ENGINEERING IN ADVANCED RESEARCH SCIENCE AND TECHNOLOGY

ISSN 2352-8648
Vol.03, Issue.01
September-2021
Pages: -204-216

EFFICIENT AND IMPROVED SECURITY FOR NON-VOLATILE MAIN MEMORY USING DSSC

[1]KUNCHE PUSHPA LATHA, [2]V V SUBHASH

[1] PG Student, Dept. of ECE, Kakinada Institute of Engineering and Technology for Women, KAKINADA, A.P

[2] Assistant professor, Dept. of ECE, Kakinada Institute of Engineering and Technology for Women, KAKINADA, A.P

ABSTRACT: The use of hardware encryption and new memory technologies such as phase change memory (PCM) are gaining popularity in a variety of server applications such as cloud systems. While PCM provides energy and density advantages over conventional DRAM memory, it faces endurance challenges. Such challenges are exacerbated when employing memory encryption as the stored data is essentially randomized, losing data locality and reducing or eliminating the effectiveness of energy and endurance aware encoding techniques. This results in increasing dynamic energy consumption and accelerated wear out. As an enhancement of this concept, new algorithm named Data Segmentation Section Code (DSSC)) based on divide-symbol is proposed to provide enhanced memory reliability. This algorithm for the detection and correction of multiple transient faults in volatile memories with low cost implementation. Data Segmentation section code is an Error Correction code based on two-dimensional code. The code in this codifies 16 data bits in 32 bits. Thus only parity bits are used for encoding data bits to reduce the area and time conception.

KEYWORDS: phase change memory, Data Segmentation Section Code, Fault tolerant, Random Access Memory.

INTRODUCTION: Faults in a scattered embedded system can be permanent, intermittent or transient. Permanent faults cause long-term malfunctioning of components. These faults emerge for a short time. Causes of intermittent faults are within system boundaries, while causes of transient faults are external to the system. They might damage data or lead to logic miscalculations, which can outcome in a fatal failure. Due to their higher rate, these faults cannot be addressed in a cost-effective way by applying traditional hardware-based fault tolerance techniques suitable for tolerating stable faults. Embedded systems along with fault tolerance have to be carefully designed and optimized, in order to assure strict timing requirements without exceeding a definite limited amount of resources. Moreover, not only performance and cost related requirements have to be considered but also other issues such as debug ability and testability have to be taken into account. Failures in VLSI systems might result from varied types of faults that can be classified as either soft (transient) or hardware ones. Transient faults are induced by temporary environmental surroundings, such as cosmic rays, EMI and for example cause information alteration in

memory elements. Permanent faults are the result of irreversible device and circuit changes, such as the following: Electromigration, which causes thinning and eventual open circuit of metal tracks. Hot carrier effect, which causes shift in device threshold voltage and it does convey conductance. Time dependant dielectric breakdown, which causes gate oxide to substrate short circuit. Increasing the yield of ICs proves especially important for new designs and manufacturing processes, which have a high density of process-induced defects and consequently a low yield. Yield improvements of early prototypes of an IC can reduce the product's introduction time and determine its commercial success. Defect tolerance has proved successful in such cases, and spectacular 30-fold increases in yield have been reported. Yield improvements due to defect tolerance tend to decrease as the manufacturing process matures. But even mature processes with lower defect densities have experienced 1.5 to 3 fold yield increases, proving the effectiveness of defect- tolerance techniques[5]. The solution to security and privacy problems is to include security features such as device identification, device/user authentication, and data encryption. These security functions are often based on the cryptographic algorithms, including public-key cryptography and symmetric cryptography, which occupy processing power and increase power and energy consumption. In contrast, IoT devices are supposed to be constrained low-cost devices with limited processing power, limited memory footprint, and even limited power/energy budget, for example, power-harvesting devices and batterybased devices. This leads to the importance of optimizing cryptographic algorithms in hardware for cost, throughput, and especially power and energy consumption. However, cost, throughput, and power/energy consumption are different features which are hard to achieve at the same time. In this paper, we chose to find a good tradeoff among them for advanced encryption standard (AES) [4], a widely-used block cipher for emerging IoT proposals, such as IEEE 802.15.4 [5], LoraWAN [6], Sigfox [7], and ZWave [8]. We also made comparison with an extreme lightweight data encryption algorithm PRESENT [9], a candidate for highly constrained devices. PRESENT is a hardware-oriented block cipher with reduced security level but it has small area footprint and very lowpower consumption. However, to the best of our knowledge, lightweight block ciphers, such as PRESENT, are not yet adopted to any IoT proposals. From its standardization in 2001 by the U.S. National Institute of Standards and Technology (NIST) to replace data encryption standard, AES has been studied by researchers in terms of security, performance, and hardware/software implementations. In terms of security, different IoT applications may require different security levels with different power/energy budgets and different throughputs. At the algorithmic level, security level depends on the design of the algorithm and the length of the key. Logic ICs: The development of efficient defect tolerant designs for random logic ICs like microprocessors is considerably more complex than for memory ICs. However, if some regularity exists in the structure of a given logic circuit, it might be possible to incorporate redundancy. A natural target for defect tolerant designs - programmable logic arrays (PLAs) have a regular structure in VLSI chips. The control sections of many microprocessors use large PLAs. Some de- signs have employed PLAs with as many as 50 inputs and almost 200 product terms. Since these PLAs require large silicon areas, the

incorporation of redundancy in their design can considerably improve the overall yield. Researchers have investigated defect tolerant designs of PLAs and proposed adding spare programmable product lines, input lines, and output lines to protect against all types of possible defects. This technique resembles the redundant row/column scheme for memory ICs.

LITERATURE SURVEY: Electrical memory testing consists of parametric testing, which includes testing DC and AC parameters, IDDQ and dynamic testing for recovery, retention and imbalance faults [9]. DC and AC parametric tests are used to verify that the device meets its specifications with regard to its electrical characteristics, such as voltage, current, and setup and hold time requirements of chip's pins. Since embedded memories in SOCs usually do not have their I/O ports directly connected to chip's pins, parametric testing for embedded memories is not a necessity. IDDQ and dynamic testing [5] need a detailed description of the specific process technology. Additional information on electrical testing can be found in [8]. This thesis focuses on technology-independent functional memory testing, whose purpose is to verify the logical behavior of a memory core. Because functional memory testing allows for the development of cost-effective short test algorithms (without requiring too much internal knowledge of the memory under test), it is widely accepted by industry as a low-cost/high-quality solution. Support multiple test algorithms: The conventional MBIST approaches usually implement a single March test algorithm. However, deep submicron process technologies and design rules introduce physical defects that are not screened when using the memory test algorithms developed for previous process generations. Therefore MBIST architectures should be programmable to support multiple memory test algorithms to increase the fault coverage and to find the most suitable algorithms for the manufacturing process at hand. 2. Diagnosis and repair support: Diagnosis support in an MBIST architecture is mandatory for manufacturing yield enhancement for new process technology and a rapid transition from the yield ramp phase to the volume production phase [19]. Furthermore, since embedded memories are subject to more aggressive design rules, they are more prone to manufacturing defects (caused by process variations) than other cores in an SOC. For large embedded memory cores, the manufacturing yield can be unacceptable low (e.g., for a 24Mbits memory core, the yield is around 20% [5]). Hence, to achieve a certain manufacturing yield, in addition to diagnosis support, it is also beneficial to introduce self-repair features comprising redundant memory cells. TPG methods like exhaustive, pseudo-random and fault simulation techniques (Hurst, 1998; Roth, 1998) are used in the test vector generation process. TPG is the process of generating the test vectors required to stimulate a circuit at the primary inputs so that effect of the considered fault (the fault effect) is propagated to the primary outputs. A difference between the fault free and faulty circuit can then be detected. It is common to derive a minimal set of test vectors as it will reduce the overall test set size and hence test time. In SOC, designers can specify the test speed, fault coverage, diagnostic options and test length for testing any random logic block. Power dissipation during the testing is one of most important issue (Crouch at. el.1999).

PHASE-CHANGE MEMORY Phase-change memory (PCM) is a key enabling technology for non-volatile electrical data storage at the nanometer scale. A PCM device consists of a small active volume of phase-change material sandwiched between two electrodes. In PCM, data is stored by using the electrical resistance contrast between a high-conductive crystalline phase and a low-conductive amorphous phase of the phase-change material. The phase-change material can be switched from low to high conductive state, and vice-versa, through applying electrical current pulses. The stored data can be retrieved by measuring the electrical resistance of the PCM device. An appealing attribute of PCM is that the stored data is retained for a very long time (typically 10 years at room temperature), but is written in only a few nanoseconds. This property could enable PCM to be used for non-volatile storage such as Flash and hard-disk drives, while operating almost as fast as high-performance volatile memory such as DRAM.

EXISTING METHOD:

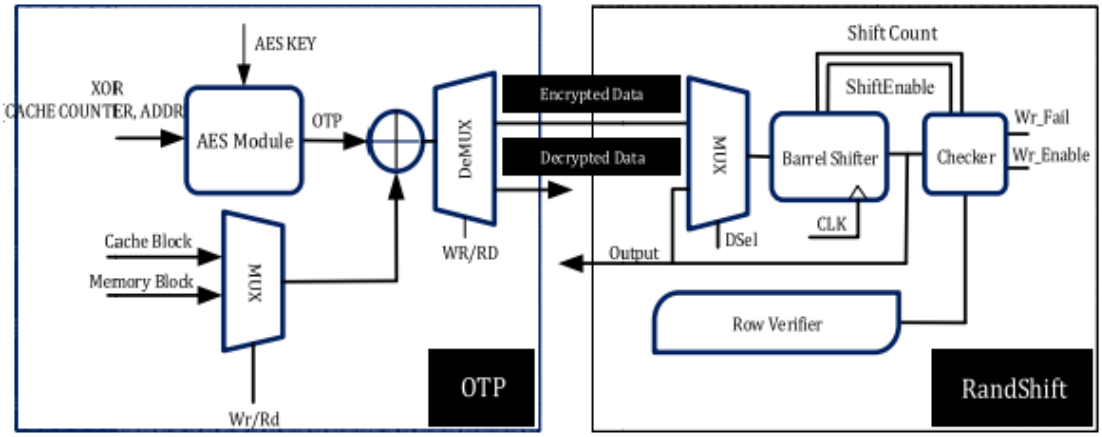

Fig1. Full architecture of RandShift

The hardware implementation of RandShift comprising Encrypt/ Decrypt and Shifter units is depicted in above Fig. After generating the encrypted data using the OTP in the Encrypt/Decrypt unit, the data are sent to the Shifter unit. The Row Verifier 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 Row Verifier. 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 bits in this article) of a row is shifted independently.

ADVANCED ENCRYPTION ALGORITHM: The Advanced Encryption Standard (AES) specifies a FIPS-approved cryptographic algorithm that can be used to protect electronic data. The AES algorithm is a symmetric block cipher that can encrypt (encipher) and decrypt (decipher) information. Encryption converts data to an unintelligible form called cipher text; decrypting the cipher text converts the data back into its original form, called plaintext. The AES algorithm is capable of using cryptographic keys of 128,

192, and 256 bits to encrypt and decrypt data in blocks of 128 bits.This standard specifies the Rijndael algorithm, a symmetric block cipher that can process data blocks of 128 bits, using cipher keys with lengths of 128, 192, and 256 bits. Rijndael was designed to handle additional block sizes and key lengths; however they are not adopted in this standard. Throughout the remainder of this standard, the algorithm specified here in will be referred to as "the AES algorithm." The algorithm may be used with the three different key lengths indicated above, and therefore these different "flavours" may be referred to as "AES-128", "AES-192", and "AES-256".

This specification includes the following sections:

1. Definitions of terms, acronyms, and algorithm parameters, symbols, and functions.
2. Notation and conventions used in the algorithm specification, including the ordering and numbering of bits, bytes, and words.
3. Mathematical properties that is useful in understanding the algorithm.
4. Algorithm specification, covering the key expansion, encryption, and decryption routines.
5. Implementation issues, such as key length support, keying restrictions, and additional block/key/round sizes.

The standard concludes with several appendices that include step-by-step examples for Key. At the start of the Cipher, the input is copied to the State array using the conventions. After an initial Round Key addition, the State array is transformed by implementing a round function 10, 12, or 14 times (depending on the key length), with the final round differing slightly from the first Nr -1 rounds. The final State is then copied to the output.

BLOCK DIAGRAM FOR ENCRYPTION &DECRYPTION:-

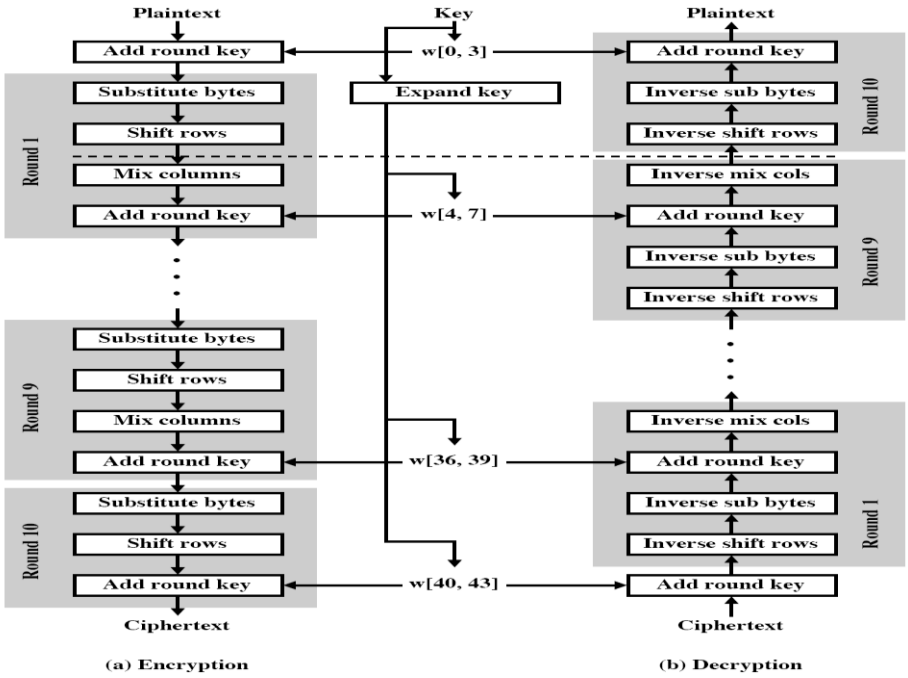

Fig 2: Encryption & Decryption block diagram

Totally, 10 bocks are used for encryption and decryption purpose. Each round performs same operations with different variable constants. In order to perform these operations, a separate key is needed. All 10 blocks operations together form encryption/decryption operations. Concerning round key generation, either the keys are precomputed and stored in the circuit or the key generation module calculates the sequence of keys. For the latter case, AES is modified in such a way that during SELF-TEST, TPG, and SA modes, the tenth round key is used as the primary key for the next round key generation. In thisway, during self-test, the key generation module receives as many different stimuli as rounds. One of the operations of the crypto-algorithm is a substitution function that is implemented by S-boxes. S-boxes represent the largest part of the crypto-cores. Their inputs are independently fed by a subpart of the round inputs.We can therefore assume that they are fed by a random source, receiving a pattern every clock cycle. S-box needs k deterministic patterns to be fully tested and it receives one random pattern every clock cycle. Other parts of the round module (mainly wires and XOR operations) receive one pattern every clock cycle as well. Since the other parts have lower complexity than the S-boxes, it can be expected that they will be fully tested by the time the S-boxes received a sufficient number of test patterns.

RAND SHIFT Method:

Fig3. Representation of ST-W and ST-R ideas.

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 above Fig, 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.

PROPOSED METHOD This concept proposes a new algorithm named as Data Segmentation Section Code (DSSC)) based on divide-symbol is proposed to provide enhanced memory reliability. This

algorithm for the detection and correction of multiple transient faults in volatile memories with low cost implementation.

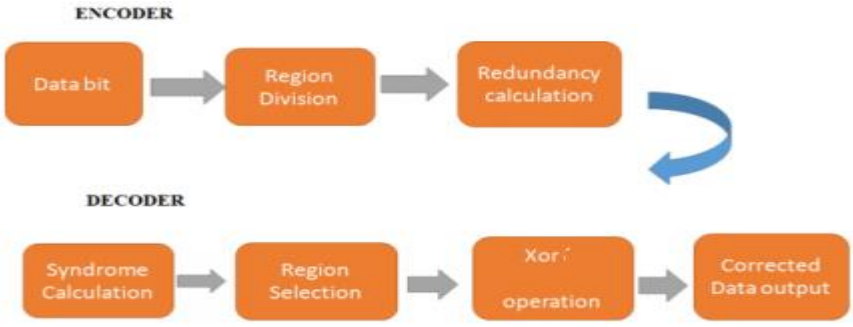

Fig4 Proposed DSSC architecture

Data Segmentation section code is an Error Correction code based on two-dimensional code. The code in this codifies 16 data bits in 32 bits. Thus only parity bits are used for encoding data bits to reduce the area and time conception.

DATA SEGMENTATION SECTION CODE ENCODING PROCESS:

Below Fig shows the structure of 32 bits of data encoded by Data Segmentation Section Code. The cells in gray was data bits, they were divided into four groups (A, B, C, D).

A1	A2	A3	A4	Di1	Di3	CbA13	CbA24
B1	B2	B3	B4	Di2	Di4	CbB13	CbB24
C1	C2	C3	C4	P1	P3	CbC13	CbC24
D1	D2	D3	D4	P2	P4	CbD13	CbD24

Fig5 DSSC Encoded data model.

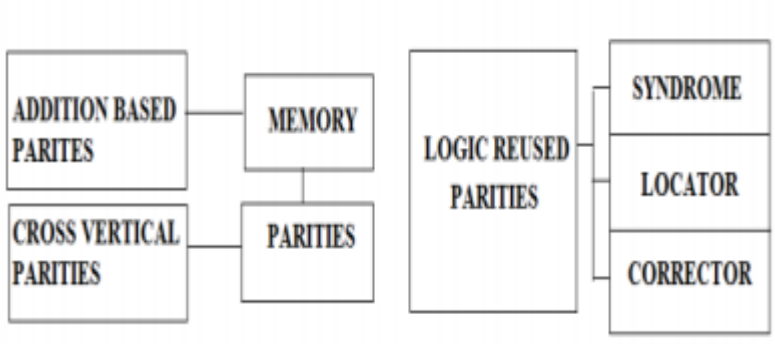

Fig6 proposed architecture

The cells in green are the Diagonal bits (Di) analyed with XOR operations in specific data bits:

$$Di_1 = A_1 \oplus B_2 \oplus C_1 \oplus D_2$$
$$Di_2 = A_2 \oplus B_1 \oplus C_2 \oplus D_1$$
$$Di_3 = A_3 \oplus B_4 \oplus C_3 \oplus D_4$$
$$Di_4 = A_4 \oplus B_3 \oplus C_4 \oplus D_3$$

The cell in blue wasParity bits (P) analyze by XOR operations in the data bits columns:

$$P_1 = A_1 \oplus B_1 \oplus C_1 \oplus D_1$$
$$P_2 = A_2 \oplus B_2 \oplus C_2 \oplus D_2$$
$$P_3 = A_3 \oplus B_3 \oplus C_3 \oplus D_3$$
$$P_4 = A_4 \oplus B_4 \oplus C_4 \oplus D_4$$

The cells orange is a Check bits (Cb) analyzed by XOR operations in interleaved bits of each group:

$$CbA_{13} = A_1 \oplus A_3$$
$$CbA_{24} = A_2 \oplus A_4$$
$$CbB_{13} = B_1 \oplus B_3$$
$$CbB_{24} = B_2 \oplus B_4$$
$$CbC_{13} = C_1 \oplus C_3$$
$$CbC_{24} = C_2 \oplus C_4$$
$$CbD_{13} = D_1 \oplus D_3$$
$$CbD_{24} = D_2 \oplus D_4$$

The redundancy bit was analyzed and, the encoding process ends and the 32 bits was stored. The Dibits and Cbbits are arranged between the data bits and Cbbits, in order to develpo the efficiency of Data Segmentation Section Code against Multiple Cell Upsets characterized by adjacent error patterns. Figure describes the mains elements of the parity operation of the Data Segmentation Section Code encoder. The decoding process of DSSC is divided into three steps: Syndrome appraisal of the redundancy bits - The syndrome appraisal consists of a XOR operation between the redundancy data stored and the recalculated redundancy bits (RDi, RP, and RCb). So the values for the Syndrome of Diagonal , Parity and Check bits are estimated by:

$$SDi = Di \oplus RDi$$
$$SP = P \oplus RP$$
$$SCb = Cb \oplus RCb$$

Verification of error decoding conditions - After the analysing of the Syndromes, one of these two conditions need to be satisfied before the error correction execution:

(i) SDi and SP vectors have at least one value similar to one;

(ii) more than one SCb value was similar to one. The conditions permite the algorithm for identify the error for data bits region. Selection and correcting the wrong data region and correction processes -In this decoding process a distinct region is selected in the data bit and corrected. The region are divided into regions and it is shown below. They split the data bits in three regions and it was explained so as to select a definitive group of bit for the correction process. This reduce the area and the time conception.

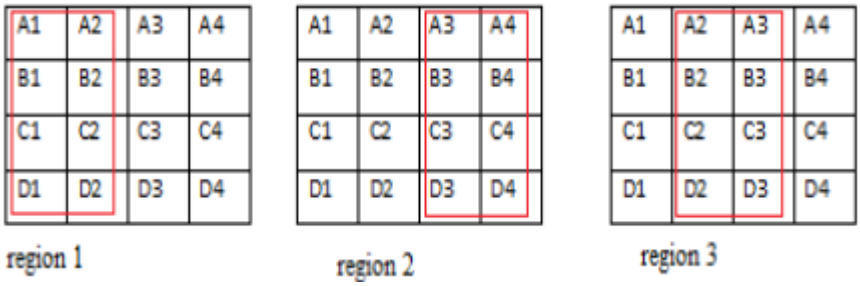

Fig7. Regions of data bits.

The above fig (a), (b) and (c) show that region 1, 2 and 3 are formed by data bits distributed in columns (1 and 2), (3 and 4) and (2 and 3), respectively. The selection of which region will be corrected is defined by the integer sum (+) of specific bits of SDiand SP. Table presents a group of equations which describes the criterion for region selection of DSSC, where the region with more syndrome bits equals to 1 is be declared as the wrong one (Region 1 or Region 2). If the sum of the equations presents equal value, then the Region 3 is selected.

Region selected	Criterion to selection
Region 1	$(SDi_1 + SDi_2 + P_1 + P_2) > (SDi_3 + SDi_4 + P_3 + P_4)$
Region 2	$(SDi_1 + SDi_2 + P_1 + P_2) < (SDi_3 + SDi_4 + P_3 + P_4)$
Region 3	$(SDi_1 + SDi_2 + P_1 + P_2) = (SDi_3 + SDi_4 + P_3 + P_4)$

Table 1: Region selection criterion.

For regions 1 and 2, the correction procedure consists in a XOR operation between the region selected and the SCbsmatrix. Region 3 is a special case where it is strictly necessary that neither of all SDi and SP bits are null, even if the condition II of step 2 is satisfied. Note that Region 3 has its first column formed by values with the even index (2), meaning that the correction performed has to be different from the other regions. If region 3 is selected, the correction procedure must be performed by SDi with shifted positions, to align the indexes of SCbs with the matrix of Region 3.

RESULTS:

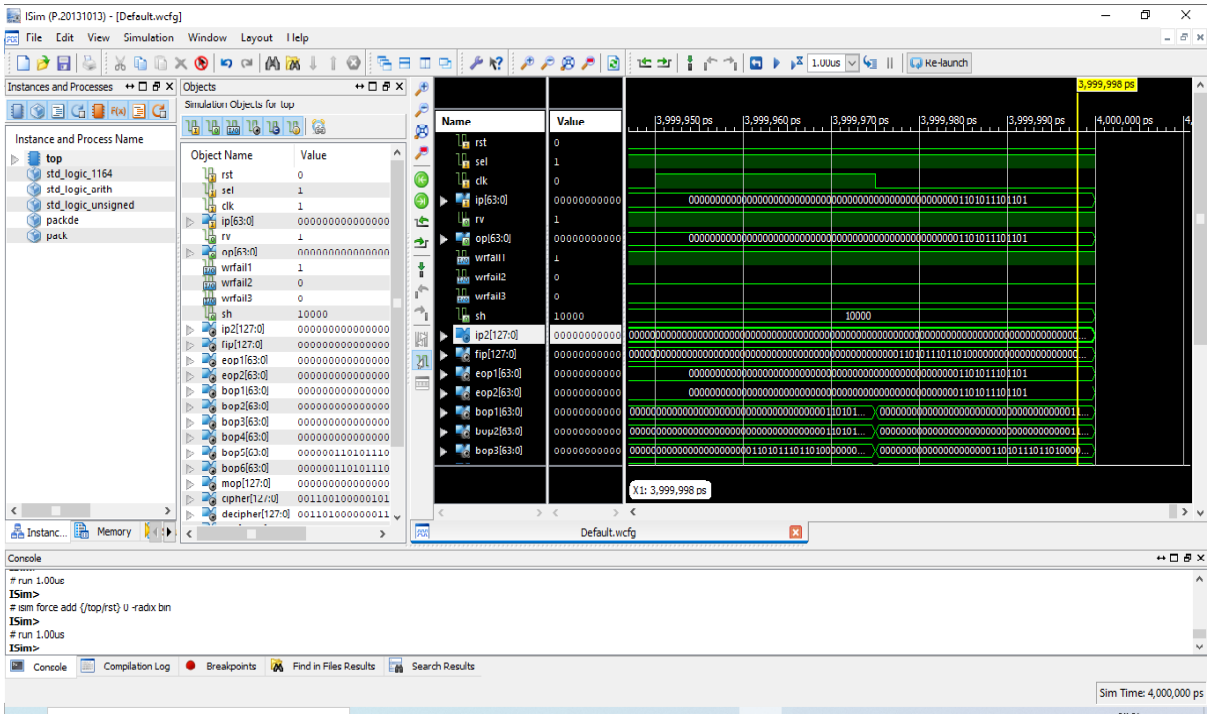

Fig8: Existing simulation output

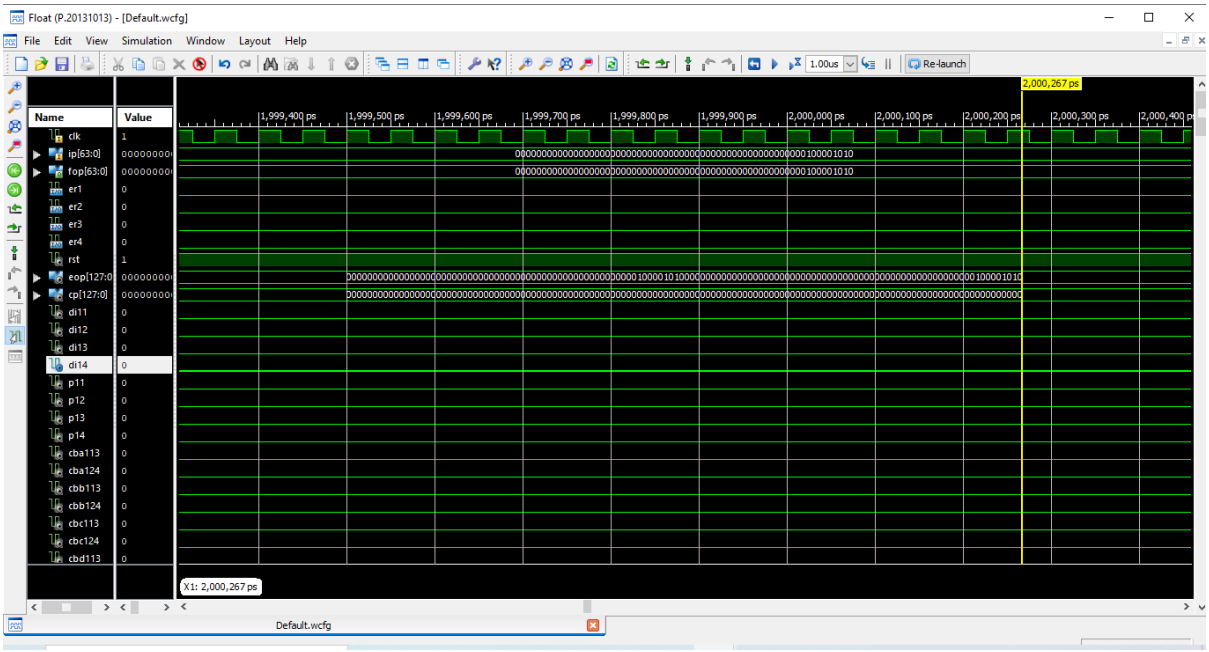

Fig9: Proposed simulation output

CONCLUSION: This project proposes with Data Segmentation Section Code this is an error detection and correction code to memory devices exposed to MCUs. By using this code on parity code and enclosed to handle with more Multiple Cell Upsets. Data Segmentation Section Code exhibited the lowest cost of coading, low area and improve time conception. Though, hamming and Extended Hamming codes in the ECCs Matrix brought advantages in error coverage it increased heavily the cost of both codes, when compared with Data Segment.

FUTURE SCOPE: The study carried out in this paper can be extended to many other potential fields. Major possibility is to develop an error correcting system which can provide better performance, with less delay overhead, lower power requirements and less area consumption. The study can be carried out by pipelining the existing codes in to an efficient form so that the delay overhead is reduced. Also by changing the adders and other elements used in realization, the area can be reduced, by the proper implementation of the above indicated two ideas the power consideration can also be considerably reduced.

REFERENCES:

[1] P.Hazucha, C. Svensson. Impact of CMOS technology scaling on the atmospheric neutron soft error rate. IEEE Transaction on Nuclear Science, v. 47, n. 6, pp. 2586-2594, Dec. 2000. [2] K.LaBel, C. Barnes, C. Marshall, A. Johnston, R. Reed, J. Barth, C. Seidleck, S. Kayali, M. O'Bryan. A roadmap for NASA's radiation effects research in emerging microelectronics and photonics. IEEE Aerospace Conference, v. 5, pp. 535-545, 2000. [3] P.Ferreyra, C. Marques, R. Ferreyra, J. Gaspar. Failure map functions and accelerated mean time to failure tests: New approaches for improving the reliability estimation in systems exposed to single event upsets. IEEE Transaction on Nuclear Science, v. 52, n. 1, pp. 494- 500, Feb. 2005. [4] V.Gherman, S. Evain, F. Auzanneau, Y. Bonhomme. Programmable extended SEC-DED

codes for memory errors. IEEE VLSI Test Symposium (VTS), pp. 140-145, 2011. [5] D.Radaelli, H. Puchner, S. Wong, S. Daniel. Investigation of multibit upsets in a 150 nm technology SRAM device. IEEE Transaction on Nuclear Science, v. 52, n. 6, pp. 2433-2437, Dec. 2005. [6] A.Chugg, M. Moutrie, R. Jones. Broadening of the variance of the number of upsets in a read-cycle by MBUs. IEEE Transactions on Nuclear Science, v. 51, n. 6, pp. 3701-3707, Dec. 2004. [7] J.Maestro, P. Reviriego. Study of the effects of MBUs on the reliability of a 150 nm SRAM device. ACM/IEEE Design Automation Conference (DAC), pp. 930-935, 2008. [8] R.Hentschke, F. Marques, F. Lima, L. Carro, A. Susin, R. Reis, Analyzing area and performance penalty of protecting different digital modules with hamming code and triple modular redundancy. Symposium on Integrated Circuits and Systems Design, pp. 95-100, 2002. [9] C.W Slayman.Cache and memory error detection,correction,andreduction techniques [10] for terrestrial servers and workstations. IEEE Electron Devices Society, v. 5 ,pp. 397 – 404, Sept. 2005 [11] M.Biberstein ; T. Etzion. Optimal codes for single-error correction, double-adjacent-error detection. IEEE Information Theory Society. v:46,pp: 2188 – 2193,sep 2000.

[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. 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/ [24] B.

Giridhar et al., "Exploring DRAM organizations for energy-efficient and resilient exascale memories," in Proc. Int. Conf. High Perform. Comput., Netw., Storage Anal., 2013, p. 23.

[24] Morteza Soltani, Mehdi Kamal , Ali Afzali-Kusha , and Massoud Pedram, "RandShift: An Energy-Efficient Fault-Tolerant Method in Secure Nonvolatile Main Memory" IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS