Implementation of Self Checking RS(n,k) Encoder Using VHDL tool

Sabita Mali

Department of Electronics and Instrumentation Engg., ITER,SOA university, Bhubaneswar, Orissa

Abstract **- In this paper we designed a Reed Solomon i.e RS(255,249) structure for wireless communication using VHDL tool. RS codes are systematic linear block codes used to detect and correct burst data errors. RS code has strong error correcting capability than other linear block codes with the same coding efficiency and it does not require a back channel.The Galois field arithmetic is used for encoding of reed Solomon codes. simulation result shows high efficiency,high reliability and low circuit complexity with good coding performance.**

Keywords: **RS, BCH, GALOIS FIELD,DTV, DVB**

I. INTRODUCTION

RS codes are systematic linear block codes resides in a subset of BCH Codes called non-binary BCH.It is block because the original message is split into fixed length blocks and each block is split into 'm' bit symbols, Linear because each m bit symbol is a valid symbol and Systematic because the transmitted information contains the original data with extra redundant or parity bits appended. Examples of important practical applications include magnetic and optical storage systems,wireless or mobile communications,satellite communications,digital television or digital video broadcast(DVB),high speed modems etc

II.RS CODING THEORY

RS codes belong to the family known as block codes. This family is so named because the encoder processes a block of message symbols and then outputs a block of codeword symbols. To be specific, RS codes are non-binary systematic cyclic linear block codes. Non-binary codes work with symbols that consist of several bits. A common symbol size for non-binary codes is 8 bits, or a byte. Non-binary codes such as RS are good at correcting burst errors because the correction of these codes is done on the symbol level. By working with symbols in the decoding process, these codes can correct a symbol with a burst of eight errors just as easily as they can correct a symbol with a single bit error.

RS codes are generally represented as an RS (n, k), with m-bit symbols, where

Block Length : n

No. of Original Message symbols: k

Number of Parity Digits: n-k=2t, n - k = 2t

Minimum Distance: d=2t+1

Such a code can correct up to (n-k)/2 or t symbol.

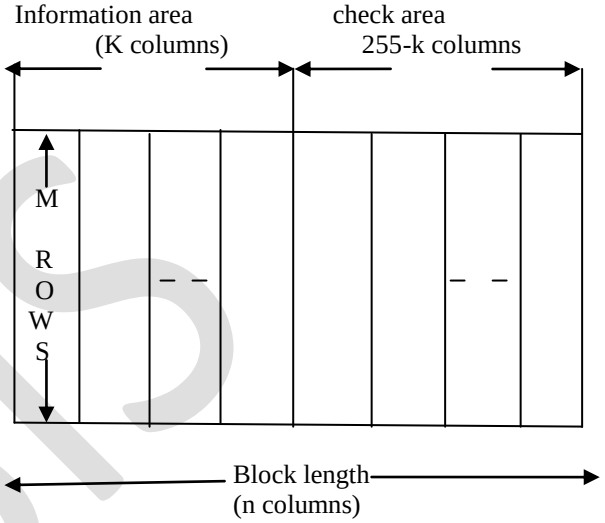


Figure 1. Reed Solomon Codeword

if the codewords can be considered as a collection of binary blocks, all of the same length, say n. Such codes are characterized by the fact that the encoder accepts k information symbols from the information source and appends a set of n-k redundant or parity symbols derived from the information symbols, in accordance with the code algorithm.
In RS (255,249) code, 255 is the total number of data symbols or block length, 249 is the number of information, 6 (255-249=6) are the parity symbols added to the information by the encoding process. Each symbols contain 8 bits (so m=8). So the code is capable of correcting any combination of t or fewer errors.

Where $t = \frac{n-k}{2} = \frac{255-249}{2} = 3$ (1)

That is why we have to add 6 parity symbols to implement any RS code, first of all we have to choose and construct a particular field. . For the implementation of RS (255,249) code, a primitive polynomial of degree 8 has been chosen.

RS (255, K) is a cyclic code, which is based on Galois field $GF^{n}(256)$. The Galois field contains the elements $GF(256) = 0, \alpha^{0}, \alpha^{1}, \alpha^{2}, \alpha^{1}, \ldots\ldots\ldots, \alpha^{254}$ (2)

Where α is the root of following polynomial.

Let's consider a polynomial of degree 8 can represent as

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

By solving $p(\alpha) = 0$ we can calculate the elements of GF(256) $= 0, \alpha^0, \alpha^2, \alpha^3, \ldots\ldots\ldots \alpha^{254}$

The generator polynomial of RS(255,249) is expressed as

$$g(X) = (X- \alpha^0)(X- \alpha^1)(X- \alpha^2)(X- \alpha^3)(x- \alpha^4)(x- \alpha^5)(x- \alpha^6) \quad (4)$$

The degree of the generator polynomial is equal to the number of parity symbols. First the information group m(x) is multiplied by x^{n-k} ,we get product m(x). X^{n-k} and divide it by g(x) to get the corresponding redundant polynomial r(x).

The input information sequence polynomial is expressed as

$$m(x)= m_{k-1} X^{k-1} + m_{k-2} X^{k-2} + \ldots\ldots\ldots + m_1 X^1 + m_0 X^0 = \sum_{i=0}^{k-1} m_i x^i \quad (5)$$

The polynomial of output system code is

$$C(x) = \sum_{i=0}^{254} c_i x^i = x^{255-k} m(x) + r(x) \quad (6)$$

Where $r(x) = \dfrac{x^{255-k} m(x)}{g(x)}$

II.DESIGN OF RS ENCODER

As RS Encoder carries out algebraic operation based on Galois field, it is different from other common binary systems. So, it is relatively complex and it's complexity depends on the size of the Galois field ,there is a need to consider the length of the codeword and the algorithm.

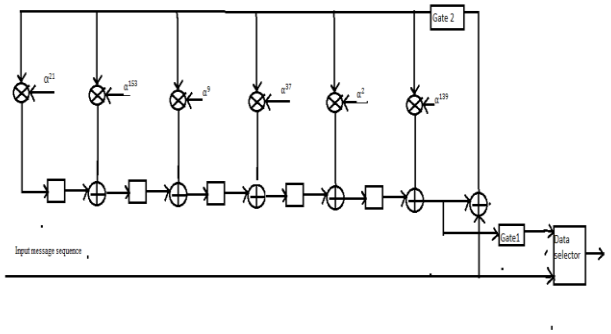


Figure 2. The structure of RS encoder

Where,

⊗ Galois field multiplier

⊕ Galois field adder

□ 8 byte shift-register

The encoder works as follows

(i) Initially all the shift registers are need to be cleared, then open gate 1 and close gate 2 and send the input messages represented by m(x) to these registers. Input messages are divided into two groups in the circuit, one group is output through the data selector, while the other group is multiplied by x^{n-k} ,then it enters the division circuit and is shifted.

(ii) After k shifts all the information enters the circuit completing the division operation , the data in shift registers are coefficients of remainder r(x),which are a parity group of RS code.

(iii) When K+1 clock arrives, it is necessary to pause the input message code group, close gate 1 and open gate 2; all the data are shifted out after going through 255-249 shifts, hence we obtain the parity or check group that follows the message or information code group forming an RS(255,249) code.

III.GALOIS FIELD MULTIPLIER

GF multiplier is the most used block in the design of encoder and decoder. Hence implementation of the multiplier is the back bone of the design. In Our design GF (2^8) is used and the primitive polynomial chosen is $p(X) = X^8 + X^4 + X^3 + X^2 + 1$. The binary representation of this primitive polynomial is 10111001. The basic difference between binary multiplication and Galois field multiplication is that Galois field is finite. So the result of the multiplication of two 8 bit Galois field elements should be of 8 bits but in case of binary multiplication it is of 16 bits. There are various approaches to implement GF multiplier we have considered.

Generally we have bit serial and bit parallel structures for a finite field multiplier. For serial and parallel multiplier various methods have been proposed. The bit-serial architectures process one bit/clock cycle, are areas - efficient and suitable for low-area applications. The bit-parallel architectures, process one word/clock cycle, are ideal for high speed applications.

Hardware implementation of serial multiplier is relatively simple, but it is difficult to achieve high speed due to its bit bit by operation . so bit parallel multipliers are used in high speed applications.

IV. THE DESIGN OF RS (255,249) ENCODER

4.1 GF (Galois Field) Elements:

A Galois field is a finite field with a finite field order.

The order of field is always prime or the power of prime. The Galois field GF(2^m) contains (2^m -1) non zero elements. All field contains of zero element and an element called the primitive element α.Such that non zero elements can be expressed as power of α.

A infinite set of elements , F , is generated starting with α and progressively multiplying the last element by α.Which can be made finite by irreducible polynomial $\boldsymbol{\alpha^{(2m-1)} + 1=0.}$

m	**Primitive polynomials**
1	$1+x$
2	$1+x+x^2$
3	$1+x+x^3$,$1+x^2+x^3$
4	$1+x+x^4$, $1+x^3+x^4$
5	$1+x^2+x^5$,$1+x+x^2+x^3+x^5$,$1+x^3+x^5$,$1+x+x^3+x^4+x^5$,$1+x^2+x^3+x^4+x^5$,$1+x+x^2+x^4+x^5$
6	$1+x+x^6$
7	$1+x^3+x^7$
8	$1+x^2+x^3+x^4+x^8$
9	$1+x^4+x^9$
10	$1+x^3+x^{10}$
11	$1+x^2+x^{11}$
12	$1+x+x^4+x^6+x^{12}$
13	$1+x+x^3+x^4+x^{13}$
14	$1+x+x^6+x^{10}+x^{14}$

Table 1: List of primitive polynomial

All the elements of the field GF (2^8) is found by raising the power of primitive polynomial in increasing order . The basic element is10000000 that denotes which is also called primitive element. The generated field elements have been used in our design. Generated GF elements against corresponding power of the primitive element are expressed in hexadecimal form. We have followed the LFSR approach to generate all the 256 elements.

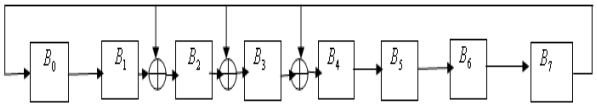


Figure. : GF multiplier for finding all the field elements for p(X) = $X^8 + X^4 + X^3 + X^2 + 1$

4.2 Generator Polynomial :

Generator Polynomial is the vital part of the RS encoder and decoder design. The generator polynomial is used for the calculation of redundant symbols in RS encoder. It is also used for detection of error in the decoder. A Reed Solomon codeword is generated using a special polynomial. All valid code words are exactly divisible by the generator polynomial.

The form of generator polynomial is for (255,249) is:

$$g(X) = (X- \alpha^0)(X- \alpha^1)(X- \alpha^2)(X- \alpha^3)(x- \alpha^4)(x- \alpha^5)(x- \alpha^6) \quad \textbf{(7)}$$

After multiplying all the terms in the above equation we get

$$g(x) = x^6 + (\alpha^6 + \alpha^5)x^5 + \alpha^{11}x^4 + (\alpha^4 + \propto^3 + \propto^2 + \propto)x^5 + (\alpha^4 + \propto^3 + \propto^2 + \propto)(\alpha^6 + \alpha^5)x^4 + (\alpha^{15} + \propto^{14} + \propto^{13} + \propto^{12})x^3 + (\alpha^4 + \propto^3 + \propto^6 + \alpha^7)x^4 + (\alpha^6 + \alpha^5)(\alpha^4 + \propto^3 + \propto^6 + \alpha^7)x^3 + (\alpha^{15} + \propto^{14} + \propto^{17} + \propto^{18})x^2 + (\alpha^9 + \propto^8 + \propto^7 + \propto^6)x^3 + (\alpha^6 + \alpha^5)(\alpha^9 + \propto^8 + \propto^7 + \propto^6)x^2 + (\alpha^{20} + \propto^{19} + \propto^{18} + \propto^{17})x + \alpha^{10}x^2 + (\alpha^{16} + \alpha^{15})x + \alpha^{21} \quad (8)$$

Putting the values of power of α generated from the field polynomial we get the following expression.

$$g(x) = x^6 + \alpha^{139}\ x^5 + \alpha^2\ x^4 + \alpha^{37}\ x^3 + \alpha^9\ x^2 + \alpha^{153}\ x + \alpha^{21} \quad (9)$$

4.3 Design Details

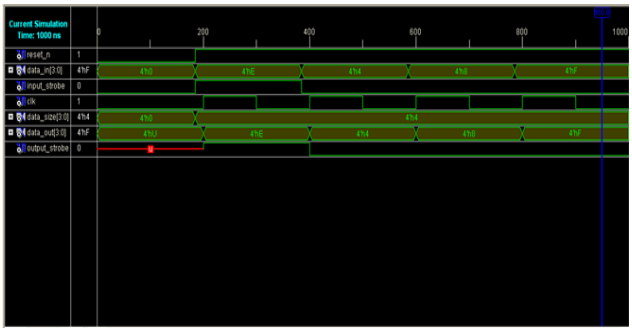
The encoder has been designed as per the architecture described above. Input output description is given in table 2.

SIGNAL	TYPE	WIDTH	DESCRIPTION
Data_in	In	4	Input Data
Input_strobe	In	1	Input strobe Signal
Clk	In	1	Input Clock
Rst_n	in	1	Active Low Reset
Data_out	out	4	Encoder output Data
Data_size	in	4	Data size

Table 2: Input & Output description of Encoder

V. SIMULATION RESULT

We have done the behavioural level RTL code and self-checking test bench and used the tools VHDL(Xilinx) and verified the output waveform. We have done the synthesis successfully.



VI. CONCLUSIONS

Reed Solomon is error correcting codes for burst errors which can correct the error both in message part and parity part. The key idea behind Reed-Solomon code is that the data are visualized as a polynomial.

We designed a Reed-Solomon (255, 249) structure for wireless communication system using VHDL tool, which is based on analysing the coding theory of Reed-Solomon (RS) code . The simulation result shows that the encoder is consistent with its theoretical analysis because its generating polynomial and the realization of constant coefficient multiplier are definite, so it can be implemented easily in hardware.

REFERENCES

[1]. Jie Meng, Min Shen, Min Zhang, "New Application of Reed-Solomon Codes in China Mobile Multimedia Broadcasting System," *IEEE Computer Society*, vol.1, pp.511-514, 2009.

[2]. Qiuyang He, "The Design and Implementation of RS encoder based on FPGA," *Electronic Science and Technology*, vol.22, no.8, pp.44-46, 2009.

[3]. Berlekamp E R., "Bit-serial Reed-Solomon encoders," *IEEE Trans Information Theory*,vol.28, no.6, 1982.

[4]. Wang C C., "VLSI architectures for computing multiplications and inverses in GF(2^m) ," *IEEE Trans Computer*, vol.34, no.8, 1985.

[5]. P.Ravi Tej ,K.Jhansi Ran , "VHDL Implementation of Reed Solomon Improved Encoding Algorithm " , *International Journal of Research in Computer and Communication Technology*, Vol 2, Issue 8, August -2013

[6]. Diplaxmi Chaudhari , Mayura Bhujade, Pranali Dhumal Hagenauer, "VHDL Design and FPGA Implementation of Reed Solomon Encoder and Decoder for RS (7,3)" , *International Journal of Science, Engineering and Technology Research (IJSETR)*, Volume 3, Issue 3, March 2014