www.ijcrt.org © 2020 IJCRT | Volume 8, Issue 7 July 2020 | ISSN: 2320-2882

IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

IMPLEMENTATION OF DECODER FOR LDPC CODES USING OFFSET MIN SUM

[1]G M G Madhuri, [2]A V K Sravanthi, [3]A Ramya Sri, 4G Sri Mrunalini, [5]G Sai Kumar

[1]Assoc. Professor, [2]Student, [3]Student, [4]Student, [5]Student
[1]Department of Electronics and Communication,
[1]PSCMR College of Engineering and Technology, Vijayawada, India

Abstract: The world is now in the hands of technology. Now-a-days the usage of network has been increasing that leads to decrease in speed of network. In order to increase the speed of network there are some techniques. Gallager introduced decoding techniques to increase the throughput and latency. In this paper we are discussing one of the decoding methods that were used in 5G technology. In 5G technology LDPC and POLAR codes plays a crucial role to increase the throughput and latency.

I. INTRODUCTION

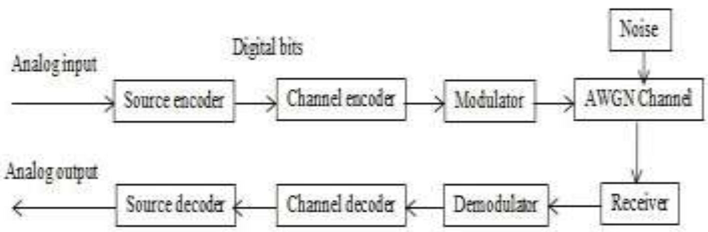
Gallager introduced Low Density Parity Check codes (LDPC) in the year 1963 [1]. The code was named Low Density Parity Check because the number of 1's in parity check matrix is less than number of 0's. The LDPC codes can be represented in two ways either in mathematical form such as matrix notation and in graphical form such as bipartite graph. In this paper the technique used for decoder is offset Min Sum algorithm. In the decoding process the final stage will be a decision making process to get the original bits. The final decision can be of two types [2], they are hard and soft decision.

Hard decision: In this process threshold value is considered to make the final decision.

Soft decision: In this the decision will be taken by considering the log likelihood ratio (ratio of probabilities) [3].
In this paper hard decision is considered to take the final decision.

II. BLOCK DIAGRAM

The input analog message is passed to the source encoder where it converts the analog signal into bit stream and then it passed the digital signal (bit stream) to the channel encoder. In the channel encoder the parity bits are added to the message bits to get a codeword [3]. The output of the channel encoder i.e., the codeword is passed to the modulator. The modulation is a BPSK modulation; it converts the bits to symbols because the bits cannot convert directly to the analog signal [7] [6]. If the digital bits are converted directly then error will be more. After converting the bits to symbol, the modulator converts the symbol to analog signal which then transmits over AWGN channel.

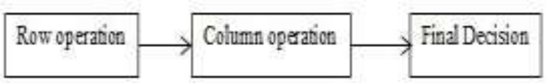


While transmitting the signal in the channel some noise is added to the signal. This noise can be removed easily. The receiver receives the signal from the channel with the noise. Then it transmits the signal to the demodulator where the demodulator converts the analog signal to the digital signal. Some operations are performed in the source decoder such as row operation, column operation and final decision. After this the symbol to bit transmission takes place during the final decision. These digital bits are transmitted over the source decoder to get the original analog signal.

III. OFFSET MIN SUM DECODER

The decoder consists of mainly three steps. They are

- Row operation
- Column operation
- Final decision



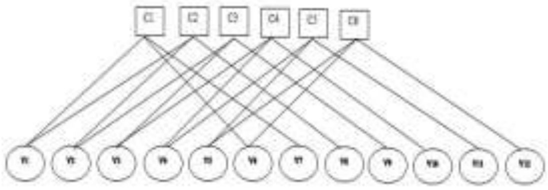
Before these operations we have initialisation step where the bits from the receiver are placed in the H- matrix in the place of 1's.

H- Matrix

This matrix is used to represent the LDPC codes. As mentioned in the introduction this matrix is the mathematical form. This matrix consists of few 1's compared to 0.

Let us consider an example to represent the matrix and its graphical notation [4].

$$H=\begin{array}{c} \begin{matrix} V1 & V2 & V3 & V4 & V5 & V6 & V7 & V8 & V9 & V10 & V11 & V12 \end{matrix} \\ \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \end{array} \begin{matrix} C1 \\ C2 \\ C3 \\ C4 \\ C5 \\ C6 \end{matrix}$$



The nodes c1, c2, c3… are known as check nodes, these represent the columns and the nodes v1, v2, v3… are known as variable nodes, these represent the rows of the parity check matrix (H- matrix) [5].

The row and column operations in the decoders will be explained in the later section.

IV. IMPLEMENTATION

This is the designing steps for Offset Min Sum decoder [9]

Step 1: Initialise the values received from the receiver in the parity check matrix.

Step 2: After initialising consider the two minimum values in each and every row. The values should be taken without considering the sign.

Step 3: Now check the parity by taking the multiplication of each and every sign of node in the row. If the parity is negative then there will be change in sign and if the parity is positive there will be no change.

Step 4: The first min value should be replaced with the second min value and other values are replaced with first minimum value. The process from **Step 2- Step 4** is **Row operation.**

Step 5: Repeat Step 2 – 4 for each and every row.

Step 6: All the values in the first column should be added. Then we get a sum value, add this sum value to the first element of the received value and consider this as new value.

$Sum_j = r_j +$ sum of all entries in column

r_j represents the received values from the channel after demodulation.

Step 7: Remove the old value (if we want to change the value first row first column then that value will be the old value) from the new value and place the value it belongs to (in the first row first column). Repeat step 6 and 7 for all the columns.

New entry = sum_j – old entry

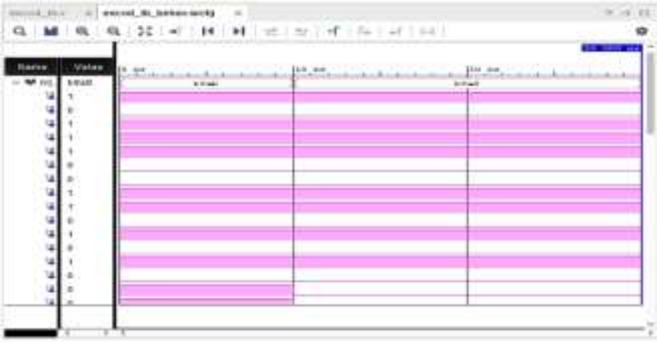
Step 8: Now the final decision to be taken by considering the threshold values [8]. If the threshold is zero, then if the sum value is greater than zero consider the value as 1 and if the value is less than zero consider it as 0.

These are the steps to be performed for Offset Min Sum algorithm for the LDPC Codes [9], [10].

Note: All the operations should not be performed in place of zeros.

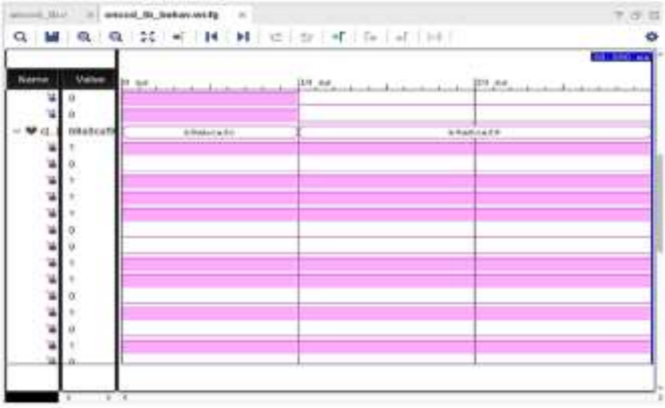
V. RESULTS AND CONCLUSION

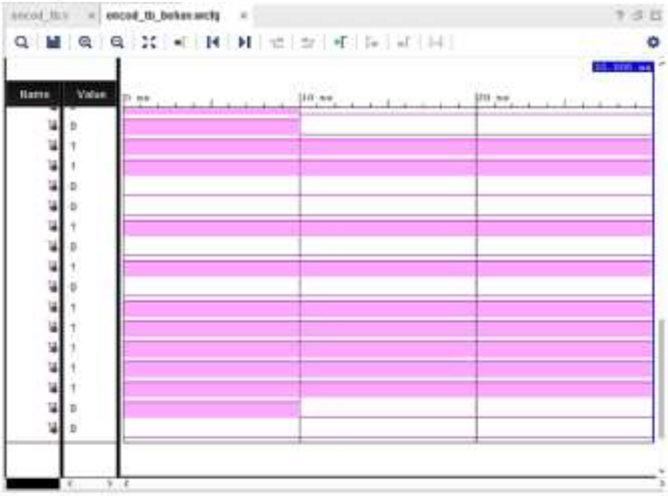
Encoder input



The graph represents the 16 bit input message bits

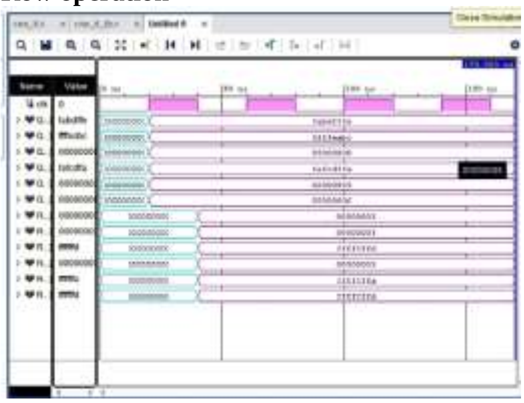
Encoder output





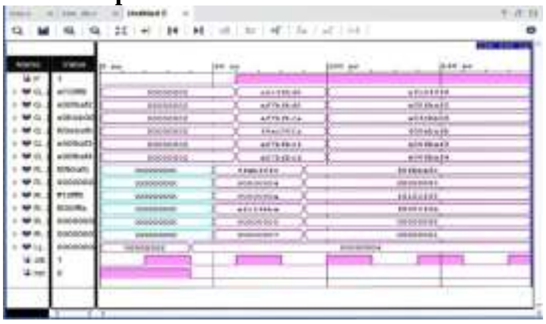
The above two graphs represent the output of the encoder (codeword). The first 16 bits represents the message bits and the next 16 bits represents the parity bits which are xor operated.

Row operation



Two minimum values are considered and they are replaced by each other after considering the parities.

Column operation



All the column values are added with the received bits and the subtracted by the new obtained value with the old entry.

VI. ACKNOWLEDGMENT

I heart fully thank the management and the members supported for the research.

REFERENCES

[1]R. Gallager, IEEE (1962), “Low-density parity-check codes”,
DOI : 10.1109/TIT.1962.1057683
[2]Rinu Jose; Ameenudeen Pe, IEEE (2015), “Analysis of hard decision and soft decision decoding algorithms of LDPC codes in AWGN”,
DOI: 10.1109/IADCC.2015.7154744
[3]M.P.C Fossorier, Shu Lin, IEEE (1995),
“Soft-decision decoding of linear block codes based on ordered statistics”,
DOI: 10.1109/18. 412683
[4]D.J.C. MacKay, IEEE (1999), “Good error-correcting codes based on very sparse matrices”,
DOI: 10.1109/18.748992
[5]Chutima Prasartkaew; Somsak Choomchuay, IEEE (2009), “A design of parity check matrix for irregular LDPC codes”,
DOI : 10.1109/ISCIT.2009.5341252
[6]Cheng Kun, Shen Qi, Liao Shengkai, Peng Chengzhi, IEEE (2019), “Implementation of encoder and decoder for LDPC codes based on FPGA”,
DOI : 10.21629/JSEE.2019.04.02
[7]John C. Porcello, IEEE (2014), “Designing and implementating Low Density Parity Check (LDPC) Decoders using FPGAs”
DOI : 10.1109/AERO.2014.6836261
[8] Venkata K. Kidambi Srinivasan, Chitranjan K. Singh, and Poras T. Balsara,
IEEE (2009) “A Generic Scalable Architecture for Min-Sum/Offset-Min-Sum Unit for Irregular/ Regular LDPC Decoder”,
DOI: 10.1109/TVLSI.2009.2023659
[9] Kai sun, Yingjie Lao, Weiqiang Liu, Xiaohu You, and Chuan Zhang, IEEE (2017), “Application of LDPC Codes on PUF Error Correction Based on Code-offset Construction”
DOI: 10.1109/ASICON.2017.8252614
[10] Xiaofu Wu; Yue Song; Ming Jiang; Chunming Zhao, IEEE (2010), “Adaptive-Normalized/Offset Min – Sum Algorithm”,
DOI: 10.1109/LCOMM.2010.07.100508