

## FPGA Implementation of A New Chien Search Block for Reed-Solomon Codes RS (255, 239) Used In Digital Video Broadcasting DVB-T

El Habti El Idrissi Anas<sup>1</sup>, El Gouri Rachid<sup>1, 2</sup>, Hlou Laamari<sup>1</sup>

<sup>1</sup>Laboratory of Electrical Engineering and Energy System. Faculty of Sciences, University Ibn Tofail Kenitra, Morocco

<sup>2</sup>National School of Applied Sciences, University Ibn Tofail Kenitra, Morocco

### ABSTRACT

The Reed-Solomon codes RS are widely used in communication systems, in particular forming part of the specification for the ETSI digital terrestrial television standard. In this paper a simple algorithm for error detection in the Chien Search block is proposed. This algorithm is based on a simple factorization of the error locator polynomial, which allows reducing the number of components required to implement the proposed algorithm on FPGA board. Consequently, it reduces the power consumption with a percentage which can reach 50 % compared to the basic RS decoder. First, we developed the design of Chien Search Block Second, we generated and simulated the hardware description language source code using Quartus software tools, finally we implemented the proposed algorithm of Chien search block for Reed-Solomon codes RS (255, 239) on FPGA board to show both the reduced hardware resources and low complexity compared to the basic algorithm.

**Keywords**-Digital video broadcasting, Euclidean Algorithm, Error locator polynomial, FPGA implementation, Reed-Solomon codes.

### I. INTRODUCTION

Communication technologies are widely used these days. Due to the growing percentage of people using these technologies, methods are needed to increase the transmission rate without reducing the quality [1]. One of these methods is the Reed-Solomon (RS) codes which are used to correct errors in many systems such as storage devices (CD, DVD, etc.). And digital video broadcasting (DVB), [2]. An RS code word is a packet of symbols which are commonly bytes (8-bit symbols), denotes as RS (n, k) where n is the number of the symbols in the encoded message and k is the number of the symbols in the original message as shown in Fig.1. Each RS code word can correct a maximum of  $t = (n-k)/2$  errors in a received packet [3]. The objective of this work is to prove that it's possible to reduce a large number of components in the Chien Search Block using a new algorithm based on a factorization method which allows conceiving another circuit of Chien Search block with an important number of minimized logic gates.

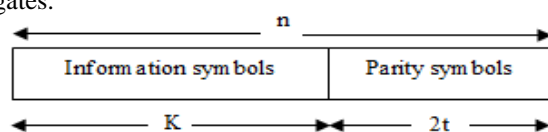


Figure.1. RS code word structure [4]

### II. ERROR LOCATOR POLYNOMIAL

The error locator polynomial  $E(x)$  is written to include only the terms that correspond to errors:

$$E(x) = Y_1 X^{e1} + Y_2 X^{e2} + \dots + Y_v X^{ev} \quad (1)$$

Where  $e1, \dots, ev$ , identify the locations of the errors in the code word as the corresponding powers of  $x$ , while  $Y1, \dots, Yv$  represent the error values at those locations [5].

The error locator polynomial,  $\Lambda(x)$ , has a degree of  $v \leq t$  and can be represented as:

$$\Lambda(x) = \prod_i^v (1 + X_i x) \quad (2)$$

$$= X_1(x + X_1^{-1}) X_2(x + X_2^{-1}) \dots \quad (3)$$

Where  $X_1 = \alpha^{e1}, X_2 = \alpha^{e2} \dots$  then clearly the function value will be zero if  $x = \alpha^{-e1}, x = \alpha^{-e2} \dots$

### III. CHIEN SEARCH ALGORITHM

This algorithm can detect the error position by calculating  $\Lambda(\alpha^i)$  where  $0 \leq i \leq n-1$ , such as  $\Lambda(x)$  is the error locator polynomial, calculated with the Euclidean algorithm. For the case of RS (n, k) we must calculate:

$$\Lambda(\alpha^{-(n-1)}), \Lambda(\alpha^{-(n-2)}), \dots, \Lambda(\alpha^{-1}), \Lambda(\alpha^0)$$

If the expression reduces to 0  $\Lambda(\alpha^i) = 0$ , then that value of  $x$  is a root and identifies the error position else the position does not contain an error.

### 3.1 Proposed algorithm

The proposed algorithm is based on a simple factorization of the error locator polynomial such as  $(x^n)$ , where  $n \geq 2$  should not be depicted in this polynomial. Besides, with this method we can conceive another circuit of the Chien Search Block i.e. we can minimize a large number of the used logic gates, and therefore we will have a low power consumption compared to the basic circuit. If we take the case of RS (15, 11) the error locator polynomial is:

$$\Lambda(x) = 14x^2 + 14x + 1 \quad (4)$$

It's a polynomial of degree 2 as type:

$$\Lambda(x) = A_2x^2 + A_1x + A_0 \quad (5)$$

We factorize the equation (4), we find:

$$\Lambda(x) = X(A_2X + A_1) + A_0 \quad (6)$$

For equation (5) the basic circuit corresponding is represented in Fig.2. The simulation of the basic circuit (equation 5) using Quartus is represented in Fig.5. For the equation (6) the modified circuit using the factorization method is represented in Fig.3. The simulation of the modified circuit (equation 6) using Quartus II is represented in Fig.6.

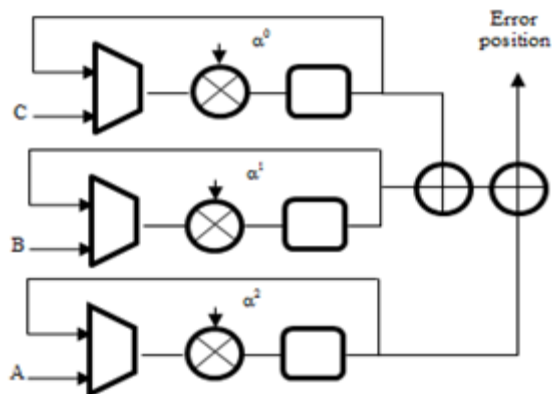


Figure.2. Basic circuit of Chien Search Block

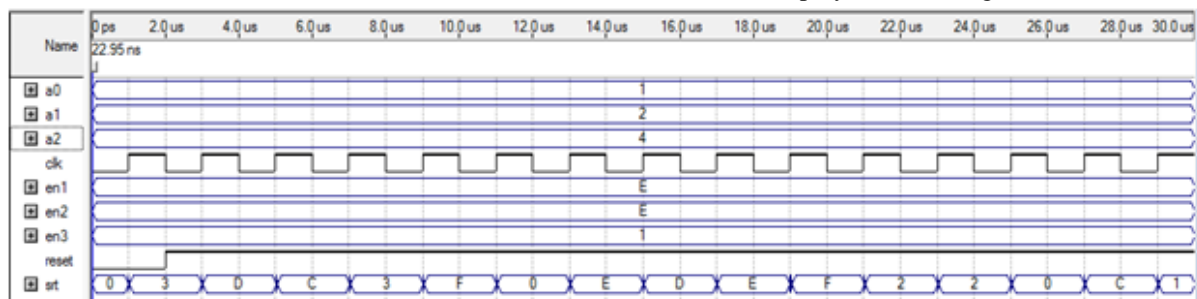


Figure.5. Simulation of the basic circuit (equation 4) using Quartus

Where  $a_0, a_1$  and  $a_2$  represent respectively  $\alpha_0, \alpha_1$ , and  $\alpha_2$  of Fig.2,  $en_1, en_2$  and  $en_3$  represent respectively  $A, B$  and  $C$ ,  $srt$  represents the error position.

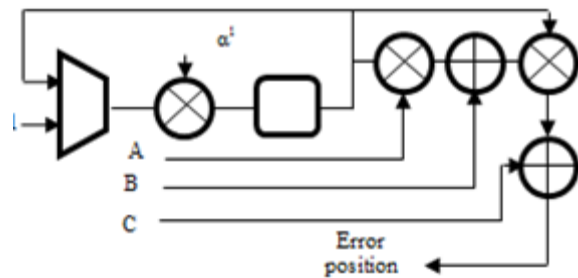


Figure.3. Modified circuit of Chien Search Block  
 For the case of a polynomial of degree 3 we have:

$$\begin{aligned} \Lambda(X) &= A_3X^3 + A_2X^2 + A_1X + A_0 \\ &= X(A_3X^2 + A_2X + A_1) + A_0 \\ &= X(X(A_3X + A_2) + A_1) + A_0 \end{aligned}$$

Generally for:

$$\begin{aligned} \Lambda(X) &= A_nX^n + A_{n-1}X^{n-1} + \dots + A_1X + A_0 \\ &= X(A_nX^{n-1} + A_{n-1}X^{n-2} + \dots + A_1) + A_0 \\ &= X(X(A_nX^{n-2} + A_{n-1}X^{n-3} + \dots + A_2) + A_1) + A_0 \\ &= X(X(X(\dots X(A_nX + A_{n-1}) + A_{n-2}) + \dots) + A_1) + A_0 \end{aligned}$$

Where the coefficients  $A_1, A_2, \dots, A_{n-1}, A_n$  differ from 0. The corresponding logic circuit is represented in Fig.4.

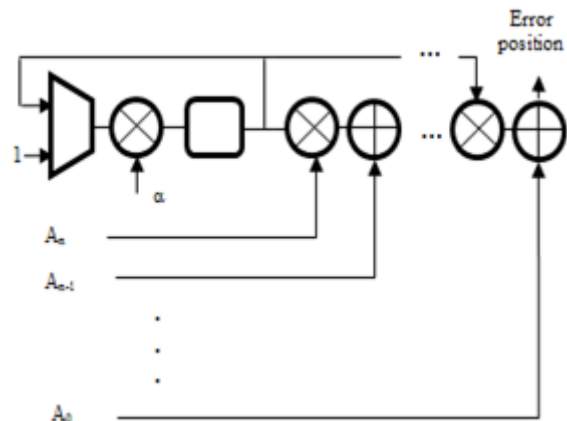


Figure.4. Modified circuit of Chien Search Block for a polynomial of degree n



Figure.6. Simulation of the modified circuit (equation 5) using Quartus

Where EP represents the error position

### 3.2 Comparison of circuits

According to the simulation we have adopted previously, the result we got is the same one in comparison with the modified circuit but with an important number of minimized logic gates. This minimization can save a percentage of power which can reach 50% compared to the basic circuit. The table 1 shows the number of the used logic gates by using both the basic circuit and the modified circuit for different error locator polynomials.

Table 1. Number of minimized logic gates for different error locator polynomials

Error locator polynomial	Number of logic gates for the basic circuit	Number of logic gates for the modified circuit	Number of minimized logic gates
$\Lambda(X) = AX^2 + BX + C$	11	7	4
$\Lambda(X) = AX^3 + BX^2 + CX + D$	15	9	6
$\Lambda(X) = AX^4 + BX^3 + CX^2 + DX + E$	19	11	8
⋮			
$\Lambda(X) = A_n X^n + \dots + A_1 X + A_0$	$3+4n$	$3+2n$	$2n$

## IV. FPGA IMPLEMENTATION

Implementation of decoders for Reed-Solomon 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 [6,7]. In this paper a parameterized hardware model of the Chien Search Block was developed using the Hardware Description Language (VHDL) and synthesized using Xilinx Synthesis Tool. The block

diagram of the proposed algorithm as implemented is shown in Fig.7.



Figure.7. Block diagram of Chien Search block

The proposed Chien Search Block consists of a global 'Clk' and the error detection process is initiated by an 'Input', the 'Error position' can be obtained immediately after entering input.

### 4.1 The new Chien Search block for RS (15, 11) codes

This algorithm can detect the error position by calculating  $\Lambda(\alpha^{-i})$  where  $0 \leq i \leq n-1$ , such as  $\Lambda(x)$  is the error locator polynomial, calculated by the Euclidean algorithm. To try the first position in the code word, corresponding to  $e_j = 14$ , we need to substitute  $\alpha^{-14}$  into the locator polynomial:

$$\begin{aligned} \Lambda(x) &= 14x^2 + 14x + 1 \\ \Lambda(\alpha^{-14}) &= 14(\alpha^{-14})^2 + 14(\alpha^{-14}) + 1 \\ &= 3 \end{aligned}$$

And the non-zero result shows that the first position does not contain an error. For subsequent position, the power of  $\alpha$  to be substituted will advance by one for the  $x$  term and by two for the  $x^2$  term, so we can calculate the calculations as shown in table 2. The two sum values of zero in table 2 identify the error positions correctly as the 6th and 13th symbols, corresponding to the  $x^9$  and  $x^2$  terms, respectively, of the code word polynomial.

Table 2. Terms in the Chien Search block

x	$x^2$ term	x term	unity	Sum
$\alpha^{-14}$	$\alpha^{13}$	$\alpha^{12}$	1	3
$\alpha^{-13}$	$\alpha^0$	$\alpha^{13}$	1	13
$\alpha^{-12}$	$\alpha^2$	$\alpha^{14}$	1	12

$\alpha^{-11}$	$\alpha^4$	$\alpha^0$	1	3
$\alpha^{-10}$	$\alpha^6$	$\alpha^1$	1	15
$\alpha^{-9}$	$\alpha^8$	$\alpha^2$	1	0
$\alpha^{-8}$	$\alpha^{10}$	$\alpha^3$	1	14
$\alpha^{-7}$	$\alpha^{12}$	$\alpha^4$	1	13
$\alpha^{-6}$	$\alpha^{14}$	$\alpha^5$	1	14
$\alpha^{-5}$	$\alpha^1$	$\alpha^6$	1	15
$\alpha^{-4}$	$\alpha^3$	$\alpha^7$	1	2
$\alpha^{-3}$	$\alpha^5$	$\alpha^8$	1	2
$\alpha^{-2}$	$\alpha^7$	$\alpha^9$	1	0
$\alpha^{-1}$	$\alpha^9$	$\alpha^{10}$	1	12
$\alpha^{-0}$	$\alpha^{11}$	$\alpha^{11}$	1	1

#### 4.2 Test procedure for RS (15, 11)

The proposed algorithm was implemented on a Xilinx Spartan 3E-500 FG 320 FPGA (xc3s500e-5fg320). A comprehensive testing environment was developed to test the implemented algorithm. The test setup is shown in Fig.8.



(a) Value of Chien Search Block for RS (15, 11)



(b) Value of Chien Search Block for RS (15, 11)

Figure.8. Values of Chien search block for RS (15, 11) codes

In Fig.8 (a) the value is different from zero, so this position does not contain an error. In Fig.8 (b) the value is equal to zero, so this position contains an error.

#### 4.3 The code specified for DVB-T

The DVB-T standard specifies a (255, 239, t=8) Reed-Solomon code, shortened to form (204, 188, t=8) code, so that the 188 bytes of the input packet will be extended with parity bytes to produce a coded block length of 204 symbols [8]. For this code, the Galois field has 256 elements (m=8) and the polynomial representation of a field element is:

$$a_7x^7 + a_6x^6 + a_5x^5 + a_4x^4 + a_3x^3 + a_2x^2 + a_1x^1 + a_0. \quad (7)$$

Corresponding to the binary numbers 00000000 to 11111111. Alternatively, we can use the decimal equivalents 0 to 255. The specification also mentions the field generator polynomial, given as:

$$P(x) = x^8 + x^4 + x^3 + x^2 + 1. \quad (8)$$

For the case of RS (255, 239) used in DVB-T standard, the decoder can detect 16 errors and correct 8 errors.

#### 4.4 Test procedure for RS (255, 239)

The proposed algorithm was implemented on a Xilinx Spartan 3E-500 FG 320 FPGA (xc3s500e-5fg320). A comprehensive testing environment was developed to test the implemented algorithm. The test setup is shown in Fig.9.



(a) Value of Chien search block for RS (255, 239)



(b) Value of Chien search block for RS (255, 239)

Figure.9. Values of Chien search block for RS (255, 239) codes

The non-zero result in Fig.9 (a) shows that this position does not contain an error, In Fig.9 (b) the value is equal to zero, so this position contains an error.

**4.5 Comparison of algorithms**

According to the tables 2 and 3, which show the FPGA device utilization summary for RS (15, 11) and RS (255, 239) by using the two algorithms, we can conclude that: The proposed algorithm presents a low complexity and a very good performance compared to the basic algorithm.

Table 3. FPGA device utilization summary for RS (15, 11)

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
Number of 4 input LUTs	12	17
Number used as logic	12	17

Table 4. FPGA device utilization summary for RS (255, 239)

Recources	proposed algorithm	basic algorithm
Device	Xilinx Spartan 3E (xc3s500e-5fg320)	
RS (n, k)	RS (255, 239)	
Number of occupied Slices	271	1080
Number used as Flip Flops	35	141
Number of 4 input LUTs	504	1992
Number used as logic	504	1992

**V. CONCLUSION**

A simplified algorithm of Chien Search Block for Reed-Solomon codes has been presented in this paper. This algorithm is based on a simple factorization of the error locator polynomial in order to reduce the number of minimized logic gates.

This minimization reduces the power consumption with a percentage which can reach 50% compared to the basic algorithm. The proposed algorithm has been implemented on a Xilinx Spartan 3E-500 FG 320 FPGA (xc3s500e-5fg320). The results show that the proposed algorithm requires reduced hardware resources compared to the basic algorithm.

**VI. ACKNOWLEDGEMENTS**

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. Shrivastava, UP. Singh, *Error Detection and Correction Using Reed-Solomon Codes, International Journal of Advanced Research in Computer Science and Software Engineering*, 3(8), 2013, 965–969.
- [2] A. Alotaibi, *Implementation of (255, 223) Reed Solomon Code Encoder /Decoder, California state university*, 2012.
- [3] A. Al Azad, MI. Shahed, *A Compact and Fast FPGA Based Implementation of Encoding and Decoding Algorithm Using Reed Solomon Codes, International Journal of Future Computer and Communication*, 3(1), 2014, 31–35.
- [4] R. Huynh, G. Ning and Y. Huazhung, *A Low Power Error detection in the syndrome calculator Block for Reed-Solomon codes RS: (208,188), J. Tsinghua Science and Technologie*, 14(4), 2009, 474-477.
- [5] V. Tilavat, Y. Shukla, *Simplification of procedure for decoding Reed-Solomon codes using various algorithms: An introductory Survey, International Journal of Engineering Development and Research*, 2(1), 2014, 279-283.
- [6] D. Chaudhari, M. Bhujade, P. Dhumal, *VHDL Design and FPGA implementation of Reed-Solomon encoder and decoder for RS (7, 3), International Journal of Science, Engineering and Technology Research*, 3(3), 2014, 563-566.
- [7] VJ. Babrekar, SV. Sakhare, *Review of FPGA Implementation of Reed-Solomon Encoder-Decoder, International Journal of Computer Applications*, 87(8), 2014, 16–19.
- [8] *BBC Research and Development, Reed-Solomon error correction, British Broadcasting Corporation*, 2002.