iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 2 **Issue:** XII **Month of publication:** December 2014

DOI:

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

Memory Protection for Encoders in OLS Codes

Michael Angelo[1], Ravi Kumar[2]
[1] M.Tech Scholar, [2]Associate Professor, Department Of Electronics and Communication Engineering

Abstract— Error correction codes (ECCs) are commonly used to look after memories next to errors. The proposed technique we are detect the one or more errors and to correct on its own bit error. Among ECCs, Orthogonal Latin Squares (OLS) codes have gain renewed interest for memory defense due to their modularity and the simplicity of the decoding algorithm that enable low down delay implementations. The general idea for achieving error detection and correction is to add some redundancy which means to add some extra data to a message, which receiver can use to check uniformity of the delivered message, and to pick up data determined to be corrupt. A significant issue is that when ECCs is used, the encoder and decoder circuits can also suffer errors. In this brief, a concurrent error detection technique designed for OLS codes encoders be proposed and evaluated. The proposed method uses the properties of OLS codes in the direction of efficiently implement a parity prediction scheme that protects the encoder.

Keywords— Concurrent error detection, error correction codes (ECC), Latin squares, majority logic decoding (MLD), parity, memory

I. INTORDUCTION

Error correction codes (ECCs) include been used to protect memories for a lot of years .There is a wide range of codes so as to are used or have been proposed for memory application. Single error correction (SEC) codes that can correct one bit per word are usually used. More advanced codes that are able to also correct double adjacent errors or double errors in general contain also been studied. The use of more complex codes that can correct additional errors is limited by their impact on delay and power, which can boundary their applicability to memory designs. To prevail over those issues, the use of codes so as to be one step majority logic decodable (OS-MLD) has freshly been proposed. OS-MLD codes can be decoded with small latency and are, therefore, used to protect memories. Among the codes so as to be OS-MLD, a type of Euclidean Geometry (EG) code has been proposed to protect reminiscences. The use of difference set code has also been freshly analysed in. Another type of code that is OS-MLD is Orthogonal Latin Squares (OLS) code .The use of OLS codes have gained renewed interest designed for interconnections, memories, and caches. The general idea for achieving error detection and correction is to add some redundancy (i.e., some extra data) to a message, which receiver can use to check consistency of the delivered message, and to pick up data determined to be corrupt. Error-detection and correction scheme can be either systematic or non-systematic .A systematic code is any error-correcting code in which the input data is embedded in the encoded output.In a systematic scheme, the transmitter sends the unique data, and attaches a fixed number of check bits (or parity data), which are derived from the data bits by some deterministic algorithm. If only the error detection is required, a receiver can simple apply the same algorithm to the received data bits and compare its output with the receive check bits; if the values do not match, an error has occurred at some point throughout the transmission. Error-correcting codes are regularly used in lower-layer communication, as well as for reliable storage in media such as CDs, DVDs, hard disks and RAM. In a non-systematic code the output does not contain the input symbols. In a system to uses a non-systematic code, the unique message is transformed into an encoded message that has at least as many bits as the unique message.Non-systematic convolution codes can provide better performance under maximum-likelihood decoding.The aim of error detection is to provide against soft errors that manifest themselves as bit-flips in memory.This is due in the direction of their modularity such that the error correction capabilities be able to be easily adapted to the error rate] or in the direction of the mode of operation. OLS codes typically require additional parity bits than additional codes to correct the similar number of errors. However, their modularity and the easy and low delay decoding implementation (as OLS codes are OS-MLD), offset this disadvantage in a lot of applications. An important subject is that the encoder and decoder circuits needed to use (ECCs) can also suffer errors. When errors affect the encoder, an incorrect word might be written into the memory. An error in the decoder can reason a correct word to be interpreted as erroneous or the additional way around, an incorrect utterance to be interpreted as a correct word. The protections of the encoders and decoders have been studied for unusual ECCs. For example, in EG codes be studied. The protection of Reed–Solomon, Hamming, and BCH encoders and decoders has as well been studied in and, and more universal techniques for systematic and cyclic codes have been projected in . Finally, the protection of encoders for SEC codes alongside soft errors was discussed in .The ECC encoder computes the parity bits, and in the majority cases the decoder starts by checking the parity bits in the direction of detect errors. This is commonly referred toward as syndrome computation. For some codes, it is probable to perform encoding and syndrome calculation serially

base on the properties of the code. However, when delays have to be low, parallel implementations are preferred. This is the case for OLS codes so as to be commonly used in high-speed applications. The reader is referred to [6] for a complete discussion of ECC encoders and decoders. After syndrome computation, when errors are detected, the relax of the decoding is done to correct the errors. These resources that generating and checking the parity bits are significant parts of the encoder and decoder circuitry. Therefore, its protection is an important issue.

II. ORTHOGONAL LATIN SQUARES CODES

OLS codes are based on the concept of Latin squares. A Latin square of size m is an m × m matrix so as to has permutations of the digits 0, 1,…, m − 1 in both its rows and columns . Two Latin squares are orthogonal if when they are superimposing every ordered pair of elements appear only once. OLS codes are resulting from OLS. These codes have k = m2 data bits and 2tm check bits, where t is the number of errors so as to the code can correct. For a double error correction code t = 2, and, therefore, 4m check bits, are used. As mentioned inside the introduction, one advantage of OLS codes is that their construction is modular. This means that to get hold of a code that can correct t +1 errors, just 2m check bits are added to the code that can correct t errors. This is able to be useful to implement adaptive error correction schemes, because discussed in and . The modular property as well enables the selection of the error correction capability designed for a given word size. As mentioned before, OLS codes be able to be decoded using OS-MLD as each data bit participates in accurately 2t check bits and each other bit participates in at the majority one of those check bits. This enables an easy correction when the figure of bits in error is t or less. The 2t check bits are recomputed and a majority vote is taken. If a value of one is obtained, the bit be in error and must be corrected. Otherwise the bit is correct. As long as the digit of errors is t or less, the remaining t −1 errors can, in the worst case, affect t −1 check bits. Therefore, still a majority of t + 1 triggers the modification of an erroneous bit. In any case, the decoding starts by recompiling the parity check bits and checking alongside the stored parity check bits.

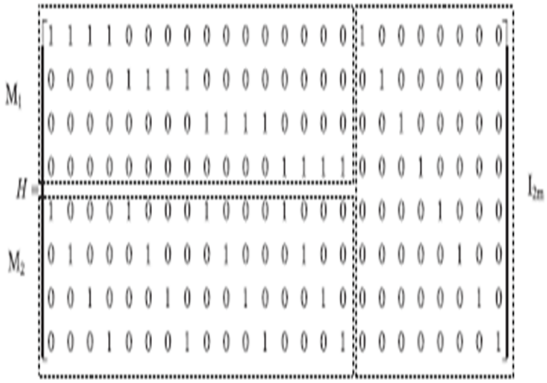


Fig 1 Parity check matrix for OLS code with k = 16 and t = 1.

The parity check matrix H for OLS codes is constructed from the OLS. As an example, the matrix intended for a code with k = 16 and 8 check bits that be able to correct single errors is shown in Fig. 1. As discussed earlier, due in the direction of the modular construction of OLS codes this matrix forms part of the H matrix for codes that be able to correct more errors. For example, to obtain a code so as to can correct two errors, eight additional rows are added to the H matrix. For an arbitrary value of k = m2, the H matrix for a SEC OLS code is constructed as follows:

$$H = \begin{bmatrix} M_1 \\ M_2 \end{bmatrix} \quad I_{2m} \qquad (1)$$

Where I2m is the identity matrix of size 2m and M1, M2 are matrices of size m × m2. The matrix M1 has m ones in each row. For the r th row, the ones are at positions (r − 1) × m + 1,(r − 1)× m + 2,…(r − 1) × m + m − 1, (r − 1) × m + m. The matrix M2 is constructed as follows:

$$M2 = [Im\ Im\ \ldots\ Im] \qquad (2)$$

For m = 4, the matrices M1 and M2 can be clearly observed in Fig. 1. The encoding matrix G is just the H matrix on which the check bits are removed

$$G = \begin{bmatrix} M_1 \\ M_2 \end{bmatrix} \qquad (3)$$

III. PROPOSED MEMORY PROTECTION TECHNIQUE FOR ENCODER

Before describing the future error detection techniques, the standard meaning of self-checking circuits that are used in this part is presented. During normal, or fault-free, operation, a circuit receives only a separation of the input space, called the input code space, and produces a separation of the output space, called the output code space. The outputs that are not member of the output code freedom from the output error space. In general, a circuit may be intended to be self-checking only intended for an assumed fault set. In this brief, we consider the responsibility set **F** corresponding to the single stuck-at fault model . A circuit is self-checking if and only if it satisfies the following properties: 1) it is self-testing, and 2) fault-secure. A circuit is self-testing if, for every fault f in the fault set **F**, present is at least one input belonging to the input code freedom, for which the circuit provides a production belonging to the output error space.A circuit is fault-secure if, for every fault f in the responsibility set **F** and for each input belonging to the input code freedom, the circuit provides the correct output, or an output belong to the output error space. The fault-secure belongings guarantee that the circuit give the correct response, or signals the presence of a fault that provides an output in the error space. Faults are always detected, since there is an input that produces an output that identifies the presence of the fault.The parity of all the check equations is just the equation obtained by compute the parity of the columns in G. For OLS codes, since every column in G has exactly 2t ones, the null equation are obtained (see, for example, Fig. 1). Therefore, the simultaneous error detection (CED) system is simply to check

$$c1 \wedge c2 \wedge c3 \wedge \cdot \cdot \cdot \wedge c2tm = 0. \quad (4)$$

This enables a proficient implementation that is not probable in other codes. For example, in a Hamming code a important part of the columns in G has an odd weight and for a number of codes the number is even larger as they are intended to have odd weights . The input code spaces of the OLS encoder correspond to the input space, since the encoder can take delivery of all the possible 2k input configurations. The output code space of the OLS encoder is collected by the outputs satisfying (4), while the output error space is the balance of the output code space. A responsibility that occurs in one of the gates composing the OLS encoder can adjust at most one of the ci check bits. When this change occurs, the OLS encoder provides outputs that do not satisfy (4), i.e., outputs belong to the output error space. Hence, these guarantee the fault-secure possessions for this circuit. Additionally, since the encoder is composed only by XOR gates, no logic masking is performed in the circuit. Therefore, when a fault is activated the error is propagating to the output. This ensures the self-testing possessions of the circuit. In order to verify if the output of the OLS encoder belongs to the output code space or the output error space, a self-examination implementation of a parity checker is used . The checker controls the equivalence of its inputs and is realized with a repetition code. The two outputs (r1, r2) are every equal to the parity of one of two disjoint subsets of the manager inputs (ci), as proposed in. When a set of inputs by means of the correct equivalence is provided, the output code takes the values 00 or 11. When the manager receives an erroneous set of inputs, the checker provides the output codes 01 or 10. Also, if a fault occurs in the manager, the outputs are 01 or 10. This guarantee the self-checking property of the parity checker. The proposed encoder is illustrate in Fig. 2

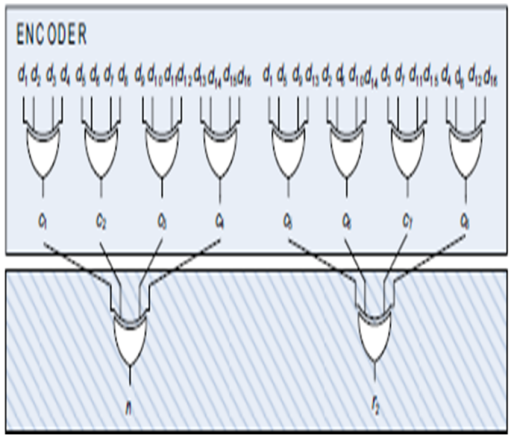


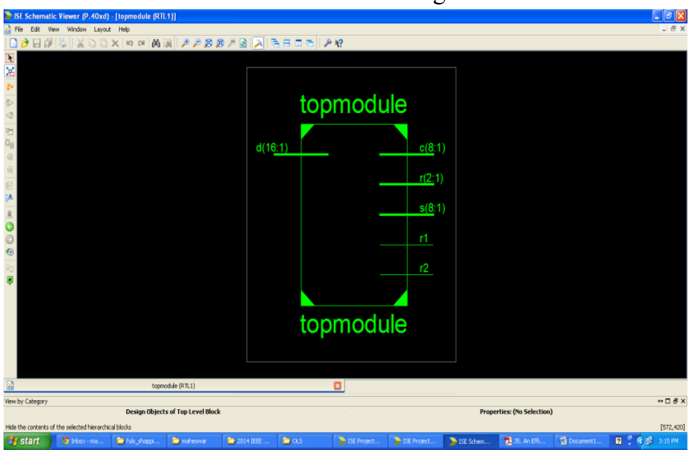
Fig: Proposed self-checking encoder for OLS code with k = 16 and t = 1.

The planned circuit can detect any error that affect an odd figure of ci bits. For a universal code, in most cases there is logic sharing in the middle of the computations of the ci bits. This means that an error may promulgate to more than one ci bit, and if the figure of bits affected is even, then the error is not detect by the proposed scheme. To avoid this subject, the computation of

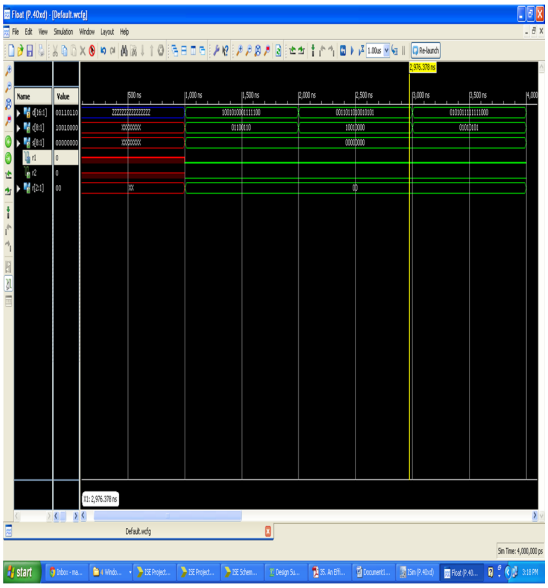
each ci bit can be complete separately. This, however, increase the circuit area of the encoder as no judgment sharing is allowed. Another option is to control the common sense in such a way that errors can only promulgate to an odd number of outputs. For OLS codes, as discussed in the preceding section a pair of data bits shares at most one equivalence check. This guarantees that there is no logic sharing in the middle of the calculation of the ci bits. Therefore, the future technique detects all errors that affect an only circuit node.

IV. SIMULATION RESULTS

Block Diagram



Simulation Output



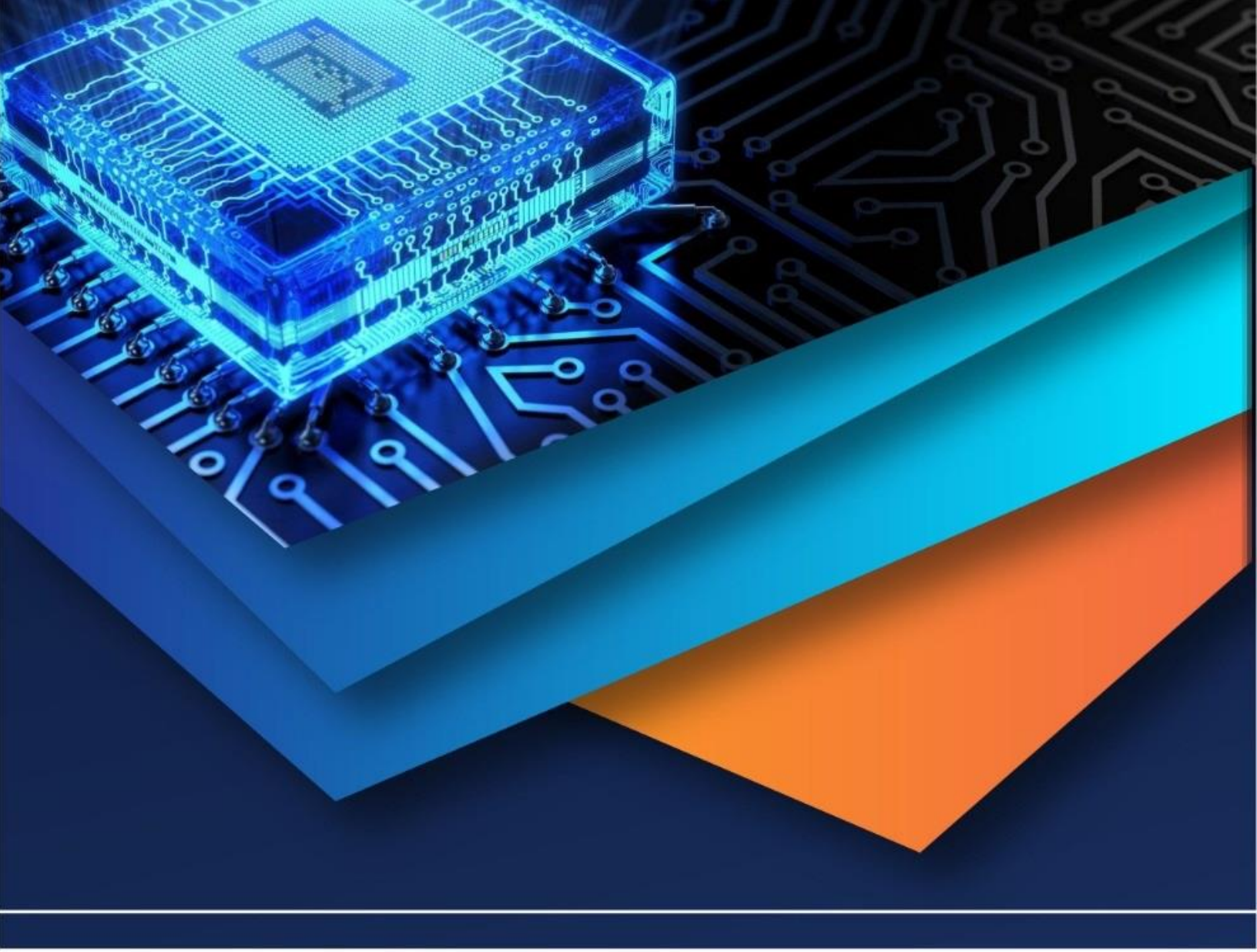
V. CONCLUSION

In this brief, a CED method for OLS codes encoders and syndrome calculation was proposed. The proposed method took advantage of the property of OLS codes to design a parity prediction scheme that could be professionally implement and protect the encoder. The method was evaluated for different word sizes, which showed that for large words the overhead is small. This is attractive as large word sizes are used, for example, in caches for which OLS codes have been recently future. Hence the

encoder is protected from errors. The Memory Protection is carried out by using the parity check matrix . It also can be connected with a syndrome computation to get correction purpose. Although the time taken for the process is somewhat noticeable a bit high but the access time is less compensating it. Therefore correct values can be sent to the decoder.

REFERENCES

[1] C. L. Chen and M. Y. Hsiao, "Error-correcting codes for semiconductor memory applications: A state-of-the-art review," IBM J. Res. Develop., vol. 28, no. 2, pp. 124–134, Mar. 1984.
[2] E. Fujiwara, Code Design for Dependable Systems: Theory and Practical Application. New York: Wiley, 2006.
[3] A. Dutta and N. A. Touba, "Multiple bit upset tolerant memory using a selective cycle avoidance based SEC-DED-DAEC code," in Proc. IEEE VLSI Test Symp., May 2007, pp. 349–354.
[4] H.M. Shao, T.K. Truong, L.J. Deutsch, J. Yuen and LS. Reed, "A VLSI Design of a pipeline Reed-Solomon Decoder", IEEE Trans. Comput., vol. C-34, no. 5, pp 393-403, May 1985. [5] Wicker, Stephen B., Bhargava, Vijay K, "Reed-Solomon Codes and the Compact Disc", IEEE Press ISBN 978-0-7803-1025-4.
[6] J.L. Massey, "Deep Space Communications and Coding: A Match Made in Heaven," in Advanced Methods for Satellite and Deep Space Communications, Lecture Notes in Control and Information Sciences, Volume 182, Berlin: Springer-Verlag, 1992.
[7] Sanjeev Kumar, Rajni Gupta, "Bit Error Analysis of Reed-Solomon Code for Efficient Communication System," International Journal of computer Applications (0975 – 8887) Volume 30- No. 12, September 2011.
[8] V.K Agrawal, Pankaj Goel, Gaurav Mittal, "Review of Reed-Solomon Code for Error Detection and correction," International Journal of Research in IT, Management and Engineering IJRIME Volume2, Issue6 (june-2012) ISSN: 2249-1619.
[9] Sung-woo Choi, sang-Sung Choi, Han-ho Lee, "RS decoder architecture for UWB", wireless home Network Research Team, ETRI ICACT, Feb 2006



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY