

# High Secured and Fault Tolerant Errorless Communication for Low Power Asynchronous Applications

<sup>1</sup>. R. Himaneesh Yadav

<sup>2</sup> I. Saidulu

<sup>1</sup>. *M.tech Student, QIS Institute of Technology, Ongole*

<sup>2</sup>. *Assistant Professor, QIS Institute of Technology, Ongole*

## ABSTRACT:

This enhanced project design a fully fault-tolerant memory architecture that is capable of tolerating hardware or software errors not only in the memory bits but also in the supporting logic including the ECC encoder and corrector with low power consumption and more security. This project uses a Euclidean Geometry codes, SEA (Scalable Encryption Algorithm). Hence proved that these codes are part of a new subset of ECCs that have FSDs. Using these FSDs we design a fault-tolerant encoder and corrector with more security, where the fault-secure detector monitors the parity-check Matrix of an FSD-ECC (fault secure detector - error correcting code) has a particular structure that the decoder circuit, generated from the parity-check Matrix, is Fault-Secure. LDPC codes satisfies a new, restricted definition for ECCs which guarantees that the ECC codeword has an appropriate redundancy structure such that it can detect multiple errors occurring in both the stored codeword in memory and the surrounding circuitries.

**Introduction:** Memory cells have been protected from soft errors for more than a decade; due to the increase in soft error rate in logic circuits, the encoder and decoder circuitry around the memory blocks have become susceptible to soft errors as well and must also be protected. A fault-tolerant nanoscale memory architecture which tolerates transient faults both in the storage unit and in the supporting logic (i.e., encoder, decoder, corrector and detector circuitries) is introduced.

Transient faults: When a node in the system loses its effective charge due to ionized particle hit or various source of noises, it may cause the value of a node to be flipped in the circuit. However, the error does not permanently change the circuit, and it only generates a faulty bit value at the node that can last for one or few cycles. The transient fault rate is the probability that a single node loses its correct value during one clock cycle. Feature-size scaling, faster clock cycles and lower power designs increase the transient fault rate. Feature-size scaling and voltage level reduction

shrinks the amount of critical charges holding logical state on each node; this in turn makes each node more susceptible to transient faults, e.g., an ionized particle strike has higher likelihood of being fatal as the critical charge is reduced in a node, which may cause a glitch or bit-flip. Furthermore operating at higher clock frequency increases the probability that a glitch in the signal is latched and propagated as an erroneous value throughout the circuit. However, many transient faults will not be latched. Some of the latched data may not be relevant to machine operation and there will be no perceivable error in the program operation. Hence, the effective error rate of a large combinational circuit needs to be derated. Particularly, a class of error-correcting codes (ECCs) that guarantees the existence of a simple fault-tolerant detector design is identified. This class satisfies a new, restricted definition for ECCs which guarantees that the ECC codeword has an appropriate redundancy structure such that it can detect multiple errors occurring in both the stored codeword in memory and the surrounding circuitries. This type of error-correcting codes are called as fault-secure detector capable ECCs (FSD-ECC). The parity-check Matrix of an FSD-ECC has a particular structure that the decoder circuit, generated from the parity-check Matrix, is Fault-Secure. The ECCs identified in this class are close to optimal in rate and distance, suggesting we can achieve this property without sacrificing traditional ECC metrics. The fault-secure detection unit is used to design a fault-tolerant encoder and corrector by monitoring their outputs. If a detector detects an error in either of these units, that unit must repeat the operation to generate the correct output vector. Using this retry technique, potential transient errors in the encoder are corrected and provide a fully fault-tolerant memory system.

## Goal

A class of error-correcting codes (ECCs) that guarantees the existence of a simple fault-tolerant detector design should be identified. This class should satisfy a new, restricted definition for ECCs which guarantees that the ECC codeword has an appropriate redundancy structure such that it can detect multiple errors occurring in both the stored codeword in memory and the surrounding circuitries. The parity-check Matrix of an FSD-ECC should have a particular structure that the decoder circuit, generated from the parity-check Matrix, is Fault-Secure. The fault-secure detector should be designed, potential transient errors in the encoder are corrected using a corrector block and should provide a fully fault-tolerant memory system

## Scope

Nanotechnology design has attracted considerable attention in recent years and seems to be the technology for the future generation of the electronic devices, either as scaled and more restricted conventional lithographic technology, or as emerging sublithographic technologies, such as nanowires, carbon nanotubes, NDR (Negative Differential Resistance) devices, or other nanotechnology devices. Each of these technologies provides one or more design benefits including feature-size scaling, high on-off ratios, and faster devices. However, all of these techniques share their most challenging design issue: reliability. Without adding much circuitry to provide reliability is very important. Protecting supporting logic is essentially done by the fault-secure detector and the only cost to achieve the logic protection is the cost that we pay for the detectors. This FSD-ECCs are used to reduce area in nanotechnology design.

## Literature Survey

The theory of error-correcting codes was originated in the late 1940's by Richard Hamming, a mathematician who worked for Bell Telephone. Hamming's motivation was to program a computer to correct "bugs" which arose in punch-card programs. Hamming's overall motivation behind the theory of error-correcting codes was to reliably enable digital communication. LDPC codes were first developed in a doctoral dissertation in 1963 by

R.G. Gallager. Gallager's work was largely ignored for approximately 30 years until connections were drawn between the iterative methods used for decoding both LDPC codes and Turbo codes. Low density parity check (LDPC) codes were first discovered by Gallager [1,2] in the early 1960's and have recently been rediscovered and generalized [3–

14]. They have experienced an amazing comeback in the last few years. Unlike many other classes of codes LDPC codes are already equipped with very fast (probabilistic) encoding and decoding algorithms. It has been shown that these codes achieve a remarkable performance with iterative decoding that is very close to the Shannon limit. Consequently, these codes have become strong competitors to turbo codes for error control in many communication and digital storage systems where high reliability is required. An LDPC code is defined as the null space of a parity check matrix  $\mathbf{H}$  with the following structural properties: (1) each row consists of  $r$  "ones"; (2) each column consists of  $c$  "ones"; (3) the number of "ones" in common between any two columns, denoted  $\lambda$ , is no greater than 1; (4) both  $r$  and  $c$  are small compared to the length of the code and the number of rows in  $\mathbf{H}$  [1, 2]. Since  $r$  and  $c$  are small,  $\mathbf{H}$  has a small density of "ones" and hence is a sparse matrix. For this reason, the code specified by  $\mathbf{H}$  is called an LDPC code. The LDPC code defined above is known as a regular LDPC code.

## Low Density Parity Check Code Construction

Basic Construction of LDPC codes Although LDPC codes can be applied in any field, they are mostly considered over the GF(2) field - the binary case. For simplicity, when referring to LDPC codes consider them in the binary case. Low Density Parity Check codes are codes of construction  $(n, c, r)$  and defined by a matrix which always has the following properties: The codes are of low density. That is, they contain mostly 0's and very few 1's.

- Contains block length  $n$ . That is, the number of columns in both the Generator Matrix and the Parity Check Matrix are of length  $n$ .
- Each row in the parity check matrix has exactly  $r$  1's. Each column in the parity check matrix has exactly  $c$  1's.
- $r/n$  and  $c/n$  are 'small' (this is to satisfy the concept of the check matrix being of 'low density'). In general,  $r/n, c/n \leq 1/4$
- The linear binary code  $C$  is defined by  $C = \{c \in \mathbb{F}_2^n \mid Hc = 0\}$ .

## Low Density Parity Check codes

### Linear Block Codes

Since LDPC codes are a special case of linear block codes (LBC), in this section we will have an overview of this class of codes to set up a ground for discussing LDPC encoding and decoding. To encode, we need to map the information

$i = [a_1, a_2, \dots, a_K]$  into a codeword  $c = [c_1, c_2, \dots, c_K, c_{K+1}, \dots, c_N]$  i.e.  $c = f(i)$ . Now the mapping can be a linear mapping. The canonical form of a linear transformation is

$$c = i * G$$

Where  $G$  is a  $K \times N$  matrix and all the code words  $\{c\}$  are distinct when the rank of  $G$  is  $K$ . The code rate of such a code is  $K/N$  i.e. there are  $K$  information bits per  $N$  coded bits.

For a linear block code, the linear combination of any subset of code words is a codeword. We describe the encoding and decoding of LBC.

We first write the basis vectors (of size  $1 \times N$ ) of  $G$  i.e.,  $[g_1, g_2, \dots, g_k]$  of  $C$  as rows of matrix  $G$  ( $K \times N$ ).

Information  $i = [a_1, a_2, \dots, a_K]$  is encoded uniquely as,

$$c = a.G = [a_1, a_2, \dots, a_K].G, \quad a_i \in GF(2)$$

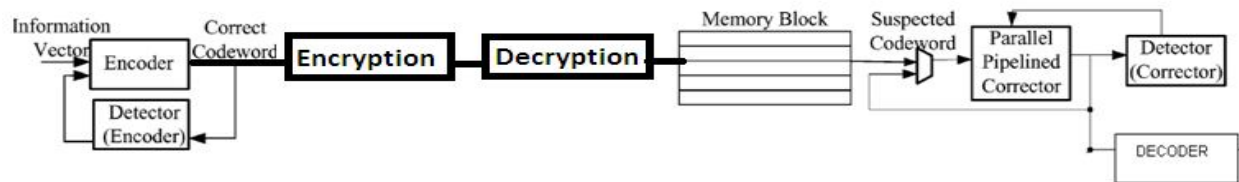
The dual space of a linear code  $C$  is denoted by  $C^T$ , which is a vector space of dimension  $(N-K)$ . A basis  $\{h_0, h_1, \dots, h_{N-K-1}\}$  for  $C^T$  can be found and used to construct a Parity.

### System overview

memory system which can tolerate errors in any part of the system, including the storage unit, encoder and corrector circuit, using the fault-secure detector is shown below. There is a fault secure detector that can detect any combination of errors in the received code-word along with errors in the detector circuit. This fault-secure detector can verify the correctness of the encoder and corrector operation.

An overview of the proposed fault secure encoder and decoder is shown in figure 3.1, and

is as described below.



Block diagram of Fault Secure Encoder and Decoder.

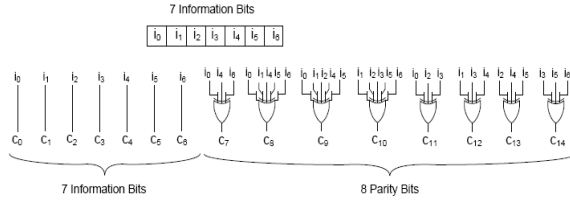
The information bits are fed into the encoder to encode the information vector, and the fault secure detector of the encoder verifies the validity of the encoded vector. If the detector detects any error, the encoding operation must be redone to generate the correct code-word. The code-word is then stored in the memory. Later during operation, the stored code-word will be retrieved from the memory unit. Since the code-word is susceptible to transient faults while it is stored in the memory, the retrieved code-word must be fed into the detector to detect any potential error and possibly to the corrector to recover any erroneous bits. In this design the corrector circuit has parallel structure and is implemented fully pipelined similar to the detector. All the memory words are pipelined through the corrector and then detector, therefore, one corrected memory word is generated every cycle. The detector following the corrector, would raise an error-detection flag only if a transient fault occurs in the corrector or detector circuitry. Due to the relative lower transient fault rate compared to the permanent defects and the relative small corrector and detector circuitry, this happens with low

frequency. Therefore, the potential throughput loss of this system is low.

### Design Structure:

In this section the design structure of the encoder, corrector, and detector units of the proposed fault secure encoder and decoder is provided.

**Encoder:** An  $n$ -bit code-word  $c$ , which encodes  $k$ -bit information vector  $i$  is generated by multiplying the  $k$ -bit information vector with  $k \times n$  bit generator matrix  $G$ , i.e.,  $c = i \cdot G$ . Figure 3.2 shows the generator matrix of  $(15, 7)$  EG-LDPC code. all the rows of the matrix are cyclic shifts of the first row. This cyclic code generation does not generate a systematic code and the information bits must be decoded from the encoded vector, which is not desirable for our fault-tolerant approach due to the further complication and delay that it adds to the operation. The generator matrix of any cyclic code can be converted into systematic form ( $G = [I : X]$ )



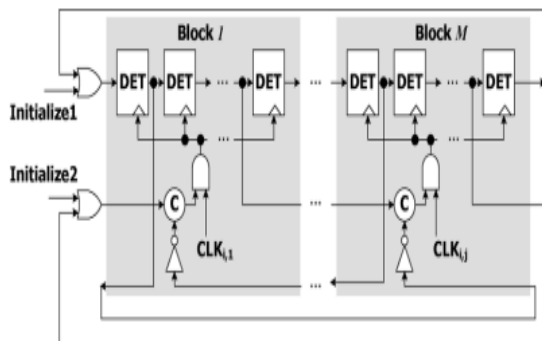
**Fault Secure Detector:**

The core of the detector operation is to generate the syndrome vector, which is basically implementing the following vector-matrix multiplication on the received encoded vector  $c$  and parity-check matrix  $H$ .  $cH^T = S$

**SECURITY USING SEA**

In this block, we consequently consider a general context where we have very limited processing resources (e.g. a small processor) and throughput requirements. It yields design criteria such as: low memory requirements, small code size, limited instruction set. In addition, we propose the flexibility as another unusual design principle.  $SE_{An,b}$  is parametric in the text, key and processor size. Such an approach was motivated by the fact that many algorithms behave differently on different platforms (e.g. 8-bit or 32-bit processors). In opposition,  $SE_{An,b}$  allows to obtain a small encryption routine targeted to any given processor, the security of the cipher being adapted in function of its key size. Beyond these general guidelines, alternative features were wanted, including the efficient combination of encryption and decryption or the ability to derive keys “on the fly”.

**MODIFIED RING COUNTER:**



Modified Ring Counter

**DET (Double edge triggered flip-flops):**

Double-edge-triggered (DET) flip-flops are utilized to reduce the operating frequency by half. The logic construction of a double-edge-triggered (DET) flip-flop, which can receive input signal at two levels the clock, is analyzed and a new circuit design of CMOS DET. In this paper, we propose to use double-edge-triggered (DET) flip-flops instead of traditional DFFs in the ring counter to halve the operating clock frequency. Double edge-triggered flipflops are becoming a popular technique for low-power designs since they effectively enable a halving of the clock frequency. The paper by Hossain et al [1] showed that while a single-edge triggered flipflop can be implemented by two transparent latches in series, a double edge-triggered flipflop can be implemented by two transparent latches in parallel.

**C ELEMENT:**

The Muller **C-element**, or Muller C-gate, is a commonly used asynchronous logic component originally designed by David E. Muller. It applies logical operations on the inputs and has hysteresis. The output of the C-element reflects the inputs when the states of all inputs match. The output then remains in this state until the inputs all transition to the other state. This model can be extended to the Asymmetric C-element where some inputs only effect the operation in one of the transitions (positive or negative). The figure shows the gate-level and transistor-level implementations and symbol of the C-element.

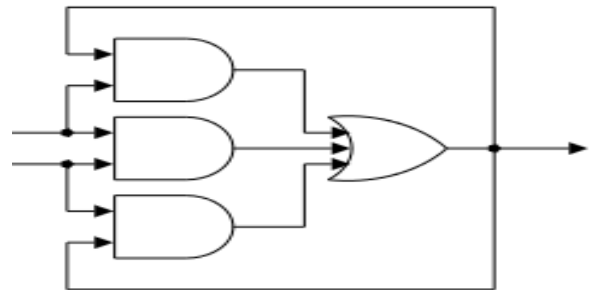
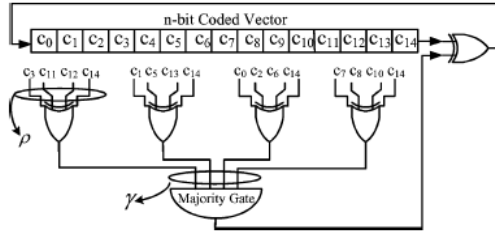


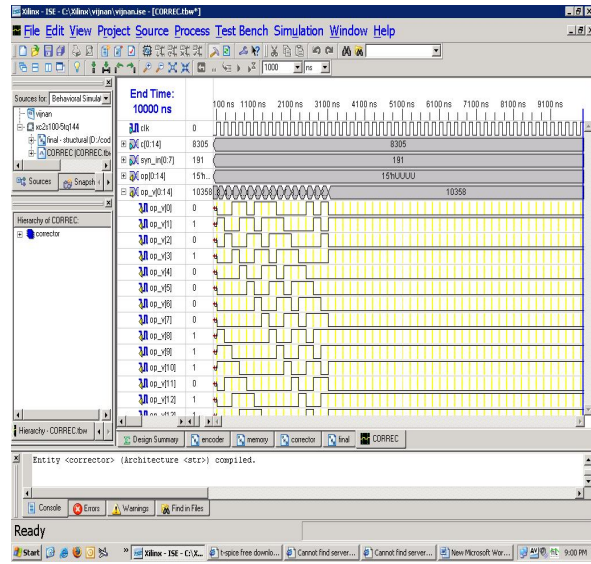
Fig 3.4: C- Element Table Corrector :

One-step majority-logic correction is a fast and relatively compact error-correcting technique. There is a limited class of ECCs that are one-step-majority correctable which include type-I two-dimensional EG-LDPC. In this section, we present a brief review of this correcting technique. Then we show the one-step majority-logic corrector for EG-LDPC codes.

- 1) The C-element stores its previous state with two cross-coupled inverters, similar to an SRAM cell. One of the inverters is weaker than the rest of the circuit, so it can be overpowered by the pull-up and pull-down networks.

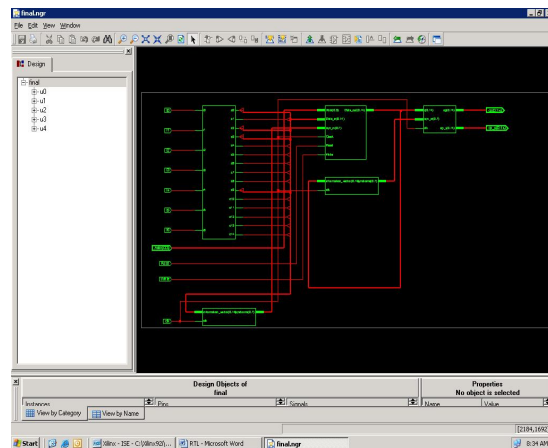


Serial one-step majority logic corrector structure  
Simulation Results



Synthesis Reports:

RTL Schematic:



38% power reduction is observed in proposed technique with more security.

References

- [1] Parker, “Prism: Probabilistic symbolic model checker,” in *Proc. TOOLS 2002*, pp. 200–204.
- [2] Y. Chen, G. Jung, D. A. A. Ohlberg, X. Li, D. Stewart, J. O. Jeppesen, K. A. Nielsen, J. F. Stoddart, and R. S. Williams, “Nanoscale molecular- switch crossbar circuits,” *Nanotechnology*, vol. 14, pp. 462–462, 2003.
- [3] Z. Zhong, D. Wang, Y. Cui, M. W. Bockrath, and C. M. Lieber, “Nanowire crossbar arrays as address decoders for integrated nanosystems,” *Science*, vol. 302, pp. 1377–1379, 2003.
- [4] International Technology Roadmap for Semiconductors: 2004 Update.
- [5][Online]. Available: [http://www.itrs.net/Common/2004Update/2004\\_05\\_ERD.pdf](http://www.itrs.net/Common/2004Update/2004_05_ERD.pdf)
- [6] S. Jin, D. Whang, M. McAlpine, and R. Friedman, “Interconnection and inte-gration of nanowire devices without registration,” *Nano Lett.*, vol. 4, pp. 915–919, 2004.
- [7] G. Wang, T. Li, and J. Chen, “A probabilistic approach to fault-tolerant routing algorithm on mesh networks,” *Proc. ICPADS 2004*, p. 577.
- [8] G. Wang and J. Chen, “Probabilistic analysis of connectivity on mesh networks,” in *Proc. ITCC 2003*, pp. 362–366.
- [9] C. Amsinck, N. Di Spigna, S. Sonkusale, D. Nackashi, and P. Franzon, “Scaling constraints in nanoelectronic random access memories,” in *Nanotechnology*. Philadelphia, PA: IOP Publ., 2005, vol. 16, pp. 2251–2260.



**R.HIMANEESH YADAV** received the B.Tech degree in ECE from Rao and Naidu Engineering college, ongole in 2012. He is doing M.tech in ECE department with a specialization of VLSI&Embedded systems in QIS Institute of Technology, Ongole. His research interest is in VLSI system designing.



**I.SAIDULU** received the B.Tech degree in ECE from Prakasam Engineering College, Kandukur in 2008. He received the M.Tech degree in ECE from Kakinada Institute of Engineering and Technology, Kakinada in 2010. He has 6 years experience in teaching for U.G and P.G students. He guided many B.Tech and M.tech projects. Now he is working as Assistant Professor in the department of ECE at QIS Institute of Technology, Ongole. His research interest is in VLSI system design.