A Survey on Design of MLDD for Error Detection and Correction

Ms. Minal M. Jaiswal **Prof. Ashish E. Bhande**

Abstract— **Memory applications generally require large errors to be corrected. So majority logic decoder (MLD) is the suitable choice for this. Another methodology of decodable logic is Majority Logic Decoder/Detector that reduces not solely the coding time however additionally access time similarly as space utilization. Euclidean Low-Density Parity-Check (EG-LDPC) codes are used for error correction, as a result of their fault-secure detector capability. EG-LDPC codes are wont to avoid high coding complexness. The application of an analogous technique to a category of Euclidean geometry low density parity check (EG-LDPC) codes that are one step majority logic decodable. The obtained results show that the strategy is additionally effective for EG-LDPC codes. The planned style of error detection and correction can be coded using VHDL, verified and synthesize on Modelsim and Xilinx FPGA severally.**

Index Terms— **Error correction, Fault detection, Majority logic decoder/detector, Memory, Serial one step MLD, Soft error, Sorting network.**

I. INTRODUCTION

Now a day, electronic communication is important a part of life and lots of data has being transferred. Several communication channels are subject to channel noise, and so errors are also introduced throughout transmission from the supply to a receiver. There are numerous ways that of hacking, once the interloper modifies the data while communication. Not solely to safeguard the confidentiality of the data however additionally to retain the correctness of the data, secure communication is extremely vital. There are numerous strategies of implementing the secure communication. Each technique has its own blessings and downsides. This project is use for the development on most of the accessible strategies for secure communication.

For reliable communication, errors should be detected further as corrected. Some multi error bit correction codes area unit BCH codes, Reed Solomon codes, however within which the algorithmic rule is extremely troublesome. These codes will correct an outsized variety of errors, however would like complicated decoders. Among the error correction codes, cyclic block codes have higher error detection capability further as low decoding complexness which area unit majority logic (ML) decodable. A low-density parity-check (LDPC) code may be a linear error correcting code, wont to avoid high decoding complexness. one specific style of low density parity check codes, specifically Euclidean Geometry-LDPC codes area unit used thanks to not solely their fault secure detector capability however additionally higher reliability and lower space overhead.

To protect the recollections from questionable soft errors error correction codes are usually used that modification the logical price of memory cells while not damaging the circuit. As technology scales memory devices become larger and additional powerful error correction codes area unit require. To the present finish recently projected the utilization of additional advanced codes. These codes will correct a bigger variety of errors, however need complicated decoders. The utilization of 1 step majority logic decodable codes was initially projected for memory applications, to avoid a high decoding complexness. One step majority logic decoder is enforced serially with terribly easy electronic equipment, however needed long decoding times. This may increase the interval in memory that is a vital system parameter.

II. LITERATURE REVIEW

[1]. Pedro Reviriego, Juan A. Maestro, and Mark F. Flanagan presented Error Detection in Majority Logic Decoding of Euclidean Geometry Low Density Parity Check (EG-LDPC) Codes. A technique was planned to accelerate the logic decryption of varied set low density parity check codes. Within the serial one step Majority Logic Decoder of EG-LDPC codes has been studied for the detection of errors throughout the primary iteration. The target was to attenuate the decoding time by stopping the decoding process once no errors are detected. The obtained simulation results show that all the tested mixtures of errors touching up to four bits are detected within the 1st 3 iterations of decoder. These results are extend those recently conferred for DS-LDPC codes, for memory application the modified one step majority logic decoder a lot of engaging. The designer currently incorporates a larger selection of word lengths moreover as error correction capabilities.

[2]. P. Kalai Mani, V. Vishnu Prasath, presented Majority Logic Decoding of Euclidean Geometry Low Density Parity Check (EG-LDPC) Codes. Error detection in memory applications was planned to accelerate the majority logic decoding of varied set low density parity check codes. LDPC is helpful as majority logic decoders are often enforced serially with easy hardware however an oversized decoding time is needed. For memory applications, this increases the operation time. This technique detects whether or not a word has errors within the 1st iterations of majority logic decoder, if there are not any errors then the decoding method is stop while not finishing the remainder of the iterations. So most words in memory are going to be error free, and then the common decoding time is greatly reduced. During this transient, the application of similar technique to a class of Euclidean geometry low density parity check codes that are one step majority logic decidable. The obtained results show that the method is additionally effective for EG-LDPC codes.

[3]. M. Pramodh Kumar, S. Murali Mohan, presented Serial one-step majority logic decoder for EG-LDPC code. During

this transient, the detection of errors throughout the primary iterations of serial one step Majority Logic decoder of EG-LDPC codes has been studied. The target was to attenuate the decoding time by stopping the decoding method once no errors are detected. The obtained simulation results show that all the tested mixtures of errors touching up to four bits are detected within the 1st 3 iterations of decoding. These result was extend those recently conferred for DS-LDPC codes, for memory application the modified one step majority logic decoder a lot of attractive. The designer currently incorporates a larger selection of word lengths moreover as error correction capabilities.

[4]. Adline Priya, presented Low Power Error Correcting Codes Using Majority Logic Decoding. Moreover, the decoder designs for LDPC codes are designed. And therefore the simulation results for encoder, decoder, memory and detector are obtained. And additionally the majority logic decoder is enforced serially.

[5]. Senbagapriya. S. presented An Efficient Enhanced Majority Logic Fault Detection with Euclidean Geometry Low Density Parity Check (EG-LDPC) Codes for Memory Applications. During this paper, the detection of errors throughout 1st iterations of serial one step Majority Logic decoder of EG-LDPC codes has been conferred. The obtained simulation results show that to decipher a codeword of fifteen-bits the one step MLD would takes 15 cycles, which might be excessive for applications. The MLD style needs tiny space however giant decryption time is needed and which may be ready to observe 2 or few errors. Hence, operation time will increase. Another technique, known as MLDD will observe up to 5 bit-flips and consumes the area of majority gate. These styles are under progress.

III. General Schematic Of MLDD

Fig.3 shows the generic schematic of a memory system with MLDD. It consists of Encoder, Memory and MLDD.

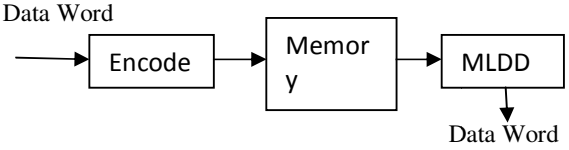

Fig.3 General Schematic of MLDD

First the data words are encoded and which is stored in the memory. When the memory is read then the codeword is fed through the MLDD before sent to the output for the processing. And this decoding process, the data word is corrected from all bit flips that it might have suffered while being stored in the memory. The advantages of this method are as follows

- High speed operation.
- Low power consumption.
- Execution time is reduced.

IV. Proposed Work

The block diagram of proposed system is shown in Fig.4 is described as follows. The block diagram consists of following blocks namely Encoder, Corrector, Detector and Memory block.

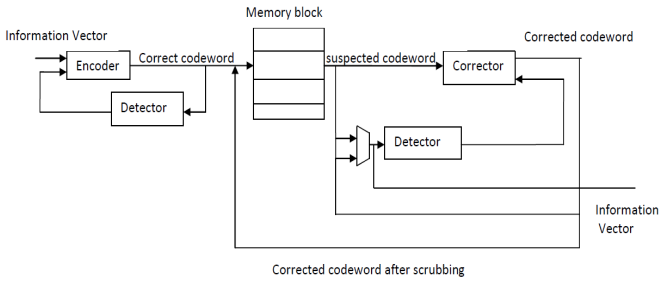

Fig.4 Block diagram of overall MLDD

A. *Encoder*

The information bits are fed into the encoder to encode the information vector. This section provides a brief introduction on linear block ECC's.

For linear codes the encoding operation essentially performs the following vector-matrix Multiplication.

$$C = I \times G$$

Let I = (i0, i1... ik−1) be k-bit information vector that will be encoded into n-bit codeword, C= (c0, c1... c n− 1) and G is a k × n generator matrix.

A code is a systematic code if any codeword consists of the original k-bit information vector followed by (n – k) parity-bits. With this definition, the generator matrix of a systematic code must have the following structure.

$$G = [I: X]$$

Where, I is a k × k identity matrix and X is a k× (n−k) matrix that generates the parity-bits.

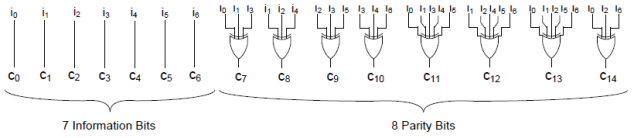

Fig. 4.1 Structure of an Encoder Circuit

Figure 4.1 shows the encoder circuit to compute the parity bits of the (15, 7, 5) EG-LDPC code. In this figure I = (i0, i1…i6) is the information vector and will be copied to C= (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.

B. *Fault secure detector*

The fault secure detector of the encoder verifies the validity of the encoded vector. If the detector detects any error, the encoding operation must be redone to generate the correct codeword. The codeword is then stored in the memory.

A code is said to be cyclic code if for any codeword c, all the cyclic shifts of C is still a valid codeword. A code is cyclic if the rows of its parity-check matrix and generator matrix are the cyclic shifts of their first rows. The checking or detecting operation is the following vector-matrix multiplication.

$$S = C \times H^T$$

Where, H is an (n−k) ×n Parity-Check matrix. The (n−k) bit vector S is called syndrome vector.

C. *Memory*

Data bits stay in memory for a number of cycles and, during this period, each memory bit can be upset by a transient fault with certain probability. Therefore, transient errors accumulate in the memory words over time. In order to avoid accumulation of too many errors in any memory word that surpasses the code correction capability, the system must perform memory *scrubbing*. Memory scrubbing is the process of periodically reading memory words from the memory, correcting any potential errors, and writing them back into the memory. To perform the periodic scrubbing operation, the normal memory access operation is stopped and the memory performs the scrub operation.

D. *Serial corrector*

During memory access operation, the stored code words will be accessed from the memory unit. Code words are susceptible to transient faults while they are stored in the memory. Therefore a corrector unit is designed to correct potential errors in the retrieved code words. In our design all the memory words pass through the corrector and any potential error in the memory words will be corrected. Similar to the encoder unit, a fault secure detector monitors the operation of the corrector unit.

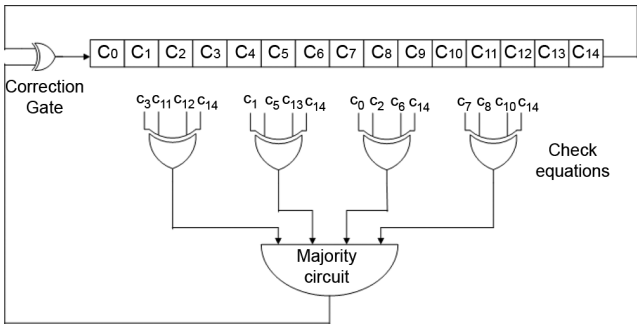

Fig.4.4 Serial one step majority logic decoder

To detect the errors serially the MLDD technique uses Serial One Step Majority Logic Decoder. The serial one step majority logic decoder is depicted in Fig.3.4. In this decoder 15 bit data is first stored in the cyclic shift register. Then the inputs are assigned to the XOR gates. Since there is 15 bit data the XOR gates required are four. The bit to be detected should be given as one of the inputs for all the XOR gates. The outputs of the XOR gates are the check sum equations. The check sum equations consist of binary data. Then the Majority circuit outputs the data which is in major number. If the output of the majority circuit is '1' then the corresponding bit has the error else the bit is error free.

The output of the Majority circuit is given as one of the input to the correction gate. Another input to the correction gate is the bit which is under test. So the corrected bit is stored into the shift register where first cyclic shift occurs. This entire process is called as one iteration. Likewise three iterations are processed. Maximum number of errors is detected within these three iterations.

V. SIMULATION RESULTS

The Coding for our proposed methodology was written using VHDL coding and simulated using Modelsim software. The simulation of overall MLDD system is shown in following figures.

Fig.5.1 (a) Simulation of encoder

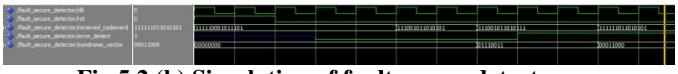

Fig.5.2 (b) Simulation of fault secure detector

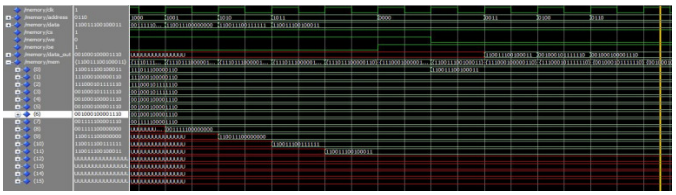

Fig.5.3 (c) Simulation of memory

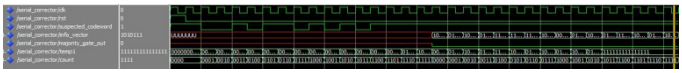

Fig.5.4 (c) Simulation of serial corrector without error

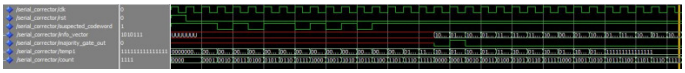

Fig.5.4 (d) Simulation of serial corrector with error

CONCLUSION

In this paper majority logic decoder/detector are often find the amount of errors and proper it. Fault secure detector are often detect error and serial one step majority logic decoder are often correct these errors. MLDD have the potential of reduces the area of majority gate by mistreatment sorting network. This method is use in an exceedingly communication system like OFDM and alternative communication system. In future work is that, we have a tendency to used parallel pipeline corrector in MLDD system and therefore the results acquire by mistreatment Xilinx and Modelsim software. The advantage is that the amount of cycles are reduces thus time is reduces and speed is increases.

ACKNOWLEDGMENT

We are solemnly grateful to our HVPM college of Engineering and Technology to support and my guide Prof. Ashish Bhande and alternative college conjointly associates of ENTC department who are directly & indirectly helped me for this paper.

REFERENCES

[1] P. Kalai Mani, V. Vishnu Prasath, "Majority Logic Decoding Of Euclidean Geometry Low Density Parity Check (EG-LDPC) Code," International Journal of Innovative Research in Computer and Communication Engineering, Vol.2, Special Issue 1, March 2014.

[2] M. Pramodh Kumar, S. Murali Mohan, "Serial one-step majority logic decoder for EG-LDPC code," IJISET - International Journal of Innovative Science, Engineering & Technology, Vol. 1 Issue 6, August 2014.

[3] A. Adline Priya, "Low Power Error Correcting Codes Using Majority Logic Decoding," International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622, March 2014.

[4] 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.

[5] Senbagapriya.S, "An Efficient Enhanced Majority Logic Fault Detection with Euclidean Geometry Low Density Parity Check (EG-LDPC) Codes for Memory Applications," International Journal of Engineering Science and Innovative Technology (IJESIT) Volume 2, Issue 6, November 2013.

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

[7] Vijaykumar K., Dr. K. Ashok Babu, "Fault tolerant nano-memory with fault secure encoder and decoder," Vol. 3 No. 1 Jan 2011.

[8] 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.