리드솔로몬 복호기에서 오류갯수를 계산하는 처리기의 산술논리연산장치 회로 최적화설계

정회원 안 형 근*

Design Optimization of the Arithmatic Logic Unit Circuit for the Processor to Determine the Number of Errors in the Reed Solomon Decoder

Hyeong-Keon, An* *Regular Member*

요 약

본 논문에선 리드 솔로몬 복호기의 오류갯수를 판별하는 마이크로콘트롤러의 새로운 설계법을 제시한다. 본 설계법을 통해 기존보다 빠르고 훨씬 회로량이 줄어든 최적화된 오류갯수 판별기용 산술논리연산장치회로를 설계할 수 있었다. 이 리드솔로몬 복호기는 거의 모든 디지털 통신및 가전기기의 데이터 보존기기의 보호장치로 사용되어 질 수 있다. 여기서는 제곱계산회로의 최소화가 가능해 병렬처리를 통해 오류갯수 판별기의 최적화를 이룰 수 있었다.

Key Words : Reed-Solomon(RS), Decoder, GF(2^4), Square computing, Digital, Number of Errors, Symbol

ABSTRACT

In this paper, we show new method to find number of errors in the Reed-Solomon decoder. New design is much faster and has much simpler logic circuit than the former design method. This optimization was possible by very simplified square calculating circuit and parallel processing. The microcontroller of this Reed Solomon decoder can be used for data protection of almost all digital communication and consumer electronic devices.

Ⅰ. Introduction

Reed - Solomon codes have since found important applications from deep-space communication to consumer electronics. They are prominently used in consumer electronics such as CDs, DVDs, Blu-ray Discs, in data transmission technologies such as DSL, in broad cast systems such as DVB and ATSC, and in computers.[2]

In this paper, the optimization means that gate counts of the proposed circuit is much smaller and the speed of it becomes much faster than the circuit by former method. Here cost function comes from speed and complexity of the circuit. The shortage of the new method is that we should transform the GF(2^8) finite field elements to GF(2^4) elements. But we need not go back to GF(2^8) field because we just need to detect zero crossing of the determinant of the characteristic matrix.[7] we propose new method to optimize the arithmatic logic unit for the processor of RS decoder, which is finding the number of errors in

* 동명대학교 정보통신공학과 (hkan@tu.ac.kr)
논문번호 : KICS2011-05-223, 접수일자 : 2011년 5월 19일, 최종논문접수일자 : 2011년 11월 11일

one codeword. To start decoding, we should first calculate the number of errors in the codeword and then 2ndly we are to find the error positions and error values. To find the number of errors we should compute the determinant value of the characteristic matrix. It is very time consumming and needs very complicated circuit to get the determinant value[1,6]. So the method presented here is showing how to get the determinant value of characteristic matrix for eight bit non binary symbol error. To optimize the arithmatic logic unit circuit of the processor ,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. Equation 1 shows the nth order characteristic matrix which is checking whether the number of errors in the codeword is greater than. In section 2 we present the flow steps to determine the number of errors in the codeword in (2^8) field[7]. In section 3, we present the processor structure and in section 4, we showed squaring and multiplying circuits of the processor.

In section 5 we compared old design and new design for 3 error determining case. Finally in section 6 we make our onclusios and show our future works.

$$A_n = \begin{matrix} S_1 & S_2 & \cdot & \cdot & S_n \\ S_2 & S_3 & \cdot & \cdot & S_{n+1} \\ \cdot & & & & \cdot \\ \cdot & & & & \cdot \\ S_n & S_{n+1} & \cdot & \cdot & S_{2n-1} \end{matrix} \quad (1)$$

In Equation (1), A_n is Nth order Characteristic matrix and S_k's are kth order syndromes.

Ⅱ. Determination of Number of Errors in the Codeword

Number of errors in one RS codeword($\in GF(2^8)$) are determined by det (A_n). If there are n errors in 1 RS codeword, det($\mathbf{A}_n$)$\neq$0 and det(A_{n+1})=0[5]. This is shown in Fig.1 flowchart. In Fig.1, we see

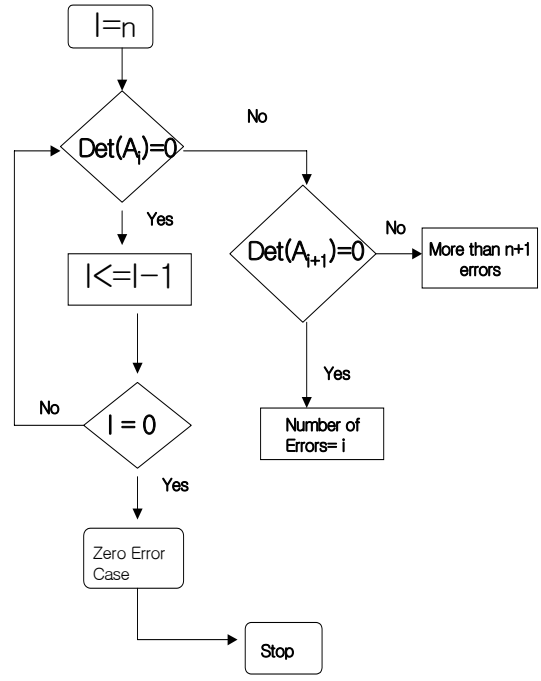

Fig. 1. Flow chart for determining number of errors in 1 RS codeword in case number of errors is less than or equal to n
그림 1. 리드 솔로몬 1코드 워드 내 오류 개수가 n개 이하일 때의 오류 개수 판정 플로우차트

that for 2 errors in one RS codeword,

$$\begin{aligned} Det(A_2) &= S_1S_3 + S_2^2 \neq 0 \text{ and} \\ Det(A_3) &= S_5(S_1S_3 + S_2^2) + S_3^3 + S_4^2S_1 = 0 \end{aligned} \quad (2)$$

For 3 errors in one RS codeword,

$$\begin{aligned} Det(A_3) &= S_5(S_1S_3 + S_2^2) + S_3^3 + S_4^2 S_1 \neq 0 \text{ and} \\ Det(A_4) &= S_1\Upsilon_1 + S_2\Upsilon_2 + S_3\Upsilon_3 + S_4\Upsilon_4 = 0 \end{aligned} \quad (3)$$

Ⅲ. Optimized Processor Structure for Determining the Number of Errors in the RS Codeword

Basic ALU structure of the processor comes from $Det(A_2) = S_1S_3 + S_2^2$ calculation since it contains multiply and square computing and all the higher order determinants of charact eristic matrices contain these multiply and square computing.

Let

$C = A \times B$, and $D = A^2$

where

$C, D, A, B \in GF(2^8)$ (4)

Here if

$C = C0 + \beta C1$, C1 and $C0 \in GF(2^4)$

Also

$D = 0 + \beta D1$, D0 and $D1 \in GF(2^4)$

Then

$$C0 = A0B0 + A1B1\ \gamma \text{ and}$$
$$C1 = A0B1 + A1B0 + A1B1 \quad (5)$$

Also

$$D0 = A0^2 + A1^2\ \gamma\ ,\ D1 = A1^2 \quad (6)$$

Equation (5) can be expressed as follows.

$$C = C0 + \beta(A0B0+(A0+A1)(B0+B1)) \quad (7)$$

From equation (6), (7), we see that we just need γ multiplier, squarer,3 mutipliers and 4 adders in $GF(2^4)$ field to compute the (5), (6) equations simulta neously so to speedup the processor computing[3,6].

The γ multiplier and squarer circuit is very simple as shown in fig 2. The Processor ALU and its bus configurations for both this optimu m structure and former structure in $GF(2^8)$ field are shown in Fig3 and 4.

The proposed processor is definitely much faster and logically simpler than the former processor so we call it optimized processor to find out the number of errors in the RS codeword[1].

In Fig.3 we see that all buses are consisted by two 4 bit wide buses and there are more than 4 execution units so need 3 bit instruction decoder to generate more than 4 enable signals.

In Fig. 4, we know we need just one bit to select one of the execution units(Multiplier and Adder). Also In fig.3, MUL means 3GF(24)

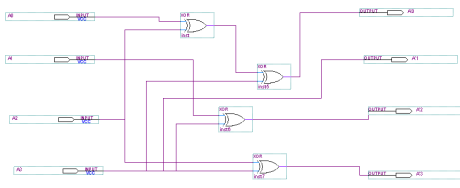

Fig. 2. Squaring circuit in $GF(2^4)$
그림 2. $GF(2^4)$장에서 제곱을 계산하는 회로

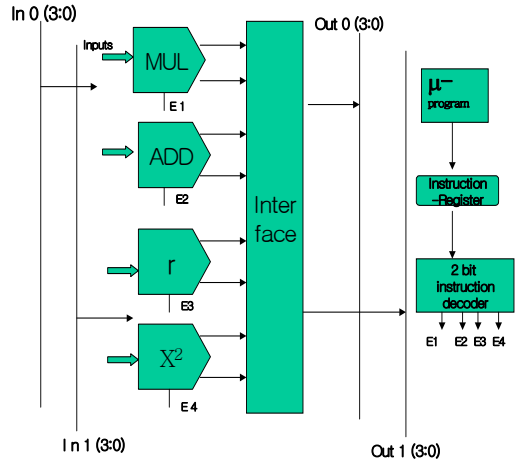

Fig. 3. Optimized Processor structure
그림 3. 최적화된 처리장치 구조

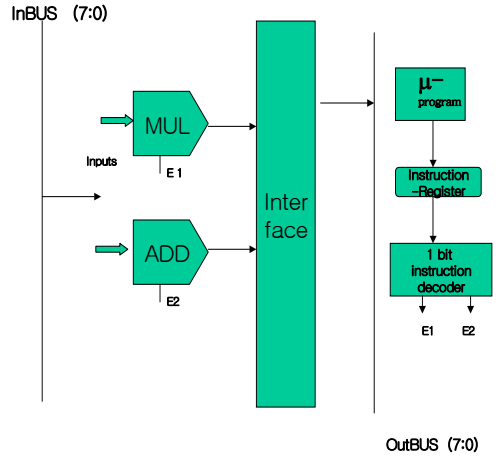

Fig. 4. Old Processor Structure
그림 4. 옛날 방식의 처리기

multipliers and ADD means 4GF(2^4) adders. In Fig. 4, MUL is 1GF(2^8) multiplier and ADD is 1GF(2^8) adder.

The gate counts of Fig. 3 and 4. ALUs are in Table 1. Table 1 shows new processor is smaller than the former one.

In Table 1, we see that new processor is not only much faster but also gatecount is m uch smaller than that of former processor[3].

Table 1. gate counts comparison between old and new processor
표 1. 새 방식 및 옛날방식의 처리기의 gate수

	New Processor	Old Processor
GF(2^8) to GF((2^4)	13	
Multipliers	93(31X3)	137
ɣ Multiplier	1	
Square circuit	4	
Adder	16(4X4)	8

Ⅳ. Performance Evaluation of the Processor for 2 Error Case

For 2 error case of the RS Decoder, we need to calculate determinants of 2nd order and 3rd order characteristic matrices. From equation (2),

$$Det(A_2) = S_1S_3 + S_2^2 \neq 0$$
$$Det(A_3) = S_5(S_1S_3 + S_2^2) + S_3^3 + S_4^2S_1 = 0$$

To compute using new processor, If

$$S_k = S_{k0} + \beta S_{k1} \quad (k=1,2,3,n) \qquad (8)$$

<Det (A_2) computing>

Then 1st step :

$S_{10}S_{30} + S_{11}S_{31}$ ɣ + ɣS_{21}^2(LSB 4bits),

$S_{10}S_{31} + S_{21}^2$ (MSB 4bits)

2nd step:

1st step result + S_{20}^2(LSB 4bits),

1st step result + $S_{11}S_{30} + S_{11}S_{31}$(MSB 4bits),

so we need 2 steps, and each step consumes the propagation delay of 1GF(2^4) multiplier (let's call it m).

To compute using former processor,

1st step : $S_1S_3$is computed.

2nd step : $S_1S_3 + S_2^2$ is computed.

So we need 2 steps, and each step consumes the propagation delay of GF(2^8) multiplier (let's call it M). Here M ≥ 2.5 m , so the speed of new processor is more than 2.5 times faster than that of the former one.

<Example> If transmitted code word is all zeroes (0,0,0 .....,0) and received codeword is (0, α,0,0,0,0,0,α,0,...,0) then there are 2 errors, and prove it.

<Proof> $S_1 = \alpha^{193} \in GF(2^8) = \alpha^{10} + \beta\alpha^6$,

$S_2 = \alpha^{130} = \alpha + \beta\alpha^3$, $S_3 = \alpha^{244} = \alpha + \beta\alpha^9$,

$S_4 = \alpha^4 = \alpha^3 + \beta\alpha^{14}$, $S_5 = \alpha^{72} = \alpha^{10} + \beta\alpha^6$.

Now using new processor we compute as in table 2. If we use old processor in GF(2^8), we have only multiplier and adder, and we can't compute squaring and multipliering sim ultaneously Also we need 2 more sequences tha n the new processorrequirng 8 steps to compute det

Table. 2. Computing Det (A_2) and Det(A_3) using new processor (In ( ), number of steps)
표 2. 새로운 연산 처리장치를 이용한 Det (A_2) 와 Det(A_3)의 계산 (( )안은 필요한 연산단계의 수)

Mnemonic	$C_0 + \beta C_1$ / C_0, C_1	Operation
Mul_n_Square (2)	α^7, α^{10}	$Det(A_2)=S_1S^3+S_2^2$
Mul (1)	0 , α^6	$(S_1S_3+S_2^2)$ S_6
Square (0)	α^{11}, α^3	S_3^2
Mul_n_Square (2)	(α^9,1) and (α^{12},α^{13})	S_3^3 and S_4^2
Mul_n_Add (1)	(α^9,α^8)+(α^9,1) =(0, α^6)	$S_4^2S_1+S_3^3$
ADD (0)	(0,α^6)+(0,α^6) = (0, 0)	$(S_1S_3+S_2^2)$ $S_6S_4^2+S_1+S_3^3$

(A_3).

In table, we see that to compute Det (A_2) and Det(A_3), we need 6 steps, so delay time is 6M (GF(2^4) multiplier delay time). Here we neglect adder delay and squaring delay comparing with multiplier. total delay of old processor is 8M (GF(2^8) multiplier delay is M = 2.5m) is much longer than new processor delay 6 m. This means New processor speed is 3 times faster than the old processor[4,5]. Also we see that Det (A_2)$\neq$0 and Det(A_3)=0 so there are two errors in the received codeword.

Ⅴ. Performance Evaluation of the Pro Cessor for 3 Error Case

For 3 error case of the RS Decoder, we need to calculate determinants of 3rd order and 4th order characteristicmatrices. From equation(3), Det (A_3) $\neq$ 0 and Det(A_4) =0. In Example 2, we see that New processor is much faster than old processor for three error case.

<Example 2>

If transmitted codeword is all zeroes(0,0,0...,0) and received codeword is (0,α,0,0,0,0,0,α,1,0,....0), Prove that there are 3 errors in received codeword.

Sol:

$S_1 = \alpha^2, S_2 = \alpha^{15}+\alpha^{16}+\alpha^3=\alpha^{182}$,

$S_3 = \alpha^{21}, S_4 = \alpha^5+\alpha^{29}+\alpha^{32} = \alpha^{197}$,

$S_5 = \alpha^{75}, S_6 = \alpha^{86}, S_7 = \alpha^{56}+\alpha^{50}+\alpha^8 = \alpha^{221}$

So, $Det(A_3) = S_5(S_1S_3 + S_2^2)+S_3^3+S_4^2S_1$

$= \alpha^{75}(\alpha^2\alpha^{21}+\alpha^{109})+\alpha^{63}+\alpha^{139}\alpha^2$

$= \alpha^{10}\alpha^{75}+\alpha^{63}+\alpha^{141} = \alpha^{85}+\alpha^{20} = \alpha^{182} \neq 0$

and $Det(A_4) = S_1\Upsilon_1+S_2\Upsilon_2+S_3\Upsilon_3+S_4\Upsilon_4$

$= \alpha^2\alpha^{214}+\alpha^{182}+\alpha^{198}+\alpha^{21}\alpha^{244}+\alpha^{197}\alpha^{198}$

$= \alpha^{216}+\alpha^{125}+\alpha^{10}+ \alpha^{140} = (00000000) = 0$

There are 3 errors in the received word[6,7].

From table 3 , we need 8 steps to get Υ_1. In same way, Υ_2, Υ_3 need 12 steps and Υ_4 needs 8 steps. So to get Det(A_4), we need 4 more Mul_n_Add operations (4steps are required) resulting totally in 44 steps.

If we use the former processor to get Det(A_4) value, we need also 44 steps but speed of new processor is 2.5 times faster than the former one since new processor uses GF(2^4) multiplier instead of GF(2^8) multiplier[10]. In Fig. 5 we show delay comparison between two processors to calculate the determinant values of characteristicmatrices.

Table. 3. Computing Υ_1 of Det(A_4) using new processor (In ( ),number of steps)
표 3. 새로운 연산처리장치를 이용한 Det(A_4) 의 Υ_1 계산 (( ) 안은 필요한 연산단계의 수)

Mnemonic	Value ∈ GF(28)	Operation
Mul_n_Square (2)	α^{96} ,α^{150}	S_3S_5 , S_5^2
Mul_n_Square (2)	α^{139}, α^{62}	S_4^2, $S_3S_5S_7$
Mul_n_Square (2)	α^{172}, α^{225}	S_6^2, S_5^3
Mul_n_Add (1)	α^{114}	$S_4^2S_7 + S_5^3$
Mul_n_Add (1)	α^{33}	$S_6^2S_3+S_4^2S_7+ S_5^3$
ADD (0)	α^{214}	$\Upsilon_1=S_3S_5S_7+S_5^3$ $+S_6^2S_3 + S_4^2S_7$

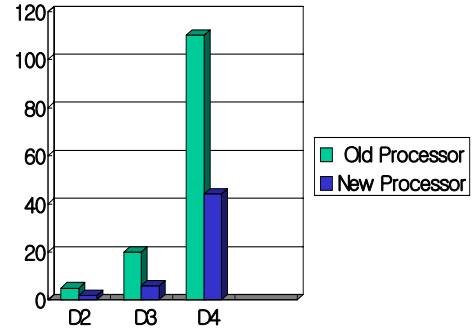

Fig. 5. Delay comparison between form er and new Processors. (Di is delay to calculate the determinant of ith order characteristic matrix, Det(A_i) Ya x is: Micro sec.
그림 5. 두 프로세서의 지연시간 Di 비교 (Di 는 Det(A_i)를 계산하는 시간): Y축 단위 (1000 나노초)

In Fig. 5., we see that as the number of errors detected is increasing, speed of new processor becomes more and more faster than the former one

Ⅵ. Conclusion

Here we proposed new method to optimize the processor design to find the number of errors in the received Reed-Solomon codeword[3,7]. Parallel processing and application of Subfield theory ($GF(2^8)$ to $GF(2^4)$) simultaneously is the main difference between new design presented here and all former design.[3] In RS decoder, this determination is essential. We show in this paper, new processor is much faster than the former design since parallel processing(Multiplication and Squaring) and shorter critical path of $GF(2^4)$ multiplier than that of $GF(2^8)$ multiplier (about 2.5 times)[9].

We know also number of gates for $GF(2^4)$ multiplier is 31 and that of $GF(2^8)$ multiplier is 137.[1,8]

References

〔1〕 안형근, "Optimizing the Circuit for finding 2 Error Positions of 2 error Correcting Reed Solomon Decoder," *한국통신학회논문지*, pp 8~13 ,2011 January.

〔2〕 J.Jiang and K. Narayanan, "Iterative soft decision decoding of Reed Solommon codes based on adaptive paritycheck matrices," in Proc. ISIT, 2004.

〔3〕 Think Silicon Ltd, "Galois Field Multiplier Generator Version 1.0", 2008

〔4〕 Joschi Brauchle, Ralf Koetter; A Systematic Reed-Solomon Encoder with Arbitrary Parity Positions, IEEE GLOBECOM, 2009 Proceedings.

〔5〕 T.K. Moon, Error Correction Coding: Mathematical Methods and Algorithms, Hoboken, NJ: John Wiley & Sons,Inc.,2005.

〔6〕 안형근, "New Efficient Design of Reed Solomon Encoder Which has Arbitrary Parity Positions Without Galois Field Multiplier", *한국통신학회논문지*, pp.984-990, Vol.35, No6

〔7〕 Ming Jang, Bin Nemat, "Implementation of Switch box Using a Subfield", USP7,532,721, 2009

〔8〕 Yosef Stein,"Compact Galois Field Multiplier Engine", USP 7,177,891, Feb 13, 2007

〔9〕 Pusan Patel, "Parallel Multiplier Designs for the Galois/Counter Mode of Operation", Univ. of Wateroo Canada, Dep. of EECS, 2008 MS Thesis.

〔10〕 D. Canright, "Avoid Mask Re-use in Galois Multipliers", Applied Math., Naval Post-graduate School, Monterey CA 93943, USA, 26 November 2008

안 형 근 (Hyeong-Keon, An) 정회원

1979년 서울대학교 전기공학과 학사 졸업

1981년 KAIST 전기 및 전자공학과 석사졸업

1988년 뉴욕주립대학교 전기공학과 박사 졸업

1988년~1998년 삼성전자 수석연구원

1998년~1999년 텔슨전자 이사

2000년~ 현재 동명대학교 정보통신공학과 교수

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