ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Implementation of Reed Solomon Decoder for Area Critical Applications

Mrs. G.Srivani
M.Tech Student
Department of ECE,
PBR Visvodaya Institute of Technology & Science,
Kavali.

Mr.R.S.Pratap Singh
Associate Professor
Department of ECE,
PBR Visvodaya Institute of Technology & Science,
Kavali.

Abstract:
In recent years the continued rise of portable data-devices such as cell phones, PDAs, and laptops has driven enormous growth in the area of wireless communications. Whenever data is sent over a wireless channel, it is subject to degradation due to multipath fading and noise. Depending on the amount of degradation, the effect can be a loss or corruption of the original data during transfer. In order to alleviate this problem and ensure the reliable transfer of data, the typical solution has been the use of an error correction coding scheme. This work will detail the implementation of a error correction coding (ECC) scheme, based on the widely used Reed-Solomon algorithm, which will be implemented in wireless communication and digital signal processing (DSP) applications due to their ease of use in high performance characteristics, and inherent configurability. The main aim of this project is to design an RS decoder which should transmit the data without having any errors by maintaining the efficiency η is 5. A new syndrome computation is proposed in the project. A detailed analysis of results is presented and the functionality is verified using ISE simulator and synthesized using XILINX ISE 12.3i with VERILOG HDL.

Keywords: ***Reed-Solomon Algorithm, Error Correction Coding, XILINX ISE, Verilog.***

I. INTRODUCTION

A Reed-Solomon (RS) code is an error-correcting code since that time they've been applied in CD-ROMs, wireless communications, space communications, DSL, DVD and digital TV. Generally encoding the data is relatively straightforward, but decoding is time-Consuming process. Only in the past few years has it become computationally possible to send high-bandwidth data using RS due to its numerous advantages. A digit is error-free if and only if all of its bits are error-free. The main aim of this project is to design an RS decoder which should transmit the data without having any errors by maintaining the η is 5.Reed Solomon codes are used to detect the errors in many applications. so, such applications are for the use of many different reed Solomon codes. Reed Solomon codes are used widely in data storage systems and communication systems for the purpose of resolve the data from possible errors which are occur during transmission process and from the disc errors respectively. One of the applications of reed Solomon codes are FEC (forward error correction) which is shown in fig.1. The encoder attaches the parity symbol before transmission of data to the data using a predetermined algorithm. During the transmission a limited predetermined number of errors are corrects and detects by the decoder. The extra parity symbols transmitting require extra bandwidth when compared to the pure data of transmitting. The additional symbols of the transmitting is introduced by the FEC it is better technique than retransmitting the whole package it is used to detect the errors by using the receiver. An efficient digital communication system provides cost effective which gives level of reliability.

Through the channel the information is transmitted to the receiver it is a prone of errors. This unit deals an error correcting techniques. This process is done by the reluctant symbols at the transmitter side. Two basic objects are there with this coding they are detection and correction.

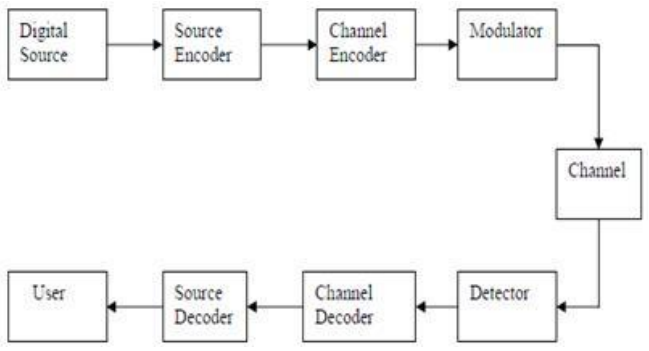


Fig.1. Block Diagram of a Digital Communication System.

By transmitting or stored it is affected by the interference which is used to distort the message symbol. The noise is influenced by the radio transmission or by any other transmitter. Apart from the noise, different types of storage systems they occurred by interference which are used to caused the damage sin the storage medium. To reduce the interference they are solve in several ways. To remove the errors their one way that is to design the message that the receiver can detect the errors which are occurred or evenly possible errors too. This whole process is achieved by error-correcting codes. By doing this in encoded message the number of symbols are increased by doing this redundancy is introduced. After detecting the errors the correction is obtained by the as shown in below

(1) Repeated incorrect code word asking for the transmission from the receiver.

(2)For the correction of the errors this method is used.

Hence it requires a transmission rate and high number of check bits, At a certain time and certain minimum probability the information is transmitted. The reverse is also true; a certain possible transmission rate is offered by the channel than FEC the ARQ permits a higher transmission rate. It is applicable only when the channel has the lower error rate. Depending up on the properties of the system or application the error-correction is to be introduced. The advantage of FEC is the transmission is never completely blocked even the channel quality falls below low level that ARQ have completely asked for retransmission. The received data do whatever it can fix or declare an alarm.

One of them symbol stream is divided in to block and coded. Hence it is called block coding. Other one is convolution operation is applied to the symbol stream. In the other one the symbol stream an operation of the convolution is applied. Hence this process is called conventional coding as shown in Fig.2.

Using the techniques the signals are enhanced and the performance is improved by the FEC.

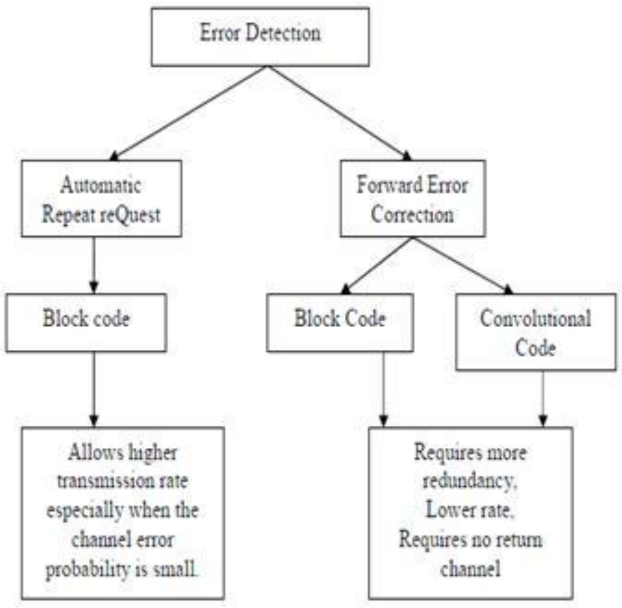

Error Detection
Automatic Repeat reQuest
Forward Error Correction
Block code
Block Code
Convolutional Code
Allows higher transmission rate especially when the channel error probability is small.
Requires more redundancy. Lower rate. Requires no return channel

Fig.2.Classifications of error detection& correction schemes.

II. ARCHITECTURE DETAILS

Reed Solomon codes are called as symmetric codes. The code word of a information symbols are placed as a higher coefficients. The information symbol requires the symbols are shifted power level to (n-1) (n-k) and reaming power is positioned from (n-k-1) to 0and they are filled with zeros. For any reed Solomon codes will perform these two operations. The operations are called as shifting and division arithmetic operation. By using these operations are easily implemented by the Linear- Feedback shift registers. At the encoder the data symbols are zero by shortened by the reed Solomon codes. These symbols are re-inserting at the decoder .the encoder consisting of the 2t shift registers each register consisting of m bit wide. Reed Solomon generator polynomials are the multiplier coefficients.

The common idea is polynomial construction. The general idea is the construction of a polynomial. At the encoder block the diagram shows that at the input each constant field element is a multiplier. With the polynomial of g(x).the message polynomial is given to the encoder by symbol by symbol. After a desired latency the symbol appears at the output. To produce the related parity through an adder feeds it back by the control logic. These whole processes continue until all k symbols are input to the encoder.

During this process the input data path enables at the output of the control logic. This process is done when the parity path disabled. At each clock cycle contains with output latency. At the (k+1)th clock pulse symbol at this time the encoder encodes the output. Also, during the first k clock cycles, the feedback control logic feeds the adder output to the bus as shown in Fig.3.

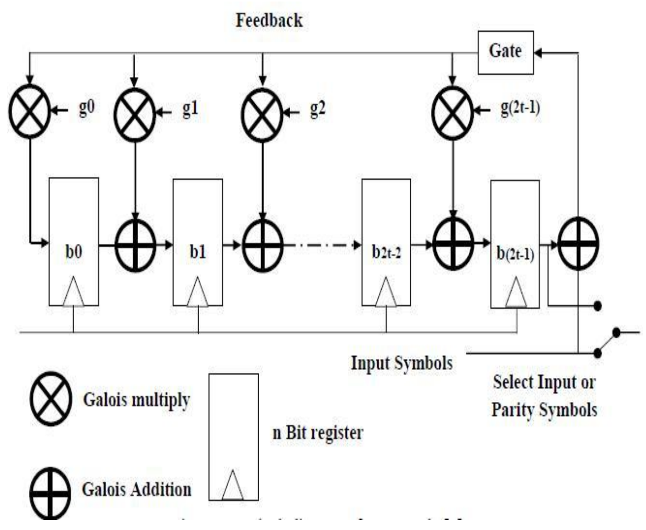


Fig.3. RS Encoder.

A. Solving the Error Locator Polynomial-CHIEN Search

In the above techniques they are error locator and error evaluator are determined the next stage is the decoding process. The decoding process is used to evaluate the error location in the received message. Also this process evaluates the roots of the polynomials. The chien search scheme in the reed Solomon decoding is used to implement the same process. The roots of the polynomial is said to be „n‟. Where n is said to be number. The result of the value in the polynomial evaluates to the zero. In the Galois the elements are adopts by the direct substitution process, until a specific i from i=0, 1... (n-1) is found such that σ (αi) = 0. In such a case αi is said to be the root and the location of the error is evaluated as σ(x).By this the number becomes zero. In the case of the error is found the error vector is updated and the evaluated process is done in all symbols.

III. PROPOSED WORK

The primitive polynomial of the Galois field is generally represent α the values are defined as shown here (0, 1,α ,α2,…α2k-2).the simple example for the finite field is the binary field the binary field as 0,1.in the form of Galois they are defined as the GF(2).this process is done by two method they are addition and multiplication by doing this operation we may reduce the module. The another example is by taing the large fields vector space leads to finite field of size 2m. These extensions are based upon the m dimensions. The Galois GF (2m) these are defined as the m as the tuple representation. By this we may defined as the m bit is referred as the GF (2m). This in turn allows applying the associated mathematical operations of the field to encode and decode data. Let the primitive polynomial be φ (x), of degree m over GF (2m). Now any ith element of the field is given by,

$$a_t(\alpha) = a_{t0} + a_{t0}\,\alpha + a_{t0} + a_{t0}\,\alpha^2 + \ldots\ldots + a_{tm-1}\,\alpha^{m-1} \quad (1)$$

So, all these elements can generator by the power of α. The representation of the finite field elements are shown below Fig.4, and also assume the leading coefficient of φ (x) to be equal to1. Table 1 shows the Finite field generated by the primitive polynomial, 1 +α +α2 +α3 +α4 +α8 represented as GF (2^8) or GF (256)

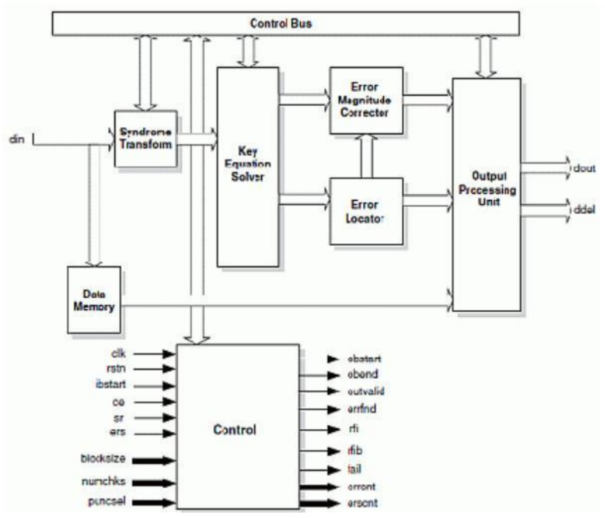


Fig.4. Proposed architecture for RS Decoder.

Initially the data din signal is applied to syndrome transform to find the error presence of the applied data. It should under goes through the KES block which should control through the controller. The same output passes through the error locator and error corrector block which should be processing through the processing unit. For storing the data a memory block is used.

A. Syndrome Calculation

The first step in decoding the received symbol is to determine the data syndrome. 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 syndromes can then be calculated by substituting the 2t roots of the generator polynomial g(x) into R(x). The syndrome polynomial is generally represented as,

$$S(x) = S_0 + S_1 x + \cdots + S_{2t-1} x^{2t-1} = \sum_{j=0}^{n-1} r_j \alpha^{tj} \qquad (2)$$

Where, α is the primitive element. The basic syndrome calculation and update architecture is shown in figs.5 and 6.

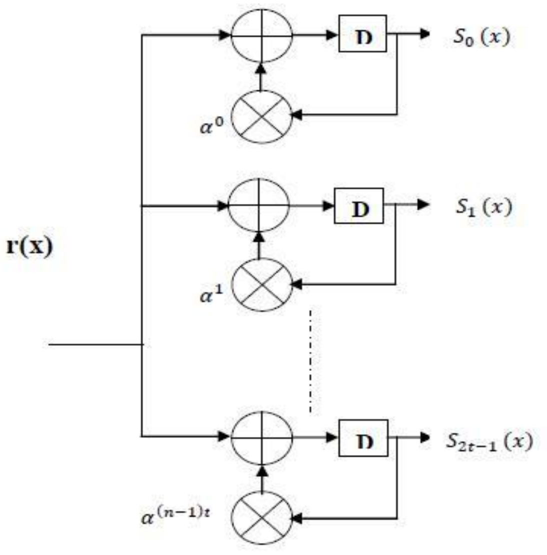
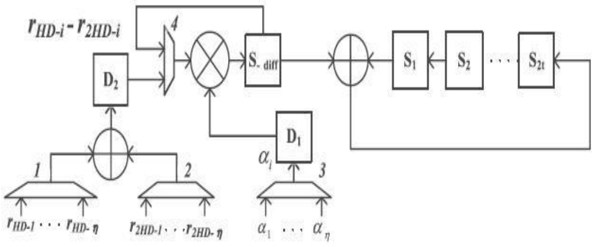


Fig.5. Syndrome calculator Architecture.


$r_{HD-i} - r_{2HD-i}$
4
D_2
S. diff
S_1
S_2
S_{2t}
D_1
α_i
1
2
3
$r_{HD-1} \cdots r_{HD-\eta}$
$r_{2HD-1} \cdots r_{2HD-\eta}$
$\alpha_1 \cdots \alpha_\eta$
Fig.6. Syndrome update architecture.

B. KES AND Polynomial Selection

Different from the iBM employed in [12], the KES in the RiBM in [13] is selected by the decoder presented in this brief. The RiBM algorithm introduces the (x) and _(x), where (x) = δ3t x3t +δ3t−1x3t−1 + • • • + δ1x + δ0 and _(x) = θ3t x3t + θ3t−1x3t−1 +• • •+θ1x +θ0. Both (x) and _(x) load the S j (x), then the RiBM simultaneously updates these two polynomials. After 2t iterations, the error locator polynomial σ(x) and error evaluator polynomial ω(x) can be obtained from the coefficients of the δ(x) polynomial. Key architecture of the RiBM is given in Fig. 7. The advantage of the RiBM is that each step in the KES algorithm of the RiBM can be implemented simultaneously. Consequently, there is no free clock period for each unit in the KES module and no KES architecture needs to be shared as in [12]. Although the RiBM requires more registers and multipliers than the iBM, it provides higher throughput which increase the decoding speed of the whole decoder. Moreover, the KES can match the speed of the syndrome update perfectly.

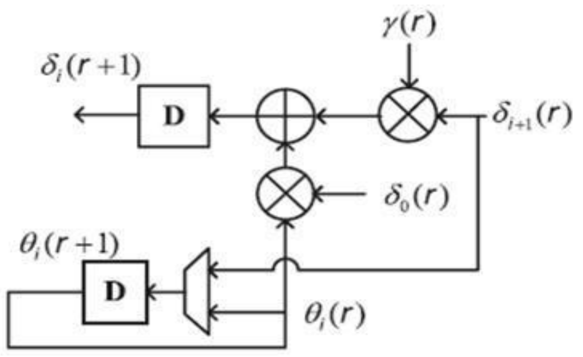


Fig.7. KES architecture in RiBM.

IV. RS DECODER ALGORITHM A. Algorithm For RS Decoder

To correct errors and erasures by calculating each syndromes for each single codeword at reed Solomon decoders. If the errors are present in the message symbol the error location are found by the Berlekamp-Massey algorithm. These algorithms create an error locator polynomial. By using the Chien search the roots of this polynomial are found. Another step in decoder is Forney‟s algorithm.

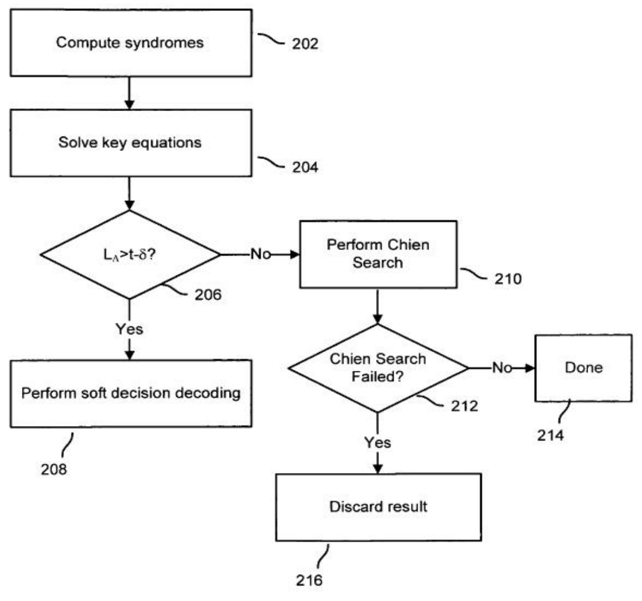


Fig. 8. Algorithm for RS Decoder.

Initially the syndrome has to be calculated first. The corresponding syndrome is applied to KES block to find out the efficiency .If it is greater than the length value then it performs soft decision decoding otherwise it will perform chien search. If the chien search fails then the result has to be discarded else the loop has to be repeated.

B. Euclid's Algorithm

For calculating the greatest common divisor we are using this algorithm .by this algorithm it produce recursive operation for two polynomials. This algorithm produces the two polynomials which is shown below, a(x) and b(x) these are two polynomials which are satisfying the above polynomials,

$$GCD[s(x), t(x)] = a(x)s(x) + b(x)t(x)$$

```
1 procedure EUCLID($a, b$)
2     $r \leftarrow a \bmod b$
3     while $r \neq 0$ do
4         $a \leftarrow b$
5         $b \leftarrow r$
6         $r \leftarrow a \bmod b$
7     end while
8     return $b$
9 end procedure
```

C. Error Value Computation-Forney Algorithm

After the errors are located the next stage is to drive the error values we use the error polynomials and the syndromes. For this purpose we use these algorithms.. This process is done by the two stages they are shown below. The first stage of this algorithm is the error evaluator polynomial(x) and is calculated. This calculation process is done by the convolving the syndromes with the error polynomial σ(x).At the location of the zero this calculation is carried out. The error symbol is arriving by the each calculation. The error magnitude at each error location xt is given by,

$$e_t = \frac{w(\alpha^t)}{\sigma^t(\alpha^t)} \quad (1)$$

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

If an error symbol is consisting of the set of bits. I.e; the corresponding bit is error in the received symbol. And they are must be inverted by using the read gain the automated symbol.

```
For $i = 1$ to $n$
If$(\Lambda(\alpha^{i}) == 0)$then
    $\hat{c}_{n-i} = r_{n-i} + \frac{\Omega(\alpha^{i})}{\Lambda 1(\alpha^{i})}$
End If
End For
```

In summary, the decoding algorithm works as follows:
Step 1: According to the equation the syndromes are calculated.
Step2: To obtain the error locator the two algorithms are used they are Berlekmap-massey or Euclid‟s algorithms and also used for the polynomial evaluator w(x).
Step 3: Perform the Chine Search to find the roots of σ (x).
Step 4: By using the Forney‟s we may find the magnitude.
Step 5: The correct word C (x) = E(x) + R (x)

The Reed-Solomon decoder tries to correct errors and/or erasures by calculating the syndromes for each codeword. Based upon the syndromes the decoder is able to determine the number of errors in the received block [1][15][17]. If there are errors present, the decoder tries to find the locations of the errors using the Berlekamp-Massey algorithm by creating an error locator polynomial. The roots of this polynomial are found using the Chien search algorithm. Using Forney's algorithm, the symbol error values are found and corrected. For an RS (n, k) code where n - k = 2T, the decoder can correct up to T symbol errors in the code word. Given that errors may only be corrected in units of single symbols (typically 8 data bits), Reed-Solomon coders work best for correcting burst errors [11]. After going through a noisy transmission channel, the encoded data can be represented as r(x) = c(x) + e(x), where e(x) represents the error polynomial with the same degree as c(x) and r(x). Once the decoder evaluates e(x), the transmitted message, c(x), is then recovered by adding the received message, r(x), to the error polynomial, e(x), as given below:

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

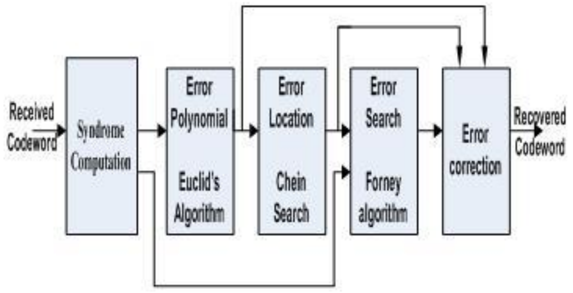


Fig.9. Proposed diagram for RS Decoder.

Note that e(x) + e(x) = 0 because addition in Galois field is equivalent to an exclusive-OR i.e e(x) XOR e(x) = 0. A typical decoder follows the following stages as shown in figure 4.2 in the decoding cycle, namely Syndrome Calculation, Determine error-location polynomial, solving the error locator polynomial - Chien search, calculating the error Magnitude, Error Correction. The details of Low complexity chase Reed-Solomon decoder flow by reduced-complexity LCC decoding based on the unified syndrome computation (USC) algorithm is proposed in which the syndromes of one test vector can be obtained from the syndromes of previous test vector, and an efficient architecture of the USC is also given. The KES in the reduced iBM (RiBM) [13] algorithm is adopted to match the speed of the USC, and a modified polynomial selection architecture is employed to choose the right error locator polynomial. As the result, the decoding latency and the area are reduced to 64% and 62% for η = 3, respectively, which leads to 2.5 times speed over area ratio than the decoder in [12]. Further analysis shows the area of the proposed decoder is 69% of the decoder based on RCMI [14] for η = 3, with 13% speed up.

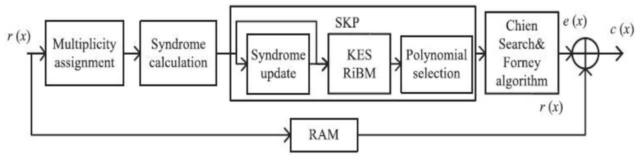


Fig.10. Existed Diagram for RS Decoder.

Reed-Solomon codec for RS (255,251) which contains code length of 255 symbols with 251 symbols of data and remaining 4 symbols of parity. It can correct 2 symbols of data in the data length of 251 symbols. The following figure describes the development and implementation of test bench for reed Solomon coder. Two modules are implemented in the code for encoder which can take stream of 251 symbols of data and add the 4 parity bits as extra symbols and decoder which can take the encoded data at the receiver end to decode and correct the errors if any found in the received stream of data. Different signals are routed for the controlling the functionality of the encoder and decoder. STR is one signal connected from test bench to encoder module to indicate the start of encoding. RD signal is asserted, when the stream of data is encoded and ready to read from encoder module. SNB is output from encoder module that indicates the end of encoding. So, by checking the SNB signal, RD signal is enabled, once the encoding is completed.

V. RESULTS

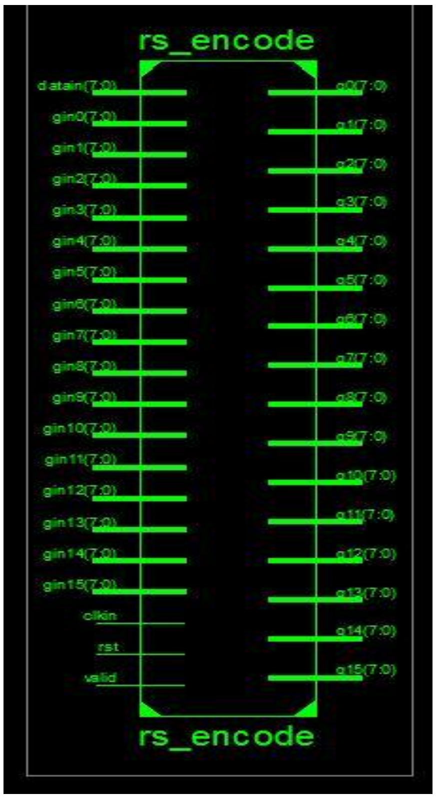


Fig.11. RTL Schematic for Encoder.

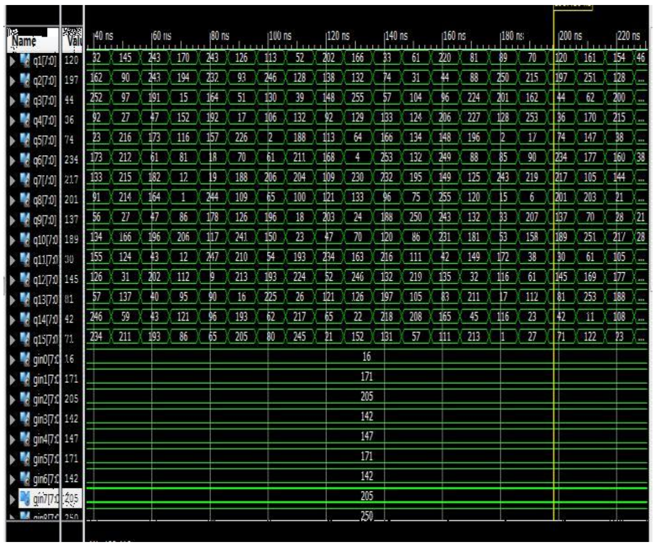


Fig.12. simulated Wave form for Encoder.

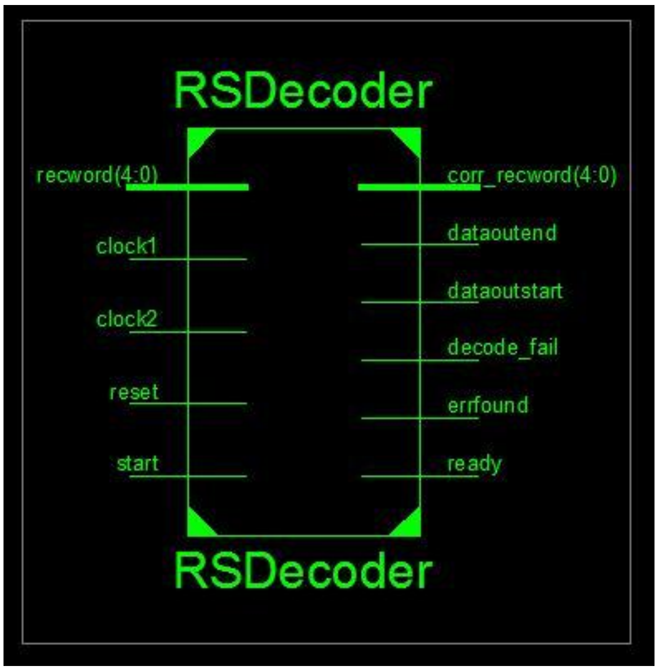


Fig.13. RTL Schematic For Decoder.

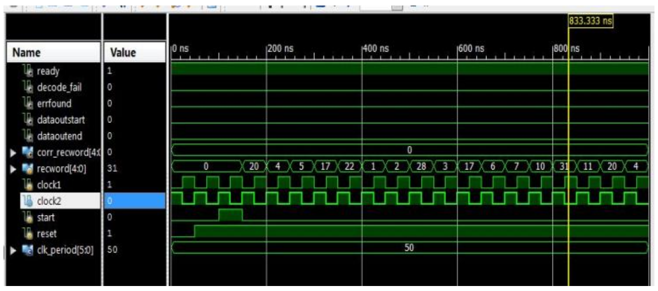
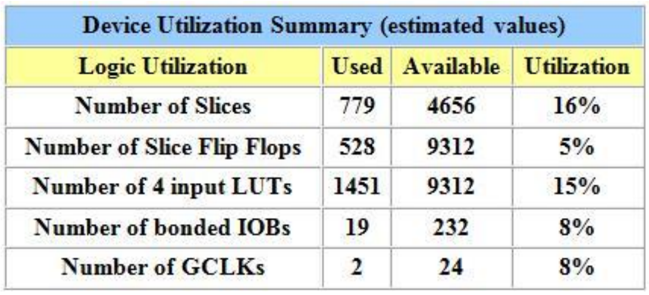


Fig.14. simulated Wave form for Decoder.

TABLE I: Device Utilization Summary



Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slices	779	4656	16%
Number of Slice Flip Flops	528	9312	5%
Number of 4 input LUTs	1451	9312	15%
Number of bonded IOBs	19	232	8%
Number of GCLKs	2	24	8%

VI. CONCLUSION

This project presents the verification of reed Solomon architectures for decoder. For the proposed design block, a detailed analysis is provided in terms of gate count, power and critical path delay. A new syndrome computation is proposed in the paper. The design of RS encoder and decoder is pipelined and reusability enables the less hardware requirement. The proposed decoder is significantly more efficient than prior designs. Next, the paper will focus on further improving the throughput of the RS decoder. From the synthesis results it can be concluded that the area required to integrate the chip is less with the LUT count of 1463 when compare with the existed LUT count of 5114 with reduced slices utilization of 528 with the existed slices utilization of 1534. The functionality is verified using ISE simulator and the synthesis is carried out using XILINX ISE 12.3i. In future there may be a chance to develop the chip with somewhat reduced area.

VII. REFERENCESS:

[1]F. Garcia-Herrero, J. Valls, and P. K. Meher, "High speed RS(255, 239) decoder based on LCC decoding," Circuits Syst. Signal Process. no. 6, pp. 1643–1669. Jun. 2011.

[2]XILINX X. Zhang and J. Zhu, "Algebraic soft-decision decoder architectures for long Reed–Solomon codes," IEEE Trans. Circuits Syst. II, no. 10, pp. 787–792, Oct. 2010.

[3]Peter Sweeney, ―Error Control Coding: From Theory to Practice,‖ John Wiley & Sons, Ltd., 2002.

[4]E. Berlekamp, "Non- binary BCH decoding," IEEE Trans. Inf. Theory, no. 2, p. 242, Mar. 1968.

[5]J. Bellorado and A. Kavcic, "A low-complexity method for Chase-type decoding of Reed–Solomon codes," in Proc. IEEE Int. Symp. Inf. Theory, Seattle, WA, Jul. 2006, pp. 2037–2041.

[6]W. J. Gross, F. R. Kschischang, R. Koetter, and P. Gulak, "A VLSI architecture for interpolation in soft-decision decoding of Reed–Solomon codes," in Proc. IEEE Workshop Signal Process. Syst., San Diego, CA, Oct. 2002, pp. 39–44.

[7]R. Koetter and A. Vardy, "A complexity reducing transformation in algebraic list decoding of Reed–Solomon codes," in Proc. IEEE Inf. Theory Workshop, Paris, France, Mar. 2003, pp. 10–13.

[8]J. Zhu and X. Zhang, "High-speed re-encoder design for algebraic soft decision Reed–Solomon decoding," in Proc. IEEE Int. Symp. Circuits Syst., New Orleans, LA, May, 2010, pp. 465–468.

[9]Z. Wang and J. Ma, "High-speed interpolation architecture for soft-decision decoding of Reed–Solomon codes," IEEE Trans. Very Large Scale Integer. (VLSI) Syst., no. 9, pp. 937– 950, Sep. 2006.

[10]X. Zhang, "Reduced complexity interpolation architecture for soft decision Reed–Solomon decoding," IEEE Trans. Very Large Scale Integr.(VLSI) Syst., no. 10, pp. 1156–1161, Oct. 2006.

[11]J. Zhu, X. Zhang, and Z. Wang, "Combined interpolation architecture for soft-decision decoding of Reed–Solomon codes," in Proc. IEEE Conf. Comput. Design, Lake Tahoe, CA, Oct. 2008, pp. 526–531.