ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 9, September 2015

Error Detection in Majority Argumentation Decryption of Parabolic Geometry Denseness Parity Analysis Codes

A.FeemaAnjum, Mekala Srinivasulu

M. Tech, Dept. of ECE, Swetha Institute of Technology& Science, Tirupati, AP, India.

Assistant Professor, Dept. of ECE, Swetha Institute of Technology& Science, Tirupati, AP, India.

ABSTRACT: In a recent paper, a process used to be proposed to accelerate the bulk original feel decoding of trade set low density parity determine codes. That is valuable as majority common sense decoding may also be implemented serially with handy hardware nonetheless requires a gigantic decoding time. For memory applica-tions, this raises the memory entry time. The approach detects whether or not a phrase has error within the first iterations of majority customary experience decoding, and when there don't seem to be any mistakes the decoding ends without finishing the rest of the iterations. Given that most phrases in a reminiscence can also be error-free, the traditional decoding time is typically lowered. In this transient, we be taught the applying of a an same system to a class of Euclidean geometry low density parity investigate (EG-LDPC) codes which probably one step majority logic decodable. The outcome obtained show off that the approach can also be mighty for EG-LDPC codes. Exten-sive simulation outcome are given to effortlessly estimate the possibility of error detection for one-of-a-sort code sizes and numbers of mistakes.

KEYWORDS: Error correction codes; Euclidean geometry low-density parity verify (EG-LDPC) codes; majority Common sense decoding; Reminiscence;

I. INTRODUCTION

Error correction codes are most commonly used to preserve reminiscences from so-called smooth blunders, which alternate the logical value of reminiscence cells without unsafe the circuit . As science scales, reminiscence gadgets come to be better and extra strong error correction codes are wanted . To this finish, the usage of more evolved codes has been not too long ago proposed . These codes can correct a better number of errors, but in most cases require complex decoders. To prevent a high decoding complexity, the usage of one step majority logic decodable codes was once first proposed for memory purposes. One step majority good judgment decoding may also be applied serially with very simple circuitry , but requires lengthy decoding occasions. In a reminiscence, this may expand the entry time which is an most important method parameter. Only some lessons of codes will also be decoded making use of one step majority good judgment decoding [9]. Amongst these are some Euclidean geometry low density parity determine (EG-LDPC) codes which have been utilized in [4], and change set low density parity check (DS-LDPC) codes .

A process was lately proposed in [10] to accelerate a serial implementation of majority good judgment decoding of DS-LDPC codes. The proposal behind the system is to make use of the primary iterations of majority logic decoding to notice if the phrase being decoded includes blunders. If there aren't any mistakes, then decoding can also be stopped with out finishing the remainder iterations, hence extensively decreasing the decoding time.

For a code with block length , majority common sense decoding (when carried out serially) requires iterations, so that because the code size grows, so does the decoding time. In the proposed technique, simplest the primary three iterations are used to observe error, thereby reaching a gigantic speed increase when is giant. It was once shown that for DS-LDPC codes, all error mixtures of as much as five errors will also be detected in the first three iterations. Also, error affecting more than 5 bits were detected with a probability very virtually one. The probability of undetected error used to be additionally found to scale back as the code block length extended. For a billion error patterns only some mistakes (or normally none) were undetected. This may be ample for some functions. An extra skills of the proposed approach is that it requires little or no further circuitry because the decoding circuitry is also used for error detection. For instance, it was shown in [10] that the additional discipline required to implement the scheme used to be simplest round 1% for massive phrase dimension.

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 9, September 2015

TABLE I
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

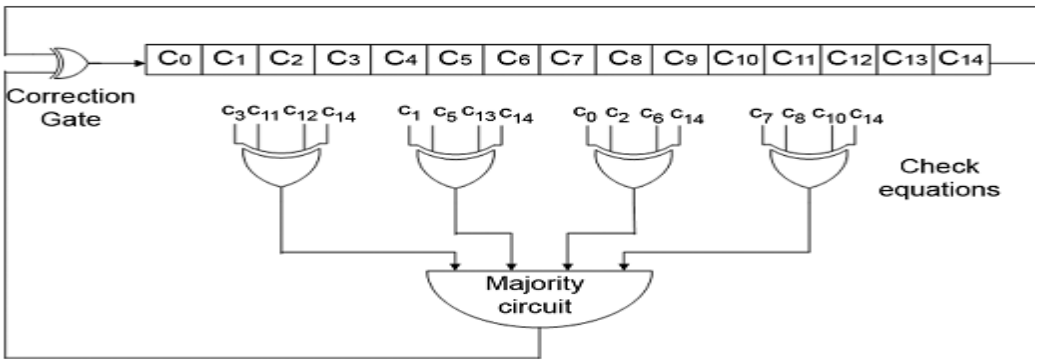

Fig. 1. Serial one-step majority logic decoder for the (15,7) EG-LPDC code.

The approach proposed in [10] relies on the properties of DS-LDPC codes and thus it is not straight applicable to different code classes. In the following, a similar strategy for EG-LDPC codes is presented. This supplies preliminaries on EG-LDPC codes, majority logic decoding and the system proposed presents the outcome of applying the approach to EG-LDPC codes, comprising simulation outcome and a speculation established on those results.

Reminiscence cells had been included from smooth error for more than a decade; as a result of the expand in gentle error rate in good judgment circuits, the encoder and decoder circuitry round the reminiscence blocks have come to be inclined to soft mistakes as good and need to even be blanketed. We introduce a new technique to design fault -at ease encoder and decoder circuitry for reminiscence designs.

Hamming codes are most often used in at present's memory techniques to proper single error and notice double errors in any memory phrase. In these memory architectures, handiest error in the memory phrases are tolerated and there is no practise to tolerate blunders within the aiding common sense (i.E. Encoder and corrector).Nevertheless combinational good judgment has already began displaying susceptibility to delicate error, and for that reason the encoder and decoder (corrector) items will no longer be immune from the transient faults. Accordingly, protecting the memory procedure help common sense implementation is extra essential. Here we proposed a fault tolerant memory system that tolerates multiple errors in each reminiscence phrase as good as more than one blunders within the encoder and corrector units.

We illustrate utilizing Euclidean Geometry codes and Projective Geometry codes to design the fault tolerant memory procedure, because of their good-suited characteristics for this utility. Low density parity-check (LDPC) codes had been first found out by means of Gallager in the early 1960s and have just lately been rediscovered and generalized .It has been shown that these codes gain a terrific efficiency with iterative decoding that is very almost the Shannon limit[3]. For this reason, these codes have end up robust rivals to faster codes for error manage in lots of verbal exchange and digital storage systems the place excessive reliability is required. LDPC codes can be developed utilizing random or deterministic strategies.In this paper, we focus on a class of LDPC codes known as Euclidean Geometric (EG) LDPC codes, which are constructed deterministically utilising the features and lines of a Euclidean geometry [1, 6].

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 9, September 2015

II. LITARETURE SURVEY

Finite geometries had been used to derive many error-correcting codes [9], [11]. One example are EG-LDPC codes which might be established on the constitution of Euclidean geometries over a Galois discipline. Amongst EG-LDPC codes there's a subclass of codes that's one step majority good judgment decodable (MLD) [9]. Codes in this subclass are additionally cyclic. The parameters for a few of these codes are given in table I, where is the block size, the quantity of information bits, the quantity of MLD determine equations and the number of errors that the code can proper utilizing one step MLD.One step MLD will also be applied serially utilising the scheme in Fig. 1 which corresponds to the decoder for the EG-LDPC code with. First the info block is loaded into the registers. Then the assess equations are computed and if a majority of them has a worth of one, the final bit is inverted. Then all bits are cyclically shifted. This set of operations constitutes a single iteration: after iterations, the bits are within the equal role where they had been loaded. In the system, each and every bit could also be corrected best as soon as. As can be noticeable, the decoding circuitry is simple, but it requires a protracted decoding time if is tremendous. The determine equations ought to have the following properties (see [9] for extra important points).

Table II
UNDETECTED blunders 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 comprise the variable whose worth is stored in the last register (the one marked as).
2) The rest of the registers are incorporated in at most one of the vital assess equations.

If error will also be detected within the first few iterations of MLD, then every time no mistakes are detected in those iterations, the decoding can be stopped with out completing the leisure of the iterations. In the first generation, error shall be detected when as a minimum one of the verify equations is suffering from an ordinary number of bits in error. In the second new release, as bits are cyclically shifted with the aid of one function, error will affect different equations such that some errors undetected in the first new release will be detected. As iterations improve, all detectable blunders will finally be detected

In [10] it was proven that for DS-LDPC codes most mistakes can also be detected within the first three iterations of MLD. Headquartered on simulation results and on a theoretical proof for the case of two mistakes, the next speculation used to be made. “Given a word read from a memory covered with DS-LDPC codes, and littered with as much as 5 bit-flips, all error can be detected in only three decoding cycles”. Then the proposed system was carried out in VHDL and synthesized, displaying that for codes with tremendous block sizes the overhead is low. This is considering that the present majority common sense decoding circuitry is reused to participate in error detection and only some further control logic is wanted.

III.IMPLEMENTATION

The method proposed in [10] has been applied to the category of 1 step MLD EG-LDPC codes. To reward the results, the conclusions are pre-sented first in terms of a speculation that is then validated by using simulation and in addition in part with the aid of a theoretical analysis.The results got can also be summarized in the following speculation. “Given a word learn from a reminiscence covered with one step MLD EG-LDPC codes, and plagued by as much as four bit-flips, all error can be detected in only three decoding cycles”.

Word that this speculation is different from the one made for DS-LDPCs codes in [10] as if so blunders affecting up to five bits were consistently detected. This is because of structural differences be-tween DS-LDPC and EG-LDPC codes, in order to be precise in the Appendix. To validate the above hypothesis, the EG-LDPC codes regarded have been carried out and demonstrated.

For codes with small words and plagued by a small quantity of bit flips, it is realistic to generate and verify all viable error combina-tions. As the code size grows and the number of bit flips raises, it's no longer feasible to exhaustively test all viable mixtures. There-fore the simulations are completed in two approaches, by way of

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 9, September 2015

exhaustively checking all error combos when it's possible and by checking randomly gen-erated combinations within the relaxation of the instances.

The results for the exhaustive exams are shown in desk II. These results show the speculation for the codes with smaller word size (15 and sixty three). For as much as three blunders were exhaustively tested whilst for best single and double error mixtures have been exhaustively confirmed.

A. Majority logic Decoder/Detector (MLDD)

To be able to overcome the obstacle of MLD system, majority logic decoder/detector used to be proposed, where the bulk logic decoder itself act as a fault detector. By and large, the decoding algorithm is still the identical as the majority logic decoder. The difference is that as an alternative of decoding all codeword bits, the MLDD process stops intermediately within the 1/3 cycle, which can competent to notice as much as 5 bit flips in three decoding cycles. So the number of decoding cycles can also be reduced to get expanded performance. The schematic of majority common sense decoder/detector is illustrated in figure3.

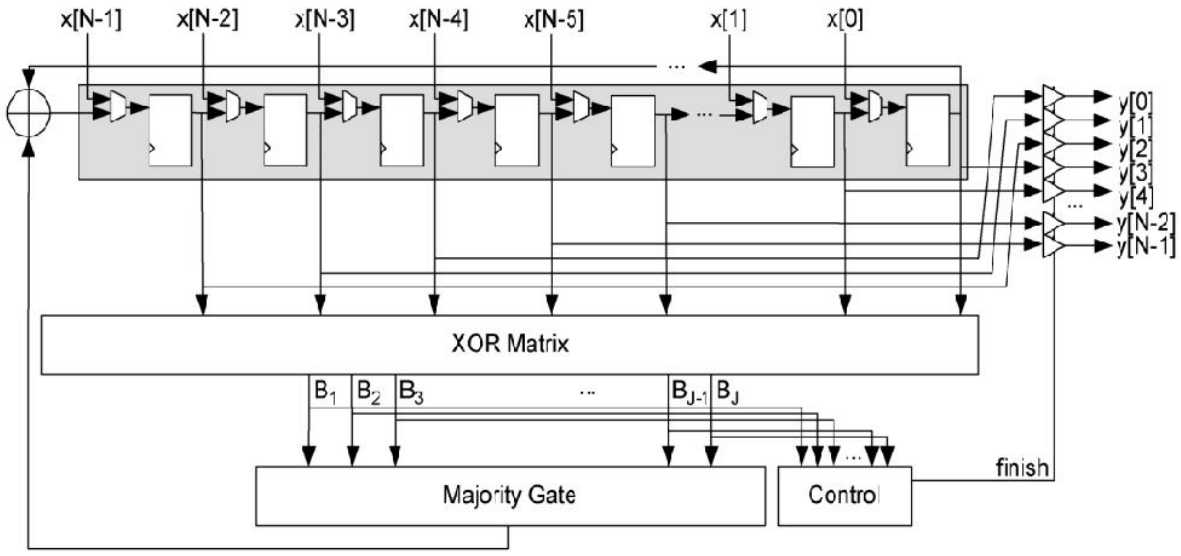

Fig 3: Schematic of Majority Logic Decoder/Detector (MLDD)

At the start the code phrase is saved into the cyclic shift register and it shifted by means of the entire faucets. The intermediate values in each and every faucet are given to the XOR matrix to perform the checksum equations. The ensuing sums are then forwarded to the bulk gate for evaluating its correctness. If the quantity of 1"s obtained is better than the quantity of zero"s which might imply that the current bit under decoding is wrong, so it transfer on the decoding procedure. Or else, the bit below decoding would be right and no additional operations would be needed on it. Decoding procedure involving the operation of the content material of the registers is circled and the above method is repeated and it stops intermediately within the third cycle. If in the first three cycles of the decoding method, the evaluation of the XOR matrix for all is "0," the code phrase is determined to be error-free and forwarded immediately to the output. If the error includes in any of the three cycles as a minimum a "1," it could proceed the entire decoding method with the intention to do away with the blunders. Subsequently, the parity check sums should be zero if the code phrase has been correctly decoded. In conclusion the MLDD approach is used to observe the five bit error and correct four bit blunders conveniently. If the code phrase include greater than five bit error, it produces the output but it surely didn't exhibit the mistakes provided within the input. This type of error is known as the silent information error. Main issue of this method is didn't detecting the silent knowledge error and it consuming the discipline of the majority gate. The schematic for this memory procedure is proven in determine 5. It is rather much like the one shown in fig. 1; additionally the manage unit was delivered within the MLDD module to control the decoding process (to become aware of the error).

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 9, September 2015

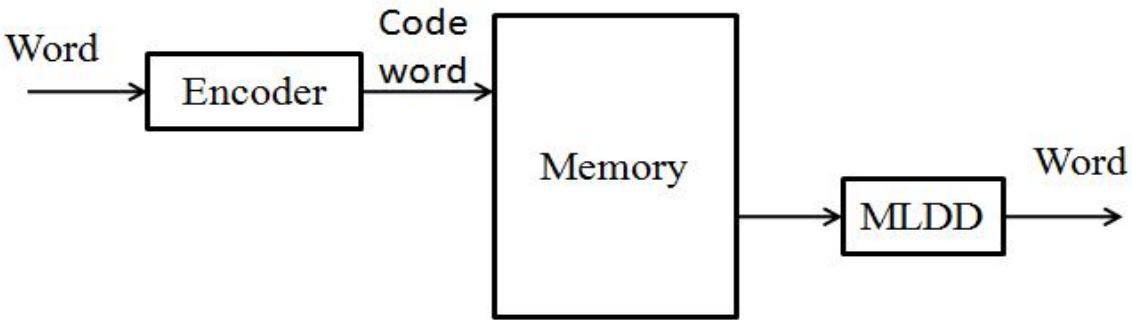

Fig 4: Schematic of memory system with MLDD

The data words are originally encoded and then the codeword is stored in the memory. When the memory is learn, the codeword is then fed through the enhanced MLDD earlier than sent to the output for extra processing. The code word comprises message bits and parity or redundant bits. The code efficiency is defined as the ratio of message bits to the number of transmitted bits per block. The silent information error detection using enhanced MLDD algorithm performs the decoding as in the MLDD with some changes. When the MLDD having greater than 5 mistakes can be detected and corrected by the enhanced MLDD method. The MLDD is used the control unit for detecting the error. If it has any error on this generation it's going to be perform with the modified algorithm is illustrated in figure 5. It is used to hinder silent knowledge corruption of the MLDD output. This would develop the error detection capabilities on the price of the error-correction capabilities.

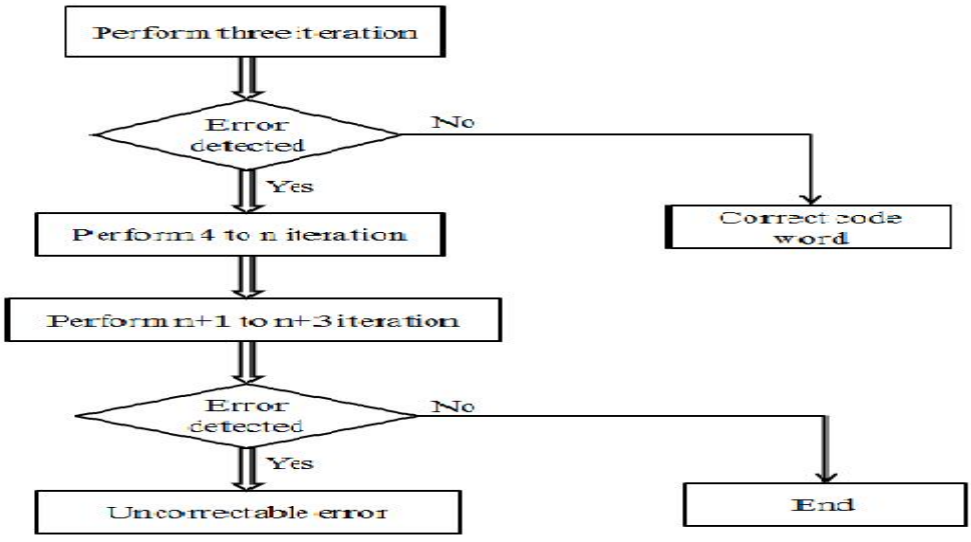

Fig 5: Enhanced MLDD algorithm

B. Sorting network

A sorting network is an abstract mathematical model of a community of wires and comparator modules that's used to type a chain of numbers. Every comparator connects two wires and types the values by using outputting the smaller value to at least one wire, and the larger to the opposite. The major change between sorting networks and evaluation sorting algorithms is that with a sorting network the sequence of comparisons is ready in advance, whatever

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 9, September 2015

the end result of earlier comparisons. This independence of evaluation sequences is priceless for parallel execution of the algorithms.

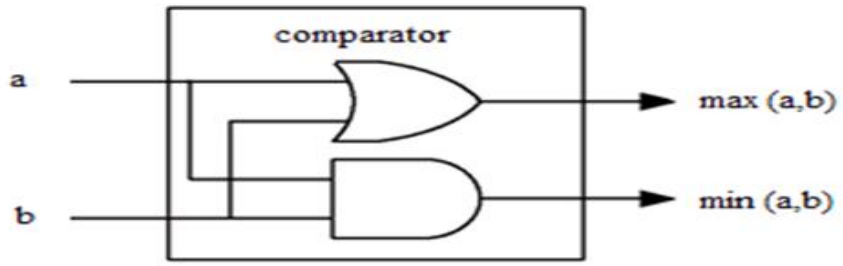

Fig 6: Comparator circuit

IV. SIMULATION RESULTS

The area utilization and energy consumption is calculated. The modelsim output shows (fig 9) the waveform of one step MLD and the decoding time. Utilizing Xilinx, the power consumption and area utilization of one step MLD are measured. The simulation outcome shows that the one step MLD would take 15cycles to decode a codeword of 15-bits. The remaining part (MLDD and enhanced MLDD) of this paper is under development.

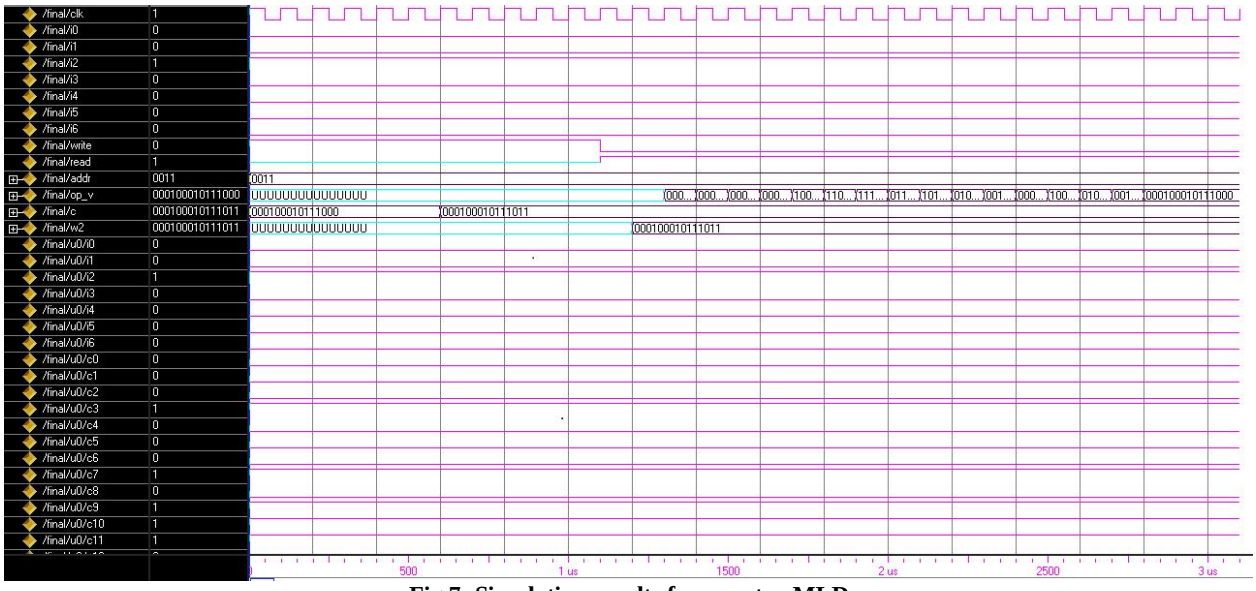

Fig 7: Simulation results for one step MLD

In summary, the primary three iterations will realize all mistakes affecting 4 or fewer bits, and close to each different detectable error affecting more bits. This can be a fairly worse efficiency than in the case of DS-LDPC codes [10] where blunders affecting 5 bits have been addition-ally consistently detected. Nevertheless, the bulk logic circuitry is less difficult for EG-LDPC codes, as the quantity of equations is a vigor of two and an approach centered on sorting networks proposed in [8] can be used to decrease the price of the majority good judgment balloting. Moreover, EG-LDPC codes have block lengths nearly a power of two, therefore becoming well to the specifications of today's

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 9, September 2015

reminiscence programs. This may increasingly imply that in some instances it could be more handy to make use of an EG-LDPC code and keep a phrase measurement suitable with present designs (energy of two) than using a DS-LDPC code requiring one other phrase size or a shortened ver-sion of that code. When using phrase measurement which is a vigour of two, there would be a bit which is not used by the EG-LDPC code (see table I). This bit can be used for a parity masking all bits within the phrase that may become aware of all blunders affecting an odd number of bits. If that's the case, the design making use of the EG-LDPC would additionally notice all errors affecting five or fewer bits.

V. CONCLUSION & FEATURE WORK

On this transient, the detection of errors for the period of the first iterations of se-rial one step Majority good judgment Decoding of EG-LDPC codes has been studied. The target was to scale back the decoding time by means of stopping the decoding process when no blunders are detected. The simulation outcome show that every one confirmed mixtures of error affecting up to 4 bits are detected within the first three iterations of decoding. These results extend the ones not too long ago awarded for DS-LDPC codes, making the modified one step majority logic decoding extra appealing for memory applica-tions. The designer now has a larger alternative of phrase lengths and blunder correction capabilities.

Future work involves extending the theoretical analysis to the circumstances of three and 4 blunders. More most of the time, picking the specified number of iterations to detect mistakes affecting a given quantity of bits seems to be an intriguing concern. A general technique to that drawback would enable a quality-grained tradeoff between decoding time and mistake detection capacity.

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. H. Naeimi and A. DeHon, "Fault secure encoder and decoder for memory applications," in Proc. IEEE Int. Symp. Defect Fault Toler. VLSI Syst., 2007, pp. 409–417.
7. Vasic and S. K. Chilappagari, "An information theoretical frame-work for analysis and design of nanoscale fault-tolerant memories based 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 for nanomemory 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. Englewood Cliffs, NJ: Prentice-Hall, 2004.
10. S. Liu, P. Reviriego, and J. Maestro, "Efficient majority logic fault 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.