Low-Density Parity-Check(LDPC)decoder using low complexity min sum algorithm

[1]Anjitha V, [2]Prof Sadanandan G K
[1]PG Student, [2] Professor
ECE department, TocH Institute of science and technology, Arakkunnam, Ernakulam

***Abstract* - In a communication system, signal is transmitted by the transmitter and is received by the receiver. But due to various external noises, errors occur in the system. To eliminate the errors various error correction techniques are used. Low-density parity-check LDPC decoder is one such technique. LDPC codes are block codes. For this, various decoding algorithms were introduced of which min sum algorithm is the most efficient method.**

***Index Terms* – LDPC ,hard decision decoding, soft decision decoding, minsum algorithm**

I. INTRODUCTION

In a communication system, information is transmitted by the transmitter and is received by the receiver. Errors can occur in the system due to a lot of reasons like noisy channel, scratches on CD or DVD, power surge in electronic circuits, etc. It is often desirable to detect and correct these errors.Other wise information loss will occur. Therefore, redundancy is used in error-correcting schemes..

A metric used to determine the "closeness" between two bit patterns is called Hamming distance. The Hamming distance between two bit patterns is defined as the number of bits that are different. Communication involves transmission of the data by the encoder where redundant bits are added and then transmitted it through the channel and in the decoder the redundant bits are removed there by original information is obtained. According to the Shannon's coding theorem, the data rate should be less than the channel capacity. But this is not possible under all circumstances. [1]

II. REPRESENTATIONS FOR LDPC CODES

There are two different ways of representing LDPC codes.It can be represented by using matrix representation aswell as graphical representation.

By matrix representation, It can be represented as a parity check matrix with dimension n×m. Two numbers are used for describing these matrix. Wr and Wc. Wr define the number of 1's in each row and Wc for the columns. For a matrix to be called low density the two conditions Wc <<n and Wr<< m must be satisfied.

Tanner introduced the graphical representation of LDPC codes.So it is also called as Tanner graph..Tanner graphs are also called as bipartite graphs. That means that the nodes of the graph are separated into two distinctive sets. The two types of nodes in a Tanner graph are called variable nodes and check nodes.[2]

III. DECODING LDPC CODES

There is mainly two type of decoding .Hard decision decoding and soft decision decoding. In hard-decision algorithms, only binary values are used. The algorithm will be explained on the basis of the example.Let an error free received codeword would be e.g. c = [1 0 0 1 0 1 0 1].Suppose that an error is occurred in the channel and the received the codeword with one error ie bit c1 flipped to 1. The algorithm uses the following steps to correct the error.

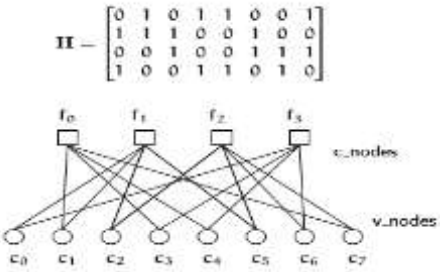

Fig 1: Tanner graph

1. In the first step all variable nodes send a message to check node that containing the bit they believe to be the correct one for them.

c-node	received/sent				
f_0	received:	$c_1 \to 1$	$c_3 \to 1$	$c_4 \to 0$	$c_7 \to 1$
	sent:	$0 \to c_1$	$0 \to c_3$	$1 \to c_4$	$0 \to c_7$
f_1	received:	$c_0 \to 1$	$c_1 \to 1$	$c_2 \to 0$	$c_5 \to 1$
	sent:	$0 \to c_0$	$0 \to c_1$	$1 \to c_2$	$0 \to c_5$
f_2	received:	$c_2 \to 0$	$c_5 \to 1$	$c_6 \to 0$	$c_7 \to 1$
	sent:	$0 \to c_2$	$1 \to c_5$	$0 \to c_6$	$1 \to c_7$
f_3	received:	$c_0 \to 1$	$c_3 \to 1$	$c_4 \to 0$	$c_6 \to 0$
	sent:	$1 \to c_0$	$1 \to c_3$	$0 \to c_4$	$0 \to c_6$

Table 1.1 overview over messages received and sent by the c-nodes in step 2 of the message passing algorithm

2. In the second step every check nodes calculate the response for every connected variable node. This response message contains the bit that fj believes to be the correct one for this.Table 1.1 gives an overview about this step.
3. Next phase: variable node receive a message back from checknode to check whether the received bit is correct or not. This is checking by a majority vote. Table 1.2 illustrates this step. Now the variable node send a message with their correction to check node.
4. Go to step 2.After so many iteration the correct decision is made. [3]

v-node	y_i received	messages from check nodes		decision
c_0	1	$f_1 \to 0$	$f_3 \to 1$	1
c_1	1	$f_0 \to 0$	$f_1 \to 0$	0
c_2	0	$f_1 \to 1$	$f_2 \to 0$	0
c_3	1	$f_0 \to 0$	$f_3 \to 1$	1
c_4	0	$f_0 \to 1$	$f_3 \to 0$	0
c_5	1	$f_1 \to 0$	$f_2 \to 1$	1
c_6	0	$f_2 \to 0$	$f_3 \to 0$	0
c_7	1	$f_0 \to 1$	$f_2 \to 1$	1

Table 1.2 Step 3 of the described decoding algorithm

IV. ENCODER

During encoding, based on the input bits, redundant bits (to help the recovery of the lost data) are added to the input sequence in order to form a code word. To distinguish between the message bits and parity bits in the code word we re-write the code parity-check constraints so that each one solves for a different code word bit .The code word bits c1, c2, and c3 contain the three bit message c1,c2,c3 while the code word bits c4, c5 and c6 contain the three parity-check bits.

Parity check bits can be calculated using the equations

$$c4 = c1 \oplus c2$$
$$c5 = c2 \oplus c3$$
$$c6 = c1 \oplus c2 \oplus c3$$

V. DECODER

The minsum algorithm here used for decoding of LDPC codes is a type of parallel iterative soft decoding algorithm.At each iteration first messages are sent from variable nodes to the check nodes, then it send back to variable node .A probability of each bit is computed and finally a hard decoding decision is made to find the correct message .

At iteration I, Variable node is updated using the equation:

$$m_{vc}^{(l)} = \begin{cases} \ln L(v^{(0)}).....ifl = 0 \\ \ln L(v^{(0)}) + \sum_{c' \& c_v \setminus \{c\}} m_{c'v}^{(l-1)} ...ifl \geq 1 \end{cases}$$

Where $\ln L(v^{(0)})$ represents the log likelihood ratio for the input data bit and v denotes the message sent from variable node to check node c at iteration 1.

Check node is updated by using the equation

$$m_{cv}^{(l)} = \prod_{v' E v_c \setminus \{v\}} \operatorname{sgn}(m_{v'c}^{(l)}) . \min_{v' E v_c \setminus \{v\}} \left| m_{v'c}^{(l)} \right|$$

$m_{cv}^{(l)}$ denotes the message sent from check node c to variable node v and sgn(v) denotes the sign of v.[4][5]

VI. MIN-SUM ALGORITHM

Initial Variable node To Check Node Pass: At beginning of the decoding process, variable nodes receive the LLR"s. Variable nodes pass these values to check nodes.

1st Iteration:

I)Check node to variable node pass:A minimum of absolute value is calculated and is send to variable nodes.
2)Variable node to check node pass: A variable node sums up all the information and send to the check node.
Subsequent Iterations:Above process is repeated until the error is corrected.

VII. SIMULATION RESULTS

During encoding, based on the input bits, parity bits (to help the recovery of the lost data) are added to the input sequence in order to form a code word. Simulation of decoder by hard decision algorithm is done in VHDL by using model sim . Error is injected to the channel and obtained an error correction up to 3bits. The system can be tested by comparing the input of encoder and the output of decoder.

Fig2:Simulation of hard decision decoding

Input of encoder	Output of encoder	Input of decoder	Output of decoder	Error?
0000000	0000000000000000	1000000000000000	0000000	No error
0000001	0000001000010111	1000001000010111	0000001	No error
1000001	1000001110011110	0000001110011110	1000001	No error
0000001	0000001000010111	0100001000010111	0000001	No error
1000001	1000001100111000	1100001100111000	1000001	No error
0000001	0000001000010111	0010001000010111	0000001	No error
1000001	1000001100111000	1010001100111000	1000001	No error

Fig 3:Comparision of input of encoder and the output of decoder

min-sum algorithm approximates the sum-product algorithm with simple check node update operations, that significantly reduce the VLSI implementation complexity .

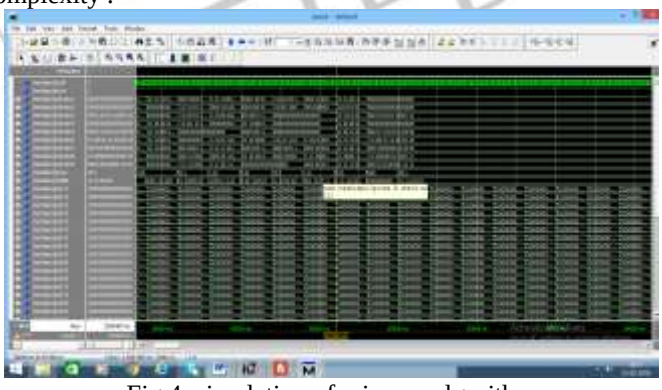

Fig 4: simulation of minsum algorithm

VIII. CONCLUSION

Amoung the various algorithm proposed for the decoding of LDPC codes ,min-sum algorithm can be easily implemented and it reduces the area and the power consumption of the decoder. Its efficiency makes it ideal for implementation.It can be done by simple check node update operations,there by reduces the VLSI implementation complexity. In this algorithm Complex exponential or logarithmic calculations are avoided thus leading to reduced hardware complexity.

REFERENCES

[1] Zhou Zhongl , Yunzhou Lil and Xiang Chenl, "Modified Min-sum Decoding Algorithm for LDPC Codes Based on Classified Correction"IEEE 2006

[2] Sarah J. Johnson , "Introducing Low-Density Parity-Check Codes"

[3] R. Gallager, "Low density parity-check codes," IRE Trans, Information Theory, pp. 21-28. January 1962

[4] Tuan Ta , "A Tutorial on Low Density Parity-Check Codes"

[5] Bernhard M.J. Leiner, LDPC Codes – a brief Tutorial