np-ECC: Nonadjacent Position Error Correction Code for Racetrack Memory

Xiaoyang Wang, Chao Zhang, Xian Zhang, Guangyu Sun
Center for Energy-efficient Computing and Applications
Peking University, Beijing, 100871, China
{yaoer, zhang.chao, zhang.xian, gsun}@pku.edu.cn

ABSTRACT

Racetrack memory is a promising non-volatile memory because of its ultra-high storage density. The data are stored along the tape-like cell, where a "shift" operation is used to move the data in a cell back and forth to be accessed. Shift operations suffer from "position error", where the shift distance is incorrect. Previous work solved the error by position error correction code (p-ECC). However, a bit error within the p-ECC bits will fail the correction mechanism. To protect p-ECC bits from bit errors, we propose a new mapping method for p-ECC, called nonadjacent position error correction code (np-ECC) in this paper. Evaluation shows significant reduction on correction mechanism failure rate.

Keywords

Racetrack memory; Error Correction Code; Position Error

1. INTRODUCTION

Racetrack memory has attracted great attention of researchers because of its ultra-high storage density [6] [5] [4] [3]. It is composed of a tape-like stripe as well as several access ports. Each stripe contains a lot of domains, which are isolated by domain walls and programmed to store bits [2]. The access ports are connected to the stripe to perform read, write, and shift operations, respectively.

The shift current drives the domains to move left or right to feed the access port with the proper data cell [8] [1]. However, an unreliable shift may result in a "position error", which means the ports will miss the target domain and access the wrong bits. It reduces the mean-time-to-failure (MTTF) to an intolerable level.

The position error correction code (p-ECC) [7] is an efficient approach to address the problem. It takes advantage of an extra area filled with circulations of "1100" and two adjacent ports. The p-ECC bits are stored beside the other normal data in the same strip. When the mechanism works, those ports access two p-ECC bits simultaneously, and the expected two bits are calculated based on the shift distance. If the shift distance is incorrect, a shift error is detected.

Nanoarch '16, July 18-20, 2016, Beijing, China

DOI: http://dx.doi.org/10.1145/2950067.2950082

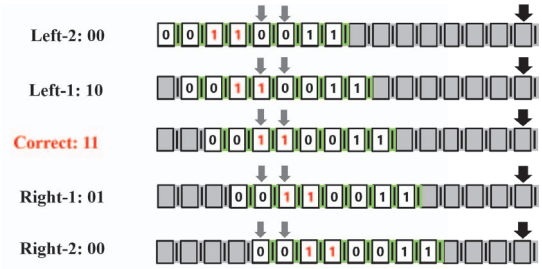

Figure 1: p-ECC method: The expected number is 11(c); 10(b) and 01(c) means a 1-step error occurs; 00(a,e) means a 2-step error occurs

Figure 1 shows a design for Single step Error Correction and Double steps Error Detection (SECDED). The expected bits after shift should be "11". If we get "01" or "10", we detect a 1-step shift error. To correct, the racetrack is need to be shifted left or right by one step. If the detected bits are "00", we only detect a 2-step error.

However, when a bit error occurs among the p-ECC bits, this method may be fooled and thus fails, which means that the p-ECC bits also need a bit-error detection mechanism just as the general ECC for the common data.

When a bit error turns the p-ECC bits "1100" into "1000", the ports are supposed to get "00" instead of "10". As shown in Figure 2, to correct the detected shift error (fooled), the p-ECC need to shift one step right to get the bits "10".

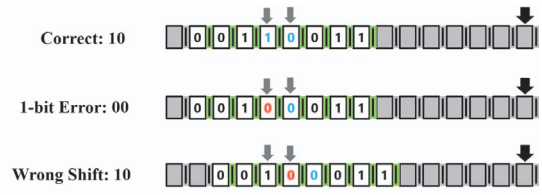

Figure 2: 1-bit error in p-ECC: (a) The expected result; (b) 1-bit error occurs which cheats the mechanism; (c) The p-ECC performs the wrong shift

To mitigate the problem above, we introduce the nonadjacent position error correction code (np-ECC), which does not induce performance penalty theoretically. We only demonstrate np-ECC for single error correction, but this method is scalable to higher error correction requirement.

2. DESIGN OF NP-ECC

p-ECC is not working in some scenarios. If the two ports travel along the p-ECC bit area and read out the bits at every step, the **read sequence** will be the loop of $\{11, 10, 00, 01\}$. When "1100" turns into "1000", the sequence changes into $\{\mathbf{10}, \mathbf{00}, 00, 01, \dots\}$, which misleads the progress.

The key point is that in the p-ECC design, a 1-bit error affects two adjacent statues in the sequence, and the nonadjacent p-ECC method (np-ECC) is designed to avoid this. We call it "nonadjacent", because the ports are seperated, which is diffenent from the p-ECC. To make np-ECC compatible with p-ECC, the bits read out at different locations should be the same as original p-ECC, which form a loop of 2-bit cyclic numbers.

Figure 3 shows an example of the np-ECC design, whose code is the repetition of "11001001". To be more clearly, we can switch the ports' order every four steps. Consistent with p-ECC, the bits read out from the ports when the domains move to left are still $\{11, 10, 00, 01, \dots\}$. Therefore, the conventional p-ECC mapping can be converted to np-ECC mapping without overhead.

Figure 3: np-ECC design

With the protection of np-ECC, the correction process follows the same way of p-ECC: It shifts the data again, trying to get the target p-ECC bits. Because the bits found in the new position reached by correction process are not affected by the bit error, it is not consistent with the expectation of the correction process.

When the correction attempt fails to get the target bits, it finds that it is a bit error, instead of a position error.

3. ANALYSIS AND DISCUSSIONS

This section analyzes the advantages of the np-ECC method from the aspects of performance, scalability, and reliability.

Changing from p-ECC to np-ECC does not require any extra ports or calculation. It means that np-ECC does not cause any performance degradation and thus is not worse than the original p-ECC design.

Compared with p-ECC, np-ECC has better scalability. Due to the adjacent ports, the read sequence for p-ECC $\{11, 10, 00, 01\}$ has the feature that the prefix of the latter element should equal to the suffix of the former one, while np-ECC does not have such limitation. For instance, if we want to detect an error longer than 2-step, three or more ports are required. Then we can define the read sequence as the loop of $\{000, 001, ..., 111\}$, which is the binary representation of 0 to 7, and the np-ECC bits will be the repetitions of $< 00001111, 00110011, 01010101 >$. It is still convenient to predict the correct bits for a given shift step. This simplifies the design of checking circuit. We may also choose the set $\{000, 011, 101, 110\}$ as the read sequence, whose element has even "1"s. When an odd-"1"s situation is observed, we can detect a bit error immediately. It is also simple to predict any steps' shift result, as we can use the two most significant bits as the counter.

To describe the reliability of the np-ECC method, we compare the p-ECC method with it. We assume that there are p 1-bit errors, and q 1-step position errors per second.

Let's consider the cases that mislead the p-ECC. If there is only a 1-step error, the ports will be confused if and only if they stop at a $0\underline{01}1$ or $1\underline{10}0$, the probability of which is $1/2$, while the expectation of a 1-bit error occurs among them is $2p(1-p)$. Therefore, the probability of this situation is $p(1-p)$. There are other cases such as a 1-step error along with a 1-bit error, or two 1-bit errors happen together. It is easy to prove that those probablities have a small magnitude compared with $p(1-p)$, and thus can be ignored. So the p-ECC method will fail about p times per second.

The scenarios that np-ECC fail are more complex. A 1-bit error along with a 1-step error is able to confuse it. If the read sequence is $\{11, 10, 00, 01\}$, a 2-bit error will also lead to an undeterministic case, where the probability is $max\{p^2, pq\}$. However, if the read sequence is reordered to $\{11, 00, 10, 01\}$, np-ECC is able to tolerate a 2-bit error, and the probability will be $max\{p^3, pq\}$. In conclusion, the np-ECC is at least $1/q$ more reliable than p-ECC.

4. CONCLUSION

Position errors in the shift operations of racetrack memory can be detected and corrected by p-ECC method. However, a bit error occurs in p-ECC region will fail the method. Our work presents a new mapping method called Nonadjacent Position Error Correction Code (np-ECC), to protect p-ECC from bit errors. This method adds no overhead both on area and calculation. It shows good scalability and improved reliability compared with original p-ECC. With the help of np-ECC, the racetrack memory can be protected against with position errors and bit errors at the same time.

5. REFERENCES

[1] W. Kang, Y. Huang, X. Zhang, Y. Zhou, W. Lv, and W. Zhao. Skyrmions as compact, robust and energy-efficient interconnects for domain wall (dw)-based systems. *CoRR*, 2016.

[2] S. S. Parkin, M. Hayashi, and L. Thomas. Magnetic domain-wall racetrack memory. *Science*, 320(5873), 2008.

[3] Z. Sun, X. Bi, W. Wu, S. Yoo, and H. Li. Array organization and data management exploration in racetrack memory. *TC*, PP(99), 2014.

[4] R. Venkatesan, V. Kozhikkottu, C. Augustine, A. Raychowdhury, K. Roy, and A. Raghunathan. Tapecache: A high density, energy efficient cache based on domain wall memory. In *ISLPED*. ACM, 2012.

[5] R. Venkatesan, S. Ramasubramanian, S. Venkataramani, K. Roy, and A. Raghunathan. Stag: Spintronic-tape architecture for gpgpu cache hierarchies. In *ISCA*, June 2014.

[6] C. Zhang, G. Sun, W. Zhang, F. Mi, H. Li, and W. Zhao. Quantitative modeling of racetrack memory, a tradeoff among area, performance, and power. In *ASP-DAC*, 2015.

[7] C. Zhang, G. Sun, X. Zhang, W. Zhang, W. Zhao, T. Wang, Y. Liang, Y. Liu, Y. Wang, and J. Shu. Hi-fi playback: Tolerating position errors in shift operations of racetrack memory. In *ISCA 2015*, June 2015.

[8] W. Zhao, Y. Zhang, H.-P. Trinh, J.-O. Klein, C. Chappert, R. Mantovan, A. Lamperti, R. Cowburn, T. Trypiniotis, M. Klaui, J. Heinen, B. Ocker, and D. Ravelosona. Magnetic domain-wall racetrack memory for high density and fast data storage. In *ICSICT*, Oct 2012.