LER: Least-Error-Rate Replacement Algorithm for Emerging STT-RAM Caches

Amir Mahdi Hosseini Monazzah, *Student Member, IEEE*, Hamed Farbeh, *Student Member, IEEE*, and Seyed Ghassem Miremadi, *Senior Member, IEEE*

***Abstract*—Spin-transfer-torque RAMs (STT-RAMs) are the most promising technology for replacing Static RAMs (SRAMs) in on-chip caches. One of the major problems in STT-RAMs is the high error rate due to stochastic switching in WRITE operations. Cache replacement algorithms have a major role in the number of WRITE operations into the caches. Due to this fact, it is necessary to redesign cache replacement algorithms to consider the new challenges of STT-RAM caches. This paper proposes a cache replacement algorithm, which is called *least error rate (LER)*, to reduce the error rate in L2 caches. The main idea is to place the incoming block in a line that incurs the minimum error rate in WRITE operation. This is done by comparing the contents of the incoming block with lines in a cache set. Compared with Least Recently Used (LRU) algorithm, LER reduces the error rate by 2× with about 1.4% and 3.6% performance and dynamic energy consumption overheads, respectively. Moreover, LER imposes no area overhead to system.**

***Index Terms*—Cache replacement algorithm, error rate, stochastic switching, STT-RAM.**

I. Introduction

THE recent challenges of manufacturing SRAM caches in nanoscale technologies have led to the idea of changing the memory cells technologies in on-chip caches [1]. Recent studies reveal that among various memory technologies, non-volatile memories are the most attractive alternative to SRAMs [2]. Many studies have been done to explore the possibility of using different non-volatile technologies, e.g., Phase Change Memories (PCMs), Resistive RAMs (RRs), and Spin Torque Transfer RAMs (STT-RAMs), across the on-chip memories (i.e. caches and ScratchPads) [1], [3]–[5]. Whereas each memory technology has its own pros and cons, STT-RAMs are the most promising alternative for SRAMs, according to the last reports of International Technology Roadmap for Semiconductors (ITRS) [2].

The main advantages of STT-RAMs over SRAMs are negligible leakage power, higher density, higher immunity to radiation-induced particles strike, and higher security [6]. However, the challenges of STT-RAMs need to be addressed to make it applicable in on-chip caches. One of the main challenges in STT-RAMs is their high error rate in write operations [7]. The stochastic switching characteristic of the storing elements, i.e., Magnetic Tunnel Junctions (MTJs), results in high error rate in write operations. This error rate can be reduced by increasing the amplitude and the duration of the write pulses [7]. However, high energy and latency imposed by this approach is unaffordable in the caches. Reducing the error rate of STT-RAM caches with acceptable overheads is a must to replace SRAMs with STT-RAMs.

Due to physical characteristics of MTJs, the error rate in STT-RAM cells is asymmetric. For a predetermined amplitude and duration of the write pulse in STT-RAM caches, the switching probability of 0 to 1 transition is different from that in 1 to 0 transition. Therefore, the probability of correct write operation for changing the MTJ state from 1 to 0 is different from changing the MTJ state from 0 to 1. As reported, the write error rate in 0 to 1 transition is higher than that in 1 to 0 transition by about two orders of magnitude [8].

Several studies addressed the write error rate reduction in STT-RAM caches [8]–[11]. However, a few researches considered the asymmetric error rate in write operations. These researches can be categorized into circuit-level techniques and RTL-level techniques. The circuit-level techniques try to reduce the error rate by manipulating the characteristics of the write pulses [8], [9]. The duration of the write pulses per memory cell in these techniques is adjusted according to the type of the transition in the cells [9]. The RTL-level techniques, on the other hand, try to reduce the *Hamming weight* of the cache lines contents by encoding the incoming data [10], [11]. The techniques in both of these two categories impose significant overheads to the caches and require redesigning of the peripheral circuits to reduce the error rates.

The error rate in write operations has a direct relationship with the number of transitions, i.e., the Hamming distance between the incoming block and current block in the cache. In L2-caches, the write operations are mainly due to cache misses and write-backs from L1-caches. Due to the diversity in the contents of the cache lines, the Hamming distances between the incoming block and various lines in a cache set can be significantly different. Therefore, the content of the target cache line that the incoming block will be written into affects the error rate in the write operation. The target cache line is determined by the cache replacement algorithm. This implies that the cache replacement algorithm can significantly affect the STT-RAM error rate.

This paper proposes a cache replacement algorithm, called *Least Error Rate (LER)*, to minimize the error rate of STT-RAM

Manuscript received February 23, 2016; accepted April 19, 2016. Date of publication May 3, 2016; date of current version June 3, 2016.

The authors are with the Department of Computer Engineering, Sharif University of Technology, Tehran 11155-11365, Iran (e-mail: ahosseini@ce.sharif.edu; farbeh@mehr.sharif.edu; miremadi@sharif.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TDMR.2016.2562021

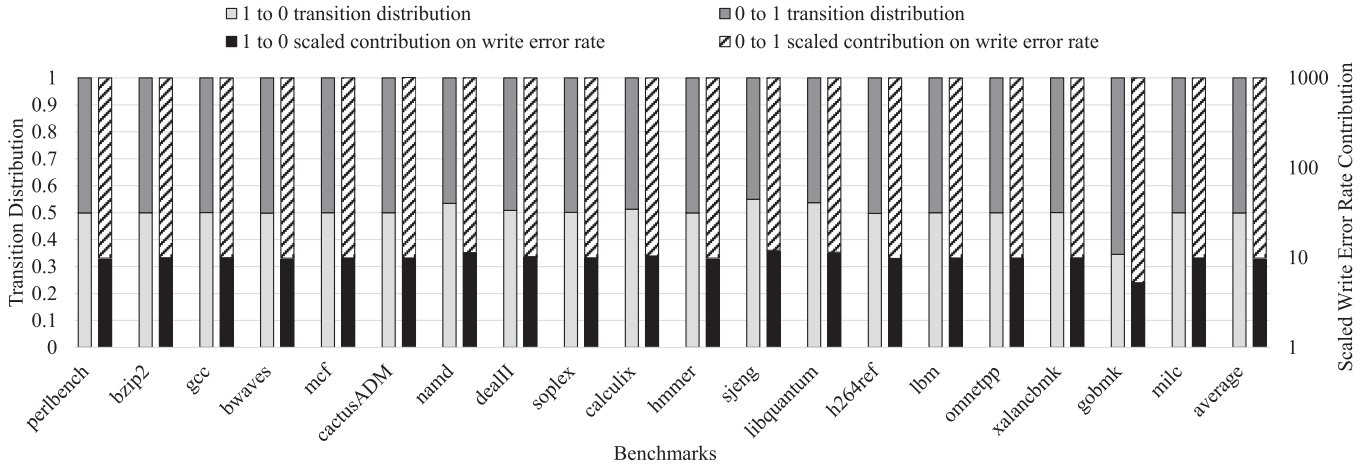

Fig. 1. Distribution of 0 to 1 and 1 to 0 transitions in WRITE operations and their contributions in the WRITE error rate.

caches in write operations. In this algorithm, the target cache line for the incoming block is a line that incurs the minimum error rate among all lines in the set. To this aim, LER compares the contents of the incoming block with the contents of the lines in the set, considering the asymmetric write error rate in STT-RAM cells. The target line is a line that incurs the minimum number of 0 to 1 transitions when writing the incoming block. To the best of our knowledge, LER is the first work that addresses the contribution of the cache replacement algorithm on the STT-RAM error rate and optimizes the cache replacement algorithm for STT-RAM cache characteristics.

LER algorithm is evaluated using gem5 simulator [12] running SPEC CPU2006 benchmark suite as the workloads [13]. The simulation results show on average 2 times improvement in the error rate of the STT-RAM caches. This is achieved by no area overhead and only 1.4% and 3.6% performance and dynamic energy consumption overheads, respectively. It is noteworthy to say that LER is completely orthogonal to previous studies which try to reduce the write error rate across the STT-RAM caches.

The remaining of this paper is organized as follows. In Section II, the proposed LER is introduced in detail. Section III includes the simulation setup and results. Finally, conclusion remarks are given in Section IV.

II. Proposed Cache Replacement Algorithm

The recent challenges in traditional SRAM caches have led to the idea of changing the memory cell technology in the caches [2]. As mentioned, STT-RAM caches are the most promising alternative for the traditional SRAM caches. Although using STT-RAMs relieves several SRAM problems in the caches, new challenges need to be addressed for STT-RAM caches. Many of these challenges can be alleviated by redesigning the traditional SRAM caches components.

This paper proposes a novel cache replacement algorithm to overcome a major challenge in STT-RAM caches, i.e., the high error rate in the write operations. Prior to introducing the proposed algorithm, called *Least Error Rate (LER)*, in the following, we first investigate the importance of the write error rate as well as the contribution of transition directions in total write error rate. Then, the LER replacement algorithm is presented in detail.

The switching operation of STT-RAM cells has a stochastic characteristic [7]. This characteristic makes the STT-RAMs write operations error prone. Considering the stochastic switching, the probability of correct write operation is increased with increasing in the amplitude and duration of the write pulse. The required amplitude and duration of the write pulses for switching the MTJ cells from 0 to 1 is significantly higher than switching MTJ from 1 to 0. Thus, for a predetermined amplitude and duration of the write pulse in STT-RAM caches the error rate of STT-RAM write operation is asymmetric by the direction of transitions [8].

Fig. 1 shows the distribution of 0 to 1 transitions and 1 to 0 transitions and their contributions in write errors. According to [8], the error rate in MTJ switching from 0 to 1 is about two orders of magnitude higher than MTJ switching from 1 to 0. On the other hand, Fig. 1 depicts that the contribution of 0 to 1 and 1 to 0 transitions in total STT-RAM cells switching are almost the same. Significantly higher error rate in one transition direction over the other direction when the numbers of transitions in these directions are the same implies that the majority of write errors is due to 0 to 1 transition. This is illustrated in Fig. 1 for the L2-cache under the workload of SPEC CPU2006 benchmark suite. As depicted, the contribution of 0 to 1 transitions in total error rate is 100$\times$ larger than the contribution of 1 to 0 transitions.

For a write operation, the number of transitions and the contribution of each transition direction depend on the bit-wise difference between the contents of the incoming block and the target cache line. On the other hand, the cache replacement algorithm determines the location of the incoming block in the cache set, i.e., the target cache line. Therefore, we can manipulate the cache replacement algorithm to deal with the write error rate.

To address the asymmetry in the error rate of STT-RAM cell transitions and overcome the high error rate of STT-RAM caches, we propose LER replacement algorithm for STT-RAM caches. Unlike the well-known LRU algorithm that replaces

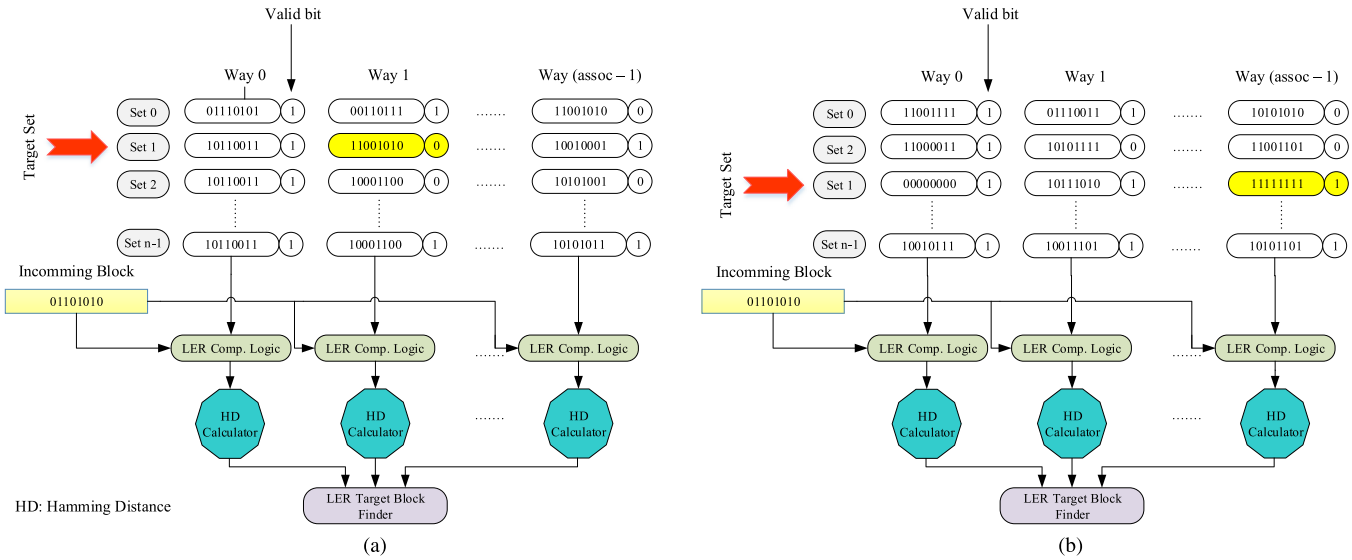

Fig. 2. Two different scenarios to explore the functionality of LER replacement algorithm: (a) when invalid lines available and (b) when no invalid line available.

the cache lines based on their reference history, LER considers the number of transitions and their direction to select the target line for error rate minimization. To this aim, a cache lines that imposes the minimum number of 0 to 1 transitions when overwritten by the incoming block is selected. Since the contribution of 1 to 0 transition in the error rate is negligible in comparison with 0 to 1 transition, LER minimizes the 0 to 1 transition and ignores 1 to 0 transitions. Algorithm 1 shows the details of LER for determining the target block to be replaced by the incoming block.

Algorithm 1 LER Replacement Algorithm

Input: new_block
Output: target_block
1: **start**
2: target_set = **extractSet**(new_block)
3: find_target = 0
// Searching the target_set to find an invalid block
4: **for** block_index = assoc **to** 0 in target_set
5: **if** block[block_index] is invalid and 0to1_tran(block[block_index]) is minimum
6: target_block = block[block_index]
7: find_target = 1
8: **end if**
9: **end for**
10: // Searching among valid blocks
11: **If** find_target = 0 // target_set has no invalid block
12: **for** block_index = assoc **to** 0 in target_set
13: if 0to1_tran(block[block_index]) is minimum
14: target_block = block[block_index]
15: find_target = 1
16: **end if**
17: **end for**
18: **end if**
19: **Return** target_block
20: **end**

To further clarify the LER replacement policy, the Algorithm 1 is traversed here according to the two scenarios depicted in Fig. 2. For each scenario, Algorithm 1 is started with the contents of the new block as an input of the algorithm. At the first step, the set that the new block should be written into (target set) is determined according to the address of the requested block. Then, based on the availability of the invalid block(s) in the target set, the algorithm decides on the target block in two ways. Considering the scenario in Fig. 2(a), the target set has some invalid lines. In this case, LER compares the contents of the invalid lines with the incoming block and the line with the minimum number of 0 to 1 transitions is selected as the target. In scenario depicted in Fig. 2(b), there is no invalid line in the set and LER compares the incoming block with all lines in the set. The target is again the line with the minimum number of 0 to 1 transitions.

Since the STT-RAM cells are used in L2 and last-level caches, the LER replacement algorithm is assumed to be applied in these levels. LER moves the victim cache line selection operation from tag comparison logic to line comparison logic. In other words, unlike the conventional cache replacement algorithms that select the target line, i.e., victim line to be replaced, by comparing the tag array contents, LER compares the data lines contents to select the target. It is noteworthy that the comparison operation between all cache lines in the selected set and the incoming block is performed in parallel with L2 miss operations.

As the comparison operations of LER does not affect the critical path latency of accessing a block in L2 cache, it does not directly impose any performance overhead to the system. On the other hand, the decisions of LER on evicting the cache lines (blocks) may be impose extra L2 cache misses to the system in comparison with traditional LRU, which may result in performance overhead, indirectly. The comparison logic designed for LER to determine the minimum number of 0 to 1 transitions is depicted in Fig. 3. As illustrated, the comparison logic is simply a combinational circuit composed of one level of NOT gates

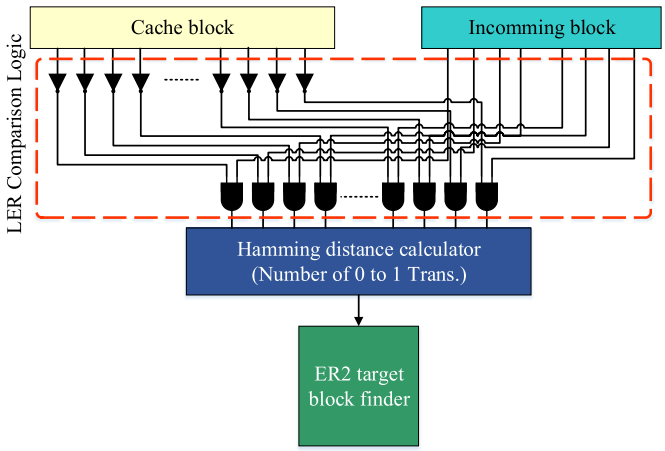

Fig. 3. Abstract of the designed circuit to implement LER.

TABLE I
SIMULATION CONFIGURATION

	PARAMETER	VALUE
	Frequency	1 GHz
	CPU Model	arm_detailed
L1 Caches	Associativity	4-way
	Size	32-KB
	Replacement Algorithm	LRU
	Block Size	64-B
L2 Cache	Associativity	16-way
	Size	1-MB
	Replacement Algorithm	LRU and LER
	Block Size	64-B

and one level of AND gates. The comparison logic performs the bit-wise AND of the incoming block and one's complement of each cache line. Our experimental results using Synopsis Design Compiler [14] with Nangate 45 nm technology node library [15], show that LER has no effect on the critical path latency of the cache access.

III. SIMULATION SETUP AND RESULTS

To evaluate LER, gem5 [12] is used as the simulation platform and SPEC CPU2006 is used as the workload. The simulations are performed for one billion instructions. The energy consumption information of STT-RAM cache are retrieved from NVSim [16]. The detail of the gem5 configuration is shown in Table I. The simulations are performed on the detailed model of the ARM processor provided by gem5. As shown in Table I, two different configurations for L2-cache algorithm are considered in the simulations, i.e., LRU and LER. We consider the LRU algorithm as the baseline to evaluate the efficiency of LER. It should be noted that we consider *Early Write Termination (EWT)* method [17] in our simulations. In the following, the effects of LER on the write error rate of the STT-RAM cache as well as on the performance are discussed.

The error rate reduction in LER is proportional to the reduction in the number of 0 to 1 transitions. Fig. 4 shows the average number of 0 to 1 transitions in cache lines per write operation. As depicted, LER reduces the number of 0 to 1 transitions by 49% in comparison with LRU, on average. In the best case, LER reduces the transitions by 93% in *lbm* workload. This reduction in the worst-case, i.e., *gobmk* workload, is 23%.

The efficiency of LER in reducing the error rate depends on the data value patterns of the workloads in the cache lines. For workloads with higher diversity in the contents of the cache lines, the incoming block incurs higher variation in the number of transitions when written into various lines. In other words, there is a large gap in the number of transitions between the best line, i.e., a line with minimum transitions, and the line selected by LRU, i.e., a line without considering the number of transitions. In this case, selecting a line by LER that imposes the minimum transitions can significantly reduce the error rate in comparison with the line selected by LRU. The contribution of this data pattern in *namd*, *libquantum*, and *lbm* workloads is higher than the other workload. LER reduces the 0 to 1 transitions by more than 60% in these workloads.

As another observation, the LER efficiency decreases when the similarity in the contents of the cache lines increases. In this case, the number of transitions incurred in a line selected by LER is not significantly lower than that of a line selected by LRU. The contribution of this data pattern in *bzip2*, *xalancbmk*, *gobmk*, and *milc* workloads is higher than the other workload. The reduction in 0 to 1 transitions in these workloads is less than 30%.

As mentioned, the error rate in the write operations in STT-RAM caches is directly affected by the number of 0 to 1 transitions on the cache lines. Here, we calculate the error rate in write operations for LRU and LER replacement algorithms. Each write operation in STT-RAM L2-cache is performed in the granularity of one cache line, i.e., 512-bit. Since the write error rate of 1 to 0 transitions is ignorable in comparison with 0 to 1 transitions, the probability of a correct write operation can be estimated according to (1)

$$P_{\text{correct write}} = (1 - 0 \text{ to } 1_{\text{error rate}})^{\#of\ 0\ to\ 1\ \text{trans.}} \quad (1)$$

The probability of an erroneous write operation can be derived from (1) and is given in (2). We consider (2) as the error rate of a cache line

$$P_{\text{erroreous write}} = 1 - P_{\text{correct write}}. \quad (2)$$

Fig. 5 depicts the error rate in cache lines per write operation for both LRU and LER replacement algorithms. The error rate of a STT-RAM cell for 0 to 1 transition is assumed to be 5×10^{-5} [8]. According to Fig. 5, LER reduces the error rate in cache lines by about 2×, on average. Same as the reduction in the number of transitions, the error rate reduction in LER is affected by the workloads. As an example, LER decreases the error rate of write operation by one order of magnitude in *lbm* workload; whereas, as the worst-case scenario, the error rate of the *gobmk* is only decreased about 33%. As another observation, there is a large variation in the absolute value of the error rate in different workloads. For example, the error

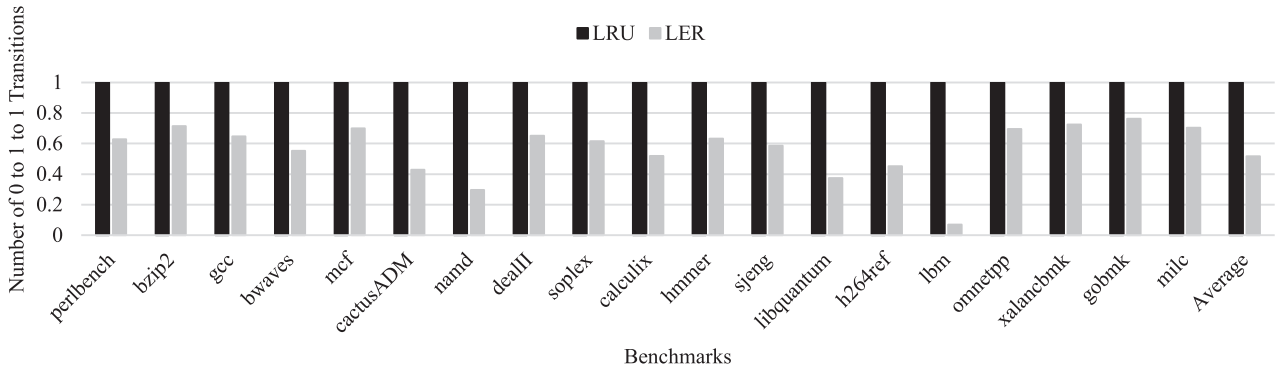

Fig. 4. Total number of 0 to 1 transitions per WRITE operation in LER normalized to LRU.

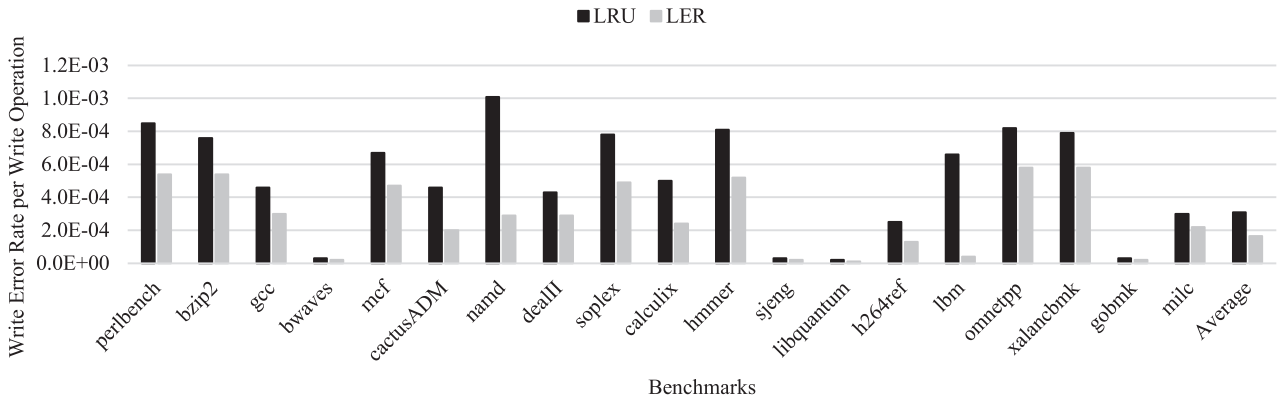

Fig. 5. Write error rate in cache lines per write operation.

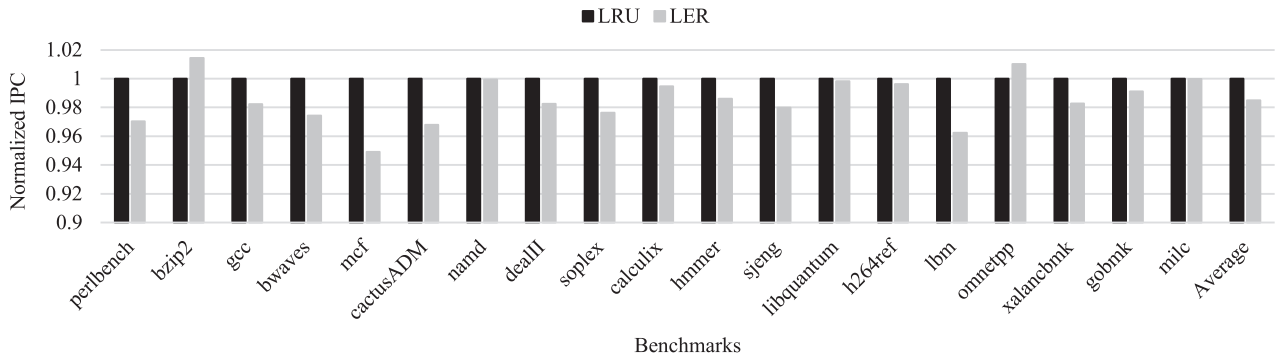

Fig. 6. Instruction per clock in LER normalized to LRU.

rates in *perlbench*, *bzip2*, *namd*, *soplex*, *hmmer*, *omnetpp*, and *xalancbmk* workloads are more than one order of magnitude higher than *bwaves*, *sjeng*, *libquantum*, and *gobmk* workloads in LRU algorithm. This implies that the number of ones in the workloads with higher error rate is larger than that in other workloads.

The average of 2× reduction in the error rate by the LER algorithm is a promising result, which enables the opportunity to significantly reduce the energy consumption and write latency of the STT-RAM caches. In other words, because of the stochastic switching characteristic of STT-RAMs which leads to high error rate, each write operation in the STT-RAM caches is consisted of several *write-read-verify* iterations [11]. These write-read-verify iterations improve the probability of the correct write in more data bits and these iterations continue until the block is correctly written. Therefore, 2× reduction in the error rate achieved by LER reduces the total iterations of write-read-verify to half, whereas the probability of the correct write operation remains unchanged. Halving the number of write-read-verify iterations by LER algorithm results in reducing the latency and energy consumption in write operations into STT-RAM caches. Furthermore, it is noteworthy to say that LER is completely orthogonal to previous studies that try to reduce the write error rate across the STT-RAM caches, Thus, LER can be used in conjunction with these methods to achieve further improvement in write error rate.

In the following, we investigate the performance of the system and the L2 cache energy consumption in LER and LRU, when assuming all write operations are error-free and no write-read-verify iteration is required. Fig. 6 depicts the *Instruction per Clock (IPC)* of the system in LER algorithm normalized to LRU algorithm. In comparison with LRU, on average IPC is

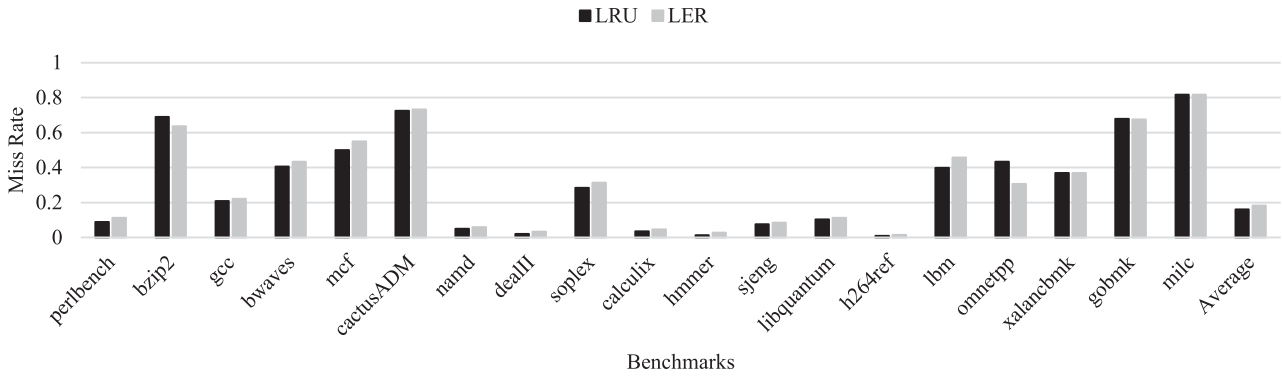

Fig. 7. Miss rate of L2 cache in different replacement policies.

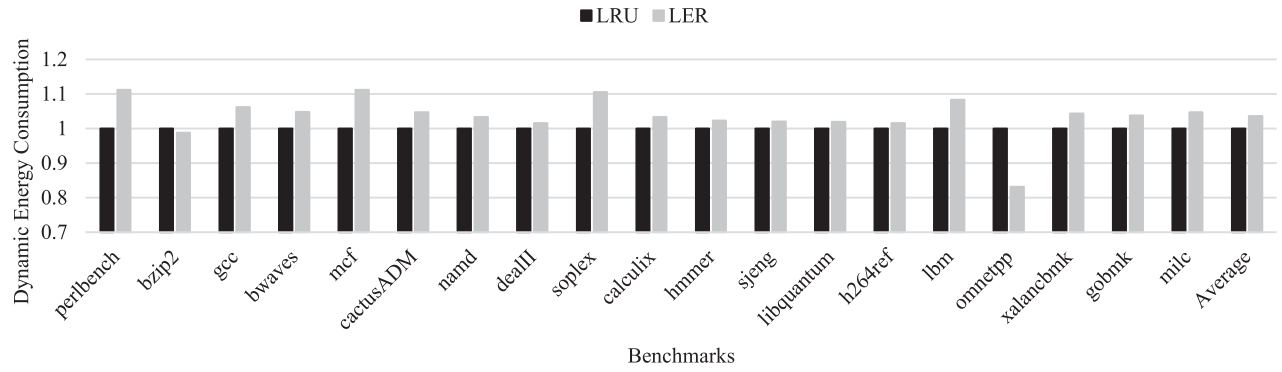

Fig. 8. Dynamic energy consumption in LER normalized to LRU.

reduced by 1.5% in LER. In the worst-case, LER reduces IPC by about 5% in *mcf* workload, whereas in some workloads like *bzip2* and *omnetpp* IPC is even increased in comparison with LRU. The miss rate of L2-cache for LRU and LER is depicted in Fig. 7. LER imposes 3% extra miss rate in comparison with LRU, on average.

Fig. 8 depicts the dynamic energy overhead of LER in comparison with LRU. While the requests hit in the L2 cache do not impose any dynamic energy overhead in LER, but the requests missed in the L2 cache enable the LER controlling circuits, which lead to imposing dynamic energy overhead to the system. In other word, for each miss occurrence, LER should read all of the cache lines (blocks) in the target set to find the optimum cache line for eviction. This operation imposes extra energy consumption to the system in comparison with LRU. It should be noted that, LER eliminates several peripheral circuits and storage elements (e.g. age bits that are assigned to the lines of each set) required for LRU. Thus, the total static energy savings that achieved from omitting these elements overcome the static energy consumption of LER peripheral circuits. In this regard, we do not consider any static energy overhead for LER in comparison with LRU. Furthermore, for the same reason, we can claim that LER does not impose any area overhead to the system in comparison with LRU. It is illustrated in Fig. 8 that on average, LER consumes 3.5% higher dynamic energy compared with LRU, when ignoring the energy dissipated by the extra write-read-verify iterations in LRU due to its higher error rate. In the worst case, LER imposes about 11% dynamic energy overhead for *perlbench*, *mcf* and *soplex*. In the best case, LER may also overcome LRU in term of dynamic energy consumption for the benchmarks like *bzip2* and *omnetpp*. Generally, in these kinds of workloads the lower miss rates of LER in comparison with LRU (as it is shown in Fig. 7) leads to lower dynamic energy consumption for LER (1% for *bzip2* and 17% for *omnetpp*).

IV. Conclusion

This paper proposes a new cache replacement policy, called *Least Error Rate (LER)*, for emerging STT-RAM caches. LER focused on the high write error rate in STT-RAM caches, which is basically originated from 0 to 1 transitions. The main idea is to place the incoming block in a cache line that incurs the minimum error rate in the write operation. This is done by comparing the contents of the incoming block with lines in a set. The simulation results show that compared with LRU algorithm, LER reduces the error rate by 2× with about 1.4% and 3.6% performance and dynamic energy consumption overheads, respectively. Moreover, LER impose no area overhead to system.

References

[1] H. Jingtong, C. J. Xue, Q. Zhuge, W. Tseng, and E. Sha, "Towards energy efficient hybrid on-chip scratch pad memory with non-volatile memory," in *Proc. IEEE 14th Int. Conf. DATE*, Grenoble, France, Mar. 14–18, 2011, pp. 1–6.

[2] [Accessed Jan. 15, 2016]. [Online]. Available: http://www.itrs.net/

[3] A. M. H. Monazzah, H. Farbeh, S. G. Miremadi, M. Fazeli, and H. Asadi, "FTSPM: A fault-tolerant scratchpad memory," in *Proc. IEEE/IFIP 43rd Annu. Int. Conf. DSN*, Budapest, Hungary, Jun. 24–27, 2013, pp. 1–10.

[4] S. G. Ghaemi, A. M. H. Monazzah, H. Farbeh, and S. G. Miremadi, "LATED: Lifetime-aware tag for enduring design," in *Proc. IEEE 11th Int. EDCC*, Paris, France, Sep. 7–11, 2015, pp. 97–107.
[5] H. Farbeh, H. Kim, S. G. Miremadi, and S. Kim, "Floating-ECC: Dynamic repositioning of error correcting code bits for extending the lifetime of STT-RAM caches," *IEEE Trans. Comput.*, doi: 10.1109/TC.2016.2557326, to be published.
[6] C. W. Smullen, V. Mohan, A. Nigam, S. Gurumurthi, and M. R. Stan, "Relaxing non-volatility for fast and energy-efficient STT-RAM caches," in *Proc. IEEE 17th Int. Symp. HPCA*, San Antonio, TX, USA, Feb. 12–16, 2011, pp. 50–61.
[7] K. Wang *et al.*, "Yield and reliability improvement techniques for emerging nonvolatile STT-MRAM," *IEEE J. Emerging Sel. Topics Circuits Syst.*, vol. 5, no. 1, pp. 28–39, Mar. 2015.
[8] B. Xiuyuan, Z. Sun, H. Li, and W. Wenqing, "Probabilistic design methodology to improve run-time stability and performance of STT-RAM caches," in *Proc. IEEE/ACM 31st ICCAD*, San Jose, CA, USA, Nov. 5–8, 2012, pp. 88–94.
[9] L. Yahya *et al.*, "Switching circuit for spin transfer torque MRAM and logic," *IEEE Trans. Magn.*, vol. 48, no. 9, pp. 2403–2406, Sep. 2012.
[10] W. Wujie *et al.*, "CD-ECC: Content-dependent error correction codes for combating asymmetric nonvolatile memory operation errors," in *Proc. IEEE 32nd ICCAD*, San Jose, CA, USA, Nov. 18–21, 2013, pp. 1–8.
[11] H. Sun, C. Liu, N. Zheng, T. Min, and T. Zhang, "Design techniques to improve the device write margin for MRAM-based cache memory," in *Proc. 21st GLVLSI*, Lausanne, Switzerland, May 2–4, 2011, pp. 97–102.
[12] N. Binkert *et al.*, "The gem5 simulator," *ACM SIGARCH Comput. Archit. News*, vol. 39, no. 2, pp. 1–7, 2011.
[13] J. L. Henning, "SPEC CPU2006 benchmark descriptions," *ACM SIGARCH Comput. Archit. News*, vol. 34, no. 4, pp. 1–17, 2006.
[14] [AccessedJan.15,2016]. [Online]. Available: http://www.synopsys.com
[15] S. Nangate, *California (2008). 45* nm *Open Cell Library*, 2008, [Accessed Jan. 15, 2016]. [Online]. Avaliable: http://www.nangate.com
[16] X. Dong, C. Xu, Y. Xie, and N. P. Jouppi, "NVSim: A circuit-level performance, energy, and area model for emerging nonvolatile memory," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 31, no. 7, pp. 994–1007, 2012.
[17] P. Zhou, B. Zhao, J. Yang, and Y. Zhang, "Energy reduction for STT-RAM using early write termination," in *Proc. 11th ICCAD*, San Jose, CA, USA, Nov. 2–5, 2009, pp. 264–268.

Amir Mahdi Hosseini Monazzah (S'14) received the B.Sc. degree in computer engineering from Islamic Azad University (South Tehran branch), Tehran, Iran, in 2009 and the M.Sc. degree in computer engineering from Sharif University of Technology (SUT), Tehran, in 2012. He is currently working toward the Ph.D. degree in computer engineering with SUT.

Since 2010, he has been a member of the Dependable Systems Laboratory, Department of Computer Engineering, SUT. His research interests include investigating the reliability challenges of emerging nonvolatile memories, fault-tolerant hybrid memory hierarchy design, designing reliable software for unreliable hardware, designing dependable embedded systems, and reliability challenges in multicore systems.

Hamed Farbeh (S'12) received the B.Sc. and M.Sc. degrees in computer engineering from Sharif University of Technology (SUT), Tehran, Iran, in 2009 and 2011, respectively. He is currently working toward the Ph.D. degree in computer engineering with SUT.

From 2014 to 2015, he was with the Embedded Computing Laboratory, Korea Advanced Institute of Science and Technology, Daejeon, Korea, as a Visiting Researcher. Since 2007, he has been a member of the Dependable Systems Laboratory, Department of Computer Engineering, SUT. His current research interests include fault-tolerant embedded system design, reliable memory hierarchy, and reliability challenges in emerging memory technologies.

Seyed Ghassem Miremadi (SM'07) received the M.Sc. degree in applied physics and electrical engineering from Linkping Institute of Technology, Linkping, Sweden, and the Ph.D. degree in computer engineering from Chalmers University of Technology, Gothenburg, Sweden.

In 1996, he initiated the Dependable Systems Laboratory with the Sharif University of Technology (SUT), Tehran, Iran, as fault-tolerant computing is his specialty, and has chaired the laboratory since then. He and his group have done research in physical, simulation-based, and software-implemented fault injection, dependability evaluation using hardware description language models, fault-tolerant embedded systems, fault-tolerant networks-on-chip, fault-tolerant real-time systems, and fault-tolerant storage systems. From 1997 to 1998, he was the Education Director; from 1998 to 2002, the Head; from 2002 to 2006, the Research Director; and from 2009 to 2010, the Director of the Hardware Group with the Department of Computer Engineering, SUT. From 2003 to 2010, he was the Director of the Information Technology Program with SUT International Campus-Kish Island, Kish, Iran. From 2010 to 2012 and since 2014, he has been the Vice President of Academic Affairs at SUT, where he is currently a Professor of computer engineering.

Dr. Miremadi is a Senior Member of the IEEE Computer and Reliability Societies. He served as the General Co-Chair of the 13th International CSI Computer Conference in 2008, the Executive Chair of the International Conference on Engineering Education in 2013, and the General Co-Chair of the International CSI Symposium on Real-Time and Embedded Systems and Technologies in 2015. He serves as the Editor of the *Scientia Transactions on Computer Science and Engineering*.