VHDL Design and FPGA Implementation of Reed Solomon Encoder and Decoder for RS (7,3)

Diplaxmi Chaudhari , Mayura Bhujade, Pranali Dhumal

***Abstract*— In this paper, Reed-Solomon (RS) encoder and decoder for RS (7,3) codec and their hardware implementation in Actel ProASIC3 (Field Programmable Gate Array (FPGA) kit is analyzed. RS codes are subclass of non binary cyclic error correcting block codes which can correct burst errors at the receiver. The RS code provides a wide range of code rates which can be chosen to obtain the optimum performance. Parity symbols are generated in the encoder using a generator polynomial by shift register concept and then concatenated with the input message symbols. RS decoder determines the location and magnitude of errors in the received polynomial caused due to noise while communication. For this, efficient decoding techniques like Chien, Forney and Berlekamp Massey algorithms are used by the decoder. The thesis proposes RS encoding and decoding algorithm, synthesis and simulation results of RS encoder using Very High Speed hardware description Language (VHDL) and ProASIC3 FPGA.**

Index Terms— **Reed Solomon (RS), FPGA, Chien, Forney, VHDL Syndrome calculator, Key Equation Solver (KES)**

I. INTRODUCTION

In practical communication system data or information may get corrupted by noise during transmission. Now a day as demand is continuously increasing for development of reliable telecommunication and wireless systems, it is important to detect and correct errors in the information received over communication channels. Therefore error control coding is important in communication system design for various applications.

Reed Solomon codes [1] are systematic linear block error correcting codes and these are sub class of non binary BCH error correcting codes. RS codes operate on the information by dividing the message stream into blocks of data. Then redundancy can be added as per block depending only on the current inputs. The symbols are elements of a finite field or Galois Field (GF). Galois field is used for encoding and decoding of Reed Solomon codes. GF multipliers are basically used for encoding purpose. The coefficients of the RS generator polynomial are nothing but the multiplier coefficients. After that, encoding is achieved by adding the remainder of a GF polynomial division into the message. For implementation of this division method, linear feedback Shift Register (LFSR) technique is used [7]. At the decoder, the syndrome calculation of the received codeword is carried out. Then we find error locations using Chien algorithm and error magnitudes using Forney algorithm. Further Massey Berlekamp is used to calculate coefficients of error locator polynomial for error locations and coefficients of error evaluator polynomial for error values.

RS codes have a widespread use to provide error correction especially for burst errors. This feature has been an important factor in adopting RS codes in many practical applications such as wireless communication system, cable modem, computer memory etc. The paper covers RS theory in section II. Architecture of RS encoder is discussed in brief in section III. Section IV includes RS decoding algorithms. Section V provides results of RS encoder. Conclusion is discussed in section VI.

Manuscript received February 2014.

Diplaxmi Sunil Chaudhari, *Electronics and Telecommunication, Pune University, Rajarshi Shahu College of Engg. Pune, India,+919028973464*

Mayura Ashok Bhujade, *Electronics and Telecommunication, Pune University, Rajarshi Shahu College of Engg. Pune,India,+919021802577*

Pranali Abhay Dhumal, *Electronics and Telecommunication, Pune University, Rajarshi Shahu College of Engg. Pune, India,+919595689264*

II. RS BASICS

The RS code is represented as RS (n,k) where n is code size in symbols, k is message size in symbols and 2t is number of parity symbols (n-k).

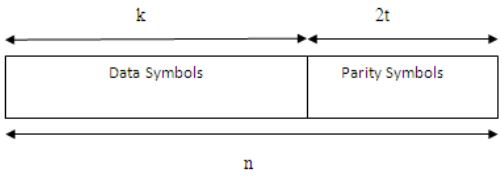

Figure 1: Structure of RS code word [2]

The relation between symbol size m, and code size n is given by

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

For RS(7,3) symbol size is m=3 and maximum correcting capability is t=2, given by

$$t = \frac{n-k}{2} \quad [5] \quad (2)$$

Firstly primitive polynomial f(X) is used to define Galois field element which is given as GF(2^m). An irreducible polynomial is said to be primitive, if the smallest positive integer n for which f(X) divides X^n+1 is $n=2^m - 1$. For (7,3) code we use primitive polynomial which is given as $1+X+X^3$.

ISSN: 2278 – 7798

III. RS ENCODER

A (7,3) cyclic code is specified by set of code word polynomials of degree 6 or less, which contains minimum degree of n-k i.e. 7-3=4 as a factor. This factor is denoted by g(X) which is called as generator polynomial of the code. That is the highest degree of generator polynomial is equal to number of parity bits in the code. The g(X) and the parity check polynomial h(X) are factors of $1+X^n$. A y factor of $1+X^n$ can be used as generator polynomial [8].

$$g(X) = g0 + g1X + g2X^2 + \cdots + g2t - 1X^{2t-1} + X^{2t}$$ [8]

For t=2, g(X) has 2t=n-k=4 roots.

$$g(X) = LCM[(X+\alpha)(X+\alpha^2)(X+\alpha^3)(X+\alpha^4)$$ [8]

By solving we get,

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

The message polynomial is

$$m(X) = \alpha X^2 + \alpha^3 X + \alpha^5 \quad (4)$$

Transmitted codeword is

$$c(X) = m(X)X^{2t} + m(X) mod\, g(X)$$ [5] (5)

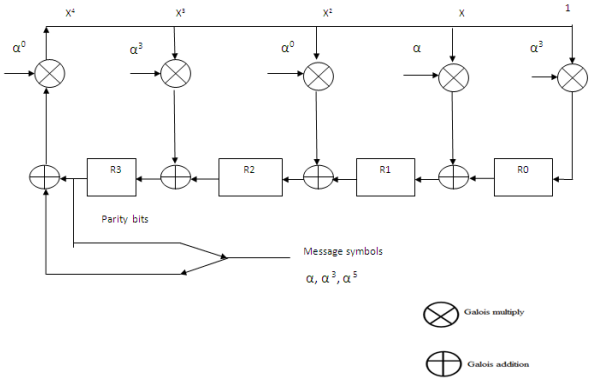

Figure 2: Architecture of RS Encoder

The encoder reads three message symbols, computes the parity symbols as 7-3=4 for total n symbols. The RS encoder consists of 2t linear feedback shift register [3] where each register is of 3 bits size. Generator polynomial coefficients are given to the multiplier coefficient. The coefficients produced will be symbols such that polynomials will exactly divide the parity polynomial. The process continues till all the 3 symbols of m(x) are given as input to the encoder. Thus during this time, the input switch is enabled while keeping the output i.e. parity switch disabled. For each clock cycle parity symbols are generated. After the last message symbol is given as input to the encoder and parity symbols are obtained, the output switch is enabled. In this way we obtain the parity symbols at the last clock cycle. Hence a new block can be started at the $(n+1)^{th}$ clock pulse.

Equations for shift registers:

$$R3 = \left(((m + R3)\alpha^0)\alpha^3\right) + R2 \quad (6)$$

$$R2 = \left(((m + R3)\alpha^0)\alpha^0\right) + R1 \quad (7)$$

$$R1 = \left(((m + R3)\alpha^0)\alpha^1\right) + R0 \quad (8)$$

$$R0 = \left(((m + R3)\alpha^0)\alpha^3\right) \quad (9)$$

Table 1: Contents of shift register

Msg Symbol	R3	R2	R1	R0
Initially	0	0	0	0
α	α^4	α	α^2	α^4
α^3	α^4	α^0	α^5	α^2
α^5	α	α^4	α^4	α^3

Transmitted symbol:

$$X^6\alpha + X^5\alpha^3 + X^4\alpha^5 \quad + \quad X^3\alpha + X^2\alpha^4 + X\alpha^4 + \alpha^3$$

Message symbols **parity symbols**

IV. RS DECODER

In communication system while transmitting input message codeword through channel noise may get added to it. Thus RS decoding [4], [5] includes detection and correction of errors at the receiver. Received codeword after being corrupted can be represented as

$$r(X) = c(X) + e(X) \quad (10)$$

where, e(x) is error polynomial with same degree as c(x) and r(x). The transmitted message c(x) is then recovered by adding received message, r(x) to error polynomial, e(x) as shown in equation 11

$$c(X) = r(X) + e(X) \quad (11)$$

RS decoding technique involves following steps:

1. Calculating the syndromes from the received codeword.
2. Computing the error locator polynomial.
3. Finding the error locations.
4. Computing error values.

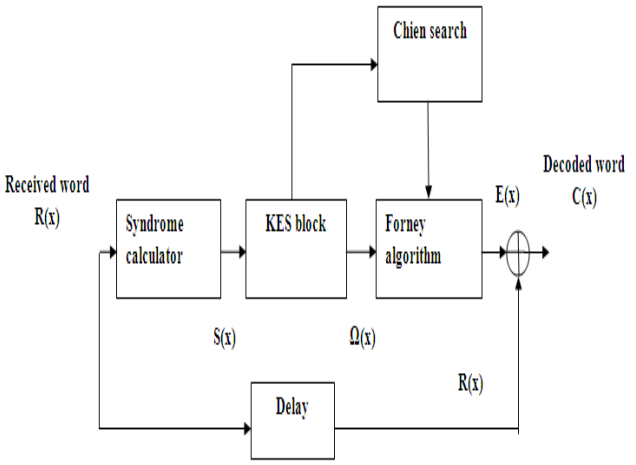

Figure 3: Architecture of RS decoder [3]

RS decoder consists of following blocks:

1) Syndrome calculator S(x) :

The syndrome is the result of a parity check performed on received polynomial to determine whether received codeword is a valid member of codeword set. Syndrome values will be calculated as follows

$$S_i = [R(X)]_{x=\alpha^i} \quad (12)$$

S(x)=0 indicates, there is no error in received codeword and if S(x)≠0 then there is error in the received codeword.

2) Key Equation Solver (KES):

This is the main block of RS decoder which solves a set of 2t linearly dependent equations. Two key equations i.e. error locator polynomial σ(x) and error evaluator polynomial Ω(x) are generated from the syndrome polynomial. By solving equation 13, we can determine above two unknown polynomials σ(x) and Ω(x).

$$S(X) * \sigma(X) = \Omega(X) mod X^{2t} \quad (13)$$

The two techniques to solve key equations are Berlekamp Massey algorithm [6] and Euclidean algorithm. We have used Berlekamp Massey algorithm as it has least hardware complexity as compared to Euclidean algorithm.

3) Forney algorithm:

Forney algorithm calculates error values e_i by using error locator polynomial σ(x) and error magnitude polynomial Ω(x).

4) Chien search:

This block is used to find the roots of σ(x) which are reciprocals of error locations. When Chien sum is zero then there is error in that particular location. In this way location of error can be computed easily using Chien search.

V. Synthesis Results

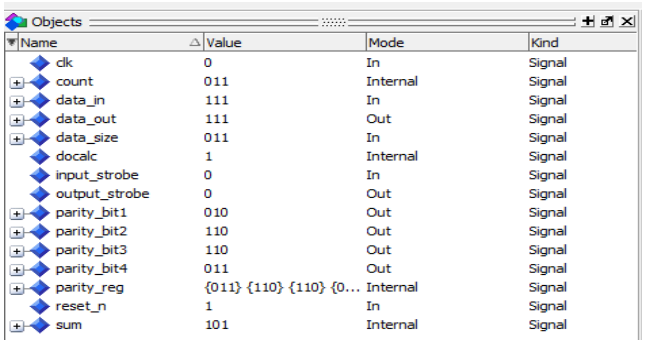

Figure 4: Objects of RS Encoder

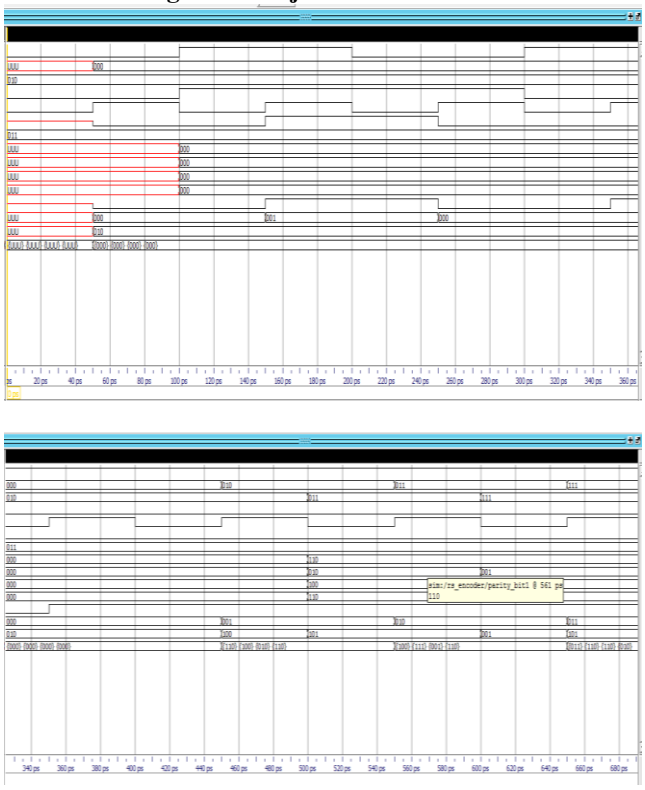

Figure 5: Synthesis result of RS Encoder

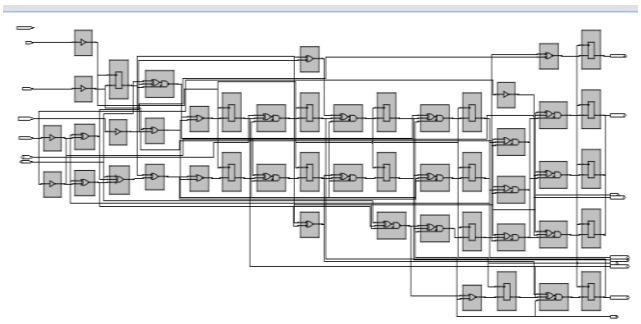

Figure 6: Gate level schematic of RS Encoder

VI. Hardware Implementation

ProASIC3 is the third generation family of Microsemi FPGAs. ProASIC3 has non-volatile flash technology with low power, secure and single chip solution. The .pdb file generated by Actel libero software is dumped into ProASIC3 A3P250 208FQGA device using FlashPro programming software and verified parity bits on Led's given on the FPGA

ISSN: 2278 – 7798

board [9].

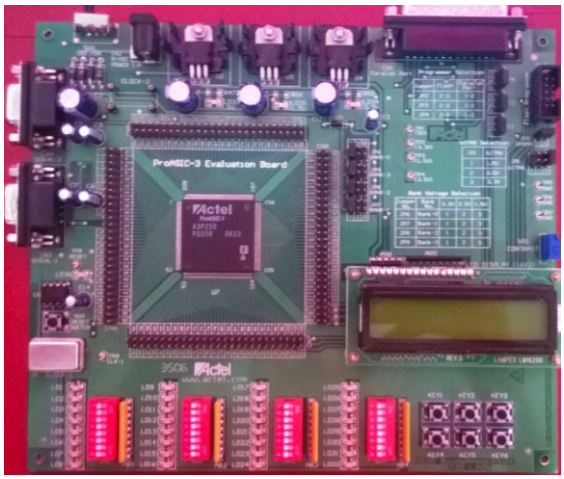

Figure 7: ProASIC3 FPGA kit

VII. CONCLUSION

In this paper, design of RS (7,3) encoder and decoder and its implementation on Actel ProASIC kit is analyzed. We have verified the parity symbols mathematically using Linear Feedback Shift Register (LFSR) and the synthesized results obtained on Libero software. At the encoder, we have successfully obtained the parity symbols for given message symbols by dumping code into FPGA kit. Thus we have performed decoding using Berlekamp-Massey algorithm. All the results are simulated using Actel Libero software.

ACKNOWLEDGMENT

At the first we would like to thank our guide Prof.S.C.Wagaj for giving us opportunity and support to work on this project. We express sincere thanks to our friends who helped us directly and indirectly for this work.

.

REFERENCES

[1] I. S. Reed and G.Solomon, "Polynomial Codes Over Certain Finite Fields", *Journal of the Society of Industrial and Applied Mathematics*, pp.300-304, 1960 printed in U.S.A.

[2] Aqib Al Azad and Md Imam Shahed, "A Compact and Fast FPGA Based Implementation of Encoding and Decoding Algorithm Using Reed Solomon Codes", *International Journal of Future Computer and Communication*, vol.2, no. 6, pp.- 31-35, February 2014.

[3] Mustapha Elharoussi, Asmaa Hamyani and Mostafa Belkasmi, "VHDL Design and FPGA Implementation of a Parallel Reed-Solomon(15,K,D) Encoder/Decoder", *International Journal of Advanced Computer Science and Applications(IJACSA)*, Vol. 4, No. 1, 2013.

[4] Bhawna Tiwari and Rajesh Mehera, "Design and Implementation of Reed Solomon Decoder for 802.16 Network using FPGA", *IEEE Signal Processing, Computing and Control (ISPCC)* , Date of Conference**:** 15-17 March 2012, Conference Location :Waknaghat Solan.

[5] Kenny Chung Chung Wai & Dr. Shanchieh Jay Yang, "Field Programmable Gate Array Implementation of Reed-Solomon Code, RS(255,239)" , Date of Conference: 2010, Conference Location: Chicago.

[6] I.S. Reed, M.T. Shih, T, K. Truong, "VLSI design of inverse free Berlekamp-Massey algorithm", *Proceeding of Computers and Digital Techniques*, Vol. 138, pp.295-298, 1991.

[7] Aqib Al Azad, Minhazul Huq, Iqbalur. Rahman Rokon, "Efficient Hardware Implementation of Reed Solomon Encoder and Decoder in FPGA using Verilog", *International Journal of Advancements in Electronics And Power Engineering(ICAEPE'2011)*, Bangkok, Dec.2011.

[8] Petrus Mursanto, "Generic Reed Solomon Encoder", *Makara, Sains*, Vol. 10, No.2, pp. 58-62, November 2006.

[9] " Microsemi ProASICs flash family FPGAs datasheet", Revision 13.

Diplaxmi Chaudhari was born in Pune, India on 12th June 1993. She is currently pursuing the under graduate course Bachelors of Engineering in Electronics and telecommunication at Rajarshi Shahu College of Engineering.

Mayura Bhujade was born in Pune, India on 10th December 1992. She is currently pursuing the under graduate course Bachelors of Engineering in Electronics and telecommunication at Rajarshi Shahu College of Engineering.

Pranali Dhumal was born in Pune, India on 17th May 1993. She is currently pursuing the under graduate course Bachelors of Engineering in Electronics and telecommunication at Rajarshi Shahu College of Engineering.