ECTI Transactions on Computer and Information Technology
Journal homepage: https://ph01.tci-thaijo.org/index.php/ecticit/
Published by the ECTI Association, Thailand, ISSN: 2286-9131

Two-Dimensional Error Correction Code for Spin-Transfer Torque Magnetic Random-Access Memory (STT-MRAM) Caches

Chatuporn Duangthong[1], Pornchai Supnithi[2] and Watid Phakphisut[3]

ABSTRACT

Spin-Transfer Torque Magnetic Random-Access Memory (STT-MRAM) is an emerging nonvolatile memory (NVM) technology that can replace conventional cache memory in computer systems. STT-RAM has many desirable properties such as high writing and reading speed, non-volatility, and low power consumption. Since the cache requires a high speed of writing and reading speed, a single-error correction and double error detection (SEC-DED) are applicable to improve the reliability of the cache. However, the process variation and thermal fluctuation of STT-MRAM cause errors. For example, writing '1' bits has more errors than writing '0' bits. We then design the weight reduction code to reduce the error caused by writing '1' bits. Moreover, the performance of an SEC-DED code is improved by constructing an SED-DED code as the product code. The simulation results demonstrate that the two-dimensional error correction code consisting of product code and weight reduction code is roughly 5.67×10^{-4} lower than the SEC-DED code when the error rate of writing '1' bits is equal to 6×10^{-3}.

Article information:

Keywords: STT-MRAM, SEC-DED Code, Error Correction Codes

Article history:
Received: November 27, 2021
Revised: February 26, 2022
Accepted: March 5, 2022
Published: June 18, 2022
(Online)

DOI: 10.37936/ecti-cit.2022163.246903

1. INTRODUCTION

Nowadays, static random-access memory (SRAM) based on cache memory is commonly utilized in computer systems. SRAM has several limitations, including high power leakage and cell instability [1, 2] when memory technologies have been down-scaled. As a result, spin-transfer torque magnetic random-access memory (STT-MRAM), which offers several advantages such as near-zero power leakage, high density, scalability, and nonvolatility [3, 4], is a potential technology for on-chip cache [5]. Even though STT-MRAM outperforms SRAM in several ways, the reliability of STT-MRAM suffers from deterioration caused by process variation and thermal fluctuation [6, 7]. These noise sources affect the STT-MRAM, causing asymmetric errors. For example, writing '1' bits has more errors than writing '0' bits [8], [9], [10], [11].

There are two major approaches available to improve the reliability of the STT-MRAM cache: improving the write error rate and improving the read error rate. By upgrading the circuits or employing signal processing techniques, the write error rate can be minimized. [14] presented the dynamic write latency approach at the architectural level, which reduces the write error rate and enhances overall system performance by up to 15.4%. [15] proposed a REACT approach for reducing read and write errors. REACT makes an effort to decrease the overall amount of '1s' block data patterns. Thus the REACT can reduce read disturbance and write failure rates by 17.4% and 18.2%, respectively. To minimize overall write latency and write error rate, [16] presented static and dynamic circuit-level approaches. A 512 kB L2-cache of a microprocessor can enhance its performance by 11% and reduce total write latency by 71%.

It is well-known that error correction codes (ECCs) have been implemented in computer memory such as static random access memory (SRAM) [17], dynamic random access memory (DRAM)[18], and Flash [19] to reduce error rates and enhance data reliability. Therefore, several works aim to design an advanced error correction code for the upcoming STT-MRAM. [20] used multi-bit error correction combined with

[1,2,3] The authors are with School of Engineering, King Mongkut's Institute of Technology Ladkrabang, Bangkok 10520, Thailand, E-mail: 58601143@kmitl.ac.th, pornchai.su@kmitl.ac.th and watid.ph@kmitl.ac.th

DRAM-style refreshing to reduce errors and provided a methodology for calculating the adequate level of correction. In [21], a novel error protection method called Sanitizer was proposed. It can enhance performance by 1.22x while end-to-end system energy use was reduced by 22%. The Bose-Chaudhuir-Hocquenghem (BCH) code for STT-MRAM was investigated in [22]. In [12], Floating-ECC architecture was proposed to enhance the lifetime of the L2 and L3 caches. By relocating the ECC bits inside the line regularly, the writing activity of the ECC portion is uniformly spread throughout the cache line. [23] proposed novel rate-compatible protograph LDPC (RCP-LDPC) codes for STT-MRAM to rectify memory cell faults and minimize raw bit error rate (BER) variability.

Since the STT-MRAM cache memory requires a high speed of writing and reading, a single-error correction and double-error detection (SEC-DED) is then applicable to improve the reliability of STT-MRAM cache memory. [24] presented the content-dependent error correction code (CD-ECC), which is comprise of dynamic differential code and SEC-DED code. However, the dynamic differential code can generate error propagation. [13] proposed the ORIENT ECC bit selection technique to decrease susceptibility fluctuation in codewords. Compared to per-word and traditional N-way interleaved bit selection techniques, the ORIENT can reduce write error rates significantly. A low-complexity SEC-DED code was employed in previous work, but only one-bit error was corrected and two bits were detected. SEC-DED and BCH product codes were utilized [22], and while they provide superior performance, they need more complexity. This additional complexity is undesirable for cache memory, which requires fast read and write rates.

Therefore, in this work, we propose to modify an SEC-DED code as a product code to improve its bit error correction. The product code can be viewed as a two-dimensional error correction code. This code is compatible with the STT-MRAM cache since all data blocks in the array are read at the same time. By using product code structure, any error bits detected by the SEC-DED code can be corrected by other SEC-DED codes. Moreover, since the process variation and thermal fluctuation impact the $0 \rightarrow 1$ switching more than $1 \rightarrow 0$ switching, it causes a highly asymmetric channel. To address this issue, we propose a weight reduction code to reduce the number of '1' bits. The concept of the weight reduction code is to reduce '1' bits before writing in the memory cell. First, the amount of '1's in the codewords will be verified. If the block length is greater than half of the data length, the bits in the codeword block are flipped and flag bit '1' is added to the codeword. Otherwise, the codeword block is not flipped and flag bit '0' is added to the codeword. Therefore, the number of '1's in the codeword is decreased to less than or equal to half of the bit length of the data. Then the codeword is stored in the STT-MRAM cache. To read data from the cache, the flag bits are always verified for bit '0' or '1'. If flag bits are '1,' the data bits are flipped. Otherwise, the data bits are not changed.

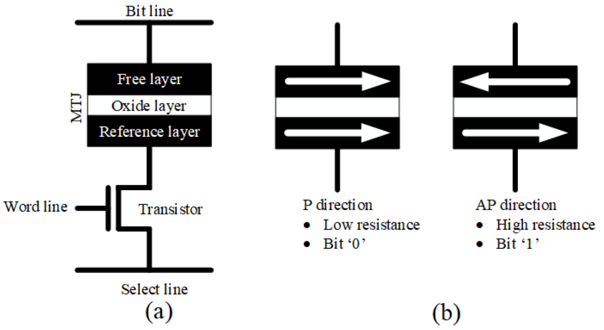

Fig.1: *(a) STT-MRAM cell structure (b) MTJ switching*

The simulation results demonstrate that the two-dimensional error correction consisting of product code and weight reduction code is roughly 5.67×10^{-4} lower than the SEC-DED code when considering the error rate of writing '1' bits to be 6×10^{-3}.

In the rest of this paper, section 2 describes the STT-MRAM channel model, cache architecture, Hamming weight distribution, and SEC-DED code. In section 3, the design of the SEC-DED product code and the design of the weight reduction code will be explained. The simulations and results are discussed in section 4. In the final part, we summarize our results.

2. BACKGROUND

2.1 Channel Model of STT-MRAM

A magnetic tunnel junction (MTJ) and an NMOS transistor are two important components of an STT-MRAM. The memory cell and cell selection operations use the MTJ and NMOS transistors, respectively. Fig. 1 (a) shows the structure of STT-MRAM. One oxide layer is placed between two magnetic layers in the MTJ element. A reference layer, in which the magnetic field direction is fixed, is one of the magnetic layers. The other is a free layer in which the magnetic field direction can be switched between parallel (P) and anti-parallel (AP). The direction of the magnetic field in the free layer determines whether the MTJ element has a high or low resistance. Low resistance is obtained by aligning the magnetic fields of the reference layer and the free layer in the P direction. On the other hand, the anti-parallel (AP) direction, where the reference layer and the free layer are aligned in opposite directions, causes high resistance. As illustrated in Fig. 1 (b), the low and high resistance of the MTJ element can be utilized to correspond to a bit '0' and '1,' respectively.

Process variation and thermal fluctuation are two significant noise sources in the STT-MRAM. The switching current in STT-MRAM varies due to process variations such as the parametric fluctuation of NMOS transistors and MTJ elements. Due to process variation, the distribution function of switching current is represented as a dual-exponential function [6]

$$p_w^{PV}(I_w) = \sigma_w^{PV}\exp\left(-\frac{|I_w - \mu_w^{PV}|}{\sigma_{PV}}\right), \qquad (1)$$

where I_w is a switching current. The standard deviation and mean of switching current are σ_w^{PV} and μ_w^{PV}, respectively. Writing bit is defined as $w \in \{0,1\}$. The Landau–Lifshitz–Gilbert (LLG) equation [6] is used to explain the magnetization dynamics of the MTJ element owing to thermal fluctuation. Since this switching time varies due to thermal fluctuation, the switching current I_w after process variation is mapped to the switching time T_w^{PV}. We will use curve fitting to estimate the current-to-time mapping in this paper. The approximated function of transition 1 → 0 is expressed by

$$T_0^{PV}(I_0) = 1.565\times10^8 - 1.039\times10^8 I_0 - 2.018\times10^7 I_0^2, \qquad (2)$$

where the switching current I_0 is a value between -400 to -130 μA. The estimation function of transition 0 → 1 is given by

$$T_1^{PV}(I_1) = 1.08\times10^8 + 7.432\times10^7 I_1 - 1.301\times10^7 I_1^2, \qquad (3)$$

where the switching current I_1 is a value between 120 to 400 μA.

The switching time T_w^{PV} is classified as thermal activation, dynamic reversal, and precessional switching in [6]. To switch the magnetic field direction in the free layer for the thermal activation mode, a lengthy switching time ($T_w^{PV} > 10$ ns) is necessary. The dynamic reversal mode requires an intermediate switching time (3 ns $\leq T_w^{PV} \leq$ 10 ns), whereas the precessional switching mode requires a very low switching time ($T_w^{PV} < 3$ ns).

The MTJ requires a longer switching time due to thermal fluctuation. $T_w = T_w^{PV} + \Delta T_w^{TF}$, where ΔT_w^{TF} is the thermal-induced deviation determined by curve σ_w^{TF}/μ_w^{TF} vs. T_w^{PV} in [6]. For the thermal activation mode, the thermal-induced deviation ΔT_w^{TF} is given by

$$\Delta T_w^{TF} = \sigma_w^{TF}(\delta_E - 1), \qquad (4)$$

where δ_E distribution function is obtained from

$$p_{Exp}(\delta_E) = \exp(-\delta_E). \qquad (5)$$

For the precessional switching mode, the thermal-induced deviation ΔT_w^{TF} is computed by

$$\Delta T_w^{TF} = \sigma_w^{TF}(\delta_G), \qquad (6)$$

where the distribution function δ_G is modeled as a Gaussian distribution [6] with

$$p_{Gauss}(\delta_G) = \frac{1}{\sqrt{2\pi}}\exp(-\frac{\delta_G^2}{2}). \qquad (7)$$

For dynamic reversal mode, the thermal-induced deviation ΔT_w^{TF} is obtained by

$$\begin{aligned}\Delta T_w^{TF} = &\frac{\sigma_w^{TF}}{\sqrt{7}}\Big(\sqrt{10 - T_w^{PV}}\,\delta_G + \\ &\sqrt{T_w^{PV} - 3(\delta_E - T_w^{PV})}\Big).\end{aligned} \qquad (8)$$

The switching time $Twrite$ is applied to all STT-MRAM cells when the final switching time is determined. The magnetization is flipped if the applied write switching time T_{write} is longer than the necessary switching time T_w. In contrast, the magnetization is not switched if the applied write switching time T_{write} is less than the necessary switching time T_w. Therefore, the STT-MRAM can be modeled as a discrete memoryless channel. The binary asymmetric channel is shown in Fig. 2. We will use the parameters of the STT-MRAM cache in [24] to model the STT-MRAM channel. The author in [24] models the STT-MRAM channel based on an in-plane MTJ with an elliptical shape of $45nm \times 90nm$ under 45nm PTM model. Therefore, the ratio of $p_{ER,0\to1}$ and $p_{ER,1\to0}$ with $T_{write} = 10ns$ is defined as

$$R_p = \frac{p_{ER,0\to1}}{p_{ER,1\to0}}. \qquad (9)$$

To obtain $p_{ER,0\to1}$ and $p_{ER,1\to0}$, the R_p is defined as 4×10^{-3} in [24]. Then, $p_{ER,0\to1}$ and $p_{ER,1\to0}$ in Fig. 2 can be arbitrary with this ratio.

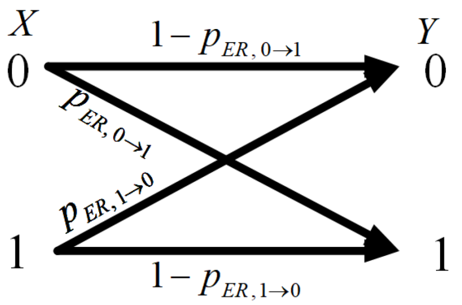

Fig.2: A binary asymmetric channel model of STT-MRAM memory.

2.2 Architecture of STT-MRAM Cache

Fig. 3 depicts the architecture of the STT-MRAM caches [25]. The data are arranged in three dimensions.

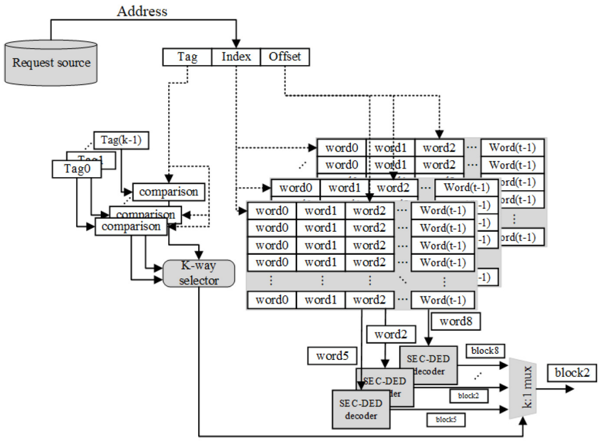

Fig.3: An architecture of STT-MRAM cache [25].

The data page is compose of a two-dimensional data block or word. Each page has several cache lines and each cache line contains several data blocks. The source provides an address consisting of tag, index, and offset to read the data block in the cache. The page number is referred to as a tag. The cache line and data block number are indicated by an index and offset, respectively. Because of parallel cache access, [25], the data blocks in the array are read at the same time. For example, when a data block is requested, the address of the data block is sent to the STT-MRAM cache. The data block on each page is read simultaneously. The tags are compared at the same time. The data blocks are decoded by the SEC-DEC decoder. Then the data block is chosen by the selector and the remaining data blocks are discarded.

2.3 Hamming Weight Distribution of User Data

Since the data blocks stored in the cache have diversity, for a randomly generated data block to cover all data formats, the data block stored in the cache is modeled in different formats. We consider that the data blocks stored in the cache have variety of Hamming weight values. The author of [24] observed that cache line data is typically highly correlated at the block level. That means neighboring blocks frequently contain the same or similar data. This in turn means that the cache line data has the same or similar Hamming weight. In the case of the same Hamming weight, the Hamming weights in the cache line do have not a variation. For the similar Hamming weight case, the Hamming weights in the cache line have some variation. Thus we will generate the data block stored in the same cache line by assigning the Hamming weight.

In this work, the data block is produced at random using a Bernoulli distribution with a probability of bit '1' (p). This probability can be seen as the average of Hamming weight. However, we will use the normalized Hamming weight (δ_{hw}), which is defined by the ratio of the number of '1' in each data block and the data block length. In this work, by using the normalized Hamming weight, the user data in each block can be generated easily. Fig. 4 demonstrates the correlation between two neighboring blocks. The correlation will be 1 if the normalized Hamming weight (δ_{hw}) of two neighboring blocks is equal to 0 or 1. The correlation is 0 when the normalized Hamming weight is equal to 0.5.

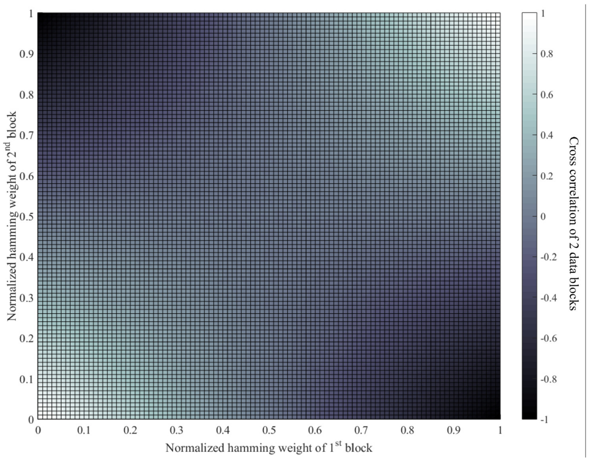

Fig.4: The correlation between data blocks in the cache line.

We also make the variation of normalized Hamming weight by utilizing the Gaussian distribution. Therefore, the normalized Hamming weight of each cache line is generated by

$$p(\delta_{hw}) = \frac{1}{\sigma_{hw}\sqrt{2\pi}}\exp\left(-\frac{(\delta_{hw}-\mu_{hw})^2}{2\sigma_{hw}^2}\right), \qquad (10)$$

where μ_{hw} and σ_{hw} are mean and standard deviation of normalized Hamming weight, respectively. If the mean is large and the standard deviation is small, it means that most stored data blocks have a large Hamming weight but the deviation of Hamming weights is small. If the mean and the standard deviation are large, it means that most stored data blocks have a large Hamming weight and the deviation of Hamming weights is large. If the mean is small and the standard deviation is large, it means that most stored data blocks have a small Hamming weight but the deviation of Hamming weights is large. If the mean and the standard deviation are small, it means that most stored data blocks have a small Hamming weight but the deviation of Hamming weights is small.

3. TWO-DIMENSIONAL ERROR CORRECTION CODE FOR STT-MRAM CACHE

Fig. 5 shows the two-dimensional error correction code for the STT-MRAM cache. The proposed code consists of the product code and weight reduction code. The product code protects the data bits and

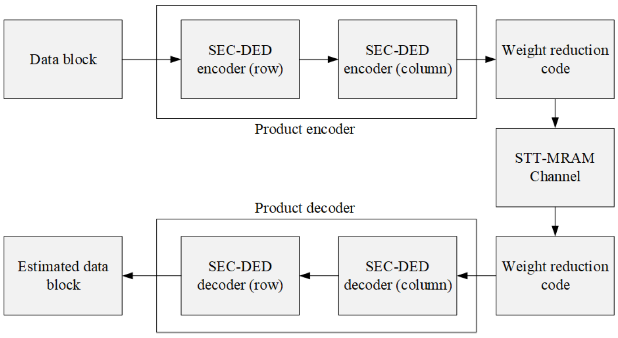

Fig.5: *The STT-MRAM cache scheme.*

the weight reduction code reduces the error probability of writing ‘1’ bits. The details of the two-dimensional error correction code are given next.

3.1 Product Code

Product coding is a method of combining short-length component codes to create a long length code with better ECC capabilities. It offers excellent performance from cross parity check [27] compared to ordinary long length codes, and minimal circuitry overhead because the component codewords have poor error correction capabilities.

In [28], a two-dimensional error code of BCH and RS codes with SEC-DED code was proposed. Although the error performance was improved, the complexity of decoders was high. [29] proposed the HARQ to increase the reliability. However, the latency was increased. In this work, we propose the SEC-DED code with single error correction capability along rows and columns to achieve high error correction capability while maintaining lower decoding complexity.

Let C_1 and C_1 be a (n_1, k_1) and (n_2, k_2) linear code. Then a $(n_1 n_2, k_1 k_2)$ linear code may be created in which each codeword is placed in a rectangular array of n_1 columns and n_2 rows as shown in Fig. 6, with each row being a codeword in C_1 and each column being a codeword in C_2. On a data array of size $k_1 \times k_2$, this code array may be created by doing row (column) encoding first, then column (row) encoding. The bottom-right cross parity block is $(n_1 - k_1) \times (n_2 - k_2)$, and it is made by encoding row (column) parity along the column parity (row). The minimum weight of the product code is equal $d_1 d_2$ [27] if code C_1 has Hamming distance d1 and code C_2 has Hamming distance d_2. Increasing the minimum weight of each code increases the number of erroneous patterns in the code array that can be corrected. [28] and [29] both utilized product codes that used single-error-correction codes in each dimension. For SRAM caches of size 256×256 bits, [28] utilized an 8-bit even-parity code in both dimensions with bit interleaving. 8-bit even-parity code was utilized in interconnection networks in [29]. In both scenarios, product codes were used to improve error correction performance.

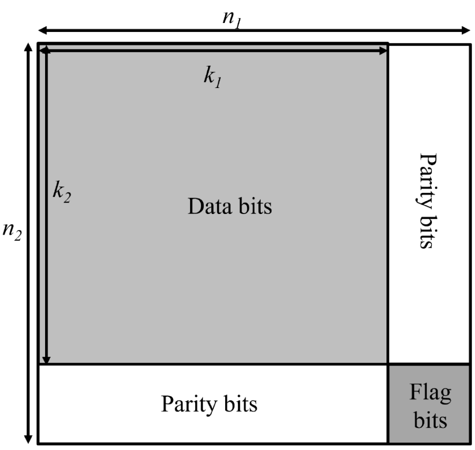

Fig.6: *Two-dimensional codeword structure.*

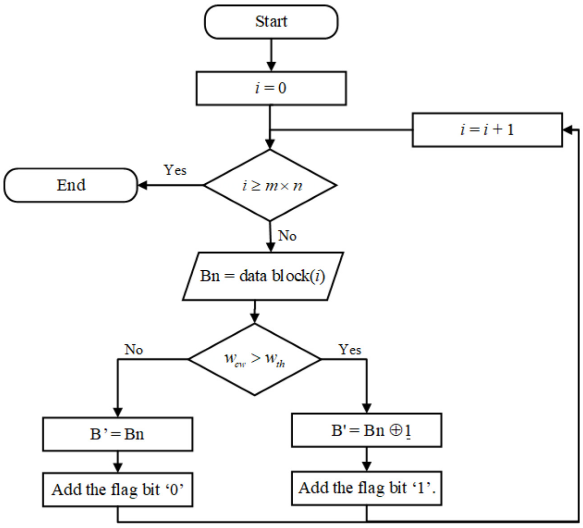

Fig.7: *The algorithm of weight reduction code with block-centric approach.*

3.2 Weight Reduction Code

The process variation and thermal fluctuation cause 0 → 1 write error. To address this issue, the number of ‘1’ bits in the data block must be reduced before writing in the STT-MRAM cache. In [24], the dynamic differential code was proposed to reduce the Hamming weight of data blocks. The data blocks with large Hamming weight in the same cache line are successively replaced with modulo-2 of their value and a flag bit ‘1’ is added to the code block. However, if the flag bit of the first data block with high Hamming weight is flipped due to a write error, the error will be propagated to the next code block. We also observe that the modulo-2 of 2 high Hamming weight code blocks provides the high Hamming weight code block output. Therefore, we propose a weight reduction code to reduce the Hamming weight without error propagation. Moreover, the proposed weight reduction code guarantees that the output code block provides the normalized Hamming weight which is less or equal to 0.5. The proposed weight reduction code is separated into two types: block-centric and page-centric approaches.

Fig. 7 depicts weight reduction codes using a block-centric approach. We will suppose that each page has $m \times n$ blocks. Before the SEC-DED encoder starts, the Hamming weight of each data block will be verified. All bits in the data block are flipped and the flag bit is set to '1' if the Hamming weight of the data block (w_{cw}) is larger than the Hamming weight threshold (w_{th}). The data block is not altered otherwise, and the flag bit is set to '0.' The procedure is repeated until all data blocks have been processed. After that, flag bits are added to the code blocks as shown in Fig. 6.

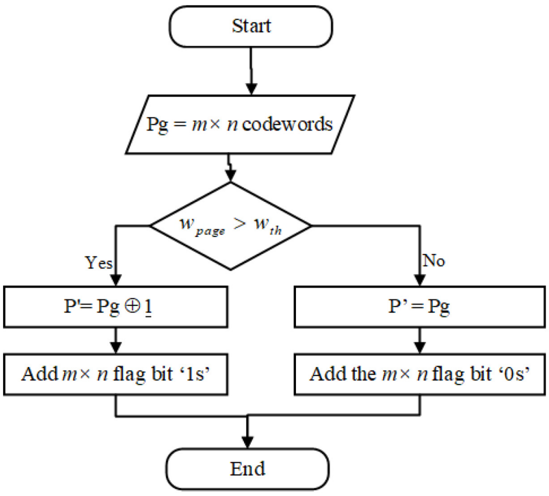

Fig.8: *The algorithm of weight reduction code with page-centric approach.*

Fig. 8 illustrates the weight reduction codes using a page-centric approach. We will suppose that each page has $m \times n$ blocks. If the total Hamming weight of each page (w_{page}) exceeds the Hamming weight threshold (w_{th}), all bits in the page are flipped, and $m \times n$ flag bits are set to '1.' Otherwise, the user data bits remain unchanged, and the flag bits for $m \times n$ are set to '0.' After that, the flag bits are added to the code blocks as shown in Fig. 6.

4. SIMULATIONS AND RESULTS

In our simulation, we suppose that each page includes 8 cache lines, each of which is made up of 8 blocks, each of which has 64 bits. The ratio of '1' to '0' bits is $R_p = 4 \times 10^{-3}$ [24]. To secure the data in each block, we employ (72, 64, 1) SEC-DED code.

The bit error performance of the STT-MRAM cache is shown in Fig. 9. The user data is generated with the normalized Hamming weight δ_{hw}. As a result of the error propagation of codewords, the SEC-DED code with the dynamic differential code [24] provides the greatest bit error rate for all $p_{ER,0\to1}$. If an error occurs in one or more codewords, it will be spread to the subsequent codewords. As a result, when compared to SEC-DED code without dynamic differential code, the dynamic differential code provides inferior error performance. When compared to SEC-DED code without weight reduction code, the proposed page-centric approach has a slightly lower bit-error rate. The proposed block-centric approach is capable of producing the best error results.

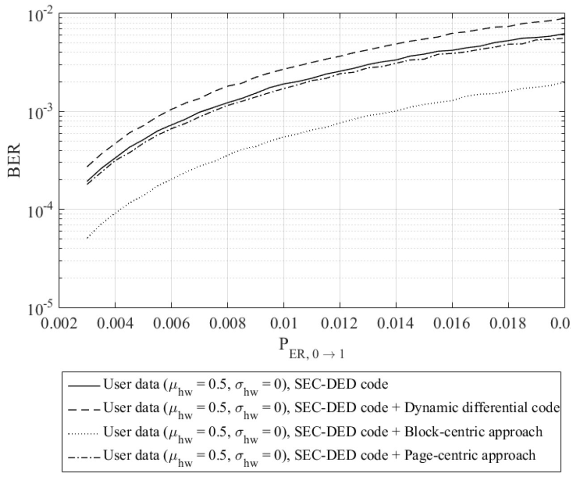

Fig.9: *The error performance comparison of weight reduction codes. The normalized Hamming weight of each cache line is distributed according to $\mu_{hw} = 0.5$ and $\sigma_{hw} = 0$.*

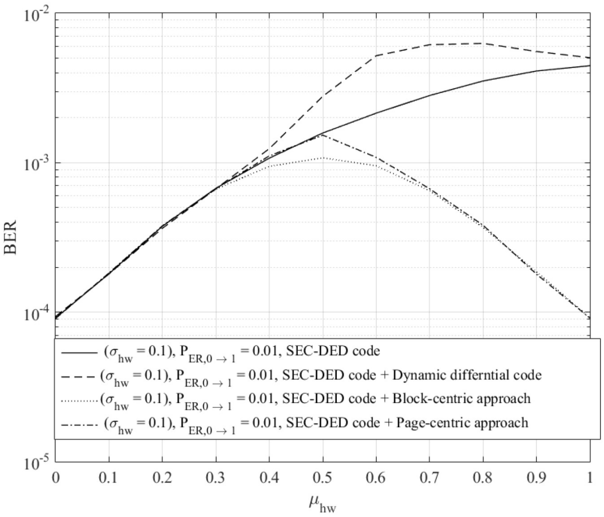

Fig.10: *The error performance comparison of weight reduction codes. The normalized Hamming weight of each cache line is distributed according to $0 < \mu_{hw} < 1$ and σ_{hw} = 0.1. The $P_{ER,0\to1}$ is 0.01*

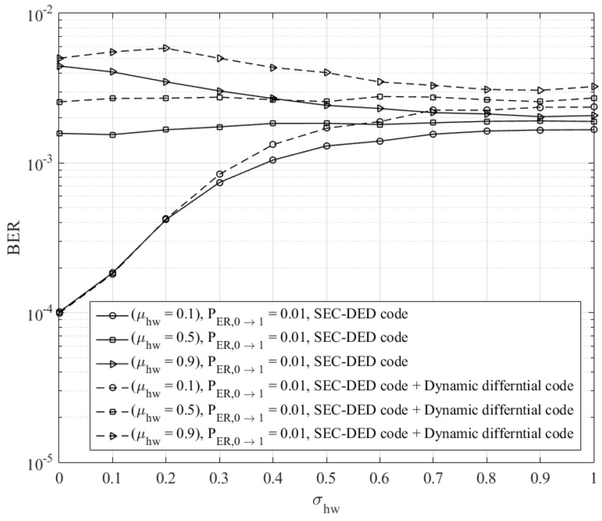

Fig.11: *The error performance comparison of dynamic differential code. The normalized Hamming weight of each cache line is distributed according to* $0 < \sigma_{hw} < 1$ *and* $\mu_{hw} = 0.1, 0.5$ *and 0.9.*

The error rate $p_{ER,0\to1}$ is set as 0.01 in the following simulations. In Fig. 10, the standard deviation of normalized Hamming weight is 0.1. The mean of normalized Hamming weight varied between 0 and 1. SEC-DED code with dynamic differential code has the greatest bit error rate (BER) for high Hamming weight codewords. When μ_{hw} is raised, the BER of the SEC-DED code with dynamic differential code is reduced. The BER curve of the SEC-DED code with proposed weight reduction codes, on the other hand, is reduced until $\mu_{hw} = 0.5$. This is due to the fact that a codeword with high Hamming weight is reduced and does not propagate errors. At μ_{hw} of about 0.5, the error performance of the block-centric approach exceeds the page-centric approach.

In Fig. 11, we set $\mu_{hw} = 0.1$, 0.5, and 0.9. The standard deviation normalized Hamming weight varied from 0 to 1. When the σ_{hw} is raised and μ_{hw} equals 0.1, the BER of $\mu_{hw} = 0.9$ is reduced. When σ_{hw} is raised for a high normalized Hamming weight ($\mu_{hw} = 0.9$), the BER improves. These three curves all meet at the same point since the normalized Hamming weight is distributed as a uniform distribution. The BER performance curves for SEC-DED code with dynamic differential code show comparable patterns for low, intermediate, and high Hamming weights. At $\mu_{hw} = 0.t$ and $0 < \sigma_{hw} < 0.4$, the SEC-DED with dynamic differential code beats the SEC-DED without dynamic differential code.

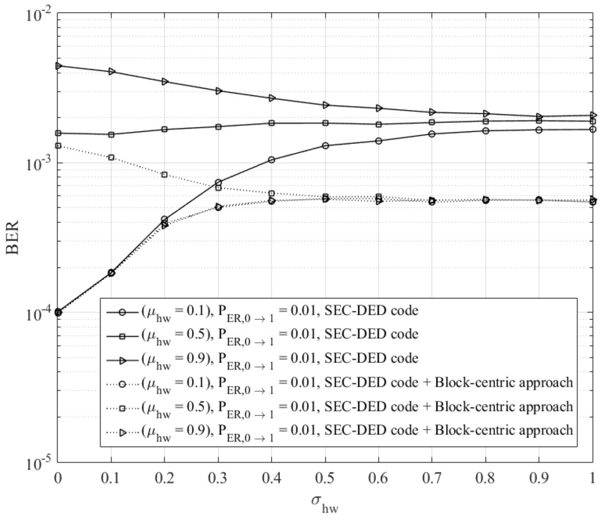

Fig.12: *The error performance comparison of block-centric approach. The normalized Hamming weight of each cache line is distributed according to* $0 < \sigma_{hw} < 1$ *and* $\mu = 0.1, 0.5$ *and 0.9.*

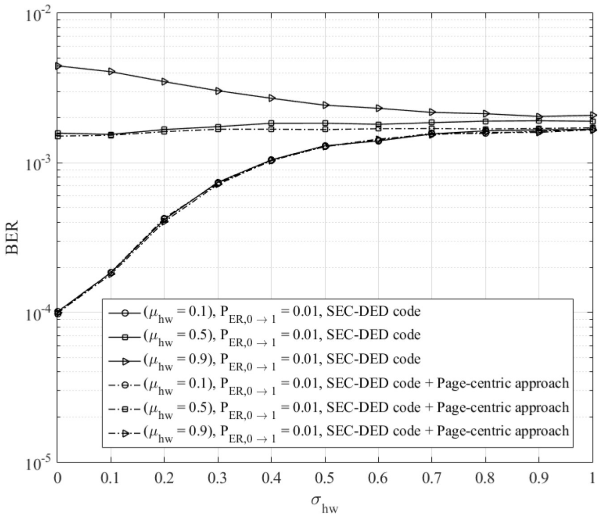

Fig.13: *The error performance comparison of page-centric approach. The normalized Hamming weight of each cache line is distributed according to* $0 < \sigma_{hw} < 1$ *and* $\mu = 0.1, 0.5$ *and 0.9.*

When the user data has $\mu_{hw} = 0.1$, 0.5, and 0.9, and the BER is considered at $p_{ER,0\to1} = 0.01$, Fig. 12 illustrates the error performance curves of weight reduction code using a block-centric approach. The BER curve for low and high Hamming weights converges around $\sigma_{hw} = 0.4$ as a consequence. The BER converges at $\sigma_{hw} = 0.4$ for the intermediate Hamming weight.

The BER curves of a weight reduction code using a page-centric approach are shown in Fig. 13. When σ_{hw} is raised at $\mu_{hw} = 0.1$ and 0.9, the BER is reduced. The BER is constant for $\mu_{hw} = 0.5$ at any σ_{hw}.

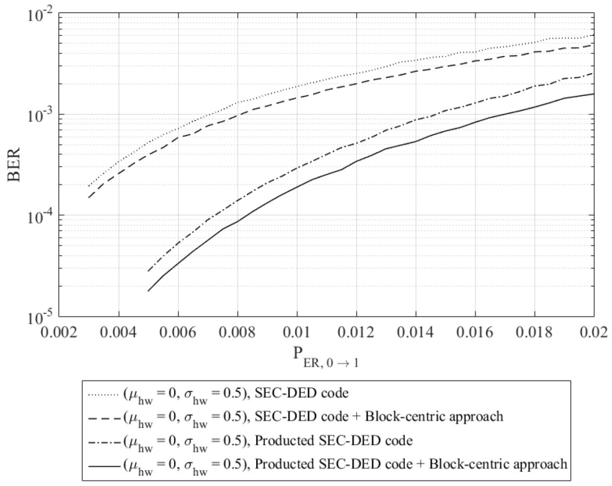

Fig.14: *The error performance comparison of SEC-DED code and product SEC-DED code with and without block-centric approach. The normalized Hamming weight of each cache line is distributed according to* $\mu_{hw} = 0.5$ *and* $\sigma_{hw} = 0$.

A comparison of the cache memory's BER is shown in Fig. 14. When comparing the performance of SEC-DED code with and without our proposed weight reduction code, it was observed that SEC-DED code with weight reduction code produced about 0.001 better performance at BER $= 10^{-3}$. At BER $= 10^{-3}$, (72, 64) SEC-DED Product code yields a better BER than standard (72, 64) SEC-DED code by around 1×10^{-2}. In comparison to the SED-DED product code (72, 64) without the weight reduction code and assuming BER $= 10^{-3}$, the (72, 64) SEC-DED product code together with the weight reduction code can enhance efficiency by around 2×10^{-3}.

5. CONCLUSIONS

In this article, the SEC-DEC code scheme is modified using a product code to enhance bit error correction performance. In addition, a weight reduction code is proposed to decrease the errors caused by writing '1' bits. When the hamming weight of user data is raised, the simulation results demonstrated that the proposed weight reduction code is superior to the dynamic differential code. Compared to the SEC-DED code without weight reduction code, the SEC-DED code with weight reduction code can minimize the errors of writing '1' bits around 0.001 at a bit error rate of 10^{-3}. When compared to the SEC-DED product code without weight reduction code, the SEC-DED product code with weight reduction code can reduce the error of writing '1' bits to about 0.002 with a 10^{-3} bit error rate. At BER $= 10^{-3}$, (72, 64) SEC-DED Product code yields a better bit error rate than standard (72, 64) SEC-DED code by around 1×10^{-2}.

ACKNOWLEDGMENT

This work was supported in part by the research and researchers for industries (RRI) under grant no. PHD61I0041.

References

[1] J. Hu, C. J. Xue, Q. Zhuge, W. Tseng and E. H. -. Sha, "Towards energy efficient hybrid on-chip scratch pad memory with non-volatile memory," *2011 Design, Automation & Test in Europe*, pp. 1–6, 2011.

[2] H. Noguchi et al., "7.2 4mb stt-mram-based cache with memory-accessaware power optimization and write-verify-write / read-modify-write scheme," *2016 IEEE International Solid-State Circuits Conference (ISSCC)*, pp. 132–133, 2016.

[3] W. Xu, H. Sun, Y. Chen, and T. Zhang, "Design of Last-Level On-Chip Cache Using Spin-Torque Transfer RAM (STT-RAM)," in*IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 19, no. 3, pp. 483–493, 2011.

[4] P. Zhou, B. Zhao, J. Yang, and Y. Zhang, "Energy Reduction for STTRAM Using Early Write Termination," *2009 IEEE/ACM International Conference on Computer-Aided Design - Digest of Technical Papers*, pp. 264–268, 2009.

[5] M. Hsiao. "A Class of Optimal Minimum Odd-weight-column SEC-DED Codes," *IBM Journal of Research and Development*, vol. 14, no. 4, pp. 95–401, 1970.

[6] J. Yang, B. Geller, M. Li, and T. Zhang, "An information theory perspective for the binary STT-MRAM cell operation channel," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 23, no. 4, pp. 979-991, Mar. 2016.

[7] C. W. Smullen, IV, A. Nigam, S. Gurumurthi, and M. R. Stan, "The stetsims stt-ram simulation and modeling system," *in ICCAD*, 2011.

[8] L. Zhang et al., "Channel Modeling and Reliability Enhancement Design Techniques for STT-MRAM," *2015 IEEE Computer Society Annual Symposium on VLSI*, pp. 461-466, 2015.

[9] B. Li, Y. Pei and W. Wen, "Efficient Low-Density Parity-Check (LDPC) Code Decoding for Combating Asymmetric Errors in STT-RAM," *2016 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, pp. 266-271, 2016.

[10] X. Zhong, K. Cai, Z. Mei and T. Q. S. Quek, "Deep Learning-Based Decoding of Linear Block Codes for Spin-Torque Transfer Magnetic Random Access Memory," in *IEEE Transactions on Magnetics*, vol. 57, no. 2, pp. 1-5, Feb. 2021.

[11] J. Yang, B. Geller, M. Li, and T. Zhang, "An information theory perspective for the binary STT-MRAM cell operation channel," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 23, no. 4, pp. 979 - 991, Mar. 2016.

[12] 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," in *IEEE Transactions on Computers*, vol. 65, no. 12, pp. 3661-3675, 1 Dec. 2016.

[13] Z. Azad, H. Farbeh and A. M. H. Monazzah, "ORIENT: Organized interleaved ECCs for new STT-MRAM caches," *2018 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 1187-1190, 2018.

[14] A. Ahari, M. Ebrahimi, F. Oboril and M. Tahoori, "Improving reliability, performance, and energy efficiency of STT-MRAM with dynamic write latency," *2015 33rd IEEE International Conference on Computer Design (ICCD)*, pp. 109-116, 2015.

[15] E. Aliagha, A. M. H. Monazzah and H. Farbeh, "REACT: Read/Write Error Rate Aware Coding Technique for Emerging STT-MRAM Caches," in *IEEE Transactions on Magnetics*, vol. 55, no. 5, pp. 1-8, May 2019.

[16] R. Bishnoi, M. Ebrahimi, F. Oboril and M. B. Tahoori, "Improving Write Performance for STT-MRAM," in *IEEE Transactions on Magnetics*, vol. 52, no. 8, pp. 1-11, Aug. 2016.

[17] S.-S. Pyo, C.-H. Lee, G.-H. Kim, K.-M. Choi, Y.-H. Jun, and B.-S. Kong, "45nm low-power embedded pseudo-SRAM with ECC-based auto-adjusted self-refresh scheme," *2009 IEEE International Symposium on Circuits and Systems*, pp. 2517-2520, 2009.

[18] P. Reviriego, A. Sanchez-Macian, and J. A. Maestro, "Low power embedded DRAM caches using BCH code partitioning," *2012 IEEE 18th International On-Line Testing Symposium (IOLTS)*, pp. 79-83, 2012.

[19] Y. Li, H. Zhong, R. Danilak, and E. T. Cohen, (Google Patents, 2014).

[20] B. Del Bel, J. Kim, C. H. Kim, and S. S. Sapatnekar, "Improving STT-MRAM density through multibit error correction," *2014 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 1-6, 2014

[21] X. Guo, M. N. Bojnordi, Q. Guo, and E. Ipek, "Sanitizer: Mitigating the impact of expensive ECC checks on STT-MRAM based main memories," in*IEEE Transactions on Computers* vol. 67, no. 6, pp. 847-860, 2018.

[22] C. Yang, Y. Emre, Y. Cao, and C. Chakrabarti, "Improving reliability of non-volatile memory technologies through circuit level techniques and error control coding," EURASIP J. Adv. Signal Process. 2012, 211.

[23] X. Zhong, K. Cai, P. Chen and Z. Mei, "Rate-Compatible Protograph LDPC Codes for Spin-Torque Transfer Magnetic Random Access Memory (STT-MRAM)," *2018 Asia-Pacific Magnetic Recording Conference (APMRC)*, pp. 1-2, 2018.

[24] W. Wen, M. Mao, X. Zhu, S. H. Kang, D. Wang, and Y. Chen, "CD-ECC: Content-dependent error correction codes for combating asymmetric nonvolatile memory operation errors," in*Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD)*, pp. 1–8, Nov. 2013.

[25] E. Cheshmikhani, H. Farbeh and H. Asadi, "Enhancing Reliability of STT-MRAM Caches by Eliminating Read Disturbance Accumulation," *in 2019 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, Mar. 2019.

[26] C. L. Chen and M. Y. Hsiao, "Error-Correcting Codes for Semiconductor Memory Applications: A State-of-the-Art Review," in *IBM Journal of Research and Development*, vol. 28, no. 2, pp. 124-134, March 1984, doi: 10.1147/rd.282.0124.

[27] S. Lin and D. J. Costello, *Error Control Coding*, 2nd ed. Upper Saddle River, NJ: Pearson Education.

[28] J. Kim et al., "Multi-bit error tolerant caches using two-dimensional error coding," in *Proc. 40th IEEE/ACM Int. Symp. Microarch.*, 2008, pp. 197–209.

[29] B. Fu and P. Ampadu, "Burst error detection hybrid ARQ with crosstalk-delay reduction for reliable on-chip interconnects," *2009 24th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems*, pp. 440–448, 2009.

Chatuporn Duangthong received the B.S. degree in Telecommunication Engineering from the King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand, in 2015 and the M.S. degree in Telecommunication Engineering from the King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand, in 2017. His research interests include channel coding, storage technology and signal processing.

Pornchai Supnithi received the B.S. degree from the University of Rochester, Rochester, NY, USA, in 1995, the M.S. degree from the University of Southern California, Los Angeles, CA, USA, in 1997 and the Ph.D. degree in Electrical Engineering from the Georgia Institute of Technology, Atlanta, GA, USA, in 2002. In 2015, he became a full professor in the Telecommunication Engineering Department, Faculty of Engineering, King Mongkut's Institute of Technology Ladkrabang. He has published over 50 journal articles, 100 conference papers and 3 book chapters. His research interests are in Telecommunications, Ionospheric and GNSS, Data storage and Engineering Education. His laboratory maintains numerous observation stations in Thailand (with ionosonde, beacon receivers, magnetometer and GNSS receivers) as well as Thai GNSS and Space Weather Information Website (http://iono-gnss.kmitl.ac.th).

Prof. Dr. Pornchai Supnithi was a mid-career Thailand Research Fund scholar from 2013-2015. He previously served as the second Vice President of ECTI Association. Currently, he serves as a subcommittee member in the National Research Council of Thailand (NRCT), International Reference Ionosphere (IRI) Committee (under URSI, COSPAR) and the National GNSS Infrastructure subcommittee, Ministry of Science and Technology, Thailand.

Watid Phakphisut received the B.Eng. degree in Telecommunication Engineering, the M.Eng. degree in Data Storage Technology, and the D.Eng. degree in Electrical Engineering from the King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand, in 2009, 2011, and 2015, respectively. His research interests include the channel coding, and artificial intelligence for communication and storage systems.