

FPGA Performance Analysis of LDPC and Turbo Codes for Communication System

Aakanksha Devrari

UPES: University of Petroleum and Energy Studies

Adesh Kumar (✉ adeshmanav@gmail.com)

University of Petroleum and Energy Studies, Dehradun India <https://orcid.org/0000-0002-0209-9206>

Amit Kumar

UTU: Uttarakhand Technical University

Research Article

Keywords: Turbo Codes, Low-Density Parity Check (LDPC) codes, Field Programmable Gate Array (FPGA), AWGN Channel, and BPSK Modulation

Posted Date: January 25th, 2022

DOI: <https://doi.org/10.21203/rs.3.rs-1288700/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

FPGA Performance Analysis of LDPC and Turbo Codes for Communication System

¹Aakanksha Devrari, ¹Adesh Kumar, ²Amit Kumar

¹*Department of Electrical and Electronics Engineering, School of Engineering, University of Petroleum & Energy Studies, Dehradun, 248007, India*

²*Faculty of Technology, Uttarakhand Technical University, Dehradun, India*

Corresponding Author: Adesh Kumar,

E-mail: adeshmanav@gmail.com

tel: +0135-2102690

The total number of words of the manuscript, including entire text from title page to figure legends: 6686

The number of words of the abstract: **204**

The number of figures: **16**

The number of tables: 3

Abstract - The wireless communication system is based on several coding schemes such as turbo codes, LDPC codes, convolutional, polar, and systematic, etc. The coding techniques should satisfy the hardware system requirements while machine and device communication is taken place. The turbo codes provide a good coding gain close to Shannon's limit, whereas LDPC codes have the ability to provide error corrected data over a noisy channel. The research article presents the comparative performance analysis of turbo and LDPC coding hardware architecture. The encoder and decoder hardware chip of turbo and LDPC is designed using Xilinx ISE 14.7 software, targeted Virtex- 5 FPGA. The performance of both coding methods is evaluated using iterative coding scheme. The FPGA hardware complexity is analyzed in terms of hardware and FPGA performance parameters such as slices, flip flops, LUTs and IoBs utilization. The performance the coding methods are also analyzed in terms of timing information related parameters such as path delay, minimum duration, minimum and maximum time of the clock signal, etc. The research work is very much helpful for 4G and 5G mobile communication requirements in device to device communication.

Keywords: Turbo Codes, Low-Density Parity Check (LDPC) codes, Field Programmable Gate Array (FPGA), AWGN Channel, and BPSK Modulation

1. Introduction

Recent communication systems are facing difficulty in order to satisfy the need and requirements of users. The next generation mobile communication systems are expected to be 5G Era with improved channel coding schemes. There are multiple channel coding schemes available such as Low-density parity check codes (LDPC), turbo codes, systematic convolutional, polar codes and non-systematic convolutional codes [1, 2]. The codes can be analyzed based on their complexity, reliability, flexibility and latency as per system requirements. In cellular systems channel coding schemes are used to ensure reliable transmission by offering minimum errors during transmission. Wireless systems have shown remarkable development in satisfying user demands for the last few years. In current scenario, the coding schemes available for 5G mobile systems are LDPC, turbo codes and polar codes.

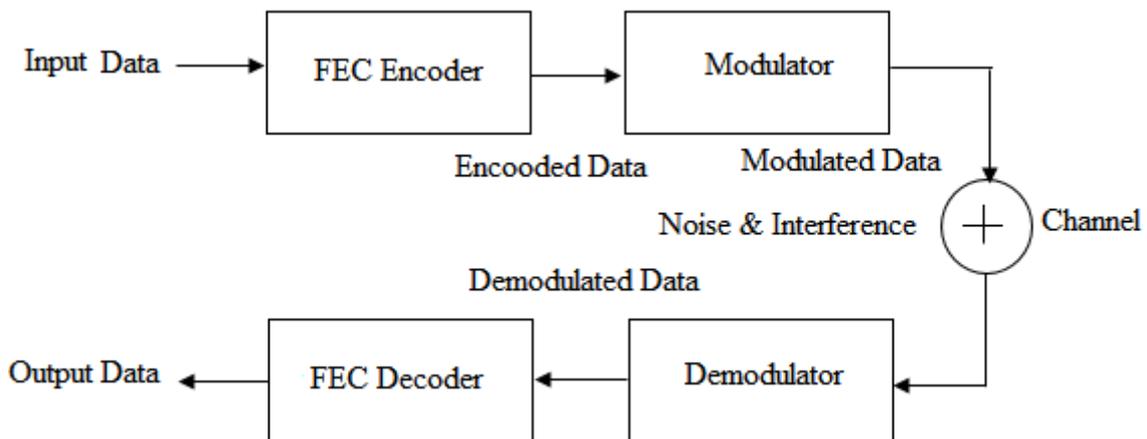


Fig. 1 Block diagram of FEC communication system

The Fig. 1 presents a forward error correction (FEC) communication system block diagram that shows the transmission of data from source to destination. The first step from source is a data encoding with the help of encoder and perform modulation using some modulation technique and then transmitted over Additive white Gaussian noise (AWGN) channel. During this transmission, noise can be added to data through the channel. This data along with noise is provided to the demodulator where carrier signal is separated from the message signal and the data is decoded back to its original form using error correcting decoder at the receiver end. The two most significant blocks in wireless communication system block diagram are encoder and decoder.

The turbo codes are a class of high performance FEC code invented by C. Berrou et al [4, 5] in 1993. The performance of the codes is close to Shannon's theoretical limit (SHA). The theorem

presents the channel capacity (C) in bits/s, to transmit information at a faster rate without error over channel bandwidth (B). The channel capacity, $C = B \log_2 (1+S/N)$, in which S/N presents signal to noise ratio. The coding community was excited to provide the promising solutions about channel capacity using an iterative decoding method based on simple constituent code. The coding block consists of the interleaver and code length is long and random in nature, can emulate the performance of the system. The convolution turbo codes are formed by parallel-concatenated recursive and systematic convolution codes, separated by an interleaver.

The LDPC codes were first introduced in the early 1960s by Gallager [28, 29]. This discovery was unnoticed by many investigators for nearly 20 years, but in 1981, a scholar named Tanner gave a new understanding of these codes from a graphical point of view. LDPC codes based on the Belief Propagation Algorithm (BPA) [26] has shown a channel's capacity performance close to Shannon's limit [7]. Consequently, LDPC codes proved powerful relative to other class of linear block codes named turbo codes for error corrections where reliability is the concern. On the other hands, turbo codes are very efficient in terms of their performance reaching the Shannon's limit [10].

2. Related Work

Andrade, J. et.al [1] presented a new class of LDPC decoder namely wide pipeline LDPC decoder for WiMAX standard 802.16e. This pipelined decoder architecture uses a high end synthesis tool to reduce validation time. The architecture passes the minimum throughput requirement for large scale integration approaches. Also, the use of 8-bit fixed point arithmetic adds extra precision in the design and delivers better performance in terms of bit error rate. **Albayrak, C. et.al** [2] presented the utilization of Belief propagation and log likelihood ratio based demodulation methods of LDPC codes to Ruby transform based decoder structure so that the computational complexity is reduced. The simulation results for both algorithms in presence of AWGN channel is obtained in terms of bit error rate. The Monte Carlo based evolution method is used for theoretical analysis. It is found that the combined approach of these two algorithms reduces the hardware complexity and improves the SNR. **Boncalo, O. et.al** [6] presented a new method for the construction of LDPC codes based on progressive edge growth mechanism. In order to achieve optimum a pipelined structure and memory organization using a single port bank is considered at the time of code construction. The hardware efficiency should be maximized using code constraints for quasi cyclic LDPC code. The design is implemented in FPGA and throughput is increased from 39 to 110% using pipelined structure.

Delomier, Y et.al [10] presented a generic behavioral model for generating LDPC decoders. The decoders are needed in applications that require reliable and fast data transmission. On the other hands, coded RTL architecture provided good performance, but path to IP design is slow down. The LDPC decoders are implemented using Xilinx Vivado HLS. It is observed that the hardware complexity is reduced by ten times and the throughput is increased by 1.5 times. The model gives performance similar to handcrafted RTL structures. **Dobkin, R. et.al [12]** presented a parallel architecture offering low latency for turbo decoding that consist of multiple single input single output elements. This architecture is compared with existing sequential architectures. For concurrent execution of the design, SISO based parallel interleaver and related algorithm is presented. The parallel architecture reduces the latency by 20 times and increases the throughput by six times in comparison to other sequential decoders. **Gilbert, J. M. et.al [16]** presented the implementation of linear, convolution and cyclic codes on FPGA using VHDL. Error control codes (ECC) are used for the purpose of error detection and correction in the noisy communication channel. This is achieved by adding redundant bits to the data at the transmitter end, and correlation of these redundant bits is used at the receiver end for error detection and error correction. **Hajiyat, Z. R. M et.al [17]** presented a new solution to channel coding schemes for a 5G mobile communication system having problems in satisfying the user requirements in machine-type communication. This work verifies various channel coding schemes such as LDPC, polar, systematic convolutional, turbo, and non-systematic convolutional codes, using BPSK modulation method in machine-type communication for 5G wireless communication system. **Kang. J et.al [18]** presented two different categories of Quasi Cyclic LDPC code out of which one is binary and other is non binary, and their construction is based upon finite field subgroups. These codes gives better performance when subjected to AWGN channel using iterative decoding algorithm. These code utilize large minimum distances as compared to finite geometry LDPC code and it balances the performance in terms of error and decoding complexity in case of the iterative decoding process. These codes provide burst errors in such environment and capable of replacing Reed Solomon codes by offering large coding gains at same code length and code rate. **Li, J. et.al [20]** presented a combination of Log-MAP decoding algorithm and a LUT-Log-MAP algorithm called LUT-Nor-Log-MAP algorithm. Also, a normalization functional unit is used for the SISO decoder unit. In simulation results, it has been observed that the resource utilization can be saved by 2.1 % using LUT-Nor-Log-MAP algorithm. Also, decoder using LUT-Nor-Log-MAP algorithm depicts a gain of 0.25~0.5 dB. Finally, while designing of turbo decoder a throughput of 36 Mbit/s can be achieved using the Cyclone IV FPGA platform. **Loi, K. C et.al**

[21] presented the implementation of LDPC decoder architecture on FPGA for application of digital video broadcasting via satellite (DVB-S2). This algorithm is applied systematically in such a manner that can supports 360 functional units of the design. Moreover, the synthesis of LDPC decoder is done on targeting two FPGA devices XC2VP100 and XC6VLX240T. **Lu, J. et.al** [22] presented a new category of LDPC code named partition and shift LDPC. The construction of this code is achieved using shift and divide the bit nodes and check nodes into subsets and connecting these nodes in subsets. A theorem is derived to prevent the cycles harmful in LDPC decoding. The simulation results of these codes depict good performance in terms of bit error rate over EPR 4 channels. **Maier, A. J. et.al** [24] presented flexible designs based on an open platform that require parallel computing method to decode the data based on an iterative minimum sum algorithm for a (3,6) regular LDPC code with variable code word length. The decoding algorithm supports parallelism for which Altera Offline compiler version 15.1 is required. This algorithm is tested on Altera Stratix VGXA7 FPGA hardware. This design gives an effective throughput of 68.22 Mbps for a 2048 length (3, 6) regular LDPC code at a clock frequency of 163.88 MHz and for a length of 1024 (3, 6) LDPC code it gives a throughput of 54.8 Mbps by showing an improvement of 7 Mbps. **Manjunatha, K. N et.al** [25] presented the design and implementation of the turbo decoder using Verilog. The design uses max-log algorithm that reduces the number of iterations and provides early termination due to which power consumption is reduced. In order to provide early termination sign difference ratio factor is used. The concept of clock gating is used to provide better power efficiency. The design is implemented and tested on Virtex-4 and Virtex-5 FPGA. **Orten, P et.al** [27] evaluated the performance of convolution codes in Rayleigh fading channel with long constraint length. As the sequential decoding cannot be applied below a threshold value of SNR. Therefore, the selected SNR per bit information is 5.7 dB. Moreover, the process of uniform quantization is chosen in such a way that it can provide negligible loss. For this design, finite interleaving method is used to an interleaver depth of (50 × 50) at a Doppler frequency of 0.01. The suggested scheme performs well in terms of bit error rate as compared to Turbo codes. **Tiwari, H. D et.al** [34] presented LDPC codes in which parity check matrix construction is done using sub matrix structure and matrix inverse is replaced by multiplication of sparse matrices. The replacement of matrices can result in efficient encoding that further reduces computational complexity. The simulation results depicted that the design has increased the throughput upto 1 Gbps. **Wang, Z et.al** [36] presented (8176, 7154) Euclidian geometry based architecture of a low complexity Quasi cyclic LDPC codes with high speed data rate. The architecture incorporated algorithmic transformation based critical path reduction and non-uniform

quantization method to minimize memory size. The parallel decoders are proposed in the design to enhance throughput. The design is implemented on Xilinx Virtex-II 6000, FPGA. The FPGA synthesis has proved that the design can achieve maximum throughput of 172 Mbps in 15 iterations. **Yan, Z. et. al [37]** presented a 3rd generation partnership LTE standard based on parallel turbo coding scheme that supports 188 block size data. In order to reduce the hardware complexity of interleaver in turbo codes, a permutation polynomial multistage network with address generator is used. An optimized decoding scheme is suggested to enhance the system performance and support high parallelism. A radix-2 and radix-4 recursion based add compare select unit is proposed for selection of block sizes that cannot be divided by 16. The design is implemented on 130 nm CMOS technology consuming 4.02 mm² core area and 1.81 architecture efficiency, while achieving 384.3 Mbps peak throughput with 5.5 iterations at a clock speed of 290 MHz. **Zuo, J. et.al [40]** presented an optimal design algorithm for irregular LDPC codes. This algorithm utilizes a shaping method with specified block length. It has been seen that LDPC codes are beating turbo codes in terms of computation cost and effective performance.

The problem statement of the research work is considered based on the hardware realization of turbo and LDPC based encoder and decoders. The comparative performance analysis of these coding based encoder and decoder architectures on FPGA platform will be the new research work so that LDPC coding based hardware can be used further for machine to machine and device to device communication as in the 5G communication system. The paper focused on the design and FPGA implementation of the turbo and LDPC coding based hardware chip, perform simulation and estimate the comparative performance.

3. Turbo Codes

The turbo decoder is based on Soft Output Viterbi Algorithm (SOVA) in which two soft-in-soft-out (SISO) decoding is followed to exchange extrinsic information for the period of each iteration. In addition, the decoder uses Bahl, Cocke, Jelinek and Raviv (BCJR) algorithm [3], based on Maximum Posteriori Probability (MAP) algorithm. The Viterbi algorithm is used for the sequence with finite or infinite length, whereas, MAP algorithm [11, 20] can only be used with sequences having finite length. The low complexity and bit error rate (BER) performance of turbo codes made this coding scheme good for 3G, 4G communication systems. Although these codes can be used for 3G and 4G systems, but their performance is limited for enhanced mobile broadband systems due to their complexity in implementing at larger block length and

code. Binary Phase Shift Keying (BPSK) is considered for modulation and demodulation at transmitting and receiving end for Additive White Gaussian Noise (AWGN) channel. Fig. 2 shows the block diagram of turbo coded digital communication system with BPSK modulation scheme on AWGN channel.

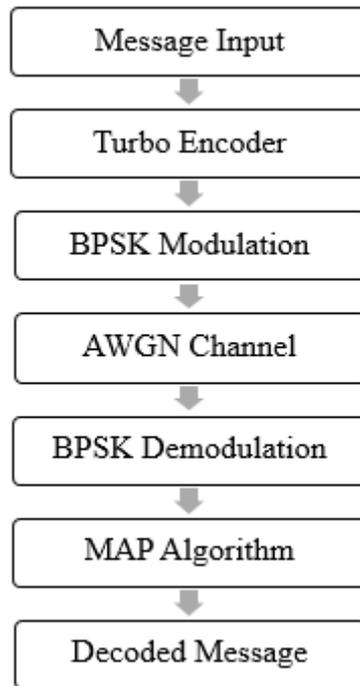


Fig. 2. Turbo Coding Process

3.1 Turbo Encoder

It is the class of linear block codes, which is constructed by parallel concatenation of two recursive blocks of systematic convolution encoders connected with the help of interleaver. The block diagram of turbo encoder, shown in Fig.3, consists of two Recursive Systematic Convolutional (RSC) encoders associated in parallel. The output of the turbo encoder consists of parity bits generated by the RCS encoder -1 and the parity bits generated by RCS encoder - 2. The two encoders produces encoded frames that includes a systematic frame and a parity frame. The RSC encoder-1 provides the forward path transfer function $(1 + D + D^3)$ and RSC encoder-r provide backward path transfer function $(1 + D^2 + D^3)$. The code rate of each RSC is $(r = 1/3)$, with $P_{1,i}$ and $P_{2,i}$ parity with length of total $3N$ bits. The encoder performs with $(M = 8)$ state transition diagram as depicted in Fig. 4. The initial state of the encoder is $(S_0 = '0')$ and subsequent states are $S_i \in \{0, 1, 2, \dots, M-1\}$ for $S_i \in \{0, 1\}$ respective message bit so that there is a potential change in state from S_{i-1} (previous state) to S_i (current state).

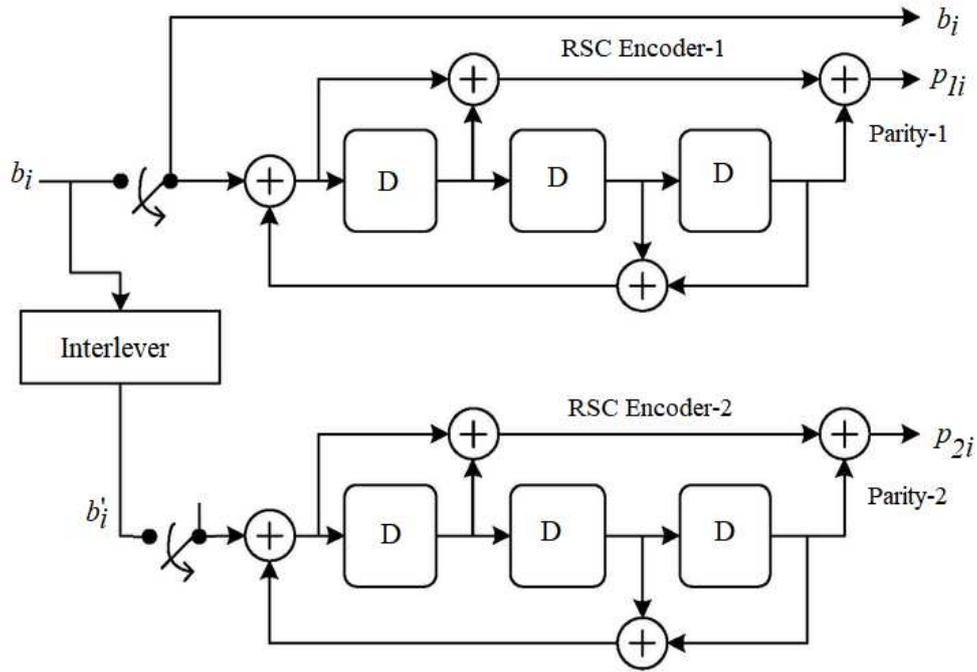


Fig. 3 Turbo Encoder

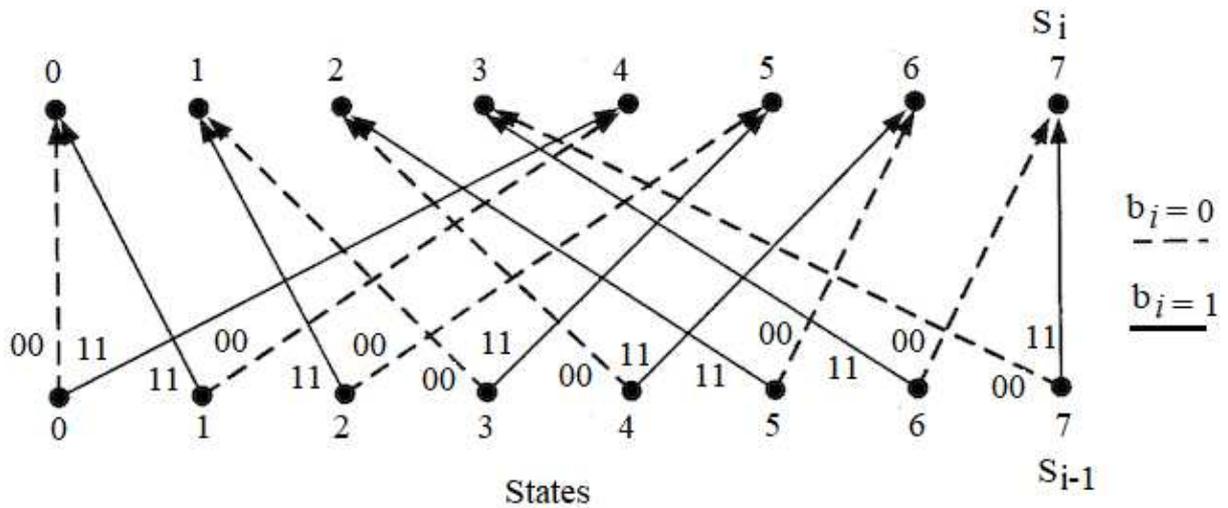


Fig.4. Trellis diagram configuration

The interleaver plays very important role in turbo codes. It is a random block arranges the data bit without repetition. The interleaver unit is used in both turbo encoder and turbo decoder. In the encoder, it produces the long block of data, whereas in decoder unit it corrects some errors after passing same data to first decoder. After it further interleaves the first decoded data and passes the same, though second decoder to correct the remaining errors. In this way, the process is repeated for a number of times.

3.2 Turbo Decoder

The decoder unit consist of the same number of decoders as used in encoder. The decoder works on an independent set of parity bits, but considers, as same information this arrangement is

Bahl et al [3] proposed the MAP algorithm in 1974, used to generate the probabilities of each bit and derive the extrinsic information. The LLR of the k^{th} symbol is given as

$$L(b_k|y) = \ln \frac{\sum_{b_k=1} \alpha'_{k-1}(s_{k-1}) \gamma'_k(s_{k-1}, s_k) \beta'_k(s_k)}{\sum_{b_k=0} \alpha'_{k-1}(s_{k-1}) \gamma'_k(s_{k-1}, s_k) \beta'_k(s_k)} \quad (3)$$

In this equation (3)

α'_k = Forward state matrix

β'_k = Backward state matrix

γ'_k = Branch matrix

s_k = Trellis diagram state at time 'k'

The MAP algorithm [30, 37] involves the additions and multiplication operations. For a larger sequence, the logarithm and approximation are applied, the eq. (3) is given as

$$\ln \left(\sum e^{x_i} \right) \approx \text{maximum} (x_i) \quad (4)$$

Then the algorithm is simplified to the equation

$$L(b_k|y) = \max_{b_k=1} (\alpha_{k-1} \gamma_k(s_{k-1}, s_k) \beta_k(s_k)) - \max_{b_k=0} (\alpha_{k-1} \gamma_k(s_{k-1}, s_k) \beta_k(s_k)) \quad (5)$$

The equation has addition and comparison operations. In the same way, the state matrix are also simplified and expressed as

$$\alpha_k(s_k) = \ln(\alpha'_k(s_k)) = \text{maximum} (\alpha_{k-1}(s_{k-1}) + \gamma_{k-1}(s_{k-1})) \quad (6)$$

$$\beta_k(s_k) = \ln(\beta'_k(s_k)) = \text{maximum} (\beta_{k-1}(s_{k-1}) + \gamma_{k-1}(s_{k-1})) \quad (7)$$

The branch matrix is expressed as

$$\gamma_k(s_{k-1}, s_k) = \left\{ \begin{array}{l} \gamma_{k00} = 0.5 (L_a(b_k) + L(d_{in}) + L(P_{k1})) \\ \gamma_{k01} = 0.5 (L_a(b_k) + L(d_{in}) - L(P_{k1})) \\ \gamma_{k10} = 0.5 (-L_a(b_k) - L(d_{in}) + L(P_{k1})) \\ \gamma_{k11} = 0.5 (-L_a(b_k) - L(d_{in}) - L(P_{k1})) \end{array} \right\} \quad (8)$$

Here,

L_a = Priori Probability

d_{in} = Systematic bit to success extrinsic information

P_{k1} = Parity bit

After calculating the LLR, the derived extrinsic information is presented as

$$L_e(b_k) = \mu. (L(b_k|y) - L(d_{in}) - L_a(b_k)) \quad (9)$$

Where μ is scaling factor, ($\mu < 1$) for extracting the extrinsic information in the MAP algorithm to compensate the losses in maximization of data.

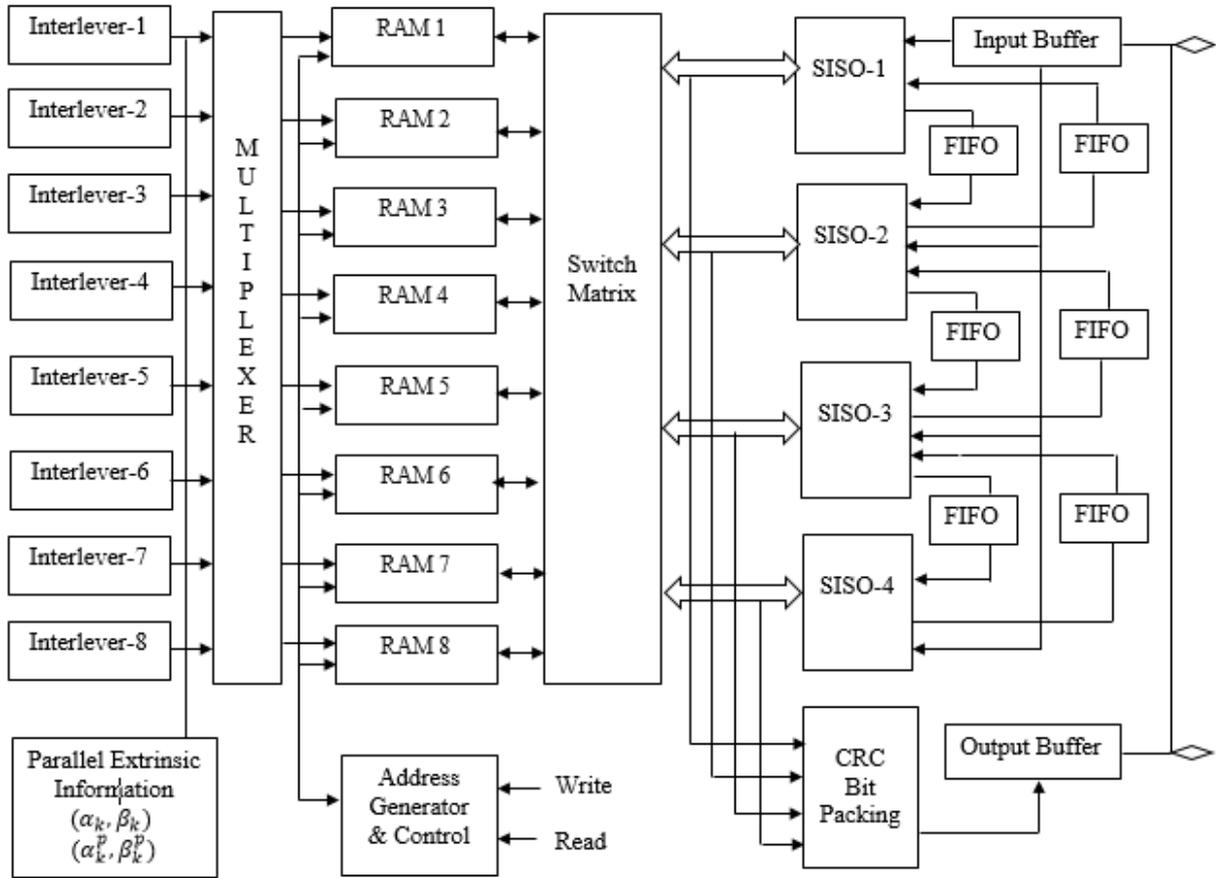


Fig. 7 Parallel execution and interleaving in turbo decoder

In the turbo channel coding scheme [33, 39], interleaver/deinterleaver blocks are structured in parallel in order to meet the decoding requirements. Memory contention is a challenge in interleaver. The Fig. 7 presents the parallel execution and interleaving in turbo decoder. The different interleaves are used to get extrinsic information and related parameters as forward variable (α) and backward variable (β). The interleavers are contention free and used for parallel execution based on block size. The interleavers/ deinterleavers are mapped to corresponding RAM memory with address generator and control unit. The address generator generates the RAM memory address for all RAM modules. For the interleaving process, the multiplexer selects real-time LLR and parallel extrinsic information [37] related to block RAMs. The write and read control signals are associated with RAM to write and read the contents sequentially of specific interleaver/deinterleaver memory. The address generator unit of interleaver/ deinterleaver is helpful to reduce the processing time and latency during each iteration. The switch matrices are used to choose the specific SISO modules and FIFO. One FIFO is attached to current SISO against interleaver/deinterleaver contends and another FIFO is for next SISO, in order to synchronize the values from input buffers and output buffers based on priority. As the

calculation of α, β , branch matrices and LLR are sequential. The input buffers and output buffers are used to maximize the throughput of the decoder.

4. LDPC Codes

LDPC code [1, 15] is a linear block code, consist of a parity check matrix ‘H’, which has less number of non-zero elements in every row and column. These codes can be regular as well as irregular, depending on the number of once available row wise and column wise. If the parity matrix has an equal number of once in each row and column, it is referred as regular LDPC code means columns weighs (w_c) is equal to row the weight (w_r). If parity check matrix ‘H’ has unequal no. of 1’s in each row and column it is irregular LDPC code. The codes are called low-density codes, as the number of 1’s are always less than the number of 0’s. In present communication systems such as mobile WiMax (802.16e), and DVB-S2 (802.11n), the LDPC codes are extensively used. LDPC codes can be represented as (n, w_c, w_r) , where n = code length, w_c = weight of column and w_r = weight of row. The parity-checking matrix ‘H’ is symbolized with the help of Tanner graphs and algebraic construction. The number of columns in ‘H’ matrix denotes the bit nodes or variable nodes(V_N) in Tanner graph, and the number of rows in H matrix denotes the check nodes (C_N) in Tanner graph and their connectivity is shown by logic ‘1’ in the ‘H’ matrix. An example of the H matrix representation is

$$H = \begin{bmatrix} C_{N1} \\ C_{N2} \\ C_{N3} \\ C_{N4} \\ C_{N5} \end{bmatrix} [V_{N1} \quad V_{N2} \quad V_{N3} \quad V_4 \quad V_{N5} \quad V_{N6} \quad V_{N7} \quad V_{N8} \quad V_{N9} \quad V_{N10}] \quad (10)$$

$$H = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{bmatrix} \quad (11)$$

The Fig.8 shows the tanner graph corresponding to the ‘H’ matrix. One method for LDPC encoding [32, 35, 38] is the pre-processing method in which for a given ‘H’ matrix a generator matrix is developed that is used for encoding some random input message bit (m) of size $(1 \times m)$. The second encoding technique uses parity check matrix directly and thus method is less complex compared to the first method. The decoding part involves bit-flipping algorithm for hard decision channels and sum-product process also called as a message passing algorithm for soft decision channels. This message-passing algorithm comprises passing of message in

forward and backward direction between variable nodes and check nodes until no further iteration is required.

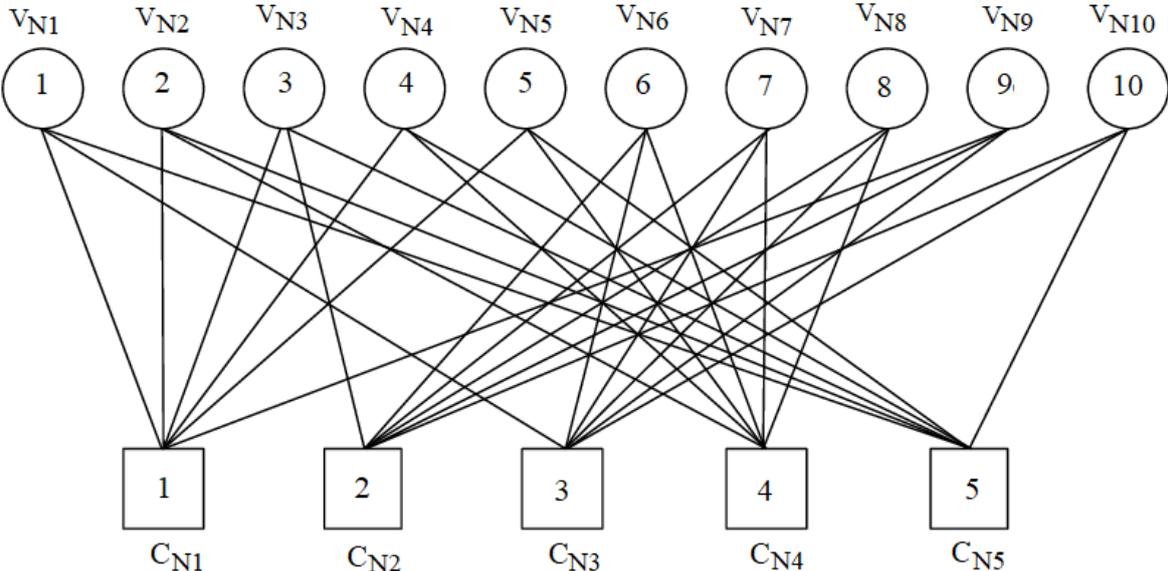


Fig.8 'H' matrix based example Tanner graph

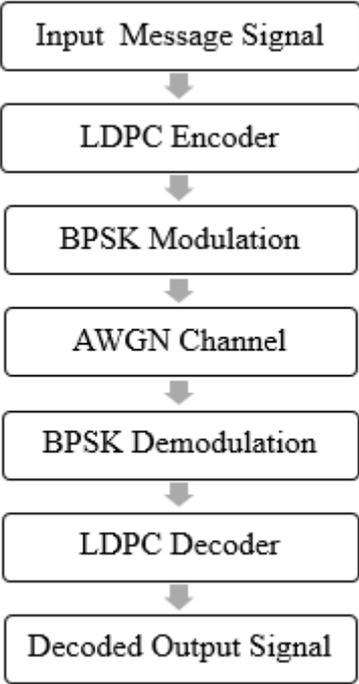


Fig.9 LDPC coding scheme

The LDPC coding scheme is presented in Fig. 9. The transmitter end shows random data bits that are modulated by BPSK modulator. LDPC encoder achieves the channel encoding and the encoded data sequence passes through the AWGN channel. The demodulation is performed

using BPSK demodulator and LDPC decoder [8, 9] is used to decode the data back to its original form at the receiving end.

Code word length plays very important role in encoding of LDPC codes. The encoding method comprises of mainly two operations: Sparse parity check formation and code word creation using sparse matrix. The decoding of LDPC codes uses iterative algorithms such as Belief Propagation algorithm [26]. A “Min- Sum” approach is adapted for the hardware implementation of Belief propagation algorithm. In this method after node, checking the information of a data node is updated for each iteration. Once the decoding process is over a hard decision is made based upon the most probable code word.

4.1 LDPC Encoder and Decoder

The LDPC codes proposed are used as cyclic codes considering the special case to conceptualize the parity check code. The generation of parity check matrix is performed using block circulant code formation. The block circulant code is considered here because of more accuracy in error correction also, it provides a structured architecture of the decoder. The parity check matrix is generated with the help of a binary matrix in which each row is created using a single cyclic right shift operation in previous row. The structure of a parity check matrix denoted by ‘H’ having size $(rT \times nT)$ is achieved by performing concatenation of $(r \times n)$ light circulant with size $(T \times T)$. The Tanner graph corresponding this parity check matrix is referred to as a protograph. In the design, the protographs [15] used are AR3A and AR4A with a code rate of $\frac{1}{2}$ are depicted in Fig. 10 (a) and (b) respectively.

The circles denote the variable nodes and the squares denote the parity check nodes. The open circles denote the penetrated symbols, whereas the solid circles represent the communicate symbols. Three level decoding process is considered: accumulate, recurrence by 3 or 4 and accumulate. The Protographs represent a circulant parity check matrix (3×5) in which the degree of consistent circulant is denoted by the number of parallel ends. The protographs cannot be extended directly without low weight code words, irrespective of the optimal selection of circulants. Therefore, protographs are only extended twice, but with small variation matrices, having a size (4×4) and (8×8) . The complete code can be built up using these circulants. In matrix representation, a dot is used for nonzero entries. For a formation of AR4A protograph the check nodes are placed in order (C_{N1}, C_{N2}, C_{N3}) and variable nodes in sequence $(4, 2, 1, 5, 3)$. The nodes are presented using solid lines initially with a (4×4) permutation and then elevating it to (16×16) circulants. The resulting construction is a highlighted (12×20) block

circulant. The concept is demonstrated using protograph example of AR3A and AR4A. In this case, the reorganization of the rows and columns in AR4A is done.

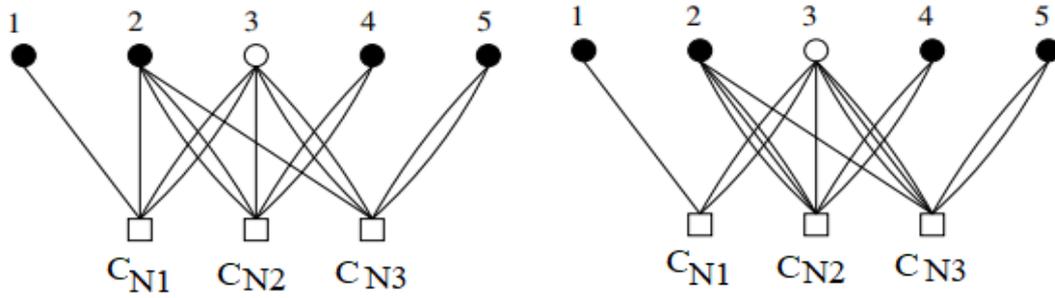


Fig.10 (a) Protograph (AR3A)

(b) Protograph (AR4A)

For the sequence (4, 2, 3, 1, 5), the base matrix (C_{N2}, C_{N1}, C_{N3}) is given as

$$H = \begin{bmatrix} 2 & 3 & 1 & 0 & 0 \\ 0 & 0 & 2 & 1 & 0 \\ 0 & 1 & 3 & 0 & 2 \end{bmatrix} \quad (12)$$

4.2 Hardware Architecture

The hardware chip implementation of LDPC encoder is based on block circulant matrix that can be created by the row wise cyclic shift performed to the contents of shift registers. The matrix is a group of $(n-k)$ encoder bits having a constraint length of T that uses a recursive convolution approach. The Fig.11 shows the feeding method where 'k' input bits are sequentially given to the encoder and then with every shift the contents of the registers are updated sequentially. Once, the updated data is stored in registers, switches are changed and data is read out from the registers along with the parity bits. The circular data is processed row-wise with $(n-k)$ soft operations. Fig. 12 presents the circular and shift operation to configure the encoder design.

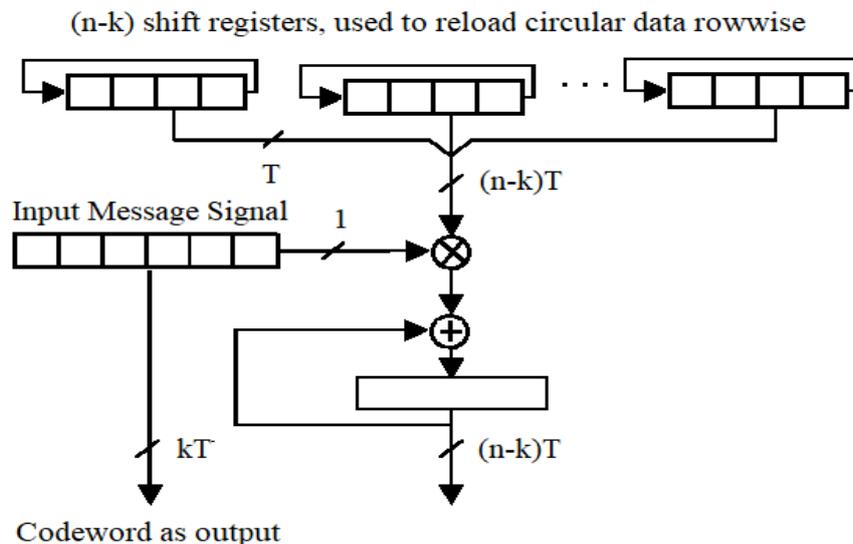


Fig. 11 Row-wise circular data processing with $(n-k)$ soft operations

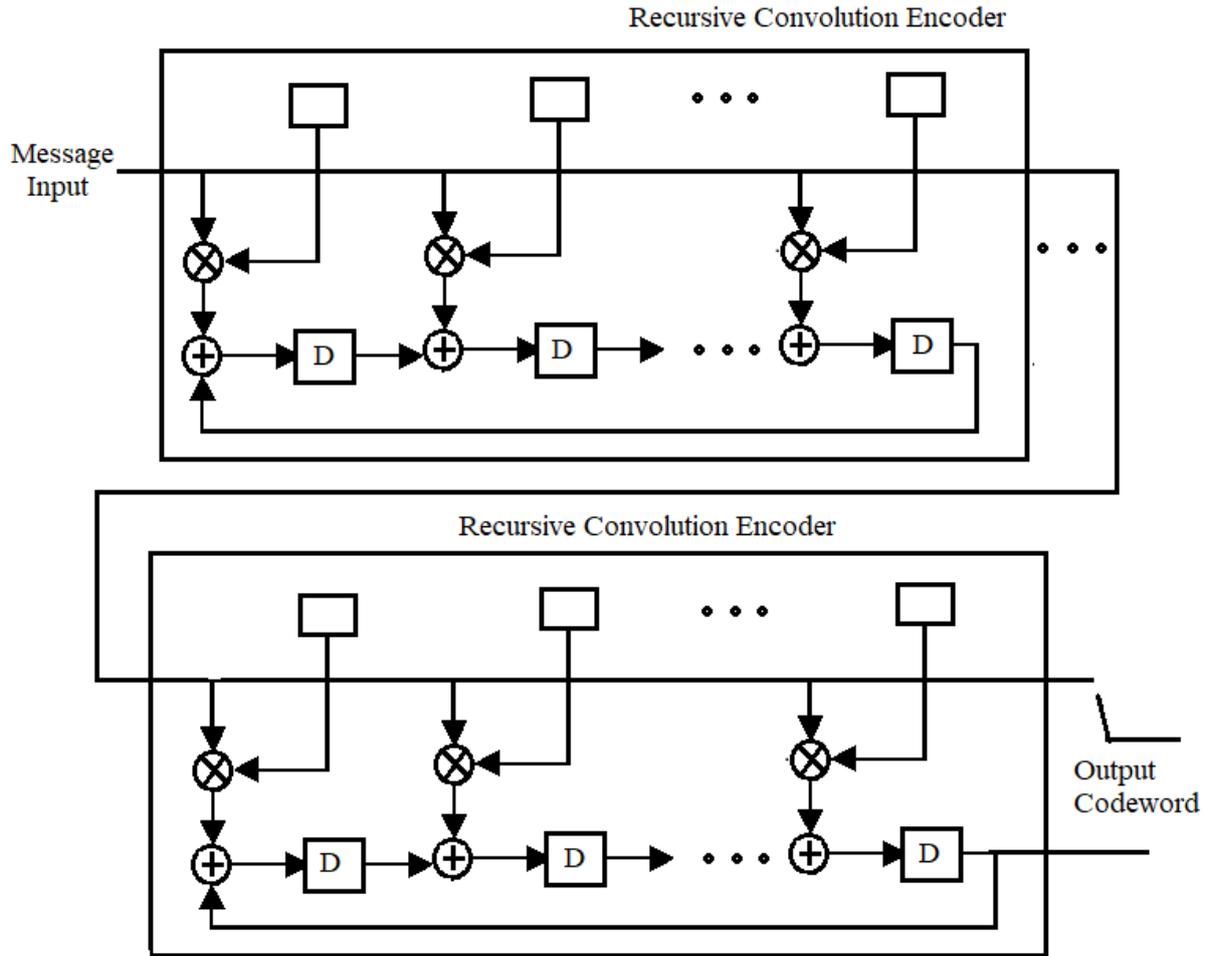


Fig. 12 Circular and shift register operation for encoder

The output of LDPC encoder is the multiplication operation performed on the input message matrix and generator matrix. For a larger message sequence, this operation becomes very complex. The following steps are involved in the LDPC encoding process.

Step-1: Compute the contents of parity check matrix 'H'

$$H = [-P^T : I_{n-k}] \quad (13)$$

Step-2: Compute the contents of Generator matrix (G)

$$G = [I_K : P] \quad (14)$$

Step-3: Determine the contents of transmitting code word (C_w)

$$C_w = m \cdot G \quad (15)$$

Circulant data encoding is used to select a parity check matrix (H) also indicates the sparse matrix with a code rate of 1/2. This size of 'H' matrix and 'G' matrix is (2048 x 4096) and (2048 x 4096) respectively. Since, the design and execution of such large data size in real time

is a challenging task. Therefore, a (32 x 64) matrix is used for the implementation of a large-scale matrix.

5. Results & Discussions

The chip design is done in Xilinx ISE 14.7 software. The Fig.13 presents the RTL view of turbo encoder and decoder. The Fig. 14 presents the RTL view of LDPC encoder and decoder. Table 1 lists the description of the pins used in the design of the encoder and decoder chip.

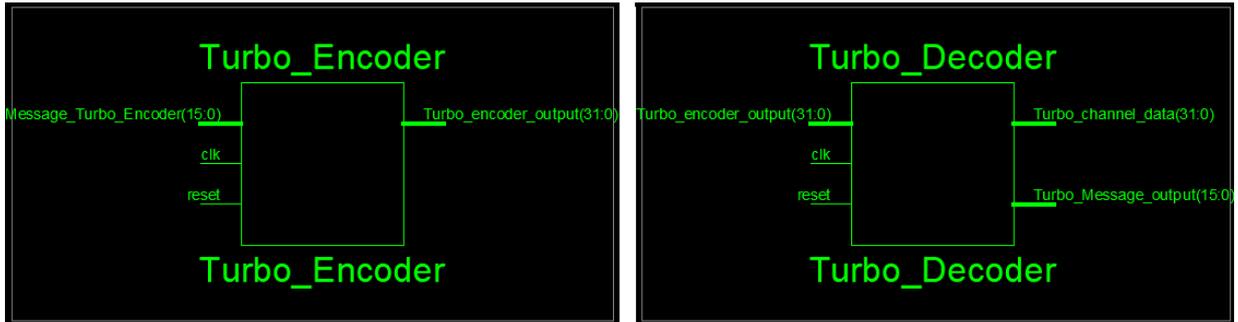


Fig. 13 RTL view of turbo encoder and decoder

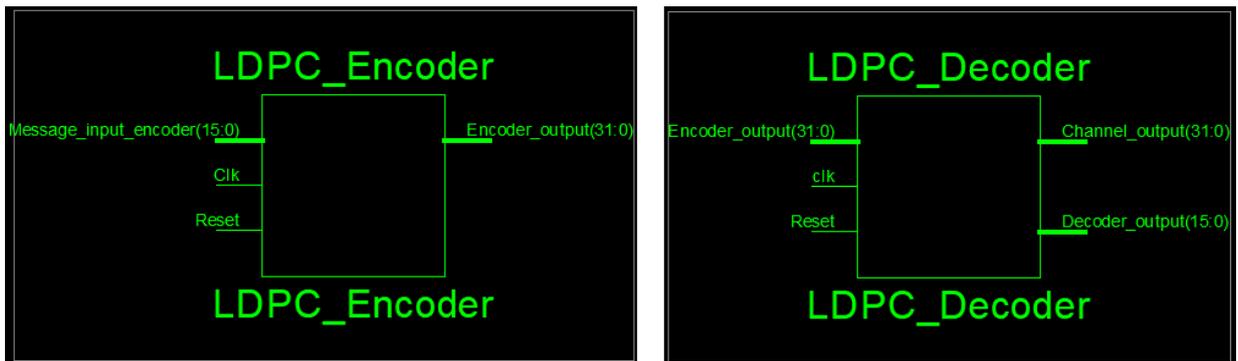


Fig. 14 RTL view of LDPC encoder and decoder

Table 1 Pin descripción of encoder and decoder designed RTL chip

Pins	Direction	Description
Clk	Input	The clock signal is an input clock signal applicable to offer the rising edge based clock signal, with 50 % duty cycle.
Reset	Input	Reset is the default input to the chip design and used to keep the contents zero for encoder and decoder.

Message_Turbo_Encoder <15:0>	Input	It presents the 16-bit input message to turbo encoder, given at the transmitted section in the design
Turbo_Encoder_output<31:0>	Inout	It is turbo encoded output of 32-bit, can be processed as input to AWGN channel.
Turbo_Channel_Data <31:0>	Inout	It is the output of the AWGN Channel and input to turbo decoder.
Turbo_Message_Output <15:0>	Output	It is 16-bit output of the turbo decoder as the decoded signal or message at the receiver end.
Message_Input_Encoder <15:0>	Input	It presents the 16-bit input message to LDPC encoder, given at the transmitted section in the design
Encoder_output<31:0>	Inout	It is LDPC encoded output of 32-bit, can be processed as input to AWGN channel.
Channel_Output <31:0>	Inout	It is the output of the AWGN Channel and input to LDPC decoder.
Decoder_Output <15:0>	Output	It is 16-bit output of LDPC decoder as the decoded signal or message at the receiver end.

The chip design of the LDPC code comprises the matrix multiplication of input message matrix and generator matrix 'G'. The size of the input message is considered of 16-bit, and the encoder output is of 32-bit, then the generator matrix is of size (16 x 32). The generator matrix requires 32 registers to store the contents. The matrix multiplication is done using AND logic and OR logic between input message (1 x 16) and generator matrix. The VHDL programming is used and corresponding test-bench are used to check the functional simulation. The output of turbo and LDPC encoder passes through AWGN channel. The behavior simulation of turbo and LDPC encoder and decoder chip is done in the Xilinx ISIM simulator. The Fig. 15 presents Xilinx ISIM simulation for turbo encoder and decoder. The Fig. 16 presents Xilinx ISIM simulation for LDPC encoder and decoder. The simulation test case stimuli are given.

Test-1 (Turbo Encoder/ Decoder): Apply Clk = rising edge clock signal, reset = '0', message_turbo_encoder [15:0] = "0000000011001100", then turbo encoder output is

turbo_encoder_output = “00000000000000001100110000000000”, turbo_channel_data <31:0> = “00000000000000001100110000000000” and decoded output is turbo_message_output <15:0> = “0000000011001100”.

Test-2 (Turbo Encoder/ Decoder): Apply Clk = rising edge clock signal, reset = ‘0’, message_turbo_encoder [15:0] = “0000000011111100”, then encoder output is turbo_encoder_output = “00000000000000001111110000000000”, turbo_channel_data <31:0> = “00000000000000001111110000000000” and decoded output is turbo_message_output <15:0> = “0000000011111100”.

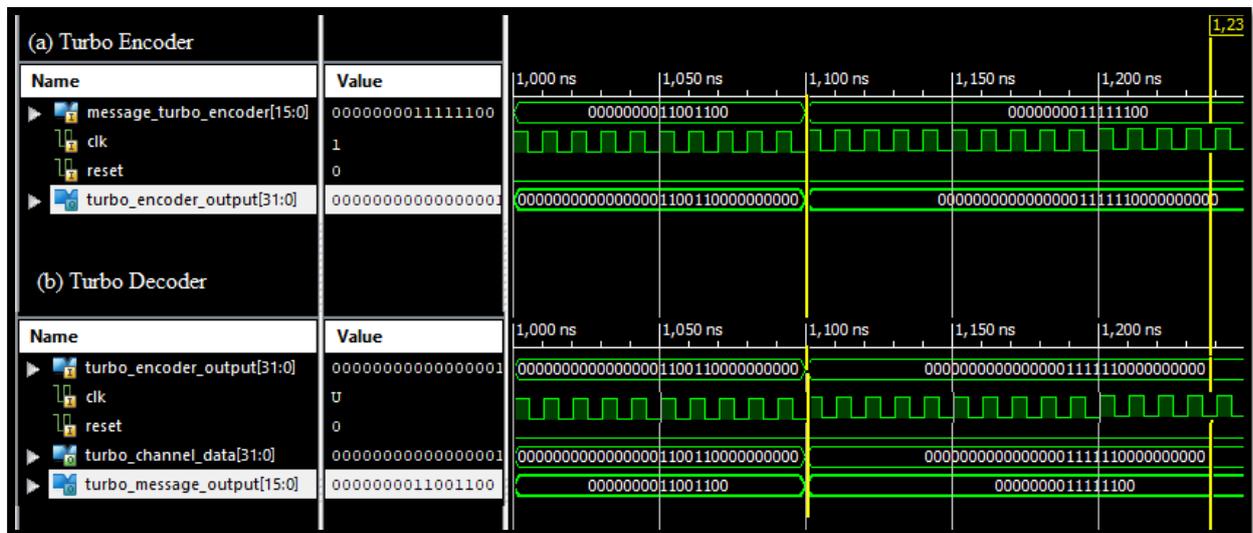


Fig. 15 Xilinx ISIM simulation for Turbo encoder and decoder

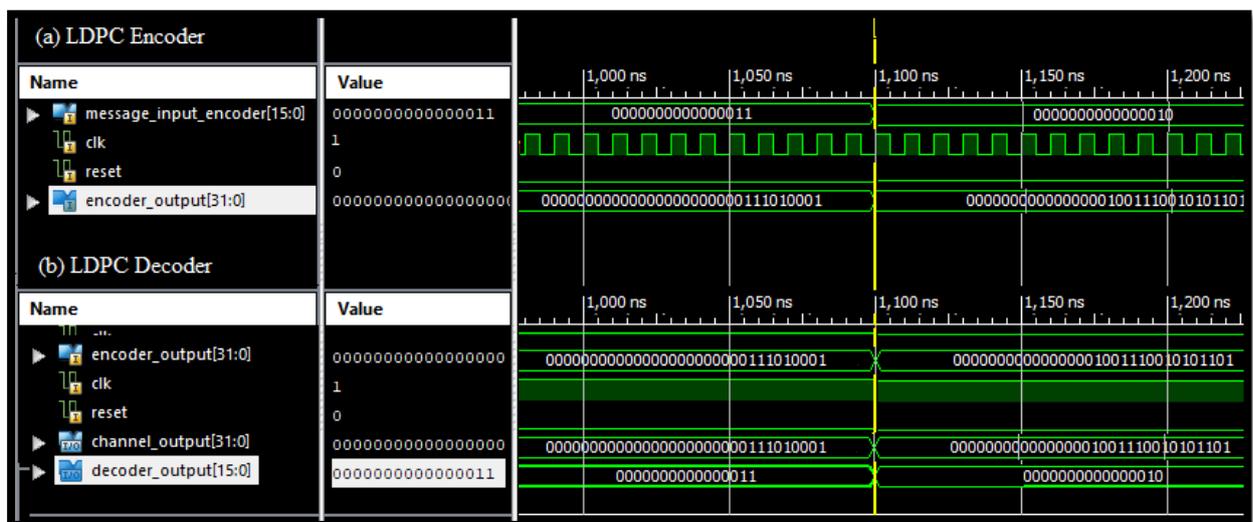


Fig. 16 Xilinx ISIM simulation for LDPC encoder and decoder

Test-3 (LDPC Encoder/ Decoder): Apply Clk = rising edge clock signal, reset = ‘0’, message_input_encoder [15:0] = “0000000000000011” then the encoder output is encoded_output = “00000000000000000000111010001”, channel_output <31:0> = “000000000000000000000000111010001” and decoded output of LDPC decoder is decoder_output <15:0> = “0000000000000011”.

Test-4 (LDPC Encoder/ Decoder): Apply Clk = rising edge clock signal, reset = ‘0’, message_input_encoder [15:0] = “0000000000000010” then the encoder output is encoded_output = “0000000000000001001110010101101”, channel_output <31:0> = “0000000000000001001110010101101” and decoded output of LDPC decoder is decoder_output <15:0> = “0000000000000010”.

Table 2 presents the hardware parameter summary of turbo and LDPC encoder and decoder hardware for different parameters targeting Virtex-5 FPGA in Xilinx ISE 14.7 software. The hardware parameters are slices, LUTs, flip-flops and IOBs. Table 3 presents the timing simulation results of turbo and LDPC encoder and decoder hardware for time related parameters such as minimum period (ns), minimum and maximum time before and after clock signal (ns), and maximum frequency.

Table 2 FPGA hardware utilization summary

Hardware	Turbo Codes		LDPC Codes	
	Encoder	Decoder	Encoder	Decoder
Slices	60/12480	145/12480	32/12480	110/12480
Flip Flops	72/12480	88/12480	45/12480	52/12480
LUTs	105/330	110/330	58/330	70/330
IOBs	50/172	72/172	50/172	72/172
GCLKs	1/32	1/32	1/32	1/32

Table 3 Timing information related parameters

Timing Parameter	Turbo Codes		LDPC Codes	
	Encoder	Decoder	Encoder	Decoder
Frequency (MHz)	310.0	315.0	375.00	390.0
Minimum period (ns)	1.615	1.365	1.053	1.275
Time (minimum) before clock (ns)	1.980	2.825	1.750	2.113
Time (maximum) after clock (ns)	4.174	4.250	2.710	2.914

Combinational path delay (ns)	7.769	8.640	6.513	6.302
Speed Grade	-5	-5	-5	-5

6. Conclusions

The hardware chip design and performance analysis of Turbo and LDPC codes for AWGN channel with BPSK modulation is done successfully in Xilinx ISE 14.7 software. The functional simulation and data communication with the designed hardware is verified using Xilinx ISIM simulation on target Virtex- 5 FPGA for device XC5V1x20t-2-ff32. The turbo encoder, hardware parameters such as slices, flip-flops, LUTs and IoBs are 60, 72, 105, and 50 respectively. The LDPC encoder hardware same parameters are 32, 45, 58 and 50 respectively. In the same ways turbo decoder has these parameters as 45, 88, 110 and 72 and LDPC decoder is having 110, 52, 70 and 72. It is analyzed that LDPC code is taking less hardware resources in comparison to turbo codes on FPGA. The turbo encoder and decoder supports 310.0 MHz and 315.0 MHz frequency respectively. In the same way, the LDPC encoder and decoder supports 375.0 MHz and 390.0 MHz frequency. It is clear that LDPC codes provide fast switching in comparison to turbo codes in FPGA hardware. Apart from this, LDPC codes provide optimal timing related parameters results in comparison to turbo codes. LDPC encoder and decoder are having combination path delay of 6.513 ns and 6.302 ns, respectively, whereas turbo codes are having the delay of 7.767 ns and 8.64 ns for encoder and decoder hardware respectively. It is analyzed that LDPC codes provide optimal solution in terms of hardware complexity, timing response parameters and high frequency support. LDPC codes are optimal, highly efficient and reliable solution in 4G and 5G wireless communication, digital broadcasting as their performance is very good in machine to machine communication and FPGA hardware.

Conflict of Interest

The authors declare no conflict of interest regarding the publication of this paper.

References

1. Andrade, J., Falcao, G., & Silva, V. (2014). Flexible design of wide-pipeline-based WiMAX QC-LDPC decoder architectures on FPGAs using high-level synthesis. *Electronics letters*, 50(11), 839-840.
2. Albayrak, C., & Turk, K. (2017). Reduced-complexity decoding of LT codes. *Wireless Personal Communications*, 94(3), 969-975.

3. Bahl, L., Cocke, J., Jelinek, F., & Raviv, J. (1974). Optimal decoding of linear codes for minimizing symbol error rate (corresp.). *IEEE Transactions on information theory*, 20(2), 284-287.
4. Berrou, C., Glavieux, A., & Thitimajshima, P. (1993, May). Near Shannon, limit error correcting coding and decoding: Turbo-codes. 1. in *Proceedings of ICC'93-IEEE International Conference on Communications* (Vol. 2, pp. 1064-1070). IEEE.
5. Berrou, C., & Glavieux, A. (1996). Near optimum error correcting coding and decoding: Turbo-codes. *IEEE Transactions on communications*, 44(10), 1261-1271.
6. Boncalo, O., Kolumban-Antal, G., Declercq, D., & Savin, V. (2018). Code-design for efficient pipelined layered LDPC decoders with bank memory organization. *Microprocessors and Microsystems*, 63, 216-225.
7. C. E. Shannon, (1948) "A mathematical theory of communication", Bell System Technology, Journal. vol. 27, pp (379-423).
8. Choi, G., Park, K. B., & Chung, K. S. (2018, November). "Optimization of FPGA-based LDPC decoder using high-level synthesis" In *Proceedings of the 4th International Conference on Communication and Information Processing* (pp. 256-259). ACM.
9. Devrari, A., Kumar, A., Chauhan, H., & Kumar, A. (2019) Design and FPGA Implementation of LDPC Decoder Chip for Communication System using VHDL, Vol 8(2), pp (200-206).
10. Delomier, Y., Le Gal, B., Crenne, J., & Jego, C. (2020). Model-Based Design of Flexible and Efficient LDPC Decoders on FPGA Devices. *Journal of Signal Processing Systems*, 1-19.
11. D. MacKay and R. M. Neal, "Near Shanon limit performance of low-density parity-check codes", *Electronic Letter*, (August 1996).
12. Dobkin, R., Peleg, M., & Ginosar, R. (2005). Parallel interleaver design and VLSI architecture for low-latency MAP turbo decoders. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 13(4), 427-438.
13. Dobkin, R., Peleg, M., & Ginosar, R. (2002, September). Parallel VLSI architecture for MAP turbo decoder. In *The 13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications* (Vol. 1, pp. 384-388). IEEE
14. Eizmendi, I., Velez, M., Gómez-Barquero, D., Morgade, J., Baena-Lecuyer, V., Slimani, M., & Zoellner, J. (2014). "DVB-T2: The second generation of terrestrial digital video broadcasting system" *IEEE transactions on broadcasting*, Vol. 60(2), 258-271.

15. Fang, Y., Chen, P., Wang, L., Lau, F. C., & Wong, K. K. (2012). Performance analysis of protograph-based low-density parity-check codes with spatial diversity. *IET Communications*, 6(17), 2941-2948.
16. Gilbert, J. M., Robbins, C., & Sheikh, W. (2019). FPGA implementation of error control codes in VHDL: An undergraduate research project. *Computer Applications in Engineering Education*, 27(5), 1073-1086.
17. Hajiyat, Z. R. M., Sali, A., Mokhtar, M., & Hashim, F. (2019). Channel Coding Scheme for 5G Mobile Communication System for Short Length Message Transmission. *Wireless Personal Communications*, 106(2), 377-400.
18. Kang, J., Huang, Q., Zhang, L., Zhou, B., & Lin, S. (2010). Quasi-cyclic LDPC codes: An algebraic construction. *IEEE Transactions on communications*, 58(5), 1383-1396.
19. Li, A., Hailes, P., Maunder, R. G., Al-Hashimi, B. M., & Hanzo, L. (2016). 1.5 Gbit/s FPGA implementation of a parallel turbo decoder designed for mission-critical machine-type communication applications. *IEEE Access*, 4, 5452-5473.
20. Li, J., Wang, X., He, J., Su, C., & Shan, L. (2019). Turbo Decoder Design based on an LUT-Normalized Log-MAP Algorithm. *Entropy*, 21(8), 814.
21. Loi, K. C., & Ko, S. B. (2011). Improvements on the design and implementation of DVB-S2 LDPC decoders. *Computers & Electrical Engineering*, 37(6), 1137-1146.
22. Lu, J., & Moura, J. M. (2005). Partition-and-shift LDPC codes. *IEEE transactions on magnetics*, 41(10), 2977-2979.
23. Luo, H., Zhang, Y., Li, W., Huang, L. K., Cosmas, J., Li, D., & Zhang, X. (2017). Low latency parallel turbo decoding implementation for future terrestrial broadcasting systems. *IEEE Transactions on Broadcasting*, 64(1), 96-104.
24. Maier, A. J., & Cockburn, B. F. (2017). Optimization of Low-Density Parity Check decoder performance for OpenCL designs synthesized to FPGAs. *Journal of Parallel and Distributed Computing*, 107, 134-145.
25. Manjunatha, K. N., & Meshram, V. A. (2017). Design and FPGA Implementation of Power Efficient Turbo Decoder for 4G LTE Standards. *International Journal of Applied Engineering Research*, 12(21), 10921-10925.
26. McEliece, R. J., MacKay, D. J. C., & Cheng, J. F. (1998). Turbo decoding as an instance of Pearl's "belief propagation" algorithm. *IEEE Journal on selected areas in communications*, 16(2), 140-152.
27. Orten, P., & Svensson, A. (2002). Sequential decoding of convolutional codes for Rayleigh fading channels. *Wireless Personal Communications*, 20(1), 61-74.

28. R. G. Gallager, "Low-density parity-check codes", IRE Transactions Information Theory, Vol. IT-8, pp. 21–28, (Jan. 1962).
29. R. G. Gallager, "Low-Density Parity-Check Codes", Cambridge, MA: M.I.T. Press, (1963).
30. Salmela, P., Sorokin, H., & Takala, J. (2008). A programmable Max-Log-MAP turbo decoder implementation. *VLSI Design, 2008*, pp (1-7).
31. Stirk, W., & Goeders, J. (2019, December). Implementation and Design Space Exploration of a Turbo Decoder in High-Level Synthesis. In *2019 International Conference on Reconfigurable Computing and FPGAs (ReConFig)* (pp. 1-5). IEEE.
32. Senthilvelan, M., Yu, M., Iancu, D., Sima, M., & Schulte, M. (2011). CORDIC instructions for LDPC decoding on SDR platforms. *Analog Integrated Circuits and Signal Processing, 69*(2-3), 191.
33. Shahabuddin, S., Janhunen, J., Juntti, M., Ghazi, A., & Silvén, O. (2014). Design of a transport triggered vector processor for turbo decoding. *Analog Integrated Circuits and Signal Processing, 78*(3), 611-622.
34. Tiwari, H. D., Tiwari, H. D., & Lee, K. Y. (2013). Designing parity check matrix to achieve linear encoding time in LDPC codes. *Digital Signal Processing, 23*(3), 994-1001.
35. T.J. Richardson and R.L Urbanke, "Efficient Encoding of Low-Density Parity-Check Codes", IEEE Transaction of Information Theory, vol. 47, pp. 638-656, February 2001.
36. Wang, Z., & Cui, Z. (2007). Low-complexity high-speed decoder design for quasi-cyclic LDPC codes. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 15*(1), 104-114.
37. Yan, Z., He, G., He, W., Wang, S., & Mao, Z. (2016). High performance parallel turbo decoder with configurable interleaving network for LTE application. *Integration, 52*, 77-90.
38. Z. Wang, Z. Cui, "Low-complexity high-speed decoder design for quasicyclic LDPC Codes," IEEE Transactions on VLSI Systems, vol. 15, no. 1, Jan. 2007.
39. Zhang, Xun, Hua Luo, Yue Zhang, Wei Li, Li-Ke Huang, John Cosmas, Dayou Li, and Carsten Maple. "Low latency parallel turbo decoding implementation for future terrestrial broadcasting systems." *IEEE Transactions on Broadcasting* (2017).
40. Zuo, J., Gui, L., & Zhang, H. (2015). Shaping Phenomenon and Optimal Design for Irregular LDPC Codes with Specified Block Length. *Wireless Personal Communications, 83*(1), 653-662.