ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

RELIABILITY AND PERFORMANCE METHOD OF CORRECTING ERRORS TRANSMISSION OF LOW DENSITY PARITY CHECK CODE USING THE BIT FLIPPING ALGORITHM

[1]LAGRINI LAKBIR, [2]ELGHAYYATY MOHAMED, [3]El HATI IDRISSI ANAS, [4]HADJOUJA ADELKADER, [5]MOULAY BRAHIM SEDRA

[1]Laboratory of Engineering Sciences and modeling. Faculty of Sciences, University Ibn Tofail Kenitra, Morocco
[2]Laboratory of Electrical Engineering and Energy System. Faculty of Sciences, University Ibn Tofail Kenitra, Morocco
[3]Laboratory of Electrical Engineering and Energy System. Faculty of Sciences, University Ibn Tofail Kenitra, Morocco
[4]Laboratory of Electrical Engineering and Energy System. Faculty of Sciences, University Ibn Tofail Kenitra, Morocco

E-mail : [1]abdo_lgn@hotmail.com, [2]l. lagrini@mem.gov.ma

ABSTRACT

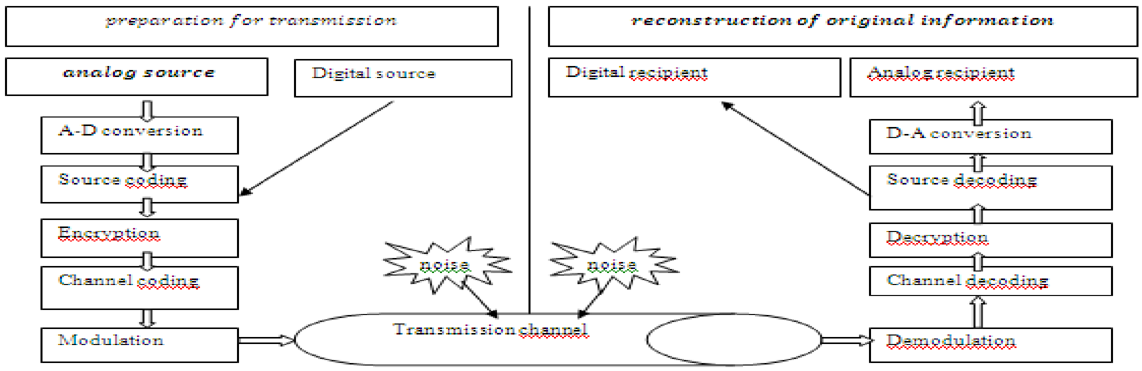
The low density parity check (LDPC) code invented by Robber Gallager in 1962, and rediscovered by Mackey in 1995, after The discovery of turbo codes in 1993 by C. Berrou, A. Glavieux, and P. Thitimajshima at the International Conference of Communication in Florida has revolutionized the means of communication. This code is one of the best performance codes in the correction of transmission errors. The purpose of this article is to study two algorithms, the first algorithm is based on the addition of another specialty to the control matrix such that the control matrix becomes at the same time an error corrector, in order to decode information without iteration compared to the bit flipping decoding algorithm based. This proposed can be applied to LDPC code regular or irregular. the result of this algorithm allows to find the position of variable noed deformed during the transmission without doing any iteration and without using any additional tool, but in this method the response exists in the control matrix, so that each syndrome found represents a column of the control matrix and each column linked by a noed variable denoted ri with i between 1 and n.

The principle for the second algorithm is to write the parity equations in a table which will facilitate the detection and correction of errors without doing the iterations, and for this algorithm, I did the simulation using a hardware description language using Quartus software tools to confirm the reliability and performance of this method. The result for second algorithm, for example the LDPC code of the matrix control H (n, k), the number of syndromes which we can find is 2^k-1 and for each syndrome different from zero, we can do almost four iterations for each nonzero syndrome, which give the number 4x(2^k-1) iterations in totality.

Keywords: *Coding, Decoding Of LDPC Code, Control Matrix.*

1. INTRODUCTION

The Low Density Parity Check (LDPC) is a block code allows of protects and corrects errors introduced by the transmission channel to reduce the probability of information loss. use of the LDPC code allows the probability of errors to become low a value as desired, and the data transmission rate can be as close to the Shannon limit [1].



LDPC code invented by Robber Gallager in 1962 [2], The LDPC code was ignored until 1981 when R. Michael Tanner [3] gave a graphic interpretation of this code LDPC, called Tanner graph, this rediscovery has been overlooked for almost 14 years. With the invention of turbo codes [4] in 1993, researchers turned to the search for low complexity code to arrive at the Shannon theory which shows the existence of a limit to the rate of information transmitted in the presence of noise. In 1995, LDPC was reinvented with the work of Mackay, and R.M. Neal [5]. Nowadays, LDPC has made its way in some modern applications such as Wi-Fi, Wi MAX, Digital Video Broadcasting (DVB)[6].

In recent years, numerous bit flipping decoding algorithms have been proposed [7], [8], [9], [10], [11], [12].However, almost all of these algorithms require some soft information from a channel.

The motivation for scientific research in the field of error correctors comes essentially from the high demand in the industrial telecommunications market and the uses of electronic equipment. The 5G [13] generation will revolutionize the communication system in the future. In this paper, the proposed algorithm also allows for the complete elimination of the decoding iteration compared to the basic algorithm, in this method the response exists in the control matrix, so that each

Figure 1: the general chain of transmission.

syndrome found represents a column of the control matrix and each column linked by a noed variable denoted ri with i between 1 and n.

The result of first proposed algorithm gives directly the variable node ri which must be toggle to find a syndrome equal to zero.

The general description of a digital telecommunications system is shown schematically in Figure 1. The overall architecture of this system was designed according to two steps, the step of preparation for transmission and the step of reconstitution of the

Original information.

2. BASIC ALGORITHM

2.1. The Bit Flipping Decoding Algorithm

The Bit-flipping algorithm is based on difficult decision message transmission technique. A binary hard decision is executed on the received channel data and transmitted to the decoder. The messages passed between the check node and variable nodes [2],[3],[5] are also single-bit hard-decision binary values. The variable node R($R=r_1\ r_2 \ldots r_n$) sends the bit information to the connected check nodes S ($S=S_1\ S_2 \ldots S_k$) 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 [7],[11].

2.2. The Steps of the Decoding Algorithm Generalization

Step 1: We use the equation $S = rH^T$ to calculate the syndrome with the received vector [2], If the binary elements constituting S are all zeros, then the vector received is correct, otherwise, go to the next step.

Step 2: we are going to calculate the set of {f0, f1, , f_{N-1}} and we are looking for 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 $S = rH^T$ 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.

2.3. Diagram of the Algorithm

The diagram 1 below represents an Low Density

Journal of Theoretical and Applied Information Technology
15th November 2020. Vol.98. No 21

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Parity Check decoder by bit flipping algorithm, such that {Ri} i = 1, N represent the code word received

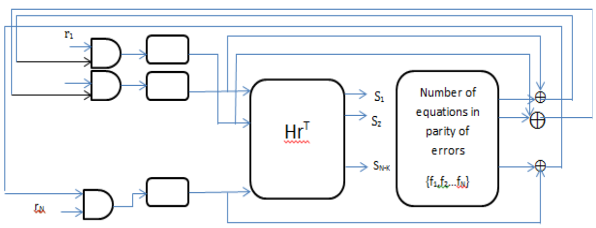


Figure 2: diagram of decoding by flipping algorithm

2.4. Example

Decoding of LDPC code: The parity matrix H of an LDPC code of length N=12 and m=6, consisting of rows such that the number of 1 in a row represents the weight wr=6 and columns such that the number of 1 in a column represents the weight wc=3, is illustrated below:

$$H = \begin{pmatrix} 110100001101 \\ 111110010000 \\ 101011000110 \\ 000111101010 \\ 011000110011 \\ 000001111101 \end{pmatrix}$$

$$rH^T = S_1S_2S_3S_4S_5S_6$$

H^T : Transposed matrix

$$\begin{pmatrix} 110100001101 \\ 111110010000 \\ 101011000110 \\ 000111101010 \\ 011000110011 \\ 000001111101 \end{pmatrix} \begin{pmatrix} R_1 \\ R_2 \\ R_3 \\ R_4 \\ R_5 \\ R_6 \\ R_7 \\ R_8 \\ R_9 \\ R_{10} \\ R_{11} \\ R_{12} \end{pmatrix} = S_1S_2S_3S_4S_5S_6$$

The parity equations resulting from this matrix multiplication:

$$S_1 = R_1 \oplus R_2 \oplus R_4 \oplus R_9 \oplus R_{10} \oplus R_{12}$$
$$S_2 = R_1 \oplus R_2 \oplus R_3 \oplus R_4 \oplus R_5 \oplus R_8$$
$$S_3 = R_1 \oplus R_3 \oplus R_5 \oplus R \oplus R_{10} \oplus R_{11}$$
$$S_4 = R_4 \oplus R_5 \oplus R_6 \oplus R_7 \oplus R_9 + R_{11}$$
$$S_5 = R_2 \oplus R_3 \oplus R_7 \oplus R_8 \oplus R_{11} \oplus R_{12}$$
$$S_6 = R_6 \oplus R_7 \oplus R_8 \oplus R_9 \oplus R_{10} \oplus R_{12}$$

Suppose the vector received is of form:r=[100001101000], we will calculate the S: S= rH^T =[010111] so the vector received incorrect, we will determine for each of the N = 12 bits the number of parity equations in error: According to the value of S = 010111 = $S_1S_2S_3S_4S_5S_6$ so the components of S that have non-zero are S_2 , S_4 , S_5 and S_6.

This node variable R1 exists only in the parity node S2 who is in error, so f1 = 1. On the other hand, R2 exists in the two nodes parity S2 and S5 which have different from zero, so f2 = 2. The same for the other cases, we find all the fi compatible with the Ri such that i integer between 1 and 12:

[f1 =1, f2 =2, f3 =2, f4 =2, f5 =2, f6 =2, f7 =3, f8 =3, f9 =2, f10 =1, f11 =2, f12 =2] So, maximum of all {fi : 1≤ i ≤12}equal {f7, f8}. Then I will toggle R7 and R8, the received message becomes r= [100001011000] (ancient r= [100001101000])

The syndrome S is calculated another time and the number of parity equations in error is sought for each of the N bits received.[f1 =0, f2 =1, f3 =1, f4 =0, f5 =0, f6 =1, f7=2, f8 =2, f9 =1, f10 =1, f11 =1, f12 =2].And max {fi, i between 1 and 12 } is R7, R8, R12.So, I go to toggle the bits R7, R8 and R12 the vector r becomes: r= [100001101001] (ancient r= [100001011000])

And one calculates the syndrome S= [110100]: this one being always different from zero, one carries out a third iteration [f1 =2, f2 =2, f3 =1, f4 =3, f5 =2, f6 =1, f7 =1, f8 =1, f9 =2, f10 =1, f11 =1, f12 =1], so max of {fi, i between 1 and 12 } is R4.

We only toggle the fourth bit of the received vector: r= [100101101001], the calculation of the syndrome gives S = [000011]. So, we did not achieve our goal because the syndrome is different from zero.

In this case, a fourth iteration of the bit flipping algorithm is performed. The errors in the parity equations for each received bit are this time: the calculation of the syndrome is null = [000000]. So the vector is now valid, then we can be decoded into a message.

3. THE FIRST PROPOSED ALGORITHM

3.1. Proposed Algorithm about the Matrix Control

The proposed method based on the realization of two conditions linked by the control matrix:

- The size of the control matrix, especially the numerical relation between the rows and the columns of the matrix so that, if the number of rows is m, the number of columns must be 2^m -1 in this case the vectors of columns represent all the possible syndromes and each vector of the columns occupies a position i among 2^m -1 columns. For example, if we did the calculation of the rH^T syndrome and we found the vector of length m which occupies the position i of the columns, so the error exists in the position i finally, it suffices to toggle the component Ri of the vectors received r = R1 R2… Ri… $R2^m$ -1 to find a null syndrome without doing any iteration.
- The independence of the 2^m -1 vectors of the columns of H (i.e. 2^m -1 different vectors from each other) such that m represents the dimension of the word and 2^m -1 represents the length of the code word and at the same time the length of the control matrix

3.2. Hamming Code of Dimension n = 7 and k = 3

The Hamming code (7,3) fulfills the first condition of the proposed algorithm such that m = 3 and 2m -1 = 7. The control matrix is

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

The first step, We calculate the syndrome S of the code word vector received is $r = R_1 R_2 \ldots R_i \ldots R_{2^m-1}$ such that $S = rH^T$

- if $rH^T = 0$ (null syndrome) so the code word received is correct, so the algorithm is finished, we decode the message received to find the valid code word .
- If $rH^T \neq 0$, the syndrome is not zero, so the code word received is incorrect.

The proposed algorithm allows to determine without iteration the node of variable Ri which it is necessary to flip for find a null syndrome.

3.3. Example 1:

Suppose that the code word received after the transmission channel is r = 0101011. We calculate the rH^T syndrome

$$\begin{pmatrix} 1001011 \\ 0101110 \\ 0010111 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 1 \end{pmatrix} = 110$$

So the syndrome is equal to 110 non-zero.
S = 110 is the same vector of the fourth column of the control matrix H, according to our algorithm it just toggle the fourth component of the vector received c = $010\underline{1}011$ so the valid vector is c = $010\underline{0}011$. I will check that the syndrome is zero

$$\begin{pmatrix} 1001011 \\ 0101110 \\ 0010111 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix} = 000$$

Finally we find the null syndrome.
The verification by the Equations of parity: The Equations of parity of the matrix product are: rH^T = S1S2S3

$$\begin{pmatrix} 1001011 \\ 0101110 \\ 0010111 \end{pmatrix} \begin{pmatrix} R1 \\ R2 \\ R3 \\ R4 \\ R5 \\ R6 \\ R7 \end{pmatrix} = S_1S_2S_3$$

$$S_1 = R_1 \oplus R_4 \oplus R_6 \oplus R_7$$
$$S_2 = R_2 \oplus R_4 \oplus R_5 \oplus R_6$$
$$S_3 = R_3 \oplus R_5 \oplus R_6 \oplus R_7$$

We are going to toggle R4 to find S1 = 0 and S2 = 0, which gives a null syndrome.

3.4. The Control Matrix H of an LDPC Code of dimension n = 15 and m = 4: (m = 4 and 2^m -1 = n)

Let the parity matrix H such that:

$$H = \begin{bmatrix} 000011111111000 \\ 011100011110100 \\ 101101100110010 \\ 110110101010001 \end{bmatrix}$$

The first step we calculate the syndrome S of the vector received r such that S = rHT and the vector received

$$r = R_1R_2R_3R_4R_5R_6R_7R_8R_9R_{10}R_{11}R_{12}R_{13}R_{14}R_{15}$$

- if $rH^T = 0$ (null syndrome) then the code word received is correct, the algorithm finished and the decoding gives the corresponding message.
- if $rH^T \neq 0$, the non-zero syndrome therefore

the code word received is incorrect. The matrix product which will give the syndrome and the equations of parity

$$S_1 = R_5 \oplus R_6 \oplus R_7 \oplus R_8 \oplus R_9 \oplus R_{10} \oplus R_{11} \oplus R_{12}$$

$$S_2 = R_2 \oplus R_3 \oplus R_4 \oplus R_8 \oplus R_9 \oplus R_{10} \oplus R_{11} \oplus R_{13}$$

$$S_3 = R_1 \oplus R_3 \oplus R_4 \oplus R_6 \oplus R_7 \oplus R_{10} \oplus R_{11} \oplus R_{14}$$

$$S_4 = R_1 \oplus R_2 \oplus R_4 \oplus R_5 \oplus R_7 \oplus R_9 \oplus R_{11} \oplus R_{15}$$

This algorithm makes it possible to determine the node of variable Ri which it is necessary to toggle without iteration to arrive at a null syndrome.

3.5. Example 2:

Suppose, we received the code word after the transmission channel in the form r = 100011100101101, we calculate the syndrome

$$\begin{pmatrix} 000011111111000 \\ 011100011110100 \\ 101101100110010 \\ 110110101010001 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \end{pmatrix} = 100$$

The syndrome is 1000 which represents the 12 th column of the matrix, so it suffices to toggle R12 to arrive at a null syndrome. So the correct message is 100011100100101.

3.6. The Control Matrix of LDPC Code of Dimension n=31 and m=5 ($n=2^m-1$ and m=5)

let H be the control matrix

$$H = \begin{bmatrix} 0000000000000001111111111111111 \\ 0000000111111110000000011111111 \\ 0001111000011110000111100001111 \\ 0110011001100110011001100110011 \\ 1010101010101010101010101010101 \end{bmatrix}$$

Suppose, we received the code word after the transmission channel in the form
r = 1000111001011010000011010010101,
We calculate the Syndrome S Such as S= rH^T, S= S_1 S_2 S_3 S_4 S_5

If at least one of the non- zero Si such as i varied between 1 and 5.
Then the received message is incorrect, i.e. the message sent initially was distorted by the constraints of heterogeneity of the Channel.
I will check the equation parity rH^T

$$\begin{pmatrix} 000011111111000 \\ 011100011110100 \\ 101101100110010 \\ 110110101010001 \end{pmatrix} \begin{pmatrix} R1 \\ R2 \\ R3 \\ R4 \\ R5 \\ R6 \\ R7 \\ R8 \\ R9 \\ R10 \\ R11 \\ R12 \\ R13 \\ R14 \\ R15 \end{pmatrix} = S_1 S_2 S_3 S_4$$

$$(10\,0\,01110\,01\,0\,110\,10\,0\,0\,0\,0\,11\,01\,0\,0\,1\,01\,0\,1) \begin{bmatrix} 00001 \\ 00010 \\ 00011 \\ 00100 \\ 00101 \\ 00110 \\ 00111 \\ 01000 \\ 01001 \\ 01010 \\ 01011 \\ 01100 \\ 01101 \\ 01110 \\ 01111 \\ 10000 \\ 10001 \\ 10010 \\ 10011 \\ 10100 \\ 10101 \\ 10110 \\ 10111 \\ 11000 \\ 11001 \\ 11010 \\ 11011 \\ 11100 \\ 11101 \\ 11110 \\ 11111 \end{bmatrix} = 00001$$

Such as H^T is written as following

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

$$H^T = \begin{bmatrix} 00001\\00010\\00011\\00100\\00101\\00110\\00111\\01000\\01001\\01010\\01011\\01100\\01101\\01110\\01111\\10000\\10001\\10010\\10011\\10100\\10101\\10110\\10111\\11000\\11001\\11010\\11011\\11100\\11101\\11110\\11111 \end{bmatrix}$$

$$(\underline{0}000111001011010000011010010101)\begin{bmatrix} 00001\\00010\\00011\\00100\\00101\\00110\\00111\\01000\\01001\\01010\\01011\\01100\\01101\\01110\\01111\\10000\\10001\\10010\\10011\\10100\\10101\\10110\\10111\\11000\\11001\\11010\\11011\\11100\\11101\\11110\\11111 \end{bmatrix} = 00000$$

And the rH^T matrix product gives 00001.
Therefore the information sent initially is disturbed at the level of transmission channel, and the syndrome calculated by the matrix product rH^T shows the existence of the error.
the vector 00001 found represents the first column of the parity matrix H, which implies that the deformed variable node is R_1 finally, just toggle R_1 to find the original information.
Therefore the information received
r = $\underline{1}$000111001011010000011010010101 becomes r'= $\underline{0}$000111001011010000011010010101. To confirm the truth of this correction, simply multiply $r'H^T = 0$,i.e

4. THE SECOND PROPOSED ALGORITHM

The design of this particular algorithm was formed by the transfer of the equations of parity to a meaningful table, which will directly give the variable nodes which must be modified to find a null syndrome without doing several iterations. This table is made up of three blocks and each of them has a specific function.

Step one: (P1) only consists of parity check equations.

Step two: (P2) includes a significant number of possibility of the syndrome that we can correct with zero iteration. The column [pp.s] which represents a clean syndrome facilitates the search on the nodes of variables which must be modified to find the null syndrome without making several iterations.

Step three: (P_3) in this part, we introduce the notion of syndrome proper of the second part in order to find the rest of the probable syndromes, it suffices to add the syndrome proper of the second part. by using

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

the following expression: $S_i \oplus S_i$ equals zero and $S_i \oplus S_j$ different from zero without doing several iterations.

4.1. The Proposed Table of a LDPC (9,4) Code :

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

We calculate the syndrome S of the received code word r such as $S = Hr^T$

- If $S = Hr^T = 0$, i.e. the syndrome of null then the received code word is correct, so the algorithm has finished, the received message is correct and we can determine the useful message by decoding it to the corresponding message.
- If $S = Hr^T \neq 0$, the non-zero syndrome, therefore the code word received is incorrect.

The proposed algorithm able to determine the variable nodes ri which must be modified to find a null syndrome.

Suppose the received code word 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 $S = Hr^T = s_1s_2s_3s_4$ ($S = s_1s_2s_3s_4$).

$$\begin{pmatrix} 011011000 \\ 101100100 \\ 111010010 \\ 000110001 \end{pmatrix} \begin{pmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \\ r_5 \\ r_6 \\ r_7 \\ r_8 \\ r_9 \end{pmatrix} = s_1s_2s_3s_4$$

Each row of H gives a parity check equation:

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

Table 1: Represents all possible syndromes 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						$s_1s_2s_3s_4$	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	

4.2. Explanatory Example:

suppose the result of Hr^T is 0111, therefore the proper syndrome is $S_2S_3S_4$ of the second concept algorithm, we will look in part two of the table on the compatible syndromes to add them up, to obtain the syndromes of parity three, then alter the suitable nodes of variable, to get the null syndrome.

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

$S_2 S_3 \oplus S_4 = S_2S_3S_4$ we are forced to toggle r_1 (case of S_2S_3) and r_9 (case of S_4) or $S_2 \oplus S_3 \oplus S_4 = S_2S_3S_4$ we are forced to toggle r_7 (case of S_2), r_8 (case of S_3) and r_9(case of S_4) or $S_2S_4 \oplus S_3 = S_2S_3S_4$ we are forced to toggle r_4 (case of S_2S_4) and r_8(case of S_3).

4.3. Simulation of LDPC (9, 4) Code after Correction:

On reception, the decoder receives the different code words after the transmission channel. The hardware description language (VHDL) VHSIC which will calculate various possible cases of syndromes based on the equation $S = Hr^T$

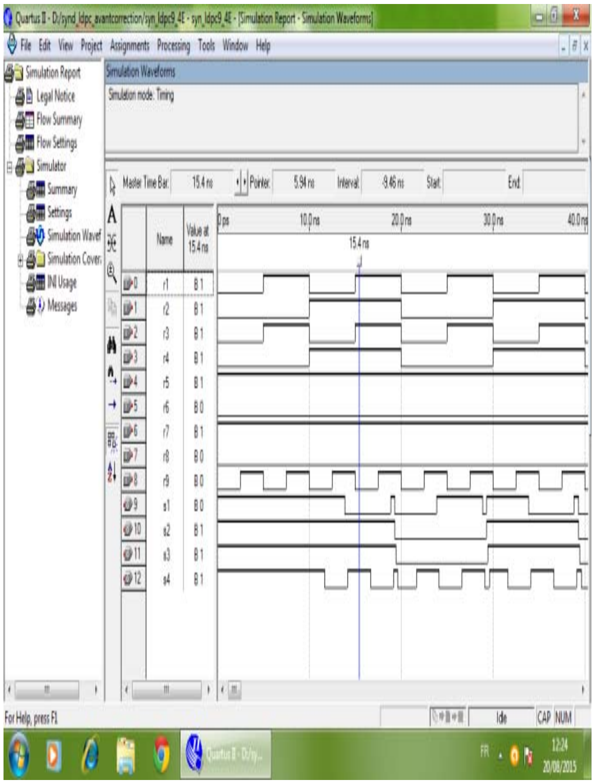


Figure3: different possible values of the syndrome for an LDPC code (9, 4).

4.4. The Simulation Result after the Correction of the Errors of LDPC (9, 4) Code:

if we compare figure 3 which represents the different values of the syndrome before the correction and figure 4 which also represents the values of the syndrome but after the correction, we notice that all the syndrome becomes null, this confirms the effectiveness and the performance of this algorithm which detects corrects all existing errors.

i.e we always find a null syndrome : $S_1S_2S_3S_4$ =0000**.**

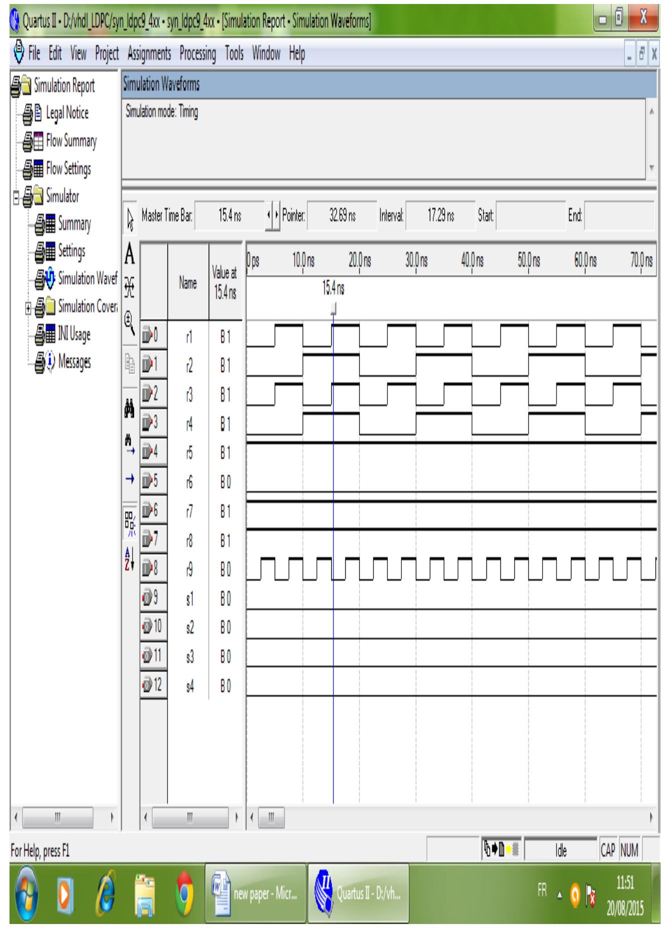


Figure 4: different values of the syndrome are zero for an LDPC code (9, 4).

4.5. Example of LDPC (7, 3) code:

Parity check matrix of the dimension (7, 3)

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

Parity check equation

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

4.5.1. Table represents the different possible cases of the syndrome

Table 2: Show all the probable syndromes of the LDPC (7,3) code

$s_1s_2s_3$	r_1	r_2	r_3	r_4	r_5	r_6	r_7	$[pp.s]$

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

s_1	x			x		x	x	
s_2		x		x	x	x		
s_3			x		x	x	x	
100	x							s_1
010		x						s_2
001			x					s_3
110				x				$s_1 s_2$
011					x			$s_2 s_3$
111						x		$s_1 s_2 s_3$
101							x	$s_1 s_3$

4.5.2. The simulation of this example before correction:

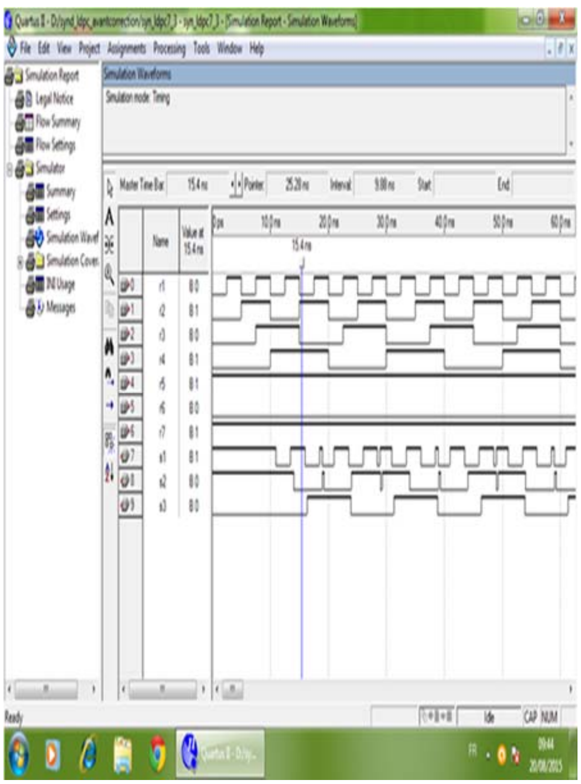


Figure 6: different possible values of the syndrome for an LDPC code (7, 3).

4.5.3. The simulation of this example after correction:

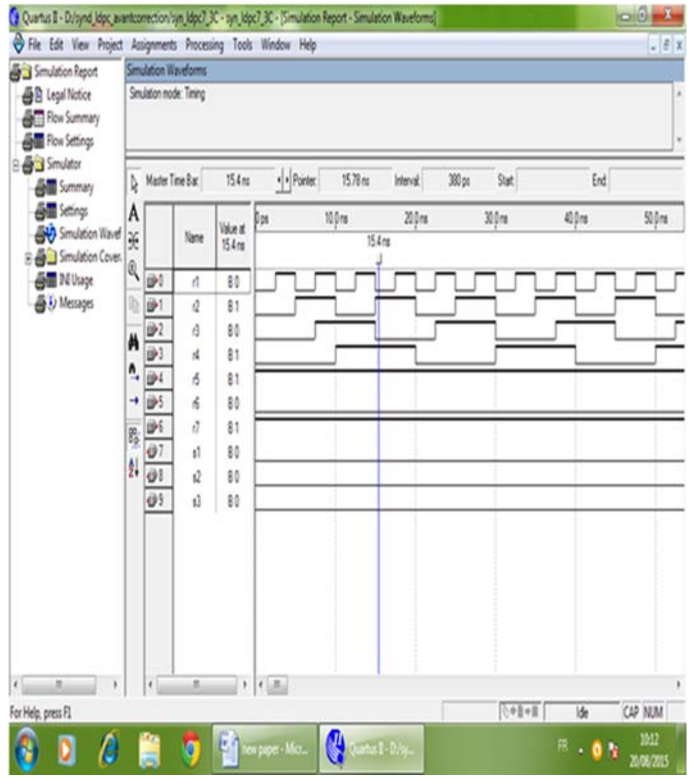


Figure 7: different values of the syndrome are zero for an LDPC code (7, 3).

4.6. Example for LDPC (12,6) Code:

The parity matrix H of an LDPC code (12.6) of dimension
12 × 6

$$H = \begin{bmatrix} 110100001101 \\ 111110010000 \\ 101011000110 \\ 000111101010 \\ 011000110011 \\ 000001111101 \end{bmatrix}$$

The parity matrix H multiplied by r transposed

$$Hr^T = S_1 S_2 S_3 \, S_4 S_5 S_6$$

Journal of Theoretical and Applied Information Technology
15th November 2020. Vol.98. No 21

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

$$\begin{pmatrix} 110100001101 \\ 111110010000 \\ 101011000110 \\ 000111101010 \\ 011000110011 \\ 000001111101 \end{pmatrix} \begin{pmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \\ r_5 \\ r_6 \\ r_7 \\ r_8 \\ r_9 \\ r_{10} \\ r_{11} \\ r_{12} \end{pmatrix} = S_1 S_2 S_3 S_4 S_5 S_6$$

The parity equations resulting from this matrix multiplication

$$S_1 = r_1 \oplus r_2 \oplus r_4 \oplus r_9 \oplus r_{10} \oplus r_{12}$$
$$S_2 = r_1 \oplus r_2 \oplus r_3 \oplus r_4 \oplus r_5 \oplus r_8$$
$$S_3 = r_1 \oplus r_3 \oplus r_5 \oplus r_6 \oplus r_{10} \oplus r_{11}$$
$$S_4 = r_4 \oplus r_5 \oplus r_6 \oplus r_7 \oplus r_9 + r_{11}$$
$$S_5 = r_2 \oplus r_3 \oplus r_7 \oplus r_8 \oplus r_{11} \oplus r_{12}$$
$$S_6 = r_6 \oplus r_7 \oplus r_8 \oplus r_9 \oplus r_{10} \oplus r_{12}$$

Table 3: This table generalizes all possible syndromes for an LDPC code (12,6)

$S_1S_2…S_6$	r_1	r_2	r_3	r_4	r_5	r_6	r_7	r_8	r_9	r_{10}	r_{11}	r_{12}
S_1	X	X		X					X	X		X
S_2	X	X	X	X	X			X				
S_3	X		X		X	X				X	X	
S_4				X	X	X	X		X		X	
S_5		X	X				X	X			X	X
S_6						X	X	X	X	X		X
111000	X											
110010		X										
011010			X									
110100				X								
011100					X							
001101						X						
000111							X					
010011								X				
100101									X			
101001										X		
001110											X	
100011												X
100010	X		X									
001010	X	X										
001100	X			X								
100100	X				X							
110101	X					X						
111111	X						X					
101011	X							X				
011101	X								X			
010001	X									X		
110110	X										X	
011011	X											X
101000		X	X									
000110		X		X								
101110		X			X							
100001		X						X				
010111		X							X			

111100		X									X	
101110			X	X								
010111			X			X						
001001			X					X				
110011			X							X		
010100			X								X	
111001			X									X
100111				X				X				
001111					X			X				
010010					X						X	
011110						X		X				
000011						X					X	
111010								X		X		
110000								X				X
100000			X	X							X	
010000	X	X	X									
001000	X							X				X
000100		X						X	X			
000010									X	X	X	
000001						X			X	X		
000101	X	X			X			X				
001011	X			X			X					
010110	X						X			X		
011000	X			X			X	X				
011001	X								X	X	X	X
011111	X			X				X				
100110		X	X								X	
101010	X				X						X	
111011		X	X					X				
111101		X			X			X				
111110	X	X		X								
101101											X	X
101111		X	X				X					

5. COMPARISON OF ALGORITHMS:

5.1. Comparison of Algorithms between the Basic and the two Algorithms Proposed

the two proposed algorithms generally allow to minimize the number of iterations compared to the basic algorithm, but each design has specific algorithms, the first proposed algorithm allows to detect and correct an error without doing any iterations, on the other hand the second algorithm makes it possible to detect and correct several errors, this process is illustrated in the table.

5.2. Comparison of Algorithms between first Proposed and Basic in a Table:

Table below shows the difference in the number of iterations between the first proposed algorithm and basic algorithm.

Table 4: represents the comparison between two algorithms

code	First Proposed Algorithm	Basic Algorithm
LDPC (7,3)	we can correct 2^3-1 error without iteration	the number of iterations to correct indeterminate errors
LDPC (9,4)	Correction 2^4 -1without iteration.	2^4-1 possible syndromes. 4 possible iteration if a received codeword equals 101010100

LDPC (10,5)	Correction 2^5-1 without iteration.	2^5-1 possible syndrome. the number of iterations to correct indeterminate errors
LDPC (12,6)	Correction 2^6-1 without iteration.	2^6-1 possible syndromes. 4 possible iterations of a single case.
LDPC (n, k)	Correction 2^k-1 Without iteration	2^k-1 possible syndromes. 4 possible iterations of a single case.

5.3. Comparison of Algorithms between the Second Proposed and the Basic in a Table:

Table below shows the difference in the number of iterations between the second proposed algorithm and basic algorithm.

Table 5: represents the comparison between two algorithms

code	Second Proposed Algorithm	Basic Algorithm
LDPC (7,3)	we can correct 2^3-1 error without iteration	the number of iterations to correct indeterminate errors
LDPC (9,4)	Part 2: correction without iteration. Part 3 : it suffices to add two proper syndrome of the part (2).	2^4-1 possible syndromes. 4 possible iteration if a received codeword equals 101010100
LDPC (10,5)	Part 2: correction without iteration. Part 3 : it suffices to add two proper syndrome of the part (2)	2^5-1 possible syndrome. the number of iterations to correct indeterminate errors
LDPC (12,6)	Part 2: correction without iteration. Part 3 : it suffices to add two proper syndrome of the part (2)	2^6-1 possible syndromes. 4 possible iterations of a single case.

6. CONCLUSION AND PERSPECTIVE

6.1. Conclusion

The first algorithm proposed in our paper is authorized to completely suppress the number of iterations, compared to based algorithm, this design added to the control matrix other specialties: detection, localization and correction of errors, but the control matrix in the basic algorithm can only detect the existence of the error, table 4 above shows the difference between the two algorithms.

The principle of the second algorithm is to write the parity equation in a table which will facilitate the detection of errors without doing several iterations; table 5 above shows the difference between the two algorithms.

The number of possible syndromes for the LDPC (12, 6) code is $2^6 - 1$ and for each syndrome different from zero, we can do four iterations by means for the syndrome to become zero in order to detect and correct the error, which implies a total number of iterations of $4 \times (2^6 - 1)$ for the basic method. On the other hand the second proposed algorithm, it removes the iteration for twelve syndromes and reduces iterations for the rest of the cases from 4 to 2 iterations.

but the first algorithm completely removes the iterations, it suffices to find the syndrome and the projection on the column which represents it then toggle the node of variable linked to the column which represents this syndrome so that the syndrome becomes zero.

6.2. Perspective:

To meet the needs of the current era, of a large number of applications for coding and decoding binary information during communication, hardware implementation solutions on reconfigurable platforms of the FPGA type are increasingly no longer used. FPGA cards present many perspectives for the implementation of algorithms in the current era. In addition, the computer-aided design tools are used to go directly from a functional description (like VHDL) to a logic gate diagram ready to implement on FPGA.

In the future, I will develop, validate, improve the performance, the reliability of the circuit and implant on the different programmable cards and especially the target FPGA card the proposed decoding algorithms.

REFRENCES:

[1] C. Shannon, A Mathematical Theory of Communication, Bell Syst. Tech. Journal, vol.27, pp. 623-656, 1948.

[2] R.G. Gallager, Low-Density Parity-Check Codes, IRE Transaction on Information Theory, vol. IT-8, pp. 21-28, January 1962

[3] D.J. MacKay and R.M. Neal, Good Codes Based on Very Sparse Matrices, in Cryptography and Coding 5th IMA Conference, vol. 1025, pp. 100-1 11, 1995.

[4] A.H .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.

[5] B. Ahn, S. Yoon, and Jun Heo, "Low complexity syndrome-based decoding algorithm applied to block turbo codes", Access IEEE, Vol. 6, PP.26693- 26706, 2018

[6] T. T. Nguyen-Ly, V. Savin, K. Le, D. Declercq, F. Ghaffari, and O. Boncalo, "Analysis and design of cost-effective, high-throughput LDPC decoders," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. PP, no. 99, pp. 1–14, Dec 2017.

[7] Soufian Addi ,Ahlam Berkani ,Ahmed Azouaoui, Mostafa Belkasmi, New hard decision decoder of LDPC codes using single Bit Flipping algorithm, DOI: 10.1109/WINCOM.2017.8238191,IEEE Xplore: 25 December 2017.

[8] T. M. N. Ngatched, M. Bossert, A. Fahrner, and F. Takawira, "Two bit flipping decoding algorithms for low-density parity-check codes," IEEE Trans. Commun., vol. 57, no. 3, pp. 591–596, Mar. 2009

[9] X. Wu, C. Ling, M. Jiang, E. Xu, C. Zhao, and X. You, "New insights into weighted bit-flipping decoding," IEEE Trans. Commun., vol. 57,no. 8, pp. 2177–2180, Aug. 2009.

[10] T. Wadayama, K. Nakamura, M. Yagita, Y. Funahashi, S. Usami, and I. Takumi, "Gradient descent bit flipping algorithms for decoding LDPC codes," IEEE Trans. Commun., vol. 58, no. 6, pp. 1610–1614, Jun. 2010.

[11] S. Elsanadily, A. Mahran and O. Elghandour, Two-Side State-Aided BitFlipping Decoding of Generalized Low Density Parity Check Codes, IEEE Commun. Lett.. vol. 21, no. 10, pp. 2122-2125, Oct. 2017

[12] Yuan-hua Liu, Xin-liangNiu and Mei-ling Zhang,"Multi-Threshold Bit Flipping Algorithm for Decoding Structured LDPC Codes," IEEE Commun Lett., vol. 19, no. 2, pp. 127-130, Feb. 2015

[13] Pierre Renaldo, Guilaume Vaquero, A new generation 5G Technology , uses to invent , WAVESTONE , 2019.