International Journal of Research
Available at https://edupediapublications.org/journals
p-ISSN: 2348-6848
e-ISSN: 2348-795X
Volume 03 Issue 13
September 2016

Improving The Life Time Reliability For Multilevel Phase Change Memory Using Non-Binary Orthogonal Latin Square Codes

LANKA PANDIT VENKATA SAI TEJASWEE[1], M.V.SREENIVASULU REDDY[2]
[1] PG Scholar, Electronics and Communication Engineering, CV Raman Institute of Technology, AP, India
[2]Assistant Professor, Electronics and Communication Engineering, CV Raman Institute of Technology, AP, India

Abstract—Frequently Error correction codes (ECCs) are used to protect memories over errors. Among ECCs, orthogonal Latin squares (OLS) codes used for memory protection due to their simplicity of the decoding algorithm that enables low delay implementations. An important issue is that when ECCs are used, the encoder and decoder circuits can also suffer errors. In this brief, a concurrent error detection technique for OLS codes encoders and syndrome computation is proposed and evaluated. The proposed method uses the properties of OLS codes to efficiently implement a parity prediction scheme that detects all errors that affect a single circuit node.
Index Terms—Concurrent error detection, error correction codes (ECC), Latin squares, majority logic decoding (MLD), memory.

I. INTRODUCTION

For many years to protect the memories by using error correction codes (ECCS)[1], [2]. There is a wide range of codes that are used or have been proposed for memory applications. Single error correction(SEC) codes that can correct one bit per word are commonly used. More advanced codes that can also correct double adjacent errors [3] or double errors in general have also been studied [4]. The use of more complex codes that can correct more errors is limited by their impact on delay and power, which can limit their applicability to memory designs [5]. To overcome those issues, the use of codes that are one step majority logic decodable (OS-MLD) has recently been proposed. OS-MLD codes can be decoded with low latency and are , therefore, used to protect memories [6]. Among the codes that are OS-MLD, a type of Euclidean geometry (EG) code has been proposed to protect memories [7], [8]. The use of difference set code has also been recently analyzed in [9]. Another type of code that is OS-MLD is orthogonal Latin squares (OLS) codes The use of OLS codes has gained renewed interest for inter connections[11], memories [12], and caches [13]. This is due to their modularity such that the error correction capabilities can be easily adapted to the error rate [11] or to the mode of operation [13].OLS codes typically require more parity bits than other codes to correct the same number of errors. However, their modularity and the simple and low delay decoding implementation (as OLS codes are OS-MLD), offset this disadvantage in many applications .An important issue is that the encoder and decoder circuits needed to use (ECCs) can also suffer errors. When an error affects the encoder, an incorrect word may be written into the memory. An error in the decoder can cause a correct word to be interpreted as erroneous or the other way around, an incorrect word to be interpreted as a correct word. The protection of the encoders and decoders has been studied for different ECCs. For example, in [8] EG codes were studied. The protection of Reed–Solomon, Hamming, and BCH encoders and decoders has also been studied in [14] and [15], and more general techniques for systematic and cyclic codes have been proposed in[16] and [17]. Finally, the protection of encoders for SEC codes against soft errors was discussed in [18].The ECC encoder computes the parity bits, and in most cases the decoder starts by checking the parity bits to detect errors. This is commonly referred as syndrome computation. For some codes, it possible to perform encoding and syndrome computation serially based on the properties of the code. However, when delay has to be low, parallel implementations are preferred. This is the case for OLS codes that are commonly used in high-speed applications. The reader is referred to [6] for a detailed discussion of ECC encoders and decoders. After syndrome computation, when errors are detected, the rest of the decoding is done to correct the errors. This means that generating and checking the parity bits are important parts of the encoder and decoder circuitry. Therefore, its protection is an important issue. In this brief, the protection of the encoders and syndrome computation for OLS codes when used in SRAM memories and caches is considered. Based on the specific properties of these codes, it is shown that parity prediction is an effective technique to detect errors in the encoder and syndrome computation. This is not the case for most other block codes for which parity prediction cannot provide effective protection. Therefore, this is another advantage of OLS codes in addition to their modularity and simple decoding.

A PCM relies on the reversible phase transformation of the chalcogenide alloy (e.g. Ge2Sb2Te5,bGST) between the amorphous and the crystalline states. The amorphous state has a high resistance and is commonly referred to as the reset state; the crystalline phase has a low resistance and is referred as the set state. If the PCM is in the Reset state (amorphous) and the voltage across the PCM cell is

International Journal of Research
Available at https://edupediapublications.org/journals
p-ISSN: 2348-6848
e-ISSN: 2348-795X
Volume 03 Issue 13
September 2016

higher than the threshold value, then a snapback behavior occurs and the resistance of the PCM is switched to the ON state value. If the PCM is in the ON state, it will switch back to the OFF state if and only if the voltage across the PCM is less than the so-called ON/OFF Intersection Point. A PCM cell can be used as a multilevel memory to increase capacity; this is made possible by its high resistance range, i.e. the difference between the resistances of the SET and RESET states. However, after a PCM cell is programmed, its resistance increases with time; this phenomenon is generally known to as the resistance drift. The resistance drift is believed to be the result of structural relaxation (SR) phenomena that are thermally activated as an atomic rearrangement of the amorphous structure [10], [13]. difference in resistance drift overtime, leading to a significant degradation in data integrity [13].. The resistance at each level varies according to a Gaussian distribution and accounts for less (more) drift when the PCM is in the (amorphous) crystalline phase. The resistance of each level changes during T (Fig. 1) and it could pass the threshold value (as separating two adjacent levels). This results in an erroneous output following a read. The erroneous effects of the resistance drift in a PCM cell can be alleviated if the threshold resistances could also vary with time. In [14], a time-aware fault-tolerant scheme is used for correcting the resistance drift of a PCM. The drift behavior of the threshold resistance is taken into account by keeping the so called lifetime of the PCM in the form of time tag bits and using them to find the threshold resistances.

is The rest of this brief is organized as follows. Section II introduces OLS codes and summarizes some of their properties that are used in the rest of this brief. In Section III, the proposed parity prediction scheme is described. Section IV evaluates its cost in terms of circuit area and delay. Finally, the conclusion from this brief is presented in Section V.

II. ORTHOGONAL LATIN SQUARES CODES

OLS codes are based on the concept of Latin squares. A Latin

square of size m is an $m \times m$ matrix that has permutations of the digits 0, 1,…, $m - 1$ in both its rows and columns [19]. Two Latin squares are orthogonal if when they are superimposed every ordered pair of elements appears only once. OLS codes are derived from OLS [10]. These codes have $k = m2$ data bits and $2tm$ check bits, where t is the number of errors that the code can correct. For a double error correction code $t = 2$, and, therefore, $4m$ check bits, are used. As mentioned in the introduction, one advantage of OLS codes is that their construction is modular. This means that to obtain a code that can correct t +1 errors, simply $2m$ check bits are added to the code that can correct t errors. This can be useful to implement adaptive error correction schemes, as discussed in [11] and [13]. The modular property also enables the selection of the error correction capability for a given word size.

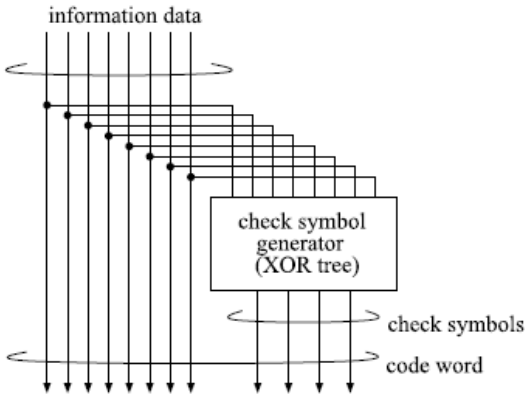


Fig 1 Proposed encoder design.

As mentioned before, OLS codes can be decoded using OS-MLD as each data bit participates in exactly $2t$ check bits and each other bit participates in at most one of those check bits. This enables a simple correction when the number 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 is in error and must be corrected. Otherwise the bit is correct. As long as the number 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 correction of an erroneous bit. In any case, the decoding starts by recomputing the parity check bits and checking against the stored parity check bits.

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

International Journal of Research
Available at https://edupediapublications.org/journals
p-ISSN: 2348-6848
e-ISSN: 2348-795X
Volume 03 Issue 13
September 2016

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 ^ c2 ^ c3 ^ • • • ^c2tm = 0. (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

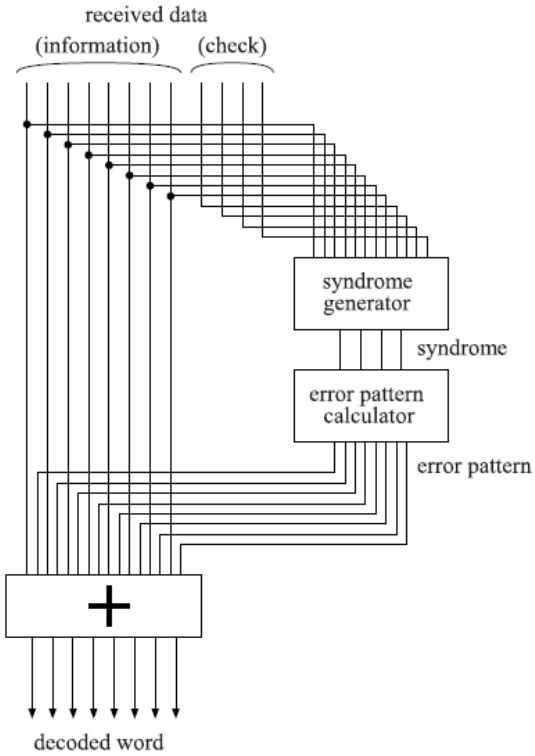


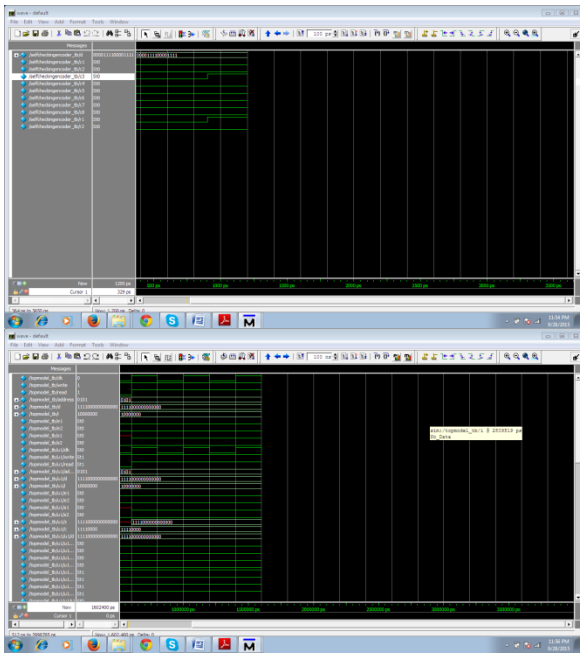
Fig 2: Proposed self-checking encoder for OLS code with k = 16 and t = 1.

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

The simulation results for the encoder and syndrome computation for OLS code with k=9 and t=1 is shown below



V. CONCLUSION

The concurrent error detection technique using the properties of OLS codes to design a parity prediction scheme is efficiently implemented and detected and corrected all errors that the single circuit node is affected. Different word sizes are evaluated using this technique. For large words the overhead is small. The availability of low overhead for encoder and syndrome computation is the reason for OLS codes in high speed memories and caches.

REFERENCES

[1] C. L. Chen and M. Y. Hsiao, "Error-correcting codes for semiconductormemory 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] R. Naseer and J. Draper, "DEC ECC design to improve memory reliability in sub-100nm technologies," in *Proc. IEEE Int. Conf. Electron., Circuits, Syst.*, Sep. 2008, pp. 586–589.

[5] G. C. Cardarilli, M. Ottavi, S. Pontarelli, M. Re, and A. Salsano, "Fault tolerant solid state mass memory for space applications," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 41, no. 4, pp. 1353–1372, Oct. 2005.

[6] S. Lin and D. J. Costello, *Error Control Coding*, 2nd ed. Englewood Cliffs, NJ: Prentice-Hall, 2004.

[7] S. Ghosh and P. D. Lincoln, "Dynamic low-density parity check codes for fault-tolerant nano-scale memory," in *Proc. Found. Nanosci.*, 2007, pp. 1–5.

[8] H. Naeimi and A. DeHon, "Fault secure encoder and decoder for nanoMemory applications," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 17, no. 4, pp. 473–486, Apr. 2009.

[9] S. Liu, P. Reviriego, and J. A. Maestro, "Efficient majority logic fault detection with difference-set codes for memory applications," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 20, no. 1, pp. 148–156, Jan. 2012.