ISSN (Online) 2394-6849

International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE)
Vol 5, Issue 2, February 2018

A Fault-Tolerant Memory System for Nano Memory Applications

[1] Pradeep Kalakoti, [2] Kalakoti. Kalyan
[1] [2] Assistant Professor
[1] Balaji Institute of Technology and Science, Warangal, TELANGANA,
[2] SVS Institute of Technology, Hanamkonda, TELANGANA

Abstract: **- Due to the increase in the soft error rate in logic circuits, the encoder and decoder circuitry around the memory blocks have become susceptible to soft errors as well and must also be protected. We introduce a new approach to design fault-secure encoder and decoder circuitry for memory designs. Hamming codes are often used in today's memory systems to correct single error and detect double errors in any memory word. In these memory architectures, only errors in the memory words are tolerated and there is no preparation to tolerate errors in the supporting logic (i.e. encoder and corrector). However combinational logic has already started showing susceptibility to soft errors, and therefore the encoder and decoder (corrector) units will no longer be immune from the transient faults. Therefore, protecting the memory system support logic implementation is more important. Here we proposed a fault tolerant memory system that tolerates multiple errors in each memory word as well as multiple errors in the encoder and corrector units. We illustrate using Euclidean Geometry codes and Projective Geometry codes to design the fault-tolerant memory system, due to their well-suited characteristics for this application.**

Keywords: **- Soft error rate, Fault-secure, Euclidean Geometry Code.**

I. INTRODUCTION

Electronic space provided by silicon chips (semiconductor memory chips) or magnetic/optical media as temporary or permanent storage for data and/or instructions to control a computer or execute one or more programs. Two main types of computer memory are: (1) Read only memory (ROM), smaller part of a computer's silicon (solid state) memory that is fixed in size and permanently stores manufacturer's instructions to run the computer when it is switched on. (2) Random access memory (RAM), larger part of a computer's memory comprising of hard disk, CD, DVD, floppies etc., (together called secondary storage) and employed in running programs and in archiving of data. Memory chips provide access to stored data or instructions that is hundreds of times faster than that provided by secondary storage. Particularly, we identify a class of error-correcting codes (ECCs) that guarantees the existence of a simple fault-tolerant detector design. This class satisfies a new, restricted definition for ECCs which guarantees that the ECC codeword has an appropriate redundancy structure such that it can detect multiple errors occurring in both the stored codeword in memory and the surrounding circuitries. We call this type of error-correcting codes, fault-secure detector capable ECCs (FSD-ECC). The parity-check Matrix of an FSD-ECC has a particular structure that the decoder circuit, generated from the parity-check Matrix, is Fault-Secure. The ECCs we identify in this class are close to optimal in rate and distance, suggesting we can achieve this property without sacrificing traditional ECC metrics. We use the fault-secure detection unit to design a fault-tolerant encoder and corrector by monitoring their outputs. If a detector detects an error in either of these units, that unit must repeat the operation to generate the correct output vector. Using this retry technique, we can correct potential transient errors in the encoder and corrector outputs and provide a fully fault-tolerant memory system.

SYSTEM OPERATION:

LOW-density parity-check (LDPC) codes were first discovered by Gallager in the early 1960s [2] and have recently been rediscovered and generalized .It has been shown that these codes achieve a remarkable performance with iterative decoding that is very close to the Shannon limit[3]. Consequently, these codes have become strong competitors to turbo codes for error control in many communication and digital storage systems where high reliability is required.

LDPC codes can be constructed using random or deterministic approaches. In this report, we focus on a class of LDPC codes known as Euclidean Geometric (EG) LDPC codes, which are constructed deterministically using the points and lines of a Euclidean geometry [1,16]. The EG

LDPC codes that we consider are cyclic and consequently their encoding can be efficiently implemented with linear shift registers. Minimum distances for EG codes are also reasonably good and can be derived analytically. Iteratively decoded EG LDPC codes also seem to not have the serious error- floors that plague randomly-constructed LDPC codes; this fact can be explained by the observation made in [6] that EG LDPC codes do not have pseudo-code words of weight smaller than their minimum distance. For these reasons, EG LDPC codes are good candidates for use in applications like optical communications that require very fast encoders and decoders and very low bit error-rates.

EUCLIDEAN GEOMETRY CODE REVIEW

The construction of Euclidean Geometry codes based on the lines and points of the corresponding finite geometries. Euclidean Geometry codes are also called EG-LDPC codes based on the fact that they are low-density parity-check (LDPC) codes [14]. LDPC codes have a limited number of 1's in each row and column of the matrix; this limit guarantees limited complexity in their associated detectors and correctors making them fast and light weight [9].

Let EG be a Euclidean Geometry with n points and J lines. EG is a finite geometry that is shown to have the following fundamental structural properties:

1) Every line consists of ρ points;
2) Any two points are connected by exactly one line;
3) Every point is intersected by γ lines;
4) Two lines intersect in exactly one point or they are parallel; i.e., they do not intersect.

Let H be a $J \times n$ binary matrix, whose rows and columns corresponds to lines and points in an EG Euclidean geometry, respectively, where $h_{i,j} = 1$ if and only if the ith line of EG contains the jth point of EG, and $h_{i,j} = 0$ otherwise.

A row in H displays the points on a specific line of EG and have weight ρ .A column in H displays the lines that intersect at a specific point in EG and have weight γ . The rows of H are called the incidence vectors of the lines in EG, and the columns of H are called the intersecting vectors of the points in EG. Therefore, H is the incidence matrix of the lines in EG over the points in EG. It is shown in [15] that H is a LDPC matrix, and therefore the code is an LDPC code.
A special subclass of EG-LDPC codes, type-I 2-D EG-LDPC, is considered here. It is shown in [15] that type-I 2-D EG-LDPC has the following parameters for any positive integer $t \geq 2$:

- Information bits, $k = 2^{2t} - 3^t$;
- Length, $n = 2^{2t} - 1$;
- Minimum distance, $d_{\min} = 2^t + 1$;
- Dimensions of the parity-check matrix, $n \times n$;
- Row weight of the parity-check matrix, $\rho = 2^t$;
- Column weight of the parity-check matrix, $\gamma = 2^t$.

It is important to note that the rows of H are not necessarily linearly independent, and therefore the number of rows do not necessarily represents the rank of the H matrix. The rank of H is $n-k$ which makes the code of this matrix linear code. Since the matrix is $n \times n$, the implementation has n syndrome bits instead of $n-k$. The $(2^{2t}-1)\times(2^{2t}-1)$, parity-check matrix H of an EG Euclidean geometry can be formed by taking the incidence vector of a line in EG and its $2^{2t}-2$ cyclic shifts as rows; therefore this code is a cyclic code.

EFFICIENCY OF EG-LDPC

It is important to compare the rate of the EG-LDPC code with other codes to understand if the interesting properties of low-density and FSD-ECC come at the expense of lower code rates. We compare the code rates of the EG-LDPC codes that we use here with an achievable code rate upper bound (Gilbert- Varshamov bound) and a lower bound (Hamming bound). The EG-LDPC codes are no larger than the achievable Gilbert bound for the same k and d value and they are not much larger than the Hamming bounds. Consequently, we see that we achieve the FSD property without sacrificing code compactness.

ENCODER

An -bit codeword c, which encodes a k-bit information vector i is generated by multiplying the k-bit information vector with a $k \times n$ bit generator matrix G ; i.e., $c = i.G$. EG-LDPC codes are not systematic and the information bits must be decoded from the encoded vector, which is not desirable for our fault-tolerant approach due to the further complication and delay that it adds to the operation. However, these codes are cyclic codes [1]. We used the procedure presented in [1] and [4] to convert the cyclic generator matrices to systematic generator matrices for all the EG-LDPC codes under consideration.

FAULT SECURE DETECTOR

The core of the detector operation is to generate the syndrome vector, which is basically implementing the following vector-matrix multiplication on the received encoded vector c and parity-check matrix H

$$s = c.H^T \text{ ……………………. (1).}$$

Therefore each bit of the syndrome vector is the product of c with one row of the parity-check matrix. This product is a linear binary sum over digits of c where the corresponding digit in the matrix row is 1. This binary sum is implemented with an XOR gate. Fig 5 shows the detector circuit for the (15, 7, 5) EG-LDPC code. Since the row weight of the parity-check matrix is ρ, to generate one digit of the syndrome vector we need a ρ -input XOR gate, or $(\rho-1)$ 2-input XOR gates. For the whole detector, it takes $n(\rho-1)$ 2-input XOR gates.

CORRECTOR

One-step majority-logic correction is a fast and relatively compact error-correcting technique [1]. There is a limited class of ECCs that are one-step-majority correctable which include type-I two-dimensional EG-LDPC. In this section, we present a brief review of this correcting technique. Then we show the one-step majority-logic corrector for EG-LDPC codes.

ONE-STEP MAJORITY-LOGIC CORRECTOR

One-step majority logic correction is the procedure that identifies the correct value of a each bit in the codeword directly from the received codeword; this is in contrast to the general message-passing error correction strategy (e.g., [5]) which may demand multiple iterations of error diagnosis and trial correction. Avoiding iteration makes the correction latency both small and deterministic. This method consists of two parts:

1) Generating a specific set of linear sums of the received vector bits and

2) Finding the majority value of the computed linear sums.

The majority value indicates the correctness of the code-bit under consideration; if the majority value is 1, the bit is inverted, otherwise it is kept unchanged. The theory behind the one-step majority corrector and the proof that EG-LDPC codes have this property are available in [1].

DESIGN FLOW CHART:

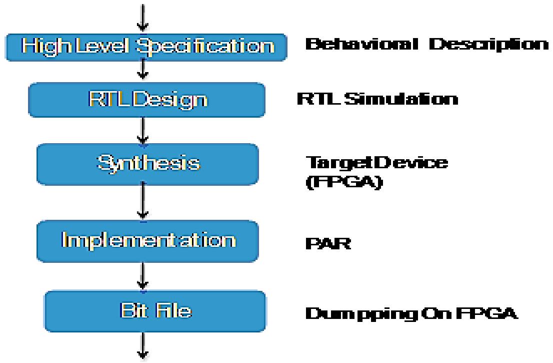

Fig: simulation waveform for the fault secure memory system.

II. CONCLUSION

We proved that these codes are part of a new subset of ECCs that have FSDs. Using these FSDs we design a fault-tolerant encoder and corrector, where the fault-secure detector monitors their operation. We also presented a unified approach to tolerate permanent defects and transient faults. This unified approach reduces the area overhead. Without this technique to tolerate errors in the ECC logic, we would required reliable (and consequently lithographic scale) encoders and decoders. Accounting for all the above area overhead factors, all the codes considered here achieve memory density of 20 to 100 Gb/nm, for large enough memory (0.1 Gb).

REFERENCES

[1]. Shu Lin and Daniel J. Costello. Error Control Coding. Prentice Hall, second edition, 2004.

[2] R. G. Gallager, “Low-density parity-check codes”, IRE Trans. Information Theory, vol. IT-8, no. 1, pp. 21–28, January 1962.

[3] D. J. C. MacKay and R. M. Neal, “Near Shannon limit performance of low density parity check codes”, Electronics Letters, vol. 32, no. 18, pp. 1645–1646, March 1997.

[4] R. J. McEliece, The Theory of Information and Coding. Cambridge, U.K.: Cambridge University Press, 2002.

[5]. M. Sipser and D. Spielman, “Expander codes,” IEEE Trans. Inf. Theory, vol. 42, no. 6, pp. 1710–1722, Nov. 1996.

[6]. D. E. Knuth, The Art of Computer Programming, 2nd ed. Reading, MA: Addison Wesley, 2000.

[7]. Allen D. Holliday, Hamming Error-Correction Codes, February 17, 1994 (revised June 15, 2002; March 1, 2004).

[8]. H. Tang, J. Xu, S. Lin, and K. A. S. Abdel-Ghaffar, “Codes on finite geometries,” IEEE Trans. Inf. Theory, vol. 51, no. 2, pp. 572–596, Feb. 2005.

[9]. H. Naeimi and A. DeHon, “Fault-tolerant nano-memory with fault secure encoder and decoder,” presented at the Int. Conf. Nano-Netw.,Catania, Sicily, Italy, Sep. 2007.

[10] S. J. Piestrak, A. Dandache, and F. Monteiro, “Designing fault-secure parallel encoders for systematic linear error correcting codes,” IEEE Trans. Reliab., vol. 52, no. 4, pp. 492–500, Jul. 2003.

[11]. G. C. Cardarilli et al. Concurrent error detection in reed-solomon encoders and decoders. IEEE Trans. VLSI, 15:842–826, 2007.

[12].Hamming, Richard W., “Error Detecting and Error Correcting Codes,” The Bell System Technical Journal 26, 2 (April 1950), 147–160.

[13]. Hill, Raymond. A First Course in Coding Theory. Clarendon Press,1986.

[14]. W. W. Peterson and E. J. Weldon, Jr., Error-Correcting Codes, 2nd Ed. Cambridge, MA: M.I.T. Press, 1972.

[15]L. Edwards, “Low cost alternative to Hamming codes corrects memory errors,” Cornput. Des., pp. 132-148, July 1981.

[16]. Y. Kou, S. Lin and M. Fossorier, Low density parity check codes based on _nite geometries: a rediscovery and more," IEEE Trans. Inform. Theory, vol. 47, pp. 2711-2736, Nov. 2001.

1.

PRADEEP KALAKOTI

2.

KALAKOTI KALYAN