리드솔로몬 복호기에서 2개의 오류시, 오류위치를 찾는 최적화 방법

정회원 안 형 근*

Optimizing the Circuit for Finding 2 Error Positions of 2 Error Correcting Reed Solomon Decoder

Hyeong-Keon, An* ***Regular Member***

요 약

본 논문에선 리드 솔로몬 복호기의 2개의 8빗트 심볼 오류정정회로의 에러위치추적기에대한 새로운 설계법을제시한다. 본 설계법을 통해 기존보다 빠르고 훨씬 회로량이 줄어든 적화된 2개의 8 빗트심볼 오류위치 추적기를 설계할수 있었다. 이 리드솔로몬 복호기는 거의 모든 디지털 통신및 가전기기의 데이터 보존장치로 사용되질수 있다. 특히 8빗트 동작을 4빗트동작으로 분화시켜 복호기의 최적화를 이뤘다.

Key Words: Reed-Solomon(RS), Decoder, GF(2^4), Communication, Digital, Error location, Symbol, GF(2^8)

ABSTRACT

In this paper, we show new method to find error locations of 2 eight bit symbol errors for 2 error correcting Reed-Solomon decoder. New design is much faster and has much simpler logic circuit than the former design method. This optimization was possible by partitioning the 8 bit operations into 4 bit arithgmatic and logic operations. This Reed Solomon decoder can be used for data protection of almost all digital communication and consumer electronic devices.

Ⅰ. Introduction

Nowadays almost all Digital electronic devices use Reed Solomon(RS) codec to protect their data. RS encoder is used to channelcode the source data, and is used in transmitter(for communication devices) and recorder(for Digital Audio/Video devices). RS decoder is used to correct the detected error data in receiver(for communication devices) and playback device(for Digital Audio/Video devices)[2].

In this paper, we propose new method to optimize the 2 eight bit symbol error correcting circuit of RS decoder. Almost all RS decoder of current digital devices use 2 ~ 4 eight bit symbol error correcting circuit in the decoder. RS decoder consists of error location finding circuit and error value evaluating circuit. To find error location in the RS codeword, we should solve nonlinear error locator polynomial. It is not easy to solve the nonlinear error locating equation. On the other hand, it is straightforward and easy to find the error values once their locations are found, since the error values can be found by solving the linear systems of equations[1,6]. So the method presented here is showing how to get the solution of non linear error locator polynomial for 2 eight bit symbol error case. To optimize the error

* 동명대학교 정보통신공학과(hkan@tu.ac.kr)
논문번호 : KICS2010-10-468, 접수일자 : 2010년 10월 4일, 최종논문접수일자 : 2011년 1월 4일

location finding circuit, we transform GF(2^8) galois elements to GF(2^4) elements[5], since GF(2^4) operation is much simpler than GF(2^8) element manipulation. Table 1 shows the GF(2^4) elements table, whose primitive polynomial is $X^4+X^3+1=0$[8]. In section 2 we present the old method to get 2 error locations in GF(2^8) field for 2 symbol error correcting RS decoder[7]. In section 3, we present the new method and in section 4, we compared the 2 methods so finding new method is much more optimized than the old method. Finally in section 5 we make our onclusions and also show our future works to make better improvements.

Table 1 GF(2^4) elements table
표 1. GF(2^4) 원소 표

GF(2^4) elements	code(z_0 z_1 z_2 z_3)
0	0000
1(=α^{15})	1000
α	0100
α^2	0010
α^3	0001
α^4	1001
α^5	1101
α^6	1111
α^7	1110
α^8	0111
α^9	1010
α^{10}	0101
α^{11}	1011
α^{12}	1100
α^{13}	0110
α^{14}	0011

From table 1, for example, make $\alpha^{11} \bullet \alpha^{13}=\alpha^{24}=\alpha^9(\because\alpha^{15})$ =1010. Also $\alpha^4+\alpha^5=0100=\alpha$.

Ⅱ. Classical Method to find 2 Error positions of 2 Error correcting Reed Solomon decoder

The error locator polynomial for 2 error case is

$$\sigma(x) = x^2 + \sigma_1 x + \sigma_2 = 0. \tag{1}$$

From Newton's identities

$$\begin{bmatrix}\sigma_1\\ \sigma_2\end{bmatrix} = \begin{bmatrix}S_1 & S_2\\ S_2 & S_3\end{bmatrix}\begin{bmatrix}S_3\\ S_4\end{bmatrix} \tag{2}$$

So from (2),

$$\sigma_1 = \frac{S_2S_3+S_1S_4}{S_2{}^2+S_1S_3} \text{ and} \qquad \sigma_2 = \frac{S_3^2+S_4S_2}{S_1S_3+S_2^2} \tag{3}$$

Now if two error positions are α^i, α^j then

$$\begin{aligned}\sigma_1{}^2/\sigma_2 &= \frac{(S_1S_4+S_2S_3)^2}{(S_2^2+S_1S_3)(S_3^2+S_4S_2)}\\ &= (\alpha^{2i}+\alpha^{2j})/\alpha^{i+j}\\ &= \alpha^{i-j}+\alpha^{j-i}\end{aligned} \tag{4}$$

Using equation (4), we construct the ROM which maps $\alpha^i+\alpha^{-i} \rightarrow \alpha^i \in$ GF(2^8) as in the table 2.

After finding $\sigma_1{}^2/\sigma_2$ value by equation (4), we get α^{i-j} from the ROM data in table 2. Then 2 error positions are as in equation (5),(6).

$$\alpha^i = (\sigma_2 \bullet \alpha^{i-j})^{\frac{1}{2}} \tag{5}$$

Table 2. $\alpha^k+\alpha^{-k}$to α^k Mapping ROM
표 2. $\alpha^k+\alpha^{-k}$를 α^k로 변환하는 리드온리메모리

Address($\alpha^k+\alpha^{-k}$)	Data (α^k)
α^{49}	α^1
α^{98}	α^2
α^{188}	α^3
α^{196}	α^4
•	•
•	•
•	•
•	•
$\alpha^k+\alpha^{-k}$	α^k
•	•
•	•
•	•
•	•
0	α^{128}

$$\alpha^j = (\sigma_1 + \alpha^i) \qquad (6)$$

Also the 2 error values are as in equation (7),(8).

$$E_j = (\alpha^i S_0 + S_1)/\sigma_1 \qquad (7)$$

$$E_i = S_0 + E_j \qquad (8)$$

〈Example〉

In GF(2^8), when the code polynomial C(x)=0 is transmitted to the receiver, the received polynomial is

$$r(x) = \alpha^3 x^2 + \alpha x^4 \qquad (9)$$

So two error values and locations are α^3 at x^2 and α at x^4. Verify these values.

〈sol〉

First calculate syndromes $S_i(i = 1,2,3,4)$.

$$\begin{aligned} &S_1 = r(\alpha) = 0. \\ &S_2 = r(\alpha^2) = \alpha^7 + \alpha^9 = \alpha^{57} \\ &S^3 = r(\alpha^3) = \alpha^9 + \alpha^{13} = \alpha^{109} \\ &S_4 = r(\alpha^4) = \alpha^{11} + \alpha^{17} = \alpha^{202} \end{aligned} \qquad (10)$$

$$\begin{aligned} \sigma_1^2/\sigma_2 &= \frac{(S_1S_4 + S_2S_3)^2}{(S_2^2 + S_1S_3)(S_3^2 + S_4S_2)} \\ &= (\alpha^{166})^2/\alpha^{114}(\alpha^{218} + \alpha^{259}) \\ &= \alpha^{77}/(\alpha^{77} + \alpha^{118}) \\ &= \alpha^{77}/\alpha^{234} = \alpha^{98} \end{aligned} \qquad (11)$$

From table 2, we see $\alpha^{i\text{-}j} = \alpha^2$. So by (5), we find

$$\begin{aligned} \alpha^i &= (\sigma_2 \bullet \alpha^{i-j})^{\frac{1}{2}} = (\frac{S_3^2 + S_4S_2}{S_1S_3 + S_2^2})^{\frac{1}{2}} \bullet (\alpha^2)^{\frac{1}{2}} \\ &= (\alpha^{106} + \alpha^{147})^{\frac{1}{2}} \\ &= \alpha^4 \text{and}\, \alpha^j = \sigma_1 + \alpha^i \\ &= \frac{S_3}{S_2} + \alpha^4 = \alpha^{52} + \alpha^4 = (\alpha^2 + \alpha^4) + \alpha^4 = \alpha^2 \end{aligned}$$

2 Error values are from equation (7) and (8),

$$E_2 = (\alpha^3 + \alpha)\alpha^2/\alpha^{52} = \alpha^{53}/\alpha^{52} = \alpha \text{ and}$$
$$E_1 = \alpha^{51} + \alpha = \alpha^3.$$

All these are correct.

Ⅲ. New Optimized Method to Find 2 Error Positions.

If there are 2 Errors in the code $\in$ GF(2^8), error locator polynomial is as equation (1). Now define $X_1 = x/\sigma_1$, and $K = \sigma_2/(\sigma_1^2)$ then error locator polynomial becomes as equation (12).

$$X_1^2 + X_1 + K = 0 \qquad (12)$$

Now convert (12) to GF(2^4) equation as follows.

$$\begin{aligned} &X_1 = Y_1 + \beta Y_2 \text{ , } K = K_1 + \beta K_2 \\ &\text{where } Y_1 \text{ \& } Y_2 \text{ and } K_1 \text{ \& } K_2 \in GF(2^4). \end{aligned} \qquad (13)$$

Substitute (13) to (12), then

$$\begin{aligned} &Y_1^2 + \gamma Y_2^2 + Y_1 + K_1 + \beta(Y_2^2 + Y_2 + K_2) = 0. \\ &(\because \beta^2 + \beta = \gamma[2]) \end{aligned} \qquad (14)$$

From (18), we see

$$\begin{aligned} &Y_1^2 + \gamma Y_2^2 + Y_1 + K_1 = 0 \text{ and} \\ &Y_2^2 + Y_2 + K_2 = 0. \end{aligned} \qquad (15)$$

This Y_2 can be found using table 1.

Table 3 is made easily because there are only 16 cases of Y_2 for each $\mathbf{K_2}$. the 2 error positions has also following relation. If Y_2 is a solution to (15), Y_2+1 is also a solution to (15) and easily verified by substitution to equation (15) itself. Once Y_2 is found, Y_1 can also be found from equation (15). From (15),

$$\begin{aligned} &Y_1^2 + Y_1 = \mathbf{K'}_1 \text{ and} \\ &\mathbf{K'}_1 = K_1 + \gamma Y_2^2 \end{aligned} \qquad (16)$$

Since Y_2 and K_1 is already known , $\mathbf{K'}_1$ is also known value so equation (16) is easily solved using the ROM shown in table 3. Fig. 1 shows $\mathbf{K}_1$ to $\mathbf{K'}_1$ converter circuit and Y_1 & Y_2 to the final error

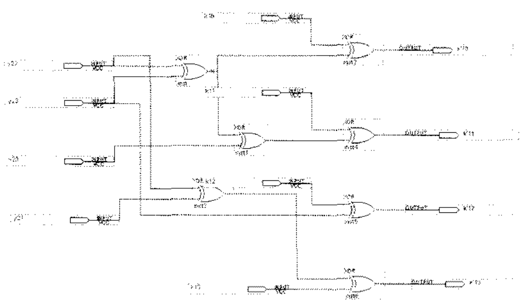

Fig. 1. K_1 to K'_1 Converter, where $K_1=(K_{10}, K_{11}, K_{12}, K_{13})$, $K'=(K'_{10}, K'_{11}, K'_{12}, K'_{13})$ and $Y_2=(Y_{20}, Y_{21}, Y_{22}, Y_{23})$.
그림 1. K_1에서 K'_1으로 바꾸는 회로

Table 3. $GF(2^4)$ 2 error position look up ROM
표 3. $GF(2^4)$ 2개의 오류위치 추적 롬

$K_2=y_i^2+y_i$	y_i
0	0
1	α^{10}
α	α^5
α^2	No Answer
α^3	No Answer
α^4	No Answer
α^5	α^7, α^{13}
α^6	No Answer
α^7	α^3, α^4
α^8	No Answer
α^9	No Answer
α^{10}	α^{11}, α^{14}
α^{11}	α^2, α^9
α^{12}	No Answer
α^{13}	α, α^{12}
α^{14}	α^6,α^8

location finding circuit. Notice one of 2 solutions of Y_2 equation (15) is invalid and can be determined by the $\mathbf{K'}_1$ and the ROM in table 1. If $\mathbf{K'}_1\neq$one of (α^5, α^7, α^{10}, α^{11}, α^{13}, α^{14}), corresponding Y_2 is invalid solution[2]. If we want to find $\mathbf{K'}_1$ **to** Y_1 converter, using primitive polynomial in $GF(2^4)$, $\gamma^4+\gamma^3+1=0$, we get logic equation 17.

$$Y_1=(y_{10},\ y_{11},\ y_{12},\ y_{13}) \\ =[K_0, K_1, K_2, K_3]=\begin{bmatrix}0101\\0000\\0100\\0011\end{bmatrix} \qquad (17)$$

where $\mathbf{K'}_1=[K_0, K_1, K_2, K_3]$. This logic is also much simpler than the $GF(2^8)$ case [7]. Fig. 2 shows the

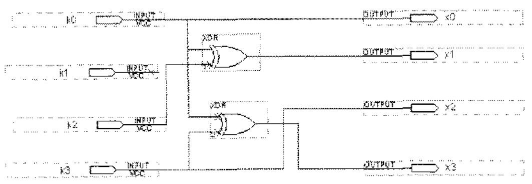

Fig. 2. K to Y_1 converter logic circuit, Y_1 =(x0, x1, x2, x3) and K=(k0, 1, k2, k3) $\in GF(2^4)$
그림 2. K에서 Y_1으로 전환하는 논리회로

circuit for K to Y_1 conve rter and Fig. 1 is $\mathbf{K}_1$ to $\mathbf{K'}_1$ Converter. Also (Y_1, Y_2) to real error position $\in GF(2^8)$ can be found using $GF(2^4)$ to $GF(2^8)$ transform circuit[2]. In this section we saw how our ne w method becomes better than classical meth od. In next section we will compare the curr ent method with the old method.

〈Example 1〉

If received polynomial is $r(x)=\alpha^3x^2+\alpha^4x^5$ for c(x)=0 (transmitted code polynomial) find 2 error locations and error values in $GF(2^8)$.

〈sol〉

$S_1=r(\alpha)=\alpha^{105}$

$S_2=r(\alpha^2)=\alpha^{119}$

$S_3=r(\alpha^3)=\alpha^{30}$

$S_4=r(\alpha^4)=\alpha^{110}$, so from equation (2), we get $\sigma_1=\alpha^{225}$ and $\sigma_2=\alpha^7 \in GF(2^8)$. Also $K=\sigma_2/(\sigma_1^2)=\alpha^{67}$. Now transform into $GF(2^4)$. Then $K=K_1+\beta K_2$, $(K_1,K_2)\in GF(2^4)$. We get $K_1=\alpha^7$, $K_2=\alpha^5$. From table 3, $Y_2=\alpha^7$, α^{13}. One of these Y_2 is invaild. Let's assume $Y_2=\alpha^7$. From (20) $\mathbf{K'}_1=\mathbf{K}_1+\gamma Y_2^2=\alpha^7+\gamma\ \alpha^{14}=(1110)+\gamma$ (0011)=(1110)+(1000)=(0110)= α^{13} from table 1. So from table 3, we see $Y_1=\alpha$ or α^{12}(Notice $\alpha^{12}+1=\alpha$). Therefore 2 Normalized solutions X_1 is

$$X_1=Y_1+\beta Y_2=(\alpha+\beta\alpha^7) \text{ or } (\alpha^{12}+\beta\alpha^7). \qquad (18)$$

Transform (18) into $GF(2^8)$[2] , then

$$X_1=\alpha^{35} \text{ or } \alpha^{32} \in GF(2^8). \qquad (19)$$

So real solution to the equation (err locator polynomial) (1) is $x=\sigma_1 X_1=\alpha^{225}\bullet(\alpha^{35} \text{ or } \alpha^{32})=\alpha^2$ or α^5 ($\because \alpha^{255}=1$). These are 2correct error positions. If we assume $Y_2=\alpha^{13}$, $\mathbf{K'}_1=\alpha^7+\gamma(\alpha^{13})^2=\alpha^7+\gamma\ \alpha^{11}$

($\because \alpha^{15}=1$ for GF(2^4) elements)=$\alpha^7+\alpha^{12}=\alpha^2 \in$ GF(2^4). From table 3 , there is no answer , so $Y_2=\alpha^{13}$ is invaild. Now 2 error values E_2 and E_5 are as follows. From equations (7) and (8)

$E_2=(\alpha^5 S_0+S_1)/\sigma_1$
$=(\alpha^5(\alpha^3+\alpha^4)+\alpha^{105})\,/\,\alpha^{225}$
$=\alpha^3 \in$ GF(2^8). and
$E_5=S_0+E_2=\alpha^3+\alpha^4+\alpha^3$
$=\alpha^4 \in$ GF(2^8). These are correct !.

<Example 2>
K to Y_1 converter :
If $K=\alpha^{13}$, find $Y_1(Y_1^2+Y_1=K_1)$.

<Sol>
From equation (17),
$K=[k_0, k_1, k_2, k_3]=[0,1,1,0]$
$Y_1=(y_{10}, y_{11}, y_{12}, y_{13})$
$=(k_0, k_0+k_2, k_3, k_3+k_0)$
$=(0,1,0,0)=\alpha$ and another solution is
$Y_1+1=(1,1,0,0)=\alpha^{12}$.
This is correct, when we see the ROM table 3.

Ⅳ. Comparison Between the New Method and Classical Method.

There are 2 big advantages over classical method, when we use new method to decode 2 error correcting RS code. First is ROM size reduction. When we compare table 2 and 3, we find 1/16 ROM size reduction. Second advantage is simpler logic for decoding circuit. For example ,to implement K to normalized error position converter, for classical method,we need 12 EXOR gates[7] , but for new method, we need just 2 EXOR gates as shown in equation 17. We summarizes the comparisons in table 4.

Table 4. Comparisons between Classical and New Methods.
표 4. 기존방식과 새방법의 비교표

	Table ROM size to find 2 error positions	Number of EXOR gates to Convert K to normalized error positions
Classical Method	(8bits/word) X (128 words)	12
New Method	(4bits/word) X (16 words)	2

Table 5. Comparisons between GF(2^4) Multiplier and GF(2^8) Multiplier.
표 5. GF(2^4)승산기와 GF(2^8)승산기의 비교

	Number of EXOR Gates	Number of AND gates
GF(2^4) case	15	16
GF(2^8) case	73	64

We see this reduction of logic gate counts not only makes the cost of the decoding circuit more economical but also the speed to find the error locations becomes greately faster because of reduced propagation delay path.

Ⅴ. Conclusions

Here we proposed new method to optimize 2error correcting Reed-Solomon(RS) decoder circuit. HDTV, Digital Audio device,Digital commun ication devices use the RS decoder to prote ct the digital data in the devices[3,7]. Norm ally the RS decoder in the devices corrects 2 ~ 4 symbol errors for 1 codeword. 4error cor recting RS decoder is actually same as 2 err or correcting RS decoder, if code word leng th becomes as half as its original word leng th. So 2 error correcting RS decoder circuit is very Important for the digital devices. In future we will try to optimize the 3 err or correcting RS decoder and the circuit to determine the number of error symbols in one code word of RS decoder circuit.

References

〔1〕 J. Jiang and K. Narayanan, "Iterative soft decision decoding of Reed Solomon codes based on adaptive parity check matrices," in Proc. ISIT, 2004.
〔2〕 안형근, "리드솔로몬 복호기의 에러값을 구하기 위한 새로운 고속의 산술논리연산장치의 설계에 대해",대한전자공학회논문지(통신TC편),제46권 제4호, pp.45-51, 2009. April.
〔3〕 Joschi Brauchle, Ralf Koetter; A Systematic Reed Solomon Encoder with Arbitrary Parity

Positions, IEEE GLOBECOM 2009 Proceedings.

[4] T.K. Moon, Error Correction Coding: Mathematical Methods and Algorithms, Hoboken, NJ: John Wiley & Sons, Inc., 2005.

[5] 안형근, "New Efficient Design of Reed Solomon Encoder Which has Arbitrary Parity Positions Without Galois Field Multiplier", pp.984-990, Vol.35, No6, 2010년 6월호, 한국통신학회논문지

[6] M. L.Cury,A.Skjellum, H.L. Ward, "Accelerating Reed-Solomon coding in RAID systems with GPUs", in Parallel and Distributed Processing,2008,IPDPS 2008. IEEE International Symposium on,2008

[7] 안형근, "디지탈오디오/비디오, 통신용 전자기기를 위한 Reed-Solomon Codec설계에 대해", pp.13-18, TC편제42호, 2005년11월, 대한전자공학회지

[8] Shu Lin, Daniel J. Costello, Jr,. " Error Control Coding," Prentice-Hall, PP 240~261(2004)

안 형 근 (Hyeong-Keon, An) 정회원

1979년 서울대학교 전기공학과 학사

1981년 KAIST 전기 및 전자공학과 석사

1988년 뉴욕주립대학교 전기공학과 박사

1988~1998 삼성전자 수석연구원

1998~1999 텔슨전자 이사

2000~현재 동명대학교 정보통신공학과 교수

<관심분야> 통신, 신호처리, 반도체, OLED / LED display 조명장치