ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com

International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE)
Vol. 3, Issue 6, June 2017

An efficient unequal error correction code using hamming code and interleaved orthogonal latin square code

Ms.J Divya Sangeetha [1], Mr.S Prabahar [2], Mrs.A Samsu Nighar [3]

Student, M.EApplied Electronics, Chandy college of engineering,
Tuticorin,Tamilnadu,India[1]
HOD/ECE, Chandy college of engineering,
Tuticorin,Tamilnadu,India [2]
AP/ECE, Chandy college of engineering,
Tuticorin,Tamilnadu,India [3]

Abstract— There has been a growing interest in multi-bit Error Correction Codes (ECCs) to protect SRAM memories. This has been caused by the increased number of multiple errors that memories suffer as technology scales. Triple adjacent errors are also common error that affects memory.To be suitable to protect an SRAM memory, an ECC has to be decodable in parallel and with low latency. Among the codes proposed for memory protection are Orthogonal Latin Square (OLS) codes that provide low latency decoding and a modular construction. For some applications, like multimedia or signal processing, the effect of errors on the memory bits can be very different depending on their position on the word. Therefore, in these cases, it is more effective to provide different degrees of error correction for the different bits. This is done with Unequal Error Protection (UEP) codes that corrects single, double and triple adjacent errors. Hamming code is used in single error correction to lower delay. The results show that the new codes can be implemented with lower decoding delay than traditional SEC-DED codes and TAEC codes. The Proposed encoder and decoder are done by Verilog and simulated in Modelsim..

Keywords — Multiple Cell Upsets (MCUs), Error Correction Codes, Orthogonal Latin Square Codes, SRAM memory.

I. INTRODUCTION

Orthogonal Latin Square (OLS) codes were introduced more than four decades ago to protect memories . They are attractive because of their modular construction and low decoding delay. In recent years, there has been a renewed interest in OLS codes to protect memories and interconnections.This interest is driven by the larger percentage of errors that affect multiple bits as technology scales [4]. For example, in [5] radiation tests were reported for 65nm and 45nm memories showing MCUs that affect more than two bits. Triple adjacent errors were the most common ones among those. To deal with those errors, several enhancements have been recently proposed to OLS codes to provide additional adjacent error correction capabilities. An initial work presented in [2] showed how Triple Adjacent Error Correction can be achieved for Double Error Correction (DEC) OLS codes using some additional parity check bits. Then in [6], an alternative scheme that also provides triple adjacent error correction without adding additional parity check bits was proposed. In this scheme, there are two correction circuits that operate in parallel: the standard OLS decoder and a triple adjacent error decoder. Errors are first classified and then the appropriate circuit is used to correct the error. The use of independent circuits for each error pattern increases the decoder complexity. For some applications, like multimedia or signal processing, the effect of errors on the memory bits can be very different depending on their position on the word.Hence unequal error protection is needed.

In this paper, a new method to implement TAEC on DEC OLS codes is presented with unequal error protection.it consists of two blocks namely SEC and DEC-TAEC block. The proposed scheme can be used for DEC OLS codes that protect data bit words of up to 64 bits. The main benefit of the new method is that a single decoder can be used for all error patterns thus reducing complexity. The new decoder is obtained by making some minor modifications to the traditional DEC OLS decoder and therefore complexity is similar to that of the traditional decoder. This is achieved by using a modified placement of data and parity check bits which ensures that triple adjacent bit errors always affect at least one parity check bit.

The rest of the paper is organized as follows. Section II introduces OLS codes and the previous method to correct triple adjacent errors. The proposed method is introduced in

Section III and evaluated in Section IV. The paper ends with some conclusions in Section V.

II. TEAC IN DEC ORTHOGONAL LATIN SQUARE CODES

OLS codes are based on the concept of Latin Squares that have many other applications . A Latin square of size *m* is an *m * m* matrix that has permutations of the numbers *0,1,..m-1* in its rows and columns. When two Latin squares are superimposed and every ordered pair of elements appears only once, the Latin Squares are orthogonal. OLS codes are obtained from orthogonal Latin squares and have $k=m^2$ data bits for Latin squares of size *m*. In general, a code that can correct *t* errors has *2tm* parity check bits. For a Double Error Correction (DEC) code *t=2* and therefore *4m* check bits are used. The number of errors that can be corrected by an OLS code can be increased by using more orthogonal Latin squares thus adding more parity check bits. This modular construction is one of the advantages of OLS codes. The other major advantage of OLS codes is that they can be decoded with low delay.

When two Latin squares are superimposed and every ordered pair of elements appears only once, the Latin Squares are orthogonal. OLS codes are obtained from orthogonal Latin squares and have $k=m^2$ data bits for Latin squares of size. A method was proposed in [6] to implement Triple Adjacent Error Correction (TAEC) in DEC OLS codes. The idea is to use the fact that a TAE will always cause three parity check bits to be in error on the M_2 group while a double error can affect at most two parity check bits on a group. This is used to distinguish between TAE and single/double errors. Then single/double errors are corrected using the standard DEC OLS decoder and TAE errors are corrected using the syndrome value of the bits in groups M_1 and M_2.

The decoder proposed in [6] is shown on Figure 2. The evaluation results showed that the decoder can effectively correct TAEs. However, the overhead in the decoder complexity was significant. In the next section, a new method that can also correct TAEs with lower cost is presented.

III. EFFICIENT UNEQUAL TAEC-DEC AND SEC METHOD

The reduced complexity triple adjacent error correction is achieved by using an optimized bit placement that interleaves data and parity check bits.

The encoder for SEC block is shown in fig 2.it is constructed using modified hamming code.

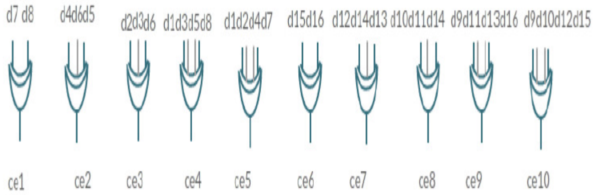

Fig 2:SEC encoder

The encoder for TAEC-DEC block is shown below for 8bit error correction.

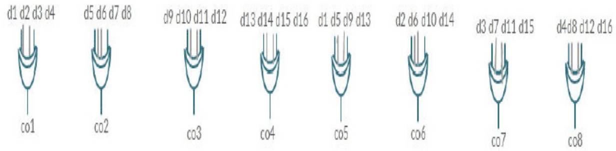

Fig3:TAEC-DEC encoder

In more detail for TAEC, a parity check bit is placed between every two data bits (d_i and p_i are the data bit and parity check bit respectively). The method used here is that the

parity check bits are selected such that the adjacent data bits participate in the parity check bit that is in between. This means that for a TAE, at most the parity check bits of the two data bits affected are in error and therefore there can be no miscorrection.

Let us for example consider a TAE that affects the first three bits (d_4,d_2,p_1), then the parity check bits in error will be 1,6,8,10,12,14 and 16 which correspond to those in which the first two data bits participate. In this example, the TAE is therefore corrected using the standard DEC-OLS decoder. Let us consider now an error that affects bits (d_3,d_1,p_5). In that case, the parity check bits in error will be 7,9,11,13 and 15. Therefore, the d_3 bit will be corrected because the majority (7, 11 and 15) of the parity check bits that it participates in will be

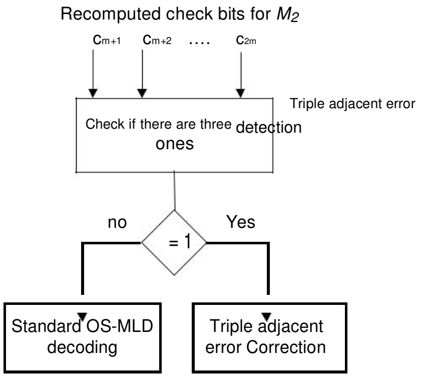
Recomputed check bits for M_2
C_{m+1}
C_{m+2}
....
C_{2m}
Triple adjacent error detection
Check if there are three ones
no
Yes
= 1
Standard OS-MLD decoding
Triple adjacent error Correction

Fig. 1: Block diagram of the modified decoding algorithm (from [6]).

However, for the d_1 bit only two of the parity check bits (9 and 13) will be in error. Therefore, the error on the d_1 bit will not be corrected using the standard DEC OLS decoder.

The parity check matrix of a DEC OLS code is formed by four groups of check bits, each corresponding to one Latin square (M_1,M_2,M_3,M_4). It can be seen that each data bit participates in exactly one parity check bit in each group and that any two bits have at most one parity check bit in common. This enables a simple correction. The four check bits for each data bit 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. Error correction is ensured as long as the number of errors is two or less, as the remaining error can, in the worst case, affect one check bit of the first data bit so that still a majority of three triggers the correction of an erroneous bit. This process is known as One Step Majority Logic Decoding (OS-MLD) and provides low complexity decoding.

OLS codes are based on the concept of Latin Squares that have many other applications [7]. A Latin square of size *m* is an $m * m$ matrix that has permutations of the numbers *0,1,..m-1* in its rows and columns. When two Latin squares are superimposed and every ordered pair of elements appears only once, the Latin Squares are orthogonal. OLS codes are obtained from orthogonal Latin squares and have $k=m^2$ data bits for Latin squares of size *m*. In general, a code that can correct *t* errors has *2tm* parity check bits. For a Double Error Correction (DEC) code *t=2* and therefore *4m* check bits are used. The number of errors that can be corrected by an OLS code can be increased by using more orthogonal Latin squares thus adding more parity check bits. This modular construction is one of the advantages of OLS codes. The other major advantage of OLS codes is that they can be decoded with low delay. In fact, they are one of the few code types that can be decoded using a technique called One Step Majority Logic Decoding (OS-MLD) [8].

The parity check matrix of a DEC OLS code is formed by four groups of check bits, each corresponding to one Latin square (M_1,M_2,M_3,M_4). As an example, the matrix for a (32,16) DEC OLS code is shown on Figure 1.

It can be seen that each data bit participates in exactly one parity check bit in each group and that any two bits have at most one parity check bit in common. This enables a simple correction.

The four check bits for each data bit 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.

Error correction is ensured as long as the number of errors is two or less, as the remaining error can, in the worst case, affect one check bit of the first data bit so that still a majority of three triggers the correction of an erroneous bit.

This process is known as One Step Majority Logic Decoding (OS-MLD) and provides low complexity decoding.

Compared to the TAEC decoder in [6], it can be observed that in all cases, the proposed TAEC decoder provides better area, power and delay. Similar results were obtained when DC was configured for maximum effort to reduce power consumption.

In summary, the evaluation shows that the proposed method has a significant advantage compared to the existing TAEC decoder presented in [6].

This simple modification ensures that all TAEs can be corrected when using the proposed method. In addition, the proposed modification cannot cause miscorrection for single and double error because a majority vote on bit d_3 can only occur if that bit is in error. A final detail is the placement of the last parity check bits. In Figure 3, the last two data bits are such that one of them participates in the next parity check bit and thus a similar decoding can be used. The rest of the parity check bits are placed in sequential order so that a TAE can affect at most two of the M_i groups so that again no miscorrection can occur. In a general case, a Hamming code will have rows with a number of ones that is roughly *k/2*. This compares with the proposed SEC codes ($w = 2$) for which the number of ones in a row is by design at most $n - k - 1$. Similarly, to locate an error a traditional SEC code requires a $n - k$ input AND gate compared with a simple two input ANDgate in the proposed code.

The parity check matrix and decoder for SEC code is shown below:

$$H = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Fig 4:SEC reduced parity check matrix

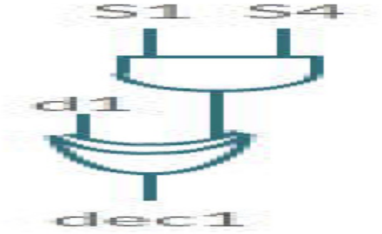

Fig 5:SEC decoder

The proposed method can be used as long as there are sufficient parity check bits to interleave among the data bits. This is the case for DEC OLS codes of up to 64 bits that can be used to protect 16 and 64 bit data words.

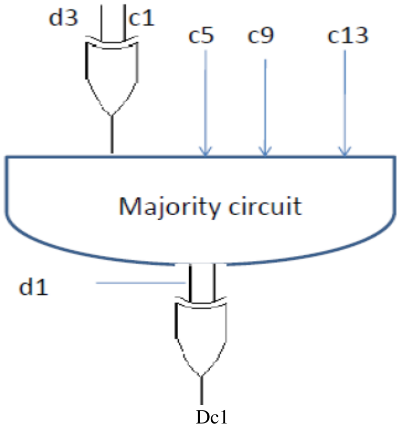

Fig. 5: Modified decoding for the d_1 bit in the (32,16) TAEC-DEC code
.

IV. Evaluation

The new triple adjacent error correction scheme has been implemented and compared with both the previous method in [6] and a standard DEC OLS decoder. The first part of the evaluation has focused on testing that the proposed decoder can correct all single, double and triple adjacent errors. To that end, all possible combinations of single, double and triple adjacent errors have been exhaustively generated and tested. In all cases, the data bits after decoding are identical to the original data bits.

The second part of the evaluation has focused on evaluating the decoder complexity. To that end, the proposed decoder, the decoder presented in [6] and a standard OLS decoder have been implemented in HDL and mapped to a 65nm library using Synopsis DC. The area, delay and power estimates are shown in Tables I to III for a synthesis with maximum effort on minimizing circuit area and in Tables IV to VI for synthesis with maximum effort on minimizing circuit delay. Compared to a standard DEC OLS decoder, the proposed decoder has only a slightly larger area and power while in the case of the delay, there is a penalty. Compared to the TAEC decoder in [6], it can be observed that in all cases, the proposed TAEC decoder provides better area, power and delay. Similar results were obtained when DC was configured for maximum effort to reduce power consumption. In summary, the evaluation shows that the proposed method has a significant advantage compared to the existing TAEC decoder presented in [6].

Table I. Decoder area (μm^2) for an area optimized synthesis

K	OS-MLD	TAEC new	Overhead	TAEC in [6]	Overhead
16	775	826	6.6%	1084	39.9%
64	3284	3474	5.8%	4289	30.6%

Table II. Decoder delay (ns) for an area optimized synthesis

K	OS-MLD	TAEC new	Overhead	TAEC in [6]	Overhead
16	1.82	2.79	53.3%	4.51	147.8%
64	1.99	3.28	64.8%	5.51	176.9%

Table III. Decoder power (mW) for an area optimized synthesis

K	OS-MLD	TAEC new	Overhead	TAEC in [6]	Overhead
16	1.48	1.54	4.1%	1.88	27.0%
64	8.32	8.67	4.2%	9.73	16.9%

Table IV. Decoder area (μm^2) for a delay optimized synthesis

K	OS-MLD	TAEC new	Overhead	TAEC in [6]	Overhead
16	1738	1988	14.4%	5005	188.0%
64	8912	9537	7.0%	14518	62.9%

Table V. Decoder delay (ns) for a delay optimized synthesis

K	OS-MLD	TAEC new	Overhead	TAEC in [6]	Overhead
16	0.71	0.73	2.8%	0.88	23.9%
64	0.93	1.07	15.1%	1.22	31.2%

Table VI. Decoder power (mW) for a delay optimized synthesis

K	OS-MLD	TAEC new	Overhead	TAEC in [6]	Overhead

V.Conclusions

In this paper, a method to implement Triple Adjacent Error Correction (TAEC) on Double Error Correction (DEC) Orthogonal Latin Square (OLS) codes has been presented with unequal error protection code is presented. The SEC decoder constructed with modified hamming code lowers the delay.The method results in a significant reduction of the decoder complexity compared with the previous method to implement TAEC in DEC OLS codes. The benefits have been evaluated using a 65nm library and confirm that circuit area, power and delay are reduced significantly.

References

[1] M.Y. Hsiao, D.C. Bossen, and R.T. Chien, ''Orthogonal Latin Square Codes,'' *IBM J. Research and Development*, vol. 14, no. 4, 1970, pp. 390-394.

[2] R. Datta and N.A. Touba, "Generating Burst-Error Correcting Codes from Orthogonal Latin Square Codes -- A Graph Theoretic Approach" *IEEE International Symposium Defect and Fault Tolerance in VLSI (DFT)*, pp. 367 – 373, 2011.

[3] S. E. Lee, Y. S. Yang, G.S Choi, W. Wu and R. Iyer, "Low-Power, Resilient Interconnection with Orthogonal Latin Squares", *IEEE Design & Test of Computers*, vol.: 28, no. 2, pp. 30 – 39, 2011.

[4] E. Ibe, H. Taniguchi, Y. Yahagi, K. Shimbo and T. Toba, "Impact of scaling on neutron-induced soft error rate in SRAMs From a 250 nm to a 22 nm Design Rule", *IEEE Trans. on Electron Devices*, vol. 57 , no. 7, pp. 1527-1538, July 2010.

[5] S. Baeg, S. Wen and R. Wong, "SRAM Interleaving Distance Selection with a Soft Error Failure Model", *IEEE Trans. on Nuclear Science*, vol.56, no.4, pp.2111-2118, Aug. 2009.

[6] P. Reviriego, S. Liu, S. Lee, N.A. Touba, J.A. Maestro, R. Datta, "Implementing Triple Adjacent Error Correction in Double Error Correction Orthogonal Latin Squares Codes", *IEEE International Symposium Defect and Fault Tolerance in VLSI* , pp. 167-171, 2013.

[7] PedroRriviriego“A method to construct a low delay single error correction codes for protecting data bits only.

[8] S. Lin and D. J. Costello, “Error control coding” (2nd Ed.). Englewood Cliffs, NJ: Prentice-Hall. 2004.