International Journal of Gender, Science and Technology
ISSN: 2040-0748
UGC Care Group I Journal.
Vol-13 Issue-02 Dec 2024
https://ijgst.com.2024.v13.i2.pp1042-1050

R and Shift: A Fault-Tolerant and Energy-Efficient Approach in Secure Nonvolatile Main Memory on FPGA

Mr.B.Ajantha Reddy [(1)], Mr.M.Ramana Reddy [(2)], Syed Inthiyaz [(3),] Yeruva Bhaskar Reddy [(4),] Araveeti Dinesh [(5),] Ravuri Kasi Ranga Sai Lokesh [(6)]

[1,2] Krishna Chaithanya Institute Of Technology & Sciences, Ece Department, Markapur, Andhra Pradesh.

[3,4,5,6,] Krishna Chaithanya Institute Of Technology & Sciences, UG Student-ECE, Markapur, Andhra Pradesh.

Abstract In this paper, we propose a straightforward, yet energy- and space-efficient technique to tolerate stuck-at errors resulting from an endurance problem in secure-resistive main memory. Many memory locations with stuck-at faults could be used in the suggested method to appropriately store the data by utilizing the rotational shift operation and the random properties of the encrypted data encoded by the Advanced Encryption Standard (AES). The suggested method's energy consumption is significantly lower than that of other recently proposed approaches because of its straightforward hardware implementation. The error correction code (ECC) and error correction pointer (ECP) are two more error correction techniques that can be used in conjunction with this one. The suggested approach is put into practice in a main memory system based on phase-change memory (PCM) and evaluated against three error-tolerating techniques to determine its effectiveness. The findings show that the suggested approach provides 82% energy reduction over the state-of-the-art method for a stuck-at fault incidence rate of 10−2 and an uncorrected bit error rate of 2 × 10−3. More broadly, we demonstrate that the fault coverage of the suggested method is comparable to the state-of-the-art method using a simulation analysis technique.

Keywords: ECC and ECE,5G Communications, FPGA, Verilog HDL, Fault Tolerant Coders and decoders.

I.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 non-volatile (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 power DDR DRAM (e.g., LPDDR3, LPDDR4) architecture. While they have lower latency and power consumption, they suffer from higher costs of the fabrication process [5]. Non-volatile 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 [4]. The cell endurances of PCM, STT-RAM, and ReRAM are on the order of 108, 4×1012, and 1011 write operation, respectively. 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.

PCMs store "0" and "1" values by changing the state of the chalcogenide materials [4]. Chalcogenide is heated to a high temperature (over 600 ◦C), 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 transient faults.

Methods for mitigating hard and soft faults in memory are divided into three general categories. In the first category, the error correction codes (ECCs) like single error correction and double error detection (SEC-DED) or the Hamming code-based BCH code are employed [7]. The required computations for encoding/decoding of the data in ECC-based algorithms increase by enlarging the number of errors imposing considerable overheads on the system [8]. In the second category, the error correction pointer (ECP) technique [9] is exploited to point the error location among the memory cells and utilizes redundant cells to replace faulty cells. This approach, which is suitable in the case of hard errors, has a high area overhead for storing the pointers and also is applicable for joining with other fault-tolerant methods with low overhead. To decrease the ECP area overhead, methods like SAFER and RDIS [6], [10] make use of clustering and re-grouping to tolerate the hard faults. In the third category, data manipulation methods (e.g., inverting data), utilizing the "0" and "1" positions for tolerating the hard faults, are employed. To explain these methods, note that the endurance problem induced by stuck-at faults may be classified into two types of errors: STuck at Right "ST-R" and STuck at Wrong "ST-W." "ST-R" ("ST-W") means the data to be written to the memory cell are equal (unequal) to the fault value. Prior work (see [11]–[13]) suggested methods to increase the "ST-R" in memory blocks by manipulating the data. A recently proposed method [12] generated new encrypted data for masking hard fault errors in the secure memory by re-encrypting the input data. Symbol-shifting [11] is another method which inverts the stored values in the case of SLC PCM memories for tolerating the ST-W hard faults.

As mentioned above, PCM is considered as a promising alternative to DRAM. On the other hand, secure data saving is an issue in cloud servers. In recent years, many approaches to increase the security in each level of memory hierarchy have been introduced [14], [15]. The nonvolatile

property of PCM memories can enable unauthorized access to confidential data by an attacker after the power shutdown of the system. In DRAM memories, the data retention time can be of several seconds after the shutdown. The data retention time in PCM cells could be many years after shutting down the system imposing a security challenge for these memories. One of the well-known methods to prevent unauthorized access to data in the memory system is data encryption. Advanced Encryption Standard (AES) describes encryption (ENC) and decryption (DEC) algorithms, which encrypt (decrypts) data blocks before storing (reading) to (from) the memory [14]. The main feature of the AES algorithm is an avalanche effect, where flipping a bit in the input causes the toggling of the output bits by 50%. Because of a high bit flip rate in the encrypted memory, this effect (using AES technique on PCM) lowers the PCM lifetime considerably [16]. Methods such as selective encryption [15] have been introduced to address this issue. If the written value on the PCM cell matches the stuck-at fault value of that cell, the error correction unit does not need to correct the error of this cell for a read operation.

we present a simple energy-efficient method to tolerate the stuck-at faults in the PCM main memory when encrypted data are stored on it. Because of the random distribution of stuckat faults in PCM, generating random values helps us to increase “ST-R” in each cell [9], therefore we suggest a method based on shift operation, which generates random values to match the stuck-at faults. The proposed method, which is among the techniques of the third category of hard fault mitigation technique, is called RandShift.

2. RELATED WORK

2.1.Main Memory Security

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 [17]. This algorithm is fast and powerful than the other symmetric encryption algorithms. The input blocks in AES are of 128-, 192-, 256-bits, whose number of rounds are 10, 12, and 14, respectively. Each round includes AddRoundKey, Substitute, ShiftRows, and MixColumn items, which all are high energy consumption operations [17]. One-time pad (OTP) is a method to accelerate the encryption process as well as increase the security of the encrypted data [18]. The details of hardware implementation of this technique are shown in the left of Fig. 5 (OTP). For the write operation, first, the cache line is XORed with the generated OTP by the AES module and then stored in the memory. Likewise, for the read operation, the memory line is XORed with the OTP and then transferred to the last level cache (LLC). The cache counter and the line address are used to make this technique a powerful encryption approach based on temporal and spatial variations, respectively.

2.2 Error Correction in PCM Main Memory

A study in [19] on PCM main memory showed that the hard fault rate, such as stuck-at faults, is at least 10× higher than the soft error rate. Many wear-leveling techniques have been suggested to postpone the occurrence of the faults on PCM memory cells (see [16], [20]). One of the basic methods for single bit error correction in memory is ECC. Typically, this type of error correction is used for soft errors in

https://ijgst.com.2024.v13.i2.pp1042-1050

DRAM memories using parity calculation [9]. While it is a proper technique for the single-bit error correction, it is not suitable for multibit error correction in faulting PCM cells. For multibit error correction cases, ECP has been proposed [9]. ECP has a pointer to each fault location in the memory block and one bit for its correct value. For example, for the 512-bit memory block with six faults, ECP leads to an area overhead of 11.9%. The SAFER method suggested in [6] clustered the memory block into some groups, where each group has one fault to solve multibit error correction. Therefore, it corrected each fault with inverting the data to increase the "STR." In [11], a symbol-shifting method that mitigates the hard fault in both SLC (single level) and MLC (multilevel) PCMs was proposed. This article inverts data in SLCs and shifts data in each MLC to increase the rate of "ST-R" values. RDIS [10] finds an invertible set of data bits, where the number of "ST-W" cases due to these bits is maximum. After finding this set that includes both "ST-W" and Not faulty (NF) bits, the bits of this set are inverted leading to maximum hard fault coverage. In [12], the counter advance method has been proposed leading to "ST-R" increase in secure PCM main memory. This article exploited the randomness feature of the AES encryption to generate data block values that are equal to stuck-at faults. For generating data, for which some of its bits should be equal to stuckat bits, it regenerated AES encryption with different counters. The data generation is stopped, if the number of "ST-W" is zero or if the number of iterations exceeds a predefined maximum iteration number. The counter advance method recomputes AES encryption with a new counter value with the granularity level of the memory block sizes of 512 bits for the row-level and 128, 64, and 32 bits for the word-level. Hence, while the implementation of this method is simple, it consumes significant amount of energy due to the large number of AES re-computations.

II.EXISTING SYSTEM AND PROPOSED SYSTEM

1.EXISTING SYSTEM

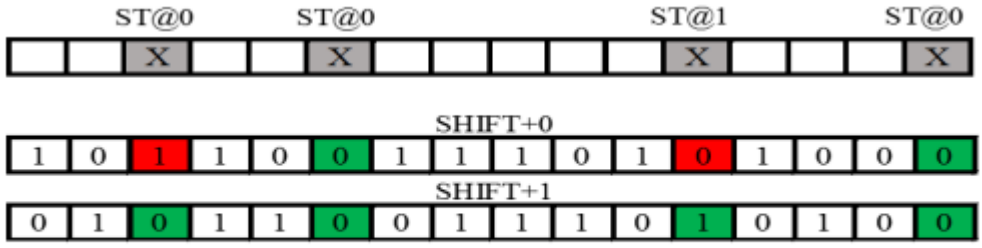

Fig. 1. Representation of ST-W and ST-R ideas.

The Advanced Encryption Standard (AES) is a symmetric encryption algorithm established by the National Institute of Standards and Technology (NIST) in 2001. It replaced the Data Encryption Standard (DES) due to its vulnerability to brute-force attacks. AES has become a widely used and trusted encryption standard for securing sensitive data.

International Journal of Gender, Science and Technology
ISSN: 2040-0748
UGC Care Group I Journal.
Vol-13 Issue-02 Dec 2024
https://ijgst.com.2024.v13.i2.pp1042-1050

2.PROPOSED SYSTEM

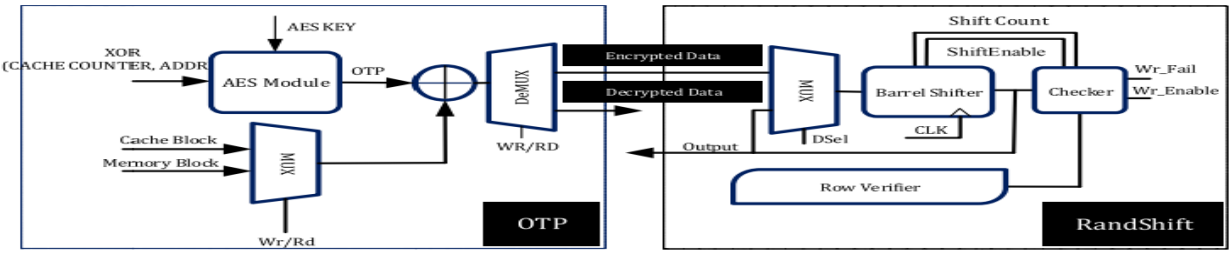

Fig 3. Full architecture of RandShift.

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 in Fig. 1, 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. As stated previously, 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. In Fig. 2, the average correlation coefficient of AES-128 encryption for 50K 128-bit encrypted memory data in the "astar" benchmark from CPU2006 is shown [22]. As Fig. 2 indicates, the illustrated average is very close to zero implying a low relationship between the output data in each iteration of the AES encryption method. The randomness feature of the output value in the AES 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., circular shifting 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 to regenerate the encrypted data. As mentioned before, due to the possibility of having to do many 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.

Randomness property of AES encryption, there is no dependency between the previous and the next data generations at

each time in the ReGen method. 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. Hence, we used a simulation approach to find the fault coverage probability of the proposed method. Fig. 3 shows the probability of matching with stuck-at faults 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. The hardware implementation of RandShift comprising Encrypt/ Decrypt and Shifter units is depicted in Fig. 3. After generating the encrypted data using the OTP in the Encrypt/Decrypt unit, the data are sent to the Shifter unit. The RowVerifier unit provides the faults' position and the value to the Checker unit. The Checker unit checks the match between the shifted data bit values and the value of the faults, which has been specified by the RowVerifier. The Shifter unit is a simple barrel shifter implemented by multiplexers. The RandShift method may be applied at the row- or word-level. In the case of the row level, all the data in the row (512 bits in this article) are shifted entirely, while in the case of the word level, each word (64 or 128 bits in this article) of a row is shifted independently.

III. RESULTS AND ANALYSIS DISCUSSION

FIG1.SIMULATION RESULT OF RANDHSIFT AES ALGORITHN

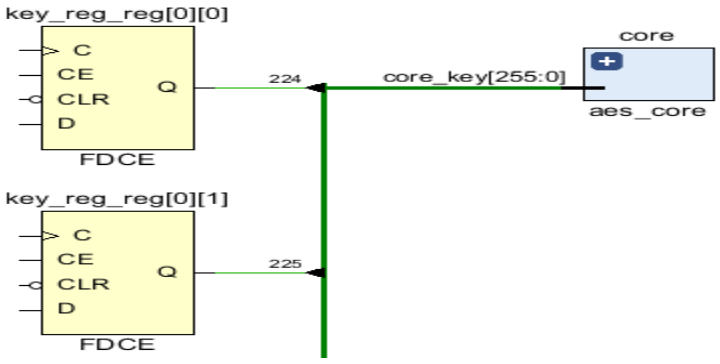

FIG2.RTL OF RANDHSIFT AES ALGORITHM

International Journal of Gender, Science and Technology
ISSN: 2040-0748
UGC Care Group I Journal.
Vol-13 Issue-02 Dec 2024
https://ijgst.com.2024.v13.i2.pp1042-1050

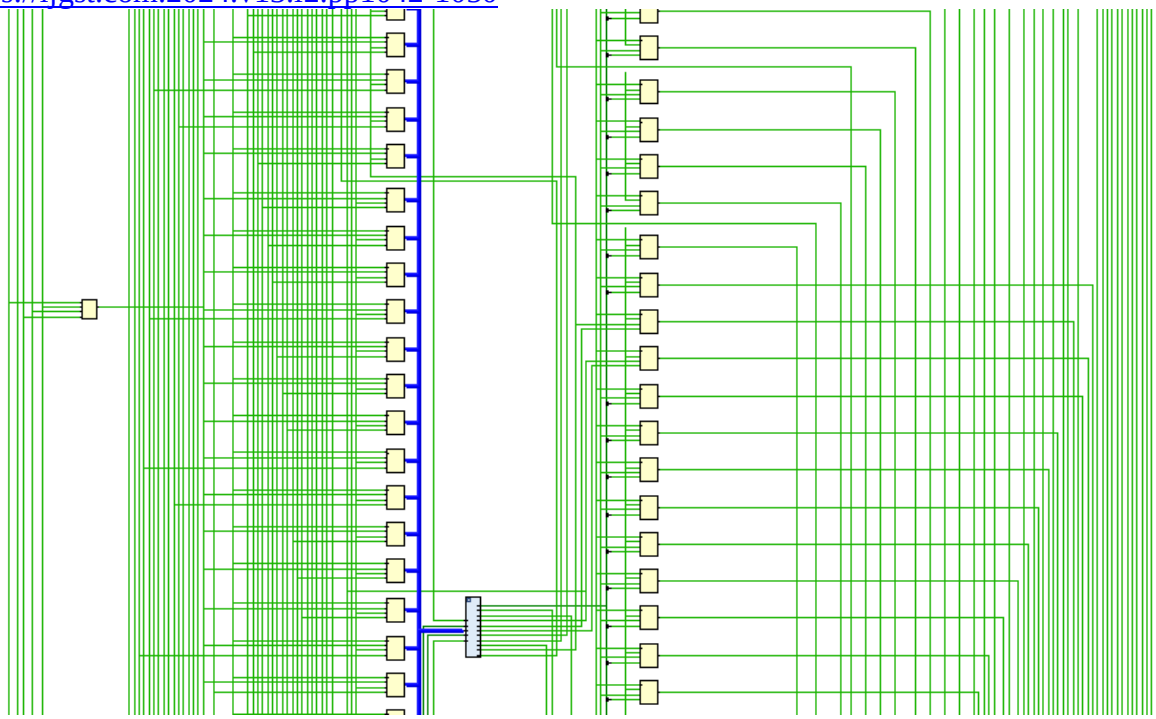

FIG.3.SYNTEHSIS DESIGN OF RANDHSIFT AES ALGORITHM

Resource	Utilization	Available	Utilization %
LUT	3224	117120	2.75
FF	2987	234240	1.28
IO	76	204	37.25
BUFG	1	352	0.28

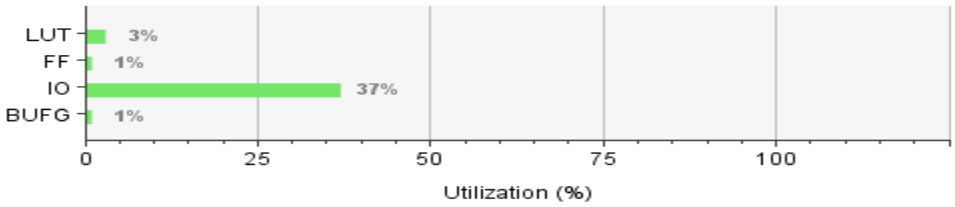

FIG.4. DESIGN UTILIZATION OF LUTS AND FFS RANDHSIFT AES ALGORITHM

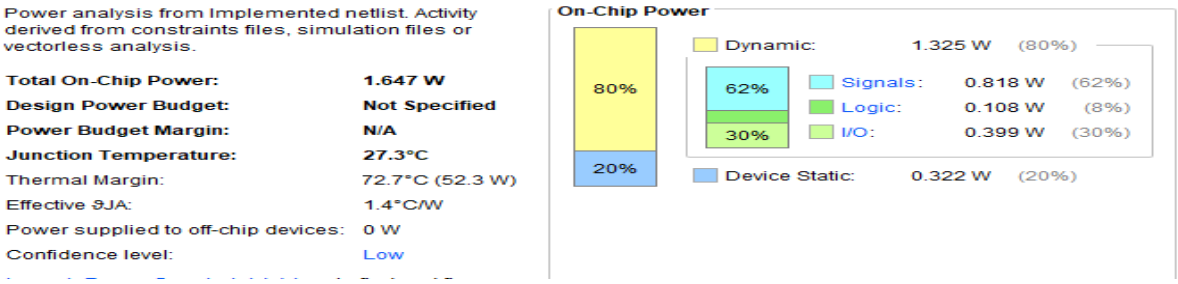

FIG.4. DESIGN POWER REPORT RANDHSIFT AES ALGORITHM

CONCLUSION

we proposed a method employing the randomness feature of AES encryption as well as rotational shift operation to tolerate hard faults in nonvolatile memory cells. This method, which was called RandShift, enjoyed the simple hardware implementation and low energy consumption. It limited the need for exploiting powerful error correction methods, such as ECC and ECP. The results of our comparative study showed up to 82% lower energy consumption for RandShift when obtaining about the same fault coverage as that of the state-of-the-art technique.

REFERENCES

[1] K. Lim, J. Chang, T. Mudge, P. Ranganathan, S. K. Reinhardt, and T. 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 stuck at 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

https://ijgst.com.2024.v13.i2.pp1042-1050

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:

[19] Z. Zhang, W. Xiao, N. Park, and D. J. Lilja, "Memory module-level testing and error behaviors for phase change memory," in Proc. 30th Int. 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.

[21] N. Binkert et al., "The gem5 simulator," ACM SIGARCH Comput. Archit. News, vol. 39, no. 2, pp. 1–7, 2011.

[22] J. L. Henning, "SPEC CPU2006 benchmark descriptions," ACM SIGARCH Comput. Archit. News, vol. 34, no. 4, pp. 1–17, Sep. 2006.

[23] (2016). NanGate—The Standard Cell Library Optimization Company. [Online]. Available: http://www.nangate.com/