Implementation of Reed Solomon Encoder

M.Revathi[1], Dr.D.Rukmani Devi[2]

[1,2]Electronics and Communication, R.M.K Engineering College, Chennai, Tamilnadu, India

Abstract

This paper presents an implementation of reed Solomon encoder using GF(2^m) multiplier. Register sharing systolic structure are used to implement this GF(2^m) to reduce the area and time delay. This technique reduces the register requirement in systolic structure and also reduces the latency.

Keywords: *Galois Field, Reed-Solomon, Register sharing systolic structure.*

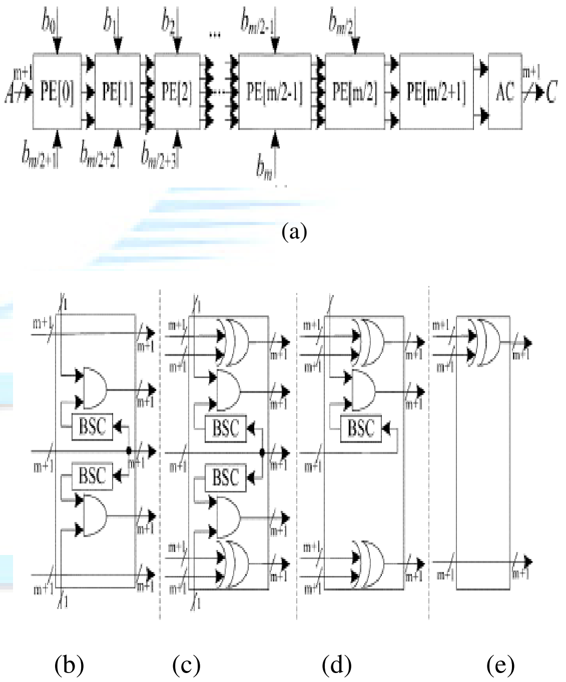
1. Introduction

Digital communication system is used to transport an information signal from the source to destination via a communication channel. A code is the set of all the encoded data, the code word that an encoder can produce. When actual set of data encoded it becomes a code. Reed-Solomon error correcting codes (RS codes) are widely used in communication systems and data storages to recover data from possible errors that occur during transmission and from disc error respectively. Application of the RS codes is the Forward Error Correction (FEC).

Finite field multipliers over GF(2^m) have wide applications in elliptic curve cryptography (ECC) and error control coding systems [1], [2]. Systolic design is a preferred type of specialized hardware solution due to its high-level of pipeline ability, local connectivity. But some issues are there in systolic structures. First the systolic structures have more registers this will consume large area and power. Second the systolic structures normally have latency nearly m cycles it is not applicable of real time applications. Therefore in this paper presents register-sharing technique to reduce the register requirements in the systolic structure. So it can reduce the register complexity and also reduce the latency. Register- sharing technique to reduce the register requirement in the systolic structure.

2. Register Sharing Systolic Structure

The proposed technique not only facilitates sharing of registers by the neighboring PEs to reduce the register complexity but also helps reducing the latency.



(a)

(b) (c) (d) (e)

Fig. 1 Low Latency register-sharing systolic structure. (a) The systolic structure. (b)Structure of PE[1].(C)Structure of a regular PE(from PE[2] to PE[m/2-1].(d)Structure of PE[m/2+1].

The register sharing systolic structure is shown in figure.1. The regular processing element consists of three basic cells. Bit shift cell, AND cell and the XOR cell. The basic design of systolic structure having (m+2) processing elements and during each cycle period not only perform the modular reduction operation but also performs the bit multiplication and bit addition.

In this register sharing systolic structure design having (m/2 +2) processing elements. From this area can be reduced in that structure[6].Using this structure design the Galois field multiplier. Galois field multiplier, as a special multiplier, is also known as finite field multiplier for all of its calculations are performed over finite field. It has been widely used in various applications of communication, such as encoding, error correction, encryption, etc. And use this multiplier design the Reed Solomon encoder.

3. Reed Solomon Theory

A Reed-Solomon code is a block code and can be specified as RS (n,k) as shown in Figure 2. RS codes are generally represented as an RS (n, k), with m-bit symbols, where

Block Length: n
No. of Original Message symbols: k
Number of Parity Digits: n - k = 2t
Minimum Distance: d = 2t + 1.

The relationship between the symbol size, m, and the size of the code word n, is given by
$n= 2^m – 1$

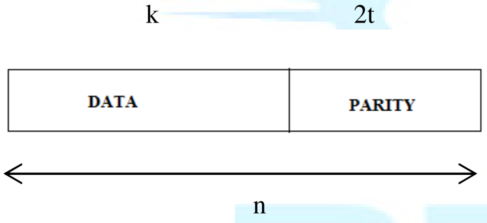


Fig. 2 Structure of a RS codeword

The RS encoder provided at the transmitter end encodes the input message into a codeword and transmits the same through the channel. Noise and any other disturbances in the channel may disrupt and corrupt the codeword. This corrupted codeword arrives at the receiver end (decoder), where it gets checked and corrected message is passed on to the receiver. In case the channel induced error is greater than the error correcting capability of the decoder a decode failure can occur. Decoding failures are said to have occurred if a codeword is passed unchanged, a decoding failure, on the other hand will lead to a wrong message being given at the output [3].

4. RS Encoder

The Reed-Solomon encoder reads in k data symbols,

calculate the n - k parity symbols, and add the parity symbols to the k data symbols for a total of n symbols. The encoder having a 2t taps shift register where each register is m bits wide. Coefficients of the multiplier are the coefficients of the RS generator polynomial. The main idea is the construction of a polynomial; the coefficients produced will be symbols such that the generator polynomial will exactly divide the data/parity polynomial [4].

The codeword is systematically encoded and defined in as a function of the transmitted message m(x), the generator polynomial g(x) and the number of parity symbols 2t as given below.

$c(x)=m(x) * 2t + m(x)modg(x)$

Where, g(x) is the generator polynomial of degree 2t and given by,

$$g(x) = \prod_{i=m0}^{m0+2t-1}(x+\alpha\ i)$$

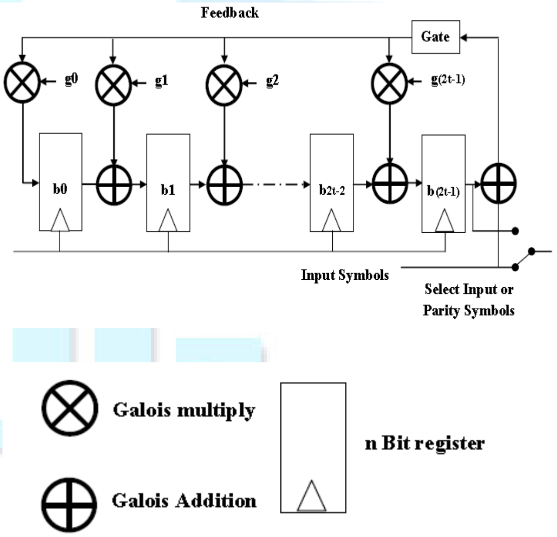


Fig. 3 Block diagram of RS encoder

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. Therefore any 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 [3][4][5].

The encoder block diagram shows that one input of the each multiplier is a constant field element, it is a coefficient of the polynomial g(x).

The information polynomial M(x) is given into the encoder symbol by symbol. After some period of latency these symbols appear at the output of the encoder, where control logic feeds it back through an adder to produce the related parity. All of the k symbols of M(x) are input to the encoder until this process will continued. At that time, the control logic at the output enables only the input data path, keep the parity path disabled. With an output latency of about one clock cycle, the output of the encoder, the last information symbol at (k+1)th clock pulse. Also during the first k clock cycles, the feedback control logic feeds the adder output to the bus.

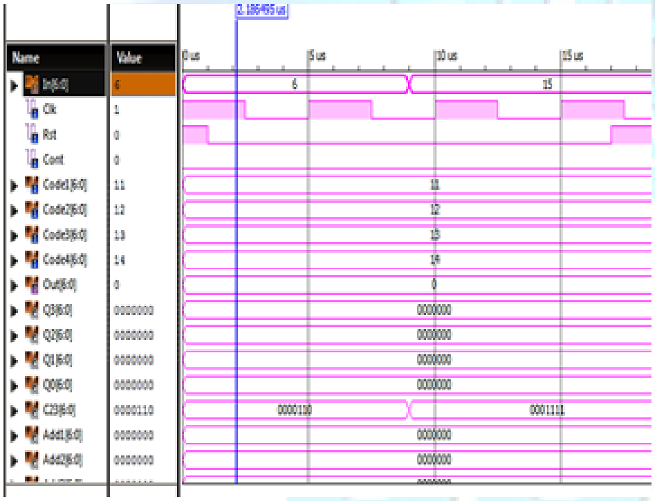


Fig. 4 Output of RS encoder.

After the last symbol has been input into the encoder (at the kth clock pulse), a wait period of at least n-k clock cycles occurs. During this waiting time, the feedback control logic disables the adder output from being fed back and supplies a constant zero symbol to the bus. Also, the output control logic disables the input data path and allows the encoder to output the parity symbols (k+2th to n+1th clock pulse). New block started at the n+1th clock pulse [3].

5. Conclusion

Reed Solomon codes are efficient and non-binary error correcting codes. Encoder is implemented using Register sharing systolic structures. In future, can design the RS decoder and also implement this in an FPGA.

References

[1] M. Ciet, , J. J. Quisquater, and F. Sica, "A secure family of composite finite fields suitable for fast implementation of elliptic curve cryptography," *in Proc. Int. Conf. Cryptol. India*,2001,pp.108-116.

[2] H. Fan and M. A. Hasan, "Relationship between GF(2^m) Montgomery and shifted polynomial basis multiplication algorithms," *IEEE Trans. Computers*, vol. 55, no. 9, pp. 1202–1206, Sep. 2006.

[3] Sandeep Kaur "VHDL Impletation of Reed-Solomon code," Thesis, Thapar Institute of Engg, 2006.

[4] "Reed-Solomon Coding Overview," VOCAL Technologies, Ltd., Rev. 2.28n, 2010

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

[6] Jiafeg Xie, Pramod Kumar Meher and Jianjun He, "Low-complexity multiplier for GF(2^m) based on all-one polynomials,"IEEE Trans.Very Large Scale Integr. (VLSI) Syst., vol. 21, no. 1, pp. 168– 173, 2013.

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

[8] H.-S. Kim and S.-W. Lee, "LFSR multipliers over GF (2^m) defined by all-one polynomial," Integr., VLSI J., vol. 40, no. 4, pp.571-578,2007.

[9] M. Sandoval, M. F. Uribe, and C. Kitsos, "Bit- serial and digit-serial GF (2^m) montgomery multipliers using linear feedback shift registers," IET Comput. Digit. Tech., vol. 5, no. 2, pp. 86–94, 2011.