International Journal of Advance Research in Science and Engineering
Volume No.06, Special Issue No.(01), December 2017
www.ijarse.com
IJARSE
ISSN: 2319-8354

ERROR DETECTION IN MAJORITY LOGIC DECODING OF EUCLIDEANGEOMETRY LOW DENSITY PARITY CHECK (EG-LDPC) CODES

[1]Anvesh Thatikonda , [2]Dr. R.P. Singh

[1]Research Scholar, Sri Satya Sai University of Technology and Medical Sciences, India)

ABSTRACT

The fault secure memory system consist encoder, decoder, detector and parallel pipelined corrector. In this the data should be transferred through the encoder and then it will be decoded, the decoder consist the detector and corrector if the data have any errors that should be detected by detector and then that data will be corrected by the parallel pipelined corrector. So it has large decoding time to correct the data by using parallel pipeline corrector. To reduce this one here we are using majority logic decoding with EGLDPC (Euclidean Geometry Low Density Parity Check) codes. In this method the system was check whether the data has any errors in the first iterations of the majority logic decoding if it does not has any errors the decoding part will be ended without completing the rest of iterations. With this most of the words have no errors in the memory and the average time is also reduced.

I INTRODUCTION

Memories are the most universal component today but they are prone to errors like soft and transient errors. Single Event Upsets (SEU) is the type of fault which alters these memories by changing its states which is caused by ions or electro-magnetic radiations. SEU occurs in digital circuits when an energized particle, or electron, causes a transistor to flip on or off from its correct state. This happens in microcircuits, including memory chips, communication devices, power circuits, and microprocessors. Such a flip of one bit can cause a computer or other electronic device to lockup or crash.

Circuit components, including configuration memory cells, user memory, and registers, can also be affected. Some type of embedded memory, such as ROM, SRAM, DRAM, flash memory etc is seen in almost all system chips. Now days, the memory failure rates and performance failures may occur for every application are increasing due to the impact of technology scaling-smaller dimensions, high integration densities, lower operating voltages etc, therefore this posing a major reliability concern for many applications. Some commonly used mitigation techniques are Triple Modular Redundancy (TMR) and Error Correction Codes (ECCs). For memories, it turned out that Error Correction Codes (ECC) is the best way to mitigate memory soft errors. Cyclic block codes have the property of

being Majority Logic (ML) decodable. Therefore cyclic block codes have been identified as more suitable among the ECC codes that meet the requirements of higher error correction capability and low decoding complexity. Euclidean Geometry Low-Density Parity Check (EG-LDPC) codes, a subgroup of the Low-Density Parity Check (LDPC) codes, which belongs to the family of the ML decodable codes. The ML decoding are that it is very simple to implement and thus it is very practical and has low complexity. The drawback of ML decoding is that, it needs as many cycles as the number of bits in the input signal, which is also the number of taps, N, in the decoder and also same decoding time for both error and error free code words. This is a great impact on the performance of the system, so to avoid this drawback the proposed ML decoding can detect all errors within three cycles. It is practical to generate and check all possible error combinations for codes with small words and affected by a small number of bit flips. Only three cycles are needed to detect all errors affecting up to four bits in EG LDPC Codes.

II PRELIMINARIES

Finite geometries have been used to derive many error-correcting codes. One example is EG-LDPC codes which are based on the structure of Euclidean geometries over a Galois field. Among EG -LDPC codes there is a subclass of codes that is one step majority logic decoding (MLD). Codes in this subclass are also cyclic. The parameters for some of these codes are given in Table I, where N is the block size, the number of information bits, K the number of MLD check equations and J the number of errors that the code can correct using one step MLD. One step MLD can be implemented serially using the scheme in Figure. 1 which corresponds to the decoder for the EG-LDPC code with .First the data block is loaded into the registers. Then the check equations are computed and if a majority of them has a value of one, the last bit is inverted. Then all bits are cyclically shifted. This set of operations constitutes a single iteration: after iterations, the bits are in the same position in which they were loaded. In the process, each bit may be corrected only once. As can be seen, the decoding circuitry is simple, but it requires a long decoding time if is large. The check equations must have the following properties.

Table 1 One step MLD EG-LDPC CODES

N	K	J	t_{ML}
15	7	4	2
63	37	8	4
255	175	16	8
1023	781	32	16

International Journal of Advance Research in Science and Engineering
Volume No.06, Special Issue No.(01), December 2017
www.ijarse.com
IJARSE
ISSN: 2319-8354

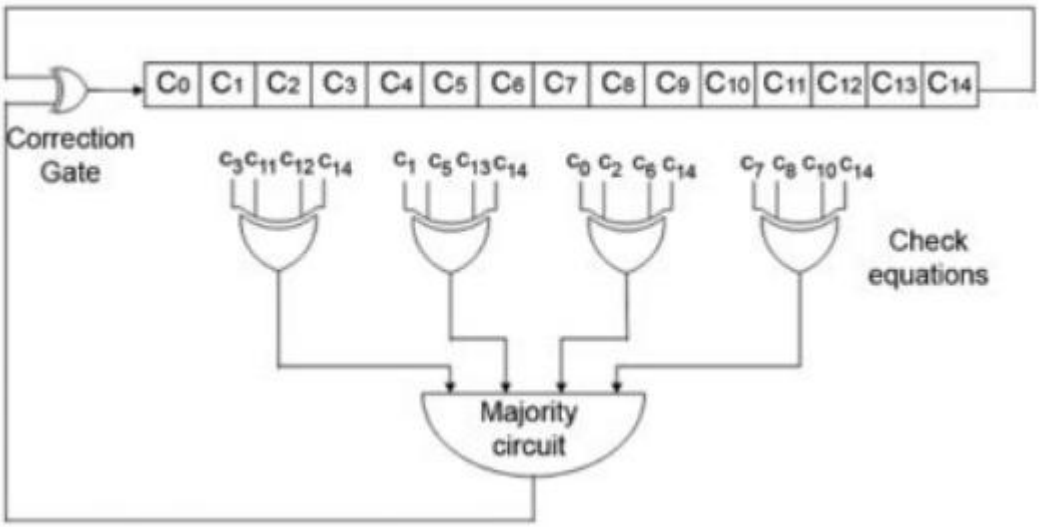


Figure 1: Serial one-step majority logic decoder for the (15, 7) EG-LPDC code

2.1 Fault Secure Detector

The center of the indicator operation is to create the disorder vector, which is fundamentally executing the accompanying vector-framework augmentation on the got encoded vector what's more, equality check grid along these lines every piece of the disorder vector is the result of with one column of the equality check grid. This item is a direct twofold whole over digits of where the relating digit in the framework line is 1. This paired whole is executed with an XOR door. Since the column weight of the equality check grid is, to produce one digit of the disorder vector we require a - info XOR entryway, or 2-information XOR doors. For the entirety locator, it takes 2-information XOR entryways. Table II delineates this amount for a percentage of the littler EG-LDPC codes. Note that executing every disorder piece with a different XOR entryway fulfills the suspicion of Theorem I of no rationale partaking in identifier circuit usage. A mistake is distinguished if any of the disorder bits has a nonzero esteem. The last blunder location sign is actualized by an OR capacity of all the disorder bits.

2.2 Encoder

A - Bit code word, which encodes a - bit data vector is created by duplicating the - bit data vector with a bit generator network; i.e.,. EG-LDPC codes are not efficient and the data bits must be decoded from the encoded vector, which is not attractive for our shortcoming tolerant methodology because of the further complexity what's more, postpone that it adds to the operation. Be that as it may, these codes are cyclic codes. We utilized the strategy

introduced as a part of and to change over the cyclic generator frameworks to deliberate generator lattices for all the EG-LDPC codes under thought.

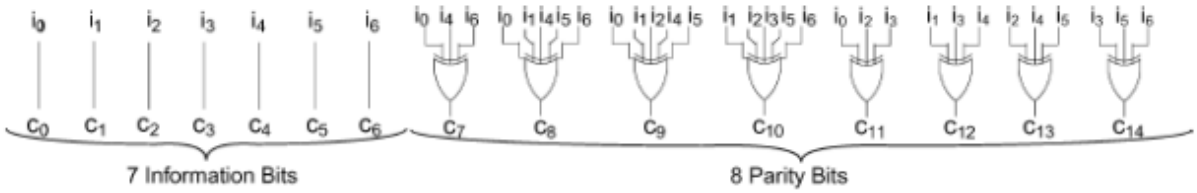


Figure 2 Structure of an encoder circuit

Above figure demonstrates the precise generator framework to create (15, 7, 5) EG-LDPC code. The encoded vector comprises of data bits took after by equality bits, where every equality bit is just an inward result of data vector and a segment of , from . Figure. 2 demonstrates the encoder circuit to figure the equality bits of the (15, 7, 5) EG-LDPC code. In this figure is the data vector and will be replicated to bits of the encoded vector, and whatever remains of encoded vector, the equality bits, are direct aggregates (XOR) of the data bits. On the off chance that the building square is two-information entryways then the encoder hardware takes 22 two-info XOR doors. Table II demonstrates the zone of the encoder circuits for each EG-LDPC codes under thought in light of their generator networks.

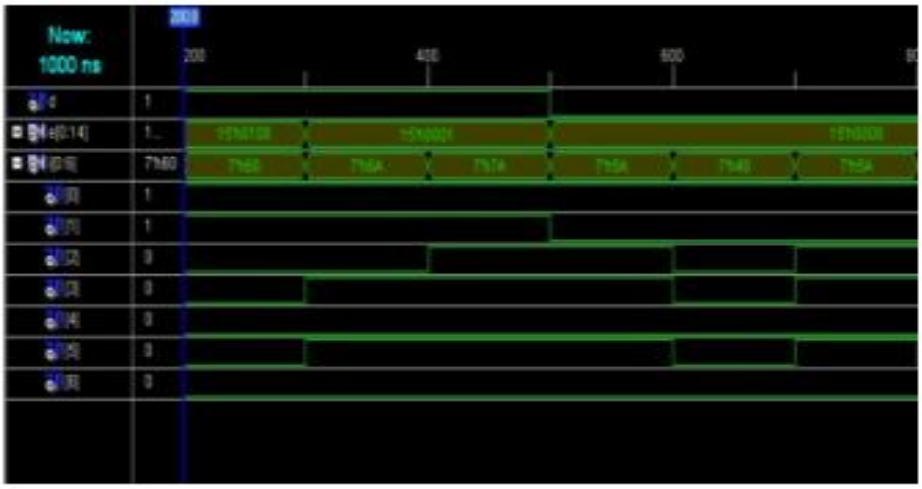
2.3 Corrector

One-stage greater part rationale remedy is a quick and moderately minimal blunder adjusting procedure. There is a constrained class of ECCs that are one-stage greater part correctable which incorporate sort I two dimensional EG-LDPC. In this segment, we exhibit a brief survey of this adjusting procedure. At that point we demonstrate the one-stage larger part rationale corrector for EG-LDPC codes. 1) One-Step Majority-Logic Corrector: One-stage majority logic amendment is the methodology that recognizes the right esteem of an every piece in the code word specifically from the got code word; this is as opposed to the general message-passing error correction procedure which may request numerous emphases of blunder finding and trial rectification. Keeping away from cycle makes the remedy dormancy both little and deterministic. This strategy can be actualized serially to give a minimized execution on the other hand in parallel to minimize remedy inactivity

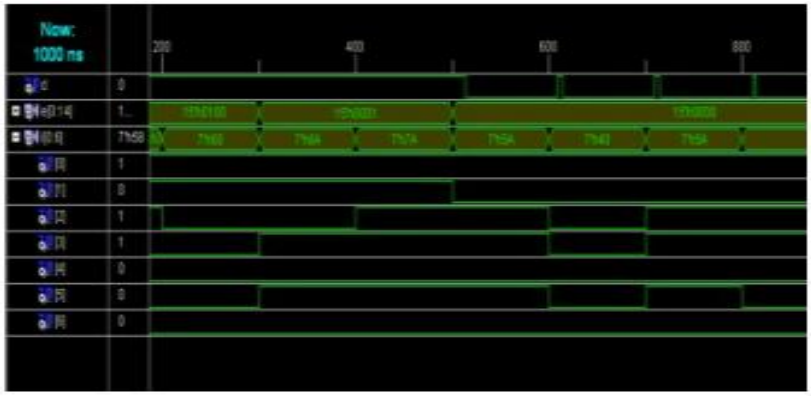
III RESULTS

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.

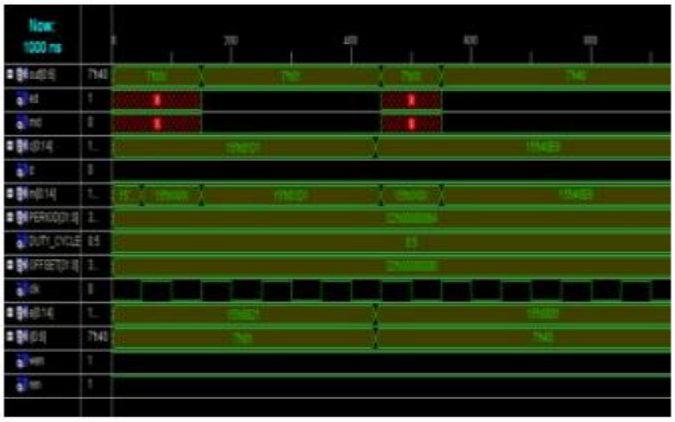
International Journal of Advance Research in Science and Engineering
Volume No.06, Special Issue No.(01), December 2017
www.ijarse.com
IJARSE
ISSN: 2319-8354



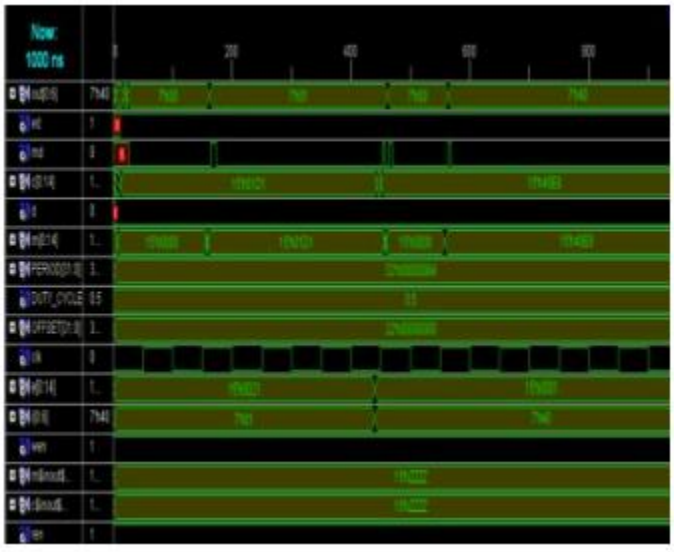
Behavioral simulation waveform for the fault secure encoder



Post route simulation waveform for the fault secure encoder



Behavioral simulation waveform for the fault secure memory system



Post route simulation waveform for the fault secure memory system

IV CONCLUSION

In this paper we have designed serial one step majority logic decoder for EGLDPC codes to detect the errors in data. The proposed design is mostly decrease the decoding time in the decoding process for error detection. The proposed simulation results can shows that the combination detection of errors for the iterations of EG LDPC codes. This design also used to detect the all errors effecting five or fewer bits.

REFERENCES

[1] R. C. Baumann, "Radiation-induced soft errors in advanced semiconductortechnologies," *IEEE Trans. Device Mater. Reliab.*, vol. 5, no. 3,pp. 301–316, Sep. 2005.

[2] M. A. Bajura, Y. Boulghassoul, R. Naseer, S. DasGupta, A. F.Witulski,J. Sondeen, S. D. Stansberry, J. Draper, L. W. Massengill, and J. N.Damoulakis, "Models and algorithmic limits for an ECC-based approachto hardening sub-100-nm SRAMs," *IEEE Trans. Nucl. Sci.*, vol.54, no. 4, pp. 935–945, Aug. 2007.

[3] R. Naseer and J. Draper, "DEC ECC design to improve memory reliabilityin sub-100 nm technologies," *Proc. IEEE ICECS*, pp. 586–589,2008.

[4] S. Ghosh and P. D. Lincoln, "Dynamic low-density parity check codesfor fault-tolerant nano-scale memory," presented at the FoundationsNanosci. (FNANO), Snowbird, Utah, 2007.

[5] S. Ghosh and P. D. Lincoln, "Low-density parity check codes for errorcorrection in nanoscale memory," SRI Computer Science Lab., MenloPark, CA, Tech. Rep. CSL-0703, 2007.

[6] H. Naeimi and A. DeHon, "Fault secure encoder and decoder formemory applications," in *Proc. IEEE Int. Symp. Defect Fault Toler.VLSI Syst.*, 2007, pp. 409–417.

[7] B. Vasic and S. K. Chilappagari, "An information theoretical frameworkfor analysis and design of nanoscale fault-tolerant memoriesbased on low-density parity-check codes," *IEEE Trans. Circuits Syst.I, Reg. Papers*, vol. 54, no. 11, pp. 2438–2446, Nov. 2007.

[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. Lin and D. J. Costello, *Error Control Coding*, 2nd ed. EnglewoodCliffs, NJ: Prentice-Hall, 2004.

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

[11] H. Tang, J. Xu, S. Lin, and K. A. S. Abdel-Ghaffar, "Codes on finitegeometries," *IEEE Trans. Inf. Theory*, vol. 51, no. 2, pp. 572–596, Feb.2005.