FPGA Implementation of Reed-Solomon Encoder and Decoder for Wireless Network 802.16

Priyanka Dayal
Post-graduate Student (ECE)
Lovely Professional University
Punjab, India.

Rajeev Kumar Patial
Assistant Professor (VLSI Design)
Lovely Professional University
Punjab, India.

ABSTRACT

A new class of cyclic codes that is Reed-Solomon codes are discussed for IEEE 802.16 wireless networks. Reed-Solomon codes are used for the error detection and correction in communication systems. This is important in information theory and coding to correct burst errors. Here Reed-Solomon code for wireless network 802.16 is synthesized using VHDL on Xilinx and simulated on ISE simulator. The Reed-Solomon encoder has been checked for different error-correcting capabilities that is 4, 6, 8 etc. Reed-Solomon decoder for IEEE 802.16 network is synthesized on VHDL for error detection and correction. Here pipelining is introduced in Reed-Solomon decoder to improve the performance. The performance of Reed-Solomon encoder RS (255,239) for IEEE 802.16 is shown and Reed-Solomon decoder is checked for both RS(255,243) and RS(255,239) and synthesizable on FPGA.

General Terms

Burst error detection and correction, IEEE 802.16 network.

Keywords

Generator polynomial, Syndrome calculation, Berkelamp-massey algorithm, Chien search, error-correction, wireless network 802.16, Pipelining, VHDL, and FPGA.

1. INTRODUCTION

The most commonly used cyclic error correcting codes are the BCH (Bose, Ray-Chaudhari, and Hocquenghem) and Reed-Solomon codes. Reed-Solomon codes are preferred in communication and storage systems because of correction of burst and random errors. Error-correction coding attaches redundancy, for example, parity-check symbols, to the data at the system's error correction encoder and uses that redundancy to correct erroneous data at the error correction decoder [1]. The purpose of error correction coding can be expressed as increasing the reliability of data communications or data storage over a noisy channel, controlling errors so the reliable reproduction of data can be obtained, increasing the overall system's signal-to-noise energy ratio (SNR), reducing noise effects within a system. The reed-Solomon error correction codes were firstly introduced in the paper "Polynomial codes over certain finite fields" in 1960 for burst error correction [2]. These codes are non-binary systematic cyclic linear block codes. These codes work with symbols that consist of several bits. The mostly used symbol size for non-binary codes is 8-bits, or a byte. A systematic code generates codeword that contain the message symbols in unaltered form. The encoder used mathematical function to the message symbols in order to generate the redundancy, or parity symbols. The codeword is formed by appending the parity symbols to the data symbols and is shown in Figure 1. RS codes are generally represented as an RS (n,k), with m-bit symbols,

$$(n,k) = (2^m\text{-}1, 2^m\text{-}1\text{-}2t) \qquad (1)$$

Where k is the number of data symbols being encoded, n is the total number of code symbols in the encoded block, and t is the symbol error correcting capability of the code. The n-k =2t is the number of parity symbols. A codeword is represented by

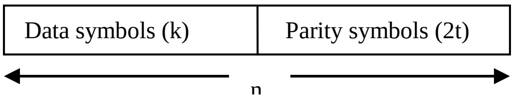

Fig 1: Reed-Solomon Codeword

For non-binary codes, the distance between two codeword's is defined as the number of symbols in which the sequences differ. For RS codes, the code minimum distance is given by

$$d_{min} = n\text{-}k\text{+}1 \qquad (2)$$

The number of advantages of RS codes makes that suitable for IEEE 802.16 wireless network for reliable communication. The IEEE 802.16 standards have been evolved as a set of air interfaces at a 10GHz-66GHz band based on a common MAC layer. IEEE std.802.16 specifies the outer code requirement for code as follows [3]:-

Code generator polynomial:

$$g(x) = (x+\mu^0)\,(x+\mu^1)\,(x+\mu^2)\,...\,(x+\mu^{2t-1}),\ \text{where}\ \mu=02\text{hex} \qquad (3)$$

Field generator polynomial:

$$p(x) = x^8 + x^4 + x^3 + x^2 + 1 \qquad (4)$$

The remaining part of the paper is organized as. Section 2 reviews the proposed Reed-Solomon encoder and Reed-Solomon decoder. Section 3 presented Implementation and results. Section 4 gives conclusion of the paper.

2. REED-SOLOMON CODES

To know the encoding and decoding of the Reed-Solomon codes, the knowledge of finite fields known as Galois Fields (GF), Primitive polynomial, Generator polynomial, Extension field etc is necessary. For any prime number p there exists a finite field denoted GF(p), containing p elements. A monic irreducible polynomial of degree at least one is called a primitive polynomial [4]. The addition and subtraction within the GF(2^m) is performed by exclusive-or or modulo-2. The multiplication and division is performed by the method of add the symbols exponents modulo 2^m-1. Reed-Solomon codes are used in number of applications like storage devices, digital video broadcasting, IEEE 802.16 etc.

2.1 Reed-Solomon Encoder

Reed-Solomon code is represented by RS (n,k) with m-bit symbols. Two types of encoders are available systematic and non-systematic. Systematic encoder is used for Reed-Solomon encoder in which data remains unaltered. This requires that data symbols must be shifted from power level of (n-1) down to (n-k) and remaining positions from power (n-k-1) to 0 be filled with zeros. RS encoder performs two main operations, shifting and division. Both operations are implemented by using linear-feedback shift registers [5]. Let a message or data is represented in the polynomial form as:

$$M(x) = m_{k-1}\, x^{k-1} + m_{k-2}\, x^{k-2} + \ldots\ldots\ldots\ldots + m_1 x + m_0 \qquad (5)$$

And the codeword is represented in polynomial as:

$$C(x) = c_{n-1}\, x^{n-1} + c_{n-2}\, x^{n-2} + \ldots\ldots + c_1 x + c_0 \qquad (6)$$

And the generator polynomial is shown below:

$$g(x) = g_0 + g_1 X + g_2 X^2 + \ldots\ldots + g_{2t-1} X^{2t-1} + X^{2t} \qquad (7)$$

And parity-check is given by:

$$CK(x) = x^{n-k} M(x) \bmod g(x) \qquad (8)$$

And codeword is given by:

$$C(x) = x^{n-k} M(x) + CK(x) \qquad (9)$$

Here, for encoding Reed-Solomon code (255,239) for IEEE 802.16 network the code generator polynomial is used as given below [6]:

$$g(x)=x^{16}+76x^{15}+34x^{14}+67x^{13}+1Fx^{12}+68x^{11}+7Ex^{10}+BBx^{9}+ E8x^{8}+11x^{7}+38x^{6}+B7x^{5}+31x^{4}+64x^{3}+51x^{2}+2Cx+4F \qquad (10)$$

There are 2t=n-k parity symbols that are 16 in this case. The error-correcting capability for this code is t= (n-k)/2 that is 8 symbols. In digital hardware, it is represented by using linear feedback shift register (LFSR) [7]. It is represented in the figure 2.

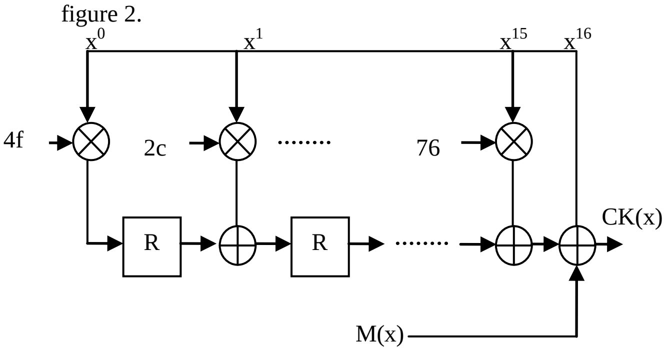

Fig 2: LFSR encoder implementation

2.2 Reed-Solomon Decoder

Decoding processes are cumbersome, difficult to understand and to implement in the hardware than encoding process. Reed-Solomon decoder works in two parts error detection and correction of that detected errors. The capability of correct error is depend on the code uses. It is denoted by t and calculated by the formula t= (n-k)/2 in which n is the length of codeword and k is the length of source data and n-k is the parity symbols. The main blocks of reed-Solomon decoder are the syndrome calculation, Berlekamp-Massey algorithm, and Chien search and error correction [8]. Syndrome means symptoms and it detect whether the error is present or not. If the result of syndrome calculator is zero then it means there is no error and if the result is non-zero then error will be there. The other blocks are used to correct the error if syndrome detects the error. All the algorithms are discussed below:

2.2.1 Syndrome calculation

At the receiver, syndrome is calculated to check the presence of errors. If the error correction capability of code is 8 then syndromes will be 16 that is 2t. The syndrome polynomial contains the location and magnitude of up to t errors in an invalid codeword. A valid codeword generates a syndrome polynomial with all zero coefficients.

$$S(x) = \sum_{i=1}^{2t} s_i\, x^{(i-1)} \quad , s_i = R(\alpha^i) \qquad (11)$$

Where $R(x)= R_0+R_1x+R_2x^2+\ldots\ldots\ldots\ldots+R_{n-1}x^{n-1}$ is the received polynomial and R_{n-1} is the first received symbol.

2.2.2 Berkelamp-Massey algorithm

The main component of Reed-Solomon decoder is the key equation solver. This equation involves 2t linearly dependent equations. It generates the key equations [9]:

$$\mu(x)\, S(x)= \Omega(x) \bmod (x^{2t}) \qquad (12)$$

Where μ(x) is the error locator polynomial and Ω(x) is the error evaluator polynomial. The error locator polynomial contains information about location of bad symbols and error evaluator polynomial contains information about the error magnitude of the bad symbols. There are different techniques for solving key equation are berkelamp algorithm [10] and Euclidean algorithm [11]. The technique used in this paper is the berkelamp-massey algorithm because this algorithm having less hardware complexity.

2.2.3 Chien search and error-correction

Chien search is used to find the roots of polynomial in reed-solomon decoder. Chien search performs the function of finding the error locations, and correcting the data, using Forney's algorithm, when the Chien sum is zero. The evaluation of the polynomials needed for error correction is pipelined. Chien search uses all possible input values and then checks to see if the outputs are zero. In the reed-solomon decoder, the delay block is the necessary to adjust for the delays in the syndrome block and startup delay in the Chien search block. The bock diagram for Reed-Solomon decoder is shown in Figure 3.

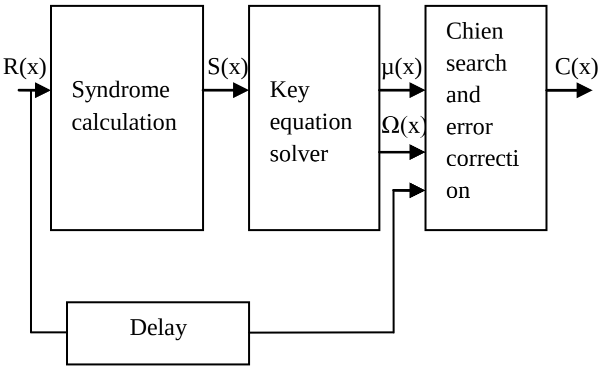

Fig 3: Reed-Solomon Decoder

In this paper, Reed-Solomon Decoder is implemented for both RS(255,243) and RS(255,239) for IEEE 802.16 wireless network. The code generator polynomials for the both codes according to IEEE 802.16 network is given as [6]:

$$G(x)=x^{12}+88x^{11}+C1x^{10}+22x^{9}+33x^{8}+83x^{7}+93x^{6}+A7x^{5}+AAx^{4}+84x^{3}+AFx^{2}+FCx+78 \quad (13)$$

$$G(x)=x^{16}+76x^{15}+34x^{14}+67x^{13}+1Fx^{12}+68x^{11}+7Ex^{10}+BBx^{9}+E8x^{8}+11x^{7}+38x^{6}+B7x^{5}+31x^{4}+64x^{3}+51x^{2}+2Cx+4F \quad (14)$$

For RS(255,243) code, the error-correction capability is 6 and parity or redundant bits are 2t=(255-243)/2 that is 12. This decoder can detect and correct six random errors. For RS(255.239) code, the error correction capability is 8 and parity or redundant bits are 2t=(255-239)/2 that is 16. This decoder can detect and correct 8 symbols. Both codes for decoder are implemented on Xilinx 12.4 using VHDL language and simulated on ISE simulator. The complexity for Reed-Solomon decoder for RS(255,239) increases than RS(255,243).

3. IMPLEMENTATION

Reed-Solomon encoder and Reed-Solomon decoder for IEEE 802.16 wireless network is written in VHDL [12],[13] on Xilinx 12.4 and simulated in ISE simulator and synthesizable on Spartan 3e FPGA. Reed-Solomon codes are implemented to reduce the hardware complexity and utilization.

3.1 Encoder Implementation

Reed-Solomon encoder for RS(255,239) for IEEE 802.16 wireless network is implemented in VHDL to encode the data symbols for reliable communication. The register transfer level (RTL) for this code is shown in figure 4. The RTL view technology for this is shown in figure 5. The design summary is shown in Table 1.

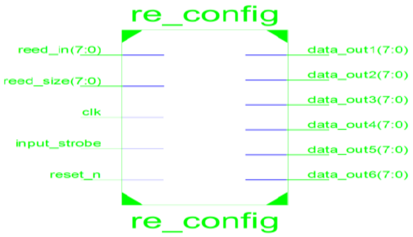

Fig 4: RTL schematic symbol for RS(255,239) encoder

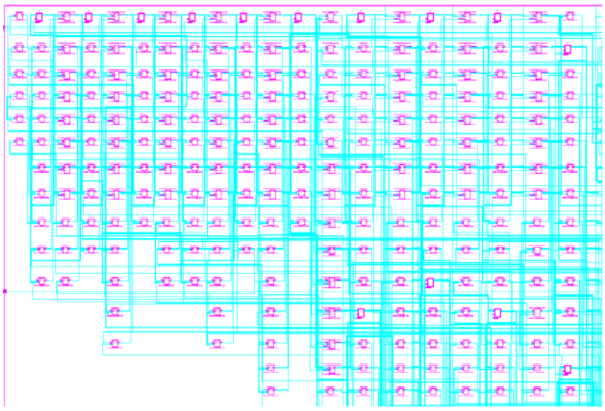

Fig 5: RTL view technology for RS(255,239) for IEEE 802.16 wireless network

Table 1. Design summary for RS(255,239) encoder for 802.16 wireless network

Logic utilization	Used	Available	Utilization
Number of slices	380	4656	8%
Number of slice flip flop	426	9312	4%
Number of 4 input LUTs	719	9312	7%
Number of bonded IOBs	67	232	28%

3.2 Decoder Implementation

Reed-Solomon Decoder for RS(255,243) and RS(255,239) codes for IEEE 802.16 wireless network are implemented in VHDL on Xilinx 12.4 and synthesized on FPGA device. The results are shown in the Figures 6, 7.

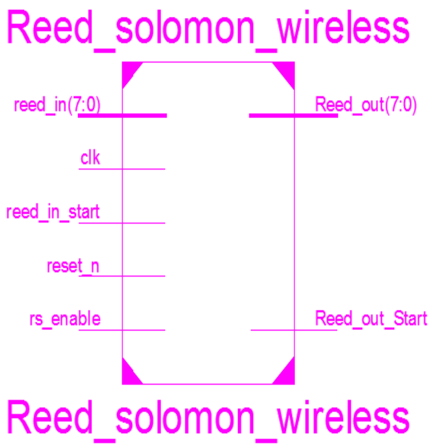

Fig 6: RTL schematic for RS(255,243) decoder

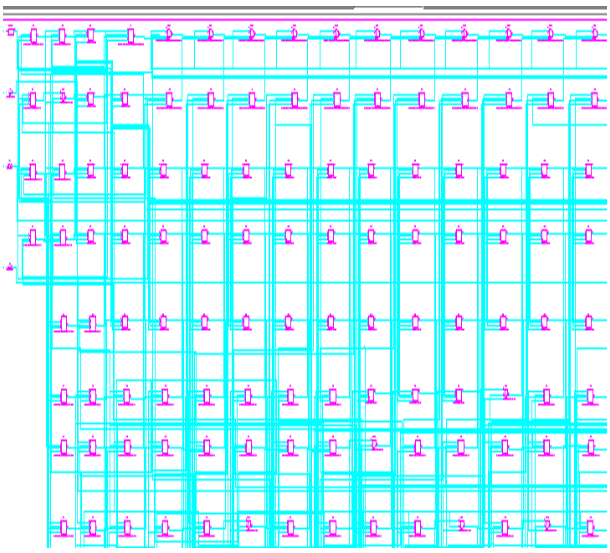

Fig 7: RTL view technology for RS(255,239) decoder

The design summary for both the decoders for IEEE 802.16 wireless network is shown in Table 2 and Table 3. It shows that RS(255,243) code having less hardware utilization than RS(255,239) code. The proposed RS decoder RS(255,239) compares resource utilization with RS decoder given in [8].

Table 2. Design summary for RS(255,239) decoder

	Reed-Solomon decoder [8]		**Proposed RS decoder (255,239)**	
Logic utilization	Used	utilization	Used	Utilization
Number of slices	1830 out of 4656	39%	1848 out of 4656	39%
Number of flip flops	1043 out of 9312	11%	751 out of 9312	8%
Number of 4 input LUTs	3408 out of 9312	36%	3447 out of 9312	37%
Number of bonded IOBs	1 out of 20	5%	21 out of 232	9%

Table 3. Design summary for RS(255,243) decoder

Logic utilization	**Used**	**Available**	**Utilization**
Number of slices	1460	4656	31%
Number of slice flip flop	597	9312	6%
Number of 4 input LUTs	2738	9312	29%
Number of bonded IOBs	21	232	9%

4. CONCLUSION

This paper implements the Reed-Solomon encoder and Reed-Solomon decoder for IEEE 802.16 wireless network. Error detection and correction technique is used here for reliable communication. The main element in the FPGA that is highest in the demand is the LUT’s (look up table). In this paper less number of LUT’s are used in RS(255,243) as comparison to RS(255,239). It means the resource utilization for RS(255,243) is less than the second code. The RS decoder RS(255,239) compares results with the results given by [8]. This represents reduction in the cost and save a lot of area. Reed-Solomon encoder RS(255,239) for IEEE 802.16 is implemented to encode the data symbols. The code is implemented in VHDL on Xilinx 12.4 and synthesized on the FPGA.

5. ACKNOWLEDGEMENT

The author would like to thank Lovely Professional University for providing this opportunity to do work in this field. And also want to thank faculty members for continuous support and encouragement.

6. REFERENCES

[1] National Aeronautics and Space Administration (NASA) technical reports, “Tutorial on Reed-Solomon error correction coding”, Lyndon B. Johnson Space Center Houston, 1990.

[2] Reed, I. S. and Solomon, G., “Polynomial Codes Over Certain Finite Fields,” SIAM Journal of Applied Math., vol. 8, pp. 300-304, 1960.

[3] IEEE draft for “Implementing an IEEE Std. 802.16-Compliant FEC Decoder”, M-WP-IEEE802.16-1.0, ver. 1.0, December 2001.

[4] Sklar, B., Digital Communications: Fundamentals and Applications, Second Edition (Upper Saddle River, NJ: Prentice-Hall, 2001).

[5] Aqib. Al Azad, Minhazul. Huq, Iqbalur, Rahman Rokon, “Efficient Hardware Implementation of Reed Solomon Encoder and Decoder in FPGA using Verilog”, International Conference on Advancements in Electronics and Power Engineering (ICAEPE'2011) Bangkok Dec., 2011.

[6] Lamia Chaari, Mohamed Fourati, Nouri Masmoudi, Lotfi Kamoun, “A Reconfigurable Fec System Based On Reed-Solomon Codec For Dvb And 802.16 Network”, Issue 8, Volume 8, August 2009.

[7] Joel Sylvester, “Reed Solomon Codes”, Elektrobit, January 2001.

[8] Bhawna Tiwari, Rajesh Mehra, “Design and Implementation of Reed Solomon Decoder for 802.16 Network using FPGA” 978-1-4673-1318, IEEE 2012.

[9] Dilip V. Sarwate, Fellow, IEEE, and Naresh R. Shanbhag, Member, IEEE, “High-Speed Architectures for Reed–Solomon Decoders”, IEEE Transactions On Very Large Scale Integration (VLSI) Systems, VOL. 9, NO. 5, OCTOBER 2001.

[10] E.R.Berklamp,” Algebric coding theory, McGraw-Hill, New york, 1968: revised edition Aegean Park Press, Laguna Lills, CA, 1984.

[11] Y.Sugiyama, M. Kasahara, S. Hirasawa, and T. Namekawa, “A method for solving key equation for decoding Goppa codes,” Information and Control, vol. 27, pp. 87–99, Jan. 1975.

[12] Peter J. Ashenden, “The Designer’s Guide to VHDL”, Second Edition, Morgan Kaufmann Publishers, 2004.

[13] Donald G. Bailey, “Design for Embedded Image Processing on FPGA’S” 2011.