[6]. Therefore, the reduction of V_d has a great impact on the switching time, especially on the RESET latency. Even 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-select cells may suffer write disturbance. To construct an efficient and reliable ReRAM-based main memory system, we should keep the V_d large enough without affecting other cells.

In circuit level, several designs have been proposed to decrease the IR drop in recent years[2][7]. DSGB scheme and dual-port write scheme have been proposed to decrease the IR drops along wordlines. However, there is no design to decrease the IR drops along bitlines for the RESET operation. Even worse, the IR drops along bitlines account for a large proportion as the size of ReRAM arrays becomes larger. It's urgent to design a novel circuit to reduce the IR drops along bitlines.

Furthermore, in a crossbar, rows near the write driver have lower wire resistance along bitlines than rows far from the write driver. It means that rows near the write driver have fewer IR drops than rows far from the write driver, causing that each row has a different voltage drop according to the distance between the row and write driver. The different voltage drop on each row causes an non-uniform access latency in ReRAM memory banks and each row has a different access latency. 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 loses the performance.

Finally, data patterns during a write operation have a great influence on the access latency and reliability of a crossbar, especially for a multiple-bits writing crossbar. Due to the nonlinearity of ReRAM cells, writing more "0" exponentially increases the access latency without buckets effect. For the sake of the write performance, we should ensure that there are fewer "0" to be written into ReRAM arrays during a multiple-bits write operation. On the other hand, when the crossbar array writes more "0", the resistance of the array will be high enough, which reduces sneak currents according to Ohm's Law and improves the reliability of the crossbar array. However, previous designs have never synthetically considered the data patterns for the access latency and reliability of ReRAM arrays.

3. THE CIRCUIT-LEVEL OPTIMIZATION

In order to reduce the IR drops along bitlines, we propose a novel circuit architecture called double-sided write driver (DSWD) for crossbar arrays. Different from the conventional design, DSWD applies additional write drivers and sense amplifiers for the top level crossbar arrays and the bottom level crossbar arrays in ReRAM banks, as shown in Figure 2. DSWD guarantees that each crossbar array has the same number of write drivers in double 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

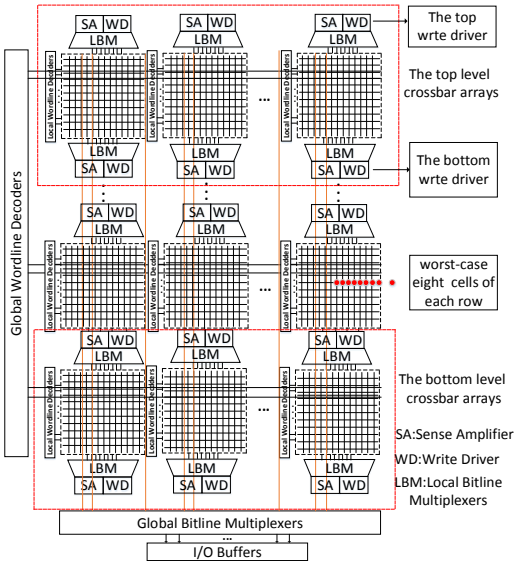


Figure 2: Schematic view of an ReRAM bank with DSWD scheme

of write drivers can be implemented through a simple selection circuit. In DSWD design, write drivers and sense amplifiers between adjacent crossbar arrays are shared as the conventional design does. Therefore, just the top level crossbar arrays and the bottom level crossbar arrays need additional write drivers and sense amplifiers in the DSWD design. Besides, the data parallelism has been maintained through alternately activating different level crossbar arrays.

To quantitatively show the advantage brought by DSWD, we build a detailed circuit model for the 1024×1024 crossbar array with Kirchhoff's Current Law[8] and we model a 1Gb $\times 8$ 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-bits writing scheme as an example rather than single-bit writing scheme, because 8-bits writing scheme is more energy efficient and has more data patterns for optimization. Table 1 shows some key parameters obtained from HfO_x-based cells[6] and IBM's MIEC device[9]. The latency of the worst-case eight ReRAM cells which are furthest from the row decoder is measured as the RESET latency of each row and the worst-case eight ReRAM cells of each row are shown in Figure 2. The RESET latency of the worst-case eight ReRAM cells can be classified into eight categories according to the number of "0" written to the row. When all the worst-case eight ReRAM cells writes "0", the RESET latency is the worst. To vividly verify the DSWD scheme, we perform a RESET operation for the worst-case eight ReRAM 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 3. The results show that rows over 512 in DSWD design have a much larger voltage drops than the conventional design (without DSWD). In the conventional design, the worst-case row is the 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 DSWD scheme, the worst-case row is the row 512 and the paths of IR drops along bitlines are tremendously reduced for the rows over 512. The DSWD scheme provides lager voltage drops for cells over row 512, improving the

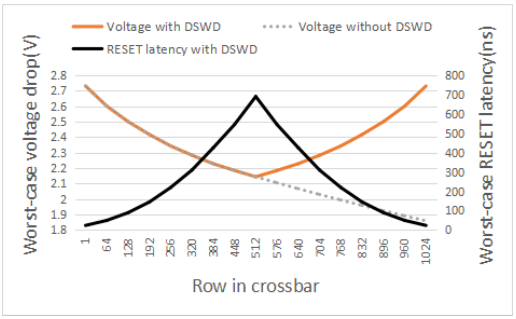


Figure 3: The relationship of worst-case voltage drop and RESET latency at different row in a crossbar array with DSWD

Table 1: Parameters in the Crossbar Array Model

Metric	Description	Value
A	Mat size: A *wordlines*$\times$ A *bitlines*	1024
n	Number of bits to read/write in a mat	8
I_{on}	Cell current of a LRS ReRAM during RESET	20μA
R_{wire}	Wire resistancec 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

access latency and reliability of these ReRAM cells.

However, the DSWD scheme brings hardware overhead because it requires another 512 write drivers and sense amplifiers for the top level crossbar arrays and the bottom level crossbar arrays, causing a 6% write driver and sense amplifier overhead and reducing the area efficiency. Considering the great reduction in RESET latency, the additional overhead is completely acceptable.

4. THE ARCHITECTURE-LEVEL OPTIMIZATION

4.1 Region Partition and Address Remapping

To make full use of the non-uniform access latency in ReRAM banks , we divide a crossbar array into fast regions and slow regions according to their access latencies. We gather every 32 rows as a region and a mat is divided into 32 regions. Since the access latencies of the crossbar array are longitudinal symmetric based on the DSWD scheme, we gather the top rows and the bottom rows as fast regions. The middle rows are grouped as slow regions, as Figure 4 shows. The ratio of fast regions to slow regions is adjustable for different applications.

Considering the access hotspots in memory banks, we propose an efficient address remapping scheme to match the access hotspots with the non-uniform access latency in ReRAM banks. The address remapping design contains three stages: logic address (LA) of memory requests to logic region number (LRN), hot data identification and LRN to physical region number (PRN). Figure 4 shows the detailed address remapping scheme.

1)LA to LRN: We gather every 64 LAs as a LRN. Because the data of a memory request is 64B and a region in our design is 4KB, the data of 64 LAs is exactly 4KB. The data of a LRN is also 4KB.

2)Hot data identification: We allocate a counter for every LRN to record the temperature according to its access frequency, where the read request and the write request have

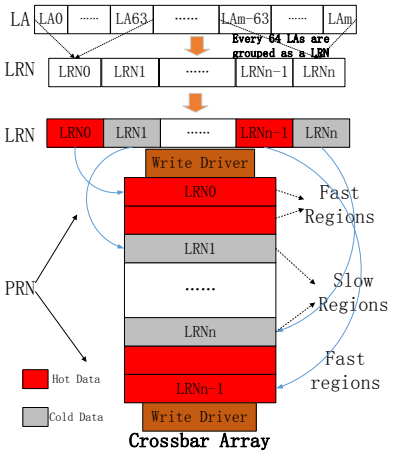


Figure 4: Region partition and address remapping in a crossbar array with DSWD

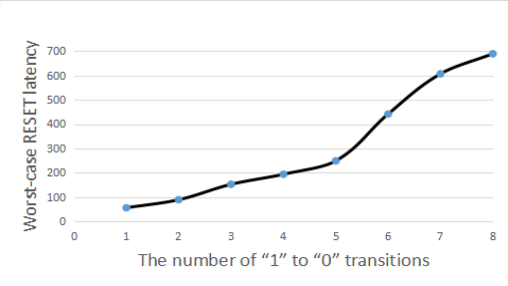


Figure 5: The relationship of worst-case RESET latency and the number of "1" to "0" transitions in a 1024×1024 crossbar array with DSWD

the same weight. We calculate the temperature every time stamp and the temperature of LRNs is divided by 2 after a time stamp. When the temperature of a LRN reaches the threshold, the data of the LRN is identified as hot data. Otherwise, the data of the LRN is 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. Perfectly, the LRN and the PRN are one-to-one, because the data of a LRN and the data of a PRN are all 4KB. When the fast regions is full, we migrate the data of the fast region with the lowest temperature to the slow region and modify the address remapping table.

4.2 Latency-Based Flip Scheme

Due to the nonlinearity of ReRAM cells, writing fewer "0" during a multiple-bits writing is benefit to the access latency without buckets effect. The number of "1" to "0" transitions has an important influence on the access latency, as Figure 5 shows. So it's necessary to reduce the number of "1" to "0" transitions for a better performance during a multiple-bits writing. A simple data inversion for ReRAM has been proposed in previous research[2], but the old data is ignored and the performance is sub-optimal. To achieve a better performance, we propose a Latency-Based Flip scheme (LBF), where the new data and the old data are all taken into account. LBF contains three phases:read, analysis and write. The processing flow of LBF is shown in Algorithm 1.

1)Read phase: LBF leverages read-before-write scheme to reduce the number of "1" to "0" transitions. LBF firstly reads out the old data and its flip flag bit $\{D', F'\}$.

2)Analysis phase: LBF compares the new data and the default flip flag bit $\{D, F\}$ with $\{D', F'\}$, then records the number of "1" to "0" transitions and assigns the number to A. Besides, LBF flips the new data and its default flip flag and gets the flipped data and its new flip flag bit $\{D'', F''\}$. Then LBF compares $\{D'', F''\}$ with $\{D', F'\}$ and records the number of "1" to "0" transitions and assigns the number to B.

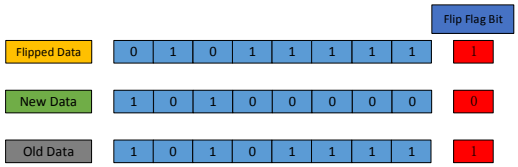


Figure 6: LBF scheme

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

Figure 6 is an example to explain the LBF. During the read phase, $\{10101111, 1\}$ is read out. During the analysis phase, LBF compares $\{10100000, 0\}$ with $\{10101111, 1\}$ and A is 5. Then LBF compares $\{01011111, 1\}$ with $\{10101111, 1\}$ and B is 2. Apparently, A is bigger than B, so during the write phase $\{01011111, 1\}$ is written to the ReRAM array. Although LBF efficiently reduces the number of "1" to "0" transitions and greatly reduces RESET latency, it causes a 11.1% storage overhead for the additional the flip flag bits.

Algorithm 1 LBF Algorithm

Require:
The old data and old flip bit, $\{D', F'\}$;
The new data and default flip flag bit F = 0, $\{D, F\}$;
The flipped data and flip flag bit of the new data, $\{D'', F''\}$;
A function to count the number of 1 to 0 transitions comparing D1 with D2, $Count\_1\_0\_transition(D1, D2)$;

Ensure:
1: read($\{D', F'\}$);
2: A = Count_1_0_transition($\{D, F\}$,$\{D', F'\}$);
3: B = Count_1_0_transition($\{D'', F''\}$,$\{D', F'\}$);
4: **if** $A > B$ **then**
5: write($\{D'', F''\}$);
6: **else**
7: write($\{D, F\}$);
8: **end if**

4.3 Reliability-Based Flip Scheme

The reliability of a crossbar is a function of the data patterns involved in a write operation. When the crossbar array writes more "0", the resistance of the array will be high enough to guarantee fewer sneak currents according to Ohm's Law. The IR drops in the crossbar will also be greatly reduced and the write/read failure may be avoided. In order to improve the reliability of the crossbar array, we propose a Reliability-Based Flip scheme (RBF). The processing flow of RBF is shown in Algorithm 2.

RBF only ensures more "0" to be written into the crossbar array. So RBF only records the number of "0" in the new data and assigns the number to C. If C is larger than N/2, where N is the number of bits to read or write in a mat, the new data will be directly written to the crossbar array with a default flip flag bit "1". Otherwise, the new data will be flipped and written to the crossbar array with flip flag bit

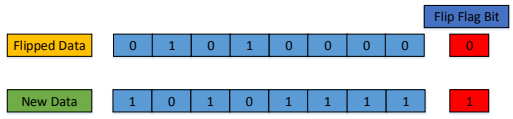


Figure 7: RBF scheme

"0". Figure 7 is an example to explain the RBF and N is 8. The number of "0" in the new data is 2 and C is smaller than N/2. So the new data is flipped and written to the ReRAM array with a flip flag bit.

Although the RBF theoretically reduces the sneak currents and improves the reliability of the crossbar array, it also causes a 11.1% storage overhead and worsens the writing latency.

Algorithm 2 RBF Algorithm

Require:

The new data and default flip flag bit F = 1, $\{D, F\}$;

The flipped data and flip flag bit of the new data, $\{D'', F''\}$;

A function to count the number of 0 in data D3, $Count\_0(D3)$;

The number of bits to read/write in a mat, N;

Ensure:

1: C = Count_0($\{D, F\}$);
2: **if** $C > N/2$ **then**
3: write($\{D, F\}$);
4: **else**
5: write($\{D'', F''\}$);
6: **end if**

4.4 Latency and Reliability Optimization for ReRAM-Based Memory System

As section 4.1 explains, the hot data tends to be written into the fast regions and the cold data tends to be written into the slow regions for the sake of performance. To achieve better performance and reliability, we propose a novel latency and reliability optimization design for ReRAM-based memory system, called LRR. LRR not only mitigates the IR drop problem with DSWD and leverages the non-uniform access latency in ReRAM memory banks through the region partition and address remapping methods, but also optimizes the data patterns for the access latency and reliability of ReRAM arrays. LRR actually implements LBF scheme in the fast regions of crossbar arrays and implements RBF scheme in the slow regions of crossbar arrays. LRR further reduces the access latency of the fast regions through LBF scheme and improves the reliability of the whole crossbar arrays through RBF scheme. Considering that the hot data in the fast regions is sensitive to the access latency, LRR can clearly optimize the access latency and maximize the performance for ReRAM-based memory system. In addition, the cold data in the slow regions is not sensitive to the access latency and it is suitable to implement RBF scheme in the slow regions. Therefore, LRR can efficiently reduce the access latency and improve the reliability of ReRAM-based memory system.

5. EXPERIMENTAL RESULTS

5.1 Experiment Setup

Table 2: 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	tRCD(18), tCL(15), tCWD(13), tFAW(30), tWTR(7.5)

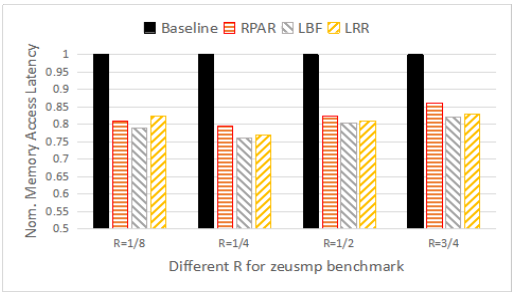


Figure 8: The average memory access latency with different R for zeusmp benchmark

To evaluate our work, we use GEM5[10] simulator with NVMain[11] as our simulation platform. Table 2 shows the detailed configurations of our simulator. We select SPEC CPU2006 benchmarks[12] with reference input size. We run all benchmarks for 500 million instructions to warmup caches and then run 1 billion instructions for our proposed design. We select the DSGB (double-sided ground biasing) design as the aggressive baseline to compare with our design. The comparison configurations are as follows: *Baseline*: A 8-bits writing crossbar array with the worst-case RESET latency under DSGB. *RPAR*: Only apply the region partition and address remapping methods under DSWD. *LBF*: Apply the LBF scheme in the whole crossbar based on RPAR. *LRR*: Apply the LBF scheme in the fast regions and apply the RBF scheme in the slow regions based on RPAR. Considering that the ratios of hot data to cold data are different in different workloads, we set different ratios of fast regions to slow regions for different workloads. The ratio of fast regions to all regions in a mat is recorded as R in our experiments.

5.2 Simulation Results

To analyze the effects of different R on the same workload, we set $R = 1/8$,$R = 1/4$,$R = 1/2$ and $R = 3/4$ for each workload. Figure 8 and Figure 9 show the average memory access latency and IPC speedup with different R for zeusmp benchmark, respectively. As for zeusmp benchmark, when R is 1/4, our methods achieve the best performance. When R is 1/8, the size of fast regions is too small to match the hot data for zeusmp benchmark, which results in sub-optimal performance. As R becomes larger, the size of fast regions and the average 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 zeusmp benchmark, which increases the access latency. Besides, we can see that the performance of LRR is worse than RPAR when R is 1/8. That's because the size of slow regions is too large and RBF scheme in slow regions cause a performance loss for LRR. But as R becomes larger, the performance loss of LRR becomes smaller.

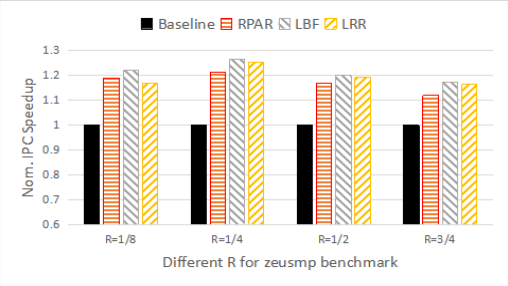


Figure 9: The IPC speedup with different R for zeusmp benchmark

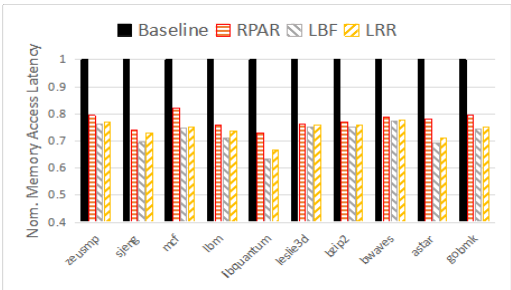


Figure 10: The average memory access latency

Each workload has a fit R for the best performance according to the ratio of hot data to cold data. Figure 10 shows the average memory access latency for different workloads with their fit R and the results are normalized to the baseline. As we can see, the proposed techniques can efficiently reduce the memory access latency. The LBF reduces access latency by 27.4% and achieves the lowest access latency, because the whole crossbar is optimal for the access latency. The LRR has an about 1.5% higher access latency than LBF, because RBF scheme causes more "1" to "0" transitions in slow regions. But the LRR makes the crossbar more reliable.

Figure 11 shows the system speedup for our design compared with the baseline under different workloads. The results show that LRR has a 30.3% IPC improvement.

Hardware Overhead: In the double-sided write driver design, additional 6% write driver and sense amplifiers overhead are needed. Besides, in the region partition and address remapping methods, the address remapping table causes some hardware 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. Additional 14 bits are required to count the temperature for each PRN and $14 \times 256K$ bits is needed. Therefore, for a 1GB memory rank, it incurs 1MB storage overhead for the region partition and address remapping methods. In LBF and RBF scheme, it needs additional 11.1% storage overhead.

6. CONCLUSION

In this paper, we propose a latency and reliability optimization design for ReRAM-based memory system. At the circuit level, the double-sided write driver design is proposed to mitigate the IR drop problem of the crossbar structure. At the architecture level, the region partition and address remapping methods are proposed to reduce access latency. We further propose a latency-based flip scheme and a reliability-based flip scheme to reduce the access latency and improve the reliability of ReRAM arrays, respectively. The experimental results show that the proposed design can

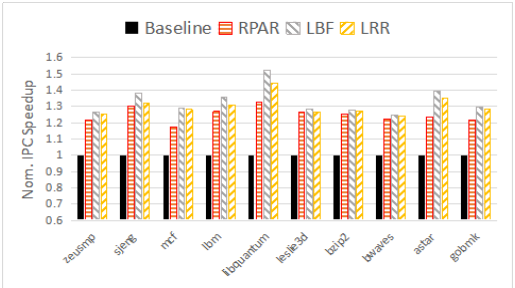


Figure 11: The IPC speedup for SPEC CPU2006

improve the system performance by 30.3% on average and reduce the memory access latency by 25.9% on average over an aggressive baseline, meanwhile the design improves the reliability of ReRAM-based memory system.

7. ACKNOWLEDGEMENTS

This work was supported by the 863 Project No.2015AA015301, No.2013AA013203, No.2015AA016701; NSFC No.61502190, No.61303046, No.61472153; Wuhan Applied Basic Research Project (No.2015010101010004).

8. REFERENCES

[1] ITRS Roadmap. International technology roadmap for semiconductors. *Semiconductor Industry Association*, 2013.

[2] Cong Xu et al. Overcoming the challenges of crossbar resistive memory architectures. In *HPCA*, pages 476–488. IEEE, 2015.

[3] Hang Zhang et al. Leader: Accelerating reram-based main memory by leveraging access latency discrepancy in crossbar arrays. In *DATE*, pages 756–761, 2016.

[4] Myoungsoo Jung et al. Design of a large-scale storage-class rram system. In *ICS*, pages 103–114, 2013.

[5] Cong Xu et al. Architecting 3d vertical resistive memory for next-generation storage systems. In *ICCAD*, pages 55–62. IEEE Press, 2014.

[6] HY Lee et al. Evidence and solution of over-reset problem for hfox based resistive memory with sub-ns switching speed and high endurance. In *IEDM*, pages 19–7. IEEE, 2010.

[7] Yang Zheng et al. Modeling framework for cross-point resistive memory design emphasizing reliability and variability issues. In *ASP-DAC*, pages 112–117, 2015.

[8] Dimin Niu et al. Design trade-offs for high density cross-point resistive memory. In *ISLPED*, pages 209–214. ACM, 2012.

[9] G. W Burr et al. Large-scale (512kbit) integration of multilayer-ready access-devices based on mixed-ionic-electronic-conduction (miec) at 100In *Symposium on VLSI Technology - IEEE Electron Devices Soc*, pages 41–42, 2012.

[10] Nathan B. et al. The gem5 simulator. *ACM SIGARCH Computer Architecture News*, 39(2):1–7, 2011.

[11] Matt Poremba et al. Nvmain: An architectural-level main memory simulator for emerging non-volatile memories. In *ISVLSI*, pages 392–397. IEEE, 2012.

[12] John L Henning. Performance counters and development of spec cpu2006. *ACM SIGARCH Computer Architecture News*, 35(1):118–121, 2007.