International Journal of Security and Its Applications
Vol.9, No.10 (2015), pp.21-30
http://dx.doi.org/10.14257/ijsia.2015.9.10.02

LDPC Encoder and Decoder Architecture for Coding 3-bit Message Vector

Gade Sribala[1], D.V.N. Sukanya[1], K.Gouthami[1] and Tai-hoon Kim[2]

[1]Department of Electronics and Communication Engineering,
Vignan's Nirula Institute of Technology and Science for Women,
Palakaluru, Guntur, AP, India
[2]Department of Convergence Security, Sungshin Women's University,
249-1, Dongseon-dong 3-ga, Seoul, 136-742, Korea
[1]g.sribala91@gmail.com, [1]sukanyadvn@gmail.com, [1]gsparuchuri@gmail.com,
[2]taihoonn@daum.net
(Corresponding Author)

Abstract

Low Density Parity Check codes are FEC codes and consequently 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 near to Shannon limit performance, Low Decoding Complexity. 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 3-bit message vector will be analyzed and also designed using VHDL.

Keywords: *LDPC, Parity matrix, Generator matrix, Shannon's Coding Theorem*

1. Introduction

LDPC codes are invented by Robert Gallger in 1960's. These codes are neglected for more than thirty years because of hardware complexity at that time. These codes are reinvented by Mackay & Neal in 1990's by constructing the codes using sparse parity check matrix.

Any linear block code can be defined by its parity-check matrix. If this matrix is sparse, i.e. it contains only a small number of 1s per row or column, and then the code is called a low-density parity-check code. Basically there are two different possibilities to represent LDPC codes: 1. Matrix Representation and 2. Graphical Representation.

Low-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.

Shannon's Coding Theorem, In 1948 Claude Shannon, generally regarded as the father of the Information Age, published the paper: "A Mathematical Theory of Communications" which laid the foundations of Information Theory. Every communication channel is characterized by a single number C, called the channel capacity.

ISSN: 1738-9976 IJSIA

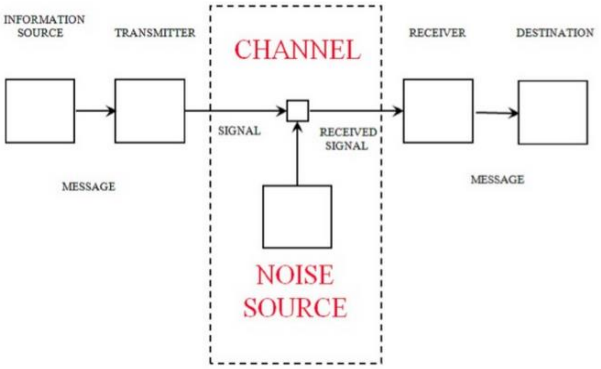

Figure 1. Noise Communication System

It is possible to transmit information over this channel reliably (with probability of error → 0) if and only if:

$$R \overset{def}{=} \frac{\#\,\text{information bits}}{\text{channel use}} < C$$

Shannon introduced the concept of codes as ensembles of vectors that are to be transmitted. To achieve reliable communication, it is thus imperative to send input elements that are correlated. This leads to the concept of a code, defined as a (finite) set of vectors over the input alphabet. The code has a rate of k/n bits per channel use, or k/n bpc (k=log2M).

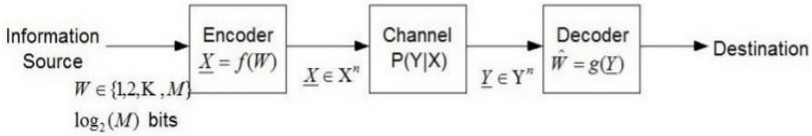

Figure 2. Message Passing between Sources to Destination

The code rate is given by:

$$R = \frac{\log_2(M)}{n} \quad \frac{\text{bits}}{\text{channel use}}$$

Reliable transmission is possible if R<C.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 Wc<<m and Wr<<n , where Wc represents the column weight and Wr 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 represent the check nodes and columns represent 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.

2. 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

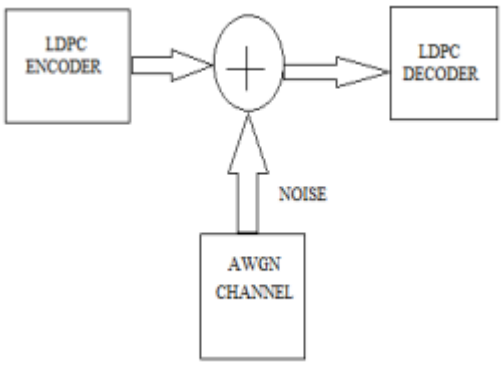

Figure 3. System Architecture

2.1. LDPC Algorithm

A codeword c is generated as C= KP (1) where K is the message vector and G is the generator matrix. A valid codeword can be verified using CHT = 0 (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 S = YHT, (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.

2.2. Design of LDPC Architecture using VHDL Coding

Where "Y" is the invalid Code Word. 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.

3. Encoder Design

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

$$H = [\ PT \mid I\ ] \text{ (or) } H=[\ In\text{-}P \mid I\ ]$$

and the generator matrix is given by

$$G = [\ I \mid PT\ ]$$

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 Wc is same for all columns and row weight is given by

$$Wr = Wc(n/m)$$

In this paper we are using regular matrix of 3X7 (or) (n,k)=(7,3) i.e., where n represents total bits and k represents message bits, n-k=7-3=4 which represents check bits or parity bits.

$$\begin{pmatrix} 1011001 \\ 1110100 \\ 1100010 \\ 0110001 \end{pmatrix}$$

Figure 4. Regular Parity Matrix

To transfer the above parity check matrix to standard form i.e H=[PT | I] 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 H is as shown in the figure 5,

$$\begin{pmatrix} 1011000 \\ 1110100 \\ 1100010 \\ 0110001 \end{pmatrix}$$

Figure 5. Standard Parity Matrix

If G is the generated matrix for (n, k) code then H is the generator matrix for (n, n-k) code. Therefore obtained parity matrix is translated to standard form generator matrix i.e., G =[I | P] as shown in fig 6,

$$\begin{pmatrix} 0001110 \\ 0100111 \\ 0011101 \end{pmatrix}$$

Figure 6. Generator Matrix

Now the information message bits are encoded by multiplying it with above generator matrix i.e., C = [M][P] to obtain the codeword. The below figure 7 shows the encoder block diagram,

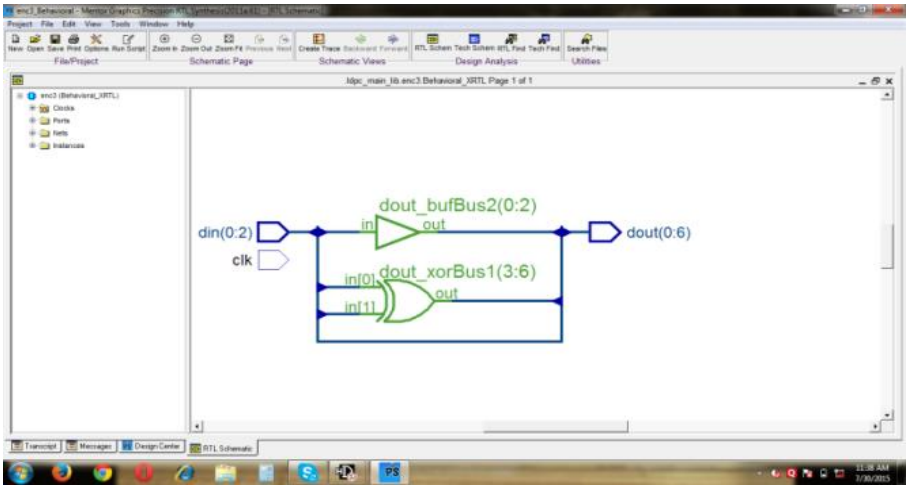

Figure 7. Encoder Block Diagram

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 n-bit information message U = [101], and C is given by

$$[101] \begin{pmatrix} 1110 \\ 0111 \\ 1101 \end{pmatrix}$$

Figure 8. Encoding

By multiplying message vector with generator matrix we obtain the codeword with parity (or) check bits C = [1 0 1 0 0 1 1]. Coding for this encoder part is done on VHDL and encoding is tested for various information bits satisfactorily.

4. 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:

The term ‘Additive’ because it is added to any noise that might be intrinsic to the information system.

Similarly for the term 'White' refers to idea that it has uniform power across the frequency band for the information system. It is an analogy to the color white which has uniform emissions at all frequencies in the visible spectrum.

Finally the term '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 VHDL using HDL DESIGNER SERIES TOOL and its simulation results are shown below.

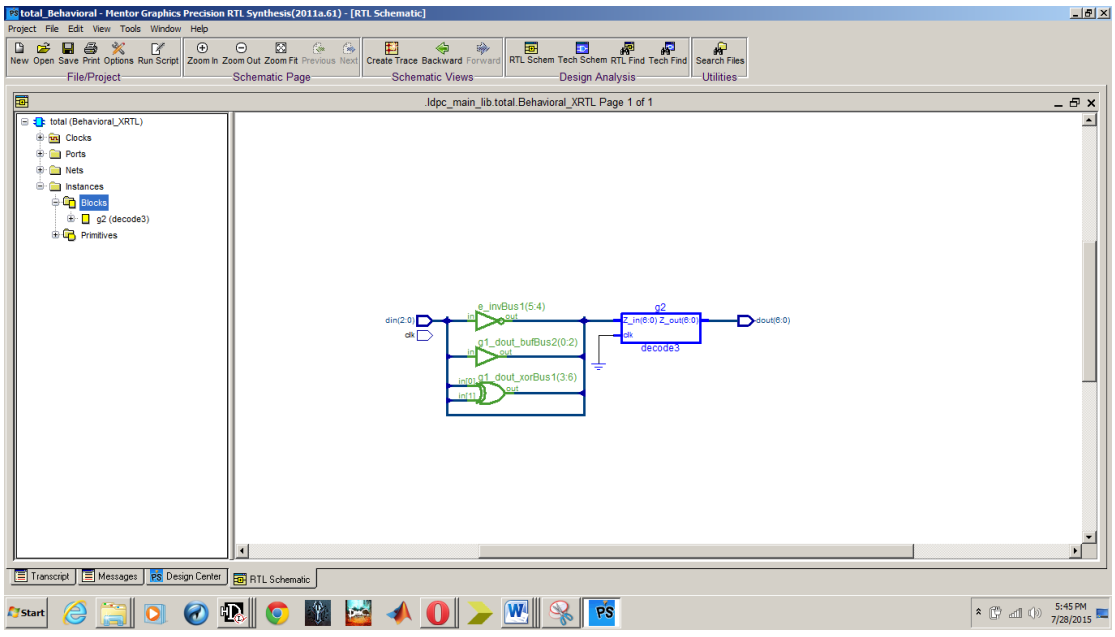

Figure 9. RTL Schematic Block Diagram

5. Decoder Design

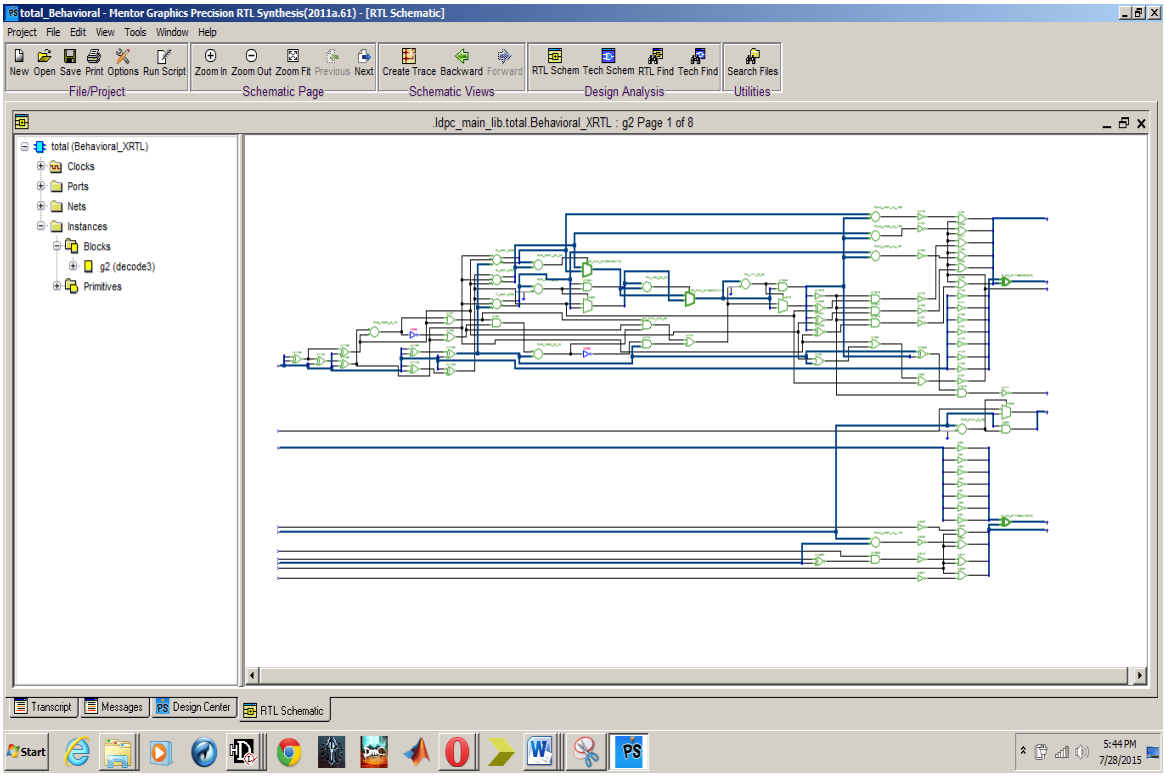

Figure 10. Decoder Block Diagram

Here in this paper we are using SYNDROME DECODING ALGORITHM CHT=0. This decoding algorithm is a hard decision message passing algorithm. 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 perform 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, i.e.,

If $cH^T = 0$ or maximum number of iterations reached, then stop otherwise continue iterations,

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 3-bit message vector and 7-bit code word with 3-message bits and 4-checkbits. Below figures shows the parity check matrix,

$$H = \begin{pmatrix} 1011001 \\ 1110100 \\ 1100010 \\ 0110001 \end{pmatrix}$$

Figure 11. Parity Matrix

In parity matrix number of rows represents the EQUATIONS and COLUMNS represent the DIGITS in the CODEWORD bits.

Number of columns in the matrix represents 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 1010011. At the decoder initially received codeword bits are assigned to the variable nodes. Then each variable node sends the message to the each connected check nodes. Then each check nodes calculates the correct message to each connected by performing modulo-2 sum operation and then sends the resultant bits to each variable node. Then each variable node performs the majority check and corrects the flipped bit. Coding for this encoding and decoding is done in VHDL using HDL designer Series and simulation results are reported below.

5.1. Simulation Results

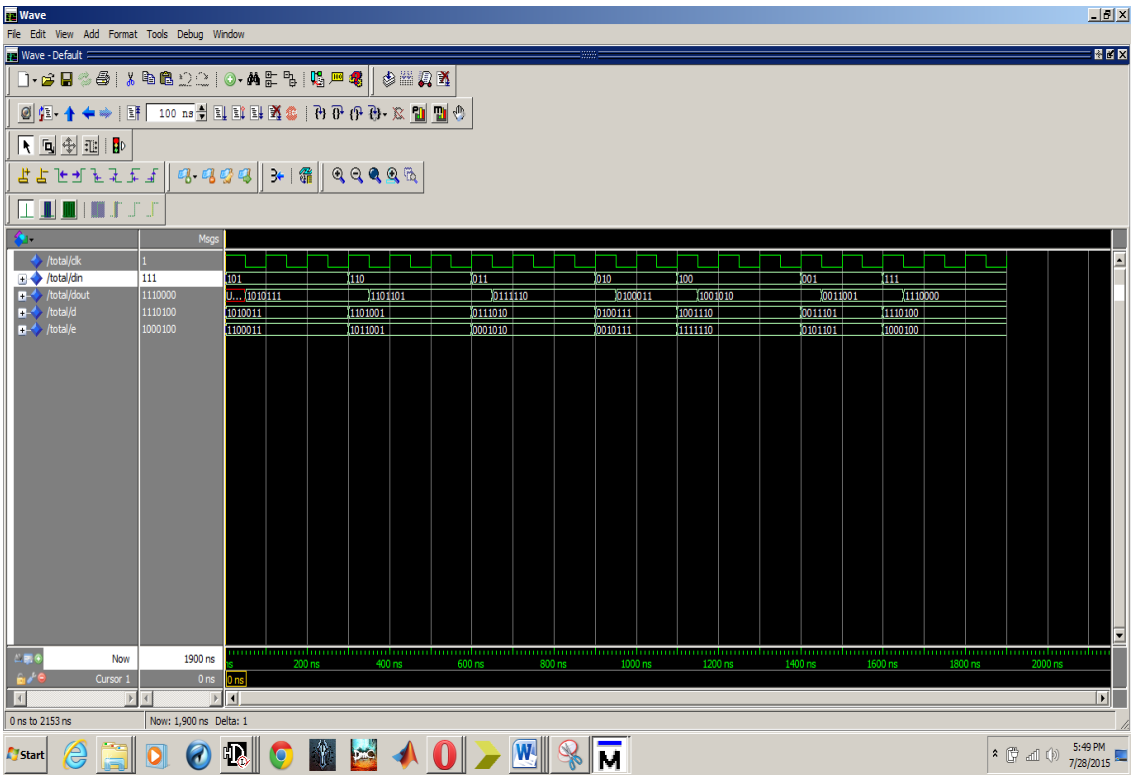

Figure 12. Simulation Result

Above figure shows the simulation results obtained from Modelsim 10.6 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 varied from 000 to 111 and encoded codeword for the input 110 is 0101110. Than to the encoded 7 bit codeword noise of 7 bit "0110000" 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 0011110. Than by bit flipping decoding error bit is flipped and the original message vector is decoded.

```
*************************************************************
Device Utilization for 6SLX16CSG324
*************************************************************
Resource                          Used    Avail   Utilization
-------------------------------------------------------------
IOS                               15      232       6.47%
Global Buffers                    0       16        0.00%
LUTs                              18      9112      0.20%
CLB Slices                        5       2278      0.22%
Dffs or Latches                   0       18224     0.00%
Block RAMs                        0       32        0.00%
DSP48A1s                          0       32        0.00%
-------------------------------------------------------------
```

Figure 13. Representing the Area Report for the LDPC Design

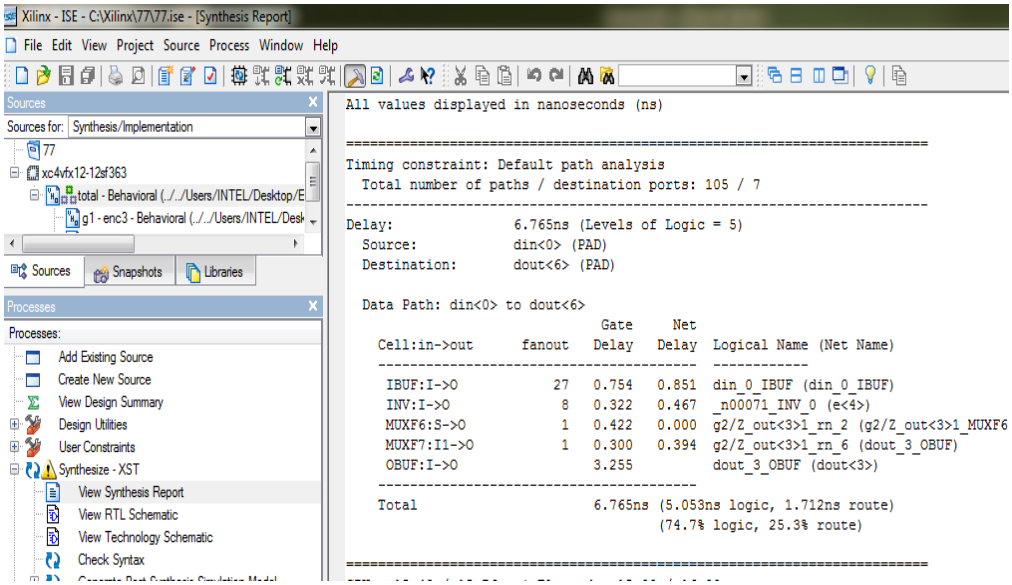

Figure 14. Representing the Estimated Delay Report

5.2. Applications of LDPC

LDPC codes have already been adopted in satellite-based digital video broadcasting and long-haul optical communication standards, are highly likely to be adopted in the IEEE wireless local area network standard, and are under consideration for the long-term evolution of third generation mobile telephony. LDPC is also used for 10GBase-T Ethernet, which sends data at 10 gigabits per second over twisted-pair cables. LDPC codes are also part of the Wi-Fi 802.11 standard as an optional part of 802.11n and 802.11ac, in the High Throughput (HT) PHY specification.

6. Conclusion

LDPC coding is a superior error correcting coding technique which allows further error modification and hence data rate of transmission is elevated. In this paper Design of Encoder, channel and Decoder is for a LDPC codes is presented. Coding is done using (7, 3) LDPC code. Coding is done on HDL Designer Series and simulation results are obtained from Modelsim. Coding satisfactorily verified for various 3-bit message vectors.

References

[1] R. G. Gallager, "Low-density parity-check codes", IRE Transactions on Information Theory, vol. 8, no. 1, (**1962**), pp. 21-28.
[2] MacKay, JC David and R. M. Neal, "Near Shannon limit performance of low density parity check codes", Electronics letters, vol. 32, no. 18, (**1996**), pp. 1645-1646.
[3] T. R. Michael, "A recursive approach to low complexity codes", IEEE Transactions on Information Theory, vol. 27, no. 5, (**1981**), pp. 533-547.
[4] S. J. Johnson, "Introducing low-density parity-check codes", University of Newcastle, Australia (**2006**).
[5] R. Tom, "Error floors of LDPC codes", Proceedings of the annual Allerton conference on communication control and computing, vol. 41, no. 3, pp. 1426-1435.
[6] T. Tao, C. Jones, J. D. Villasenor and R. D. Wesel, "Construction of irregular LDPC codes with low error floors", IEEE International Conference on Communications, ICC'03, vol. 5, (**2003**), pp. 3125-3129.
[7] L. Wei, "Several properties of short LDPC codes", IEEE Transactions on Communications, vol. 52, no. 5, (**2004**), pp. 721-727.
[8] H. Zhong and T. Zhang, "Block-LDPC: A practical LDPC coding system design approach", IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 52, no. 4, (**2005**), pp. 766-775.
[9] X.-Y. Hu, E. Eleftheriou and D. M. Arnold, "Regular and irregular progressive edge-growth tanner graphs", IEEE Transactions on Information Theory, vol. 51, no. 1, (**2005**), pp. 386-398.
[10] T. Etzion, A. Trachtenberg and A. Vardy, "Which codes have cycle-free Tanner graphs?", IEEE Transactions on Information Theory, vol. 45, no. 6, (**1999**), pp. 2173-2181.
[11] P. Straus and Z. Kolka, "Simulation and implementation of LDPC code in FPGA", 23rd International Conference on Radioelektronika (RADIOELEKTRONIKA), IEEE, (**2013**), pp. 346-349.