Performance Analysis of (255, 239) Reed Solomon Code for Efficient Knowledge-based Systems in Ubiquitous Environment

Wonshik Na, Jae-Yeon Choi

Abstract*: *The Reed-Solomon code is used for the error correction with a wide range of applications in wireless digital communication and data storage system. This paper analyzes the performance of (255, 239) Reed-Solomon code, and evaluates its performance in the environment of additive white Gaussian channel. The performance is estimated in bit error rate and signal energy to noise ratio. (255, 239) Reed Solomon code can be applicable to the efficient knowledge-based system as well as the smart communication system. We design a bit-serial Reed Solomon encoder that operates in parallel on a polynomial basis, using a polynomial divider and a bit serial multiplier with adder located outside the shift register. Also, we apply it to the Reed Solomon code for (255, 239) and the encoder is designed. This paper is aimed to evaluate the performance of Reed Solomon code over the noisy communication channel. As the code length increases with similar error correcting capability, the performance of bit error rate improves and redundancy of the code also increases.

Index Terms*: *Reed-Solomon code, Knowledge-based, RS encoder, Knowledge systems

I. INTRODUCTION

In the field of correcting errors occurred due to noise, the Reed Solomon code is extensively used in many communication systems and digital storage systems because it performs an efficient error correction function for random errors as well as for burst errors [1]. Especially, in the satellite communication system, transmission errors can occur because of the long distance from the ground, the exposure to various electromagnetic stimuli, and the attenuation due to the weather change. One of the ways to overcome these problems is to use Reed Solomon codes, which react strongly to both random errors and burst errors. Reed Solomon codes have been utilized in space communications and satellite communications with improving performance since they started to be commonly used in Voyager probes. Consultative Committee for Space Data Systems, a satellite communication standardization organization, strongly recommends the use of Reed Solomon codes as error correcting codes [2]. This error correcting code technique for the burst error like Reed Solomon code can be applied to the efficient knowledge-based communication and smart communication system.

The Reed Solomon encoder is implemented as a division circuit that divides information polynomials into generator polynomials on a finite field $GF(2^m)$ [3]. Since the Reed Solomon encoder operates m bits in parallel, the circuit becomes complicated if the error correction capability and m bits are increased. To reduce this complexity, Berlekamp proposed a bit-serial Reed-Solomon encoder with Reed Solomon coder serialized on a dual basis [4]. Since the encoder operates in series, it becomes much simpler in hardware than the conventional parallel-operated encoder. However, this encoder requires a converter to convert the output of the encoder to a polynomial basis since the output appears in the form of dual basis representation [5][6]. This paper is aimed to design the encoding and decoding procedure of RS code and analyse the performance of the Reed Solomon code on the $GF(2^8)$.

Revised Manuscript Received on May 23, 2019.

Wonshik Na, Professor, Department of Computer Science, Namseoul University, 91, Daehak-ro, Seonghwan-eup, Seobuk-gu, Cheonan, Chungcheongnam-do, 31020, Republic of Korea

Jae-Yeon Choi(Corresponding author), Professor, Department of Information and Communication Engineering, Namseoul University, 91, Daehak-ro, Seonghwan-eup, Seobuk-gu, Cheonan, Chungcheongnam-do, 31020, Republic of Korea

II. RELATED WORK

A. Finite Field Theory

Finite Field Arithmetic has been largely applied in fields such as error correction coding and cryptographic theory [1]-[3]. In the finite field $GF(2^m)$, the elements can be represented as the polynomial that has the order of *m-1* or less than *m-1* and has the coefficients 1 and 0.

When the field elements are represented by polynomials in the $GF(2^m)$ finite fields, addition of elements can be easily implemented by bitwise XOR, whereas multiplication and division are very complicated. As division can be implemented by repetition of exponential multiplication and multiplication, a multiplication is the key operation in the finite field $GF(2^m)$.

However, if a finite field $GF(2^m)$ has subfields, a finite field multiplier using the bit parallel multipliers on the $GF(2^m)$ can be implemented with fewer hardware than the conventional bit parallel multiplier [3][4].

B. RS Encoder

A Reed-Solomon code (RS code), which can correct symbol errors occurred due to noise with cyclic characteristics[6], has been extensively used in many computer storage systems and communications [7][8].

Specifically, a concatenated encoding system with its convolutional code has been utilized for the downlink channel of space communication and for knowledge-based intelligent wireless communication systems. In this system, a convolutional code can function as the inner code, producing good results with comparatively high code rates [9]. One of the most efficient ways is to use the codes with short constraint length and the decoder with Viterbi algorithm as the convolutional code can perform at a relatively high error rate. In addition, the RS code is largely used in secure communication systems to protect from jamming. Since this code has an error control function, it is adopted in the digital systems for storage such as optical data storage and magnetic storage systems. The RS code can be shortened and used in some storage system with 512 bytes of each data block. Some parity-check bytes are added to the data blocks for error control during the recording operation[10].

The RS code has both the superiority in error correction and the availability of the efficient decoding method. The Berlekamp iterative algorithm is an algorithm that finds the smallest linear feedback shift register that can produce a given sequence. This algorithm is very efficient for the RS code decoder and requires the computation of the error values for the slight modification.

If m is the number of bits per symbol of the RS code, the code has 2^m binary symbols block sequence of the finite field $GF(2^m)$. The coefficients of a code polynomial $c(x) = c_0+c_1x+c_2x^2+\cdots+ c_{n-1}x^{n-1}$ show the symbol sequences, which is the field elements of the finite field $GF(2^m)$.

The parameters of an (n, t) RS code can be schematized as follows:

$n=2^m-1$: code length in symbols
$k=n-2t$: number of information symbols
$n-k=2t$: number of check symbols
$d_0=2t+1=d_{min}$: designed distance = minimum distance

Here, t represents the number of correctable error symbols.

C. Encoding of RS codes

If we assume that the parity check polynomial function is $p(x)=c_0+c_1x+c_2x^2+\cdots+ c_{n-k-1}x^{n-k-1}$ and the information polynomial function is $d(x)= c_{n-k}x^{n-k}+ c_{2n-k+1}x^{n-k+1}+\cdots +c_{n-1}x^{n-1}$ at the RS codes with code symbols from $GF(2^m)$ where m is the number of bits per symbol, then the code polynomial can be expressed as

$$c(x)=\sum_{i=0}^{n-1} c_i x^i = d(x)+p(x) \quad (1)$$

Here, the coefficients c_i, $0 \le i \le n-1$ are the elements in $GF(2^m)$. An n symbol vector, (c_0, c_1, $\cdots$, c_{n-1}) is a code data if its corresponding code polynomial $c(x)$ can be obtained by the multiple of $g(x)$ that functions as the polynomial generator as below.

$$d(x)=g(x)q(x)+\gamma(x) \quad (2)$$

Here, $\gamma(x)$ is a remainder and $q(x)$ is an irrelevant quotient.

$$c(x)=p(x)+g(x)q(x)+\gamma(x) \quad (3)$$

If we assume that the negatives of the coefficients of $\gamma(x)$ are the check digits, that is, $p(x)= -\gamma(x)$, it is represented as in the following

$$c(x)=g(x)q(x) \quad (4)$$

This shows that a multiple of $g(x)$ is the code polynomial $c(x)$.

In a *t-error* correcting RS code with length 2^m-1 where α is an original element of $GF(2^m)$, the generator polynomial can be expressed as follows

$$g(x)=\sum_{i=0}^{2t} g_i x^i = (x+\alpha)(x+\alpha^2)\cdots(x+\alpha^{2t}) \quad (5)$$

In the (n, t) RS encoder design, the denseness is a result of the calculation of the check symbols c_i, where i varies between 0 and $2t$-1. This complex calculation can be attained from any RS encoder. However, for a RS encoder, Berlekamp has developed a new algorithm about the bit serial multiplier that can be implemented in an integrated circuit. While Figure 1 shows the typical encoder of Reed Solomon code, Figure 2 represents the serial multiplier for Reed Solomon encoder[11].

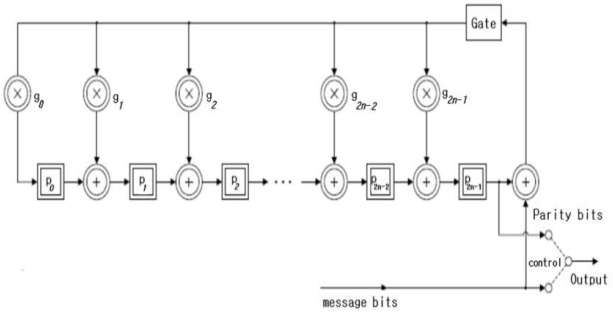

Fig. 1: Reed Solomon encoder

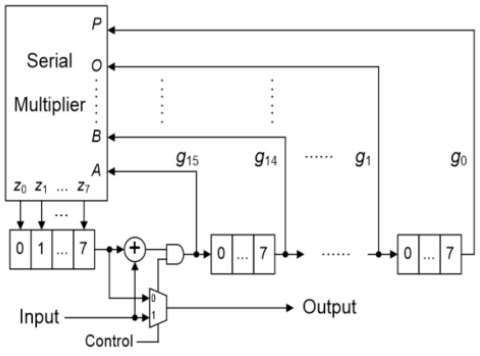

Fig. 2 : Serial multiplier for Reed Solomon encoder

D. Syndrome computation for RS codes

Follow the binary BCH code, if the error pattern polynomial $e(x)$ includes ν errors, the numbers of error location can be represented as follows.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

$$\beta_l = \alpha^{j_l} \qquad 1 \le l \le n \tag{6}$$

The $2t$ syndrome components s_i can be acquired by substituting α_i, $0 \le i \le 2t$, into the received polynomial $r(x)$, and they can be also calculated by dividing $r(x)$ by $x+\alpha_i$ as seen below. The components are either

$$s_i = r(\alpha^i) = \sum_{l=1}^{v} e_{j_l} \alpha^{(j_l) i} \tag{7}$$

or

$$r(x) = q_i(x)(x+\alpha^i) + \gamma_i \tag{8}$$

The result of this division is

$$s_i = r(\alpha^i) = \gamma_i \tag{9}$$

, if α_i is substituted in formula (8).

Therefore, the syndrome computation can be achieved by the syndrome generator division circuit for Reed Solomon codes. Figure 3 displays this.

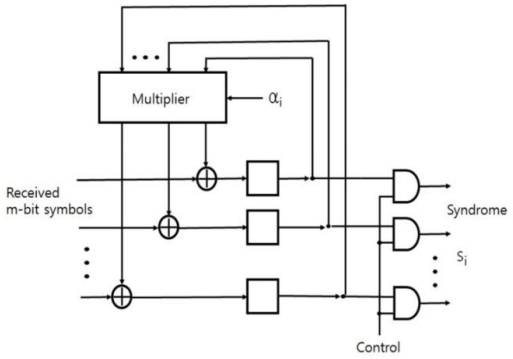

Fig. 3 : Syndrome generator for Reed Solomon codes from GF(2^m)

E. Error evaluator polynomial and error values

If m is the number of symbol bits and α is an original element in $GF(2^m)$, the generator polynomial of the RS code with the code length 2^m-1 and t error correcting capability is expressed by formula (5), then the error pattern caused by the channel noise becomes

$$e(x) = r(x) + c(x) = \sum_{i=0}^{n-1} (r_i + c_i) x^i = \sum_{i=0}^{n-1} e_i x^i \tag{10}$$

If v errors are included in the error pattern polynomial function $e(x)$, we can get

$$e(x) = e_{j_1} x^{j_1} + e_{j_2} x^{j_2} + \cdots + e_{j_v} x^{j_v} \tag{11}$$

the error values e_{jk}, and the error locations x^{jk} must be needed in order to determine error polynomial function $e(x)$.

As with binary BCH codes, the $\sigma(x)$ which is the polynomial for RS code has the information about the error location and can correct v errors. It can be represented as follows

$$\sigma(x) = \prod_{l=1}^{v} (1+\alpha^{j_l} x) = 1 + \sigma_1 x + \sigma_2 x^2 + \cdots + \sigma_{v-1} x^{v-1} + \sigma_v x^v \tag{12}$$

This can also be obtained from Berlekamp's algorithm. Let the syndrome polynomial be

$$s(x) = s_1 x + s_2 x^2 + \cdots + s_v x^v = \sum_{\lambda=1}^{v} s_\lambda x^\lambda \qquad n \le 2t \tag{13}$$

et the $\Omega(x)$ be the polynomial function of the error evaluator, then $\Omega(x)$ can be defined by performing the product between $\sigma(x)$ and $s(x)$ like below

$$\begin{aligned} \Omega(x) &= s(x)\sigma(x) \\ &= 1 + (s_1 + \sigma_1)x + (s_2 + s_1\sigma_1 + \sigma_2)x^2 + \cdots + (s_v + s_{v-1}\sigma_1 + \cdots + \sigma_v)x^v \end{aligned} \tag{14}$$

At each error location, the error evaluator polynomial $\Omega(x)$ is adopted to identify the error value for non-binary RS codes as expressed below. If v errors occur in positions in accordance with the indices $j_1 < j_2 < \ldots\ldots < j_v \le n-1$, then the syndrome components can be expressed as

$$s_\lambda = \sum_{k=1}^{v} e_{j_k} (\alpha^{j_k})^\lambda \qquad 1 \le \lambda \le 2t \tag{15}$$

Here, α^{j_k} are defined as the error location numbers at position indices j_k.

$$s(x) = \sum_{k=1}^{v} \frac{e_{j_k}}{1 + e_{j_k} x} \tag{16}$$

$$\Omega(x) = \sum_{k=1}^{v} e_{j_k} \prod_{l=1, l\ne k}^{v} (1+\alpha^{j_l} x) \tag{17}$$

$$e_{j_m} = \frac{\Omega(\alpha^{-j_m})}{\prod_{l=1, l\ne m}^{v} (1+\alpha^{j_l}\alpha^{-j_m})} \tag{18}$$

The error pattern polynomial $e(x)$ is composed from this error magnitude.

F. Burst error correction

The symbol of the RS code is a non-binary code and has the t error correcting capability. If each symbol is represented by its corresponding m bit byte, $a\ (n, k)$ linear binary code with the parameters of $k=m(2^m-1-2t)$ and $n=m(2^m-1)$ can be obtained. This code can modify any error pattern that affects m bit bytes of t bits or less. It is not important whether there exists one bit error in a byte or there exist several bit errors in a byte. The error is still counted as one byte error. The word that is received is divided into 2^m-1 bytes, and then individual byte can be changed into a symbol from $GF(2^m)$. Therefore, if the error pattern affects bytes below t, it will affect t or fewer symbols in a RS code. This code is called as a multiple phased burst error correcting code. Table 1 displays the different RS code parameters. As seen in Table 1, if the code rate reduces the code redundancy, then the minimum distance and the asymptotic coding gain increase.

Table 1 : Reed-Solomon code parameters

Code length	Minimum distance	Redundancy	Code rate	Coding gain(dB)
(255, 233)	33	14.4	0.87	14.6
(255, 239)	17	6.7	0.94	12.0
(255, 247)	9	3.2	0.97	9.4

The simulations are run on the output BER of some RS codes. Net electrical coding gains are represented 5.3, 4.4, 3.5 for (255, 247), (255, 239) and (255, 233) respectively. At the (255, 239), the code's redundancy is almost twice compared to (255, 247) and the code offers more than 1dB coding gain.

III. RESEARCH APPROACH

The bit error rate is used to analyse the performance of RS code. Figure 4 shows the performance of RS (255, 233) and RS (15, 11). The channel is assumed to be AWGN channel and PSK, respectively.

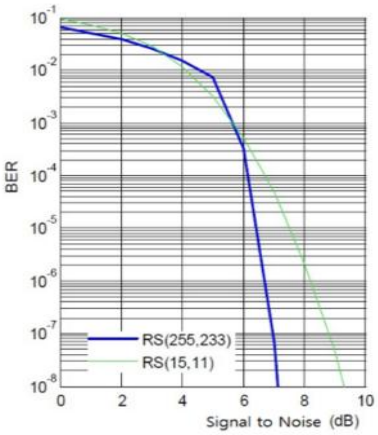

Fig. 4 : Performance of RS (15, 11) and RS (255, 239)

When the length of code word increases, the complexity of calculating and implementation also increases. It is obvious that if the signal power increases, the error rate decreases. If the code length changes from 15 to 255, the error rate decreases in accordance with the signal power increase. Also, the same performance changes between RS (255, 239) and RS (255, 233) are observed. One of data symbols is 239, and the other of data symbols is 233. As the signal power increases, the error rate decreases. It is shown that the coding gain is 0.36 dB for a BER 10^{-5} when the data symbols of the code word changes from 233 to 239.

IV. CONCLUSION

This paper presents the design of (255, 239) Reed Solomon encoder. In order to design a low-complexity encoder, a bit-serial encoder is designed by serializing Reed Solomon encoder, which operates in parallel using a polynomial divider circuit and a bit serial multiplier, in which the adder is placed outside the shift register. The designed bit serial coder is very simple in hardware and has the advantage of avoiding complicated circuit connection as compared to the existing encoder operation in parallel. Also, since the designed encoder operates on the polynomial basis, it solves the disadvantages of the base conversion of bit serial coder operating on dual base and it is very suitable for VLSI because the circuit structure is regular and simple. The performance of Reed-Solomon code is presented using the terms with the noise power density ratio of the signal energy and the bit error rate. (255, 239) Reed Solomon code system can be applicable to the efficient knowledge-based system and smart communication system.

ACKNOWLEDGMENT

Funding for this paper was provided by Namseoul University.

REFERENCES

1. Lin. S, Costello. D, Error Control Coding: Fundamentals and Applications, Pearson Prentice-Hall, 2nd ed, 2004.
2. CCSDS 131.0-B-2, CCSDS Recommended Standard for TM Synchronization and Channel Coding, The Consultative Committee for Space Data Systems, Technical Report, August 2011.
3. Peterson. W, Weldon. E.J, Error- Correcting Codes, MIT Press, Cambridge, Mass, 1972.
4. Berlekamp. E.R., Bit-Serial Reed-Solomon Encoders, IEEE Transactions on Information Theory, 28, 1982: 869-874.
5. Tyagi. R., Srivastava. P.C, Kumar. M., 2 Performance Comparison of RS Code and Turbo Codes for Optical Communication, International Journal of Electronics Engineering, 3(2), 2011: 251– 256.
6. Abdesslam. H, Anas. E, Ahmed. E. A., Mohamed. H, Performance Study of RS (255, 239) and RS (255.233) Used Respectively in DVB-T and NASA, Int. Journal of Engineering Research and Application, 6(11), 2016: 10-14.
7. D. Subhashree, VHDL implementation of Reed-Solomon coding, National Institute of Technology Rourkela, 2011.
8. Lee. H, High-speed VLSI architecture for parallel Reed-Solomon decoder, IEEE transactions on very large scale integration (VLSI) systems, 11(2), 2003: 288-294.
9. Rhee. M.Y, Error Correcting Coding Theory, McGraw-Hill, 1989.
10. L. Borkar. Harshada, and V. N. Bhonge ,Review: Design And Implementation Of Reed Solomon Encoder And Decoder, SSRG International Journal of Electronics and Communication Engineering (SSRG-IJECE) ,2(1) , 2015:29-33.
11. Reed-Solomon (RS) Coding Overview, VOCAL Technologies, Ltd., Rev. 2.28n, 2010.

AUTHORS PROFILE

First Author Wonshik Na He is a professor of Department of Computer Science at Namseoul University in Korea. His research interests include Network Security, Information Protection, Medical Information, Multimedia, U-computing, Wireless LAN. He has a Ph.D. in Computer Engineering from Kyunghee University in Korea.

Second Author Jae-Yeon Choi is a professor in the Department of Information and Communications Engineering at Namseoul University. He received the B.S., M.S., and the Ph.D. degree in electronic communication engineering from Hanyang University, Seoul, Korea (respectively 1985, 1987, 1998). From 1987 to 1989, he was with Samsung Advanced Institute of Technology. From 1989 to 1992, he was with LG Information & Communication Research Center.