ENGINEERING IN ADVANCED RESEARCHSCIENCE AND TECHNOLOGY

ISSN 2352-8648
Vol.04, Issue.02
November-2021
Pages: -366-379

A HIGH SPEED AREA EFFICIENT FAULT TOLERANT METHOD IN SECURE PCM MEMORY

[1]PALLEPOGU UJWALA BHAVITHA, [2]B. SAMBA SIVA RAO

[1]M. Tech Student, Dept. of ECE, NOVA COLLEGE OF ENGINEERING AND TECHNOLOGY, IBRAHIMPATNAM, A.P

[2] Associate Professor, Dept. of ECE, NOVA COLLEGE OF ENGINEERING AND TECHNOLOGY, IBRAHIMPATNAM, A.P

ABSTRACT: In this paper, we will propose a method for tolerating the stuck-at faults caused by an endurance issue in secure-resistive main memories. In the proposed method, by employing the random characteristics of the encrypted data encoded by the Modified Advanced Encryption Standard (MAES) as well as a rotational shift operation, a large number of memory locations with stuck-at faults could be employed for correctly storing the data. The proposed RandShift technique, which is simple and energy efficient due to its one-time computation of the MAES encryption for each data written to the PCM main memory. Due to the simple hardware implementation of the proposed method, its energy consumption is considerably smaller than that of other recently proposed methods. The fault coverage of the proposed method is similar to that of the existing method. In this article, we present a simple method to tolerate the stuck-at faults in the PCM main memory when encrypted data are stored on it. The synthesis and simulation are verified by using Xilinx ISE 14.7 version tool.

Keywords: PCM Memory, MAES, Randshift technique, Checker, Row verifier.

INTRODUCTION: Computational processing has increased in cloud servers, requiring larger core counts and higher memory densities. The number of processor cores doubles every two years, while the DRAM DIMM capacities double every three years [1]. This causes a large gap between the core count and the memory density. On the other hand, traditional DRAM chips consume more than 40% of power of the servers [2]. Also, DRAM scaling to reach a higher memory density has some challenges such as high leakage current, reduced memory cell reliability, and more complex fabrication processes [3]. Emerging memory technologies, which are categorized into volatile (DRAM-based) and nonvolatile

(resistive-based (have been introduced to solve scaling and power consumption problems [4]. Some of the volatile DRAM-based memories include reduced latency and tiered latency DRAM (RL-DRAM and TL-DRAM) and low powerDDR DRAM (e.g., LPDDR3, LPDDR4) architecture. While they have lower latency and power consumption, they suffer from higher costs of the fabrication process [5]. Nonvolatile memories, such as spin-transfer torque RAM (STT-RAM), phase-change memory (PCM), resistive RAM (ReRAM), and 3DXpoint, while enjoying from high scalability and low leakage power suffer from high latency, high dynamic power, and low endurance. While the endurance of PCM is not high, owing to its better scalability characteristics and lower power consumption property, it has a higher chance of becoming the next generation of main memories in computing systems. In this article, we focus on increasing the endurance of the PCM. It is important that any solution for increasing the reliability of the main memory be power and area efficient.

Phase Change Memory (PCM): Phase-change memory (also known as PCM, PCME, PRAM, PCRAM, OUM (ovonic unified memory) and C-RAM or CRAM (chalcogenide RAM) is a type of non-volatile random- access memory. PRAMs exploit the unique behavior of chalcogenide glass. In the older generation of PCM, heat produced by the passage of an electric current through a heating element generally made of titanium nitride was used to either quickly heat or quench the Glass, making it amorphous, or to hold it in its crystallization temperature range for some time, thereby switching it to a crystalline state. PCM also has the ability to achieve a number of distinct intermediary states, thereby having the ability to hold multiple bits in a single cell, but the difficulties in programming cells in this way has prevented these capabilities from being implemented in other technologies (most notably flash memory) with the same capability. Phase Change Memory (PCM) is one of the emerging memory technologies that have received a lot of attention from both the academic and industrial communities in recent years. Currently, Samsung Electronics and Micron Technology have made commercial PCM chips available in the market. Architectural efforts show great potential to use PCM as a DRAM alternative for future memory systems because PCM is non-volatile memory and the latency is close to DRAM. Compared to Flash Memory, PCM is byte-addressable with a much longer lifetime and much faster read and write cycles. It is also not necessary to explicitly erase before writing PCM cells. PCMs store “0” and “1” values by changing the state of the chalcogenide materials [4]. Chalcogenide is heated to a high temperature (over 600 degree centigrade), which changes to the liquid phase. Once cooled, it is frozen to an amorphous glass-like state with high electrical resistance. To achieve a low resistance

state, the chalcogenide should be transformed into its crystallization state, which is achieved by heating to a temperature above its crystallization point. Frequent heating and cooling processes of a cell material lead to the creation of a hard fault, thereby decreasing the lifetime of the cell. Hard faults show themselves as stuck-at faults (stuck-at Logical "1" or "0") [6]. Soft errors are another type of fault, which is more common in DRAM-based memory. They are generated by striking alpha particles and are considered to be transient faults. Modified Advanced Encryption Standard (MAES): The Modified Advanced Encryption Standard (MAES) [13], a symmetric key block which is published by the National Institute of Standards and Technology (NIST) in December 2001. It is a non-Feistel block cipher that encrypts and decrypts a fixed data block of 128-bits. There are three different key lengths. The encryption/decryption consists of 10 rounds of processing for 128-bit keys, 12 rounds for 192-bit keys, and 14 rounds for 256-bit keys. MAES performs several rounds where each round is made of several stages. A data block is transformed from one stage to another. Before and after each stage, the data block is referred to as a state. Each round, except the last, performs four transformations which are invertible. The last round implements the rest three transformations except the Mix Columns stage. Below figure shows the AES cipher structure.

Substitute Bytes: The first transformation, Sub Bytes, is used at the encryption site. It is a non-linear byte substitution that operates independently on each byte of the state using a substitution table (S-Box). All the 16 bytes of the state are substituted by the corresponding values which are found from the lookup table. In decryption, InvSubBytes is used. Bytes of a state are substituted from InvSubBytes table.

Shift Rows: In the encryption, the state bytes are shifted left in each row. It is called Shift Rows operation. The number of the shifts depends on the row number (0, 1, 2 or 3) of the state matrix. Row 0 bytes are not shifted and row 1, 2, 3 are shifted to 1, 2, 3 bytes left accordingly.

Mix Column: The Mix Columns transformation operates at the column level. It transforms each column of the state to a new column. The transformation is actually the matrix multiplication of a state column by a constant square matrix. All the arithmetic operations are conducted in the Galois Field (Finite Field). The bytes are treated as polynomials rather than numbers.

Add Round Key: AddRoundKey precedes one column at a time. It is similar to Mix Columns in this respect. AddRoundKey adds a round keyword to each column matrix. Matrix addition operation is performed in the AddRoundKey stage. In encryption, Sub Bytes, Shift Rows, Mix

Columns, and AddRoundKey are performed in all rounds except the last round.Mix Columns transformation operation is not performed in the last round of encryption. The decryption process essentially follows the same structure as the encryption, in addition to the nine rounds of Inverse Shift Rows, Inverse Sub Bytes, Inverse AddRoundKey and Inverse Mix Columns Transformation. In the final round, Inverse Mix Columns is no longer performed.

EXISTING METHOD: To prevent the leakage of important data into cloud servers and personal computers, one may encrypt data stored in the memory cells. AES is the most popular algorithm for data encryption in unsecure memory. This algorithm is fast and powerful than the other symmetric encryption algorithms. The input blocks AES are of 128-, 192-, 256-bits, whose number of rounds are10, 12, and 14, respectively. Each round includes AddRoundKey, Substitute, ShiftRows, and MixColumn items, which all are high-energy consumption operations.

RANDSHIFT METHOD: As mentioned before, owing to the limited write endurance of the PCM, some of the cells are worn-out, permanently become stuck-at "1" or "0" value. The idea of fault coverage based on "ST-R" and "ST-W" is demonstrated by a memory word shown below figure, where 4-bit positions have stuck-at faults (i.e., the bit positions of 0, 4, 10, and 13). For example, in this memory word, storing "0xB3A8" leads to ST-W at the 4th- and 13th-bit positions. In this case, a 1-bit circular shift to the right causes the value to become "0x59D4" giving rise to ST-R at the 4th- and 13th-bits without inducing any other ST-W.

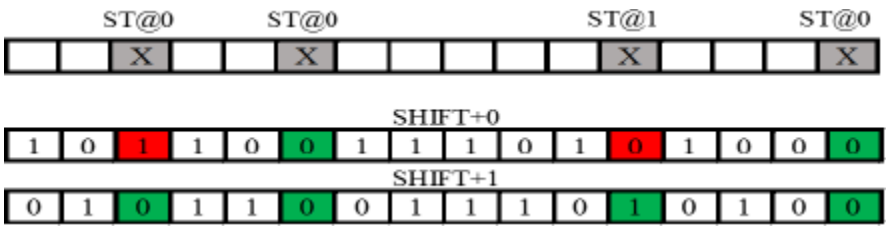

Fig..1 Representation of ST-W and ST-R ideas.

In our proposed method, the correlation of any two AES encrypted data is almost zero, and thus one may consider the output of the AES encryption as a random number. By using this random number we will encrypt and decrypt the data. The randomness feature of the output value in the AES may be employed as a solution to tolerate stuck-at faults in PCM cells. If the encrypted data fail to fully match with the stuck-at faults (the number of "ST-W" is zero), one may use the technique suggested in [12] to regenerate the encrypted data. As mentioned before, due to the

possibility of having to do a large number of regenerations, this technique is a power-hungry approach. Each mismatch between the writing data and the stuck-at fault values causes to add ten rounds of add-Round-Key, substitute, shift-Rows, and mix-Column in AES-128. Although data shifting approach, as well as the generation of new encrypted data may not completely keep the randomness of the data, it may increase the number of "ST-R" in the case when there are not many stuck-at fault bits in the memory block. Obviously, the former approach is considerably faster and more energy efficient. Since the Randshift approach is applied on the encrypted data whose bits enjoy a high degree of randomness, the disturbance in the randomness in the Randshift approach is similar to that in the regeneration approach.

Barrel Shifter:

A barrel shifter is a digital circuit that can shift a data word by a specified number of bits. The shifting of days may be circular left shift or circular right shift. In this proposed method we implemented circular right shift. One way to implement it is as a sequence of multiplexers where the output of one multiplexer is connected to the input of the next multiplexer in a way that depends on the shift distance or we can assign loop statements if the block size is more.

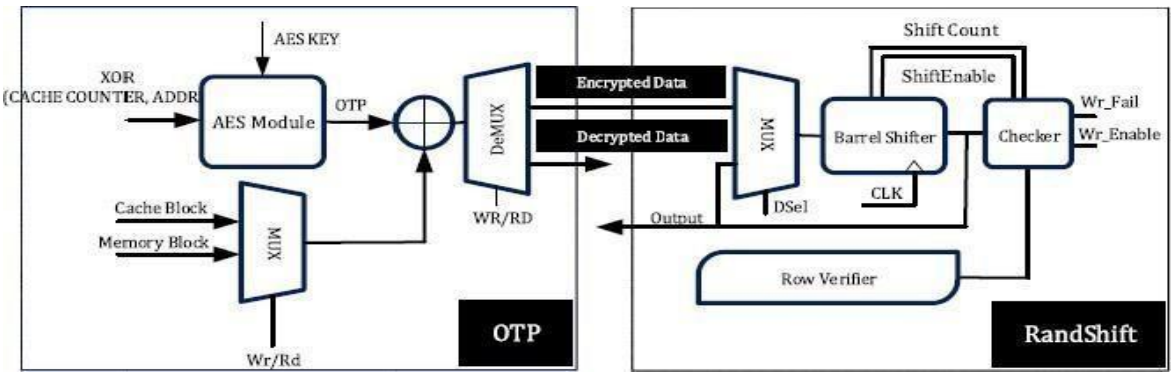

Fig.2 Full architecture of Randshift.

AES: It is a non-Feistel block cipher that encrypts and decrypts a fixed data block of 128-bits. There are three different key lengths. The encryption/decryption consists of 10 rounds of processing for 128-bit keys, 12 rounds for 192-bit keys, and 14 rounds for 256-bit keys. In the proposed method they have performed for 128 bit length input data and the key of same length of 128 bit data. Each time a new key is used to encrypt the data.

AES performs several rounds where each round is made of several stages. A data block is

transformed from one stage to another. Before and after each stage, the data block is referred to as a state. Each round, except the last, performs four transformations which are invertible. The last round implements the rest three transformations except the MixColumns stage. Below figure shows the AES cipher structure.

PROPOSED METHOD

In the proposed method Modified Advanced Encryption Standard encryption was proposed. MAES is the most popular algorithm compared to AES for data encryption in unsecure memory. This algorithm is fast and powerful than the other symmetric encryption algorithms. MAES input keys are of 128-, 192-, 256-bits, and the number of rounds are10, 12, and 14, respectively. Each round contains AddRoundKey, Substitute, ShiftRows, and MixColumn operations.

RANDSHIFT METHOD:

As mentioned before, owing to the limited write endurance of the PCM, some of the cells are worn-out, permanently become stuck-at "1" or "0" value. The idea of fault coverage based on "ST-R" and "ST-W" is demonstrated by a memory word shown below figure, where 4-bit positions have stuck-at faults (i.e., the bit positions of 0, 4, 10, and 13). For example, in this memory word, storing "0xB3A8" leads to ST-W at the 4th- and 13th-bit positions. In this case, a 1-bit circular shift to the right causes the value to become "0x59D4" giving rise to ST-R at the 4th- and 13th-bits without inducing any other ST-W.

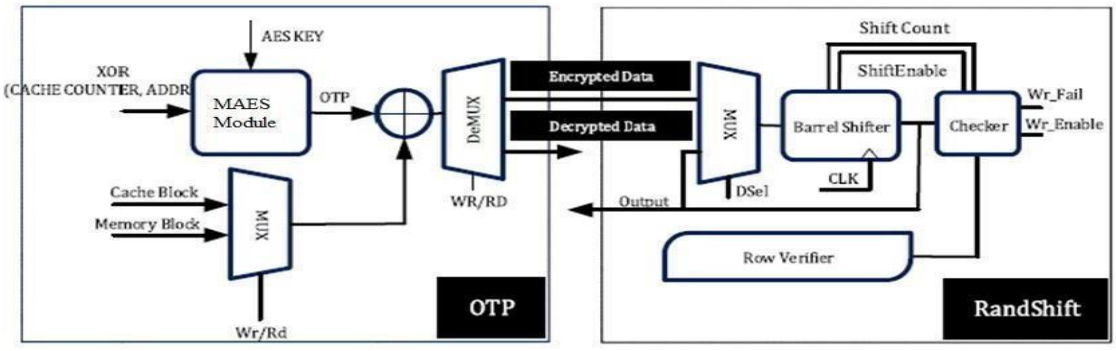

Fig3 Full Architecture of Proposed Randshift.

In our proposed method, the correlation of any two MAES encrypted data is almost zero, and thus one may consider the output of the MAES encryption as a random number. By using

This random number we will encrypt and decrypt the data. The randomness feature of the output value in the MAES may be employed as a solution to tolerate stuck-at faults in PCM cells. However, in the case of raw data (data that are not encrypted), due to the existence of spatial/temporal correlations among the data blocks, the use of manipulating methods. (e.g., circularshifting or inversing) may not be appropriate for overcoming the problem of the stuck-at faults. If the encrypted data fail to fully match with the stuck-at faults (the number of "ST-W" is zero), one may use the technique suggested in [12] to regenerate the encrypted data. As mentioned before, due to the possibility of having to do a large number of regenerations, this technique is a power-hungry approach. Each mismatch between the writing data and the stuck-at fault values causes to add ten rounds of add-Round-Key, substitute, shift-Rows, and mix-Column in MAES-128. Although data shifting approach, as well as the generation of new encrypted data may not completely keep the randomness of the data, it may increase the number of "ST-R" in the case when there are not many stuck-at fault bits in the memory block. Obviously, the former approach is considerably faster and more energy efficient. Since the Randshift approach is applied on the encrypted data whose bits enjoy a high degree of randomness, the disturbance in the randomness in the Randshift approach is similar to that in the regeneration approach. To evaluate the efficacy of the proposed Randshift method compared to the one proposed in [12] (Which is denoted ReGen in the rest of this article), probabilities of the stuck-at fault coverage's of the two methods should be compared. Because of the randomness property of MAES encryption, there is no dependency between the previous and the next data generations at each time in the ReGen method [12]. Because of the data dependency between the shifted bits in the Randshift method, extracting a close-form formula to calculate the fault coverage probability is not possible. For Randshift and ReGen methods for different iteration counts when the number of stuck-at fault in each row of PCM memory is from one to six and the size of each data block is 128 bits. As mentioned before, in each iteration of the Randshift method, a circular shift to right is performed, while in the ReGen method, an MAES encryption uses a new value for the counter to generate a different encrypted data. The coverage probabilities of Randshift and ReGen were obtained by applying one million

Encrypted data as stimuli. In this article, the number of faults in 128 bits of data does not exceed six Faults.

RESULTS

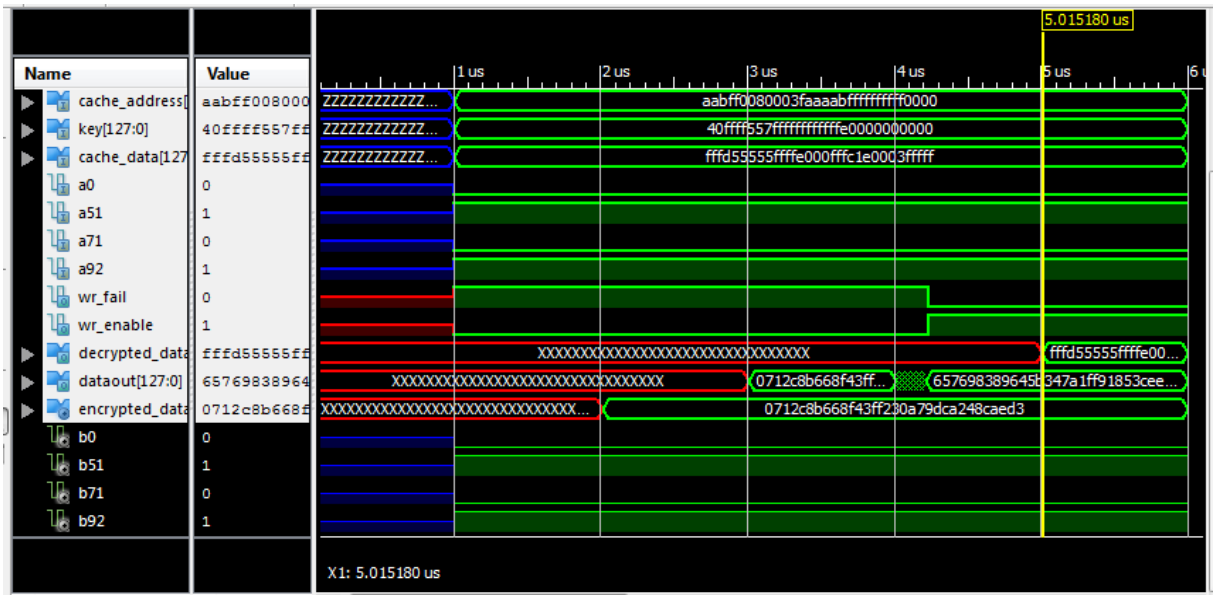

Fig Simulation for proposed Randshift method

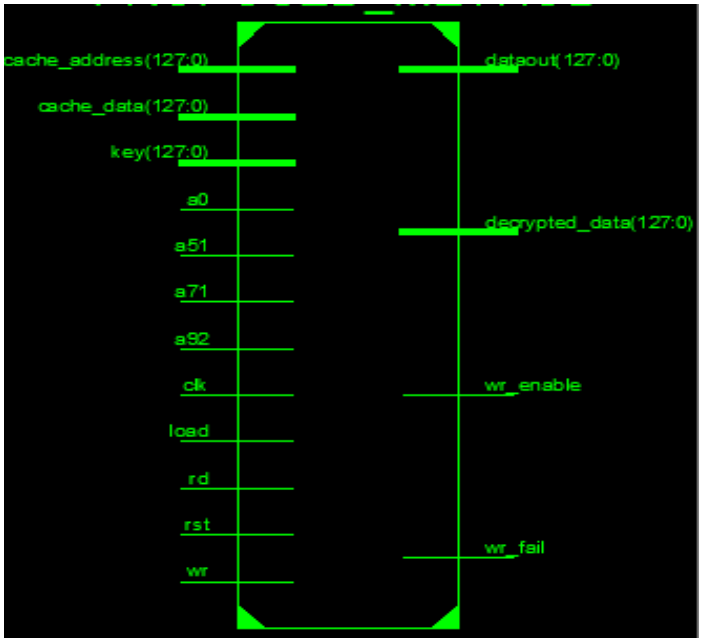

Fig. RTL Schematic of Proposed Randshift

CONCLUSION In this project, we proposed a method employing the MAES encryption as well as rotational shift operation to tolerate hard faults and randomqualityof nonvolatile memory cells. This method, which was called Randshift, which is implemented with simple hardware implementation. This method is to tolerate the stuck-at faults in the PCM main memory when encrypted data are stored on it. It limited the need for exploiting powerful error correction methods, such as Error correction method and Error correction pointer. The proposed method result shows that the data is securely encrypted and decrypted and stored in memory without any faults. The synthesis and simulation are verified by using Xilinx ISE 14.7 version tool.

FUTURE SCOPE In this paper we have used MAES module to encrypt and decrypt the data that we want to store in the memory. All the other modules will consume less area and less delay except the AES module. We will replace the MAES module with Pseudo Random Number Generator or Linear Feedback Shift Register to encrypt and decrypt the data. This way of implementation will improve the parameters like area and delay.

REFERENCES:

[1] K. Lim, J. Chang, T. Mudge, P. Ranganathan, S. K. Reinhardt, and. F. Wenisch, "Disaggregated memory for expansion and sharing in blade servers," ACM SIGARCH Comput. Archit. News, vol. 37, no. 3, pp. 267–278, 2009.

[2] M. Ware et al., "Architecting for power management: The IBM power7 approach," in Proc. High Perform. Comput. Archit. (HPCA), Jan. 2010, pp. 1–11.

[3] O. Mutlu, "The RowHammer problem and other issues we may face as memory becomes denser," in Proc. Conf. Design, Autom. Test Eur., 2017, pp. 1116–1121.

[4] A. Chen, "A review of emerging non-volatile memory (NVM) technologies and applications," Solid-State Electron., vol. 125, pp. 25–38, Nov. 2016.

[5] O. Mutlu, "Rethinking memory system design for data-intensive computing," in Proc. SAMOS, 2015, p. 1.

[6] N. H. Seong, D. H. Woo, V. Srinivasan, J. A. Rivers, and H.-H. S. Lee, "SAFER: Stuck- at-fault error recovery for memories," in Proc. 43rd Annu. IEEE/ACM Int. Symp. Microarchitecture, Dec. 2010, pp. 115–124.

[7] D. Strukov, "The area and latency tradeoffs of binary bit-parallel BCH decoders for prospective nanoelectronic memories," in Proc. Signals, Syst. Comput. (ACSSC), Oct.

2006, pp. 1183–1187.

[8] R. C. Bose and D. K. Ray-Chaudhuri, "On a class of error correcting binary group codes," Inf. Control, vol. 3, no. 1, pp. 68–79, Mar. 1960.

[9] S. Schechter, G. H. Loh, K. Strauss, and D. Burger, "Use ECP, not ECC, for hard failures in resistive memories," ACM SIGARCH Comput. Archit. News, vol. 38, no. 3, pp. 141–152, 2010.

[10] R. Maddah, R. Melhem, and S. Cho, "RDIS: Tolerating many stuckat faults in resistive memory," IEEE Trans. Comput., vol. 64, no. 3, pp. 847–861, Mar. 2015.

[11] R. Maddah, S. Cho, and R. Melhem, "Symbol shifting: Tolerating more faults in PCM Blocks," IEEE Trans. Comput., vol. 65, no. 7,pp. 2270–2283, Sep. 2016.

[12] D. Kline, Jr., R. G. Melhem, and A. K. Jones, "Counter advance for reliable encryption in phase change memory," IEEE Comput. Archit. Lett., vol. 17, no. 2, pp. 209–212, Jul. 2018.

[13] M. K. Qureshi, A. Seznec, L. A. Lastras, and M. M. Franceschini, "Practical and secure PCM systems by online detection of malicious write streams," in Proc. High Perform. Comput. Archit. (HPCA), Feb. 2011, pp. 478–489.

[14] S. Chhabra and Y. Solihin, "i-NVMM: A secure non-volatile main memory system with incremental encryption," in Proc. 38th Annu. Int. Symp. Comput. Archit. (ISCA), Jun. 2011, pp. 177–188.

[15] M. Jalili and H. Sarbazi-Azad, "Endurance-aware security enhancement in non-volatile memories using compression and selective encryption," IEEE Trans. Comput., vol. 66, no. 7, pp. 1132–1144, Dec. 2017.

[16] S. Cho and H. Lee, "Flip-N-Write: A simple deterministic technique to improve PRAM write performance, energy and endurance," in Proc. IEEE/ACM Int. Symp. Microarchitecture, Dec. 2009, pp. 347–357.

[17] S. Mathew et al., "53 Gbps native GF (24)2 composite-field AES-encrypt/decrypt accelerator for content-protection in 45 nm high-performance microprocessors," in Proc. VLSI Circuits (VLSIC), Jun. 2010, pp. 169–170.

[18] S. Haber and P. K. Manadhata, "Improved security for non volatile main memory," Tech. Discl. Commons, Feb. 2017. [Online].Available: https://www.tdcommons.org/dpubs_series/396

[19] Z. Zhang, W. Xiao, N. Park, and D. J. Lilja, "Memory module-level testing and error behaviors for phase change memory," in Proc. 30thInt. Conf. Comput. Design (ICCD), Dec. 2012, pp. 358–363.

[20] M. Soltani, M. Ebrahimi, and Z. Navabi, "Prolonging lifetime of nonvolatile last level caches with cluster mapping," in Proc. Int. Great Lakes Symp. VLSI, May 2016, pp. 329–334.