INTERNATIONAL JOURNAL OF PURE AND APPLIED RESEARCH IN ENGINEERING AND TECHNOLOGY

A PATH FOR HORIZING YOUR INNOVATIVE WORK

FPGA IMPLEMENTATION OF EUCLIDEAN GEOMETRY LOW DENSITY PARITY CHECK (EG-LDPC) CODES FOR ERROR CHECKING IN MAJORITY LOGIC DECODING

DARSHANKUMAR K. WAGH[1], S. S. THORAT[2]

1. Electronics Engineering Department, M. Tech. In Electronics System and Communication, Government College of Engineering Amravati.
2. Electronics Engineering Department, Assistant Professor, Government College of Engineering Amravati.

Accepted Date: 05/03/2015; Published Date: 01/05/2015

Abstract: In this paper, a technique was proposed to protect memory cells, which are more susceptible to soft errors. These memory cells are to be protected with effective error correction codes. Majority logic decodable codes are suitable for memory applications because of their capability to correct large number of errors. Another method of decodable logic is Majority Logic Decoder/Detector which reduces not only the decoding time but also memory access time as well as area utilization. Euclidean Geometry Low-Density Parity-Check (EG-LDPC) codes are used for error correction, because of their fault-secure detector capability. EG-LDPC codes are used to avoid high decoding complexity. The application of a similar technique to a class of Euclidean geometry low density parity check (EG-LDPC) codes that are one step majority logic decodable. The proposed design of error detection and correction will coded using VHDL, be verified and synthesized on Modelsim and Quartus II software respectively.

Keywords: Error Correction Codes, Euclidean Geometry Low-Density Parity Check (EG-LDPC) Codes, Majority Logic Decoding, and Memory.

Corresponding Author: MR. DARSHANKUMAR K. WAGH

Access Online On:

www.ijpret.com

How to Cite This Article:

Darshankumar K. Wagh, IJPRET, 2015; Volume 3 (9): 1142-1148

PAPER-QR CODE

INTRODUCTION

Memory is the basic resource in every digital system and therefore for reliable and secure performance the protection of memories is very essential in the present day electronic system plan. Now a days the Single Event Upsets (SEU) which is also called 'soft errors' occurs when radiation event causes enough charge disturbance to change the logical value of memory cells, registers without damaging the circuit or device. The error is called 'soft' because it flips the data state of the memory cell without damaging the circuit. More than a single bit may be affected if the radiation event is of high energy which creates the Multi Bit Upsets (MBU).

As soft errors change the logical value of the memory cells without damaging the circuits, errors must be detected and corrected for reliable communications. As technology emerges, memory devices become larger. It requires powerful error correction codes to correct the errors. Due to this the use of more advanced codes has been recently proposed. These types of codes can correct a larger number of errors, but they require mixed decoders. To avoid this difficult decoding method, the different set codes and one step majority logic decodable codes were first proposed for memory applications, decoding of low density parity check codes is done using majority logic decoding.

One step majority logic decoding can be implemented serially with very simple circuit, but requires long decoding times and it increases the delay. In a memory, this would increase the access time which is an important system constraint. But for few modules of codes is decoded using one step majority decoding. Among those are some Euclidean geometry low density parity check (EG-LDPC) codes which were used.

A method was recently proposed to accelerate a serial implementation of majority logic decoding of EG-LDPC codes. The design behind the method is to use the first iterations of majority logic decoding to detect if the word decoded contains errors. If it is found there are no errors, then decoding process can be stopped. Decoding time is much more reduced because of stopping the iterations before fully completing. For a code with block length N, majority logic decoding which is implemented serially requires N iterations, so that the sizes of the code increase, so the decoding time also increase. In the proposed system, the errors are detected in parallel and in pipelining method. The detection of errors requires only single iteration where most of the errors are detected. The delay time is reduced for this proposed method is low compared to the prior technique.

I. EXISTING MLD TECHNIQUES

Majority logic decoding is a simple and effective scheme for decoding certain class of block codes. Majority logic decoding codes are cyclic codes. These codes are constructed based on finite geometrics such as Euclidean geometrics and projective geometrics. Majority Logic Decoder (MLD) technique is generally based on number of parity check equations

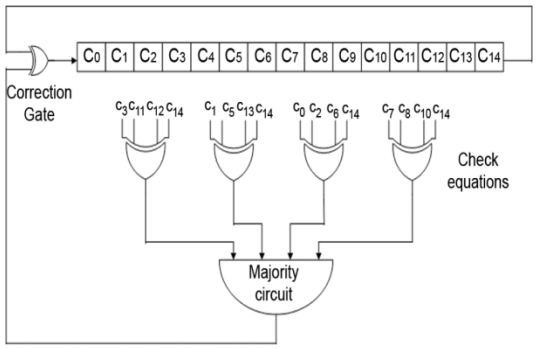

Fig.1: One step Majority Logic Decoder

(15,7) EG-LDPC codes

Initially the input is stored into the cyclic shift register and it shifted through all the bits. The intermediate values in each bit are given to the XOR matrix which is used to perform the checksum equations. The resulting sums are then forwarded to the majority gate for evaluating its correctness. If the number of 1's received is greater than the number of 0's which would mean that the current bit under decoding is wrong, so it move on the decoding process. It is used to produce the accurate result of the MLD.

II. PROPOSED SYSTEM

There are various error detection techniques can be used to avoid the soft error. One of the methods is majority logic decoder which used to detect and correct the error in simple way. This method uses the First iteration of majority logic decoding to detect the error present in the word.

If there are no errors, then the decoding process can be stopped without completing the remaining iterations. The main reason for using Majority Logic Decoding (MLD) is that it is very easy to implement and has a low complexity. The major drawback of this method is, it increases the average latency of the decoding process because it depends on the size of the code, thus increases the memory access Time. Another method is syndrome fault detector which is an XOR

matrix that calculates the syndrome based on the parity check matrix. This method results in a quite complex module, with a large amount of hardware and power consumption in the system.

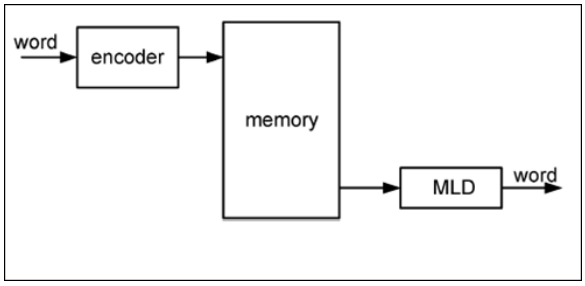

Fig.2: Memory System Schematic With MLD

A generic schematic of a memory system is depicted in fig.2 for the usage of an ML decoder. The general schematic of memory system implemented with majority logic decoder is depicted in fig.3. Initially the data words are encoded and then stored in the memory. When the memory is read, the code word is then fed through the Majority Logic Decoder (MLD) before sent to the output. In this decoding process, the code word is corrected from all bit flips it might have suffered while being stored in the memory.

A. **Encoder :**

In encoder the input bit of 7 bits of information is encoded. It produces encoded bits in which it has information as well as parity check bits.

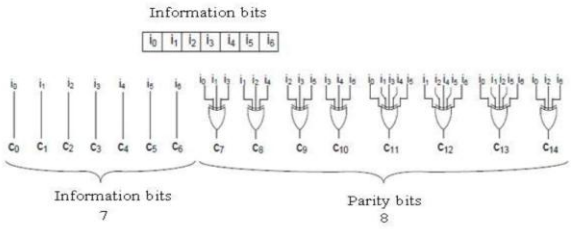

Fig.3: Encoded data from encoder

(7-bits of information**)**

These parity bits are responsible for detecting the presence of error and hence correct the erroneous word. The output produced will be encoded. In general the encoder transfers the

following input information to the memory. Here the input is the 7-bits of information, and then the encoded input is stored in the memory devices of the electronic devices.

B. Proposed Majority Logic Decoder and Detector technique :

The modified version of majority logic detector that overcomes the disadvantages of

Majority logic decoder and syndrome vector with Majority logic decoder method. MLDD is straightforward, power decoder and capable of correcting several random bit flips that depending in the number of the parity check equation. In the MLDD method, the 15-bit codeword input is encoded and decoded. If codeword does not contain any error, then the output will be processed in three iterations.

The advantages of this method are as follows,

a) Ability to correct large number of errors.

b) Sparse encoding, decoding and checking circuits synthesizable into simple hardware.

c) Modular encoder and decoder blocks that allow an efficient hardware implementation.

d) Systematic code structure for clean partition of information and code bits in the memory.

In this proposed technique the error detection process is done in parallel and in pipelining manner.

1. Parallel Processing*:-*

Single iteration is required for detection of any number of errors. Thus the delay time is reasonably low. Also the power consumption and the area requirement are low. The entire error in any bit of the given data is detected simultaneously in single iteration. But there is no cyclic shift as in serial error detection process. The logic blocks are same for the parallel MLDD as in the serial MLDD. But required is more number of majority gates and correction gates, each gate is assigned for a single bit. Error detection process is also done in pipelining manner for the parallel technique. In this process area requirement is further reduced compared to parallel processing. The memory schematic for parallel processing MLD is shown in fig.4.

In Parallel schematic each bit of the code word fed for error detection and correction consist of its parity check equation, correction gate and majority gate. Area increase in the parallel process because of using individual gates for each bit and power also increases as the area increases.

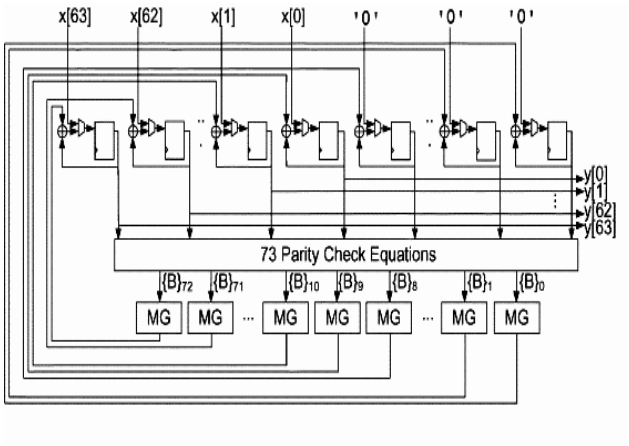

Fig.4: Schematic of the proposed MLDD for N-bit codeword

2. Pipelining Parallel Processing :-

The pipelining process is done for the proposed Parallel processing technique by adding registers. So that the delay is reduced Compared to parallel processing, power and area is compared with the parallel technique increases. In pipelining technique, using partitioning method the input data is divided into several sets of inputs. The cut-set partitioning algorithm is used here. Since the inputs are partitioned, the area and power has been little bit increased compared to the Parallel technique. But the delay is reduced compared to the Serial and parallel techniques.

CONCLUSION

In this brief, using parallel one step majority logic decoding technique the detection of errors single iteration is more practical than the serial manner. The EG-LDPC code was used to analyze errors up to four bit flips. The proposed method of parallel and its pipelining process detects any number of errors in a single iteration. Further in the future work includes extending the practical analysis to more number of errors with much more reduction of area and power.

ACKNOWLEDGMENT

I express my sincere gratitude to 'S. S. Thorat', Assistant Professor, Electronics Engineering Department, Government College of Engineering, Amravati, for extending his valuable insight for completion this work.

REFERENCES

1. Pedro Reviriego, Juan A. Maestro, and Mark F. Flanagan, (Jan 2013) "Error Detection in Majority Logic Decoding of Euclidean Geometry Low DensityParity Check (EG-LDPC) Codes," IEEE Trans. Very Large Scale Integr.(VLSI) Syst..vol. 21, no. 1,

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

3. Lin.S and D. J. Costello, (2004) Error Control Coding, 2nd ed. Englewood Cliffs, NJ: Prentice-Hall,.

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

5. R. G. Gallager, *Low-Density Parity-Check Codes*. Cambridge, MA: MIT press, 1963.

6. 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

7. Y. Kou, S. Lin, and M. P. C. Fossorier, 'Low-density parity-check codes based on finite geometries," IEEE Trans. Inf. Theory, vol. 47, no. 7, Jul. 2001.

8. H. Naeimi, A. DeHon, "Fault secure encoder and decoder for Nano Memory applications," IEEE Trans. Very Large Scale Integration.(VLSI) Syst.. vol. 17, no. 4,April 2009.