DESIGN OF LDPC ARCHITECTURE USING VERILOG CODING

[1]Manjunatha P N , [2]Prof .T.S Bharath kumar, [3]Dr. M Z Kurian

Abstract— **Low Density Parity Check codes are FEC codes and hence data rate is more. They are linear error correcting codes for transmitting a message over a noisy transmission channel. LDPC codes are finding increasing use in applications requiring reliable and highly efficient information transfer over noisy channels. These codes are capable of performing close to the shannon's capacity[2]. The main advantage of the parity check matrix is the decoder can correct all single-bit errors. In this Paper LDPC encoder and decoder architecture for coding 8-bit message vector will be analyzed and designed using verilog code.**

Index Terms— **Bit Flipping , G-matrix, H-matrix, LDPC.**

I. INTRODUCTION

LDPC codes are invented by Robert Gallger in 1960's. This codes are neglected for more than thirty years because of hardware complexity at that time. This codes are reinvented by Mackay & Neal in 1990's by constructing the codes using sparse parity check matrix. **L**ow-Density parity-check (LDPC) codes have recently attracted tremendous research interest because of their excellent error-correcting performance and highly parallel decoding scheme. LDPC codes have been selected by the digital video broadcasting (DVB) standard and are being seriously considered in various real-life applications such as magnetic storage, 10 Gb Ethernet, and high-throughput wireless local area network . They are class of linear block codes, as their name suggests low density means number of 1's in the parity matrix is very small compare to 0's. Condition for low density is W_c <<m and W_r <<n , where W_c represents the column weight and W_r represents the row weight. The sparseness of parity matrix guarantees that the complexity of decoding algorithm increases only with increasing code length. In this coding technique we are going to use two matrixes one is generator matrix G at encoder and parity check matrix H at decoder. Rows of the parity matrix represents check nodes and columns represents the variable nodes of the tanner graph. Bits in the codeword are based on the variable nodes and bits in the message vector are based on the check nodes. They are two types of parity check matrix, one is Regular in which column weights and row weights are same for all columns and rows respectively and other one is Irregular in which column and row weights are different for each columns and rows respectively.

Manuscript received April, 2014.
[1]Manjunatha P N ,4th sem, M.Tech (VLSI and Embedded systems), SSIT,
[2].T.S Bharath kumar,Professor. Dept of ECE, SSIT, Tumkur
[3]Dr. M Z Kurian ,Dean & HOD, Dept of ECE, SSIT, Tumkur, Karnataka

II. SYSTEM DESIGN

Here we have divided our entire LDPC system in to three major blocks mainly,

1) Encoder block
2) Noise insertion block (AWGN - channel)
3) Decoder block

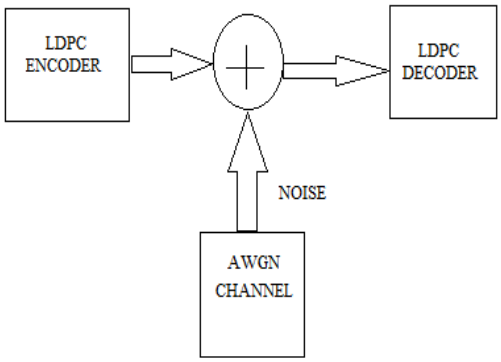

Fig 1: System Architecture

LDPC Algorithm:
A codeword **c** is generated as

$$\mathbf{C = KG} \quad (1)$$

where **K** is the message vector and **G** is the generator matrix. A valid codeword can be verified using

$$\mathbf{HC^T} = 0 \quad (2)$$

Where **H** is the parity check matrix. If the result in (2) is nonzero, the codeword **C** is invalid and an error correction procedure should be used in this case. The Bit flipping method uses a vector, called syndrome , which is computed as

$$\mathbf{S = HY^T}, \quad (3)$$

Where **Y** is the invalid codeword. The syndrome indicates which row in the **H** is not zeroed by vector **Y** and some bits have to be repaired in the decoder .If the parity check matrix has low size, we can find an error floor of the LDPC code, where one erroneous bit is repaired and BER is close to zero or is zero.

III. ENCODER DESIGN

Encoder uses generator matrix to encode the information bits in to the code word. Both generator and parity check matrix are in inter related, parity check matrix is given by

$$\mathbf{H=[A \mid I_{n-k}]}$$

and generator matrix is given by

$$\mathbf{G=[I_k \mid A^T]}$$

Initially parity check matrix is generated ,using that matrix generator matrix is created by Gaussian elimination method. There are two types of parity matrices in LDPC coding one is Regular matrix and another one is irregular matrix. Regular matrix is one in which column W_c is same for all columns and row weight is given by $W_r = W_c(n/m)$. In this paper we are using regular matrix of 8x16.

1	0	0	1	1	0	0	0	0	0	0	0	0	0	1	0
1	0	0	0	0	0	0	0	0	1	0	0	1	1	0	0
0	0	1	0	1	0	0	0	1	0	0	1	0	0	0	0
0	0	0	1	0	0	0	1	0	0	1	1	0	0	0	0
0	0	1	0	0	1	0	0	0	0	0	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	1	0	0	0	0	0
0	1	0	0	0	0	1	0	0	0	0	0	0	0	1	1
0	1	0	0	0	1	0	1	0	0	0	0	1	0	0	0

Fig 2:Regular parity matrix

To transfer the above parity check matrix to standard form i.e $H=[A \mid I_{n-k}]$ Gaussian elimination method is applied to the above matrix. The matrix H is put into this form by applying elementary row operations which are interchanging two rows or adding one row to another modulo 2. The resulting parity matrix in its standard form is as shown in the fig 3,

$H =$

0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
0	0	1	1	1	0	1	1	0	1	0	0	0	0	0	0
0	0	1	1	1	0	0	1	0	0	1	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0	1	0	0	0	0
0	1	0	0	0	0	1	0	1	0	0	0	0	1	0	0
1	1	1	1	1	1	1	0	0	0	0	0	0	1	0	0
1	0	0	1	1	0	0	0	0	0	0	0	0	0	1	0
1	1	0	1	1	0	1	0	0	0	0	0	0	0	0	1

Fig 3: standard parity matrix

Obtained parity matrix is translated to standard form generator matrix i.e $G=[I_k \mid A^T]$ as shown in fig 4,

$G =$

1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
0	1	0	0	0	0	0	0	0	0	0	0	1	1	0	1
0	0	1	0	0	0	0	0	0	1	1	1	0	1	0	0
0	0	0	1	0	0	0	0	0	1	1	0	0	1	1	1
0	0	0	0	1	0	0	0	0	1	1	1	0	1	1	1
0	0	0	0	0	1	0	0	0	0	0	0	1	1	0	0
0	0	0	0	0	0	1	0	0	1	0	0	0	1	0	1
0	0	0	0	0	0	0	1	0	1	1	0	1	0	0	0

Fig 4: Generator matrix

Now the information message bits are encoded by multiplying it with above generator matrix i.e C=[U][G] to obtain the codeword. The below fig 5 shows the encoder block diagram,

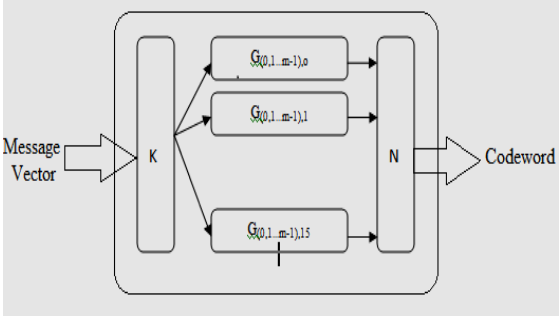

Fig 5:Encoder Block Diagram

Each structure labeled $G_{\{0,1,,,m-1\},i}$ are XOR structures performs modulo-2 operations on the incoming message bits and the resultant code words will be of N bits. Let us consider an 10 bit information message U =[0 0 0 0 1 1 0 0],

$[1\,0\,0\,1\,0\,1\,0\,0]$

1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
0	1	0	0	0	0	0	0	0	0	0	0	1	1	0	1
0	0	1	0	0	0	0	0	0	1	1	1	0	1	0	0
0	0	0	1	0	0	0	0	0	1	1	0	0	1	1	1
0	0	0	0	1	0	0	0	0	1	1	1	0	1	1	1
0	0	0	0	0	1	0	0	0	0	0	0	1	1	0	0
0	0	0	0	0	0	1	0	0	1	0	0	0	1	0	1
0	0	0	0	0	0	0	1	0	1	1	0	1	0	0	0

$C = [1\,0\,0\,1\,0\,1\,0\,0\,0\,1\,1\,0\,1\,1\,0\,0]$

Fig 6: Encoding

By multiplying message vector with generator matrix we obtain the codeword of 10 bits C =[1 0 0 1 0 1 1 1 0 1] . Coding for this encoder part is done on verilog and encoding is tested for various information bits satisfactorily.

IV. CHANNEL DESIGN

When encoded of signals are transmitted through channel , coded signal may get corrupted by noise in the channel and other interferers. To model that effect here we are adding AWGN noise to the coded signal.

Additive white Gaussian noise (AWGN) is a basic noise model used in Information theory to mimic the effect of many random processes that occur in nature. The modifiers denote specific characteristics:

- 'Additive' because it is added to any noise that might be intrinsic to the information system.
- 'White' refers to idea that it has uniform power across the frequency band for the information system. It is an analogy to the colour white which has uniform emissions at all frequencies in the visible spectrum.
- 'Gaussian' because it has a normal distribution in the time domain with an average time domain value of zero.

Coding for this channel is done in verilog using Xilinx and its simulation results are shown below.

RTL schematic of Channel:

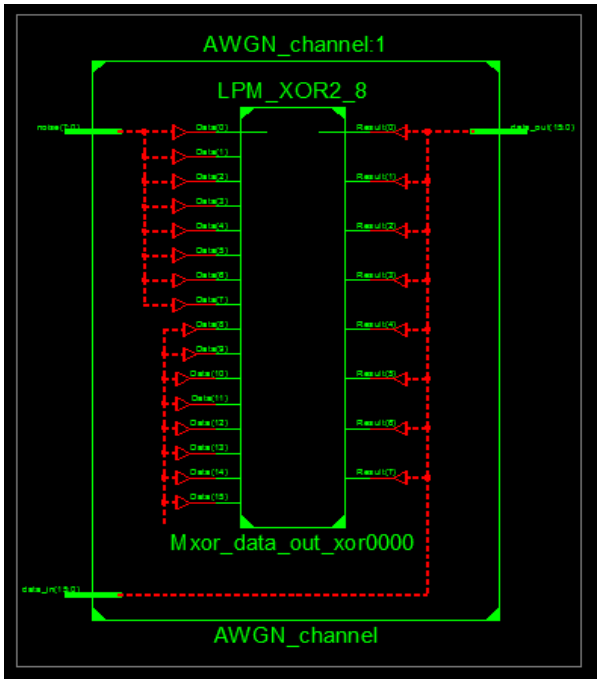

Fig 7 :RTL schematic

V. DECODER DESIGN

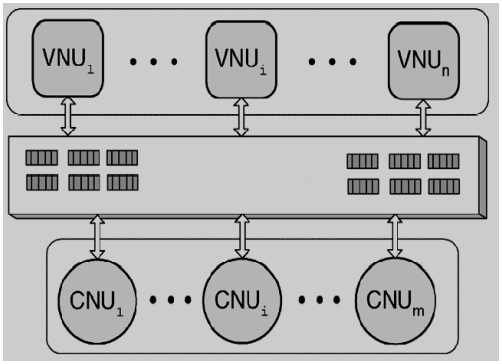

Fig 8: Decoder Block Diagram

Here in this paper we are using Bit Flipping decoding algorithm. This decoding algorithm is a hard decision message passing algorithm. A hard decision is made on each received bit and than those bits are transferred to tanner graph structure. In this algorithm message passed along the edges of tanner graph are binary bits. Initially a variable node sends a message to check nodes declaring if it is a 1 or 0. Than each check nodes calculates message for each variable node that what bit it should receive based on the information available to check node form other connected variable nodes i.e check node performs modulo-2 sum to verify the parity check equations. If the sum is zero than equation is satisfied otherwise bit is flipped and sent back to variable node. Now variable nodes have several bits one is initially received bit and other are various bits received from connected check nodes. Than variable nodes performs the majority check if result of the majority checks are same as the initial received bit than bit remains same else bit is flipped. This above process is continued until the all the parity check equations are satisfied and all errors are detected.

This bit flipping decoding algorithm immediately terminates as the all the parity equations are satisfied and valid codeword is detected, hence it has two advantages

i. Additional unnecessary iterations are eliminated after codeword detected.

ii. Any failure to converge to a codeword is always detected.

This algorithm is based on the principal that a codeword involved in a large number of incorrect check equations is likely to be incorrect itself . The sparseness of parity check matrix helps in spread out of the bits in checks so that the parity equations are unlikely to contain the same se of codeword bits.

This paper presents decoder design for 8-bit message vector. Below figures shows the parity check matrix,

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

Fig : parity matrix

In parity matrix number of rows represent the number of bits in the message vector and here we are considering 8 bit message vector therefore 8 rows. Number of columns in the matrix represent the number of bits in the resultant codeword. In this parity matrix each rows represents the parity equations. Tanner graph consist of two types of nodes, 1)Check nodes and 2)Variable nodes. Number check nodes equal to number of rows in parity matrix and number of variable nodes equal to number of columns in parity matrix. These nodes are connected by the edges. In the above figure we observe that each check node is connected to four variable nodes.

Let us consider the received codeword is 00001100. At the decoder initially received codeword bits are assigned to the variable nodes. Than each variable nodes sends the message to the each connected check nodes. Than each check nodes calculates the correct message to each connected by performing modulo-2 sum operation and than sends the resultant bits to each variable nodes. Than each variable nodes performs the majority check and corrects the flipped bit. Coding for this encoding and decoding is done in verilog using Xilinx and simulation results are reported below.

Simulation Results:

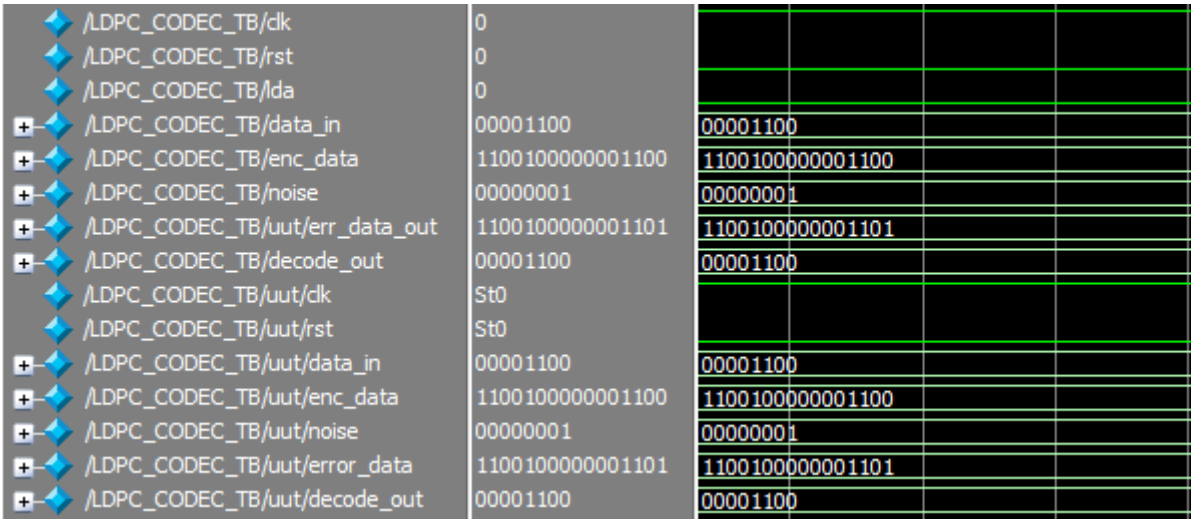

Fig :Simulation Result

Above figure shows the simulation results obtained from Modelsim 6.3 software. Here initially reset is made high that output goes low and as reset goes low input is encoded to give codeword . Here input is 00001100 and it's encoded codeword is 1100100000001100. Than to the encoded 16 bit codeword noise of 8 bit 00000001 is added to model the effect of noisy channel. Due the added noise some bits in the codeword gets flipped and the resulting corrupted codeword is 1100100000001101. Than by bit flipping decoding error bit is flipped and the original message vector is decoded.

VI. CONCLUSION

LDPC coding is a good error correcting coding technique which allows forward error correction and hence data rate of transmission is high. In this paper Design of Encoder, channel and Decoder is for a regular LDPC codes is presented. Coding is done using 8x16 parity check matrix with row weight of 4 and column weight of 2. Coding is done on Xilinx and simulation results are obtained from Modelsim. Coding satisfactorily verified for various 8-bit message vectors.

REFERENCES

[1] R. G. Gallager, "Low density parity check codes," IRE Trans. Inform. Theory, vol. IT-8, no.1, pp. 21–28, Jan. 1962

[2] D. J. C. MacKay, R. M. Neal, "Near Shannon limit performance of low density parity check codes," Electronics letters, vol. 32, no.18, pp.1645– 1646.

[3] R. M. Tanner, "A Recursive Approach to Low Complexity Codes," IEEE Trans. Inform. Theory, vol. IT-27, no.5, pp. 533–547, Sep. 1981

[4] S. J. Johnson, "Introducing Low-Density Parity-Check Codes," unpublished.

[5] T. Richardson, "Error floors of LDPC codes," in Proc. Annual Allerton Conference on Communication, Control, and Computing, Monticello, IL, pp. 1426-1435, Oct. 2003.

[6] T. Tian, C. Jones, J. Villasenor, and R. D. Wesel, "Construction of irregular ldpc codes with low error floors," in Proceedings IEEE International Conference on Com-munications, 2003.

[7] L. Wei, "Several properties of short LDPC codes," IEEE Trans. Commun., vol. 52, pp. 721-727, May 2004.

[8] H. Zhong and T. Zhang, "Block-LDPC: A practical LDPC coding system design approach,"IEEE Trans. Circuits Syst. I, vol. 52, no. 4, pp. 766 – 775, Apr. 2005.

[9] Xiao-Yu Hu; Eleftheriou, E.; Arnold, D.M, "Regular and irregular progressive edge-growth tanner graphs", IEEE Transactions on Information Theory, 2005, 51, (1), pp. 386 – 398

[10] T. Etzion, A. Trachtenberg, and A. Vardy, "Which codes have cycle-free Tanner graphs?,"IEEE Trans. Inf. Theory, vol. 45, no. 6, pp. 2173–2180, Sep. 1999

[10] Simulation and Implementation of LDPC Code in FPGA
Pavel Straus, Zdenek Kolka Department of Radio Electronics
Brno University of Technology Brno, Czech Republic
xstrau00@stud.feec.vutbr.cz,kolka@feec.vutbr.cz

