DESIGN OF IMPROVED MAJORITY LOGIC FAULT DETECTOR/ CORRECTOR BASED ON EUCLIDEAN GEOMETRY LOW DENSITY PARITY CHECK (EG-LDPC)CODES

A.JAYASREE, M.TECH
DEPT OF ELECTRONICS AND COMM.
GOLDEN VALLEY ENGINEERING COLLEGE
MADANAPALLE, INDIA

P. DHANEEF KUMAR, M.TECH, ASST PROF
DEPT. OF ELECTRONICS AND COMM.
GOLDEN VALLEY ENGINEERING COLLEGE
MADANAPALLE, INDIA

Abstract -- As the engineering enhances the memory devices get to be bigger, so capable error correction codes will be required. Error correction codes are by and large used to ensure memories from soft errors, which change the logical codes of memory cells without harming the circuit. These codes will redress countless, yet typically require complex decoders. To make this decoding less difficult in this project it utilizes Euclidean geometry LDPC (EG-Ldpc)codes with one step majority decoding system, on account of their fault-secure detector capacity. This strategy identifies words contains error in the beginning cycle of the majority logic decoding process and decreases the decoding time by completion the decoding methodology when no errors are detected and reductions the memory access time. The result got past this system will demonstrate that it is a successful and smaller error correcting procedure. It is suitable for commonsense VLSI execution in applications obliging quick decoders.

Index Terms – Memory, Error correction codes, majority logic decoding, Euclidean geometry low density parity check (EG-LDPC) codes.

I. INTRODUCTION

The reliability and security of memories are basic contemplation in the present day electronic system plan. Soft error happens when a radiation occasion causes a sufficient charge irritation to Reverse or flip the data state of a memory cell, register [2]. As engineering scales, memory devices become greater and all the more compelling error amendment codes are obliged to guarantee memories from soft errors [3], [4]. The error is "soft "because it will change the reason estimation of memory cells without hurting the circuit/device. The soft error is similarly implied as a Single Event Upset (SEU). In case the radiation occasion is of high imperativeness, more than a lone bit conceivably impacted, making a Multi Bit Upset (MBU) [2].

For dependable correspondence, errors must be identified and corrected. Some multi error bit correction codes are BCH codes, Reed Solomon codes, however in which the computation is to a great degree troublesome. These codes can amend incalculable, however oblige complex decoders [10], [11]. Among the error correction codes, cyclic square codes have higher error distinguishment limit, low decoding multifaceted nature and that are majority logic (ML) decodable. A low-density parity check (LDPC) code is an immediate error altering code, used to avoid a high interpreting multifaceted nature [6]-[9]

In this paper, one specific sort of low density parity check codes, to be particular Euclidean Geometry-LDPC codes [1] are used due to their weakness secure locator Capacity, higher unwavering quality and lower extend overhead. The EG-LDPC codes which are one step majority logic decoder.

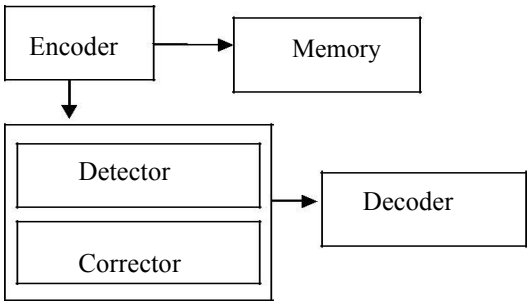


Fig 1.Detector and corrector in EG geometry

Distinctive error discovery systems are used to avoid the soft error [10]. One of the schedules is majority logic decoder which used to get and right the error in essential way. This system uses the first iteration of majority logic decoding to distinguish the error present in the interpretation. In case there are no errors, then the decoding approach may be stopped without completing the remaining iterations [1]. The key reason behind using Majority Logic Decoding (MLD) is that it is not hard to complete and has a low many-sided quality [11]. As depicted in former, Majority-logic decoder is a fundamental and feasible decoder prepared for reviewing different bit flips depending upon the amount of equality check aggregate examinations. It contains four segments: 1) a cyclic movement enlist; 2) a XOR network; 3) a larger part entryway; 4) an EXOR entryway for error correction, as outlined in figure 2.

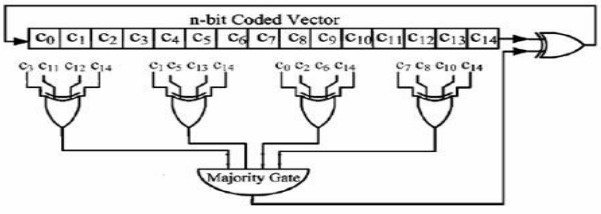


Fig 2: Decoder of One step Majority Logic for (15, 7) G-LDPC Codes

ISSN: 2278 – 1323

In one step majority logic decoding [1], from the get go the code word is stacked into the cyclic shift register. By then the check comparisons are registered. The following results are then sent to the majority gate for evaluating its accuracy. In case the amount of 1‟ s got in is more significant than the amount of 0‟ s which intimates that the current bit under decoding isn't correct, and a sign to modify it would be activated. For the most part the bit under decoding is correct and no extra operations would be needed on it. In next, the substance of the registers are rotated and the above system is reiterated until code word bits have been readied. Finally, the parity check whole would be zero if the code word has been faultlessly decoded. In this process, each bit may be reconsidered simply once.

Hence, the decoding circuitry is fundamental, yet it obliges a long decoding time if the code word is considerable. Along these lines, by one majority logic decoding, the code is prepared for reviewing any error outline with two or less errors. For example, for a code outflow of 15-bits, the decoding would take 15 cycles, which would be excessive for by and large applications. The majority logic decoders itself go about as a fault detector.

MLD decoding all code word bits, the MLD methodology stops transitionally in the third cycle, which can fit to find up to five bit flips in three decoding cycles. So the amount of decoding cycles could be reduced to get upgraded execution. The schematic of majority logic decoder/detector is represented in figure3.

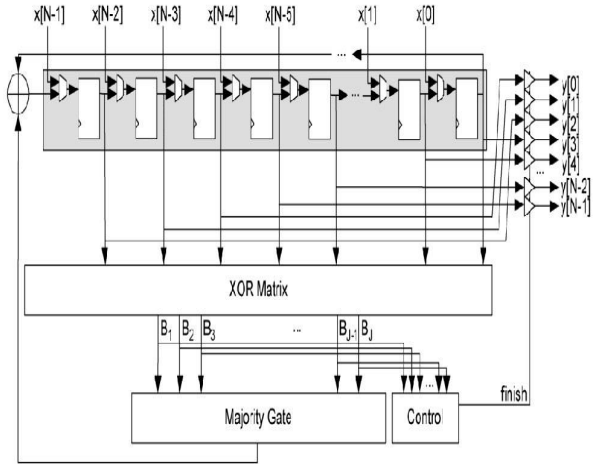


Fig 3: Schematic of Majority Logic Decoder/Detector (MLD)

At the outset the code word is secured into the cyclic shift register and it moved through all the taps. The widely appealing values in every one tap is given to the XOR matrix to perform the check whole comparisons. The resulting wholes are then sent to the majority part passage for evaluating its rightness. If the amount of 1‟ s got is more unmistakable than the amount of 0‟ s which would suggest that the current bit under decoding isn't correct, so it move ahead forward the decoding technique. For the most part, the bit under decoding would be correct and no extra operations would be needed on it. Decoding methodology including the operation of the substance of the registers is turned and the above framework is reiterated and it stops transitionally in the third iteration. On the off chance that in the beginning three cycles of the decoding process, the appraisal of the XOR matrix for all is "0," the code word is determined to be blunder free and sent particularly to the yield. In case the mistake contains in any of the three cycles at any rate a "1," it would move ahead with the whole decoding methodology to execute the errors.

Finally, the parity check totals should be zero if the code word has been exactly decoded. Considering everything the MLD system is used to recognize the five bit errors and right four bit errors effectively. In case the code word contains more than five bit error, it makes the yield anyway it didn't show the errors exhibited in the data. This kind of error is known as the silent information error. Drawback of this method is did not identifying the silent information error and it eating up the locale of the majority part gateway. The schematic for this memory structure is demonstrated in figure 5. It is really apparently equivalent to the one showed in fig. 1; additionally the control unit was incorporated the MLD module to manage the interpreting procedure (to detect the error).

The two basic steps involved in calculating in detected code word are as follows:

1) Create parity check wholes by registering the internal aftereffect of the got vector and the correct rows of parity check matrix.

2) The register wholes are managed with a majority gate. The yield of the majority gate corrects the bit by irritating the quality if the yield of majority gate is "1".

The serial bigger part corrector takes cycles to change an off base codeword. In case the flaw rate is low, the corrector piece is used at times; since the ordinary case is lapse free code words, the idleness of the corrector won't have a genuine impact on the typical memory read inactivity.

The one stage majority logic detector may be used to redress all the n bits of the got codeword of a cyclic code. To identify every one code bit, the got encoded vector is cyclic moved and nourished into the XOR gates. The information bits are bolstered into the encoder to encode the information vector. The deficiency secure detector of the encoder checks the authenticity of the encoded vector. On the off chance that the detector identifies any blunder the encoding operation must be patched up to deliver the right codeword. The codeword is then secured in the memory. Amid memory access operation, the set away code words will be gotten to from the memory unit.

A corrector unit is intended to rectify potential errors in the retrieved codeword's.

MLD is focused around different parity check equations which are orthogonal to each other, so that at

ISSN: 2278 – 1323

each cycle each codeword bit shares in one helps all equations. For this reasons the larger part eventual outcome of these parity check equations pick the rightness of the right bit under decoding. Generic schematic of a memory framework is portrayed for the utilization of a ML decoder.

In this method at the outset, the data words are encoded and after that set away in the memory the following sums are then sent to the majority logic gate for surveying its rightness. If the amount of 1" s got in is more imperative than the amount of 0" s, which would intimate that the current bit under decoding isn't correct and a signal to redress it would be enacted. In general, the bit under decoding would be correct and no extra operations would be needed on it and it fabricates decoder. Regardless they oblige a far reaching decoding time that impacts memory performance

To improve the decoder execution, choice outlines may be used. One possibility is to incorporate a shortcoming detector by figuring the disorder, so simply defective code words are decoded. Since the majority logic of the code word will be error free, no further correction will be obliged thus execution won't be affected. Notwithstanding the way that the utilization of a SFD diminishes the typical of the translating technique.

A. Encoder

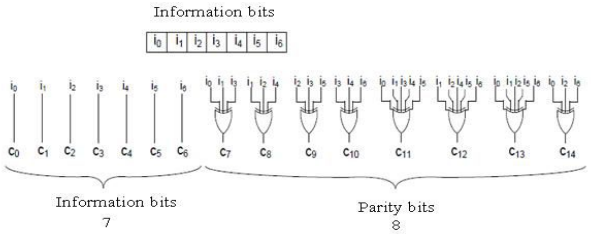


Fig 4.Encoded data from encoder (7-bits of information)

In encoder the information bit of 7bits of data is encoded .It makes encoded bits in which it has data and what's more parity check bits. These parity bits are accountable for detecting the region of error and subsequently right the inaccurate word.

The output conveyed will be encoded bits c1,c2,c3,c4,c5,c6,c7,c8,c9,c10,c11,c12,c13,c14. With everything taken into account the encoder trades the going hand in hand with information data to the memory. Here the data is the 7-bits of data, and a while later the encoded data is secured in the memory devices of the electronic devices.

B .Detector & Corrector

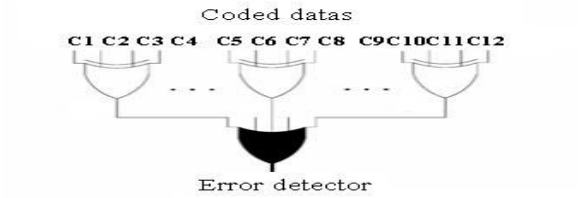


Fig.5. Fault secure detector for (15, 7, 5) EG-LDPC codes

The basic operation of the detector is to deliver the syndrome vector. This is gotten by a matrix duplication on the got coded vector c and the transpose of the parity check matrix H

,S=C*(H^T)

The encoded bits and one line of the parity check matrix is copied to get the syndrome vector. On the off chance that the learned syndrome includes the region of ones, then an error is acknowledged to have happened. All zeros demonstrate no error.

The vector matrix growth is executed with the aid of XOR gates. The illumination to these gates depends on upon the weight of parity check matrix. On the off chance that the weight is thought to be p then p data XOR gates are required. Else two (p-1) information gates are required. As showed in the above outline, each syndrome bit is figured using separate XOR gates. Consequently there is no rationale conferring in the detector execution and in this way is a shortcoming secure detector. The last error identification signal is realized by an OR limit of all the syndrome bits. The yield of this n-incorporate OR entryway is the error detector signal.

C. Decoder

In the decoder the balanced bits are decoded. When in doubt the decoder recovers the first data which is send by the sender. The information of the decoder is completely 15-bits of data in which it embodies 7bits of data and 8 bits of encoded bits. This recovery is centered around Euclidean geometry which means uproot between 1" s must be little.

According to this geometry it picks the data bits and performs the XOR operation. Therefore the output will be the error free data bits.

The 15-bits of data is given as output to decoder. These 15 bits of data includes 7bits of data bits and 8 bits of parity bits. By then figure the disorder bits and check whether the disorder bit is "1" or "0".if the disorder bit is "0" then it is thought to be error free however in the event that the disorder bit is "1" then it is considered to be error.

A method was starting late proposed into animate a serial execution of majority logic decoding of DS-LDPC codes. The thought behind the framework is to use the first iteration of majority logic decoding to detect if the maxim being decoded contains errors. If there are no errors, then decoding may be stopped without completing the remaining cycles, thusly phenomenally lessening the decoding time.

For a code with block length, majority logic decoding (when executed serially) requires iterations, so that as the code size creates, so does the decoding time. In the proposed methodology, simply the initial three cycles are used to detect errors, in this way accomplishing a far reaching speed in- fold when is

ISSN: 2278 – 1323

gigantic. In it was exhibited that for DS-LDPC codes, all error mixes of up to five errors could be detected in the first.

TABLE I

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

Table. 1. Serial one-step majority logic decoder for the EG-LPDC code, three iterations.

Moreover, errors impacting more than five bits were detected with a possibility close to one. The possibility of undetected errors was in like manner found to decrease as the code piece length stretched. For a billion error plans simply a few errors (or sometimes none) were un- detected. This may be sufficient for a couple of uses

An interchange inclination of the proposed framework is that it requires just about no additional circuitry as the decoding circuitry is moreover used for error detection. For example, it was shown in that the additional district required to realize the arrangement was just around 1% for considerable word sizes.

The system proposed in relies on upon the properties of DS-LDPC codes and as needs be it is not particularly applicable to other code classes. In the going with, a relative procedure for EG-LDPC codes is presented. Whatever is left of this short is parceled into the going with territories. Range II gives preliminaries on EG-LDPC codes, majority logic decoding and the framework proposed in. Region III presents the results of applying the framework to EG-LDPC codes, involving reenactment results and a theory centered around those results. This is supplemented by a hypothetical check for the occasions of one and two errors that is given in an informative supplement.

II. PRELIMINARIES

Constrained geometries have been used to focus various error-reviewing codes. One example is EG-LDPC codes which are centered around 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 decodable (MLD). Codes in this subclass are similarly cyclic. The parameters for some of these codes are given in Table I, where is the piece measure the amount of data bits, the amount of MLD check mathematical statements and the amount of errors that the code can cure using one step MLD.

One step MLD may be realized serially using the arrangement in fig.1 which contrasts with the decoder for the EG-LDPC code with. In the first place the data square is stacked into the registers. By then the check mathematical statements are transformed and if a majority of them has an estimation of one, the last bit is vexed. By then all bits are cyclically moved. This set of operations constitutes a single emphasis: after cycle, the bits are in the same position in which they were stacked. The entire time, each bit may be updated simply once. As may be seen, the decoding hardware is clear, yet it obliges a long decoding time if is sweeping. The check mathematical statements must have the going hand in hand with properties.

TABLE II
UNDETECTED ERRORS IN EXHAUSTIVE CHECKING

N	1 error	2 errors	3 errors	4 errors
15	0	0	0	0
63	0	0	0	0
255	0	0	0	--
1023	0	0	--	--

1) All equations incorporate the variable whose quality is put away in the last register (the one checked as).

2) Whatever remains of the registers are incorporated in at most one of the check equations.

If errors could be detected in the introductory couple of iteration of MLD, then at whatever point no errors are detected in those cycles, the unraveling may be ended without completing whatever is left of the iterations. In the first iteration, errors will be detected when no short of what one of the check equations is affected by an odd number of bits in slip. In the second cycle, as bits are cyclically moved by one position, errors will impact distinctive equations such that a couple of errors undetected in the first iteration will be detected. As cycle's improvement, all perceptible errors will in the long run be detected.

In it was exhibited that for DS-LDPC codes most errors could be detected in the introductory three cycles of MLD. In perspective of entertainment results and on a hypothetical affirmation for the occurrence of two errors, the going with theory was made.

"Given an statement that read from a memory guaranteed with DS-LDPC codes, and affected by up to five bit-flips, all errors could be detected in only three decoding cycles".

By then the proposed system was completed in VHDL and integrated, showing that for codes with broad block sizes the overhead is low. This is on account of the current majority logic decoding hardware is reused to perform blunder detection and simply some additional control logic is required

III. RESULTS

The system proposed in has been associated with the class of one step MLD EG-LDPC codes. To present the results, the conclusions are displayed first the extent that a theory that is then endorsed by multiplication and more over almost by a hypothetical examination presented in the Appendix. The results obtained could be dense in the going hand in hand with assumption.

ISSN: 2278 – 1323

"Given a word read from a memory secured with one step MLD EG-LDPC codes, and impacted by up to four bit-flips, all errors may be detected in only three decoding cycles".

Note that this theory can't avoid being different from the one made for DS-Ldpc's codes in as in light of current circumstances errors affecting up to five bits were always detected. This is a result of structural differences between DS-LDPC and EG-LDPC codes, which will be ordered in the Appendix. To endorse the above speculation, the EG-LDPC codes considered have been implemented and tested. For codes with small words and impacted by a small number of bit flips, it is practical to make and check all possible lapse unions. As the code size creates and the amount of bit flips stretches, it is no more achievable to completely test all possible combinations. In this way the simulations are done in two courses, by exhaustively checking all blunder merging when it is feasible and by weighing arbitrarily made combos in whatever is left of the cases.
The results for the exhaustive weighs are shown in Table II. These results exhibit the speculation for the codes with more humble word measure (15 and 63). For up to three lapses have been extensively attempted while for simply single and twofold mistake unions have been altogether attempted.

To supplement the aftereffects of the thorough checks for bigger codes and number of errors, reenactments utilizing irregular error examples have been utilized.
TABLE III

UN DETECTABLE ERRORS IN ONEBILLION ERROR COMBINATIONS

N	1 error	2 errors	3 errors	4 errors
15	0	0	0	0
63	0	0	0	0
255	0	0	0	--
1023	0	0	--	--

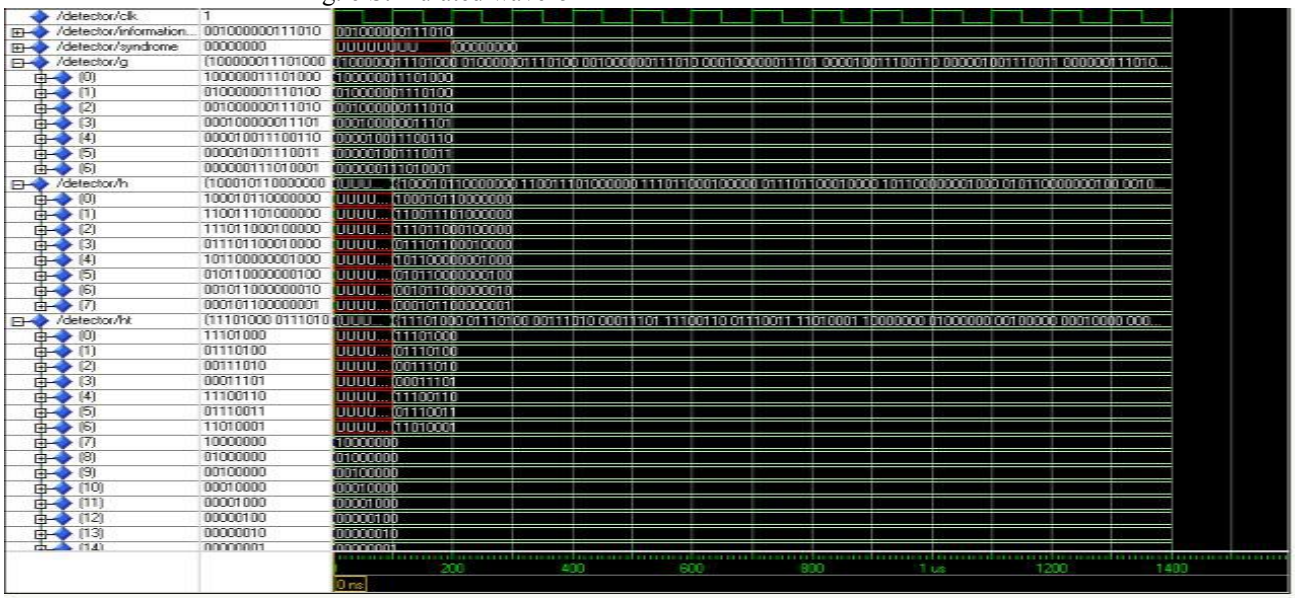
In all the tests, one billion error combinations are tried. The results for errors impacting more than four bits are demonstrated in Table III,

since for errors influencing up to four bits there were no undetected errors. It could be watched that for errors influencing more than four bits there is a little number of mistake fusions that won't be detected in the initial three iterations. This number diminishes with word size furthermore with the quantity of errors. The lessening with the word size might be clarified as takes after: the bigger the word estimate, the bigger the quantity of MLD check equations (see Table I) and along these lines it is all the more farfetched that errors happen in the same equation. With respect to the quantity of errors, a comparable thinking applies: the more errors happen, the bigger the likelihood that an odd number of errors happens in no less than one equation.At last it must be noted that the probabilities of undetected errors are diverse for an even and an odd number of errors as in the recent case, one of the errors must happen in a bit which is not checked by any equation. The simulation results exhibited propose that all errors influencing three and four bits would be recognized in the initial three iterations.

In summary, the initial three iterations will detect all errors influencing four or less bits, and practically every other detectable mistake influencing more bits. This is a marginally more regrettable execution than on account of DS-LDPC For errors influencing a bigger number of bits, there is a little probability of not being detected in those iterations. For vast word sizes, the probabilities are sufficiently little to be worthy in numerous applications. LDPC codes where errors influencing five bits were expansion constantly detected. Be that as it may, the majority logic circuitry is easier for EG-LDPC codes, moreover, EG-LDPC codes have block lengths near a power of two, therefore fitting admirably to the necessities of advanced memory frameworks. This may imply that sometimes it might be more advantageous to utilize an EG-LDPC code and keep a word size perfect with existing outlines (power of two) than utilizing a DS-LDPC code obliging an alternate word size or an abbreviated ver- indication of that code. At the point when utilizing word size which is a power of two, there would be a bit which is not utilized by the EG-LDPC code (see Table I).

EXPERIMENTAL RESULTS

Fig. 6 Stimulated waveform



ISSN: 2278 – 1323

This bit might be utilized for a parity coating all bits in the word that would detect all errors influencing an odd number of bits. All things considered, the configuration utilizing the EG-LDPC would additionally detect all errors influencing five or less bits.

APPLICATIONS

- Data stockpiling devices
- Remote cell system
- Satellite show

Fault secure detectors identifies fault in the encoded bits .If fault is caught the encoding operation is redone. So the middle operation of the FSD is to find the syndrome vector . So first the generator matrix is delivered and its transpose is found. By then the syndrome vector is found. By then the syndrome vector is found by recreating the coded bit with the transpose of parity check matrix. In the blink of an eye if all the bits of syndrome vector are „0" then there is no fault the parity check matrix. In a matter of seconds if all the bits of the syndrome vector are „0" then there is no fault.

IV. CONCLUSION

In this brief, the detection of errors amid the first iterations of serial one step Majority Logic Decoding of EG-LDPC codes has been considered. The target was to diminish the decoding time by stopping the decoding technique when no errors are identified. The simulation results show that all attempted mixes of errors affecting up to four bits are discovered in the introductory three cycles of decoding. These results intensify the ones starting late showed for DS-LDPC codes, making the changed one stage majority logic decoding more appealing for memory applications. The architect now has a greater choice of word lengths and error correction capabilities.

Future work joins extending the hypothetical examination to the cases of three and four errors. All the more generally, choosing the obliged number of cycles to discover errors impacting a given number of bits seems, by all accounts, to be an entrancing issue. A general response for that issue would enable a fine-grained tradeoff between decoding time and error detection capability

REFERENCES

[1] Pedro Reviriego, Juan A. Maestro, and Mark F.Flanagan," Error Detection in Majority Logic Decoding of Euclidean Geometry Low Density Parity Check (EG-LDPC) Codes" IEEE Trans. Very Large Scale Integration (VLSI) Systems, Vol. 21, No. 1, January 2013.

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

[3] 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 approach to hardening sub-100-nm

[4] SRAMs," IEEE Trans. Nucl. Sci., vol. 54, no. 4, pp. 935–945, Aug. 2007.

[5] R. Naseer and J. Draper, "DEC ECC design to improve memory reliability in sub-100 nm Technologies," Proc. IEEE ICECS, pp. 586–589, 2008.

[6] S. Ghosh and P. D. Lincoln, "Dynamic low-density parity check codes for fault-tolerant nano-scale memory," presented at the Foundations Nanosci. (FNANO), Snowbird, Utah, 2007.

[7] S. Ghosh and P. D. Lincoln, "Low-density parity check codes for error correction in nanoscale memory," SRI Computer Science Lab., Menlo Park, CA, Tech. Rep. CSL-0703, 2007.

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

[9] B. Vasic and S. K. Chilappagari, information theoretical framework for analysis and design of nanoscale fault-tolerant memories based on low-density parity-check codes," IEEE Trans. CircuitsSyst. I, Reg. Papers, vol. 54, no. 11, Nov. 2007.

[10] H. Naeimi and A. DeHon, "Fault secure encoder and decoder for nanomemory applications," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 17, no. 4, pp. 473–486, Apr. 2009.

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

ABOUT AUTHORS

1. A.Jayasree received B.Tech. degree from MITS, Madanapalle, Chittoor District, A.P., India. Persuing M.Tech. in GVIC, Madanapalle, Chittoor (D), A.P. She worked as a Asst Prof in MITS, Madanapalle during 2011-12. Her interested areas are Electronic devices & circuits, Communication Engg, Digital Electronics, HDL.

2. Mr.P.Dhaneef Kumar received Diploma degree from G.M.R Polytechnic College, Madanapalle, Chittoor Dist, A.P, India and the B.E degree from SATHYABAMA UNIVERSITY, Chennai, T.N, M.Tech (ES) from BHARATH UNIVERSITY CHENNAI, TN. He worked as Asst. Professor in Vemu Institute of Technology during 2011-12. Presently working as Asst.Prof. in GVIC, Madanapalle Chittoor (D) A.P. His Interested areas are Digital Electronics, Embedded systems and Microprocessors.

ISSN: 2278 – 1323