RESEARCH ARTICLE OPEN ACCESS

Conception of a new Syndrome Block for BCH codes with hardware Implementation on FPGA Card

El HabtiEl IdrissiAnas[1],El Gouri Rachid[1, 2],Ahmed Lichioui[3], Hlou Laamari[1]
[1]Laboratory of Electrical Engineering and Energy System. Faculty of Sciences, University Ibn Tofail Kenitra, Morocco
[2]National School of Applied Sciences, University Ibn Tofail Kenitra, Morocco
[3]National Society of Radio and Television (SNRT), Rabat, Morocco

ABSTRACT
Error Correcting Codes are required to have a reliable communication through a medium that has an unacceptable bit error rate and low signal to noise ratio. Data gets corrupted during the transmission and reception due to noises and interferences. The Bose, Chaudhuri, and Hocquenghem (BCH) codes are being widely used in variety communication and storage systems. In this paper, a simplified algorithm for BCH decoding is proposed with a view to reduce the number of iterations for error detection in the syndrome calculator block of BCH decoders with a percentage of up to 80 % compared to the basic syndrome block. First, we developed the design of the proposed algorithm second, we generated and simulated the hardware description language source code using Quartus software tools and finally we implemented the new algorithm of syndrome block on FPGA card.

Keywords-Digital video broadcasting-satellite-second generation,BCH decoder,Syndrome Block, iterations, add syndromes, FPGA implementation.

I. INTRODUCTION

Error correcting codes are used in satellite communication, cellular telephone networks, body area networks and in most of the digital applications. There are different types of error correcting codes based on the type of error expected, expected error rate of the communication medium, and whether re-transmission is possible or not. Few of them are BCH, Turbo, Reed Solomon, Hamming and LDPC. These codes differ from each other in their implementation and complexity [1]. BCH codes are large class of multiple error-correcting codes. 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 [2].

In practice, the binary BCH and Reed-Solomon codes are the most commonly used variants of BCH codes. They have applications in a variety of communication systems: digital subscriber loops, wireless systems, networking communications, and magnetic and data storage systems.Continuous demand for ever higher data rates and storage capacity provide an incentive to devise very high-speed and space-efficient VLSI implementations of BCH decoders. The first decoding algorithm for binary BCH codes was devised by Peterson in 1960. The objective of this work is to reduce a large number of iterations in the syndrome Block using a new algorithm which allows conceiving another circuit of syndrome block. The rest of the paper is organized as follows: an overview of syndrome calculator block of BCH decoders is Provided in section 2. Section 3 discusses the proposed algorithm and simulation. Finally, FPGA implementation details, hardware performance results are presented in section 4, followed by a conclusion.

II. BACKGROUND AND RELATED WORK

The BCH code is characterized as (n, k, t), where n is the code length, k is the data length, and t is the error correction capability.The n-bit code word $(r_0, r_1, \dots, r_{n-1})$ can be interpreted as a received polynomial [3], $R(x) = r_0 + r_1 x^1 + r_2 x^2 + \cdots + r_{n-1} x^{(n-1)}$.In Syndrome Calculation, 2t syndromes are computed using the following equation:

$$S_i = R(\alpha^i) = \sum_{j=0}^{n-1} r_i\, \alpha^{i.j} = r_0 + r_1\alpha^i + r_2\alpha^{2i} + \cdots + r_{n-1}\alpha^{i(n-1)} \quad (1)$$

The syndrome calculator block provides two results. The first result is whether the received code word is correct. The second result provides the syndrome polynomial which will be used to correct the code word if the code word is erroneous. The most common algorithm to perform the syndrome calculation needs 2t basic cells as defined in Fig.1.Where 1≤i≤2t.For each Syndrome Si, where 1≤i≤ 2t, n iterations or computations are needed to compute the Syndrome Polynomial. All the syndrome coefficients will be equal to 0 if the received code word is correct with at least one coefficient different from 0 if the code word is not correct.Much work has been done to reduce time-consuming computational steps. Costa et al. [4] developed a new fast Fourier transform (FFT) method based on the cyclotomic FFT to compute the syndrome using a method that corresponds to the partial discrete Fourier transform (DFT) of D(x). This algorithm is better than the Horner rule and the Zakharova method.

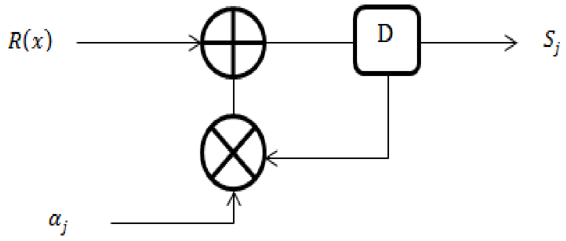

Figure.1. Basic syndrome calculator cell

III. PROPOSED ALGORITHM

The proposed algorithm is based on a change [5, 6] of the received code word such as (j divides n, j is an odd number) in the code BCH (n, k, t) to calculate the add syndromesS_j. Besides, with this method we can conceive another circuit of the Syndrome Block i.e. we can reduce latency by decreasing the number of iterations compared to the basic circuit. For instance, in the case of BCH (15, 7, 2) where: N= 15, k=7 and t=2 => we have 2t = 4 Syndromes: S_1, S_2, S_3 and S_4.

$$R(x) = r_{14}x^{14} + r_{13}x^{13} + \cdots + r_1x^1 + r_0 \qquad (2)$$

The received code word. Knowing that S_1 can be calculated by the direct method and the even syndromes can be calculated using the following relationship $S_{2i} = S_i^2$. To calculate the odd syndromes [7] we use the proposed algorithm as follows:

$$S_3 = R(\alpha^3) = R_{14}\alpha^{42} + R_{13}\alpha^{39} + R_{12}\alpha^{36} + R_{11}\alpha^{33} + R_{10}\alpha^{30} + R_9\alpha^{27} + R_8\alpha^{24} + R_7\alpha^{21} + R_6\alpha^{18} + R_5\alpha^{15} + R_4\alpha^{12} + R_3\alpha^9 + R_2\alpha^6 + R_1\alpha^3 + R_0\alpha^0 \qquad (3)$$

$$R_M = A\alpha^{12} + B\alpha^9 + C\alpha^6 + D\alpha^3 + E\alpha^0 \qquad (4)$$

Where: $A = R_4 + R_9 + R_{14}$
$B = R_3 + R_8 + R_{13}$
$C = R_2 + R_7 + R_{12}$
$D = R_1 + R_6 + R_{11}$
$E = R_0 + R_5 + R_{10}$

So we were able to reduce the sixteen iterations (basic circuit) to six iterations (modified circuit) despite of the fact that we may have more logic gates for the new method. For the case of BCH (255, 231, 3) where:N= 255, k=231 and t=3 => we have 2t = 6 Syndromes: S_1, S_2, S_3, S_4, S_5 and S_6. S_1 can be calculated by the direct method and the even syndromes can be calculated using the following relationship $S_{2i} = S_i^2$.For example, to calculate the odd syndrome S_5 we use the proposed algorithm as follows:

$$R(x) = r_{254}X^{254} + r_{253}X^{253} + \ldots + r_2X^2 + r_1X1 + r_0X^0 \qquad (5)$$

$$S_j\,(j=5) = \sum_{i=0,\ stepof\ 5}^{n-5} A_i * \alpha^i = A_0 + A_1\alpha^5 + A_2\alpha^{10} + \ldots + A_{50}\alpha^{250} \qquad (6)$$

Where:

$$A_0 = r_0 + r_{51} + r_{102} + r_{153} + r_{204}$$
$$A_1 = r_1 + r_{52} + r_{103} + r_{154} + r_{205}$$
$$A_2 = r_2 + r_{53} + r_{104} + r_{155} + r_{206}$$
$$\vdots$$
$$A_{50} = r_{50} + r_{101} + r_{152} + r_{203} + r_{254}$$

So we were able to reduce the 256 iterations (basic circuit) to 52 iterations (modified circuit) despite of the possibility of having more logic gates for the new method.For equation (2) the basic circuit corresponding is represented in Fig.1. The simulation of the basic circuit (equation 2) using Quartus is represented in Fig.2. For the equation (4) the modified circuit using the factorization method is represented in Fig.3. The simulation of the modified circuit (equation 4) using Quartus is represented in Fig.4.

3.1) Syndrome block used in DVB-S2

The DVB-S2 standard lists includes 21 variants of long BCH codes. Each variant is identified by its code block size N_{bch}, uncoded block size K_{bch}, error correction capability t and frame type. Table 1 represents some examples of the 21 variants listed in tables 5a and 5b of the specifications [8].

Table 1. BCH codes used in DVB-S2

Kbch	Nbch	t	Frame
16008	16200	12	Normal
51648	51840	12	Normal
53840	54000	10	Normal
58192	58320	8	Normal
3072	3240	12	short

The codes for normal frames are computed in GF (2^{16}), whereas the short frame codes are computed over GF (2^{14}). The primitive polynomials used to generate the Galois fields are:

$x^{16} + x^5 + x^3 + x^2 + 1$ for GF (2^{16}) and
$x^{14} + x^5 + x^3 + x + 1$ for GF (2^{14})

For the case of BCH (3240, 3072, 12) where:
N= **3240**, k=**3072** and t=**12** => we have 2t = 24 Syndromes: S_1, S_2, S_3, S_4, S_5 … S_{24}.S_1 can be calculated by the direct method and the even syndromes can be calculated using the following relationship $S_{2i} = S_i^2$.To calculate the odd syndromes we use the proposed algorithm as follows:

$$S_j\,(j=9) = \sum_{i=0,\ stepofj}^{n-j} A_i * \alpha^i = A_0 + A_1\alpha^9 + A_2\alpha^{18} + \ldots + A_{359}\alpha^{3231}$$

$$A_0 = r_0 + r_{360} + r_{720} + r_{1080} + r_{1440} + r_{1800} + r_{2160} + r_{2520} + r_{2880}$$
$$A_1 = r_1 + r_{361} + r_{721} + r_{1081} + r_{1441} + r_{1801} + r_{2161} + r_{2521} + r_{2881}$$
$$A_2 = r_2 + r_{362} + r_{722} + r_{1082} + r_{1442} + r_{1802} + r_{2162} + r_{2522} + r_{2882}$$
$$\vdots$$
$$A_{358} = r_{358} + r_{718} + r_{1078} + r_{1438} + r_{1798} + r_{2158} + r_{2518} + r_{2878} + r_{3238}$$
$$A_{359} = r_{359} + r_{719} + r_{1079} + r_{1439} + r_{1799} + r_{2159} + r_{2519} + r_{2879} + r_{3239}$$

So we were able to gain 2880 iterations, 3241 iterations (basic circuit) to 361 iterations (new method) despite of the fact that we may have more logic gates for the new method. The same asS_j(j=3), S_j(j=5) and S_j(j=15).Generally for a code characterized as (n, k, t), where n is the code length, k is the data length, t is the error correction capability and j divides n

We have:

$$S_j = \sum_{i=0,\ stepof j}^{n-j} A_i * \alpha^i$$
$$A_l = \sum_{i=l,\ stepof\ (n/j)}^{i<n} r_i$$

Where:

- j is an odd number
- j divides n
- S_j are the odd syndromes

The corresponding logic circuit is represented in Fig.5.

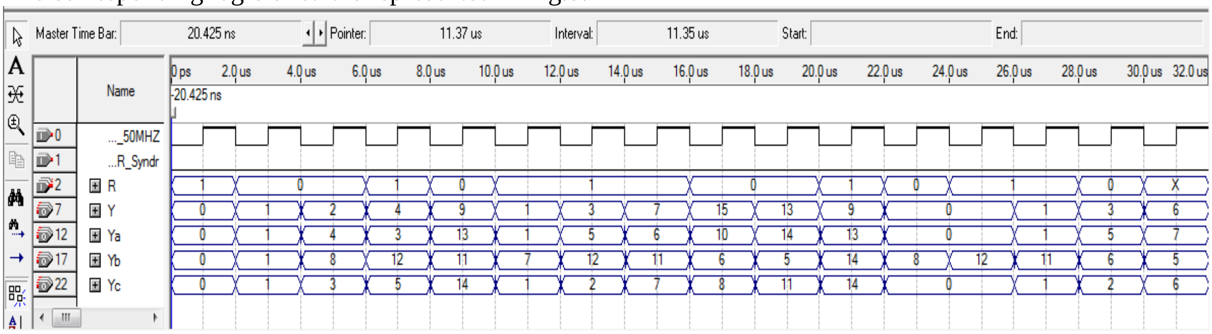

Figure.2. Simulation of the basic circuit (equation 2) using Quartus

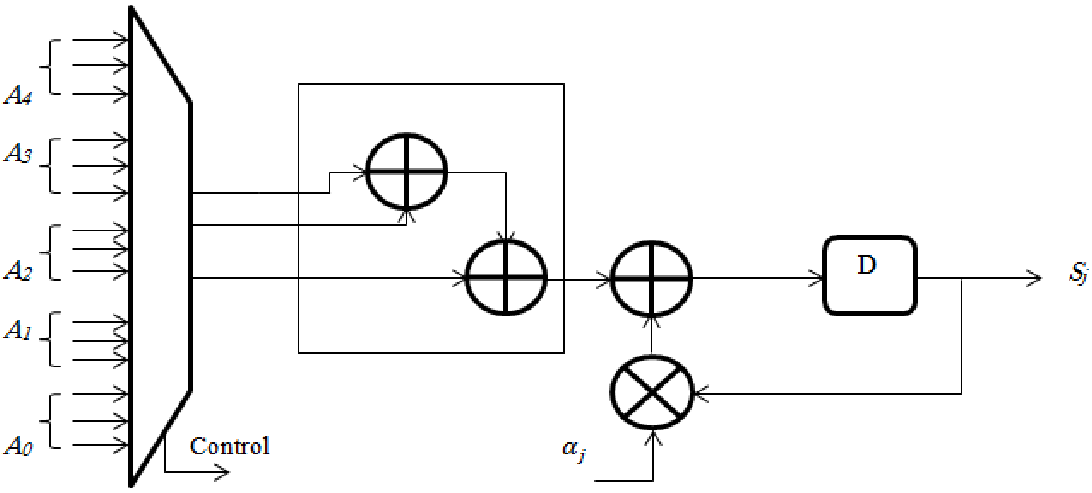

Figure.3. Modified syndrome calculator cell

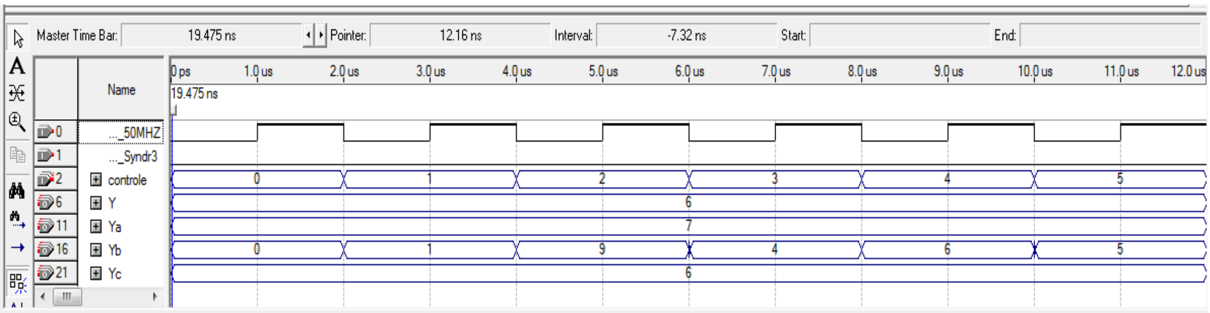

Figure.4. Simulation of the modified circuit (equation 2) using Quartus

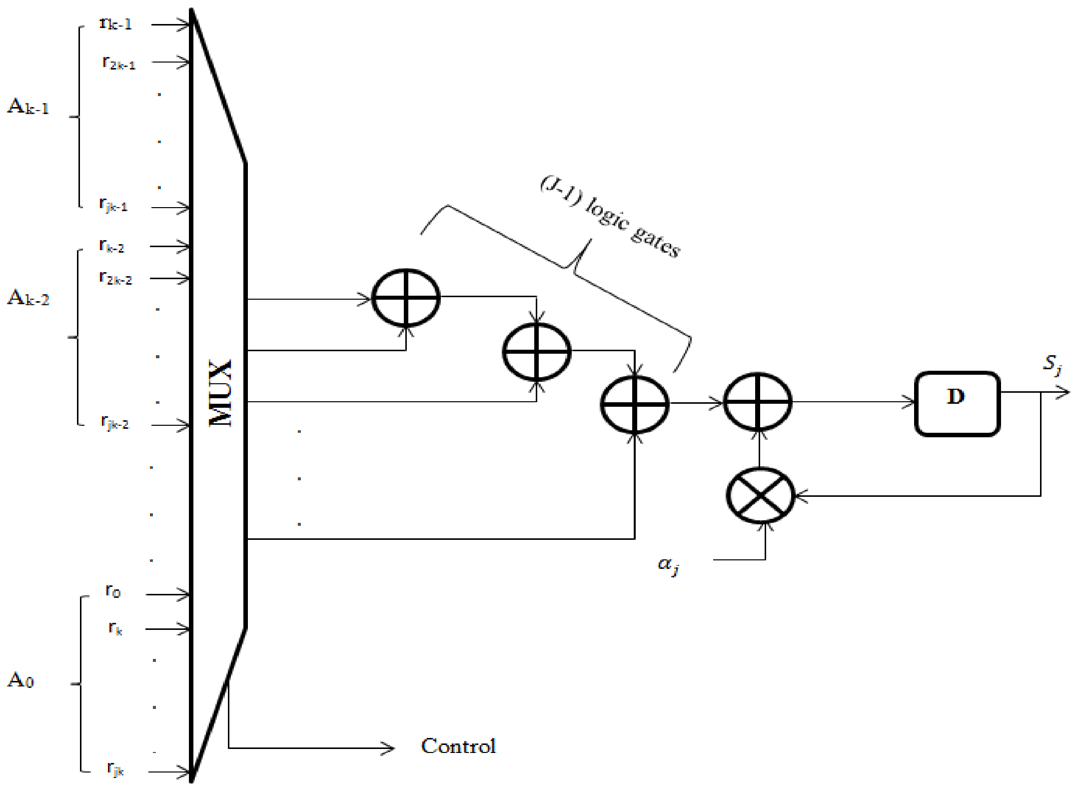

Figure.5. Modified syndrome calculator cell

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 iterations. This reduction of iterations can reach 80% compared to the basic circuit. The table 2 shows the number of the gained iterations by using both the basic circuit and the modified circuit for different BCH codes.

Table 2. Number of gained iterations for different BCH code

BCH Codes	Number of syndromes	value of J	Number of iterations for basic circuit	Number of iterations for modified circuit	Number of gainediterations
BCH (15, 7, 2)	2t = 4	J = 3	16	6	10
BCH (255, 231, 3)	2t = 6	J = 3	256	86	170
		J = 5	256	52	204
BCH (3240, 3072, 12)	2t = 24	J = 3	3241	1081	2160
		J = 5	3241	649	2592
		J = 9	3241	361	2880
		J = 15	3241	217	3024
BCH (n, k, t)	2t	J	n+1	(n/J) + 1	n. {(J-1)/J}

IV. FPGA IMPLEMENTATION

Implementation of decoders for BCH codes has been problematic due to the very large number of resources required. Our quest is to implement a new Syndrome 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 Syndrome 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.6. The proposed Syndrome Block consists of a global 'Clk' and the error detection process is initiated by an 'Input', the 'Syndromes Sj' can be obtained immediately after entering input.

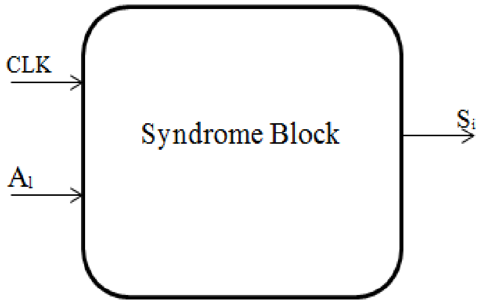

Figure.6. Block diagram of syndrome Block

V. COMPARISON OF ALGORITHMS

According to the tables 3, 4 and 5 which indicate the FPGA device utilization summary for BCH (15, 7) and BCH (255, 231) codes 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 BCH (15, 7, 2)

a) FPGA device utilization summary for modified circuit

Device Utilisation Summary			
LogicUtilisation	Used	Available	Utilisation
Number of Slices	24	4656	0%
Number of Slice Flip Flpos	38	9312	0%
Number of 4 input LUTs	46	9312	0%
Number of bondedIOBs	9	232	0%
Number of GCLKs	1	24	0%

b) FPGA device utilization summary for basic circuit

Device Utilisation Summary			
LogicUtilisation	Used	Availabl e	Utilisatio n
Number of Slices	21	4656	0%
Number of Slice Flip Flpos	34	9312	0%
Number of 4 input LUTs	41	9312	0%
Number of bondedIOBs	13	232	0%
Number of GCLKs	1	24	0%

Table 4. Timing summary for BCH (15, 7, 2)

Timing Summary	Basic circuit	Modified circuit
Minimum input arrival time before clock (ns)	2.992	2.637
Maximum output required time after clock (ns)	7.233	6.580
Maximum combinational path delay (ns)	8.012	7.302

Table 5. FPGA device utilization summary for BCH (255, 231, 3)

a) FPGA device utilization summary for Basic circuit

Device Utilisation Summary			
Logic Utilisation	Used	Availabl e	Utilisatio n
Total Number Slice Registers	56	9312	1%
Number used as Flip Flpos	50		
Numberused as Latches	6		
Number of 4 input LUTs	114	9312	1%
Number of occupied Slices	74	4656	1%
Number of Slices containing only related logic	74	74	100%
Number of Slices containing unrelated logic	0	74	0%
Total Number of 4 input LUTs	144	9312	1%
Numberused as logic	144		
Number used as a route-thru	30		
Number of bondedIOBs	13	232	5%
IOB Latches	3		
Number of BUFGMUXs	2	24	8%
Average Fanout of Non-Clock Nets	3,15		

b) FPGA device utilization summary for Modified circuit

Device Utilisation Summary			
Logic Utilisation	Used	Available	Utilisation
Total Number Slice Registers	56	9312	1%
Number used as Flip Flpos	50		
Numberused as Latches	6		
Number of 4 input LUTs	117	9312	1%
Number of occupied Slices	75	4656	1%
Number of Slices containing only related logic	75	74	100%
Number of Slices containing unrelated logic	0	74	0%
Total Number of 4 input LUTs	147	9312	1%
Numberused as logic	147		
Number used as a route-thru	30		
Number of bondedIOBs	13	232	5%
IOB Latches	3		
Number of BUFGMUXs	2	24	8%
Average Fanout of Non-Clock Nets	3,31		

VI. CONCLUSION

In this paper, we have presented a simplified algorithm of Syndrome Block for BCH codes. This algorithm is based on a simple change of the received code word in order to reduce the number of iterations, the comparison drawn between circuits in Table 1 shows that the BCH code (J = 15) used in DVB S2 has 217 iterations using the modified method instead of 3241 iterations using the basic method. In Table 4 the timing summary for BCH (15, 7, 2) shows that the decrease of the number of iterations reduces latency 8.012ns to 7.302ns and presents a slight increase in hardware resources compared to the basic algorithm. The proposed algorithm has been implemented on a Xilinx Spartan 3E-500 FG 320 FPGA (xc3s500e-5fg320).

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. Mathew, L. Augustine, Sabarinath G and T. Devis, *Hardware Implementation Of (63, 51) Bch Encoder And Decoder For Wban Using Lfsr And Bma,International Journal on Information Theory, 3(3), 2014, 1–11.*

[2] M. Prashanthi, P. Samundiswary, *International Journal of Engineering Trends and Technology, 10(13), 2014, 616–620.*

[3] *Y. Lee, H. YooAnd Ic. Park, Small-Area Parallel Syndrome Calculation For Strong Bch Decoding, IeeeXplore, 2014, 1609-1612.*

[4] *E. Costa, SV. Fedorenko and P. Trifonov, On computing the syndrome polynomial in Reed-Solomon decoder, Eur.Trans. Telecommun, 15(4), 2004, 337-342.*

[5] *A. El habti, R. El gouri and H. Laamari, A low power error detection in the Chien Search Block for Reed-Solomon code, IEEE Xplore*, 2012, 1-3.

[6] *A. El habti, R. El gouri and H. Laamari, 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, 4(8),2014,82-86.*

[7] *J. Yeon, S-J. Yang, C. Kim, and H. Lee, Low-Complexity Triple-Error-Correcting Parallel Bch Decoder, Journal Of Semiconductor Technology And SCIENCE, 13(5),2013,465-472.*

[8] *ETSI EN 302 307, Section 5.3 FEC encoding.*