VLSI implementation of Reed Solomon encoder

Chanchal Ghadse
M.V.Vyawahare
Priyadarshini College of Engineering
Nagpur, India
cghadse@yahoo.com

***Abstract*—Reed Solomon codes are widely used to identify and correct data errors in transmission and storage system. RS is a block forward error correction capable of correcting multiple errors. This code is widely used in wireless and mobile communication units. This paper present a design of (7, 3) Reed Solomon encoder using VHSIC hardware description language.**

***Keywords*- encoder, Reed Solomon codes, VHDL.**

I. Introduction

Reed-Solomon codes are block-based error correcting codes with a wide range of applications in digital communications and storage such as

- Storage device (hard disks, compact disks, DVD).
- Wireless communication (mobile phone, microwave links).
- Digital television.
- Satellite communication.
- Broadband modems (ADSL).

One typical application of the RS codes is the Forward Error Correction (FEC), shown in Fig(1)

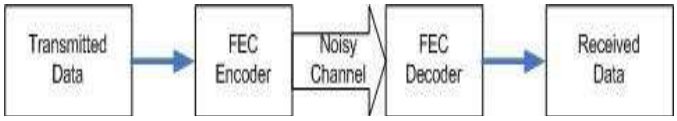

Fig. 1. Forward Error Correction Concept

Reed Solomon codes work by adding extra "redundant bits" to the original data. The encoded data can then be transmitted or stored. During transmission error may happen for a number of reasons e.g. (scratches on CD, radio frequency interference with mobile phone reception, noise etc.) At the receiving side, the decoder detects and corrects a limited predetermined number of errors occurred during transmission.

II. Reed Solomon theory

A Reed-Solomon code is a block code and can be specified as RS (n, k) as shown in Fig. 2. The variable n is the size of the codeword with the unit of symbols, k is the number of data symbols and 2t is the number of parity symbols. Each symbol contains m number of bits.

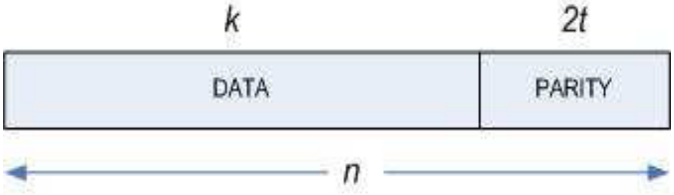

Fig. 2. The structure of a RS codeword

The relationship between the symbol size, m, and the size of the codeword, n, is given by (1). This means that if there are m bits in one symbol, there could exist 2^m-1 distinct symbols in one codeword, excluding the one with all zeros.

$$n = 2^m - 1 \qquad (1)$$

The RS code allows correcting up to t number of symbol errors where t is given by

$$t = (n - k) / 2 \qquad (2)$$

III. Galois field

Reed Solomon codes works on Galois field. The Galois field is a finite set of elements which has defined rules for arithmetic. For any prime number, *p*, there exists a finite field denoted by GF (*p*) that contains *p* elements. It is possible to extend GF (*p*) to a field of p^m elements, called an *extension field* of GF (*p*), and denoted by GF (p^m), where *m* is a nonzero positive integer. Symbols from the extension field GF (2^m) are used in the construction of Reed-Solomon (RS) codes. Elements in the extension field are represented with a new symbol α. Each nonzero element in GF (2^m) can be represented by a power of α. the elements of the finite field, GF (2^m), are as follows:

$$GF(2^m) = \{0, \alpha^0, \alpha^1, \alpha^2, \ldots, \alpha^{2m-2}\} \qquad -(3)$$

Each of the 2^m elements of the finite field, GF (2^m), can be represented as a distinct polynomial of *degree m* - 1 or less. The degree of a polynomial is the value of its highest-order exponent. We denote each of the nonzero elements of GF (2^m) as a polynomial, $a_i(X)$, where at least one of the *m* coefficients of $a_i(X)$ is nonzero. For $i = 0,1,2,\ldots,2^m - 2$.

$$\alpha^i = a_i(X) = a_{i,0} + a_{i,1}X + a_{i,2}X^2 + \ldots + a_{i,m-1}X^{m-1} \qquad -(4)$$

In this project considering m=3 the finite field is denoted GF(2^3). Fig. 3 shows the mapping of the seven elements $\{\alpha^i\}$ and the zero elements, in terms of the basis elements $\{X^0, X^1, X^2\}$ described by (3).

TABLE 1: BASIS ELEMENTS

Exponent	Polynomial	Binary
	X^0 X^1 X^2	X^0 X^1 X^2
0		0 0 0
α^0	1	1 0 0
α^1	x	0 1 0
α^2	x^2	0 0 1
α^3	1 + x	1 1 0
α^4	$x + x^2$	0 1 1
α^5	$1 + x + x^2$	1 1 1
α^6	$1 + x^2$	1 0 1
$\alpha^7 = \alpha^0$	1	1 0 0
$\alpha^8 = \alpha^1$	x	0 1 0

Fig.3: Elements for GF (2^3) with $F(X) = 1 + x + x^3$.

A. Operations in Galois Field

An addition of two elements in the Galois field is simply the Exclusive-OR (XOR) operation. However, a multiplication in the Galois field is more complex than the standard arithmetic. It is the multiplication modulo the primitive polynomial used to define the Galois field. For example, a Galois field, GF (8), is constructed with the primitive polynomial $F(X) = 1 + x + x^3$.

IV. Encoder

Reed-Solomon codes operate on the information by dividing the message stream into blocks of data, adding redundancy per block, dependent only on the current inputs. The symbols in Reed Solomon coding are elements of a Galois Field (finite field). Encoding **is** achieved by appending the remainder of a Galois field polynomial division into the message. This division is done by a *Linear Feedback Shift Register* (LFSR) implementation. The LFSR is the main computational element of the RS Encoder. The mathematics of RS encoding is based on Finite Field operations. To define Galois field, primitive polynomial used is

$$G(X) = 1+X+X^3 \qquad -(3)$$

The system implemented in this project was a (7,3) system, in which (n, k) denotes an output codeword length of n and an input word of length k.

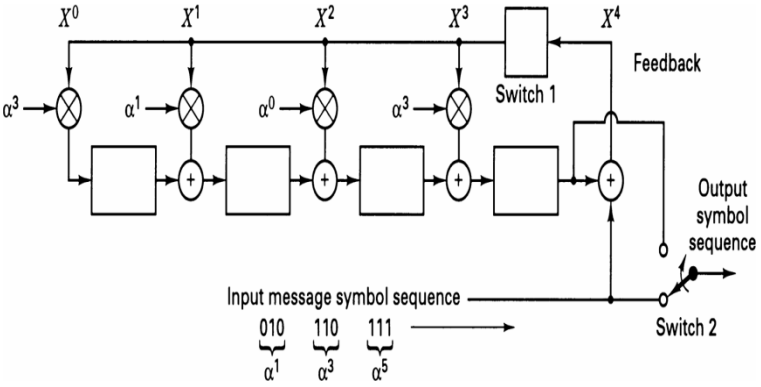

Fig.4: Reed-Solomon Encoder Block Diagram (LFSR)

It has a symbol size (m) equal to three.

$$n-k = 2t$$

where t is error correcting capability so encoder has double error correcting capability.

$$7-3 = 2t$$

Therefore $t = 2$

The encoder forms a code word $X^{n-k}\,\mathbf{m(X) + p(X)}$ by means of the following equations:

$$p(X) = X^{n-k}\, m(X) \bmod g(X) \qquad -(5)$$

$$U(X) = p(X) + X^{n-k}\, m(X) \qquad -(6)$$

where g(X) is generator polynomial and *m(X)* is message, *p(X)* is parity and *U(X)* is coded message polynomial.
The generating polynomial for an R-S code takes the form:

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

Degree of the generator polynomial is equal to the number of parity symbols. Generator polynomial has $2t = n - k = 4$ roots, they are

$$g(X) = (X-\alpha)(X-\alpha^2)(X-\alpha^3)(X-\alpha^4)$$
$$= X^4 - \alpha^3 X^3 - \alpha^0 X^2 - \alpha^1 X + \alpha^3$$

In binary field +1= -1.for this case generator polynomial is given by

$$g(X) = \alpha^3 + \alpha^1 X + \alpha^0 X^2 + \alpha^3 X^3 + X^4 \qquad -(8)$$

In digital hardware, the encoder is an LFSR with internal feedback connections corresponding to *g(X.* The operations involved are GF addition and multiplication. To encode a three symbol sequence in systematic form with (7, 3) RS code described by g(X) is implemented using LFSR circuit shown in fig .4

The result of the simulation is shown in Fig.5.

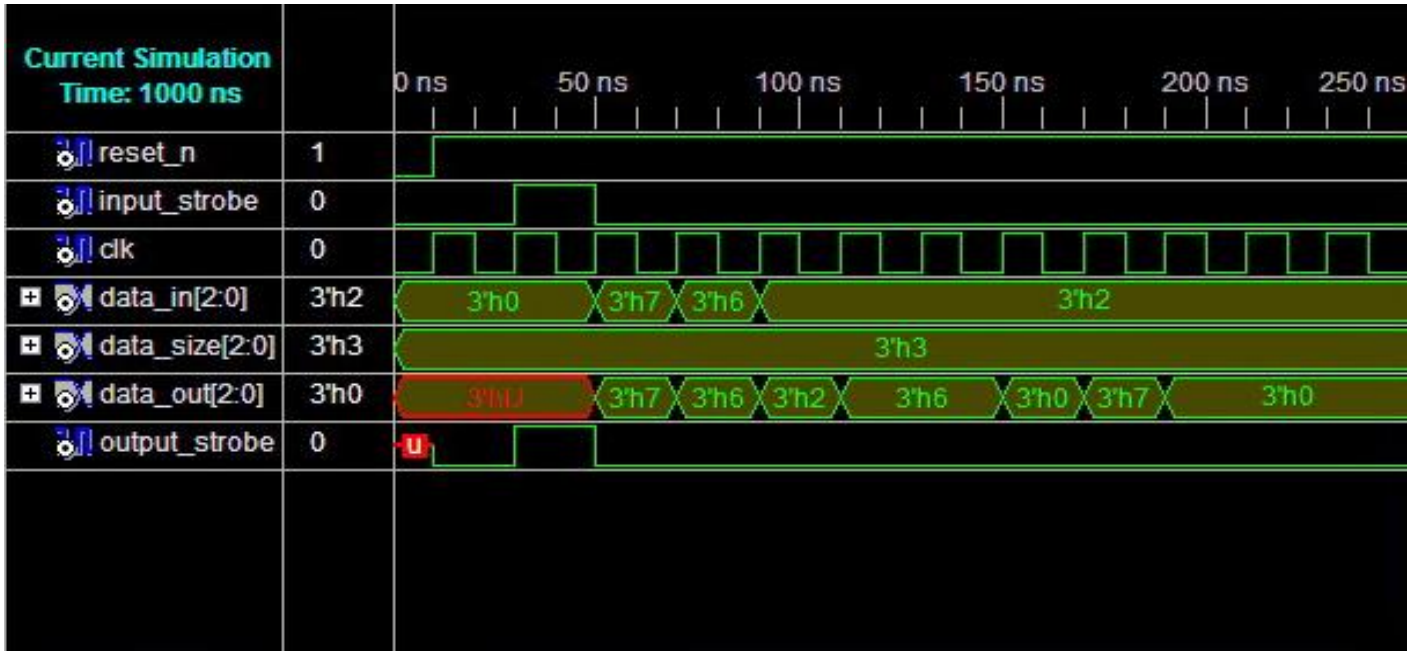

Fig.5: Simulation Result

References

[1] A. Amina, P. Chio, I. A. Sahagun and D. J. Sabido IX, "VLSI Implementation of A (255,223) Reed-Solomon Error- Correction Codec," Roc. Of Second National ECE Conference.

[2] Shahab Ardalan, Kaamran Raahemifar, Fei Yuan and Vadim Geurkov ," Reed-Solomon Encoder & Decoder Design, Simulation and Synthesis ," Electrical and Computer Engineering Department Ryerson university Toronto, Canada CCECE2003ZCGEI 2003, Montrial, Maylmai *2003* 0-7803-7781-8/03/$17.00 *2003 IEEE*.

[3] D. V. Sarwate and N. Rshanbhag, "High-speed Architectures for Reed-Solomon Decoder," IEEE transaction on VLSI system, OCT, 2001.

[4] H. C. Chang and C. B. Shung, "New Serial Architecture for Berlekamp-Massey Algorithm," IEEE transaction on communications, April 1999.

[5] Bernard Sklar "Digital Communications: Fundamentals and Applications".