HIGH SPEED DESIGN APPROACHES FOR CCSDS LDPC ENCODER SYSTEMS

A Dissertion

Submitted by

JEESHMA T N

(CB.EN.P2VLD22019)

in partial fulfillment of the requirements for the award of

the degree of

Master of Technology

in

VLSI Design

under the supervision of

Mr. Pargunarajan K

Department of Electronics and Communication Engineering,
Amrita School of Engineering,
Amrita Vishwa Vidyapeetham,
Coimbatore – 641112

December 2023

Declaration

I do hereby declare that this dissertation titled **"High Speed Design Approaches For CCSDS LDPC Encoder Systems",** submitted in partial fulfillment of the requirements for the award of the degree of **Master of Technology in VLSI Design**, is a true record of work carried out by me, under the supervision of **Mr. Pargunarajan K** and that all information contained herein, which do not arise directly from my work, have been properly acknowledged and cited, using acceptable international standards. Further, I declare that the contents of this thesis have not been submitted, in part or in full, for the award of any other degree or diploma.

Coimbatore – 641112 **JEESHMA T N**

Date :

Abstract

Low density parity check (LDPC) codes are linear block codes that are commonly used for error detection and correction in high-speed digital communication systems such as digital broadcasting, optical fiber communications, and wireless local area networks. Because of its substantial performance in error correction, LDPC codes have been the focus of extensive research.as is required.

LDPC Code is a type of Block Error Correction code that has been discovered and has performance that is extremely close to Shannon’s limit. Reliable communication is enabled by good error correction performance. More research is being conducted for its efficient design and application since its discovery by Gallagar. Although there is no one-of-a-kind way for creating LDPC codes. The LDPC Code is implemented by taking several elements into account, such as error rate, decoder parallelism, ease of implementation, and so on.

The work is about FPGA implementation of LDPC codes . Protograph codes were introduced and analyzed by NASA's Jet Propulsion Laboratory in the early years of this century. CCSDS Experimental specification has been contributed to CCSDS by NASA. It describes a set of Low Density Parity Check (LDPC) codes including definition of their code structures, encoder implementations and experimental performance results. I have used MATLAB for generating parity check matrix and generator matrix.The main objective is Create a more efficient technique of encoding Quasi-Cyclic Low-Density Parity-Check codes .Use an FPGA to demonstrate the effectiveness of this method.

Table of Contents

List of Tables

List of Abbreviations

CCSDS	Consultive Committee for Space Data Systems
FPGA	Field Programmable Gate Array
LDPC	Low Density Parity Check Codes
VLSI	Very Large Scale Integration

List of Symbols

$\in$	Belongs
∞	Infinity

List of Tables

List of Figures

1. Introduction

The mathematical rules that control the speed at which data can be securely conveyed across a noisy channel were deduced by Claude Shannon in his groundbreaking 1948 article. Information theory, a brand-new science dedicated to studying it and its sister discipline, error-correcting codes, was founded on this mathematical framework.

According to Shannon's noisy channel coding theorem, there is a maximum rate at which we may communicate with vanishing error probabilities for any given channel. The channel's capacity is referred to as this highest rate. Shannon also demonstrated that practically any very long code can reach this capacity. But this was not a constructive proof. Technically speaking, a long, random code would work fine, but the encoding and decoding durations would be unacceptably high.

Building capacity-achieving codes with reasonable encoding and decoding speeds has been the ultimate goal of coding theory in the decades after Shannon's breakthrough. The introduction of turbo codes in 1993 was a significant accomplishment in this endeavor. Iterative decoding, which closed the gap between high performance and low complexity, was introduced with turbo codes. More specifically, with a complexity that increases only linearly with code length, iterative decoding can attain performance that approaches theoretical bounds.

The topic had a surge of interest in study with the discovery of turbo codes, which specifically resulted in the rediscovery of Gallager's 1963 work on low-density parity check (LDPC) codes. Despite the fact that Gallager's work had been mostly overlooked because of the restricted processing power.

Luby et al. presented the concept of irregular codes in 1998 as a means of enhancing according to Gallager's usual codes. A protograph code was first proposed by NASA's Jet Propulsion Lab (JPL) five years later. Compared to irregular codes, protograph codes are more organized, allowing for simpler code descriptions without compromising performance .Protograph codes are an example of the multi-edge type construction first described by Richardson and are closely connected to Tanner's codes derived from seed graphs.It has been demonstrated that LDPC codes with iterative decoding perform exceptionally well on a variety of channels, almost reaching capacity on the additive white Gaussian noise (AWGN) channel and, when the code's length increases to infinity, reaching it on the binary erasure channel (BEC).

1.1 The Motivation

By utilizing parallelism, this study seeks to improve the efficiency of Quasi-Cyclic Low-Density. Parity-Check (QC- LDPC) code encoding. The creation of an enhanced encoding technique and its application on FPGA for real-world validation are the main goals. The main objective is to significantly outperform current techniques in terms of encoding speed while putting a focus on resource efficiency. The goal of this research is to improve LDPC encoding methods and make them more practical and efficient for use in real-world settings.

1.2 Objectives

Create a more effective way to use inherent parallelism to encrypt Quasi-Cyclic Low-Density ParityCheck codes. Use FPGA to test this approach's effectiveness and viability in real-world scenarios. The primary objective is to maintain low resource consumption while achieving a significant speedup in encoding speed when compared to current approaches.

1.3 Overall work and thesis outline

The structure of the thesis is as follows:

Chapter 2 The relevant study that was done to comprehend the implementation of the LDPC Encoder.

Chapter 3 gives an extensive review of LDPC encoding is provided.

Chapter 4 gives an extensive overview of hardware modeling.

.

2. Literature Review

The paper [1] presents about space communications, quasi-cyclic low-density parity-check (QC-LDPC) codes are frequently employed. The Consultative Committee for Space Data Systems (CCSDS) has proposed encoder layouts that are currently inefficient for high-throughput applications that use a a lot of registers and logical resources. We introduce a novel encoder architecture designed to tackle these problems. Our method finds common subexpressions (CS) throughout the encoding process by using a grouping technique. To further minimize logical resources, we merge circuit structures that are comparable using a two-layer architecture. We present a preprocessing technique for a certain size of the generator matrix. Furthermore, a control unit is replaced by configuration registers. When compared to earlier designs, this shows a notable increase in throughput and resource utilization.

The paper [2] presents about a two-step encoding strategy for the 12 quasi-cyclic (QC)-low-density parity-check (LDPC) codes. Instead of targeting each code individually, the proposed strategy takes into account all codes in the specific set. The suggested approach uses inverse matrices to perform multiplication. The introduced encoding method greatly reduces the complexity of the multiplications. It enables the creation of full-parallel architectures for any of the supported codes that run the encoding process in a single clock cycle, or more for pipelined implementations. In addition, an XOR-gate tree-based VLSI encoding architecture is proposed. The proposed technique makes use of the structure and properties of the matrices involved to extract common subexpressions (CSs) utilizing common sub-expression analysis.

The paper [3] presents about the, Consultative Committee for Space Data Systems (CCSDS)-adopted quasi-cyclic low-density parity-check (QC-LDPC) codes are commonly used in Deep Space (AR4JA) and Near-Earth (C2) communications. A wide variety of encoder architectures that adhere to the CCSDS standard have been proposed. However, conventional architectures are inefficient in high-throughput implementations; many systems include a large number of logical resources and registers. We provide a unique architecture with reduced resource use in this brief. A grouping method is utilized to retrieve the encoding scheme's common subexpressions.

In paper [4], we suggest a simple yet effective coding strategy for transmitting extra bits by cyclically shifting a low-density parity-check (LDPC) coded sequence with no increased bandwidth or transmission energy consumption. The payload data is encoded into an LDPC coded sequence and then cyclically shifted according to the pattern provided by the additional bits at the transmitter. The extra bits are first retrieved at the decoder by estimating the amount of cyclic shifts based on the hard or soft syndromes. The LDPC decoder recovers the payload data after removing the effect of cyclic shift. The simulation findings show that with a 5G LDPC code of length 1910, up to 10 additional bits can be reliably sent while having no effect on the reliability of the code.

In paper [5] presents an efficient reconfigurable encoder that complies with the IEEE 1901 standard. First, we present a reconfigurable LDPC-CC encoder to meet the multirate requirement while also improving the architecture through fine-tuned parallelization that takes full advantage of the codeword structure's properties.

In paper [6] propose finite-length irregular protograph-based quasi-cyclic low density parity-check codes with good waterfall performance and a low error floor in this paper. To obtain a low error floor, we exclude a certain collection of dominant elementary trapping sets from the code's Tanner graph. The design is built on a search technique that determines whether or not any instance of any structure inside exists in the generated code's Tanner graph. Because the search algorithm does this operation with lowest complexity.

In paper [7] Because of their high performance curves in the waterfall and error floor areas, quasi-cyclic LDPC codes and are two fundamental families of LDPC codes that are preferred above others.

So here Creating a more efficient method of encrypting Quasi-Cyclic LowDensity Parity-Check codes using inherent parallelism. FPGA can be used to assess the efficacy and viability of this method in real-world circumstances. When compared to present techniques, the key goal is to retain minimal resource use while significantly increasing encoding speed.

3. Basics Of LDPC Encoding

3.1 Linear Block Codes

The order of bits to be transferred in a digital communication system is set up in blocks of k bits. Thus, there are 2^k-tuples that could be sent. When a block code is used, the encoder gives each k- tuple where $n > k$ n bits. In order for a block code to be functional, each of the 2^k , n- tuples, also known as codewords, must be unique. That is, the encoder's input (represented by u) and output (represented by v) should correspond one to one.

If the codewords don't follow a specific structure, the encoding process will be excessively complicated, which also affects the decoding process. We are interested in linear block codes because of this.

A code is said to be linear if any two of its codewords may be combined linearly to form another codeword, or vice versa: What linear block codes are defined as: If and only if the 2^k codewords in a block code of length n constitute the k-dimensional subspace of the vector space of nn- tuples over $G(2)$, then the block code is a (n, k) linear code. matrix generator A k-dimensional subspace of all the binary n-tuples (Vn) is a linear (n, k) code C. Finding k linearly independent elements of C, such as $b0$, $b1$, ⋯, and $gk-1$, will allow us to write any $v \in V$ as follows:

$$v = u_0 g_0 + u_1 g_1 + \cdots . u_{k-1} g_{k-1}. \tag{3.1}$$

$$\mathrm{G} = \begin{bmatrix} g_0 \\ g_1 \\ \\ g_k \end{bmatrix} \quad \begin{bmatrix} g_{0,0} & g_{0,1} & g_{0,n-1} \\ g_{1,0} & g_{1,1} & g_{1,n-1} \\ \\ g_{k-1,0} & g_{k-1,1} & g_{k-1,n-1} \end{bmatrix} \tag{3.2}$$

where G is a binary matrix of size $k \times n$. Assume that the message to be sent is $u = (u0, u1, \cdots, uk-1)$. The codeword can then be stated as follow

$$V = u \,.\, G = (u_0, u_1, \ldots \ldots, u_{k-1}) \begin{bmatrix} g_0 \\ g_1 \\ \\ \end{bmatrix} = u_0\, g_0 + u_1 g_1 \ldots . + u_{k-1} g_{k-1} \tag{3.3}$$

g_k

3.1.1 Parity-check matrix

An (n - k) x n matrix with (n - k) independent rows is the parity-check matrix H. It is the code c's dual space, or G.$\boldsymbol{H^T}$ = 0.

$$H = \begin{pmatrix} h_1 \\ h_2 \\ \\ h_{n-k} \end{pmatrix} \begin{pmatrix} h_{1,1} & h_{1,2} & h_{1,3} & h_{1,n} \\ h_{2,1} & h_{2,2} & h_{2,3}, & h_{2,n} \\ \multicolumn{4}{c}{\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots} \\ h_{n-k,1} & h_{n-k,2} & h_{n-k,3} & .h_{n-k,n} \end{pmatrix} \quad (3.4)$$

The parity check matrix, cH^T= 0, provides the parity check equations.H is a detailed specification for a certain block code.

3.1.2 Block Codes in Systematic form

The codeword is arranged systematically, with k message bits and (n−k) parity-check bits in order of precedence.

For a systematic linear block code $c(n,k)$, the generator matrix G is defined as follows:$[I_k\ p_{k(n-k)}]G = [I_k \mid p_{k(n-k)}]$ is a compact notation where $I_k$is the identity matrix of size $k \times k$ that corresponds to the message bits.$p_{k(n-k)}$ is the parity matrix representation.

$$G= \left(\begin{array}{cc|ccc} 1 & 0 & 0\ldots\ldots\ldots\ldots 0 & p_{1,k+1} & p_{1,k+2} & p_{1,n} \\ 0 & 1 & 0\ldots\ldots\ldots\ldots 0 & p_{2,k+1} & p_{2,k+2} & p_{2,n} \\ 0 & 0 & 1\ldots\ldots\ldots\ldots 0 & p_{3,k+1} & p_{3,k+2} & p_{3,n} \end{array}\right) \quad (3.5)$$

3.2 How to decode linear block codes

The received vector $r = (r_1, r_2, \ldots\ldots r_n,)$ may contain mistakes when a codeword is transmitted over a noisy channel. An error vector, or pattern, $e = (e_1, e_2, \ldots\ldots e_n)$, where $e = r + c$, is used to model errors.The sent codeword in this case is c. The condition $c^T . H^T$=0, where H is the parity-check matrix, is necessary for

error detection. One computes the syndrome vector, $s = r \cdot H^T$. The received vector r is multiplied by the transpose of the parity-check matrix H^T to get the syndrome vector $s = (s_1, s_2 \ldots\ldots\ldots\ldots s_n)$. The received vector is a valid codeword if s is the all-zero vector. If not, the received vector contains mistakes. When errors are found, the syndrome array is examined to determine which error pattern, e_j, corresponds to each $1,2,\ldots\ldots, j = 1,2,\ldots, n$.The message that has been decoded is $m' = r + ej$.

3.3 The LDPC codes

LDPC codes are denoted as $(n, k)\ or\ (n, w_c, w_r)$, where n is the codeword length. k is the message bit length, and wc is the column weight, which represents the number of nonzero elements in a column of the parity-check matrix.,w_r is the row weight, which represents the number of nonzero elements in a parity-check matrix row.LDPC codes have two distinct characteristics: Representation of a Parity-Check Matrix: A binary parity-check matrix H represents LDPC codes. The matrix H must meet the c^T. H^T=0 condition, where c is a codeword. The parity-check matrix H is sparse, which means that the number of '1's in the matrix is substantially lower than the number of '0's. The condition $c^T\ H^T$=0, where c is a codeword, must be met by the matrix H.

The parity-check matrix H exhibits a low-density property wherein the number of '1's in the matrix is significantly less than the number of '0's. This sparseness of H further contributes to its low computing complexity**.** The parity-check matrix H exhibits a low-density property wherein the number of '1's in the matrix is significantly less than the number of '0's. This sparseness of H further contributes to its low computing complexity.

Tanner graphs Tanner (1981) developed a bipartite Tanner graph as a way to express LDPC codes. Two sets of vertices make up the graph: check nodes (k vertices for parity-check equations) and variable nodes (n vertices for codeword bits). If the matching bit is present in the parity-check equation, an edge links a variable node to a check node. The Tanner graph has the same number of edges as the parity-check matrix's number of ones.

Cycle: A series of interconnected vertices. It has the same vertex at both ends. In a bipartite Tanner graph, the length of each cycle is even.

Girth: An essential component of LDPC code efficiency, the girth of a Tanner graph is the smallest cycles in the graph.

Regular LDPC codes have the benefit of regularity; the parity check matrix in regular LDPC codes has uniform column and row weights. When it comes to implementation, regular codes are simpler than irregular LDPC codes.

With respect to the code length n and the number of rows in H, both wc and wr should be modest values. LDPC codes typically have a coding rate of $R = 1 - \frac{Wc}{Wr}$.

Of general, irregular LDPC codes perform better at error-correcting than regular LDPC codes. The parity check matrix of irregular LDPC codes has non-uniform column and row weights.

3.4 Quasi-cyclic (QC) LDPC codes

In contrast to LDPC codes that are generated at random, QC-LDPC codes are structured constructions with a minimum girth of 6. Shift registers can be used to encode them in linear time. The low encoding complexity and low memory requirements of QC-LDPC codes are well recognized. They continue to perform well at error correction in spite of this. The parity-check matrix of QC-LDPC codes, which is made up of tiny square blocks, is its main characteristic. These blocks can be circulant permutation matrices or zero matrices. A $q \times q$ matrix that moves the identity matrix I i times to the right is called the circulant permutation matrixp^i.For an all-zero matrix, p^∞ is used. An $mq \times nq$ matrix, with m and n being multiples of an integer q, is the parity-check matrix $H. A_{i,j} \in \{0,1,\ldots,q-1,\infty\}$ are the elements.

$$H = \begin{pmatrix} p_{11}^a & p_{12}^a & p_{1(n-1)}^a & p_{1n}^a \\ p_{21}^a & p_{22}^a. & p_{2(n-1)}^a & p_{2n}^a \\ \cdots & \cdots & \cdots & \cdots \\ p_{m,1}^a & p_{m,2}^a & p_{m,n-1}^a & p_{m,n-1}^a \end{pmatrix} \quad (3.6)$$

With a codeword size of N=nq and an information bit size of M=(n−m)q, the code has full rank. The formula for the coding rate is

$$R = 1 - \frac{m}{n}. \quad (3.7)$$

Increasing the size of circulant permutation matrices p^i yields larger block LDPC codes. Compared to randomly built LDPC codes, the memory needed to store the parity-check matrix of QC-LDPC codes can be decreased by a factor of $1/q$.

3.4.1 Protograph codes

Protographs are LDPC code ensembles shown graphically. Three protographs from the standard (3,6) ensemble are shown in Figure.

Distinctions in Structure: The structural appearance of the protographs in Figure differs. The center and right protographs both have double edges, however the left protograph does not.

Comparable Spectral Forms: All three protographs share identical spectral forms with the regular (3,6) ensemble, despite their structural variances. The term "spectral shape" describes the performance traits associated with the codes' ability to repair errors.

LDPC code ensembles are represented graphically in prototographs.Figure displays three protographs from the conventional (3,6) ensemble.

Differentiations in Structure: The protographs in Figure have different structural looks. While the left protograph lacks double edges, the center and right protographs do.

Similar Spectral Structures: Despite their structural differences, all three protographs have the same spectral forms as the ordinary (3,6) ensemble."Spectral shape" refers to the performance characteristics linked to the error-repairing capabilities of the codes.

AR4JA protograph : The suggested AR4JA LDPC codes have a comparatively large minimum distance in relation to their block length. For any given operating signal-to-noise ratio, undetected error rates are several orders of magnitude lower than detected frame and bit error rates. A QC LDPC code is produced through the direct QC (Quasi-Cyclic) expansion of the AR4JA protograph, which is represented as a matrix. Two steps are used in the expansion procedure for AR4JA codes as stated in the experimental CCSDS standard. At rate-½, a new larger type-I weight matrix is created after the first cyclic expansion by a factor of 4.Since there are only ones and zeros in the type-I weight matrix, the corresponding protograph lacks parallel edges. In the second phase, Matrix 1. is cyclically enlarged to yield three block lengths, which equate to t0 k=1024 information bits for a QC LDPC code.

3.5 Codes used

The circulant matrices, which are square matrices of binary elements with each row representing a one-position right cyclic shift of the preceding row, are the basis for this initial LDPC code. Therefore, the first row of a circulant determines the circulant as a whole, and low-weight circulants are used to build low-density parity check matrices.

AR4JA LDPC code combines the structure Quasi cyclic added with permutation basing on the basic protograph structure. The various code rates are generated by expanding, copying and permuting the protograph structure. here the sub blocks are related through permutation to make a systematic structure. The advantage of this code is that the BER convergence is faster with lesser number of decoder iterations.

The H matrices for the rate-1/2 codes are specified as follows

$$H_{1/2}= \begin{pmatrix} 0_M & 0_M & 0_M & I_M & I_M \ xor \ \pi_1 \\ I_M \ xor \ \pi_1 & I_M \ xor \ \pi_7 \ xor \ \pi_6 & 0_M & 0_M & I_M \\ 0_M & I_M & I_M \ xor \ \pi_5 & 0_M & \pi_4 \ xor \ \pi_3 \ xor \ \pi_2 \end{pmatrix} \quad (3.8)$$

where I_M and 0_M are the $M \times M$ identity and zero matrices, respectively, and Π_1 through Π_8 are permutation matrices. The H matrices for the rate-2/3 and rate-4/5 codes are specified with additional columns and permutation matrices as follows.

$$H_{2/3} = \left(\begin{array}{cc|c} 0_M & 0_M & \\ I_M & \pi_4 \ xor \ \pi_{10} \ xor \ \pi_9 & H_{1/2} \\ \pi_{14} \ xor \ \pi_{13} \ xor \ \pi_{12} & I_M & \end{array}\right) \quad (3.9)$$

$$H_{2/3} = \left(\begin{array}{ccccc|c} 0_M & 0_M & 0_M & 0_M & & \\ I_M & \pi_{23} \ xor \ \pi_{22} \ xor \ \pi_{21} & I_M & \pi_{17} \ xor \ \pi_{16} \ xor \ \pi_{15} & & H_{2/3} \\ \pi_{26} \ xor \ \pi_{25} \ xor \ \pi_{24} & I_M & \pi_{20} \ xor \ \pi_{10} \ xor \ \pi_M & I_M & & \end{array}\right) \quad (3.10)$$

Permutation matrix Πk has non-zero entries in row i and column πk (i) for i ∈ {0,..., M-1} and

$$\pi_k(i)=\frac{M}{4}((\theta_k + [4i/M])mod\ 4\ ) + (\emptyset_k([4i/M]) + i)mod\ \frac{M}{4}\ ) \qquad (3.11)$$

permutation matrix Πk has non zero entry in row i and column πk(i) for i = 0 to M-1. For different submatrix sizes M= {128,256,512,1024}, the values of θk and Φk . , the values of θk and Φk are taken from CCSDS specification document.

Table 3-1 Codeblock length (bits) for supported code rate

	Submatrix size (M)		
Information Block lenth(n)	Rate 1/2	Rate 2/3	Rate 4/5
1024	512	256	128
2048	1024	512	256
4096	2048	1024	512

Table 3-2 Values of Submatrix Size M for Supported Codes

	Code block length(n)		
Information block length(k)	Rate 1/2	Rate 2/3	Rate 4/5
1024	512	1536	1280
2048	1024	3072	2560
4096	2048	6144	5120

A direct QC expansion of the AR4JA protograph in matrix below will create a QC LDPC code. The AR4JA codes defined in the experimental CCSDS standard use a two step expansion process.

After a first cyclic expansion by a factor of 4, a new larger type-weight matrix obtained in matrix for rate-½.

These codes are QC with a sub block size equal to the second step expansion factor (k). In other words, the two-step process is not equivalent to any single step cyclic expansion. Hence after the expansion according to the desired specifications the resultant matrix has a dimension of(3072 X 5120) and the matrix contains a total of 15230 non zero elements “1”.

4. ENCODER

Large codeword length codes can have more complicated LDPC encoding, mostly because matrix multiplication involving the generating matrix G and the message word requires a lot of computing power. A lot of work has gone into creating low-complexity LDPC encoding methods. These approaches, which seek to streamline the encoding procedure, frequently concentrate on effective strategies built around the parity-check matrix H. It is imperative that the LDPC encoding procedure be appropriate for a range of channels, taking into account diverse communication situations.

4.1 Circulant of H matrix

Each row in a circulant matrix is created by cyclically moving the row above it to the right. The matrix is square. A circular pattern is produced by the first row, which is a cyclic shift of the last row. Each column in a circulant matrix is the leftmost column's downward cycle shift. The final column's cyclic shift is represented by the first column. Weight (w) is the uniform weight of a circulant matrix. The number of non-zero components in each row or column is indicated by the weight, w. The circulant becomes a permutation matrix if the weight w equals 1.One particular kind of circulant matrix is a permutation matrix. The generator of the circulant is fully specified by the first row (or first column) of a circulant matrix. The generator determines the overall circulant structure. The set of columns (read from top to bottom) in a circulant matrix is identical to the set of rows (read from right to left).This symmetry simplifies the circulant's representation.

4.2 Codeword generation

$$\boldsymbol{H_{qc}} = \begin{pmatrix} A_{1,1} & \cdots & A_{1,2}, \ldots & A_{1,t} \\ A_{2,1} & \cdots & A_{2,2}, \ldots & A_{2,t} \\ & & & \\ A_{c,1} & \cdots & A_{c,2}, \ldots & A_{c,t} \end{pmatrix} \quad (4.1)$$

If the rank of a b×b circulant over GF(2) is b, then every row in the circulant is linearly independent. The number of rows that are linearly independent is indicated by the rank. The null space of an array of sparse circulants, all of the same dimension b×b, is used to create a QC-LDPC code. With c≤t, an array of b×b circulants over GF(2) is considered, with c and t being positive integers. The array's circulants are light in relation to their size (b).A small number of non-zero elements are present in each circulant.

4.3 Finding generator matrix

Examine the QC-LDPC code provided by the parity-check matrix's null space . if the rank of and cb are the same. Assumed to be set up so that the rank of the subsequent sub array, c x c of, is cb, the same as the rank of, are the columns of circulants of. Additionally, we assume that the (t-c)b information bits match the first (t-c)b columns of. The following is the desired generating matrix of:

$$Gqc = \left[\begin{array}{c|c|c} \begin{matrix} G_1 \\ G2 \\ \\ \\ G_{t-c} \end{matrix} & \begin{matrix} I\ 0\ldots\ 0 \\ 0\ I\ 0 \\ \\ \\ 0\ 0\ldots..I \end{matrix} & \begin{matrix} G_{1,1}\ G_{1,2}\ldots\ G_{1,c} \\ G_{2,1}\ G_{2,2}\ldots\ G_{2,c} \\ \\ \\ G_{t-c,1}\ G_{t-c,2}\ldots\ \ldots G_{t-c,c} \end{matrix} & \begin{matrix} I_{(t-c)b} \end{matrix} \end{array}\right] \quad (4.2)$$

The necessary and sufficient criteria for being a Cqc generator matrix is that [0] represents the zero matrix. Once we know,'s, we can form all the circulants G_{ij}'s of G_{qc}. As a result, is totally defined by a set of $c(t-c)$ circulant generators known as G_{qc} generators.

Let $u = (1,0,\ldots,0)$ represent the unit b-tuple with a "1" in the first position, and 0 = (0,...,0) represent the all-zero b-tuple. For example, the first row of the submatrix of Gi is $g_i = (0 \ldots\ldots.0\ u\ 0 \ldots 0\ g_{i,1}\ g_{i,2}\ldots.g_{i,c})$.

$H_{qc}\ gi^T = 0$.results in the following equality:

$$M_i u^T + Dz_i^T = 0$$

Which gives $$z_i^T = D^{-1} M_i u^T \quad (4.3)$$

Where, for $1 \leq i \leq -t-c$, z_i= ($g_{i,1}$ $g_{i,2}$ …….. $g_{i,c}$) (ie. The last c sections of) and the ith column of circulants of given by M_i = $[A_{1,i}^T \ldots\ldots A_{c,i}^T]^T$

Solving equation 1 we obtain , , , for, $1 \leq i \leq t-c$. From z_1 ,z_2,z_{t-c}we can find all , $g_{i,j}$ ‘s from which G_{qc}can be easily constructed.

4.4 The encoding procedure

The duty of creating a systematic codeword for the input message string (a) applied to the input block is handled by the encoding process. The following equation can be used to determine this process

$$C = aG \tag{4.4}$$

When implementing hardware, input is given one at a time. This compels us to express the preceding equation as:

$$C = a_i G_{i,j} = a_{(i-1)b+1} g_{i,j}^{(0)} + \cdots \ldots \ldots \ldots \ldots + a_{(ib)} g_{i,j}^{(b-1)} \tag{4.5}$$

Here are the steps to encrypt a message string:

1. The generator matrix register loads a new row on the positive going clock cycle edge.

2. The input, which consists of the message bit and the generating matrix, will be ANDed.

3. Next, the previous step output and the contents of the temporary output register are XORed.

4. Next, the previous step output and the contents of the temporary output register are XORed.

5. Until every row in the generator matrix has been traversed once, this operation is repeated. The process flow chart is shown in *Fig. 4.2* and Hardware architechture is given in *Fig. 4.1*.

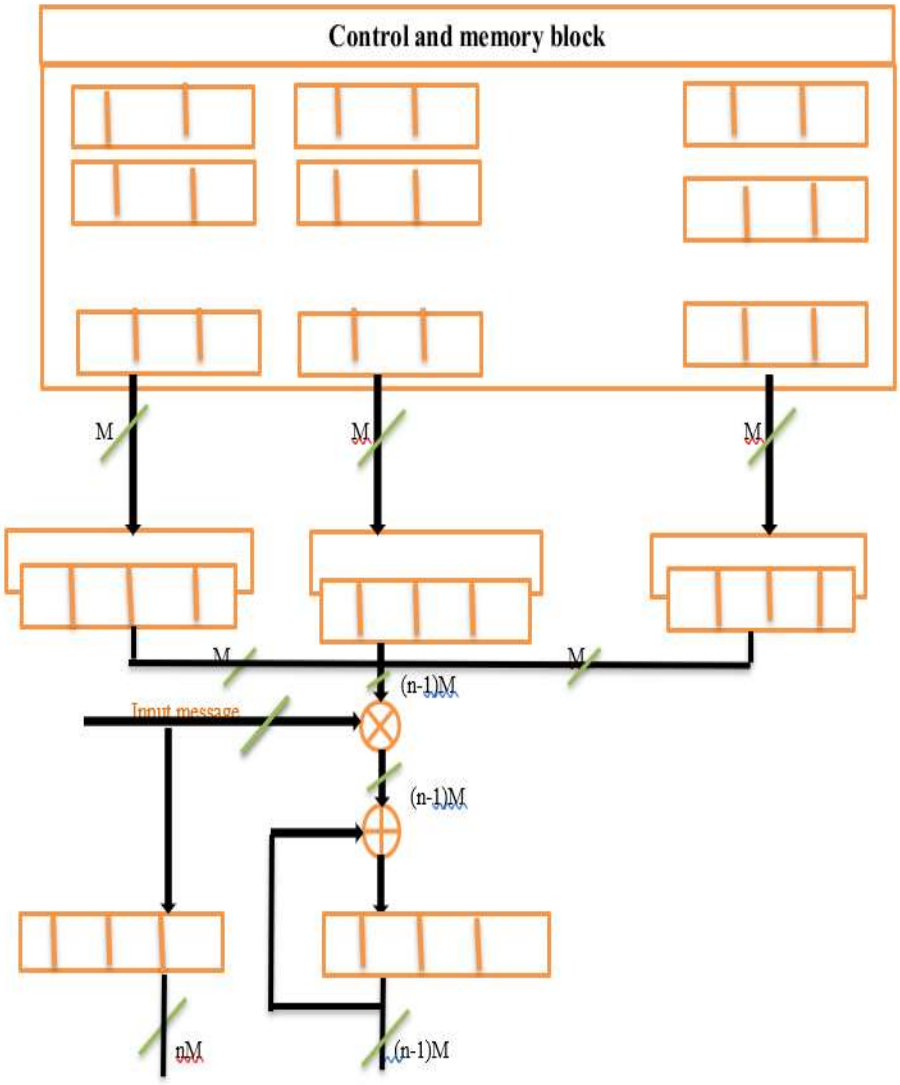

Fig. 4.1 Hardware Architecture

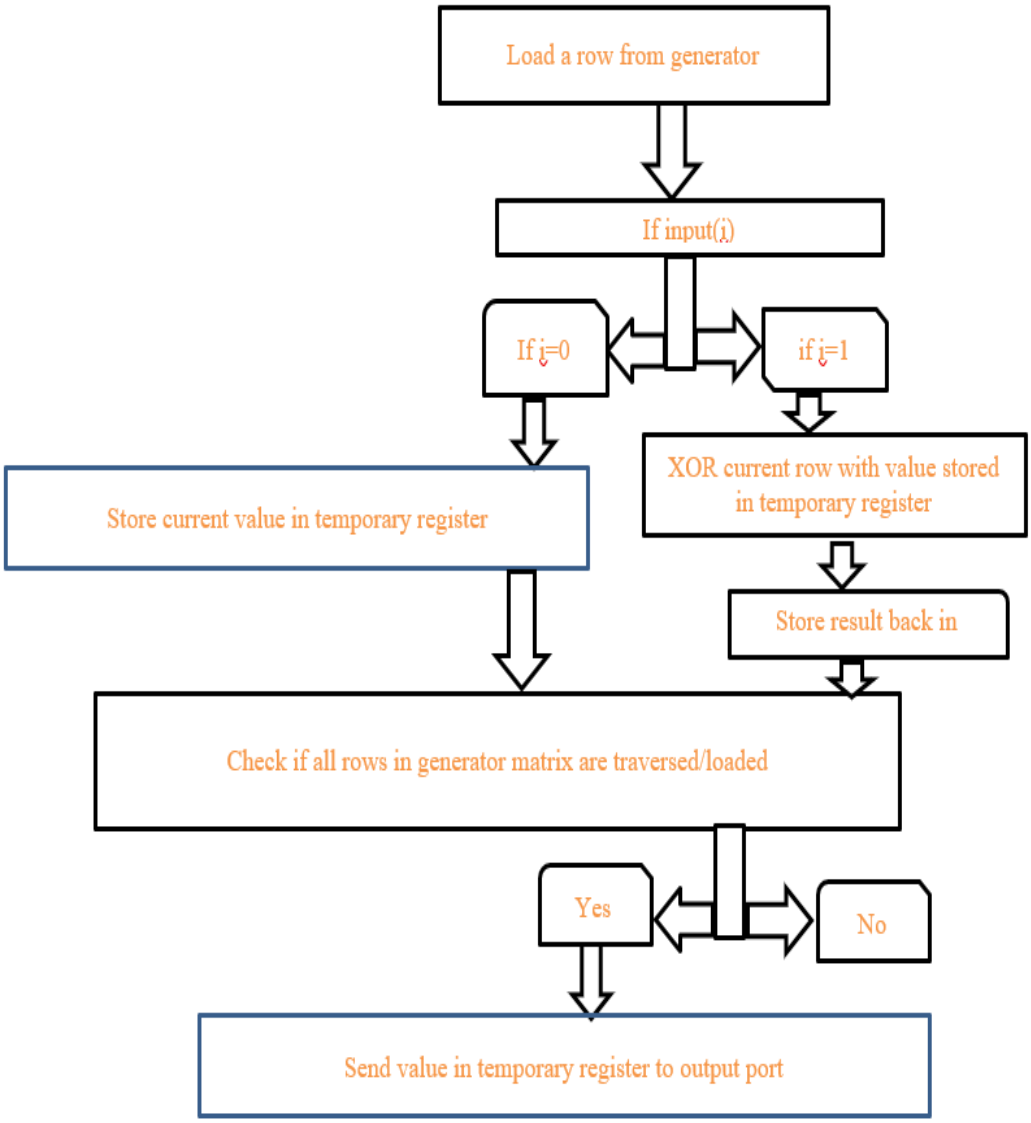

Fig. 4.2 Encoder flow chart

5 Results and Disscussion

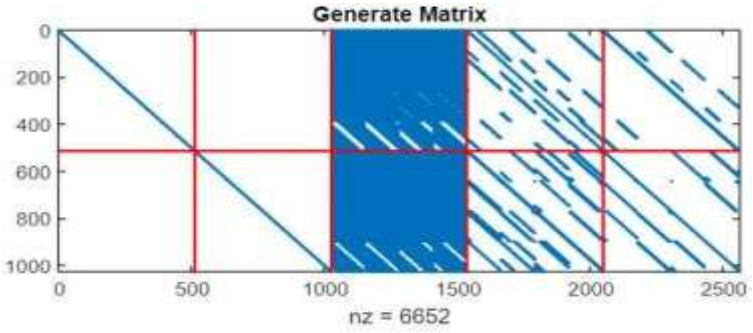

Fig 5.1 Generator matrix

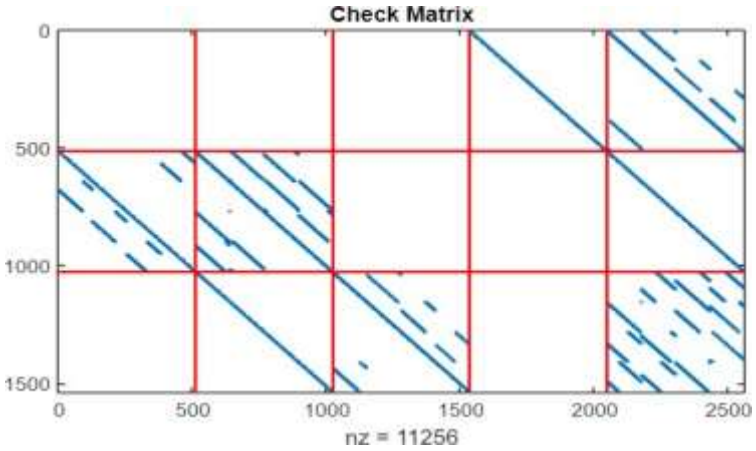

Fig. 5.2 Parity check matrix

Generator matrix and Parity check matrix are given in *Fig. 5.1* and *Fig. 5.2* .

Name	Value	Size	Class
BER	[0,0,0,0]	1x4	double
EbN0_dB	[1.5000,1.7...	1x4	double
FILE_NAME	'LDPC_CC...	1x23	char
FRAMES_NUM	100	1x1	double
G	1024x2560 ...	1024x2560	double
H	1536x2560 ...	1536x2560	double
HColNum	1x2560 dou...	1x2560	double
HRowNum	1x1536 cell	1x1536	cell
INFO_LENGTH	1024	1x1	double
MAX_ERROR_	200	1x1	double
MAX_ITER_NU	200	1x1	double
NORM_FACTO	0.8000	1x1	double
RATE	0.5000	1x1	double
SIZE_M	512	1x1	double
SNR	1.4125	1x1	double
SNR_dB	1.5000	1x1	double
ans	33	1x1	double
bitError	[0,0,0,0]	1x4	double
c_mark	7680x1 dou...	7680x1	double
encodeData	1x2560 dou...	1x2560	double
fid	13	1x1	double
frameError	[0,0,0,0]	1x4	double
iterNumTotal	[0,0,0,0]	1x4	double
message	1x1024 dou...	1x1024	double
nEbN0	1	1x1	double
nF	1	1x1	double
noise	1x2560 dou...	1x2560	double
r_mark	7680x1 dou...	7680x1	double
receiveSignal	1x2560 dou...	1x2560	double
rowH	1536	1x1	double

Fig. 5.3 output report from MATLAB

The H matrix code is responsible for generating the parity-check matrix (H) for LDPC codes based on the CCSDS standards. Iterates 26 times, corresponding to the 26 different submatrices used to construct the overall H matrix. The code handles LDPC codes with different information rates, specifically 1/2, 2/3, and 4/5. The information rate determines the efficiency of error correction. The H matrix is constructed using predefined vectors (theta and fai) and submatrices (C). These vectors are based on parameters specified in the CCSDS documents. The spy plots are used to visualize the sparsity pattern of the generated H matrix. The sparsity pattern is crucial for understanding the structure of the LDPC code.

References

[1] Liu J. and Feng Q., 'A Miniaturized LDPC Encoder: Two-Layer Architecture for CCSDS Near-Earth Standard,' in *IEEE Transactions on Circuits and Systems II*: *Express Briefs*, vol. 68, no. 7, pp. 2384-2388, July 2021.

[2] Mahdi A., Kanistras N. and Paliouras V., 'A Multirate Fully Parallel LDPC Encoder for the IEEE 802.11n/ac/ax QC-LDPC Codes Based on Reduced Complexity XOR Trees,' in *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 29, no. 1, pp. 51-64, Jan. 2021.

[3] Liu J. and Feng Q., 'A Miniaturized LDPC Encoder: Two-Layer Architecture for CCSDS Near-Earth Standard,' *in IEEE Transactions on Circuits and Systems II Express Briefs*, vol. 68, no. 7, pp. 2384-2388, July 2021.

[4] Wang Y., Jiang M. and X Ma ., 'Transmitting Extra Bits With Cyclically Shifted LDPC Codes,' in *IEEE Wireless Communications Letters*, vol. 10, no.12, pp. 2824-2827, Dec. 2021.

[5] Chen Y., *Cui H. and Wang Z., '*An Efficient Reconfigurable Encoder for the IEEE 1901 Standard*,' in IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 30, no. 9, pp. 1368-1372, Sept. 2022.

[6] Karimi B. and Banihashemi A. H., 'Construction of Irregular Protograph-Based QC-LDPC Codes With Low Error Floor,' in *IEEE Transactions on Communications*, vol. 69, no. 1, pp. 3- 18, Jan 2021.

[7] Sadeghi M. R., 'Optimal Search for Girth-8 Quasi Cyclic and Spatially Coupled Multiple-Edge LDPC Codes,' in *IEEE Communications Letters*, vol. 2, no. 9, pp. 1466-1469, Sept. 2019.

[8] Tao X., Xin Y., Wang B.and Chang L.,'Layered Construction of Quasi-Cyclic codes,'in *IEEE-Communications Letters*,vol.24,no. 5, pp. 946-950, May 2020.

[9] Tao X., Chen X. and Wang B., 'On the Construction of QC-LDPC Codes based on Integer Low Error Floor', in *IEEE Communications Letters*,vol.26 no. 10, pp.2267-2271,Oct.2022.

[10] Zhou M., Zhu H., Xu H., Zhang B. and Xie K., 'A Note on the Girth of (3, 19)-Regular Tannaer's Quasi Cyclic LDPC Codes,' in *IEEE Access*, vol. 9, pp. 28582- 28590,2021.