Implementation of Low Delay Error Correction Codes

P.Rahul Reddy [1], N.Swapna [2]
[1] *Associate Professor, Dept of ECE, Swami Ramananda Tirtha Institute of Science & Technology,*
[2]*Assistant Professor, Dept of ECE, Ramanandatirtha Engineering College,*
Nalgonda, India

Abstract- **In the contemporary world, communication has got many applications such as telephonic conversations etc. in which the messages are encoded into the communication channel and then decoding it at the receiver end. During the transfer of message, the data might get corrupted due to lots of conflicts in the communication channel. So it is essential for the decoder tool to also have a function of correcting the error that might happen. Reed Solomon codes are type of burst error detecting codes which has got many applications due to its burst error detection and correction nature.**

Index Terms- **Reed-Solomon (RS), Galois Field (GS), Block length, Bit Error Rate (BER), Signal Noise Ratio (SNR)**

I. INTRODUCTION

Channel coding is an important signal processing operation for the efficient transmission of digital information over the channel. In channel coding the number of symbols in the source encoded message is increased in a controlled manner in order to facilitate two basic objectives at the receiver one is Error detection and other is error correction. Error detection and error correction to achieve good communication is also employed in devices. It is used to reduce the level of noise and interferences in electronic medium. The amount of error detection and correction required and its effectiveness depends on the signal to noise ratio (SNR) [1]. A channel code is a broadly used term mostly referring to the forward error correction code. Forward error correction (FEC) is a system of error control for data transmission, whereby the sender adds redundant data to its messages, also known as an error correction code. This allows the receiver to detect and correct errors without the need to ask the sender for additional data. FEC is applied where retransmissions are relatively costly or impossible. FEC information is usually added to most mass storage devices to protect against damage to the stored data [2]. There are many types of block codes, but the most notable is Reed Solomon coding, Golay, BCH, Multidimensional parity, and Hamming codes are other example of block codes.

Reed Solomon is an error-correcting coding system that was devised to address the issue of correcting multiple errors – especially burst-type errors in mass storage devices (hard disk drives, DVD, barcode tags), wireless and mobile communications units, satellite links, digital TV, digital video broadcasting (DVB), and modem technologies like xDSL [3]. Reed-Solomon codes are an important subset of non-binary cyclic error correcting code and are the most widely used codes in practice. These codes are used in wide range of applications in digital communications and data storage. Reed Solomon describes a systematic way of building codes that could detect and correct multiple random symbol errors. By adding t check symbols to the data, an RS code can detect any combination of up to t erroneous symbols, or correct up to ⌊t/2⌋ symbols. Furthermore, RS codes are suitable as multiple-burst bit-error correcting codes, since a sequence of b + 1 consecutive bit errors can affect at most two symbols of size b. The choice of t is up to the designer of the code, and may be selected within wide limits.

II. RELATED WORKS

After getting familiarized with classification and properties of forward error correction codes some of the error detection and correction codes are explained and compared in this section.

• **Hamming Code** - In a hamming encoder parity bits are inserted into the message bits. These parity bits are decided so as to impose a fixed parity on different combinations of data and parity bits. In decoder those combinations are checked for that fixed parity.

Accordingly decoder parity bits are set. Binary equivalent of this combination decides the location of the error. Then that particular bit is flipped to correct the data. Hamming code is a single error correction code. Double errors can be detected if no correction is attempted.

• **Berger Code** - Berger code is a unidirectional error detection code. It means it can only detect error either '1' flipped to '0' or '0' flipped to '1' but not both in a single code. If designed for detecting errors with '1' flipped to '0' then binary equivalent of number of 0s in the message is sent along with the message. Similarly when design for detecting '0' flipped to '1' error binary equivalent of the number of 1s in the message are sent along with the message. Decoder compares the number of 0s or 1s as per the design with the binary equivalent received. Mismatch between the two indicates the error. It can be used where error is expected to be unidirectional.

• **Constant weight code** -In this code a valid code word always have a constant weight. It means number of 1s in a valid code word is fixed. Hence any variation in this is an indication of error. It is simple but not efficient way of encoding as multiple errors can cancel out each other.

• **M out of N code** - In an M out of N encoder message is mapped to a N bit code word having M number of 1s in it. The N-M bits of message are appended with additional M number of bits which are used to adjust the number of 1s in the code. If the message consists of no 1s in it then all the M bits are set to '1'. It is also not an efficient code in terms of coding rate.

• **Erasure code** - Erasure means error when its location is known in advance from previous experience. Erasure code is able to correct such errors. In this type of code the decoder circuit does not need an error locator as it is already known. Hence only error magnitude is calculated by the decoder to correct the erasure.

• **Low Density Parity check code** - Low density parity check code is a linear block code. The message block is transformed into a code block by multiplying it with a transform matrix. Low density in the name implies low density of the transform matrix. That means number of 1s in the transform matrix is less. It is the best code as far as the coding gain is concerned but encoder and decoder design is complex. Mainly used in Digital Video Broadcasting.

• **Turbo Code** - It is a convolutional code. Encoding is simple convolutional encoding. It is defined by (n, k, l) turbo code where n is the number of input bits, k is the number of output bits and l is the memory of the encoder. Decoding is done in two stages. First one is soft decoding stage then a hard decoding stage. It has very good error correcting capability i.e. coding gain. The main drawback is that it has low coding rate and high latency. Hence it is not suitable for many applications. But in case of satellite communication as the latency due to the distance itself is so high this additional latency is negligible. Hence it is used mainly in satellite communication.

• **Reed Solomon Code** - Reed Solomon code is a linear cyclic systematic non-binary block code. In the encoder Redundant symbols are generated using a generator polynomial and appended to the message symbols. In decoder error location and magnitude are calculated using the same generator polynomial. Then the correction is applied on the received code. Reed Solomon code has less coding gain as compared to LDPC and turbo codes. But it has very high coding rate and low complexity. Hence it is suitable for many applications including storage and transmission.

III. PROPOSED REED SOLOMON CODES

In this section the encoder and decoder of REED SOLOMON is explained with the help of block diagrams.

Encoder hardware: The pipelined calculation performed using the conventional encoder circuit shown in Fig. 1. All the data paths shown provide for 4-bit values. During the message input period, the selector passes the input values directly to the output and the AND gate is enabled. After the eleven calculation steps shown above have been completed (in eleven consecutive clock periods) the remainder is contained in the D-type registers. The control waveform then changes so that the AND gate prevents further feedback to the multipliers and the four remainder symbol values are clocked out of the registers and routed to the output by the selector.

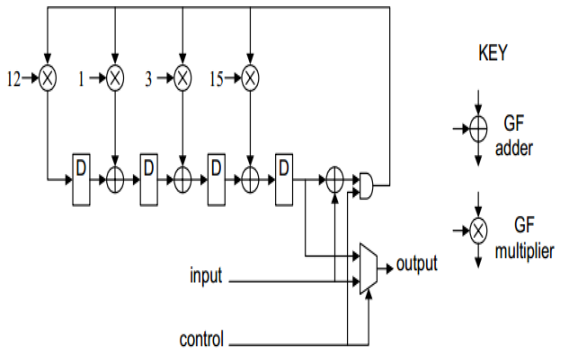

Fig. 1 - A (15, 11) Reed-Solomon encoder

Galois field adders: The adders of Fig. 1 perform bit-by-bit addition modulo-2 of 4-bit numbers and each consists of four 2-input exclusive-OR gates. The multipliers, however, can be implemented in a number of different ways.

Galois field constant multipliers: Since each of these units is multiplying by a constant value, one approach would be to use a full multiplier and to fix one input. Even though a full multiplier is significantly more complicated, with an FPGA design, the logic synthesis process would strip out at least some of the unused circuitry.

Dedicated logic constant multipliers: For the logic circuit approach, we can work out the required functionality by using a general polynomial representation of the input signal $a3\alpha^3 + a2\ \alpha^2 + a1\alpha + a0$. This is then multiplied by the polynomials denoted by the values 15, 3, 1 and 12 from Table 1.

Table 1.

index form	polynomial form	binary form	decimal form
0	0	0000	0
α^0	1	0001	1
α^1	α	0010	2
α^2	α^2	0100	4
α^3	α^3	1000	8
α^4	$\alpha + 1$	0011	3
α^5	$\alpha^2 + \alpha$	0110	6
α^6	$\alpha^3 + \alpha^2$	1100	12
α^7	$\alpha^3 + \alpha + 1$	1011	11
α^8	$\alpha^2 + 1$	0101	5
α^9	$\alpha^3 + \alpha$	1010	10
α^{10}	$\alpha^2 + \alpha + 1$	0111	7
α^{11}	$\alpha^3 + \alpha^2 + \alpha$	1110	14
α^{12}	$\alpha^3 + \alpha^2 + \alpha + 1$	1111	15
α^{13}	$\alpha^3 + \alpha^2 + 1$	1101	13
α^{14}	$\alpha^3 + 1$	1001	9

This involves producing a shifted version of the input for each non-zero coefficient of the multiplying polynomial. Where the shifted versions produce values in the α^6, α^5 or α^4 columns, the 4-bit equivalents (from Table 1) are substituted. The bit values in each of the α^3, α^2, α^1 and α^0 columns are then added to give the required input bit contributions for each output bit.

Look-up table constant multipliers: Alternatively, each multiplier can be implemented as a look-up table with 2m = 16 entries. The entry values can be obtained by cyclically shifting the non-zero elements from Table 1 according to the index of the multiplication factor.

Code shortening: For a shortened version of the (15, 11) code, for example a (12, 8) code, the first three terms of the message polynomial, equation (10), would be set to zero.

The arrangement of the main units of a Reed-Solomon decoder reflects, for the most part, the processes of the previous Section

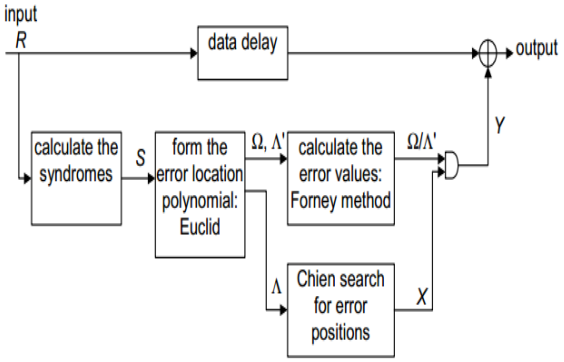

Fig. 2- Main processes of a Reed-Solomon decoder

Thus, in Fig. 2, the first process is to calculate the syndrome values from the incoming code word. These are then used to find the coefficients of the error locator polynomial Λ_1 .... Λ_v and the error value polynomial Ω_0 .... Ω_{v-1} using the Euclidean algorithm. The error locations are identified by the Chien search and the error values are calculated using Forney's method. As these calculations involve all the symbols of the received code word, it is necessary to store the message until the results of the calculation are available. Then, to correct the errors, each error value is added (modulo 2) to the symbol at the appropriate location in the received code word.

IV. SIMULATION RESULTS

Reed Solomon Error Probability: Reed Solomon codes are mainly used for burst error correction. However the code has its own error correcting capability. So, the error probability plays a crucial role in saving our time detecting and correcting the

error. Let us assume that the code can correct 4 error symbols in an (255,251) RS code. A maximum of 32 bits of error can be corrected. So, if we can calculate the bit error rate properly and then manage the syndrome calculation part as if the decoder calculates more than 32 bits of error, then send a signal that decoder cannot correct. Therefore plotting the bit error probability (P) against the SNR will help. We can have a range of SNR for which the error can be corrected. However the range will include many parts like the percentage of probability that the signal will get detected.

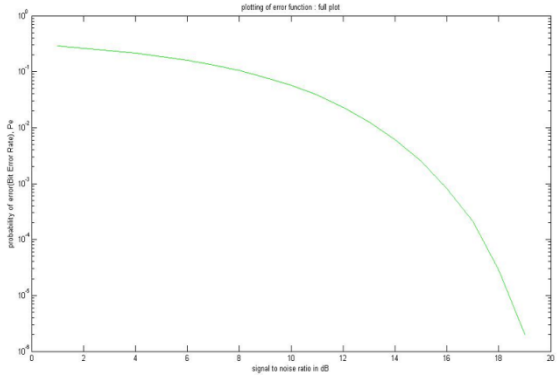

Fig. 3: Graph between SNR and Bit error rate (BER)

Fig. 3 shows the plot between bit error probability and SNR. The code is for a random of about 255 symbols where each symbol contains 8 bits being transmitted. These 255 symbols form a code word. And there are about 500 such code words. However the range estimation for different capability of correcting errors can be calculated.

V. CONCLUSION

This paper presents clear understanding of Reed-Solomon codes used in error detection and correction.The main purpose of this paper was to study the Reed-Solomon code encoding and decoding process and also the error probability for the RS code. The encoding process and the block diagram have been discussed and also the different step for decoding process has been discussed. The error probability for RS code shows that the BER performance also improves for large block length and shows a poor BER performance for lower SNR as the SNR value increases the curve becomes steeper.

REFERENCES

[1] Carl Eklund, et.al. "WirelessMAN: Inside the IEEE 802.16 Standard for Wireless Metropolitan Area Networks," IEEE Press, 2006.

[2] Hagenauer, J., and Lutz, E., "Forward Error Correction Coding for Fading Compensation in Mobile Satellite Channels," IEEE JSAC, vol. SAC - 5, no. 2, Feb 1987, pp. 215 -225.

[3] High-speed VLSI Architecture for Parallel Reed-Solomon Decoder", IEEE Trans. On VLSI, April 2003.

[4] J. Jittawutipoka, J. Ngarmnil, "Low complexity Reed Solomon encoder using Globally optimized finite field multipliers" , IEEE Region 10 conference, Nov. 2004.

[5] Lee H., "High Speed VLSI Architecture for parallel Reed solomon Decoder", IEEE Transactions on VLSI systems, 2003.

[6] Wilhelm W., "A New Scalable VLSI Architecture for Reed-Solomon Decoders", IEEE Journal of Solid State Circuits, 1999

[7] Y. J. Jeong and W Burleson, "VLSI Array synthesis for polynomial GCD Computation and Application to Finite Field Division", IEEE Transactions on Circuits and Systems, 1994

[8] Wicker, Stephen B., Bhargava, Vijay K, "Reed-Solomon Codes and the Compact Disc", IEEE Press ISBN 978-0- 7803-1025-4.