Triple Error Correcting Reed Solomon Decoder Design Using Galois Subfield Inverse Calculator And Table ROM

Hyeong-Keon An* ***Reguler Member***, Young-Jin Hong** ***Lifelong Member***

ABSTRACT

A new RS(Reed Solomon) Decoder design method, using Galois Subfield GF(2^4) Multiplier, is described. The Decoder is designed using Normalized error position stored ROM. Here New Inverse Calculator in GF(2^8) is designed, which is simpler and faster than the classical GF(2^8) direct inverse calculator, using the Galois Subfield GF(2^4) Arithmatic operator.

Key Words : RS(Reed Solomon), Syndrome, Encoder, Decoder, Inversion, Error Locator polynomial, Galois Field(GF)

Ⅰ. Introduction

Reed Solomon coding theory is very famous well known nonbinary error correction method for Digital Electronic Devices(Consumer and Communication products.)[3].

In this paper, new RS(Reed Solomon) Decoder, which is correcting 3 symbol errors, design method is proposed using Normalized error position stored ROM[2]. Especially new Inverse calculator in GF (2^8) is implemented using its Galois Subfield GF(2^4) Arithmatic operator. The new Subfield operator is much simpler and faster than before, So More efficient RS Decoder design is Possible[1].

In chapter 2, we briefly described RS(Reed Solomon) ECC algorithm. For example we describe how to calculate syndromes, solve Newtonian identitiy equations.

In chapter 3,we describe the New RS Decoder algorithm which is correcting 3 symbol error in the codeword. Example is showing the algorithm is working well. In MD(Mini Disc Player), DCC, HDTV, Main computer magnetic storage system, this 3 symbol error correcting RS decoder is used.

In chapter 4, the new Inverse calculator, in GF(2^8), design method is described. Dividing can be done using Multiplier and this inversion circuit. Definitely the new circuit, using Galois subfield GF(2^4) arithmetic operator, is much more efficient than the direct GF(2^8) operator. For more clarity, we show inversion example to describe the step of the algorithm.

In chapter 5 conclusions are made. Future works on 4 symbol error correcting RS decoder is briefly mentioned. Also Divider in GF(2^8) design method is also discussed.

Ⅱ. Syndromes and Error Locator polynomial

An RS(Reed Solomon) codes are based on finite fields, often called Galois fields.

In CDP, RSC(32,28), on GF(2^8) field, codes is used and up to 2 symbol errors can be corrected [2].

* Department of Information & Communications Engineering, College of Engineering (hkan@tit.ac.kr)
** Department of Information Security, College of Engineering (gryjhong@tit.ac.kr)
논문번호 : KICS2005-10-423, 접수일자 : 2005년 10월 19일

An RS code with 8bit symbols will use a Galois field GF(2^8), consisting of 256 symbols. In decoding Reed-Solomon code, we should calculate the Syndromes as in equation 1.

Let

$$C(X)= \Sigma_{j=0}^{n-1} C_j X^j$$

Be the Transmitted polynomial, and let

$$r(X)= \Sigma_{j=0}^{n-1} r_j X^j$$

Be the received polynomial. Then error pattern of the channel is

$$E(X)= \Sigma_{j=0}^{n-1} E_j X^j$$

Where E_j(j=0 to n-1) are error values. Here Syndromes are defined as

$$S_i = E(\alpha^i) \ (i=0,1,\cdots, 2t-1) \cdots \quad (1)$$

For t error correction coding.

In this paper, for finding Error values and positions, syndrome calculator shown in Fig. 1 is used[3, 6].

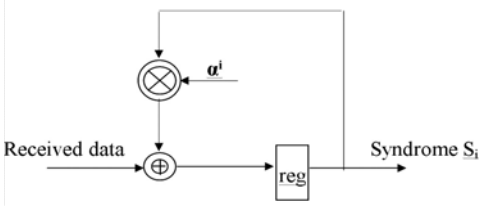

Fig. 1. Syndrome calculator of RS codec

Now if there are t errors, error values are E_n (n=0, 2⋯, t-1) and their positions are α^{jn}(n=0,1, ⋯, t-1).

Then Let

$$\beta_j(j=0,1,\cdots, t-1) = \alpha^{jn}(\ n=0,1,\cdots,t-1)$$

and Error Locator polynomial is defined as

$$\delta(X)=(X-\beta_0)(X-\beta_1)\cdots(X-\beta_{t-1})= \Sigma_{k=0}^{t} X^k\delta_{t-k}\cdots \quad (2)$$

Now Newton's identities are following set of equations.

$$\Sigma_{j=1}^{t} S_{t-j+v}\delta_j = S_{v+t} \ (v=0, 1, 2\cdots, t-1)\cdots \quad (3)$$

These equations are for t error correcting Reed-Solomon codec[4, 5]. If we apply these equations to 3 symbol error correction case (t=3), all the δ_i(i=1, 2, 3) are got as described in the next section[7].

Ⅲ. Triple Error Correcting Reed Solomon Decoder Design

In GF(2^8), If there are 3 symbol errors in the received codeword, we can find 3 error positions and

Error values as follows[7].

Here we use ROM tables as in 2 symbol error case[2].

In this case, Error Locator polynomial is

$$X^3 + \delta_1X^2 + \delta_2X + \delta_3=0 \cdots \quad (4)$$

Here

$\delta_1 = (S_1S_3^2 + S_1^2S_5+S_2^2S_3 + S_0S_2S_5 + S_0S_3S_4 + S_1S_2S_4) \ / \ \chi$,

$\delta_2= (S_0 \ S_4^2 + S_2S_3^2 + S_2^2S_4 + S_0S_3S_5 + S_1S_2S_5+S_1S_3S_4) \ / \ \chi$,

$$\delta_3= (S_3^3 + S_1S_4^2+S_2^2S_5 + S_1S_3S_4 \ ) \ / \ \chi \cdots \quad (4\text{-}1)$$

where $\chi = S_2^3+S_0S_3^2 + S_1^2S_4 + S_0S_2S_4$.

From equation 13, if $X=\delta_1+y$, also if $Œ = \delta_1^2 + \delta_2$, $\delta=\delta_1 \ \delta_2 + \delta_3$ then

We get

$$Y^3 + ŒY + \delta = 0 \cdots \quad (5)$$

If Œ=0, $Y=(\delta)^{1/3}$, otherwise

Let's define $Z_i = Œ^{-1/2} \ Y_i$(i=1,2,3)

$$Z^3 + Z + \delta \ /Œ^{3/2} =0 \cdots \quad (6)$$

From Equation 6, corresponding to Add= $\delta/Œ^{3/2}$ we can construct ROM table of root of equation (6) as in Fig. 2.

<Z_i ROM table>

Address($\delta / \Phi^{3/2}$)	data (Z_i, i=1,2,3)
0	0,1,1
1	•
α	•
α^2	•
•	•
•	•
•	•
α^{239}	α^{157}, α^{181}, α^{156}
•	•
α^{254}	•

Fig. 2. ROM table corresponding to equation 6. When Address = 0, Only 2 roots exist.

Once we find Z_i(i=1,2,3), Y_i(i=1,2,3)=$\Phi^{1/2}Z_i$

So exact Error positions are

$$X_i=Y_i+\delta_1 \quad (i=1,2,3) \cdots \qquad (7)$$

Also Error values are

$$E_i =(S_0\delta_3/X_i+S_1(\delta_1+X_i)+S_2)/(X_i^2+\delta_2) \quad (i=1,2,3) \cdots (8)$$

As we see, there are many GF(2^8) Arithmatic operations to compute the Error values and positions. In Fig. 3. we show how to compute square values. Other operations needed are $\alpha^{1/2}$, α^j/α^i. Allthese operations can be done using only Inverse calculator and Multiplier.

Next section, we show how to compute inversion values and GF(2^8) division.

In Fig. 4 GF(2^8) Inversion circuit diagram is shown using subfield GF(2^4) arithmetic operator,so those circuit can be implemented very efficiently [1].

<Square ROM table>

Address α^i	**Data α^{2i}**
0	**0**
1	**1**
α	α^2
α^2	α^4
α^3	α^6
•	•
•	•
α^{254}	α^{253}

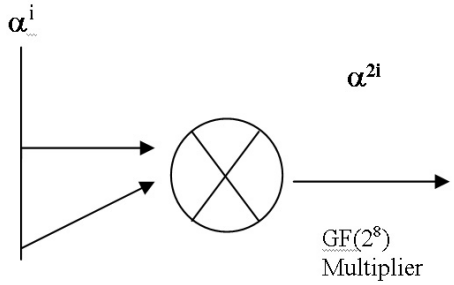

Fig. 3. Square calculator with and without ROM
1) Square calculation using ROM
2) Square calculation with Multiplier

<Example> Triple Error correcting Reed Solomon Decoder example :

From the Transmitter, we send all 0 data so Code polynomial C(x)=0. In Receiver, Received polynomial R(x)= α + αx+ α^2x^2.

In this case, find 3 error values and positions. All code symbols are 8 bits wise so GF(2^8) field elements are used.

<Solution>

We first find Syndromes : $S_0=\alpha^2$, $S_1=\alpha+\alpha^2+\alpha^4=\alpha^{239}$. $S_2=\alpha^{37}$, $S_3=\alpha^{75}$, $S_4=\alpha^{219}$, $S_5=\alpha^{24}$. From Equations (4-1), we find out $\delta_1=\alpha^{198}$, $\delta_2=\alpha^{199}$, $\delta_3=\alpha^3$. So from equations (5), $\Phi=\alpha^{248}$, $\delta=\alpha^{101}$. Hence $\delta/\Phi^{3/2}=\alpha^{239}$.

So from following equation,

$$Z^3 + Z + \alpha^{239} = 0.$$

So Using ROM table in fig. 5, we find that $Z_i=\alpha^{157}$, α^{181}, α^{156}.

Therefore

$$Y_i =\Phi^{1/2} Z_i= \alpha^{26}, \alpha^{50}, \alpha^{25} (i=1,2,3).$$

So from equation (7),

$$X_i=Y_i+\delta_1 \quad (i=1,2,3) = \alpha^0, \alpha^1, \alpha^2.$$

These are 3 correct Error positions and 3 error values are calculated from equation (8), as α, α, α^2. These are also correct 3 error values as we see from received polynomial r(x).

Ⅳ. New GF(2^8) Inverse element calculating circuit design

In this section, we describe how to simplify the Inversion circuit using Galois subfield[1]. The circuit is used for divider HW in RS Codec. Using this and multiplier described in the former Author's paper[2], Most RS Codec circuit can be simplified and faster. In Fig. 5 we draw the New inversion circuit block diagram[1]. Here all arithmetic operationa are done in GF(2^4) field so Dramatically reducing gate counts and computational speed much fasrer than the case in GF(2^8). Multipler design using GF(2^4) Sub field is described in the Author's another paper[2].

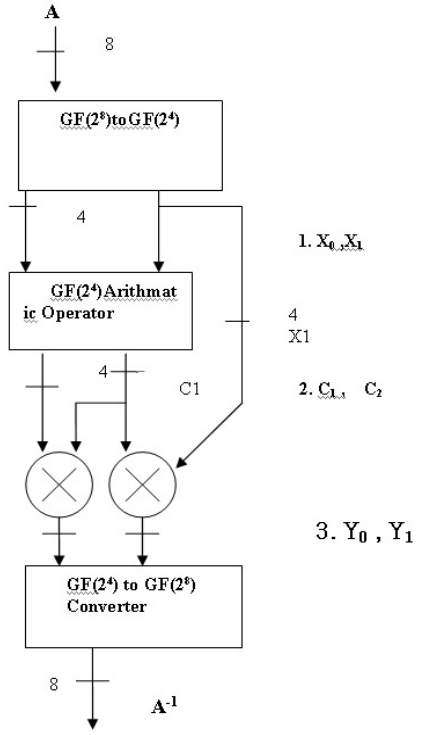

Fig. 4. Inversion Circuit in GF(2^8)
1. X0,X1 each 4 bits
2. C1=((X0+X1)X0+ $\gamma X1^2)^{-1}$,C2=X0+X1

GF(2^8) to GF(2^4) is processed as follows.

Let α^k is in GF(2^8) field as (b_0, b_1, ..., b_7), it can be expressed as $\alpha^k = a + b\beta$ where a and b is in GF(2^4) field and β is in GF(2^8). Here a and b are (z_0,z_1,z_2,z_3) and (z_4,z_5,z_6,z_7) respectively. All b_j, z_j (j=0 to 7) are in GF(2) = (0,1). This means $\alpha^k = \sum_{I=0}^{3} (z_I + \beta z_{I+3})\ \gamma^I$, $\gamma \in$GF(2^8) and $\gamma^4 = \gamma^3+1$ (GF(2^4) Primitive Polynomial).

Then

$$\begin{aligned} Z_0 &= b_0+b_1+b_5 \\ Z_1 &= b_1+b_3+b_5 \\ Z_2 &= b_2+b_3+b_6 \\ Z_3 &= b_1+b_3+b_4+b_6 \\ Z_4 &= b_1+b_2+b_3+b_5+b_6+b_7 \\ Z_5 &= b_2+b_5+b_6 \\ Z_6 &= b_1+b_2+b_3+b_4+b_5+b_6 \\ Z_7 &= b_1+b_3+b_4+b_5 \ \ldots \end{aligned} \tag{9}$$

In the same way, From (9), we find GF(2^4) to GF(2^8) converter equation is, for example

$$\begin{aligned} B_0 &= Z_1 + Z_0 + Z_2 + Z_6 + Z_7 \\ B_1 &= Z_2 + Z_1 + Z_5 \\ B_2 &= Z_3 + Z_5 + Z_7 \\ B_3 &= Z_2 + Z_6 + Z_7 \\ B_4 &= Z_1 + Z_7 \\ B_5 &= Z_5 + Z_6 + Z_7 \\ B_6 &= Z_3 + Z_6 + Z_5 \\ B_7 &= Z_1 + Z_6 + Z_4 + Z_7 \ \ldots \end{aligned} \tag{10}$$

Now A, A^{-1} in GF(2^8) can be expressed as follows.

$$\begin{aligned} A &= X_0+X_1\beta \\ A^{-1} &= Y_0+Y_1\beta \ \ldots \end{aligned} \tag{11}$$

So

$$\begin{aligned} X_0Y_0 + \gamma X_1Y_1 &= 1 \\ X_1\ Y_0 + (\ X_0\ + X_1)\ Y_1 &= 0 \ \cdots \end{aligned} \tag{12}$$

Here X_0, X_1, Y_0, $Y_1 \in$ GF(2^4), $\beta$and $\gamma \in$ GF (2^8) also $\beta^2=\beta+\gamma$, then Y_0, Y_1 are represented as in (13)[1]:

$$\begin{aligned} Y_0 &= (X_0+X_1)/B \\ Y_1 &= X_1/B \\ B &= X_0(X_0+X_1)+\gamma(X_1^2) \ \ldots \end{aligned} \tag{13}$$

Also if X=(x_0, x_1,x_2,x_3), γX^2=(x_2+x_3, x_0 +x_2 +x_3, X_3, x_1 +x_2).

All these are implemented in Fig. 4.

Fig. 5 is a Divider circuit using this inversion circuit.

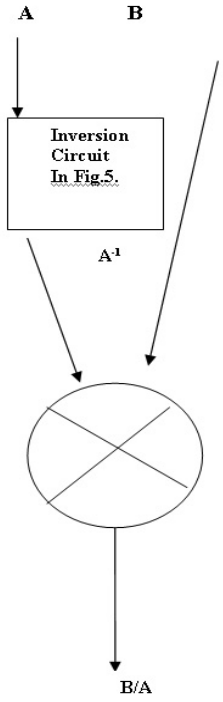

Fig. 5. Divider Circuit in $GF(2^8)$ using Circuit in Fig. 4, where $A,B \in GF(2^8)$.

if we compare the Number of gates when we compute $Y=A^2B \in GF(2^8)$ between two cases GF (2^4) transformation case and $GF(2^8)$ case (Two Multipliers are used),

1) $GF(2^8)$ case

The total number of gates breaks down as follows.

AND gate	EXOR gate
2X64 =128	2X73=146

2) $GF(2^4)$ transformation case

AND	EXOR
$GF(2^8)$ to $GF(2^4)$	13
2Multiplier($GF(2^4)$) 48X2	65X2
$GF(2^4)$ to $GF(2^8)$	13
96	156

So totally 22 gates are saved when we use $GF(2^4)$ transformation method. This cost reduction is even further larger if the computation becomes more complex[1].

<Example>

Let's Find Inverse of α^5, $\alpha^{-5} \in GF(2^8)$ using Subfield $GF(2^4)$ Arithmatic operation.

<Solution>

$A= \alpha^5 \in GF(2^8) = X_0 + X_1 \beta$.

From Eq 9., $X_0 = \alpha^{12}$, $X_1 = \alpha^6 \in GF(2^4)$.

From Eq 13., $Y_0 = \alpha^{14}/(\alpha^{13}+\alpha^{12}\ \alpha^{14}) = \alpha^9$,

Here $\gamma X_1^2 = \alpha^{13}$.

Also $Y_1=1/(\alpha^5+\alpha^7) = \alpha^{-14}= \alpha$.

Now Convert these to element in $GF(2^8)$.

Then $b_0 = b_1 = b_4=b_7=0$ and $b_2=b_3=b_5=b_6=0$.

Hence this b_i (i=0 to 7) represents $\alpha^{250} = \alpha^{-5}$ so Correct.

Ⅴ. Conclusions

With the implementation of the inverse Calculator[1], the divider can be easily implemented with multiplier circuit by using the subfield $GF(2^4)$ arithmetic operation.

The idea presented in the paper simplifies the circuit and performs high speed operation by decreasing the number of logic gates[1]. The drawback of this transformation method is there is no advantage when the arithmatic computation is simple because in this, we should transpose $GF(2^8)$ to $GF(2^4)$ and inverse transpose $GF(2^4)$ to $GF(2^8)$ too.

Also RS 3 Symbol Error correcting Decoder can be implemented by using the circuit of table ROM. And this kind of 3 symbol Error correction RS decoder is used for most of the Current Digital Audio/Video devices, CDP, MP3, MD, HDTV, etc.

Our future works will be 4 symbol error correcting RS decoder, and direct $GF(2^8)$ Divider using also $GF(2^4)$ sub field, resulting in even further greately reducing RS codec HW circuitry[3].

REFERENCES

[1] US patent number 5227992, "Operational Method and Apparatus over GF(2^m) using a Subfield GF($2^{m/2}$)", Man-young Lee, Hyeong-Keon An et al., 1993 Jul. 13

[2] Hyeong-Keon An, "2 Error Correcting RS Decoder design", IDEC Conference Paper, KOEX, October 20-24, 2004

[3] Hyeong-Keon An, TS Joo et al, "The New RS Ecc Codec For Digital Audio and Video", IEEE CES Conference paper, PP. 112-115, 1992

[4] Lee Man Young, "BCH coding and Reed-Solomon Coding theory," 1990, Minumsa (Daewoo Academic Press).

[5] Sunghoon Kwon and Hyunchul Shin, "Anarea-efficient VLSI architecture of Reed-Solomon decoder/encoder for digital VCRs," IEEE Transactions on Consumer Electronics, Vol. 43, No.4, Nov. 1997

[6] Kwang Y.Liu, " Architecture for VLSI design of Reed-Solomon Decoders, "IEEE Transactions on Computers. Vol.33, No.2, Feb. 1984

[7] 岡野博 : ROM used 2 to 3 error correcting BCH Decoder Improvement, 信學技報, AL 82-56 (1982).

[8] Shu Lin, Daniel J. Costello, Jr., "Error Control Coding," Prentice-Hall, pp.240-261 (20044).

An Hyeong-Keon ***Reguler Member***

He received B.Engineering Degree in electrical engineering from Seoul National University, Seoul, KOREA, in 1979 and M.S degree in electrical science from Korea Advanced Institute of Science and Technology, Seoul, Korea in 1981, and the Ph.D. degree in electrical engineering from State University of New York at Stony Brook, NY, USA., in 1988. In 1988, he joined Samsung Electronics Co.Ltd as a Senior Researcher working for designing System LSI for 10 years. From 1998 to 1999, He worked for Telson Electronics Corp. working for CDMA handphone design. In 2000, he joined Tong Myoung University in Busan as a Professor in Dept. Of Information and Telecommunication engineering He has interests in designing CDMA and GSM hand phone and also in System LSI(Non Memory) design. He also operates Venture Comapany for Producing various Mobile phones and GPS/MP3 Engines.

Hong, Young-Jin ***Lifelong Member***

He received the B. S. E. E. degree from Seoul National University. Seoul. Korea, in 1978 and the M. S. E. E. and Ph. D.(E. E.), from the State University of New York at Stony Brook in 1982 and 1985, respectively.

From January 1986 until May 1986 he was with the Department of Electrical Engineering at the State University of New York ay Stony Brook, as an Assistant Professor. In June 1986 he joined LNR Communications, Inc., Hauppauge, NY, where he was a Research Staff Engineer and working on spread spectrum systems and satellite communications. In 1992 he came back to Korea to join Samsung Advanced Institute of Technology (SAIT), where he had been leading several research projects including CT2, VSAT and TDMA cellular basestations for two years. Since then he has broadened the spectrum of his career path to include not only the area of R&D(CTO of Eastel Systems from 1994 through 1997; CTO of Sungil Telecom in the year of 2004) sector but also the business area(executive managing director of SKC&C from 1997 to 2003). He is currently an Associate Professor in the Department of Electrical and Electronics Engineering, Tongmyong University, Busan, Korea. His research interests are in the areas of smart antenna system, adaptive signal processing and communication systems. Dr. Hong is a member of Korean Institute of Communication Sciences, The institute of Electronics Engineers of Korea. he is also a member of IEEE.