International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2017 IJSRCSEIT | Volume 2 | Issue 6 | ISSN : 2456-3307

Orthogonal Latin Squares Encoders and Syndrome Computation with Concurrent Error Detection

P. Divya

Department of Electronics and Communication Engineering, Medha College of Engineering, Affiliated to Jawaharlal Nehru Technological University, Hyderabad, Telangana, India

ABSTRACT

Error Correction Codes (ECCS) Are Commonly Used To Protect Memories Against Errors. Among Eccs, Orthogonal Latin Squares (OLS) Codes Have Gained Renewed Interest For Memory Protection Due Tot heir Modularity And The 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.

Keyword : Concurrent Error Detection, Error Correction Codes (ECC), Latin Squares, Majority Logic Decoding (MLD), Memory, OLS Codes.

I. INTRODUCTION

Error Correction Codes (ECCS) Have Been Used To Protect Memories For Many Years [1], [2]. 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) Code [10].

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 To As Syndrome Computation. For Some Codes, It Is 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.

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.

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.

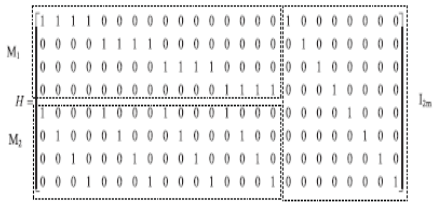

Figure 1. Parity check matrix for OLS code with $k = 16$ and $t = 1$.

The Parity Check Matrix H For Ols Codes Is Constructed From The Ols. As An Example, The Matrix For A Code With $K = 16$ And 8 Check Bits That Can Correct Single Errors Is Shown In Fig. 1. As Discussed Earlier, Due To The Modular Construction Of Ols Codes This Matrix Forms Part Of The H Matrix For Codes That Can Correct More Errors. For Example, To Obtain A Code That 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:

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

Where $I2m$ Is The Identity Matrix Of Size $2m$ And $M1$, $M2$ Are Matrices Of Size $M \times M2$. The Matrix $M1$ Has M Ones In Each Row. Fort He R Th Row, The Ones

Are At Positions $(R-1)\times M+1,(R-1)\times M+2,\ldots(R-1)\times M+M-1$, $(R-1)\times M+M$. The Matrix$m2$ Is Constructed As Follows: $M2=[Im\ Im\ \ldots\ Im]$. (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

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

In summary, the encoder takes $k=m2$ data bits $(di\ )$ and computes$2tm$ parity check bits $(ci\ )$ using a matrix G, which is derived from Latin squares and has the following properties.

1) Each data bit participates exactly in $2t$ parity checks.
2) A pair of data bits participates (both bits) in at most one of the parity checks.

These properties are used in the next section to discuss the proposed technique.

III. PROPOSED CONCURRENT ERROR DETECTION TECHNIQUE

A circuit is self-checking [20] if and only if it satisfies thefollowing properties: 1) it is self-testing, and 2) fault-secure.

A circuit is self-testing if, for each fault f in the fault set $\boldsymbol{F}$, thereis at least one input belonging to the input code space, for which thecircuit provides an output belonging to the output error space.

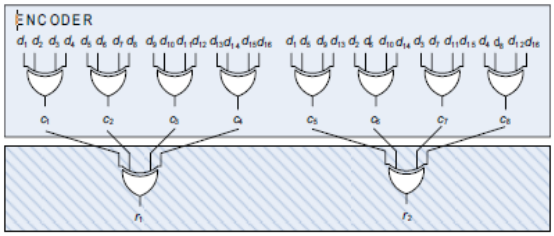

Figure 2. Proposed self-checking encoder for OLS code with $k=16$ and $t=1$.

A circuit is fault-secure if, for each fault f in the fault set $\boldsymbol{F}$ and for each input belonging to the input code space, the circuit provides the correct output, or an output belonging to the output error space.
The technique that we propose is based on the use of parity prediction, which is one of the techniques commonly used to detect error in general logic circuits [21], [22]. In our case, the problem is substantially simpler, given the structure of the OLS codes. For the encoder, it is proposed that the parity of the computed check bits $(ci\ )$ is compared against the parity of all the check equations .The parity of all the check equations is simply the equation obtained by computing the parity of the columns in G. For OLS codes, since ach column in G has exactly $2t$ ones, the null equation is obtained(see, for example, Fig. 1). Therefore, the concurrent error detection(CED) scheme is simply to check.

$$c1\oplus c2\oplus c3\oplus\cdots\oplus c2tm=0. \quad (4)$$

For the syndrome computation, the parity prediction can be implemented by checking that the following two equations take the same value

$$r1=s1\oplus s2\oplus s3\oplus\cdots\oplus s2tm \quad (5)$$
$$r2=c1\oplus c2\oplus c3\oplus\cdots\oplus c2tm \quad (6)$$

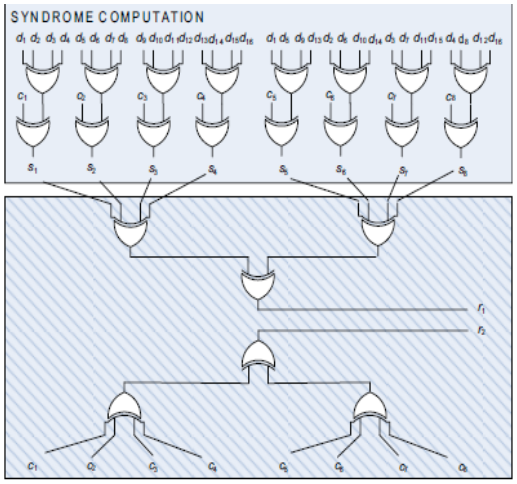

Figure 3. Proposed self-checking syndrome computation for OLS code with $k=16$ and $t=1$.

TABLE I
OVERHEAD OF PROPOSED CED FOR SEC-OLS CODES

k	m	Encoder	Syndrome
16	4	29.17%	43.75%
64	8	13.39%	23.44%
256	16	6.46%	12.11%

$$o_{\text{encoder}}=\frac{(2tm-1)}{(2tm(m-1))} \qquad (7)$$

$$o_{\text{syndrome}}=\frac{(4tm-2)}{(2tm^2)}. \qquad (8)$$

For Large Values Of M, The First Equation Can Be Approximated By$1/M$ And The Second By $2/M$. This Shows That As The Block Size Of The code $K = M2$ Grows, The Overhead Becomes Smaller And Independent of T . This Is Illustrated In Table I, Where The Overheads Predicted By(7) And (8) Are Illustrated For $T = 1$ And Different Values Of K.

TABLE II
LAYOUT AREA FOR ENCODER (in μm^2)

k	m	Unprotected	With CED	Overhead
16	4	184.6	236.2	27.9%
64	8	847.0	948.7	12.0%
256	16	3594.1	3820.1	6.2%

TABLE III
LAYOUT AREA FOR SYNDROME COMPUTATION (in μm^2)

k	m	Unprotected	With CED	Overhead
16	4	237.7	344.1	45.1%
64	8	951.1	1176.4	23.6%
256	16	3804.4	4276.8	12.4%

TABLE IV
DELAY ESTIMATES FOR ENCODER (in ns)

k	m	Unprotected	With CED	Overhead
16	4	1.26	1.65	30%
64	8	1.52	2.01	28%
256	16	1.71	2.41	40%

TABLE V
DELAY ESTIMATES FOR SYNDROME COMPUTATION (in ns)

k	m	Unprotected	With CED	Overhead
16	4	1.32	1.75	32%
64	8	1.54	2.05	33%
256	16	1.75	2.42	38%

IV. EVALUATION

The Proposed CED Mechanisms Have Been Implemented In VHDL For Codes With $T = 1$ And The Values Of K Used In Table I. The Designs Have Been Implemented For The 45-Nm Osu free PDK Standard Cell Library [25] Using Synopsys Design Compiler for The Synthesis And Cadence Encounter For Placement And routing.

The Area Overhead Of Avoiding Logic Sharing Was 35%. The Cost Of The checker Is Also Larger For Hamming. In This Particular Case, The Total overhead For The Proposed Scheme Was Over 80%. This Confirms That the Proposed Checker Is Not Effective In A General Case And Relies On The Properties Of OLS Codes To Achieve An Efficient Implementation.

V. CONCLUSION

In This Brief, A CED Technique For OLS Codes Encoders And Syndrome Computation Was Proposed. The Proposed Technique Took Advantage Of The Properties Of OLS Codes To Design A Parity Prediction Scheme That Could Be Efficiently Implemented And Detects All Errors That Affect A Single Circuit Node. The Technique Was Evaluated For Different Word Sizes, Which Showed That For Large Words The Overhead Is Small.

VI. 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 Practicalapplication. New York: Wiley, 2006.

[3]. A. Dutta And N. A. Touba, "Multiple Bit Upset Tolerant Memory Using Aselective Cycle Avoidance Based Sec-Ded-Daec Code," In Proc. Ieeevlsi Test Symp., May 2007, Pp. 349-354.

[4]. R. Naseer And J. Draper, "Dec Ecc Design To Improve Memory Reliabilityin 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, "Faulttolerant 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. Englewoodcliffs, Nj: Prentice-Hall, 2004.

[7]. S. Ghosh And P. D. Lincoln, "Dynamic Low-Density Parity Check Codesfor Fault-Tolerant Nano-Scale Memory," In Proc. Found. Nanosci., 2007,Pp. 1-5.

[8]. H. Naeimi And A. Dehon, "Fault Secure Encoder And Decoder Fornanomemory 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 Logicfault 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.

[10]. M. Y. Hsiao, D. C. Bossen, And R. T. Chien, "Orthogonal Latinsquare Codes," Ibm J. Res. Develop., Vol. 14, No. 4, Pp. 390-394,Jul. 1970.

[11]. S. E. Lee, Y. S. Yang, G. S. Choi, W. Wu, And R. Iyer, "Low-Power,Resilient Interconnection With Orthogonal Latin Squares," Ieee Designtest Comput., Vol. 28, No. 2, Pp. 30-39, Mar.-Apr. 2011.

[12]. R. Datta And N. A. Touba, "Generating Burst-Error Correcting Codes Fromorthogonal Latin Square Codes-A Graph Theoretic Approach," In Proc. Ieeeint. Symp. Defect Fault Tolerance Vlsi Nanotechnol. Syst., Oct. 2011, Pp. 367-373.

[13]. A. R. Alameldeen, Z. Chishti, C. Wilkerson, W. Wu, And S.-L. Lu,"Adaptive Cache Design To Enable Reliable Low-Voltage Operation," Ieee Trans. Comput., Vol. 60, No. 1, Pp. 50-63, Jan. 2011.

[14]. G. C. Cardarilli, S. Pontarelli, M. Re, And A. Salsano, "Concurrent Error Detection In Reed-Solomon Encoders And Decoders," Ieee Trans.Very Large Scale Integr. (Vlsi) Syst., Vol. 15, No. 7, Pp. 842-846,Jul. 2007.

[15]. I. M. Boyarinov, "Self-Checking Circuits And Decoding Algorithmsfor Binary Hamming And Bch Codes And Reed-Solomon Codes Overgf(2m)," Prob. Inf. Transmiss., Vol. 44, No. 2, Pp. 99-111, 2008.

[16]. H. Jaber, F. Monteiro, S. J. Piestrak, And A. Dandache, "Designof Parallel Fault-Secure Encoders For Systematic Cyclic Block Transmissioncodes," Microelectron. J., Vol. 40, No. 12, Pp. 1686-1697,Dec. 2009.

[17]. S. J. Piestrak, A. Dandache, And F. Monteiro, "Designing Fault-Secureparallel Encoders For Systematic Linear Error Correcting Codes," Ieee Trans. Reliab., Vol. 52, No. 4, Pp. 492-500, Apr. 2003.

[18]. J. A. Maestro, P. Reviriego, C. Argyrides, And D. K. Pradhan, "Faulttolerant Single Error Correction Encoders," J. Electron. Test., Theoryappl., Vol. 27, No. 2, Pp. 215-218, Apr. 2011.

[19]. J. Denes And A. D. Keedwell, Latin Squares And Their Applications.San Francisco, Ca: Academic, 1974.

[20]. P. K. Lala, Self-Checking And Fault-Tolerant Digital Design. Sanmateo,Ca: Morgan Kaufmann, 2001.