ISSN: 2278 – 909X
International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE)
Volume 5, Issue 4, April 2016

VHDL IMPLEMENTATION OF REED SOLOMON (32, 28) ENCODER AND DECODER

Upasana Roy Chaudhary, Shafali Jagga

***Abstract*— Reed Solomon codes are block based codes which are non- binary, cyclic, systematic and linear in nature. These are used to detect and correct burst errors. In this paper, implementation of RS (32, 28) encoder and decoder has been discussed.RS Encoder is implemented using galois field multipliers .At the decoder, syndromes of received codeword is calculated using generator polynomial to detect number of errors. Euclid's algorithm is used to create an error locator polynomial, Chien search algorithm and Forney algorithm is used to correct the errors.**
Reed Solomon encoder and decoder is synthesized using VHDL on Xilinx 14.6 ISE Design Suite.

***Index Terms*—Decoder, Encoder, Reed Solomon codes, VHDL.**

I. INTRODUCTION

Reed Solomon codes are linear block error correcting codes [1]. Cyclic codes were conferred in a successions of thesis and notes written by E. Prange [2],[3],[4] amid 1957 to 1959.Eventually this led to research work prepared by R.C. Bose and D.K. Ray Chaudhari in 1960 when they discovered BCH codes [5],[6],[7]. I.S. Reed and G. Solomon has illustrated a new class of error correcting codes named Reed Solomon codes[8].In the beginning Reed -Solomon codes were encoded and decoded by the use of finite field arithmetic [9],[10].

These codes are used in correcting burst errors and have found wide ranging applications throughout the fields of digital communication and storage. [1]
Some of which include:

- Storage Devices (hard disks, compact disks, DVD, Barcodes)
- Wireless Communication (mobile phones, microwave links)
- Digital Television
- Broadband Modems (ADSL, xDSL, etc).
- Deep Space and Satellite Communications Networks (CCSDS).

***Upasana Roy chaudhary**, Department of ECE, A.P.J. Abdul Kalam University/ IPEC, Ghaziabad ., Ghaziabad , U.P. , India, Mobile No 8860478669*
***Shafali Jagga**, Asst Professor, IPEC Ghaziabad ,India ,Mobile No 9560709212 .*

A Reed Solomon code is a block code which is designated as RS (*n*, *k*) as shown in the fig.1 below

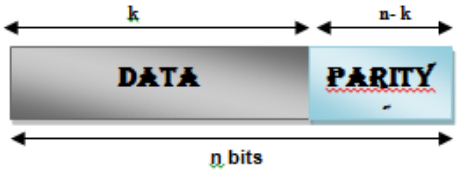

Fig.1 Structure of a RS codeword

- *m* = number of bits per symbol
- *n* = code length in symbols (up to 2^m-*1*)
- *k* = original message length in symbols
- *r* = n-k = number of check symbols
- *t = 1/2 (n-k)* = error correction capability

There is an vital relationship between m and n is

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

II. GENERATOR POLYNOMIAL

Reed Solomon codes are constructed using a special class of polynomial denoted as *g(x)*.Each valid codeword is divisible by *g(x)*. The general form of generator polynomial [11] is

$$g(x) = (x+\alpha^i)(x+\alpha^{i+1})\ldots\ldots\ldots(x+\alpha^{i+2t-2})(x+\alpha^{i+2t-1}) \quad (2)$$

For the RS (32, 28) generator polynomial is

$$g(x)= x^4 + g_3 x^3 + g_2 x^2 + g_1 x^1 + g_0 \quad (3)$$

To calculate the codeword C(x) refer to (4)

$$C(x) = m(x) * 2t + m(x) \bmod g(x) \quad (4)$$

where *m(x)* is the block of information as the incoming input.

III. RS (32, 28) ENCODER ARCHITECTURE

Reed Solomon Encoder takes in k bits as input, evaluates n-k parity bits and attaches these *n- k* bits to the *k* bits or symbols to construct a total of *n* bits. The *2t* parity symbols in a systematic reed Solomon code is given in (5)

ISSN: 2278 – 909X
International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE)
Volume 5, Issue 4, April 2016

$p(x) = m(x) * x^{n-k} \mod g(x)$
(5)

Construction of codeword is done by multiplying *m(x)* with x^{n-k}, dividing the product by *g(x)* such that remainder is evaluated and compensating the remainder by subtracting it as shown in (6).

$C(x) = m(x) * x^{n-k} - p(x)$
(6)

By substituting the value of *p (x)* from above we reach to the conclusion of calculating the *C (x)* as shown in (7)

$C(x) = q(x) * g(x)$
(7)

Here the quotient polynomial (8) is

$q(x) = \frac{m(x)}{g(x)} x^{n-k}$
(8)

RS encoder design should effectively perform the following two operations, namely division and shifting. Both operations can be easily implemented using *Linear-Feedback Shift Registers.*
The following is the diagram of architecture of systematic RS (32, 28) coder in fig.2. Each of the 4 registers holds a symbol (8 bits). The arithmetic operators carry out finite field addition and multiplication on a complete symbol.

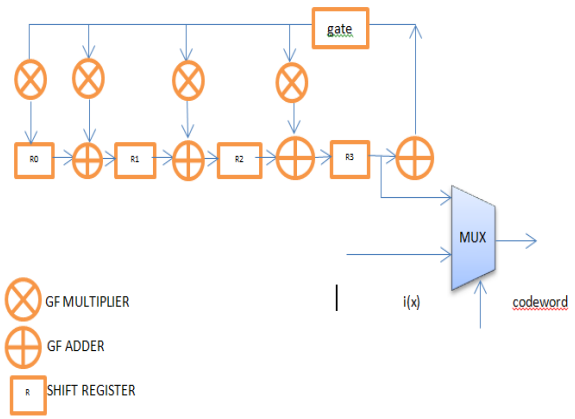

Fig. 2 Structure of RS (32, 28) encoder

IV. RS (32, 28) DECODER ARCHITECTURE

The decoder corrects errors which may be introduced during transmission and thus received codeword *r (x)* is given below (9)

$r(x) = c(x) + e(x)$
(9)

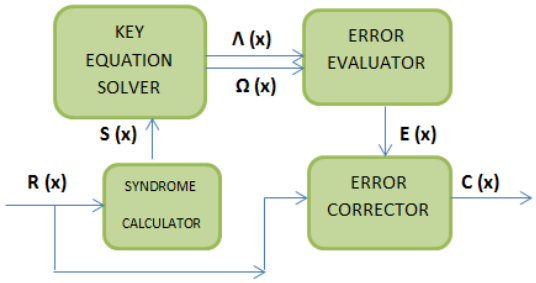

Fig. 3 Structure of RS (32, 28) decoder

A. SYNDROME CALCULATION

Here the input received symbols are divided by the generator polynomial. The result should be zero. The parity is placed in the codeword to ensure that code is exactly divisible by the generator polynomial. If there is a remainder, then there are errors. The remainder is called the syndrome. The syndrome polynomial *S(x)* which is given in (9)

$S(x) = S_0 + S_1 x + \ldots S_{2t-1} x^{2t-1} = \sum_{j=0}^{n-1} R_j * \alpha^{tj}$
(9)

where α is a primitive element

B. EUCLID'S ALGORITHM

This is a recursive method [14] of evaluating the GCD of two polynomials satisfying the following equation given in (10)

$\text{GCD}[S(x), t(x)] = a(x) * s(x) + b(x) * t(x)$
(10)

C. CHIEN SEARCH ALGORITHM

Once the error locator $\Lambda(x)$ and error evaluator $\Omega(x)$ polynomials have been determined the next step in the decoding process is to evaluate the error polynomial and obtain its roots. The roots thus obtained over a finite field using chien search algorithm [12]

D. FORNEY ALGORITHM

It evaluates the error values at known error locations. It is an efficient way of performing a matrix inversion [13], and involves two main stages.

- Calculate the error evaluator polynomial as shown in (11)

$\Omega(x) = S(x)\Lambda(x) \pmod{x^{2t}}$
(11)

- Evaluate error values

$e_j = -[\Omega(X_j^{-1}) / \Lambda'(X_j^{-1})]$
(12)

ISSN: 2278 – 909X
International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE)
Volume 5, Issue 4, April 2016

V. PROPOSED ENCODING AND DECODING SYSTEM AND ITS IMPLEMENTATION

Fig. 4 shows a block diagram of typical RS (32, 28) transmission system.

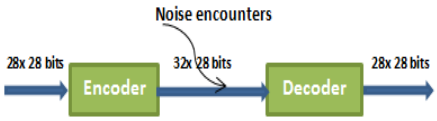

Fig.4 RS (32, 28) Transmission System

RS (32, 28) is written in VHDL on Xilinx 14.6 and simulated in ISE simulator.

A. ENCODER IMPLEMENTATION

Reed-Solomon encoder for RS (32, 28) is implemented in VHDL to encode the data symbols for reliable communication. The register transfer level (RTL) for this code is shown in following fig. 5 and fig. 6 respectively.

Top level block of RS (32, 28) encoder entity is shown in fig. 5

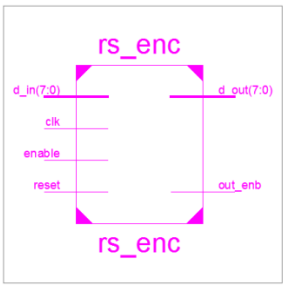

Fig. 5 Top Level Block of entity of RS (32, 28) Encoder

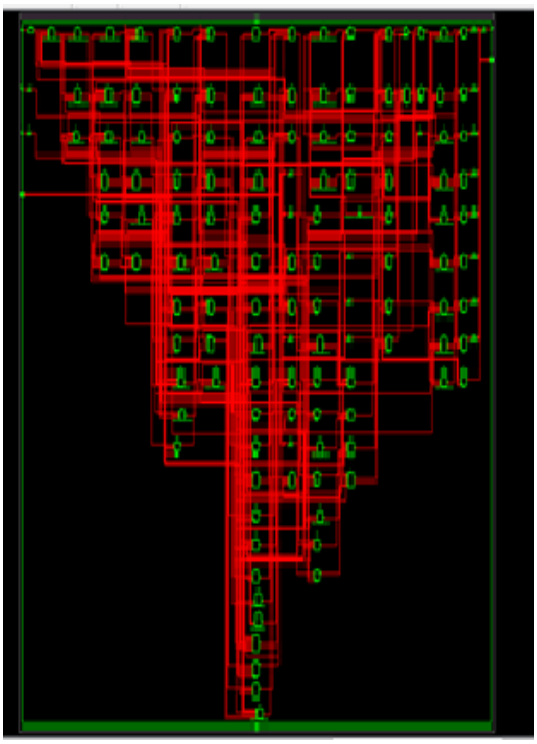

Fig. 6 RTL view of RS (32, 28) Encoder

Simulation result of RS (32, 28) Encoder is shown in fig. 7 below,

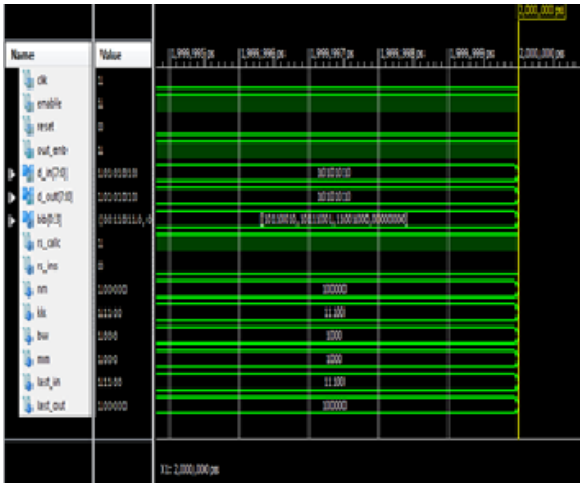

Fig. 7 Simulation result of Entity RS (32, 28) Encoder

B. DECODER IMPLEMENTATION

Reed-Solomon Decoder for RS (32, 28) is implemented in VHDL on Xilinx 14.6 successfully.

RTL of top level block of RS (32, 28) decoder is shown in fig 8.

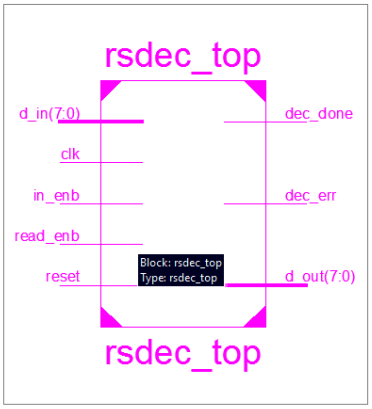

Fig. 8 Top Level Block of RS (32, 28) decoder

Internal architecture of RS (32, 28) decoder is shown in fig. 9

Fig.9 RTL view of RS (32, 28) decoder

Simulation result of RS (32, 28) Decoder is done on Xilinx 14.6 shown in fig.10

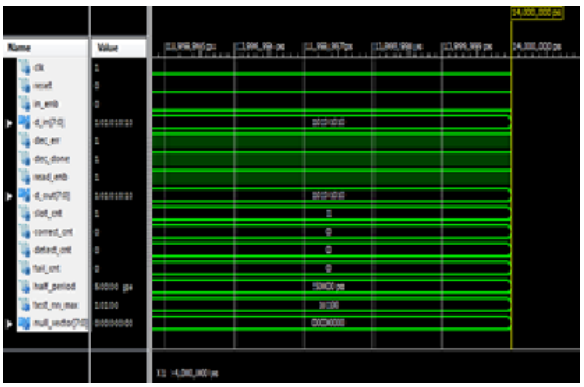

Fig. 10 Simulation result of Entity RS (32, 28) decoder

VI. CONCLUSION

This paper implements RS (32, 28) encoder and decoder where m=8 bits, $2t = n\text{-}k = 4$ hence error correction capability't' = 2. The generator polynomial taken here *is* $g(x) = x^4 + g_3 x^3 + g_2 x^2 + g_1 x^1 + g_0$. For decoding Syndrome computation, Euclid's algorithm, Chien search algorithm and Forney algorithm is applied and hence decoding is done successfully.

VII. REFERENCES

[1] C.E. Shannon, "A Mathematical Theory Of Communication ", *Bell System Technology Journal*, volume 27, pp. 379-423, 623-656, 1948.S
[2] E. Prange, "Cyclic Error-Correcting Codes in Two Symbols," *Air Force Cambridge Research Center-TN-57-103*, Cambridge, Mass., September 1957.
[3] E. Prange, "Some Cyclic Error-Correcting Codes with Simple Decoding Algorithms," *Air Force Cambridge Research Center-TN-58-156*, Cambridge, Mass., April 1958.
[4] E. Prange, "The Use of Coset Equivalence in the Analysis and Decoding of Group Codes," *Air Force Cambridge Research Center-TR-59-164*, Cambridge, Mass., 1959.
[5] R. C. Bose and D. K. Ray-Chaudhuri, "On a Class of Error Correcting Binary Group Codes," *Information and Control*, Volume 3, pp. 68-79, March 1960.
[6] R. C. Bose and D. K. Ray-Chaudhuri, "Further Results on Error Correcting Binary Group Codes," *Information and Control*, Volume 3, pp. 279-290, September 1960.
[7] A. Hocquenghem , "Codes correcteurs d'erreurs," *Chiffres*, Volume 2, pp. 147-156, 1959
[8] I. S. Reed and G. Solomon , "Polynomial Codes over Certain Finite Fields," *SI AM Journal of Applied Mathematics*, Volume 8, pp. 300-304, 1960.
[9] R. J. McEliece, Finite Fields for Computer Scientists and Engineers, Boston: Kluwer Academic, 1987
[10] S. B. Wicker, Error Control Systems for Digital Communication and Storage, N.J.: Prentice-Hall,1994.
[11] .Gorenstein and N. Zierler, "A Class of Error Correcting Codes in pm Symbols," *Journal of the Society of Industrial and Applied Mathematics*, Volume 9, pp. 207-214, June 1961.
[12] R. T. Chien, "Cyclic Decoding Procedure for the Bose- Chaudhuri-Hocquenghem Codes," *IEEE Transactions on Information Theory*, Volume IT-10, pp. 357-363, October 1964.
[13] G. D. Forney, "On Decoding BCH Codes " *IEEE Transactions on Information Theory*, Volume IT-11, pp. 549-557, October 1965.
[14] Y. Sugiyama, Y. Kasahara, S. Hirasawa, and T. Namekawa, "A Method for Solving Key Equation for Goppa Codes," *Information and Control*, Volume 27, pp. 87-99, 1975.