ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

An Enhanced Majority Logic Fault Detection with EG- LDPC for Memory Applications

R.Hemalatha
PG Scholar,
Dept of VLSI & ES,
Gates Institute of Technology,
Gooty, Anantapure, AP, India.

K.Maheswari
Associate Professor,
Dept of ECE,
Gates Institute of Technology,
Gooty, Anantapure, AP, India.

Abstract:

Memory cells have been protected from soft errors for more than a decade; due to the increase in soft error rate in logic circuits, Low density parity check codes are used to detect whether a word has errors in the first iterations of majority logic decoding, and when there are no errors the decoding ends without completing the rest of iterations. A method was proposed to accelerate the majority logic decoding of difference set low density parity check codes. It is useful as majority logic decoding can be implemented serially with simple hardware but requires a large decoding time. This project goal is to reduce the decoding time by stopping the decoding process when no errors are detected. In the first iteration, errors will be detected when at least one of the check equations is affected by an odd number of bits in error. In the second iteration, as bits are cyclically shifted by one position, errors will affect other equations such that some errors undetected in the first iterations will be detected. Results shows that a word can be read from a memory protected with one step MLD EG-LDPC codes, and affected by up to four bit errors, and all these errors can be detected in only three decoding cycles. The conclusion includes DS-LDPC codes are recently presented ones in the simulation results, by making the modified one-step majority logic decoding more attractive for memory applications this increases the memory access time.

KeyWords:

Error correction codes, Euclidean geometry low-density parity check (EG-LDPC) codes, majority logic decoding, memory.

I. INTRODUCTION:

Error correction codes are commonly used to protect memories from so-called soft errors, which change the logical value of memory cells without damaging the circuit.

As technology scales, memory devices become larger and more powerful error correction codes are needed. To this end, the use of more advanced codes has been recently proposed. These codes can correct a larger number of errors, but generally require complex decoders. To avoid a high decoding complexity, the use of one step majority logic decoding codes was first prop osed in for memory applications. Further work on this topic was then presented in one step majority logic decoding can be implemented serially with very simple circuitry, but requires long decoding times. In a memory, this would increase the access time which is an important system parameter. Only a few classes of codes can be decoded using one step majority logic decoding. Among those is some Euclidean geometry low density parity check (EGLDPC) codes which were used in, and difference set low density parity check (DS-LDPC) codes. A method was recently proposed into accelerate a serial implementation of majority logic decoding of DS-LDPC codes. For terrestrial radiation environments where there is a low soft error rate (SER), codes like single error correction and double error detection (SEC–DED), are a good solution, due to their low encoding and decoding complexity. However, as a consequence of augmenting integration densities, there is an increase in the number of soft errors, which produces the need for higher error correction capabilities The usual multi error correction codes, such as Reed–Solomon (RS) or Bose Chaud- huri–Hocquenghem (BCH) are not suitable for this task. The reason for this is that they use more sophisticated decoding algorithms, like complex algebraic decoders that can decode in fixed time, and simple graph decoders, that use iterative algorithms Both are very complex and increase computational costs Among the ECC codes that meet the requirements of higher error correction capability and low decoding complexity, cyclic block codes have been identified as good candidates, due to their property of being majority logic (ML) decodable A sub- group of the low-density parity check (LDPC) codes, which be- longs to the family of the ML decodable

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

codes, has been re- searched in [9]-[11]. However, many transient faults will not be latched. Some of the latched data may not be relevant to machine operation and there will be no perceivable error in the program operation. Hence, the effective error rate of a large combinational circuit needs to be detectedThe main reason for using ML decoding is that it is very simple to implement and thus it is very practical and has low complexity. The drawback of ML decoding is that, for a coded word of bits, it takes cycles in the decoding process, posing a big impact on system performance One way of coping with this problem is to implement parallel encoders and decoders. This solution would enormously increase the complexity and, therefore, the power consumption. As most of the memory reading accesses will have no errors, the decoder is most of the time working for no reason. This has motivated the use of a fault detector module that checks if the codeword contains an error and then triggers the correction mechanism accordingly. In this case, only the faulty codewords need correction, and therefore the average read memory access is speeded up, at the expense of an increase in hardware cost and power consumption.

A similar proposal has been presented in for the case of flash memories The simplest way to implement a fault detector for an ECC is by calculating the syndrome, but this generally implies adding another very complex functional unit. This paper explores the idea of using the ML decoder circuitry as a fault detector so that read operations are accelerated with almost no additional hardware cost. The results show that the properties of DSCC-LDPC enable efficient fault detection 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 referred. This is the case for OLS codes that are commonly used in high-speed applications The remainder of this paper is organized as follows. Section II gives an overview of existing ML decoding so- lutions; Section III presents the novel ML detector/decoder

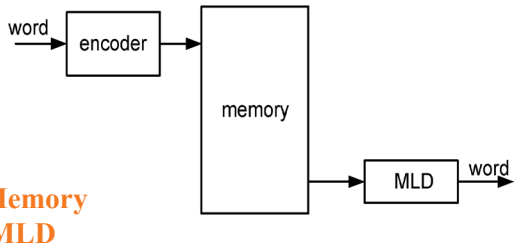

Fig1.Memory with MLD

(MLDD) using difference-set cyclic codes; Section IV discusses the results obtained for the different versions in respect to effectiveness, performance, and area and power consumption. Finally, Section V discusses conclusions and gives an outlook onto future work.

II.MAJORITY LOGIC DECODING (MLD):

An n-bit code-word c, which encodes k-bit information vector i is generated by multiplying the k-bit information vector with k × n bit generator matrix G, i.e., c = i • G. Figure 3.2 shows the generator matrix of (15, 7) EG-LD-PC code. all the rows of the matrix are cyclic shifts of the first row. This cyclic code generation does not generate a systematic code 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.

The generator matrix of any cyclic code can be converted into systematic form (G = [I : X]) .

	c_0	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_{10}	c_{11}	c_{12}	c_{13}	c_{14}
i_0	1	0	0	0	1	0	1	1	1	0	0	0	0	0	0
i_1	0	1	0	0	0	1	0	1	1	1	0	0	0	0	0
i_2	0	0	1	0	0	0	1	0	1	1	1	0	0	0	0
i_3	0	0	0	1	0	0	0	1	0	1	1	1	0	0	0
i_4	0	0	0	0	1	0	0	0	1	0	1	1	1	0	0
i_5	0	0	0	0	0	1	0	0	0	1	0	1	1	1	0
i_6	0	0	0	0	0	0	1	0	0	0	1	0	1	1	1

Figure.2.The generator matrix of EG-LDPC code of (15, 7) in cyclic format.

figure 1 shows the systematic generator matrix to generate (15, 7) EG-LDPC code. The encoded vector, which is generated by the inner product of the information vector and the generator matrix, consists of information bits followed by parity bits, where each parity bit is simply an inner product of information vector and a column of X, from G = [I : X]. The encoder circuit to compute the parity bits of the (15, 7) EG-LDPC code. In this figure i = (i0, ..., i6) is the information vector and will be copied to c0, ..., c6 bits of the encoded vector, c, and the rest of encoded vector, the parity bits, are linear sums (xor) of the information bits. If the building block is two-input gates then the encoder circuitry takes 22 twoinput xor gate. Since the systematic generator matrix of EG-LDPC and PG-LDPC codes does not have the standard row and column density,

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

To compute the area of an encoder circuitry the corresponding systematic generator matrix has to be constructed. Once the systematic generator matrix is constructed the fanin size of the xor gates can be determined by the column densities of the generator matrix.One-step majority-logic correction is a fast and relatively compact error-correcting technique. 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.

1) 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 , which may demand multiple iterations of error diagnosis and trial correction. Avoiding iteration makes the correction latency both small and deterministic. This technique can be implemented serially to provide a compact implementation or in parallel to minimize correction latency. 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.

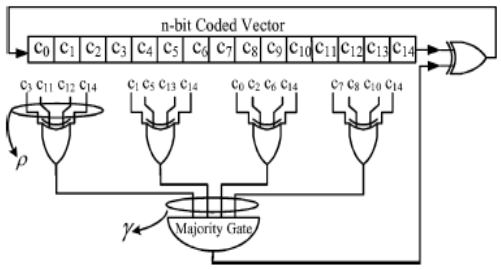

Figure.3.Serial one-step majority logic corrector structure.

A linear sum of the received encoded vector bits can be formed by computing the inner product of the received vector and a row of a parity-check matrix. This sum is called Parity-Check sum. A set of parity-check sums is said to be orthogonal on a given code bit if each of the parity-check sums include the code bit but no other code bit is included in more than one of these parity-check sums. If for each code bit there are j parity-check sums that are orthogonal on it, then the code is one-step majority-logic correctable up to bj/2c bit errors. In a cyclic code, a set of j parity-check sums orthogonal on a code-word bit is orthogonal on all the n code-word bits. Therefore, using one set of parity-check matrix rows orthogonal on one code bit, we can design a majority circuit that corrects all the other bits, serially.The one-step majority logic error correction is summarized in the following procedure.These steps correct a potential error in one code bit, e.g., cn−1. 1. The j parity-check sums orthogonal on cn−1 are formed by computing the inner product of the received vector and the appropriate rows of parity-check matrix. 2. The J orthogonal check sums are fed into a majority gate. The output of the majority gate corrects the bit cn−1, by inverting the value of cn−1 if the output of majority gate is “1”.The circuit implementing a serial one-step majority logic corrector for (15, 7) EGLDPCcode is shown in figure 2 is circuit generates parity-check sums with xor gates and then computes, the majority value of the parity-check sums. Since each parity-check sum is computed using a row of the parity-check matrix and the row density of EG-LDPC codes are _ then each xor gate that computes the linear sum has _ inputs. The single xor gate on the right, corrects the code bit cn−1, using the output of the majority gate. Once the code bit cn−1 is corrected the code-word is cyclic shifted and code bit cn−2 is placed at cn−1 position and will be corrected. The whole code-word can be corrected in n rounds.

V. IMPLEMENTATION AND RESULTS:

The proposed Memory testing using MLD is designed using Verilog hard ware description language and structural form of coding. The basic block of paper is Xor matrix, majority gates and cyclic shift registers. The design is completely synchronized by the clock. The code is completely synthesized using Xilinx XST and implemented on device family Spartan 3e, device XC3S50OE, package FG320 with speed grade -4.

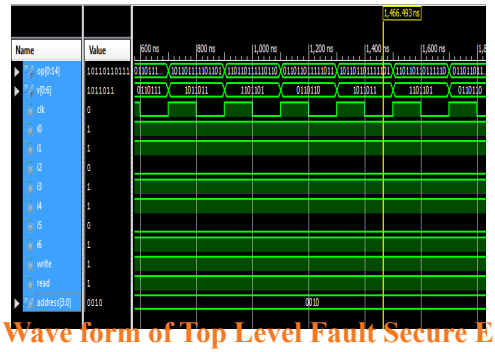

Figure 3 Wave form of Top Level Fault Secure Encoder and Decoder.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

V. CONCLUSION:

In this brief, the detection of errors during the first iterations of serial one step Majority Logic Decoding of EG-LDPC codes has been studied. The objective was to reduce the decoding time by stopping the decoding process when no errors are detected. The simulation results show that all tested combinations of errors affecting up to four bits are detected in the first three iterations of decoding. These results extend the ones recently presented for DS-LDPC codes, making the modified one step majority logic decoding more attractive for memory applications. The designer now has a larger choice of word lengths and error correction capabilities. Future work includes extending the theoretical analysis to the cases of three and four errors. More generally, determining the required number of iterations to detect errors affecting a given number of bits seems to be an interesting problem. A general solution to that problem would enable a fine-grained tradeoff between decoding time and error detection capability.

REFERENCES:

[1] R. G. Gallager, Low-Density Parity-Check Codes. Cambridge, MA:MIT Press, 1963.

[2] R. G. Gallager, "Low Density Parity Check Codes" , PhD dissertation, MIT, 1963 .[3] R. J. McEliece, The Theory of Information and Coding. Cambridge,U.K.: Cambridge University Press, 2002.

[4] W. E. Ryan, "An Introduction to LDPC Codes", in CRC Handbook for Coding and Signal Processing for Recording Systems (Ed. B. Vasic), CRC Press, 2004.

[5] M. Karkooti, J.R. Cavallaro, "Semi-Parallel Reconfigurable Architectures for Real-Time LDPC Decoding", ITCC 2004.

[6] H. Zhong, T. Zhang, "Design of VLSI Implementation-Oriented LDPC Codes", IEEE Semiannual Vehicular Technology Conference (VTC), Oct. 2003.

[7] E. Yeo, B. Nikolic, and V. Anantharam, "Architectures and Implementations of Low-Density Parity Check Decoding Algorithms", IEEE International Midwest Symposium on Circuits and Systems, August. 2002.

[8] H. Naeimi and A. DeHon, "Fault secure encoder and decoder for memory applications," in Proc. IEEE Int. Symp. Defect Fault Tolerance VLSI Syst., Sep. 2007, pp. 409–417.

[9] S.L. Howard, V.C. Gaudet, and C. Schlegal, "Soft-Bit Decoding of Regular Low-Density Parity Check Codes", IEEE Transactions on Circuits and Systems.

[10] 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.

[11] 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..

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

Author's Details:

R.Hemalatha
pursuing her M.Tech (VLSI &Embedded Systems Design) Stream From Gates Institute of Technology , Goofy , Anantapure ,Andhra pradesh Her areas of interest area in VLSI and Embedded system designing Fields.

K.Kmaheswari
Mtech,(cne) working as a Associate professor For ECE Department in Gates Institute of Technology ,Gooty , Anantapure Andhra Pradesh Her ,Areas of interest in Mobile communication , wireless communication ,cryptography .