A Review Paper on VHDL Implementation of RS-Encoder and Decoder in Spacecraft Technology Security Enhanced

Surbhi Dubey[1] Pallavee Jaiswal[2]
[1]M.Tech Student [2]Asst. Professor
[1,2]Dept. Electronics & Communication Engineering, Dr. C.V. Raman University, Bilaspur, Chhattisgarh-India

Abstract— *In present era of wireless communication error free transmission became necessary. The error correction and detection play very important role for error free transmission of information from one place to another. In long distance communication such as satellite, spacecraft communication there is a need of powerful encoder and decoder. RS Encoder and Decoder is powerful tool for satellite data error correction and detection.*

Keywords— VHDL, RS-Encode, RS-Decoder

I. INTRODUCTION

A VLSI encoder is presented that can function either as an encoder or decoder for Reed-Solomon codes. VLSI is one approach to implementing high-performance Reed-Solomon decoders. A code is the set of all the encoded words, the code word that an encoder can produce. When actual set of data encoded it becomes a code. Reed Solomon error correcting codes (RS codes) are widely used in communication systems and data storages to recover data from possible errors that occur during transmission and from disc error respectively. One typical application of the RS codes is the Forward Error Correction (FEC), the scheme is presented in Figure 1.1.

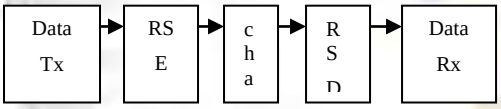

Figure 1.1: Forward Error Correction concept

Before data transmission, the encoder attaches parity symbols to the data using a predetermined algorithm before transmission. At the receiving side, the decoder detects and corrects a limited predetermined number of errors occurred during transmission. Transmitting the extra parity symbols requires extra bandwidth compared to transmitting the pure data. However, transmitting additional symbols introduced by FEC is better than retransmitting the whole package when at least an error has been detected by the receive

Reed-Solomon Code

The Reed-Solomon code was developed in 1960 by I. Reed and G. Solomon. This code is an error detection and correction scheme based on the use of Galois field arithmetic. This section provides background information on binary and extended Galois fields and summarizes the essence of the Reed-Solomon codes.

Galois Fields

A number field has the following properties:

- Both an addition and a multiplication operation that satisfy the commutative, associative, and distributive laws.
- Closure, so that adding or multiplying elements always yields field elements as results.
- Both zero and unity elements. The zero element

Here is a one-to-one correspondence between any binary number and a polynomial in that every binary number can be represented as a polynomial over GF (2), and vice versa. A polynomial of degree D over GF (2) has the following general form:

$$f(x) = f_0 + f_1x + f_2x^2 + f_3x^3 \ldots\ldots. + f_D x^D \quad \ldots\ldots..$$

Where the coefficients f_0..... f_D are taken from GF (2). A binary number of (N+1) bits can be represented as an abstract polynomial of degree N by taking the coefficients equal to the bits and the exponents of x equal to the bit locations. For example, the binary number 100011101 is equivalent to the following polynomial:

$$100011101 \leftrightarrow 1 + x^2 + x^3 + x^4 + x^8 \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$

The bit at the zero position (the coefficient of x^0) is equal to 1, the bit at the first position (the coefficient of x) is equal to 0, and the bit at the second position (the coefficient of x^2) is equal to 1, and so on. Operations on polynomials, such as addition, subtraction, multiplication and division, are performed in an analogous way to the real number field. The sole difference is that the operations on the coefficients are performed under modulo-2 algebra. For example, the multiplication of two polynomials is as follows:

$$(1 + x^2 + x^3 + x^4).(x^3 + x^5) = x^3 + x^5 + x^5 + x^6 + x^7 + x^7 + x^8 + x^9 = x^3 + x^6 + x^8 + x^9$$

This result differs from the result obtained over the real number field (the middle expression) due to the XOR operation (the + operation). The terms that appear an even number of times cancel out, so the coefficients of x^5 and x^7 are not present in the end result.

Extended Galois Fields GF (2^m)

A polynomial p(x) over GF (2) is defined as irreducible if it cannot be factored into non-zero polynomials over GF (2) of smaller degrees. It is further defined as primitive if $n = (x^n + 1)$ divided by p(x) and the smallest positive integer n equals $2^m - 1$, where m is the polynomial degree. An element of GF (2^m) is defined as the root of a primitive polynomial p(x) of degree m. An element α is defined as primitive if

$$\alpha^{i \bmod (2m-1)}$$

where $i \in N$, can produce 2^{m-1} field elements (excluding the zero element). In general, extended Galois fields of class GF (2^m) possess 2^m elements, where m is the symbol size, that is, the size of an element, in bits. For example, in ADSL systems, the Galois field is GF (256). It is generated by the following primitive polynomial

$$1+x^2+x^3+x^4+x^8$$

This is a degree-eight irreducible polynomial. The field elements are degree-seven polynomials. Due to the one-to-one mapping that exists between polynomials over GF(2) and binary numbers, the field elements are represent as binary numbers of eight bits each, that is, as bytes. In GF (2^m) fields, all elements besides the zero elements can be represented in two alternative ways: In binary form, as an ordinary binary number. In exponential form, as α^p. It follows from these definitions that the exponent p is an integer ranging from 0 to (2^m–2). Conventionally, the primitive element is chosen as 0x02, in binary representation. As for GF (2), addition over GF (2^m) is the bitwise XOR of two elements. Galois multiplication is performed in two steps: multiplying the two operands represented as polynomials and taking the remainder of the division by the primitive polynomial, all over GF (2). Alternatively, multiplication can be performed by adding the exponents of the two operands. The exponent of the product is the sum of exponents, modulo 2^m –1.

Polynomials over the Galois field are of cardinal importance in the Reed-Solomon algorithm. The mapping between bit streams and polynomials for GF (2^m) is analogous to that of GF (2). A polynomial of degree D over GF (2^m) has the most general form:

$$f(x) = f_0 + f_1x + f_2x^2 + f_3x^3 \ldots\ldots\ldots\ldots + f_Dx^D \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots..$$

where the coefficients $f_0 – f_D$ are elements of GF (2^m). A bit stream of (N+1)m bits is mapped into an abstract polynomial of degree N by setting the coefficients equal to the symbol values and the exponents of x equal to the bit locations. The Galois field is GF (256), so the bit stream is divided into symbols of eight consecutive bits each. The first symbol in the bit stream is 00000001. In exponential representation, 00000001 becomes α^0. Thus, α^0 becomes the coefficient of x^0. The second symbol is 11001100, so the coefficient of x is α^{127} and so on

Reed-Solomon Encoder and Decoder

Reed-Solomon codes are encoded and decoded within the general framework of algebraic coding theory. The main principle of algebraic coding theory is to map bit streams into abstract polynomials on which a series of mathematical operations is performed. Reed-Solomon coding is, in essence, manipulations on polynomials over GF (2^m). A block consists of information symbols and added redundant symbols. The total number of symbols is the fixed number 2^m –1. The two important code parameters are the symbol size m and the upper bound, T, on correctable symbols within a block. T also determines the code rate, since the number of information symbols within a block is the total number of symbols, minus 2T. Denoting the number of errors with unknown locations n errors and the number of errors with known locations as $\mathbf{n}_{\text{erasures}}$, the Reed-Solomon algorithm guarantees to correct a block, provided that the following is true: $\mathbf{2n_{errors} + n_{erasures} \leq 2T}$, where T is configurable. The current implementation does not deal with erasures, and this document considers only error correction.

Encoding

When the encoder receives an information sequence, it creates encoded blocks consisting of symbols each. The encoder divides the information sequence into message blocks of K ≡ N– 2T symbols. Each message block is equivalent to a message polynomial of degree K –1, denoted as m(x). In systematic encoding, the encoded block is formed by simply appending 2T redundant symbols to the end of the K-symbols long-message block, as shown in Figure 3. The redundant symbols are also called parity-check symbols.

$$N = 2^m - 1$$

K Message Symbols	2T Redundant Symbols

←------- N = K+2T Block Symbols----------→

Figure 1.2. Block Structure

The redundant symbols are obtained from the redundant polynomial p(x), which is the remainder obtained by dividing $x^{2T}m(x)$ by the generator polynomial g(x):

$$p(x) = (x^{2T}m (x)) \bmod g(x)$$

where is the generator polynomial. We choose the most frequently used generating polynomial:

$$g(x) = (x + \alpha^{p1}) (x + \alpha^{p2})(x + \alpha^{p3})(x + \alpha^{p2T})$$
$$g(x) = (x + \alpha) (x + \alpha^2) (x + \alpha^3) (x + \alpha^{2T})$$

The code-word polynomial c(x) is defined as follows:

$$c(x) = x^{2T}m (x) + p (x)$$

Since in GF (2^m) algebra, plus (+), and minus (–) are the same, the code word actually equals the polynomial x^{2T} m(x) minus its remainder under division by g(x). It follows that c(x) is a multiple of g(x). Since there is a total of 2^{mK} different possible messages, there are 2^{mK} different valid code words at the encoder output. This set of 2^{mK} code words of length N is called an (N, K) block code.

Decoding

When a received block is input to the decoder for processing, the decoder first verifies whether this block appears in the dictionary of valid code words. If it does not, errors must have occurred during transmission. This part of the decoder processing is called error detection. The parameters necessary to reconstruct the original encoded block are available to the decoder. If errors are detected, the decoder attempts a reconstruction. This is called error correction. Conventionally, decoding is performed by the Petersen-Gorenstein-Zierler (PGZ) algorithm, which consists of four parts:

1. Syndromes calculation.
2. Derivation of the error-location polynomial.
3. Roots search.
4. Derivation of error values.

The error-location polynomial in this implementation is found using the Berlekamp-Massey algorithm, and the error values are obtained by the Forney algorithm. The four decoding parts are briefly outlined as follows:

1. Syndromes calculation: From the received block, the received polynomial is reconstructed, denoted

as c(x). The received polynomial is the superposition of the correct code word c(x) and an error polynomial e(x)

$$r(x) = c(x) + e(x)$$

The error polynomial is given in its most general form by:

$$e(x) = e_{i_0}x^{i_0} + e_{i_1}x^{i_1} + e_{i_2}x^{i_2} \ldots\ldots e_{i_{(v-1)}} x^{i(v-1)}$$

where i_0, i_1 and so on denote the error location indices, and ν the actual number of errors that have occurred. The 2T syndromes are obtained by evaluating the received polynomial r(x) at the 2T field points:

$$\alpha, \alpha 2, \alpha^3 \ldots\ldots, \alpha 2T$$

Since c(x) is a multiple of g(x), it has the following general form:

$$c(x) = q(x)\, g(x)$$

where q(x) is a message-dependent polynomial. It follows from the definition of g(x) that the following field points:

$$\alpha, \alpha^2, \alpha^3 \ldots, \alpha^{2T}$$

are roots of g(x). Hence c(x) vanishes at the 2T points and the syndromes:

$$S_1, S_2, S_3 \ldots\ldots\ldots, S_{2T}$$

contain only of the part consisting of the error polynomial e(x):

$$S_1 = e(\alpha)$$
$$S_2 = e(\alpha^2)$$
$$S_3 = e(\alpha^3)$$
$$\ldots\ldots\ldots$$
$$S_{2T} = e(\alpha^{2T})$$

If all 2T syndromes vanish, e(x) is either identically zero, indicating that no errors have occurred during the transmission, or an undetectable error pattern has occurred. If one or more syndromes are non-zero, errors have been detected. The next steps of the decoder are to retrieve the error locations and the error values from the syndromes. Denoting the actual number of errors as ν, α^{ik} as X_k and the error values e^{ik} as Y_k, the 2T syndromes $S_1 - S_{2T}$ can then be expressed as follows:

$$S_1 = Y_1X_1 + Y_2X_2 + Y_3X_3 \ldots\ldots\ldots\ldots\ldots Y_v X_v$$

$$S_2 = Y_1(X1)^2 + Y_2(X2)^2 + Y_3(X_3)^2 \ldots\ldots\ldots Y\nu\, (X_v)^2$$

$$S_3 = Y_1(X_1)^3 + Y_2(X_2)^3 + Y_3(X_3)^3 \ldots\ldots\ldots Y_v\, (X_v)^3$$

$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$

$$S_{2T} = Y_1(X1)^{2T} + Y_2(X_2)^{2T} + Y_3(X3)^{2T} \ldots\ldots (Y\nu)\,(X_v)^{2T}$$

Thus, there are 2T equations to solve that are linear in the error values Y_k and non-linear in the error locations X_k.

2. **Derivation** of the error-location polynomial: The output of the Berlekamp-Massey algorithm is the error-location polynomial Λ(x), defined as:

$$\Lambda(x) = (1 + xX_1)(1 + x X\ \ )(1 + xX_3)\ldots(1 + xX_v)$$
$$= 1 + \lambda_1 x + \lambda_2 x^2 + \lambda_3 x^3 + \ldots\ldots \lambda_v x^v$$

Λ(x) has at most ν different roots. The inverses of the roots have the form α^{ik}, where i_k is the error-location index. It can be proven that the so-called Newton identity holds for the coefficients of Λ(x) and the syndromes:

$$S_{j+v} + \lambda_1 S_{j+v-1} + \lambda_2 S_{j+v-2} \ldots\ldots \lambda_v S_j = 0$$

The Berlekamp-Massey algorithm is an iterative way to find a minimum-degree polynomial that satisfies the Newton identities for any j. If the degree of Λ(x) obtained by the Berlekamp-Massey algorithm exceeds T, this indicates that more than T errors have occurred and the block is therefore not correctable. In this case, the decoder detects the occurrence of errors in the block, but no further attempt of correction is made, and the decoding procedure stops at this point for this block.

3. **Roots** search: The roots of the error-location polynomial are obtained by an exhaustive search, that is, by evaluating Λ(x) at all Galois field elements, checking for possible zero results. The exponents of the inverses of the roots are equal to the error-location indices.If the number of roots is less than the degree of Λ(x), more than T errors have occurred. In this case, errors are detected, but they are not corrected, and decoding stops at this point for this block.
4. **Derivation** of error values: The error values are obtained from the Forney algorithm in this implementation. Once the error locations X_k are found, the error values Y_k are found from the ν first syndromes equation by solving the following:

$$\begin{bmatrix} X_1 & X_2 & \ldots & X_v \\ \ldots & \ldots & \ldots & \ldots \\ X_1^v & X_2^v & \ldots & X_v^v \end{bmatrix} \begin{bmatrix} Y_1 \\ \ldots \\ Y_v \end{bmatrix} = - \begin{bmatrix} S_1 \\ \ldots \\ S_v \end{bmatrix}$$

The Forney algorithm is an efficient way to invert the matrix and solve for the errors values $Y_{1-}Y_k$.

II. Literature Review

In the year 2014 Ying Xhang et.al published a paper on IEEE signal processing conference that Error correction coding technology is a valid way to improve the reliability of communication system in which information will be distorted duo to various noises in the channel. This paper considers RS code as the error correction code, and gives the implementation of the RS encoder under CCSDS standard. The implementation of the parallel multiplier with constant coefficients under the natural base is proposed on the basis

of the coding theory of the RS code and the design of the basic unit circuit. In addition, the RS encoder is modeled by MATLAB which prepares procedure's debug for the implementation of the RS encoder. Furthermore, the RS encoder is designed on FPGA by using VHDL, and is simulated under the environment of Modelsim SE 6.5d. The performance of the RS encoder is verified by comparison with the theoretical results. The FPGA implementation shows that the design of the RS encoder occupies minimal hardware resources and can apply to high-speed occasions.

In the year 2014 Aqib Al Azad et.al published a paper on IJFCC This paper presents a compact and fast Field Programmable Gate Array (FPGA) based implementation technique of Encoding and Decoding Algorithm using Reed Solomon (RS) codes, widely used in numerous applications ranging from wireless and mobile communications units to satellite links for correcting multiple errors especially burst type errors. The main objective of this paper is to provide the reader with a deep understanding of the theory of RS code and encoding and decoding of the codes to achieve efficient detection and correction of the errors. The paper will cover the properties of RS code, RS Encoding and Decoding algorithm, simulation, synthesis and Verilog HDL based hardware implementation in FPGA device of the proposed RS Encoder and Decoder architecture. The results of fast and compact implementations of RS Encoder and Decoder architecture using Xilinx's Vertex and Spartan3E FPGA device are presented and analyzed. The design can also be synthesized to other FPGA architectures and is fully flexible & parameterized since block lengths and symbol sizes can be readily adjusted to accommodate a wide range of message sizes.

In the year 2014 Ranjita Singh et.al published a paper on International Journal of Analytical, Experimental and Finite Element Analysis that a system based on Reed-Solomon codec for WiMax networks. The proposed architecture implements various programmable primitive polynomials. A lot of VLSI implementations have been described in literature. This paper introduces a highly parametrical RS-coder-decoder. The implementation, written in a hardware description language (HDL), is based on an Berlekamp massey, Chien Search and Formey Algorithms. Wehave defined an advanced RS encoder-decoder architecture approach which is a key solution for systems. Our approach is used inorder to implement on Xilinx a generic RS coder-decoder for WiMax networks. IEEE Std.802.16 specifies that the codec performs a variable number of check symbols in a codeword. The Reed-Solomon encoder has been checked for different error-correcting capabilities that is 4, 6, 8 etc. Reed-Solomon decoder synthesized using VHDL on Xilinx and simulated on ISE Simulator.The RS decoder implementation written in VHDL is based on Berlekamp Massey,Forney and Chien Search Algorithm. The performance of Reed-Solomon encoder RS (7,3)and,RS(15,9) is shown and Reed-Solomon decoder is checked for RS(7,3)and synthesizable on Xilinx and simulated on ISE Simulator. The performance are shown using two device Virtex 5 and Spartan 3e.

In the year 2013 P.RaviTej et.al published a paper on International Journal of Research in Computer and Communication Technology that The effective Reed Solomon encoder design is based on the improved algorithm widely used in wireless communication system. Taking RS encoder in Digital VideoBroadcasting system, we introduce the structure of RS encoder. Improved algorithm on Galois Field multiplier and Galois Field addition is proposed thatsaves the numbers of the XOR gates. Galois Field depends on Irreducible Polynomial. The proposed architecture implements various programmable primitive polynomials. The feasibility and validity of our proposed work has been verified through Xilinx Simulator.

In the year 2013 Priyanka Dayal et.al published a paper on IJCA that 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.

In the year 2013 NaveenRaj.M et.al published a paper on IJRD that normally decoding is done to correct errors present in the codeword by calculating syndrome. syndrome calculation for large size codeword makes decoding more complex**.** This paper presents an error detection and correction schemes using Majority Logic Decoding(MLD) which provides simple decoding process to detect and correct errors, the existing MLD requires number of clock cycles equal to size of codeword to detect error but the proposed decoding significantly reduces the iterations by detecting error within three clock cycles for any size of codeword using control logic .Also the proposed scheme uses detection logic to detect the uncorrectable codeword if error occured is more than actual capacity of system. Hence this makes error detection and correction schemes more efficient.

III. PROBLEM IDENTIFICATION

In previous work we are transmitting data from earth station to the satellite. The satellite communication is long distance communication and there is a requirement of high speed, high bandwidth and error free transmission. The security of the information is very important in this transmission. The Transmitter and Receiver need powerful encoder and decoder. The RS encoder and decoder is used for error free transmission .The limitation of existing work is that there is lack of security which is to be overcome. There different parameters like speed, bandwidth security which play very important role in error free transmission. Detection and correction codes have been widely used in the last decades; because of them it is possible to guarantee reliable transmissions, avoiding the loss of information. It is possible to send information to so long distances as those we can find

in the space and with no loss of information. Because of this, the study of the correction and detection codes has been studied all codes which have been implemented; there is one which is presented in the most applications.

IV. METHODOLOGY

The parameters like speed, bandwidth and most important security is overcome by Reed-Solomon codes which is implemented in areas such as mobile telephony, storage devices (CD), on satellite transmissions of Digital Video Broadcasting (DVB), digital television ISDB-T and on xDLS systems of wired communications. There is different kind of error correction and detection codes. The importance of the Galois Fields to build up these codes and how the Galois addition and multiplication operations are implemented on hardware in order to work with Reed-Solomon codes on a FPGA. Later, it is described the idea behind Reed-Solomon coding, the coding algorithm is shown along with its implementation. This implementation is generalized to any Reed-Solomon code. After that we explain Reed-Solomon decoding and each module that composes it. We generalize the algorithm and the implementation to be able to build up any Reed-Solomon decoding.

V. EXPECTED OUTCOME

The security and error free transmission is very important in space craft communication technology. The different parameters like speed, bandwidth, security, error correction and detection is to be improved by using new Reed Solomon encoder and decoder communication with UART. We are going to compare these parameters with previous work done. The expected outcome is to be improved.

REFERENCE

[1] Ying Xhang. Implementation of RS encoder for CCSDS in IEEE Signal Processing (ICSP), 2014 12th International Conference on IEEE

[2] Aqib Al Azad and Md Imam Shahed IJFCC. A Compact and Fast FPGA Based Implementation of Encoding and Decoding Algorithm Using Reed Solomon Codes

[3] Ranjita Singh1 Krikshankant Pathak2 VHDL Implementation of Reed-Solomon Encoder-Decoder for WiMax Network

[4] I. S. Reed and G. Solomon, ― Polynomial codes over certain finite fields‖, *Journal of the Society for Industrial and Applied Mathematics*, 8:300-304, 1960.

[5] T. R. N. Rao and E. Fujiwara, Error Control Coding for Computer Systems, *Prentice-Hall, Englewood Cliffs, NJ, USA*, 1989.

[6] H.C.Chang, C.B.Shung, CY.Lee, ―A Reed- Solomon product-code (RS-PC) decoder chip for DVD applications‖, *Solid-State Circuits, IEEE Journal* , Volume: 36 Issue: 2 , Feb. 2001 Page(s): 229 –238.

[7] Y.Cho, K.Kang, J.Lee and H.Shin ―Proactive Reed-Solomon Bypasss (PRSB): A Technique for Real-Time Multimedia Processing in 3G Cellular Broadcast Networks‖, *The 11th IEEE International Conference on Embedded and Real-Time computing system and application*, 17-19 August 2005, Hong Kong

[8] J.SHIAN LI M.W. GUO, ―Improving 802.11 Wireless TCP Performance with Adaptive Reed-Solomon Codes: An Experimental Study‖, *Journal Of Information Science And Engineering* 21, 1201-1211 (2005). International Journal of Analytical, Experimental and Finite Element Analysis (IJAEFEA), Issue. 3, Vol. 1, July 2014 © 2014 RAME IJAEFEA 6 Research Association of Masters of Engineering www.rame.org.in

[9] M.F.Arnal, ―Optimisation de la fiabilité pour des communications multipoint par satellite géostationnaire, *Thèse, Ecole Nationale supèrieure des télecommunications*

[10] DVB, ―Framing structure, channel coding and modulation for digital terrestrial television,‖ *ETSI EN 300* 744, vol. 4.1, January 2001

[11] Lee H., ―A high speed, low complexity Reed-Solomon decoder for optical communications,‖ IEEE Transactions on Circuits and Systems II, PP. 461-465, 2005

[12] WiMAX Forum: ‖Mobile WiMAX. Part I: A TechnicalOverview and Performance Evaluation,‖ August 2006.

[13] Peter J. Ashenden, ―The Designer's Guide to VHDL‖, Second Edition, Morgan Kaufmann Publishers, 2004.