Performance study and synthesis of new Error Correcting Codes RS, BCH and LDPC Using the Bit Error Rate (BER) and Field-Programmable Gate Array (FPGA)

El habti El idrissi Anas†, El gouri Rachid†, ††, Ahmed Lichioui††† and Hlou Laamari†

† Laboratory of Electrical Engineering and Energy System. Faculty of Sciences, University Ibn Tofail Kenitra, Morocco
†† Laboratory of Electrical Engineering and Telecommunications Systems, National Schools of Applied Sciences, University Ibn Tofaïl Kenitra, Morocco
††† National Society of Radio and Television (SNRT), Rabat, Morocco

Abstract
Most communication networks involve physical phenomena of distortion and attenuation of the transmitted information and external disturbances associated with transport conditions or communication equipment. Therefore, this information is often corrupted in reception. The RS codes (Reed-Solomon), LDPC codes (Low-Density Parity-Check) and BCH codes (Bose, Ray-Chaudhuri and Hocquenghem) are being widely used in variety communication and storage systems. In this paper we present a synthesis of the new algorithms for RS, BCH and LDPC codes that have been recently proposed in this field. we also give a performance study of these algorithms, the number of the minimized logic gates and iterations will be detailed by using The bit error rate (BER) and the FPGA card Xilinx Spartan 3E-500 FG 320 FPGA (xc3s500e-5fg320).

Key words:
Error correcting codes, FPGA implementation, Hardware Description Language (VHDL), FPGA Device Utilization Summary, Bit Error Rate, Minimization Rate.

1. Introduction

The error correcting codes are widely used in telecommunications, where they permit a reliable transmission in the noisy communication mediums, such as wireless channels. Few of them are RS, BCH and LDPC codes. These codes differ from each other in their implementation and complexity [1]. The Reed-Solomon codes are considered a subgroup of cyclic codes of the BCH codes family, the binary BCH and Reed-Solomon codes have found applications in the conservation of the digital information, and in the communication systems like wireless systems, networking communications and data storage systems, we find for example the code RS (255, 233, 33) which is used by the NASA in space communication [2]. The Low density parity-check codes (LDPC) are linear codes; LDPC was developed by Robert Gallager in 1962 and rediscovered by Mackay in 1996, in 2003, an LDPC code has been preferred to six Turbo Codes to become error correction code in digital video broadcasting-satellite-second generation (DVB-S2) [3]. The objective of this work is to present both a performance study and a synthesis of the basic and proposed algorithms for RS, BCH and LDPC codes bearing in mind many works have been done in this field, we also show that with a bit error rate of 10-4 for the basic circuit, when the minimization rate increases, the bit error rate decreases. The rest of the paper is organized as follows: an overview of error correcting codes RS, BCH and LDPC is presented in section 2. The proposed algorithms, simulation, FPGA implementation details, performance study of these codes are presented in section 3, followed by a conclusion.

Manuscript received May 5, 2016
Manuscript revised May 20, 2016

2. Background and related work

2.1. Reed-Solomon decoders

The main purpose of decoders is the detection and correction of errors introduced by transmission channels, knowing that, errors can be added to the coded message C (x) [4], the received code word is given by:

$$R(x) = C(x) + E(x) \quad (1)$$

Where:

$$E(x) = E_{n-1}X^{n-1} + E_{n-2}X^{n-2} + \cdots + E_1X^1 + E_0 \quad (2)$$

$R(x)$: The received polynomial

$C(x)$: The coded polynomial

$E(x)$: The error polynomial

2.1.1. Main units of a Reed-Solomon decoder

Several steps are necessary of a Reed-Solomon decoder:

- Calculate the syndrome
- Use the Euclidean algorithm to calculate the error locator polynomial and amplitude polynomial
- Calculate the error values: Forney method
- Chien search for error position

2.1.2. Performance of Reed-Solomon codes RS (15, 11) and RS (255, 239) using the bit error rate term (BER)

To analyze the performance using the bit error rate term (BER) as shown in figure 1, two codes are presented: RS (15, 11) and RS (255, 239. The channel and the modulation type are respectively AWGN and PSK where the order of modulation is equal to 2.

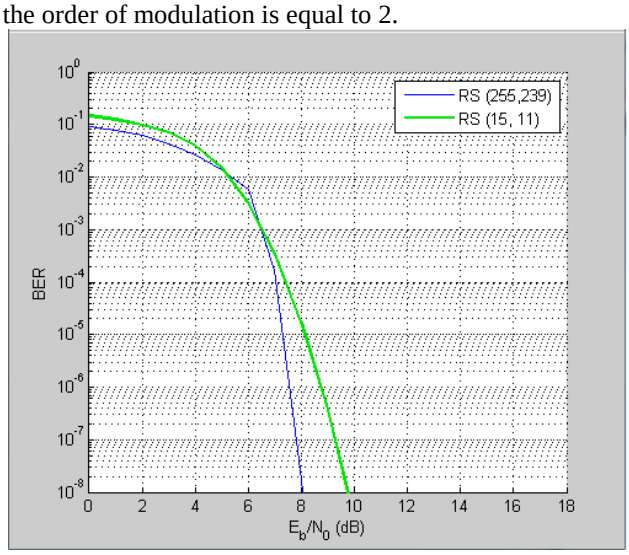

Fig.1. performance of RS (15, 11) and RS (255, 239) codes using the bit error rate term (BER)

The figure 1 shows the performance of two codes: RS (15, 11) and RS (255, 239). One of length 15, and the other of length 255. It is observed that the coding gain is 0, 6 dB for a BER = 10-4 when the length of the code word changes from 15 to 255. However, when the code word length increases, the complexity of calculating and implementation increases too.

2.2. BCH decoder

2.2.1. Main units of a BCH decoder

Three main steps for BCH decoding are represented as follows:

- Step 1: Computation of syndromes.
- Step 2: Berlekamp-Massey algorithm.
- Step 3: detection of error position using Chien Search Block.

2.2.2. Performance of BCH codes BCH (15, 7, 2) and BCH (255, 231, 3) using the bit error rate term (BER)

To analyze the performance using the bit error rate term (BER) as shown in figure 2, two codes are presented; BCH (15, 7, 2) and BCH (255, 231, 3). The channel and the modulation type are respectively AWGN and PSK where the order of modulation is equal to 2.

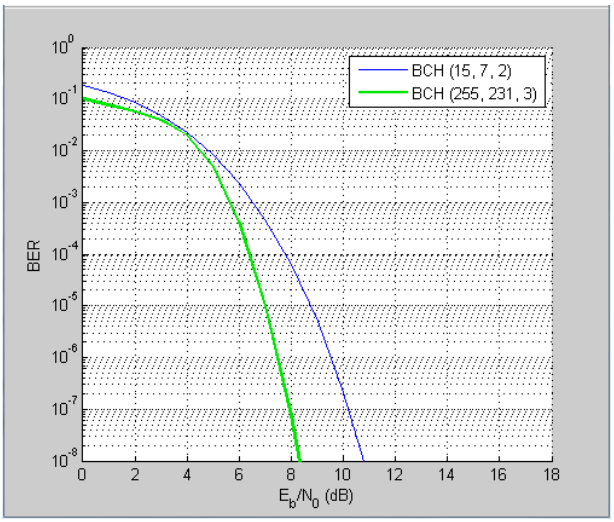

Fig.2. performance of BCH (15, 7, 2) and BCH (255, 231, 3) codes using the bit error rate term (BER)

The figure 2 shows the performance of two codes: RS (15, 7, 2) and RS (255, 231, 3). One of length 15, and the other of length 255, it is observed that the coding gain is 1, 4 dB for a BER = 10-4 when the length of the code word changes from 15 to 255. However, when the code word length increases, the complexity of calculating and implementation increases too.

2.3. LDPC decoder

Low density parity-check code (LDPC) is an error correcting code used in noisy communication channel to correct the errors introduced by the transmission channel. There are many methods to decode the LDPC codes, among these methods we find: the sum-product algorithm and the Bit-Flipping algorithm (BFA). 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 main aim of Bit-Flipping Algorithm is to correct the bits of the received code word by observing the output of the set of parity equations. First, we calculate the received code word syndrome, if it is zero; it is assumed that the received code word is correct. If the syndrome differs from zero, the received code word contains errors. The algorithm consists to flip the received bits causing a greater number of parity errors. The syndrome is again calculated and the received bits are flipped again until the syndrome equal to zero [5].
The algorithm is shown as below:

Step 1: We use the equation $s = r. H^t$ 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 $\{f_0, f_1, \cdots, f_{N-1}\}$ 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 $S = r H^t$ 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.

3. FPGA implementation and performance of proposed algorithms

3.1. Proposed algorithm of Reed-Solomon decoder

The proposed algorithm is based on a simple factorization of the error locator polynomial where (xn, where $n \geq 2$) should not be appeared in this polynomial. Besides, we can conceive another circuit of the Chien Search Block i.e. we can minimize a large number of the used logic gates, which allows reducing the power consumption compared to the basic circuit [6]. For the case of RS (15, 11), the error locator polynomial is given by:

$$\Lambda(x) = 14x^2 + 14x + 1 \quad (3)$$

It's a polynomial of degree 2 as type:

The basic circuit is represented in figure 3:

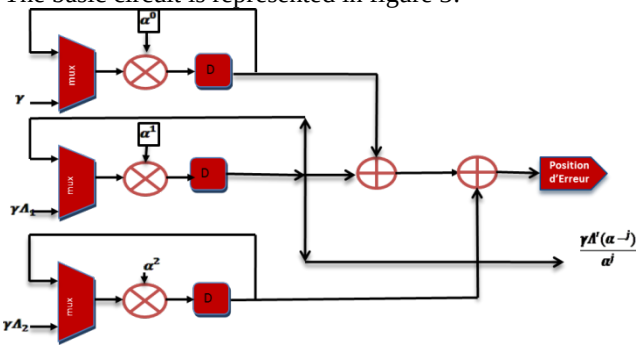

Fig.3. Basic circuit of Chien Search Block for RS (15, 11) codes

The simulation of the Chien Search Block using Quartus is represented in figure 4:

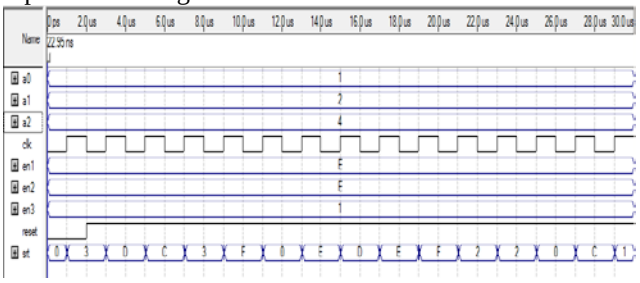

Fig.4. Simulation of the basic circuit of the Chien Search Block using Quartus for RS (15, 11) code

For the case of an error locator polynomial as degree 8, we have:

$$\Lambda(x) = Ax^8 + Bx^7 + Cx^6 + Dx^5 + Ex^4 + Fx^3 + Gx^2 + Hx + I \quad (4)$$

The basic circuit is given by:

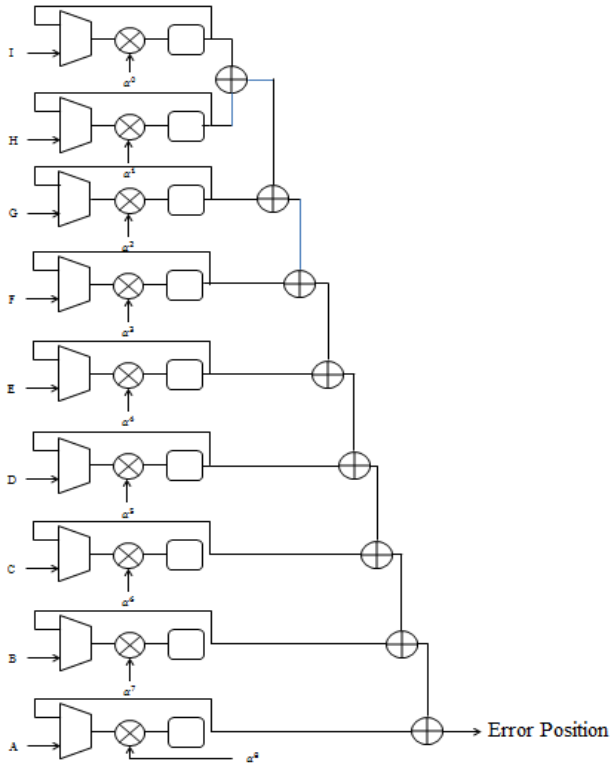

Fig.5. Basic circuit of Chien Search Block for an error locator polynomial as degree 8

The simulation is represented below in figure 6:

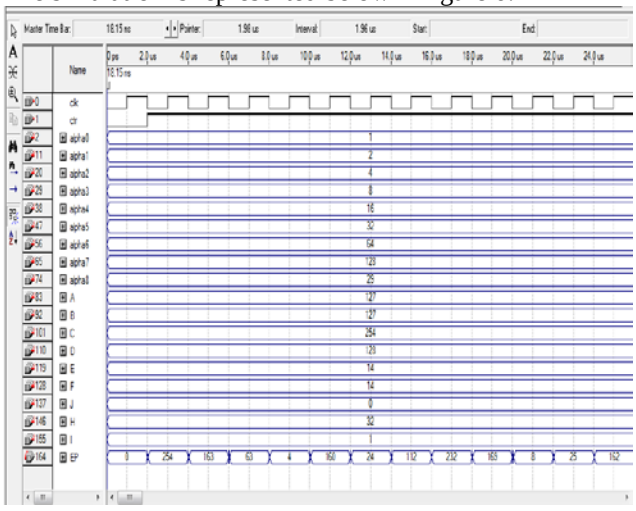

Fig.6. Simulation of the basic circuit of the Chien Search Block using Quartus for an error locator polynomial as degree 8

The modified circuit of Chien Search Block for Reed-Solomon codes RS (15, 11) is represented in figure 7:

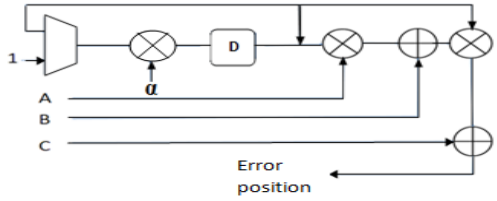

Fig.7. Modified circuit of Chien Search Block for RS (15, 11) code

The simulation is represented in figure 8:

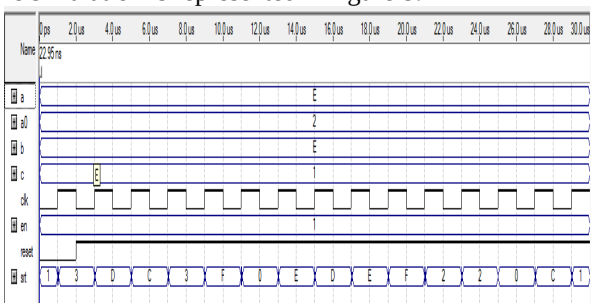

Fig.8. Simulation of the modified circuit of the Chien Search Block using Quartus for RS (15, 11) code

For the case of an error locator polynomial as degree 8, the modified circuit is given by:

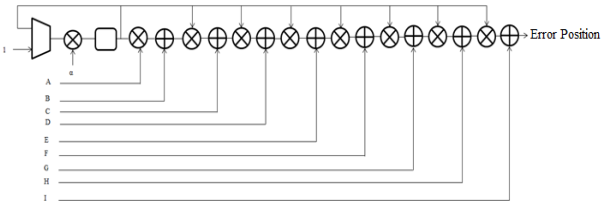

Fig.9. Modified circuit of Chien Search Block for an error locator polynomial as degree 8

The simulation is represented in figure 10:

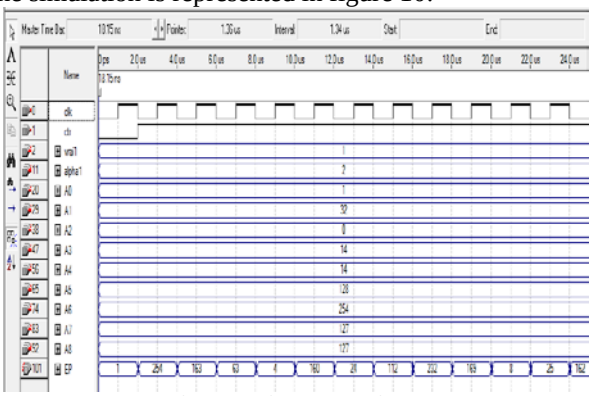

Fig.10. Simulation of the modified circuit of the Chien Search Block using Quartus for an error locator polynomial as degree 8

3.1.1. FPGA implementation

Implementation of decoders for Reed-Solomon (RS), Bose, Ray-Chaudhuri et Hocquenghem (BCH) and Low-Density Parity-Check (LDPC) codes has been problematic due to very large amount of resources required We seek to implement a new Chien Search Block on FPGA based on the proposed algorithm to judge the savings in hardware resources. In this paper a parameterized hardware model of the decoders Block (RS, BCH and LDPC codes) was developed by using the Hardware Description Language (VHDL) and synthesized by using Xilinx Synthesis Tool. The block diagram of the proposed algorithm as implemented is shown in figure 11.

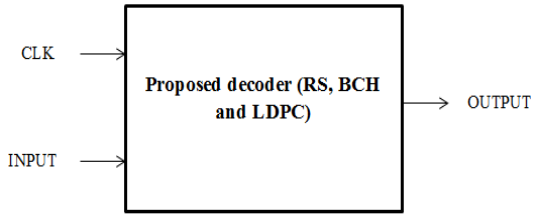

Fig.11. Block diagram of a decoder block

3.1.2. Experimental results and performance

Both the proposed and the basic algorithms have been implemented on FPGA card Xilinx Spartan 3E-500 FG 320 FPGA (xc3s500e-5fg320) for RS (15, 11) and RS (255, 239) codes i.e. for the error locator polynomial as degree respectively 2 and 8.

- In the case of an error locator polynomial as degree 2:

The basic circuit contains 34 Slice Flip Flops, while the modified one contains 30, which give a minimization rate as:

$$\frac{(34-34)}{34} \times 100 = 11\%$$

The minimization rate of LUTs 4 inputs is:

$$\frac{(42-37)}{42} \times 100 = 12\%$$

The minimization of occupied slices is:

$$\frac{(25-19)}{25} \times 100 = 24\%$$

- In the case of an error locator polynomial as degree 8:

The basic circuit contains 95 Slice registers, while the modified one contains only 39, which increases the minimization rate with a percentage:

$$\frac{(95-39)}{95} \times 100 = 59\%$$

According to the tables 1 and 2 below which show, the FPGA device utilization summary for RS (15, 11) and RS (255, 139) codes using the two algorithms, it can be concluded that the proposed algorithm presents a low complexity and a very good performance compared to the basic algorithm.

Table 1: FPGA device utilization summary for RS (15, 11) codes

Recources	*Proposed Algorithm*	*Basic Algorithm*
Device	Xilinx Spartan 3E (xc3s500e-5fg320)	
RS (n, k)	RS (15, 11)	
Number of occupied Slices	19	25
Number used as Flip Flops	30	34
Total number of 4 input LUTs	37	42
Number used as logic	12	17

Table 2: FPGA device utilization summary for RS (255, 139) codes

Recources	*proposed algorithm*	*basic algorithm*
Device	Xilinx Spartan 3E (xc3s500e-5fg320)	
RS (n, k)	RS (255, 239)	
Number of occupied Slices	271	119
Total Number Slice Registers	39	95
Number of 4 input LUTs	504	196
Number used as logic	504	196

3.1.3. Comparison of circuits

According to the simulation we have adopted previously, the result we got is the same in comparison similar to the modified circuit but with an important number of minimized logic gates. This minimization can reduce the energy consumption with a percentage which reaching 50% compared to the basic circuit. The table 3 shows the number of the used logic gates by using both the basic circuit and the modified circuit for different error locator polynomials.

Table 3: Number of minimized logic gates for different error locator polynomials

Error locator polynomial	*Number of logic gates for the basic circuit (N_b)*	*Number of logic gates for the modified circuit*	*Number of minimized logic gates (N_m)*	*Minimization rate ($\frac{N_m}{N_b} \times 100$) %*
$\Lambda(X) = AX^2+BX+C$	11	7	4	36,36
$\Lambda(X) = AX^3+BX^2+ CX+D$	15	9	6	40
$\Lambda(X) = AX^4+BX^3+ CX^2+DX+E$	19	11	8	42,10
$\Lambda(X) = A_nX^n + \ldots+A_1X+A_0$	3+4n	3+2n	2n	$(\frac{2n}{3+4n} \times 100)$

3.1.4. Graphic representation

We can plot the results of the factorization method used in the Chien Search block using the degree of the error locator polynomial. The Figures 12 and 13 represent respectively, the minimization rate depending on the degree of the error locator polynomial and the evolution of the bit error rate BER depending on the minimization rate for a BER = 10-4 of the base circuit.

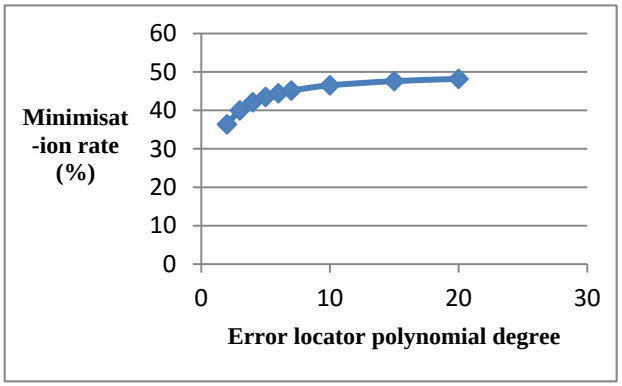

Fig.12. Evolution of the minimization rate depending on the degree of the error locator polynomial

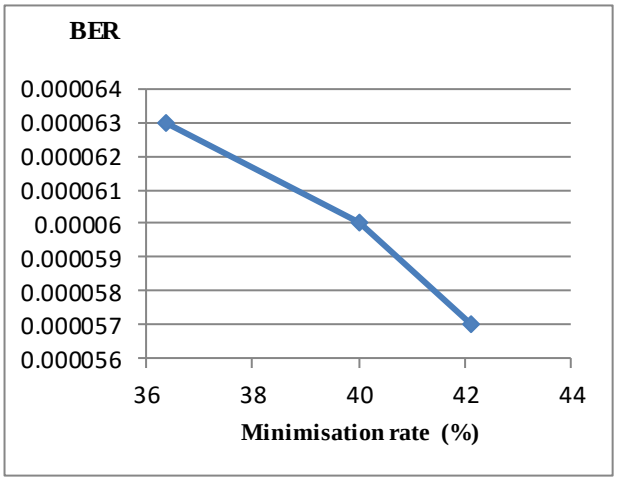

Fig.13: Evolution of the bit error rate BER depending on the minimization rate for the modified circuit with a BER = 10-4

According to the Figure 13, which represents the evolution of the bit error rate BER depending on the minimization rate, we can note that for a bit error rate BER = 10^{-4} of the base circuit, when the minimization rate increases the bit error rate decreases, for a minimization rate equal to 36% BER = 63.10^{-6}, if MR = 40% BER = 60.10^{-6}, for MR = 42% BER = 57.10^{-6}.

3.2. Proposed algorithm of BCH decoder

The binary BCH codes were discovered in 1959 by Hocquenghem and independently in 1960 by Bose and Ray-Chaudhuri. Later, Gorenstein and Zierler generalized them to all finite fields [7]. The purpose of this work is to improve the BCH decoder compared to the basic algorithm. First, we conceive another circuit of the syndrome block [8]. Second, we optimize the Berlekamp's algorithm using the error locator polynomial. Third we improve the Chien search block in order to reduce a large number of logic gates Bearing in mind that many works have been done in this field [9] [10]. Fourth, the proposed algorithm was implemented on a Xilinx Spartan 3E-500 FG 320 FPGA (xc3s500e-5fg320). Finally, we study the evolution of the bit error rate BER depending on the minimization rate for a BER = 10-4 of the base circuit.

3.2.1. Proposed BCH decoder

The proposed BCH decoder contains three main steps represented as follows [11]:

- Step 1: proposed syndrome block.
- Step 2: Calculation of error locator polynomial through the Berlekamp-Massey algorithm.
- Step 3: proposed Chien Search Block.

The hardware algorithm and simulation for proposed BCH decoder are represented respectively in figures 11 and 12.

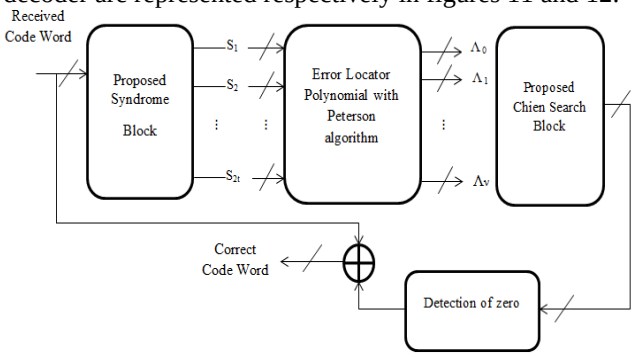

Fig.14: Modified BCH decoder

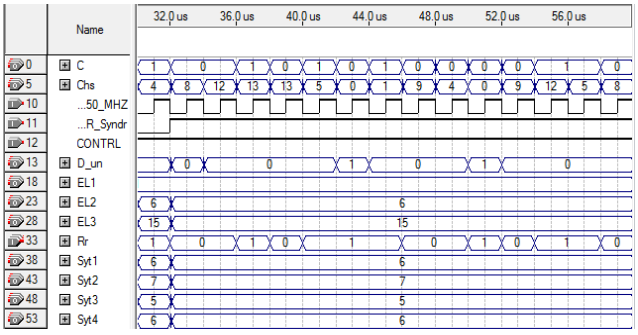

Fig.15: VHDL Simulation of Modified BCH decoder with Quartus

3.2.2. FPGA implementation and performance

Both the proposed and the basic algorithms have been implemented on FPGA card Xilinx Spartan 3E-500 FG 320 FPGA (xc3s500e-5fg320) for BCH (15, 7, 2) and BCH (31, 16, 3) codes i.e. for a different correction capability and number of information symbol.

- For the case of BCH (15, 7, 2) code :

The basic circuit contains 73 Slice Flip Flops, while the modified one contains 50, which give a minimization rate as:

$$\frac{(73-50)}{73} \times 100 = 31,5\%$$

The minimization rate of LUTs 4 inputs is:

$$\frac{(94-90)}{94} \times 100 = 4,25\%$$

The minimization rate of occupied slices is:

$$\frac{(68-47)}{68} \times 100 = 30\%$$

- In the case of BCH (31, 16, 3) code :

The basic circuit contains 606 Slice Flip Flops, while the modified one contains 399, which give a minimization rate as:

$$\frac{(606-399)}{606} \times 100 = 34,1\%$$

The minimization rate of LUTs 4 inputs is:

$$\frac{(1040-714)}{1040} \times 100 = 31,34\%$$

According to the Tables 4 and 5 and the figure 13 which represent respectively "the number of used logic gates" for BCH (15, 7, 2) and BCH (31, 16, 3) using the two algorithms and the evolution of the bit error rate BER depending on the minimization rate for a BER = 10^{-4} of the basic circuit. We can conclude that the proposed algorithm presents a reduced complexity and a very good performance in comparison to the basic algorithm.

Table 4: FPGA device utilization summary for BCH (15, 7, 2) codes

Recources	***proposed algorithm***	***basic algorithm***
Device	Xilinx Spartan 3E (xc3s500e-5fg320)	
BCH (n, k, t)	BCH (15, 7, 2)	
Number of occupied Slices	47	68
Total Number Slice Registers	50	73
Number of 4 input LUTs	90	94

Table 5: FPGA device utilization summary for BCH (31, 16, 3) codes

Recources	**proposed algorithm**	**basic algorithm**
Device	Xilinx Spartan 3E (xc3s500e-5fg320)	
BCH (n, k, t)	BCH (31, 16, 3)	
Total Number Slice Registers	399	606
Number of 4 input LUTs	714	1040

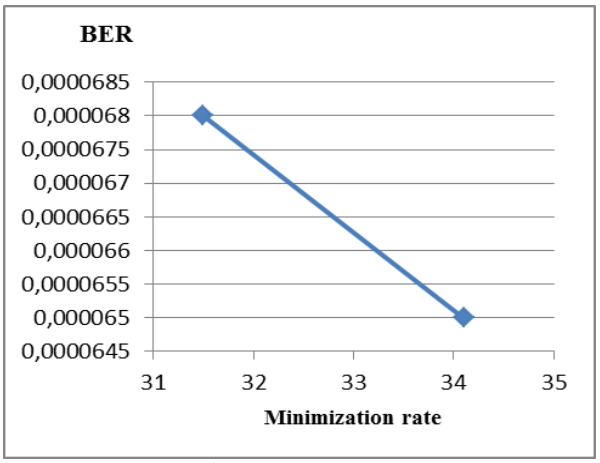

Fig.16: Evolution of the bit error rate BER depending on the minimization rate for the modified circuit with a BER = 10-4

According to the Figure 13, which represents the evolution of the bit error rate BER depending on the minimization rate, we can note that for a bit error rate BER = 10^{-4} of the base circuit, when the minimization rate increases the bit error rate decreases, for a minimization rate equal to 31, 5% BER = 68.10^{-6}, if TM = 34, 1 % BER = 65.10^{-6}.

3.3. Proposed algorithm of LDPC decoder

In this work, a decoding algorithm for LDPC codes is proposed in order to reduce the implementation complexity; this algorithm is based on solving a matrix equation so as to calculate all possible solutions for this equation [12]. First we conceive the design of the proposed circuit. Then, we simulate the algorithm using VHDL hardware description language and Quartus. Finally, we implement it on the FPGA board.

3.3.1. Proposed LDPC decoder

The proposed algorithm:

- Step 1: We use the equation $S = r \cdot H^t = 0$ to calculate all possible solutions.
- Step 2: Calculate the errors produced during the transmission channel using a simple logic gate XOR.
- Step 3: Calculate the error using the equation $S = e. H^t = 0$.
- Step 4: Calculate the correct code word using the logic gate XOR.

The hardware algorithm and simulation for proposed BCH decoder are represented respectively in figures 14 and 15.

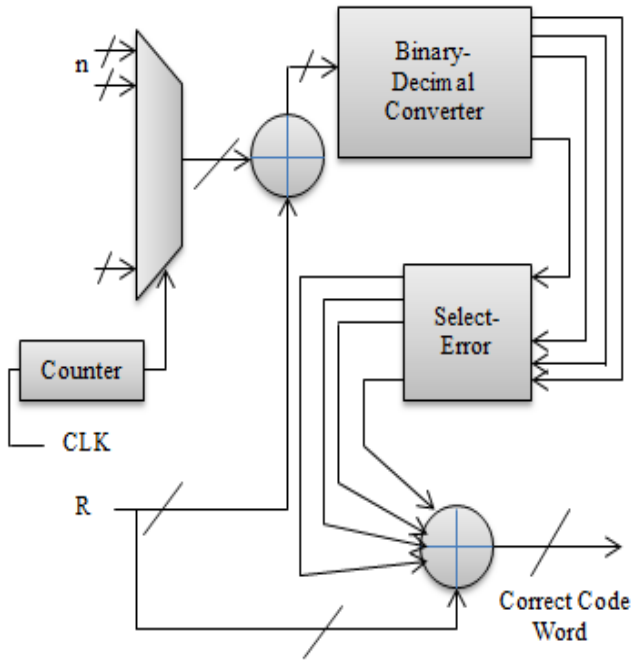

Fig.17: Scheme of LDPC Decoder.

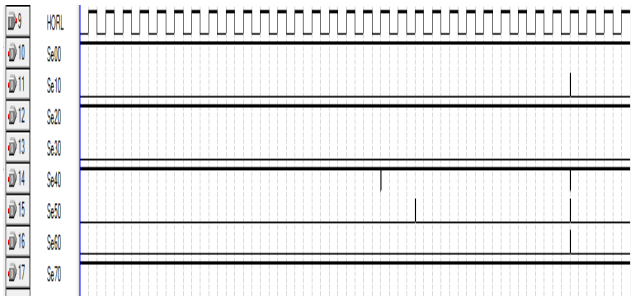

Fig.18: VHDL Simulation of ½ Rates (2, 4) 8-Bit LDPC Code with Quartus.

3.3.2. FPGA implementation

The basic circuit contains 33 Slice Flip Flops, while the modified one contains 14, which give a minimization rate as:

$$\frac{(33-14)}{33} \times 100 = 57\%$$

According to the table 1 which represents FPGA Device Utilization Summary using the two algorithms, we can conclude two things: On the one hand, the proposed algorithm presents a low complexity and a very good performance compared to the bit-flipping algorithm .On the other hand, this algorithm does not use the iterations method which allow the detection of errors introduced by the transmission channel easily and quickly.

Table 6: FPGA Device Utilization Summary for LDPC (8, 4) code

Recources	***Proposed Algorithm***	***Basic Algorithm***
Device	Xilinx Spartan 3E (xc3s500e-5fg320)	
LDPC (n, k)	LDPC (8, 4)	
Number of occupied Slices	50	21
Number used as Flip Flops	14	33
Number of 4 input LUTs	100	40

4. Conclusion

This paper is twofold : The former is the optimization and conception of decoding algorithms for RS, BCH and LDPC codes in order to reduce respectively the complexity and the number of iterations starting by the RS codes (Reed-Solomon) and LDPC (Low-Density Parity-Check) and ending by the BCH (Bose, Ray-Chaudhuri and Hocquenghem). The latter is the implementation of these algorithms on Xilinx Spartan 3E-500 FG 320 FPGA (xc3s500e-5fg320) using the hardware description language (VHDL). The results obtained in this paper show that the proposed algorithms present a low complexity and a very good performance compared to the basic algorithms.

Acknowledgment

This work was supported by the Laboratory of Electrical Engineering and Energy System, Faculty of Science, University Ibn Tofail Kenitra, the National Society of Radio and Television (SNRT), Rabat and the National School of Applied Sciences, University Ibn Tofail Kenitra, Morocco.

References

[1] P. Mathew, Lismi Augustine, Sabarinath G and Tomson. Devis, "HARDWARE IMPLEMENTATION OF (63, 51) BCH ENCODER AND DECODER FOR WBAN USING LFSR AND BMA", International Journal on Information Theory, Vol. 3, No. 3, 2014, pp. 1–1.

[2] P. Djiknavorian, Codes de Reed-Solomon Etude et simulation, avril 2007.

[3] B. Shin, H. Park, S. Hong, J. Seon and SH. Kim "Quasi-cyclic LDPC codes using overlapping matrices and their layered decoders", International Journal of Electronics and Communications VOL.68(5), (2014),pp: 379–383.

[4] BBC Research and Development, Reed-Solomon error correction, British Broadcasting Corporation, 2002.

[5] Jean-yves chouinard, théorie et pratique des codes correcteurs, janvier 2013.

[6] A. El habti, R. El gouri and H. Laamari, "A low power error detection in the Chien Search Block for Reed-Solomon code", Proceedings of International Conference on Complex Systems (ICCS), Agadir Morocco, November 5-6, 2012, pp. 1-3.

[7] M. Prashanthi, P. Samundiswary, "An Area Efficient (31, 16) BCH Decoder for Three Errors", International Journal of Engineering Trends and Technology, Vol. 10, No. 13, 2014, pp. 616–620.

[8] A. El habti, R. Elgouri and L. Hlou, Conception of a new Syndrome Block for BCH codes with hardware Implementation on FPGA Card, Int. Journal of Engineering Research and Applications, Vol. 5, Issue 5, May 2015, pp. 80-85.

[9] A. El habti, R. Elgouri and L. Hlou, FPGA Implementation of A New Chien Search Block for Reed-Solomon Codes RS (255, 239) Used In Digital Video Broadcasting DVB-T, Int. Journal of Engineering Research and Applications, Vol. 4, Issue 8(Version 1), August 2014, pp. 82-86.

[10] R. Huynh, GN. Ning and Yang Huazhung, "A Low Power Error detection in the syndrome calculator Block for Reed-Solomon codes RS: (208,188)", J. Tsinghua Science and Technology, Vol. 14, No. 4, 2009, pp. 474-477.

[11] A. El habti, R. Elgouri, A. Lichioui and L. Hlou, FPGA IMPLEMENTATION OF A NEW BCH DECODER USED IN DIGITAL VIDEO BROADCASTING - SATELLITE - SECOND GENERATION (DVB-S2), Journal of Theoretical and Applied Information Technology, Vol. 3(1), September 2015, pp. 22-28.

[12] A. El habti, R. Elgouri and L. Hlou, Conception of a new LDPC decoder with hardware implementation on FPGA card, International Journal of Engineering & Technology, Vol. 3(4), 2014, pp. 451-456.