International Journal of Applied Engineering Research
ISSN 0973-4562 Volume 9, Number 20 (2014) pp. 6855-6862

http://www.ripublication.com

Study of Reed Solomon Encoders and its Architectures

A. Deepa[1] and C.N. Marimuthu[2]

[1]Assistant Prof. Department Of ECE J.K.K. Muniraja College of Technology, Tamilnadu, India
[2]Prof. & Dean Department of ECE Nandha Engineering College, Tamilnadu, India
E-mail: [1]adeepaeaswaramoorthy@gmail.com, [2]muthu_me2005@yahoo.co.in

Abstract

Detection and correction of errors in digital data is a vital issue for the recent communication systems. Errors can creep into message data during transmission or reception, altering or erasing one or more message bytes. Sometimes, errors are introduced deliberately to sow disinformation or to corrupt data. Hence an efficient error control code is needed to protect the digital data. Reed Solomon code is widely used to identify and correct data errors in transmission and storage systems. We give a wide knowledge of the Reed Solomon (RS) encoder and study has been made on several RS encoder architectures by reviewing several papers.

Keywords: Reed Solomon, Galois field (GF), Generator polynomial, Bit error rate.

1. Introduction

When compared to analog communication, digital communication is immune to noise and errors. But there exists some errors due to change in the bits. To overcome these errors, error correcting codes have a wide range of applications in different fields like digital data communications, memory system design are some of them [9].

Reed Solomon code is an error correcting coding system discovered by Irving S. Reed and Gustave Solomon in 1960 and is used for correcting multiple errors especially burst type errors. Demand for higher data rates makes it necessary to devise very high speed implementation of RS codes. It is a cyclic, non binary linear code and a kind of BCH code with strong error correction capability. Compared with other

linear block codes RS code has strong error correcting capability with the same coding efficiency's code can correct not only random errors but also unexpected errors [9]. Hence it is widely used in deep-space communication systems, digital subscriber loops, wireless systems, data storage systems, digital television transmission systems as well as in memory. For example RS (28, 24) and RS (32, 28) codes with interleaving are popularly used for storage in CD.RS (255, 223) code has been used for digital microwave radio. RS (23, 17) codes are used to preserve important information in MB-OFDM UWB [13].RS code RS (255,223) has been selected by the Consultative Committee for Space Data Systems (CCSDS) as a correction coding tool for the forward and backward signals in the communication link of Advance Orbiting Systems (AOS) [12].

RS coding is a method of forward error correction in the form of block coding. Block coding consists of calculating a number of parity symbols over a number of message symbols. The parity symbols are appended to the end of the message symbols forming a code word [9].

2. Reed Solomon Codes

Reed Solomon code can be specified as RS (n, k) as shown in Fig.1

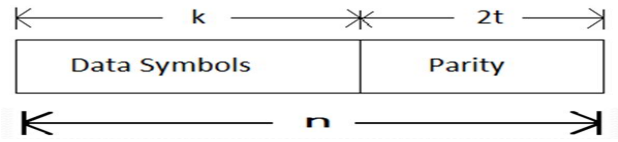

Fig. 1: Structure of a RS codeword

where,
k-number of message symbols in each block
n-size of the output code word
t-number of symbols that can be corrected by RS code, t = (n-k) / 2
2t-number of parity symbols
d- minimum distance, d = 2t + 1

Each symbol has 'm' number of bits. The relationship between the symbol size 'm'and the size of the output code word 'n' is given by, $n = 2^m - 1$.

3. Reed Solomon Data Transfer Channel

A Reed-Solomon protected communication or data transfer channel is shown in Fig.2

The RS encoder at the transmitter encodes the input message into a codeword. Then it transmits the same through the channel. At the channel the codeword gets corrupted by the noise. The corrupted codeword is checked and corrected by the

decoder at the receiver. The original corrected message is later fed to the receiver. If the error induced at the channel is greater than the error correcting capability of the decoder then there occurs a failure in decoding. These decoding failures occur if the codeword is passed unchanged to the receiver. On the other hand decoding failure will lead to a wrong message being given at the output [1].

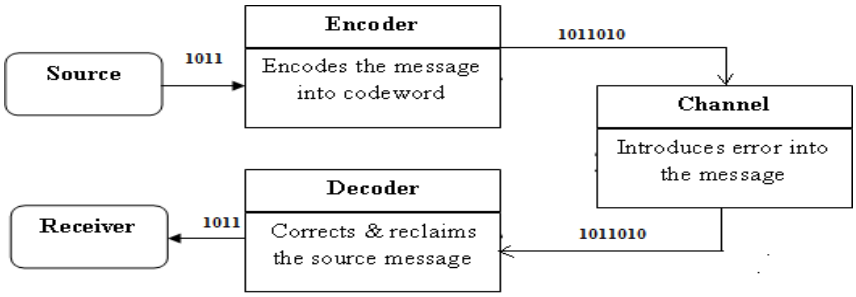

Fig. 2: Reed-Solomon data transfer channel.

4. Generator Polynomial and Galois Field

The Reed-Solomon code is defined in the finite field called the Galois field. This contains a finite set of numbers where any arithmetic operations on elements of that set will result in an element belonging to the same set. In Galois field every element except zero can be expressed as a power of a primitive element, α of the field. The non-zero field elements form a cyclic group defined based on a binary primitive polynomial. Addition in the Galois field is simply the XOR operation [1].Multiplication in the Galois field is more complex than the standard arithmetic. In the multiplication module the primitive polynomial is used to define the Galois field [1].

A Reed Solomon code is a special case of a BCH code in which the length of the code is one less than the size of the field over which the symbols are defined. It consists of sequences of length (q-1) whose roots include 2t consecutive powers of the primitive element of GF (q).Alternatively, the Fourier transform over GF(q) will contain 2t consecutive zeros. To construct the generator for a Reed Solomon code it is necessary to construct the appropriate finite field and choose the roots. If the roots are from α^{i} to α^{i+2t-1}, the generator polynomial will be

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

5. Reed Solomon Encoder

The Reed Solomon encoder reads in k data symbols computes the n-k symbols; append the parity symbols to the k data symbols for a total of n symbols.RS codes are systematic, so for encoding, the information symbols in the codeword are placed as the

higher power coefficients. This requires that information symbols must be shifted from power level of n-1 down to n-k and the remaining positions from power n-k-1 to 0 be filled with zeros[6].Any RS encoder design should effectively perform two operations namely division and shifting [1].Both operations can be easily implemented using Linear Feedback Shift Registers [6], [7],[8].

The parity symbols are computed by performing a polynomial division using GF algebra. The steps involved in this computation are as follows:

- Multiply the message symbols by x^{n-k} .This shifts the message symbols to the left to make space for the n-k parity symbols.
- Divide the message polynomial by the code generator polynomial using GF algebra.
- The parity symbols are the remainder of this division.

These steps are accomplished in hardware using a shift register with feedback, Galois field adder, Galois field multiplier and switches. The architecture [1] for the encoder is shown in Fig.3

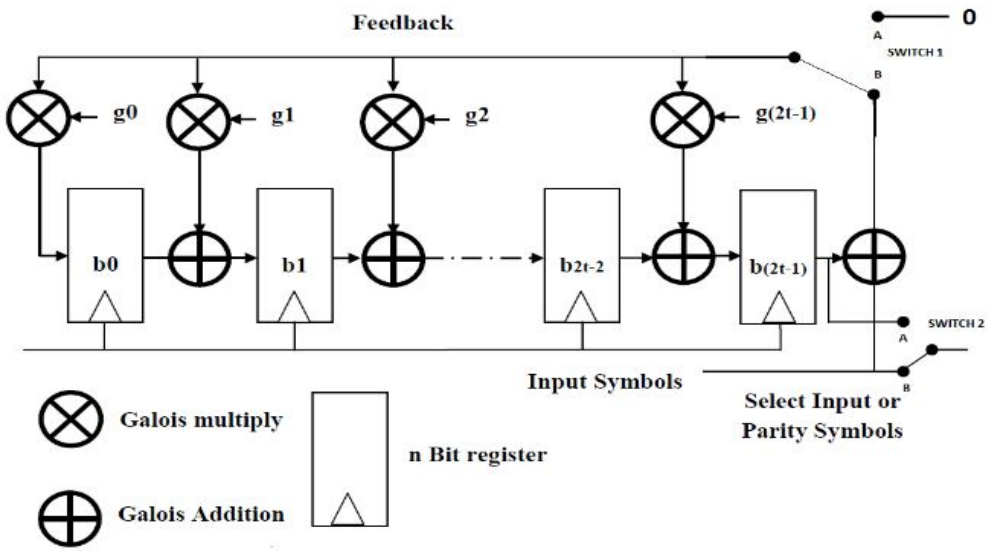

Fig. 3: Block Diagram of Reed-Solomon Encoder.

The complexity of coding structure depends upon the multiplier. So the selection of the multiplier structure is a critical issue. Since the domain is characterized by base-two systems, the Galois field adders and multipliers are implemented by EXOR logic. The encoder works as follows [11].

The encoder is a 2t tap shift register, where each registers are m bits wide. The multiplier coefficients g0 to g (2t-1) are coefficient of the RS generator polynomial. The coefficients are fixed, which can be used to simplify the multiplier if required. The only hard bit is working out the coefficients, and for hardware implementations the value can often be hard coded.

At the beginning of a block all the register are set to zero. From then on, at each clock cycle the symbol in each register is added to the product of the symbol and the fixed coefficient for that trap, and passed on to the next register. The symbol in the last register becomes the feedback value on the next cycle. When all n inputs symbols have been read in, the parity symbols are sitting in the register, and it just remains to shift them out one by one.

6. Review On RS Encoder Architectures

For designing the RS encoder two components are used. One is LFSR, which is the main component for designing the encoder. The other is Registers and EXOR gates to form the LFSR. One important advantage of the LFSR is that it produces immediate response for every clock pulse. But it transfers the data in serial manner. Another component is Galois field multiplier. This multiplier performs the multiplication by the use of finite field theory [18].In all the arithmetic operations required in the implementation of RS codes finite field multiplication is the most frequently studied. The operations like inversion and division can be decomposed into repeated multiplications [14].The conventional encoder depends on the structure for polynomial division. Since polynomial division requires multiplications in GF (2^m), the complexity of the structure depends on the multiplier used. The finite field multipliers can be divided into standard- basis, normal basis and dual-basis multipliers according to different field elements base. In general the finite field multiplier consists of bit-serial and bit-parallel structures. Various methods have been proposed for the serial and parallel multiplier structures of RS encoder.

6.1 Berlekamp ‘s Bit Serial Multiplier Algorithm

Berlekamp developed a bit serial multiplication algorithm for the encoding of RS codes, using a dual basis over a Galois field. The conventional RS encoder for long codes often requires lookup tables to perform multiplication of two field elements.Berlekamp’s algorithm requires only shifting and EXOR operations. Berlekamp’s multiplier (BM) has easy-to-drive structure. The only disadvantage of the Berlekamp‘s multiplier is that it operates over two bases, the dual basis and the polynomial basis [14]. (i.e.) the polynomial basis for the multiplier and the dual basis for the multiplicand and the product. It is normal to input all data in the same basis; this means that some basis transformation circuits will be required. Even including the extra hardware for basis conversions, the BM is known to have the lowest hardware requirement [2].RS encoder proved to perform well with this algorithm, however in this design ,the multiplicand is a fixed finite field constant which is inconvenient if one desires to change the multiplicand. The disadvantage of this multiplier is that it operates over both the dual and the polynomial basis, and so basis converters may be required.

6.2 Massey-Omura Multiplication Algorithm

Massey Omura Multiplier (MOM) is also a bit serial multiplication algorithm for the encoding of RS codes and operates entirely over the normal basis and hence no basis converters are required. The idea behind the Massey-Omura multiplier is that if the Boolean function generating the first product bit has the inputs cyclically shifted, then this same function will also generate the second product bit. Further with each subsequent cyclic shift a further product bit is generated [16].However, the MOM requires more hardware than the BM and cannot efficiently carry out constant multiplication [14].

6.3 Bit Parallel Multiplier Algorithm

Hardware implementation of serial multiplier is relatively simple, but it is difficult to achieve high speed and high throughput due to its bit-by-bit operation [12].Bit parallel multiplier algorithms overcomes the drawback of bit serial multiplier algorithm. It is generally used in high speed applications. Multiplicands use the standard basis which does not need the base conversion. It has simple structure and easy to expand to higher order in Galois field.

- Surendra K. Jain, Keshab K. Parhi Presents an efficient Reed-Solomon encoder based on standard basis. The key operation in Reed-Solomon encoding is the multiplication of a feedback term with several (possibly) known terms, an efficient structure is implemented for this operation. The hardware complexity of this encoder is identical to the well-known Berlekamp encoder. It however, offers two advantages over the Berlekamp encoder-a critical path independent of the order of Reed-Solomon code being implemented and the ability to encode without any need for basis conversion [3].
- Taking RS encoder in CMMB system for example, Zefu Tan, Hong Xie, Guangjie Wu and Mingxia Liao explains about an improved algorithm according to the method of bit-parallel multiplier based on dual basis. This algorithm can achieve a higher rate, and the RS encoder can be applied in a wide range of systems when implemented in FPGA [11].
- Based on the GF (2^8) polynomial multiplication principle, the conventional Reed-Solomon (RS) coding structure is used to implement the RS (255,223) encoder from CCSDS (Consultative Committee for Space Data System).Zhang Jinzhou, Liang Xianfeng,Wang Zhugang and Xiong Weiming explains about the remainder program of polynomial multiplication and the test method of RS encoder. The encoder is highly efficient and suitable to high speed data transmission for the satellite downlink [17].
- A.R. Dash, T.R. Lenka designed an encoder which is implemented with 32 optimized finite multipliers. The multiplication operation in the structure is converted into a group of digital logic operations. The multiplication by a constant approach reduces the number of logic operations, thereby reducing the number of gates. The number of modulo 2 additions or XOR gates is

minimized by reducing the redundant operations so that the multiplier in RS encoder partly shares the same hardware operations. Thereby the computational complexity for one multiplier is improved. Further redundancies were identified and reduced to only one XOR operation per group of redundancies which is known as global optimization. Multiplier structure used is simple and ensures high speed operations. It is highly efficient, low complexity and good coding performance. The encoder is implemented with 32 optimized finite multipliers [12].

7. Proposed Work

In RS encoder design, Linear Feedback Shift Register (LFSR) is the main component. LFSR allows only serial inputs. When the clock pulse is applied, it transfers the data immediately. Future work is to design a RS encoder so that parallel implementation is done using unfolding algorithm to overcome the drawback of serial transmission of LFSR. By doing this, the sampling period is reduced which is further exploited to increase the clock speed.

8. Conclusion

This paper gives a brief explanation on Reed-Solomon code, generator polynomial that is used to generate the code, the Reed-Solomon encoding process and its block diagram .The main purpose of this paper is to study about the different methods that are used to reduce the complexity of the RS encoder by reducing the finite field computations and multipliers.

References

[1] Sandeep Kaur, *VHDL Implementation of Reed-Solomon code*, Thesis, Thapar Institute of Engg, 2006.

[2] I.S. Hsu, I. S. Reed, T. K. Truong, Ke Wang, Chiunn-Shyong Yeh and L. J. Deutsch, *The VLSI implementation of a Reed-Solomon encoder using Berlekamp 's bit-serial multiplier algorithm*, IEEE Trans.Comput, No.10,906, Oct. 1984 .

[3] Surendra K. Jain and Keshab K. Parhi, *Efficient standard basis Reed-Solomon encoder*, IEEE International conference, vol.6, pp. 3287-3290, May 1996.

[4] Xiaojun Wu, Xianghui Shen and Zhibin Zeng, *An improved RS encoding algorithm*, IEEE CECNet International conference, pp. 1648-1652, April 2012.

[5] J. Jittawutipoka and J. Ngarmnil, *Low complexity Reed Solomon encoder using Globally optimized finite field multipliers*, IEEE Region 10 conference, vol. 4, pp 423-426, Nov. 2004.

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

[7] *Reed-Solomon (RS) Coding Overview*, VOCAL Technologies, Ltd., Rev. 2.28n, 2010.

[8] J. Y. Chang and C. B. Shung, *A high speed Reed-Solomon codec chip using look forward architecture*, IEEE APC CAS'94, pp. 212-217, Dec. 1994.

[9] B. Sklar., *Digital Communications: Fundamentals and Applications*, 2nd ed., Prentice-Hall, 2001.

[10] C.K.P. Clarke., *Reed-Solomon error correction, BBC R&D White Paper*, WHP 031, July 2002.

[11] Zefu Tan, Hong Xie, Guangjie Wu and Mingxia Liao, *Design and implementation of Reed-Solomon encoder in CMMB system*, in Proc. of 6th IEEE Int. Conf. Wireless Communications Networking and Mobile Computing, WiCOM 2010, pp. 1–4, Sep.2010.

[12] A.R. Dash and T.R. Lenka, *VLSI Implementation of Reed-Solomon Encoder Algorithm for Communication Systems*, Radio electronics and Communications Systems, Springer, Vol. 56, No. 9, pp. 441–447, Sep 2013.

[13] Anindya Sundar Das, Satyajit Das and Jaydeb Bhaumik, *Design of RS (255, 251) Encoder and Decoder in FPGA*, International Journal of Soft Computing and Engineering,Volume-2, issue-6, Jan. 2013.

[14] S.T.J.Fenn, M.G.Parker, M.Benaissa and D. Taylor, *Bit-serial Multiplication in GF(2m) using irreducible All-one polynomials*, IEEE Proc.Comput. Digit. Tech., Vol. 144, No. 6, pp.391-393, Nov. 1997.

[15] I.S.Hsu, T.K.Truong, H.M.Shao and L.J.Deutsch, I.S.REED, *A Comparison of VLSI architecture of finite field multipliers using Dual, Normal or standard basis*, TDA Progress Report, pp. 42-90, April 1987.

[16] Hardik Punamch, Sutaria, *Review Paper: Bit-Serial Multiplier Techniques For Finite Fields*, Journal Of Information, Knowledge And Research In Electronics And Communication Engineering, Vol.- 02, Issue- 02, pp. 648-651, Nov. 2012.

[17] Zhang Jinzhou, Liang Xianfeng, Wang Zhugang and Xiong Weiming, *The Design of a RS Encoder*, Future computing, communication, control and Management, Springer, Vol.144, pp.87-91, 2012.

[18] Keshab K. Parhi, *Eliminating the Fan out Bottleneck in Parallel Long BCH Encoders*, IEEE Transaction on Circuits and Systems, Vol. 51, No.3, Mar. 2004.