International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 03 Issue: 03 | Mar-2016 www.irjet.net p-ISSN: 2395-0072

Design and Implementation of LDPC codes and TURBO Codes using FPGA

Nikita J. Gaurihar[1], Ishita R. Khadse[2], Trivenee S. Ghonade[3], Amit Borkar[4], Ashish Singh[5], Mrs. M. R. Patil[6]

[12345]*Electronics & Communication Department, Dr. Babasaheb Ambedkar CER, Nagpur, India*
[6]*Asst. Prof & HOD of Electronics & Communication Department, Dr. Babasaheb Ambedkar CER, Nagpur, India*

Abstract - *Error correction code technique are utilized for correction of error in the transmitted data at receiver end. Thus a brief comparison of LDPC codes and turbo codes allows significant reduction in memory consumption and provides the encoder design a great complexity. Turbo codes has become the coding technique of choice in many communication and storage system due to its near Shannon limit error correction capacity. In this paper we are undertaking the comparison of LDPC and TURBO codes based on several parameters like speed, efficiency and delay. This parameters will be judged on the basis of bit error rate ratio (BER). Thus this paper will consider both the classes of codes and compare the performance and complexity of these codes.*

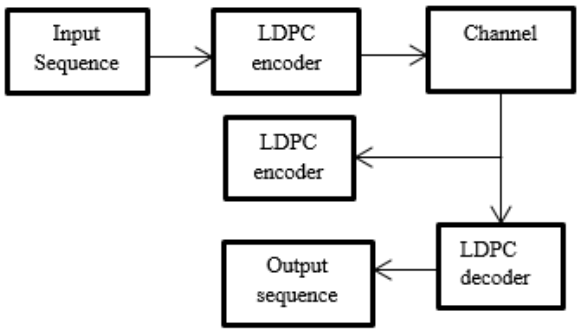
1. INTRODUCTION

Digital communication deals with efficient sending and receiving of data wirelessly without corrupting the original information. Hence we often use codes to improve performance of the system. Thus we require codes with less memory storage and area along with reduction in coding delay. Error correcting codes like LDPC and TURBO codes are the combinational objects designed to cope with the problem of reliable transmission of information on a noisy channel. Codes are used to detect the error and to correct those error. Thus the main objective of this paper is to compare the LDPC codes and turbo codes in VHDL using Modelsim SE 6.3f. First of all the design of the encoder and the decoder for LDPC and TURBO code is done[1].

2. LDPC CODES

LDPC codes were first introduced by Robert G. Gallager in his PhD thesis in 1960, but due to the computational efforts in implementing encoder and decoder and the introduction of the reed Solomon codes, they were mostly ignored until recently.

The basic block diagram of LDPC codes is shown in fig(a). Low density parity check codes are a class of recently rediscovered highly efficient linear block codes made from many single parity check codes (SPC).They can provide performance very close to the channel capacity using an iterated soft decision decoding approach at linear time complexity in terms of their block length. In day to day life, When one party wishes to communicate a message to another distant party, more often than not, the intervening communication channel is noisy and distorts the message during transmission[2]. The problem of reliable communication of information over such a noisy channel is a fundamental and challenging one. One of the error correcting codes like LDPC codes are the objects designed to cope with this problem. Thus, LDPC codes can provide a systematic way of adding redundancy to message before transmitting message to the receiver to figure out the original message that the sender had intended to transmit.



Fig(a) Block Diagram of LDPC codes

At encoder, it will now generate a code for the dedicated input sequence given and forms redundancy for the corresponding input sequence. This code word is then passed through the channel. Channel is the most important part of the communication system while taking into consideration the efficient parameters. If the channel is noisy, the data is ought to get corrupted and the received signal gets distorted [3].

So to check whether reliable transmission of information is done, again this received data is forced to pass through encoder to get the code for the received signal. Then the code words for both, the transmitted message and the received message is compared to detect the error. This can be done using bit error rate ratio (BER) which can be discussed further.

At the decoder side, this code word which are formed by xoring the adjacent bits are checked using BER ratio with the original code word, the corrupted bit positions can be detected using BER and becomes easy to correct the error and get the original information[5].

3. TURBO CODES

Turbo code is a great achievement in the field of communication system. It can be created by connecting a turbo encoder and decoder serially. A turbo encoder is build with parallel concentration of two simple convolution codes. By varying the number of memory element, code rate, block size of data and interleaver unit . BER performance depends on the interleaver size.

Turbo codes have received considerable attention and first came in existence in 1993.Turbo codes provides good performance with low implementation cost. Turbo codes uses convolutional encoder for encoding of codes and a Viterbi decoder for decoding of codes[4].

A Convolution encoder is a finite state machine that processes information bits in serial manner. A convolution coding is done by combining the fixed number of input. The input bit is stored in the fixed length shift register and they are combined with the help of adder. This operation is equivalent to binary convolution coding. A convolution encoder is shown in fig(b).

Whenever the message bit is shifted to position 'm0' the new values of v0, v1 and v2 are generated depending upon m0,m1 and m2. m1 and m2 store the previous two message bits. The current bit is present in m0.

Thus we can write :

$$v0 = m0 \oplus m2 ,$$
$$v1 = m0 \oplus m1 \oplus m2 ,$$
$$v2 = m0 \oplus m1 \oplus m2.$$

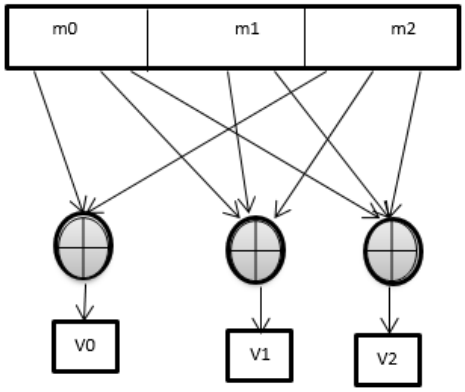
The shift register then shifts contents of m1 and m2 and contents of m0 to m1. The next input bit is then taken and stored in m0. Again v0,v1 and v2 are generated according to this new combination of m0,m1,m2.

Here for every input message bit three encoded output bits v0,v1 and v2 are transmitted. Number of message bits,k=1, number of encoded output bits for one Message bits=3, therefore r = k/n = 1/3.
It is observed that whenever particular message bit enters a shift register, it remains in shift register for three shift.

So for every 1 bit we will get 3 bit output data.we are doing encoding of 12 bits and so we will be getting a 36 bit output sequence.

At the decoder side, this code word is of 36 bits is given as an input. As per the rate of the convolutional encoder, there will be a 12 stage decoder for 36 bit input at decoder. But since it is impossible to draw the Viterbi diagram for 12 stage, let us consider an example of 10 bits input for a ½ rate encoder.

Let's represent the received signal by Y. convolution encoding operates continuously on input data. Hence there are non code vector and block as such. Let's assume that the transmission error probability of symbol 1's and 0's is same[7].



Fig(b): block diagram of convolutional encoder

Let's define an integren variable metric as follows.

Metric: It is the discrepancy between the received signal y and the decoded signal at particular node. This metric can be added over few nodes for a particular path.

Surviving: This is the path of the decoded signal with minimum metric.

In viterbi decoding, a metric is assigned to each surviving path. (Metric of a particular path is obtained by adding individual metric on the nodes along that path). Y is decoded as the surviving path with smallest metric.

Consider the following example of Viterbi decoder. Let this signal being received is encoded by the encoder. For this encoder code trellis is shown fig(c).let the first ten received bits be Y=1101011001 these output are received at the decoder and represented by Y. Thus Y above represents the output for three successive message bits. Assume that the decoder is at same node a0.

Consider the code trellis dig of fig(c) for this encoder. It shows that if the current state is 'a', then next state will be 'a'or 'b'. this is shown in fig(d). Two branches are shown from a0. One branch is at next node a1 representing decoded signal as 00 and other branch is at b1 representing decoded signal as11.

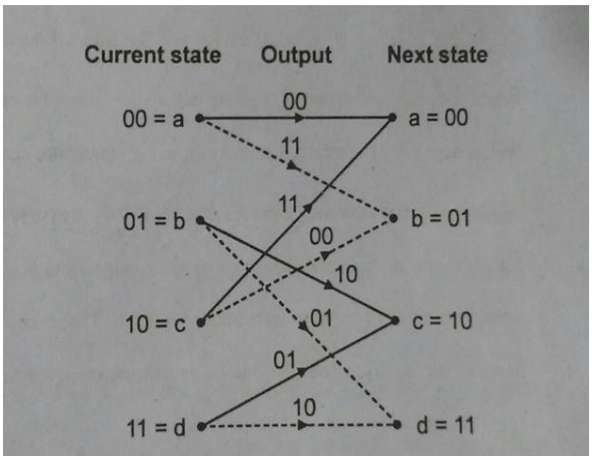
The branch from a0b1 represents decoded output as 11 which is same as received signal at that node i.e.11. Thus there is no discrepancy between received signal and decoded signal. Hence metric of that branch is 0. This metric is shown in brackets along that branch.

The metric of branch from a0 to a1 is 2. The encoded number near a node shows path metric reaching to the node. When the next part of bits Y=01 is received at nodes a1 and b1 then from nodes a1 and b1 four possible next states a2,b2,c2 and d2 are possible.

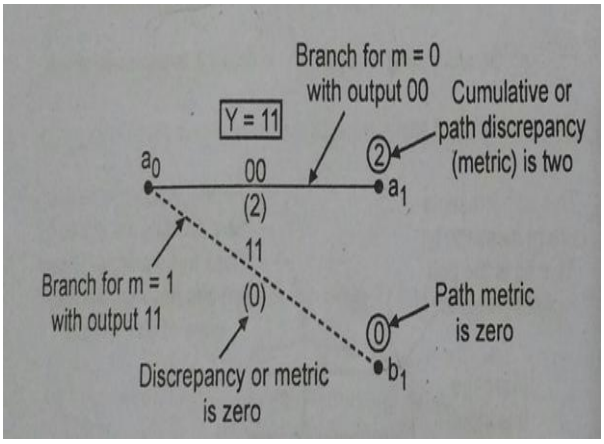
Fig(d): shows all these branches, their decoded outputs and the branch metric corresponding to those decoded output. The number near a2,b2,c2 and d2 indicate the path metric emerging from a0.

For example the path metric of path a0,a1,a2 is 'three'. The path metric of a path a0b1d2 is zero. Fig(e) shows the trellis diagram for all the ten bits for Y. fig(d): shows the possible path for decoder. fig(e) shows the nodes with their path metric on the right hand side at the end of tenth bit of Y. Thus two path are common to node 'a'[6].

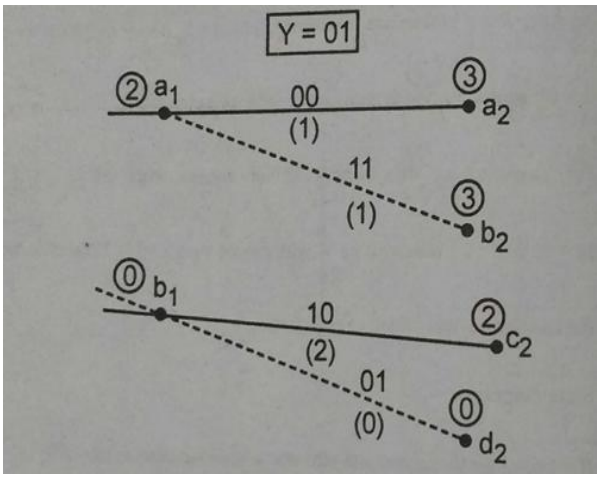
Similarly there are two paths at the other nodes also. According to Viterbi decoding, only one path with a lower metric should be retained at a particular node. As observed in fig(e): the paths that have higher metrics than other path coming to that particular node are not considered. These four paths with a lower metrics are stored in the decoder and the decoding continues to next received bits.



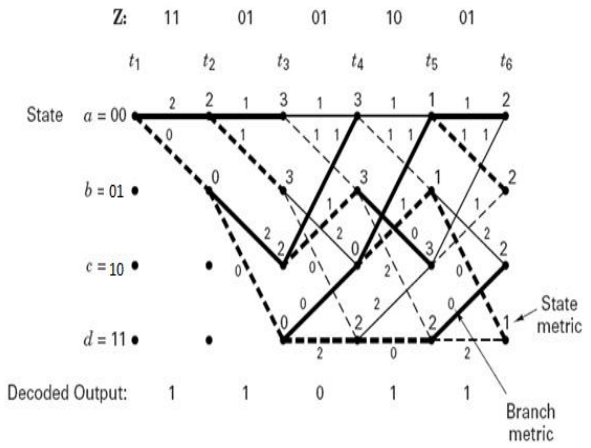
Fig(c): code trellis of convolutional encoder



Fig(d):Viterbi decoder results for first message bit.



Fig(d):Viterbi decoder results for second message bit



Fig(e):path and their metrics after viterbi decoding

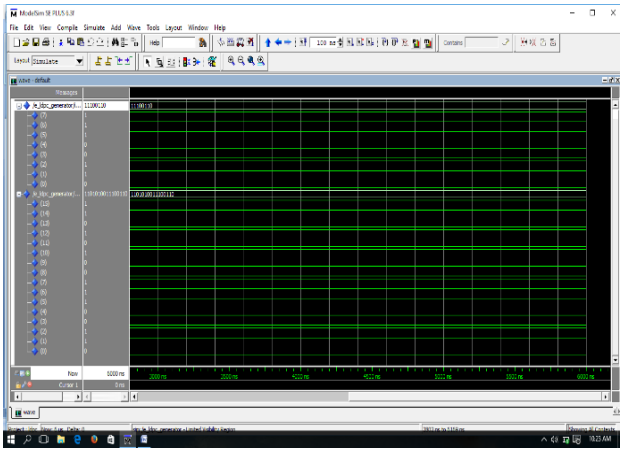
4. BIT ERROR RATE

In digital transmission, the number of bit errors is the number of received bits of a data stream over a communication channel that have been altered due to noise, interference, distortion or bit synchronization errors.

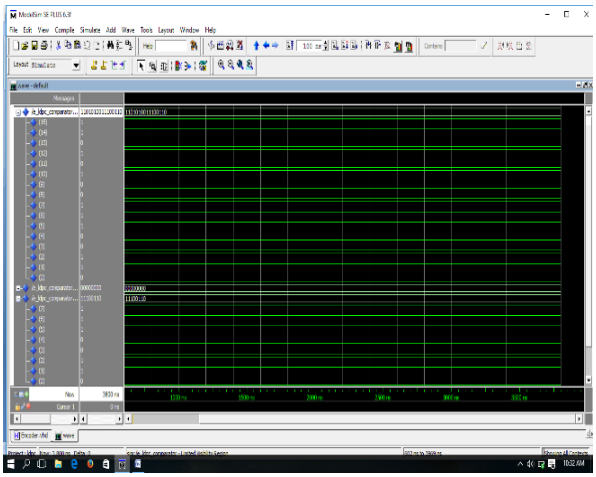
The bit error rate is the number of bit errors per unit time. The bit error ratio is the number of bit errors divided by the total number of transferred bits during a studied time interval.

5. RESULT

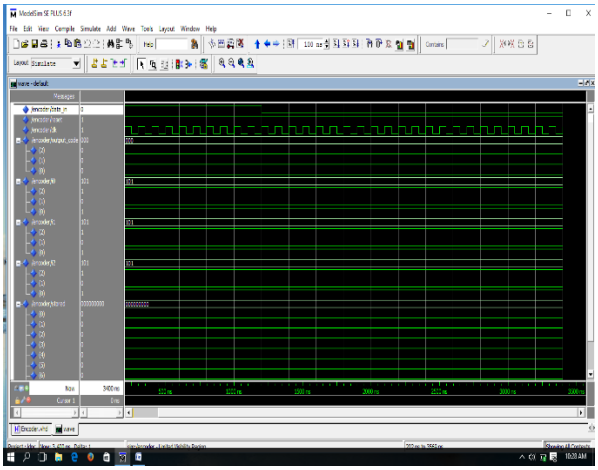
Thus we have studied and performed the encoding and decoding of LDPC and TURBO codes in VHDL using modelsim 6.3f software. The corresponding outputs of LDPC encoding, TURBO encoding, LDPC decoding is shown below. Turbo decoding is yet to perform but estimated results are expected depending upon the progress made till now.



Fig(f): LDPC encoder



Fig(g): LDPC decoder



Fig(h): TURBO encoder

6. CONCLUSION

Thus, theoretically it is concluded that decoding complexity is higher in LDPC codes than in TURBO codes. But LDPC are more efficient codes than TURBO codes. It is also observed that LDPC execution time is low while that of turbo is high. Also, LDPC has low BER ratio than TURBO codes.

7. FUTURE SCOPE

We can dump these codes on FPGA kit for more higher and complex input sequence

8. REFERENCES

[1] Ahmad Hasan Khan , Dr K C. Roy ,Comparison Of Turbo Codes And Low Density Parity Check Codes,

Volume 6,Issue 6,(IOSR-JECE), (Jul. – Aug. 2013), PP 11- 18.

[2] Mr. Shuang Wang, Mr.Lijuan Cui, Mr. Samuel Cheng, Mr.Yan Zhai, Mr. Mark Yeary, and Mr.Qiang Wu. "Noise Adaptive LDPC Decoding Using Particle Filtering." Education, Volume. 59, NO. 4, APRIL 2011.

[3] Mrs. Sarah J. Johnson on "Introducing Low-Density Parity-Check Codes".

[4] Mr. Grace Oletu and Mr. Predrag Rapajic , "The Performance Of Turbo Codes For Wireless Communication Systems.", Computer research and development (ICCRD),IEEE, VOL. 4, March 2011.

[5] T Brack, M Alles, T Lehnigk-Emden, F. Kienle , N. Wehn, N.E.L'Insalata, F. Rossi, M. Rovini, L. Fanucci," Low Complexity LDPC Code Decoder For Next Generation Standards", IEEE, April 2007.

[6] C. Berrou, A. Glavieux, and P. Thitimajshima. Near Shannon limit error correcting coding and decoding: Turbo codes. In Proceedings of the IEEE International Conference on Communications, Geneva, Switzerland, May 2003.

[7] A. M. Viterbi. Shannon theory, concatenated codes and turbo coding. http://occs.donvblack.com/viterbi/index.htm, 1998.