A brief study on LDPC codes

[1]Ranjitha CR,[1] Jeena Thomas,[2] Chithra KR
[1]PG scholar,[2] Assistant professor,Department of ECE, Thejus engineering college
Email:cr.ranjitha17@gmail.com

Abstract: Low-density parity-check (LDPC) codes are a class of linear block LDPC codes. The name comes from the characteristic of their parity-check matrix which contains only a few 1's in comparison to the amount of 0's. This paper represents LDPC code characteristics, encoding and iterative decoding approaches to achieve channel capacity. Low-density-parity-check codes have been studied a lot in the last years and large progresses have been made in the understanding and ability to design iterative coding systems. The transmission quality is basically concerned with the probability of bit error at the receiver with respect to communication. This is an attempt to obtain highest capacity with minimum error rate by implementing modern codes named as LDPC(Low Density Parity Check codes). At the present time, LDPC codes has received a superior interest because of the error correction performance and world wide applications.

Keywords: LDPC, Encoding, Decoding, Construction, Tanner graph, Hard decision decoding, Soft decision decoding

INTRODUCTION

Error correction codes are one of the widely using tools available for achieving reliable data transmission in communication systems. For a wide variety of channels, the Noisy Channel Coding Theorem of Information Theory proves that the probability of decoding error can be made to approach zero exponentially with the code length if properly coded information is transmitted at a rate below channel capacity. It has been over 70 years since Claude Shannon published his famous "A Mathematical Theory of Communication", the foundation of the vast fields of channel coding, source coding and information theory, in which Shannon proved the existence of channel codes that are able to provide reliable communication as long as the code rate does not exceed the channel capacity. During the 1990s, the situation changed dramatically with the invention of Turbo Codes and the rediscovery of low-density parity-check (LDPC) codes , both of which have near-capacity performance. Coding schemes play an essential role in ensuring successful transmission of information, which is represented by a sequence of bits, from one point to another. In order to combat channel noise, coding strategy is devised that can construct codewords by adding redundancy to the transmitted bits, such that the original information can be perfectly decoded even with a certain number of errors. One of the most advanced classes of channel codes is the class of LPDC codes, which were first proposed by R.G Gallager in the early 1960s and rediscovered and generalized by MacKay et al. in the 1990s. As strong competitors to Turbo Codes, LDPC codes are well known not only for their near-capacity performance but also for their manageable decoding complexity[1]. More importantly, LDPC codes have some of the advantages of linear block codes, such as their simplicity and sparse (low-density) parity-check matrices which can be depicted as a graphical model called a Tanner graph (TG).
The name of LDPC code arrives from parity- check matrix concept which has only few one's when compared with zeros. Nowadays parallel architecture is also in use which will again increase the performance. Thus these codes are suited for implementation of current systems. The forward error correction codes are used more frequently on those days due to highly structured algebraic block and convolution codes. Nowadays LDPC codes are commonly used in Wi-max for microwave communications, CMMB i.e. china multimedia mobile broadcasting, Digital video broadcasting and for Wi-Fi standard. Low Density Parity Check (LDPC) codes gained significant research attention in current years due to their powerful decoding performance than turbo codes. All LDPC decoding algorithms are usually iterative in nature, so the performance and cost of using LDPC codes are partly determined by the choice of decoding algorithm. The decoding algorithm operate by exchanging messages between basic processing nodes. Design of power efficient LDPC encoders and decoders with low biterror rate (BER) in low signal-to-noise ratio (SNR) channels is critical for these environments.

REPRESENTATION OF LDPC CODES

The paper [2] explained, Low-density parity-check codes are error correction codes specified by a matrix containing mostly 0's and only a small number of 1's. Such a structure give both: a lower decoding complexity and good distance properties. Generally there are two methods to represent LDPC codes. Like all linear block codes they can be described via matrices and second method is a graphical representation. The parity check matrix is a mxn matrix with m number of rows and n number of columns. An example of parity check matrix is given in the figure 1.

$$H= \begin{pmatrix} 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \end{pmatrix}$$

Figure 1: Example of parity-check matrix[2]

We can now define two numbers describing these matrix. A w_r for the number of 1's in each row and w_c for each columns. For a matrix to be called low-density the two conditions $w_c \ll n$ and $w_r \ll m$ must be satisfied. In order to do this, the parity check matrix should usually be very large, so the example matrix given above can't be really called low-density. The graphical representation of the above parity check matrix is given in the figure 2.

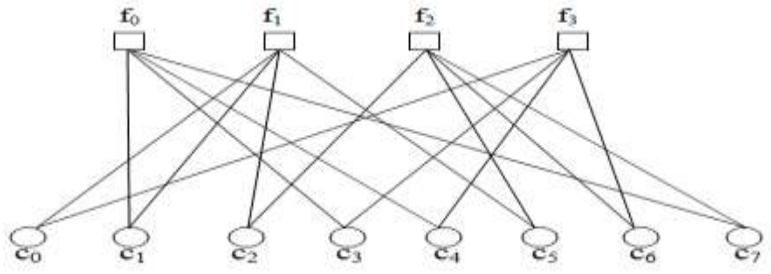

Figure 2.Tanner graph corresponding to the parity check matrix[2]

Tanner introduced an effective graphical representation for the LDPC codes. Tanner graphs are bipartite graphs. That means that the nodes of the graph are separated into two distinctive sets and edges are connecting nodes of two different sets[8]. The two types of nodes in a Tanner graph are called variable nodes (v-nodes) and check nodes (c-nodes). Considering the graph ,there is no connectivity between C_0 and F_0 hence the first place is having 0,1 in the second place means that connectivity lies between F_0 and C_1. Check nodes are specified as m nodes that are number of parity bits and variable nodes are n which are known as number of bits in code word.

If w_c is constant for every column and $w_r = w_c \cdot (n/m)$ is also constant for every row then the LDPC code is called a regular LDPC code. If H is low density but the numbers of 1's in each row or column are not constant the code is called a irregular LDPC code.

DIFFERENT ENCODING SCHEMES OF LDPC CODE

Encoding of codes, specially for higher block length codes can be quite difficult to implement in hardware but there are several methods for generating *H* such that encoding can be done via shift registers[7]. If the generator matrix *G* of a linear block code is known then encoding can be done using Parity check matrix H. The cost of the method depends on the Hamming weights i.e. the no of 1's in G. If the vectors are dense, then cost of encoding using this method is proportional to n^2. If *G* is sparse then this cost becomes linear with n. Here note that by performing Gauss-Jordan elimination on H to obtain it in the form a generator matrix *G* for a code with parity-check matrix H which can be found as per following[3].

$$H=[A \;\; I_{n-k}] \qquad (1)$$

Where A is (n-k)×k binary matrix and I is the size (n-k x n-k) identity matrix. The generator matrix is then

$$G=[I_k \;\; A^T] \qquad (2)$$

The message can encode into code words for LDPC Codes which requires the generation of parity check matrix H. The encoding method is through the use of a generator matrix, denoted by G. A code word C is formed by multiplying source input u by the generator matrix which is represented as

$$C=u*G \qquad (3)$$

The above explained method is the basic concept about the LDPC encoding. Many other methods are proposed by different authors to reduce the complexity of the LDPC encoding. Compared to turbo codes which is widely used in communication systems the main disadvantage of LDPC code is the complexty of encoding technique, in paper 'Efficient Encoding of Low-Density Parity-Check Codes'[11] by Thomas J. Richardson and Rüdiger L considered the encoding problem for LDPC codes specified by sparse parity-check matrices. The efficiency of the encoder arises from the sparseness of the parity-check matrix H and the algorithm can be applied to any (sparse) H. Although the example is binary, the algorithm applies generally to matrices H whose entries belong to a field F also assume that the rows of H are linearly independent. Assume given an mxn parity-check matrix H over F . By definition, the associated code consists of the set of n-tuples x over F such that

$$Hx^T=0^T \qquad (4)$$

Probably the most straightforward way of constructing an encoder for such a code is the following. By using Gaussian elimination H is converted to an equivalent lower triangular form as shown in Fig. 3.

n-m m

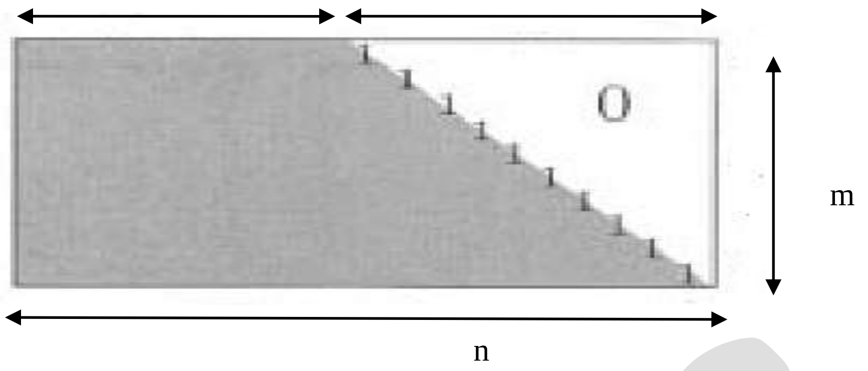

Figure .3. An equivalent parity-check matrix in lower triangular form.

Split the vector x into a systematic part s, sϵ F^{n-m} and a parity part p , pϵF^m such that x=(s,p). Construct a systematic encoder as follows:

i) Fill s with the (n-m) desired information symbols.

ii) Determine the m parity-check symbols using back-substitution.

More precisely, for l ϵ [m] calculate

$$P_l = \sum_{j=1}^{n-m} H_{l,j} s_j + \sum_{j=1}^{l-1} H_{l,j+n-m}\, p_j \qquad (5)$$

The complexity of this scheme is Bringing the matrix H into the desired form requires O(n^3) operations of preprocessing. The actual encoding then requires O(n^2) operations since, in general, after the preprocessing the matrix will no longer be sparse.

The proposed encoder in this paper is motivated by the above example. Assume that by performing row and column permutations only we can bring the parity-check matrix into the form indicated in Fig. 4. We say that H is in approximate lower triangular form. Note that since this transformation was accomplished solely by permutations, the matrix is still sparse. More precisely, assume that we bring the matrix in the form

$$H = \begin{pmatrix} A & B & T \\ C & D & E \end{pmatrix}$$

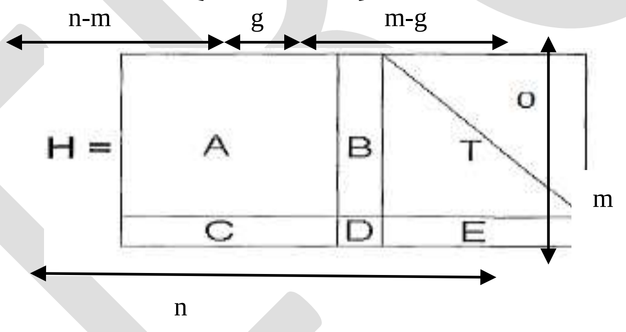

Figure 4. The parity-check matrix in approximate lower triangular form.

Where A is (m-g) x (n-m), B is (m-g) x g, T is (m-g) x (m-g), C is g x (n-m), D is g x g and E is g x (m-g) matrices. Further, all these matrices are sparse matrices and T is lower triangular with ones along the diagonal. Multiplying this matrix from the left by

$$\begin{pmatrix} I & 0 \\ -ET^{-1} & I \end{pmatrix} \qquad (6)$$

We get

$$\begin{pmatrix} A & B & T \\ -ET^{-1}A+C & -ET^{-1}B+D & 0 \end{pmatrix} \qquad (7)$$

Let x=(s,p_1, p_2) where s denotes the systematic part, p_1 and p_2 combined denote the parity part, p_1 has length g , and p_2 has length(m-g) . The defining equation $Hx^T=0^T$ splits naturally into two equations, namely

$$As^T+Bp_1^T+Tp_2^T=0 \qquad (8)$$

and

$$(-ET^{-1}A+C)s^T+(-ET^{-1}B+D)p_1^T=0 \qquad (9)$$

define Φ= $-ET^{-1}B$+Dand assume for the moment that Φ is non singular. then

$$p_1^T = -\Phi^{-1}(-ET^{-1}A+C)s^T \qquad (10)$$

Hence, once the g x(n-m) matrix - $\Phi^{-1}(-ET^{-1}A+C)s^T$ has been precomputed, the determination of can be accomplished in complexity O(g x(n-m))simply by performing a multiplication with this (generically dense) matrix. In the similar manner p_2 can also precomputed with less complexity. So the proposed system consist only two steps mainly, preprocessing and actual encoding. In the preprocessing step, we first perform row and column permutations to bring the parity-check matrix into approximate lower triangular form with as small a gap g as possible.

DIFFERENT DECODING SCHEMES OF LDPC CODES

There are different type of iterative decoding algorithms are used for decoding the LDPC codes. They are mainly classified as hard decision decoding and soft decision decoding respectively. The decision made by the decoder based on the received information is called a hard-decision if the value of bit can either be 0 or 1. If the decoder is able to distinguish between a set of values between 0 and 1, then it is called a soft-decision decoder[5]. The decoding of LDPC code is performed through iterative processing using the Tanner graph, to satisfy the parity check conditions. The condition $CH^T=0$ is the parity check condition, where C is the codeword and H is the parity check matrix. If $CH^T = 0$ then the received codeword is said to be valid, that is the received code word is error free.

A. HARD DECISION DECODING

Bit flipping algorithm is the best example for hard decision decoding. In bit flipping decoding the message would be binary, different from belief propagation decoding. In belief propagation decoding the probabilities of incidence of the code word bits constitute the message. The message and the edges in the tanner graph are passed together in bit flipping algorithm. The message send by the message node contain the information that the bits available at the message node is zero or one to the check node. The check node returns a response message to the message node. This response is initiated by using the parity check equation which is based on the modular sum of bits available at the check node is equal to zero. Let the code word be $C = [11001000]^T$ and the received cord word $Y = [10001000]^T$. This implies error occurred in C. The fig.2 illustrate the tanner graph, used for the decoding algorithm. The steps involved in the bit flipping algorithm is given below. Explained in[2]

Step1: All message nodes send a message to their corresponding check nodes connected to it. In this case, the message is the bit they believe to be correct for them. Here C_2 receives 0 (as per the codeword received Y) and it will send to f_1 and f_2. Similarly all message nodes will send messages to their corresponding check nodes as illustrated in the table 1.

Step 2: Every check nodes calculate a response to their connected message nodes using the messages they receive from step 1. The response from check node is calculated by using parity check equations which force all message nodes to connect to a particular check node to sum to 0 (mod 2). If sum of bits received is zero then the same bit which they received from the message node will send back. If it is not zero the the check node will flip the bit that received from message node and send back. Move to step 3.

Step 3: The message nodes use the messages they get from the check nodes and they received from transmitter to decide if the bit at their position is a 0 or a 1 by majority rule. The message nodes then send this hard-decision to their connected check nodes. Table2 illustrate this step.

Step 4: Repeat step 2 until either exit at step 2 or a assigned number of iterations has been passed.

Table 1: Overview of messages received and sent by the check nodes[2]

c-node	Activities
f_0	Received:C_2-0 C_4-0 C_5-1 C_8-0 Sent: 1-C_2 1-C_4 0-C_5 1-C_8
f_1	Received:C_1-1 C_2-0 C_3-0 C_6-0 Sent: 0-C_1 1-C_2 1-C_3 1-C_6
f_2	Received:C_3-0 C_6-0 C_7-0 C_8-0 Sent: 0-C_3 0-C_6 0-C_7 0-C_8
f_3	Received:C_1-1 C_4-0 C_5-1 C_7-0 Sent: 1-C_1 0-C_4 1-C_5 0-C_7

Table 2:Message nodes decisions for hard decision decoder.[2]

Message nodes	Y_1	Message from check node		Decision
C_1	1	f_2-0	f_4-1	1
C_2	0	f_2-1	f_1-1	1
C_3	0	f_2-1	f_3-0	0
C_4	0	f_1-1	f_4-0	0
C_5	1	f_1-0	f_4-1	1
C_6	0	f_2-1	f_3-0	0
C_7	0	f_3-0	f_4-0	0
C_8	0	f_1-1	f_3-0	0

The bit flipping algorithm can be classified and explained in the paper [12], 'Hard decision and soft decision algorithms of LDPC and comparison of LDPC with Turbo codes, RS codes and BCH codes'.

Weighted Bit-Flipping Algorithm(WBF):

The WBF algorithm finds the most unreliable message node of each individual check. The magnitude the received value y_i determines the reliability of the hard decision z_i, the least reliable message node's magnitude for each individual check during the algorithm's first step is given by:

$$Y_m^{min} = \min_{n\epsilon N(m)} |yn| \quad (11)$$

where |yn| denotes the absolute value,ie, the magnitude of the nth message node's soft value while Y_m^{min} is the lowest magnitude of all message nodes participating in the mth check. In the iterative WBF process,the bit sequence z obtained by hard decision is multiplied with the transpose of H matrix, and the resultant syndrom vector s is derived. For each message node at the position n, the WBF algorithm computes:

$$En = \sum_{m\epsilon M(n)} (2sm-1) y_m^{min} \quad (12)$$

where En is the error term, it is used to evaluate the probability that the bit position n would be flipped. Thus in the next step of WBF algorithm, the bit having highest error term En will be regarded the least reliable bit and hence flipped.ie, flip the particular bit in z which has the highest error term En. The forgoing steps are repeated until an all zero syndrome vector is obtained.

Improved Weighted Bit-Flipping Algorithm (IWBF):

The WBF algorithm considers the check node based information during the assesment of the error term En. By contrasting, the IWBF algorithm proposed by Zhang and Fossorier increased the performance of the WBF algorithm by considering both the check node and message node based information during the evaluation of En. As seen in the WBF, when the error term En is high, the corresponding bit is likely to be ab erroneous bit and hence it to be flipped. However when the soft value $|y_n|$ of a certain bit is high, the message node itself is expressing some confidence that the corresponding bit should not be flipped. Hence the above equation can be modified as:

$$En = \sum_{m\epsilon M(n)} (2sm-1) y_m^{min} - \alpha |y_n| \quad (13)$$

this equation considers the extra information provided by the message node itself, thus a message node having higher soft value has a lower chance of being flipped, regardless having a high error term En.

Gradient Descent Bit-Flipping Algorithm(GDBF):

The numerical problem for a differentiable function, the GDBF method is a natural choice. The partial derivative of f(x) with respect to the variable x(k) ϵ [1,n] can be derived from the definition of f(x):

$$\partial/\partial x_k \, f(x) = y_k + \sum_{i\epsilon M(k)} \prod_{j\epsilon N(i)\backslash k} x_j \quad (14)$$

Consider the product x_k and partial derivative of x_k in x given as:

$$x_k \partial/\partial x_k \, f(x) = x_k y_k + \sum_{i\epsilon M(k)} \prod_{j\epsilon N(i)} x_j \quad (15)$$

one better way to finding the position of flipping is to choose the position at which the absolute value of partial derivative is large.

Reliability-Ratio Based Weighted Bit-Flipping Algorithm(RRWBF):

The optimum value has to be found specifically for each particular column weight and its value should be optimized for each individual SNR are the main draw backs of I-WBF. In RRWBF introduce a new quantity termed as the reliability ratio (RR) defined as:

$$R_{mn} = \beta . | y_n | / |y_m^{max}| \quad (16)$$

Where $|y_m|$ is used to denote the highest soft value of all the message nodes participating in the mth check. The variable β is a normalization factor for ensuring that we have $\sum n{:}n\epsilon N(m)\ R_{mn} = 1$.
Hence, instead of calculating the error term En using y_m^{min}, propose the employment of the following formula:

$$En = \sum_{m\epsilon M(n)} (2sm-1)/R_{mn} \quad (17)$$

the rest of the RR-WBF algorithm is same as the standard WBF algorithm and the iteration will be terminated when a all zero syndrome vector obtain.

Self Reliability Based Weighted Bit Flipping Algorithm(SRWBF):

According to the previous methods there are two kind of information need to be considered in evaluating the error term for each bit: the information from check node and the intrinsic information. It is noticed that the 2sm-1 term may bring enough information from check nodes. Hence, the self reliability $|y_n|$ should be considered more in contrast to the reliability of the neighbor variable nodes participating in same check nodes. In consideration of this, a new self reliability ratio based weighted bit flipping decoding algorithm is introduced. The new error term used is:

$$En = \sum_{m\epsilon M(n)} (2sm-1)/|y_n| \quad (18)$$

The ignorance of the reliability of neighbor variable nodes can largely reduce the decoding complexity.

Check Reliability Based Bit-Flipping (CRBF) Algorithms:

Two CRBF algorithms are proposed: the soft check reliability based bit flipping (soft-CRBF) algorithm, which proposes the received channel values when decoding, and its hard decision counterpart which sends the hard decision counter which sends the hard decision demodulated bit streams to the decoder. The soft CRBF outperforms the WBF decoding algorithm and its variants and is comparable to SPA for some LDPC codes.

Two novel check reliability based soft decision bit flipping decoding algorithms are used to improve the performance of the WBF algorithm and its variants for decoding LDPC codes. At each iterations, the cost/ reliability for each bit is computed. The bit with reliable is flipped. The check reliability is also defined for each check node and is used to update the related bit node reliabilities. The sum of bit cost/reliability is to be a relaxed version of the ML decoding metric.

B. SOFT DECISION DECODING

Soft-decision decoding gives better performance in decoding procedure of LDPC codes which is based on the idea of belief propagation. In soft scheme, the messages are the conditional probability that in the given received bit is a 1 or a 0. The sum-product algorithm is a soft decision message-passing algorithm. Priori probabilities for the received bits is the input probabilities as here they were known in advance before running the LDPC decoder. The bit probabilities returned by the decoder are called the a posterior probabilities[4].

In the paper [3], the sum-product algorithm is a soft decision message-passing algorithm which is similar to the bit-flipping algorithm described in the previous section, but the major difference is that the messages representing each decision with probabilities in SPA. Whereas bit-flipping decoding on the received bits as input, accepts an initial hard decision and the sum-product algorithm is a soft decision algorithm which accepts the probability of each received bit as input. For example here initially take a guess that suppose a binary variable x, then it is easy to find P(x = 1) given P(x = 0), since P(x = 1) = 1-P(x = 0) and so here it is needed to store one probability value for x. Log likelihood ratios are introduced here to do so. They are used to represent the metrics for a binary variable by a single value as per following:

$$L(x)= \text{Log}\ (P(x=0)/P(x=1)) \qquad (19)$$

The aim of sum-product decoding algorithm here is first to compute the maximum a posteriori probability (MAP) for each codeword bit. Now here it is the probability that the i-th codeword bit is a 1 conditional on the event N and that all parity-check constraints are satisfied[10]. The sum-product algorithm iteratively computes an approximation of the MAP value for each code bit. The a posteriori probabilities returned by the sum-product decoder are only exact MAP probabilities if the Tanner graph is cycle free[9].The extra information about bit i received from the parity-checks is called as extrinsic information for bit i. Until the original a priori probability is returned back to bit i via a cycle in the Tanner graph, the extrinsic information obtained from a parity check constraint in the first iteration is independent of the a priori probability information for that bit and information provided to bit i in subsequent iterations which remains independent of the original a priori probability for bit i. In sum-product decoding the extrinsic message from check node j to variable node i, $E_{j,i}$, is the LLR of the probability that bit i causes paritycheck j to be satisfied.

The probability that the parity-check equation is satisfied if bit i is a 1 is,

$$P_{j,i}^{ext} =1/2-1/2\ \pi_{i' \epsilon B j,i' \neq i} (1-2P_{i'}^{int}) \qquad (20)$$

Where $P_{j,i}^{ext}$ is the current estimate, available to check j, of theprobability that bit i‘ is a one. If bit i is a zero, The probability that the parity-check equation is satisfied is thus $(1-2P_{i}^{ext})$ .Here it is expressed as a log-likelihood ratio,

$$E_{j.i} = \text{LLR}\ P_{j,i}^{ext}= \text{Log}\ [(1-2P_{i}^{ext})/ P_{j,i}^{ext}] \qquad (21)$$

We get,

$$E_{j.i}= \text{Log}\ [1+ \pi_{i' \epsilon Bj,i' \neq i} \tan h\ (M_{j,i}'/2)] / [1- \pi_{i' \epsilon Bj,i' \neq i} \tan h\ (M_{j,i}'/2)] \qquad (22)$$

Where,

$$M_{j,i}'= \text{LLR}\ (P_{j,i'}^{int}) = \log\ [(1- P_{j,i'}^{int}) / P_{j,i'}^{int}] \qquad (23)$$

Here Each bit has access to the input a priori LLR, ri, and the LLRs from every connected check node. The total LLR of the i-th bit is the sum of these LLRs:

$$L_i = \text{LLR}\ (P_i^{int}) = r_i + \sum_{j \epsilon Ai} E_{j.i} \qquad (24)$$

The messages sent from the bit nodes to the check nodes, $M_{j,i}$, are not the full LLR value for each bit here. The equation Hx[mod 2] = 0 is satisfied (where x[mod 2] is received codeword) or maximum number of iterations set.

The paper [13] ‘Channel Coding using Low Density Parity Check Codes in AWGN’ compared performance between a hard decision decoding algorithm (bit flipping) and a soft decision decoding algorithm (belief propagation). The analysis is based on the Bit Error Rate of decoding outputs. The result shows that the Soft decision decoding gives better performance than the hard decision decoding. LDPC code with soft decision decoding enhances the system performance and makes the long distance communication fast and error free.

APPLICATION OF LDPC CODES

It is error correcting codes in DVB-s2 standard for satellite communication for digital television. It is also used in Ethernet 10 base T. It is also a part of Wi-Fi 802.11 standard. The optional parts of it are 802.11 ac and 802.11n. It is also used in OFDM networks where data transmission to be without error. It is even with low bit rate also.

CONCLUSION & FUTURE SCOPE

Low-density parity-check (LDPC) code, a very promising near-optimal error correction code (ECC), is being widely well thought-out in next generation industry standards. LDPC code implementations are widely used in DVB-S2, T2 or Wi-MAX standards. Unlike many other classes of codes, LDPC codes are already equipped with very fast (probabilistic) encoding and decoding algorithms. These algorithms can recover the original codeword in the face of large amounts of noise. The iterative decoding approach is already used in turbo codes but the structure of LDPC codes give even better results. In many cases they allow a higher code rate and also a lower error floor rate. Furthermore they make it possible to implement parallelizable decoders..

REFERENCES:

[1] R. G. Gallager, 'Low-Density Parity-Check Codes'. Cambridge, MA: M.I.T. Press, 1963.

[2], Bernhard M.J. Leiner, LDPC Codes – a brief Tutorial Stud.ID.: 53418L bleiner@gmail.com,April 8, 2005.

[3] Namrata P. Bhavsar, Brijesh Vala,' Design of Hard and Soft Decision Decoding Algorithms of LDPC'. International Journal of Computer Applications (0975 – 8887)

[4] Ashish Patil, Sushil Sonavane, Prof. D. P. Rathod, "Iterative Decoding schemes of LDPC codes", International Journal of Engineering Research and Applications (IJERA), March -April 2013.

[5] Sarah J. Johnson, 'Introducing Low-Density Parity-Check Codes', School of Electrical Engineering and Computer Science, The University of Newcastle, Australia.

[6] William E Ryan, ' An introduction to LDPC codes', department of electrical and computer engineering. The University of Arizona,Australia,2003.

[7] M. Jadhav, ankit pancholi, dr. A. M. Sapkal,'Analysis and implementation of soft decision decoding algorithm of ldpc', pune university phase-i, d,-402, g.v 7, ambegaon, pune (india),[IJETT],2013.

[8] B. Vasic , Ivan B. Djordjevic, Low density parity check code and iterative decoding for long haul optical communication system,Journal of light wave technology , vol.21.no.2. February 2003

[9] Lakshmi.R, Tilty Tony, Abin Johns Raju, An Analytical Approach to The Performance of Low Density Parity Check Codes, International Conference on Advanced Computing and Communication Systems (ICACCS -2013), Dec. 19 21, 2013

[10] Vikram Arkalgud Chandrasetty, Syed Mahfuzul Aziz, FPGA Implementation of a LDPC Decoder using a Reduced Complexity Message Passing Algorithm , Journal of Networks, Vol. 6, no. 1, January 2011

[11] Thomas J. Richardson and Rüdiger L. Urbanke 'Efficient Encoding of Low-Density Parity-Check Codes' IEEE TRANSACTIONS ON INFORMATION THEORY, VOL. 47, NO. 2, FEBRUARY 2001.

[12] CHINNA BABU.J, V.USHA SREE,andS.PRATHYUSHA, **'**Hard decision and soft decision algorithms of LDPC and comparison of LDPC with Turbo codes, RS codes and BCH codes'. Proceedings of 09th IRF International Conference, 27th July-2014, Bengaluru, India, ISBN: 978-93-84209-40-7.

[13] Rinu Jose and Ameenudeen P.E, 'Channel Coding using Low Density Parity Check Codes in AWGN'. International Conference on Emerging Trends in Technology and Applied Sciences (ICETTAS 2015)