RESEARCH ARTICLE OPEN ACCESS

Optimization and simulation of a New Low Density Parity-Check Decoder using the Syndrome Block

Lagrini Lakbir[1], Sedra Moulay Brahim[2]
[1,2]Laboratory of High Energy and Engineering Science sand Reactors the University Ibn to fail of Kenitra at Morocco.

ABSTRACT
The Low Density Parity-Check codes are one of the hottest topics in coding theory nowadays equipped with very fast encoding and decoding algorithms, LDPC are very attractive both theoretically and practically.In this article, we present a simulation of a work that has been accepted in an international journal. The newalgorithm allows us to correct errors quickly and without iterations. We show that the proposed algorithm simulation can be applied for both regular and irregular LDPC codes. First, we developed the design of the syndrome Block Second, we generated and simulated the hardware description language source code using Quartus software tools, and finally we show low complexity compared to the basic algorithm.
Keywords-Bit Flipping Algorithm, Low Density Parity Check decoder, proper syndrome, simulation of the syndrome block.

I. INTRODUCTION

The Low density parity-check code (LDPC) is an error correcting code used to protect information against noise introduced by the transmission channel and reduce the probability of loss information.Grace Low Density Parity Check, the information transmission rate can be as close to Shannon limit in the same noisy channel. LDPC was developed by Robert Gallager [1] in his doctoral dissertation at MIT in 1960. Theses codes were ignored until 1981when R. Michael Tanner gave them a new interpretation of the graphical representation of the codes [2]. In 1993, with the invention of turbo codes, researchers switched their focus to finding low complexity code which can approach Shannon channel limit. Nowadays, LDPC have made its way into some modern applications such as 10GBase-T Ethernet, Wi-Fi, Wi MAX, Digital Video Broadcasting (DVB) [3], which is based on hard-decision decoding, has the least complexity but suffers from poor performance of iteration.

In this paper, a low complexity for the syndrome Block algorithm of LDPC decoding is proposed to achieve a trade-off between implementation complexities compared to the bit-flipping algorithm (BFA),particularly in the second part of the table. At the same time this method reduces the number of iteration by introduction the simulation of a new concept of proper syndromes in part 3.

II. Bit-Flipping Algorithm

The Bit-Flipping algorithm is based on hard-decision message passing technique. A binary hard-decision is done on the received channel data and then passed to the decoder. The messages passed between the check node and variable nodes are also single-bit hard-decision binary values. The variable node (r) sends the bit information to the connected check nodes (S) over the edges. The check node performs a parity check operation on the bits received from the variable nodes. It sends the message back to the respective variable nodes with a suggestion of the expected bit value for the parity check to be satisfied [5].

The algorithm:

Step 1: We use the equation $S = Hr^T$ to calculate the syndrome with the received vector. If the elements in the set of S are all zeros, it's terminated with the correct vector, otherwise, go to the next step.

Step 2: Calculate the set of {f0, f1, · · ·, fN−1} and find the largest fj .Then transfer the corresponding rj to its opposite number (0 or 1), get a new vector r'.

Step 3: Calculate the vector $S = Hr^T$ with the new vector r'. If the elements of S are all zeros or the iterations reach the maximum number, the decoding is terminated with the current vector, otherwise, the decoding go back to step 2.

Clearly the BFA has simple check node and variable node operations, thus making it a very low complexity decoding algorithm compared to the other algorithms. But this advantage comes with a poor decoding performance.

III. proposed algorithm

3.1 Construction of a new method of LDPC codes Table

The proposed algorithm consists of accomplishing a table, which is divided into three parts or steps.

Step 1: (P1) contain parity check equations.

Step2: (P2) which runs diagonally includes some possible syndromes from parity syndromes of part 1, without making any calculations and without any iteration. All we need to do is flipping the variable node r_i in order to reach a Null Syndrome.

Step 3: (P3) we introduce the notion of proper syndrome of the second part, this may we only need to make the addition of the proper syndrome of the second part in order to find the remaining possible syndromes bearing in mind the algorithm of the flowing calculation:

3.2. Parity check matrix Hofthe dimension 3x7:

Here we will only consider binary messages and so the transmitted messages consist of strings of 0's and 1's.

Example1:

Example parity check matrix

$$H = \begin{bmatrix} 1001011 \\ 0101110 \\ 0010111 \end{bmatrix}$$

In the first place,

We calculate the syndrome S of the received codeword r such as S=Hr^T.

 If Hr^T=0 (null syndrome) then the received codeword is correct, therefore terminate the algorithm, decoding it with the corresponding message.

 If $Hr^T \neq 0$, the non-zero syndrome, therefore the codewordreceived is incorrect.

The proposed algorithm can determinate the variable node$r_i$that we have toflip in order to find a null syndrome.

Supposethe codeword received after the transmissionchannel is $r = r_1r_2r_3r_4r_5r_6r_7$where each r_i is either 0 or 1

AndHr^T = $S_1S_2S_3$ (the syndrome S=$S_1S_2S_3$)

$$\begin{pmatrix} 1001011 \\ 0101110 \\ 0010111 \end{pmatrix} \begin{pmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \\ r_5 \\ r_6 \\ r_7 \end{pmatrix} = (S_1S_2S_3)$$

Each row of H gives a parity check equation:

$$S_1 = r_1 \oplus r_4 \oplus r_6 \oplus r_7$$
$$S_2 = r_2 \oplus r_4 \oplus r_5 \oplus r_6$$
$$S_3 = r_3 \oplus r_5 \oplus r_6 \oplus r_7$$

Table1.différent cases of syndrome for LDPC (7, 3) codes

$s_1s_2s_3$	r_1	r_2	r_3	r_4	r_5	r_6	r_7	$[pp.s]$	$[pi]$
s_1	x			x		x	x		p_1
s_2		X		x	x	x			
s_3			x		x	x	x		
100	x							s_1	p_2
010		X						s_2	
001			x					s_3	
110				x				s_1s_2	
011					x			s_2s_3	
111						x		$s_1s_2s_3$	
101							x	s_1s_3	

$[pi]$:Parts$_i$

$[pp.s]$: Proper syndrome

X in $p_2$representsr_i which we have to flip in order to find anull syndrome.

3.2.1 Explanation of the table:

Example1:

Explanation of the proper syndrome:

In column six (column r_5), the case between line S_1 and column r_5($part$ 1) is empty, but between line S_2 and column r_5 ($part$ 1), there is an X and between line $S_3$and column r_5 ($part$ 1) there is also an X, which gives the following Expression S_2S_3 (proper syndrome in part 2), which is represented in binary by 011(column 1 part 2).

Example 2:

Supposing after the calculation of Hr^T, we find the syndrome 011 which, according to the Proposed Algorithm, is equivalent to S_2S_3

In this case we only need to flip r_5 to get the null syndrome

If r_5=1 will be changed to 0

If r_5=0 will be changed to 1

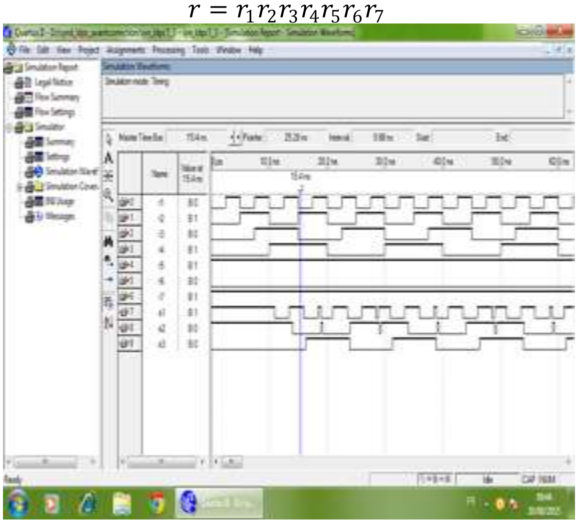
We will explain how to read this table:

- ϖ The first part (P1) represents parity check equation S_1, S_2 and S_3 the X represents the $r_i$forming the S_i
- ϖ The second part (P2) which runs diagonally in the table represents the possible syndromes from equations of part 1 without making any calculations. And the X represents the r_i(=1 or 0) which we have to flipping in order to get the syndrome which corresponds to zero.
- ϖ The third part (P3) which is in the other tables represents the rest of the possible syndromes which are possible to calculate by introducing the new notion of adding proper syndromes of part 2 (P2).

3.2.2 Simulation results of the LDPC code (7.3) with Quartus before the error correction:

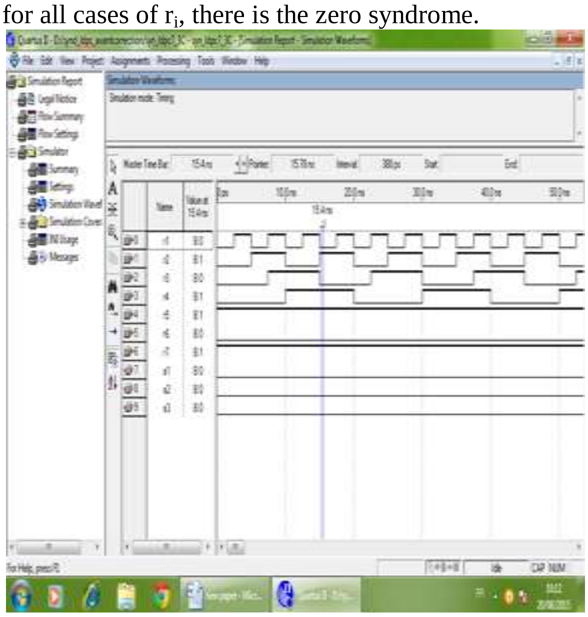
The decoder receives the different code word erroneous by the transmission channel .the VHSIC Hardware Description Language (VHDL) that will calculate various cases possible syndromes based on the equation S =Hr^T.Such as S=$S_1S_2S_3$ and

$$r = r_1r_2r_3r_4r_5r_6r_7$$



3.2.3Simulation results of the LDPC (7, 3) code with Quartusafterthe correction of errors

for all cases of r_i, there is the zero syndrome.



3.3Parity check matrix H of the dimension 4x9 Parity check matrix

$$\boldsymbol{H} = \begin{bmatrix} 011011000 \\ 101100100 \\ 111010010 \\ 000110001 \end{bmatrix}$$

In the first place,

We calculate the syndrome S of the received codeword r such as S =Hr^T

ϖ IfHr^T=0(null syndrome) then the received codeword is correct, therefore terminate the algorithm, decoding it with the corresponding message.

ϖ If Hr^T ≠0, the non-zero syndrome, therefore the codeword received is incorrect.

The proposed algorithm can determinate the variable node r_i that we have to flip in order to find a null syndrome,

Suppose the received codeword after the transmission channel is $r = r_1r_2r_3r_4r_5r_6r_7r_8r_9$Where each$r_i$ is either 0 or 1and Hr^T = $S_1S_2S_3S_4$(the syndrome S= $S_1S_2S_3S_4$).

$$\begin{pmatrix} 011011000 \\ 101100100 \\ 111010010 \\ 000110001 \end{pmatrix} \begin{pmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \\ r_5 \\ r_6 \\ r_7 \\ r_8 \\ r_9 \end{pmatrix} = S_1S_2S_3S_4$$

Each row of H gives a parity check equation:

$$S_1 = r_2 \oplus r_3 \oplus r_5 \oplus r_6$$
$$S_2 = r_1 \oplus r_3 \oplus r_4 \oplus r_7$$
$$S_3 = r_1 \oplus r_2 \oplus r_3 \oplus r_5 \oplus r_8$$
$$S_4 = r_4 \oplus r_5 \oplus r_9$$

Table2.différent cases of syndrome for LDPC (9, 4) codes

$s_1s_2s_3s_4$	r_1	r_2	r_3	r_4	r_5	r_6	r_7	r_8	r_9	$[pp.s]$	$[pi]$
s_1		x	x		x	x					p_1
s_2	x		x	x			x				
s_3	x	x	x		x			x			
s_4				x	x				x		
0110	x									s_2s_3	p_2
1010		x								s_1s_3	
1110			x							$s_1s_2s_3$	
0101				x						s_2s_4	
1011					x					$s_1s_3s_4$	
1000						x				s_1	
0100							x			s_2	
0010								x		s_3	
0001									x	s_4	
1111		x		x						$s_1s_2s_3$	p_3
0111	x								x	$s_2s_3s_4$	
1101				x		x				$s_1s_2s_4$	
1001						x			x	s_1s_4	
0011								x	x	s_3s_4	
1100	x	x								s_1s_2	

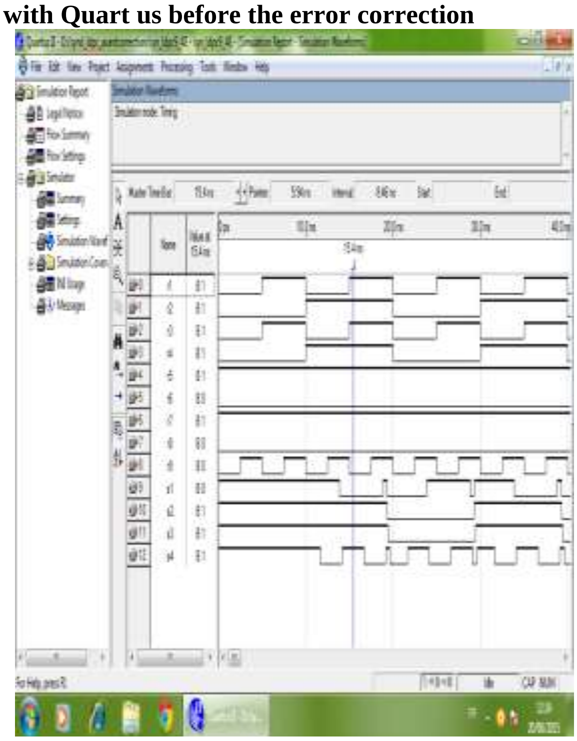
3.3.1Explanation of the table:

If we make the product of Hr^T and we find for example the syndrome 0111 which is equivalent proper syndrome $S_2S_3S_4$ from the proposed algorithm; we will look in the table, especially the two part (P2), the r_i need to flip to achieve null syndrome.

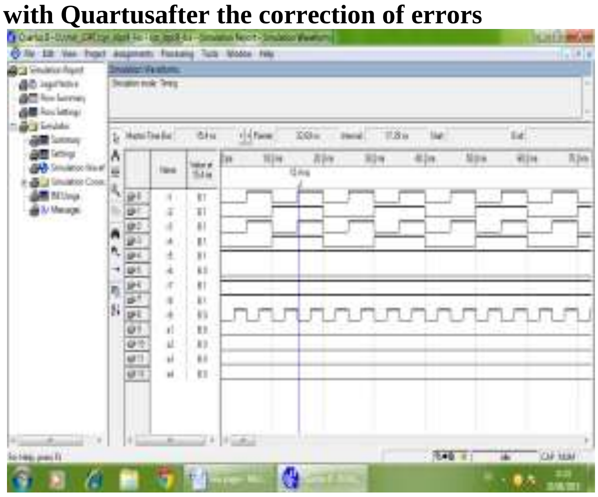
The proposed algorithm uses the addition of proper syndrome as follows:

- $S_2S_3 \oplus S_4 = S_2S_3S_4$ forcing us to flip r1(case of S_2S_3) etr_9 (caseof S_4)or
- $S_2 \oplus S_3 \oplus S_4 = S_2S_3S_4$ forcing us to flip r_7(case of S2), r_8 (case of S3) andr_9(case of S4) or
- $S_2S_4 \oplus S_3 = S_2S_3S_4$ forcing us to flip r_4(case of S_2S_4) andr_8(case of S_3).

3.3.2Simulation results of the LDPC code (9.4) with Quart us before the error correction



3.3.3 Simulation results of the LDPC (9, 4) code with Quartusafter the correction of errors



3.4Comparison of algorithms

The proposed algorithm presents a without iteration and a very good performance compared to the basic algorithm.

	Proposed Algorithm	Basic Algorithm
LDPC (7,3)	Without any iteration for all possible syndromes (2^3-1 possible syndromes)	2^3-1 possible syndromes and 2 possible iterations
LDPC (9,4)	-Part 2 : without iteration -Part 3 :the additions of proper syndromes of part 2	2^4-1 possible syndromes, 4 possible iterations. Case of codeword received is r=101010100
LDPC (10,5)	-Part 2 : without iteration -Part 3 : adding proper syndromes of the part 2	15 possibles syndromes possibles iteration
LDPC (12,6)	-part 2 : without iteration -part 3 :the addition of proper syndromes of part 2	2^6-1 possibles syndromes and 4 possibles iteration
LDPC (n, k)	if $n=2^k-1$ and the columns are linearly independent, we find all directly syndrome without iteration in the part 2 (P2) of the table and permanently delete part 3 (P3).	if $n=2^k$ -1 and the columns are linearly independent, basic algorithm several iteration

IV. Conclusion

- The proposed algorithm based on the new notion of the proper syndrome, and the shape of the table, which allows us in a simple and easy way to find the variable nodes that we have to flip in order to find a null syndrome instead of the classical iteration method.
- The proposed algorithm allows us as well to completely eliminate and delete the classical decoding iteration, particularly in relation to a certain number n of the syndrome, such as n the length of the matrix of the parity H.
- The first part in the table gives directly the variable nodes ri that we have to flip to find a null syndrome, and by the addition of proper syndromes, we can find the ri that we have to flipping in order to have the rest of the possible null syndromes.

¬ The proposed algorithm can be improved in future work about the implementation of carte FPGA, Digital Video Broadcasting-second generation (DVB-S2).

V. Acknowledgements

This work was supported by the Laboratory of High Energy and Engineering Sciences and reactors The University IbnTofail of Kenitra, Morocco.

References

[1] R.G.Gallager, Low-Density Parity-Check Codes. IRE Transactions on Information Theory, January 1962,

[2] R. M. Tanner, "A Recursive Approach to Low Complexity Codes", IEEE Trans. Inform. Theory, vol. IT-27, pp. 533-547, 1981.

[3] AH. El-Maleh, MA. Landolsi and EA. Alghoneim,“Window-constrained interconnect-efficient progressive edge growth LDPC codes”, international journal of electronics and communications VOL. 67, 2013.

[4] Lagrini Lakbir, Sedra Moulay Brahim, Optimization of A New Low Density Parity-Check Decoder using The Syndrome Block, The University Ibn Tofail of Kenitra at Morocco, International Journal Of Scientific Progress And Research (Ijspr) Volume-10, Number - 03, 2015

[5] NP. Bhavsar, B. Vala, “Design of Hard and Soft Decision Decoding Algorithms of LDPC”, International Journal of Computer Applications VOL. 90(4), (2014), http://dx.doi.org /10.5120/15803-4653.