에러 트래핑 방법을 이용한 효율적인 비이진 코드의 디코더 설계

°정 일 석, 강 창 언
연세대학교 전자공학과

Design of an Efficient Nonbinary Decoder Using the Error-Trapping Method

Il Suk Jeong, Chang Eon Kang
Dept. of Electronic Eng. Yonsei Univ.

Abstract

The error-trapping decoder is the simplest way of decoding cyclic codes satisfying R<1/t, where t is the maximum number of errors to be corrected and R is the code rate.

Here the error-trapping decoder for use on nonbinary cyclic codes not satisfying R<1/t is modified.

In this paper, the error-trapping decoder is constructed for the (15,11)-2-error correcting Reed-Solomon code using the concept of covering monomial.

The experimental results have shown good agreement with the theoretical results.

1. 서론

순환코드의 한 디코딩 방법인 Error-Trapping 방법은 짧은 코드나(코드의 길이 : ~32심볼) 혹은 적은 에러를 정정하는(2~3개 심볼 에러 정정, R<1/t) 긴 코드에서는 매우 효율적인 방법이다. [1][2][7]

특히 Error-Trapping 방법은 BCH 코드나 Reed-Solomon 코드를 디코딩하는 데 널리 쓰이고 있는 Berlekamp 디코더가 가지는 Galois Field 곱셈 및 나눗셈을 필요로하지 않아 회로를 간단히 구성할 수있다.

그러나 코드 레이트 R<1/t를 만족하지 않는 코드까지 확장하기 위해 Error-Trapping 디코더를 변형해야 하며, 특히 Kasami는 Covering 다항식을 이용하여 binary 코드에 대한 변형된 Error-Trapping 디코더를 제시했다.[4] 이런 연관성으로 최근에는 nonbinary 순환 코드를 설계하기 위하여 Covering 다항식 대신, 더 축소된 개념인 Covering 단항식이 도입되었다. [1]

본 연구에서는 Covering 단항식 개념을 도입하여 (15,11)-2-에러 정정 Reed-Solomon 코드의 디코더를 Error-Trapping 방법으로 설계했다.

2. 이론 해석

(1) Error-Trapping 방법

코드 레이트 R<1/t를 만족하는 t-에러 정정 순환 코드를 생각할 때, v(X)를 전송 코드워드라 하고, r(X)를 수신된 코드워드라고 한다.

이 때 통신 채널에 의해 야기된 에러의 형태는 e(X)=r(X)+v(X)로 나타난다. 이 경우 수신된 코드워드 r(X)의 신드롬(syndrome) s(X)는 e(X)를 생성 다항식 g(X)로 나눈 나머지이다. 즉 식(1)과 같이 나타난다.

$$e(X)=q(X)g(X)+s(X) \quad \text{------(1)}$$

에러 e(X)가 r(X)의 n-k 패리티 검사(parity check) 부분에 한정될 때, e(X)의 차수는 n-k 보다 작아져 식(1)에서 몫 q(X)=0이 되고 e(X)=s(X)가 된다. 즉 r(X)의 에러가 n-k 패리티 검사 부분에 존재하면, r(X)의 신드롬은 에러와 같은 형태가 되어, 단순히 r(X)의 패리티 부분에서 신드롬의 값을 빼주면 된다.

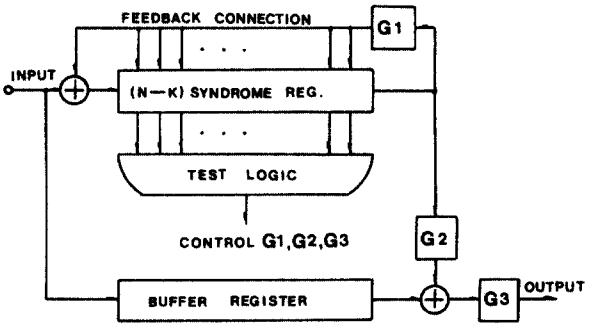
만약 에러의 형태가 r(X)의 n-k 패리티 검사 부분에 한정되지 않을 경우에, r(X)를 순환 이동시켜 에러의 형태가 n-k 패리티 검사 부분에 한정되게 한 후, 같은 방법으로 정정이 된다.

그림(1)에는 위의 방법을 토대로 해서 n-k 신드롬 레지스터를 사용하여 Error-Trapping 디코더를 구성한 것이다.

(2) Reed-Solomon 코드

Reed-Solomon(RS) 코드는 순환 심볼 정정 코드로서, 채널을 통하여 코드워드의 전송시 발생하는 binary burst 에러를 정정할 수 있다.

RS 코드는 Galois Field 심볼의 블락 구조로



그림(1) 일반적인 Error-Trapping 디코더

되어 있으며, 각 심볼은 Galois Field GF(2^m)의 element이다.

RS 코드의 매개 변수는 다음과 같다.

심볼당 비트(bit) 수 : m

코드워드의 길이 : $n=2^m-1$

정정할 수 있는 최대 에러 수 : t

최소거리(mininum distance) : $d=2t+1$

패리티 체크 심볼의 수 : $2t$

정보 심볼의 수 : $k=n-2t$

(15,11)-2-에러 정정 RS 코드에서는 $m=4, n=15, t=2, d=5, 2t=4, k=11$이 된다.

RS 코드의 생성 다항식은 식(2)로 주어지며 본 연구에서는 회로를 간단히 하기위해 $b=2$를 선택했다.

$$g(X)=\prod_{j=b}^{b+2t-1}(X-\alpha^j)=\sum_{i=0}^{2t}g_iX^i \quad \text{-------(2)}$$

(3) Covering 단항식을 이용한 에러 정정 방법

에러 다항식 $e(X)$를 정보 부분과 패리티 검사 부분으로 나누어 나타내면 식(3)과 같다.

$$e(X)=e_I(X)+e_p(X) \quad \text{----------(3)}$$

여기서 $e_I(X)=e_{n-k}X^{n-k}+\cdots+e_{n-1}X^{n-1}$, $e_p(X)=e_0+e_1X+\cdots+e_{n-k-1}X^{n-k-1}$이다.

만약 $e_I(X)=\varepsilon X^{n-k+a}$ ($\varepsilon\neq 0, 0\le a<k$)의 단항식 형태로 두면 $e_I(X)$는 $n-k$ 차 이상이므로 $g(X)$로 나누면 식(4)와 같다.

$$e_I(X)=q(X)g(X)+\varepsilon\rho_a(X) \quad \text{-----(4)}$$

여기서 $\rho_a(X)=\rho_{a,0}+\rho_{a,1}X+\cdots+\rho_{a,n-k-1}X^{n-k-1}$이다. 따라서 신드롬은 식(5)와 같이 된다.

$$s(X)=\varepsilon\rho_a(X)+e_p(X) \quad \text{----------(5)}$$

식(4),(5)로 부터 정보 구간 내의 에러 패턴 $e(X)$를 구할 수 있다. 따라서 에러 다항식은 다음과 같이 된다.

$$e(X)=s(X)+\varepsilon\rho_a(X)+\varepsilon X^{n-k+a} \quad \text{------(6)}$$

여기서 $s(X)+\varepsilon\rho_a(X)$는 단항식 εX^{n-k+a}와 열치할 때 패리티 검사 구간 내에 존재하는 에러 형태이다.

임의의 i ($0\le i\le n-k-1$)에 대해 $u_{a,i}=\rho_{a,i}^{-1}s=\varepsilon+\rho_{a,i}^{-1}e_i$라 둘 때, $u_a(X)$는 다음과 같이 정의한다.

$$u_a(X)=u_{a,0}+u_{a,1}X+\cdots+u_{a,n-k-1}X^{n-k-1} \quad \text{------(7)}$$

만약 $t-1$ 이하의 에러가 패리티 검사 부분에 있다면 $e_p(X)$는 $t-1$ 이하의 영이 아닌 상수를 가지고 $u_a(X)$는 ε와 동일한 $n-k-t+1$개 이상의 상수를 가진다.

식(4)에 의해 $e(X)$의 정보 부분 혹은 $e(X)$의 순환 이동의 정보 부분이 단지 X^{n-k+a}에 영이 아닌 상수를 가진다면 단항식 X^{n-k+a} ($0\le a<k$)는 에러 단항식 $e(X)$를 covering 한다.

설계코자 하는 디코더는 $S=\{0\}\cup\{X^{n-k+a}, 0<a<k\}$인 Covering 단항식 set가 필요하며 이것은 t이하의 영이 아닌 상수를 가진 모든 에러 $e(X)$를 covering한다.

zero 단항식은 $e(X)$의 정보 부분 혹은 순환 이동의 정보 부분이 zero인 에러 $e(X)$를 covering한다.

(4) 바이진 코드의 Error-Trapping 디코딩 과정

Covering set S가 주어졌을 때, 디코딩 과정은 다음과 같다. 여기서 신드롬 $s(X)$의 Hamming Weight $w[s(X)]$는 $s(X)$의 영이 아닌 상수를 말한다.

[제1단계] 신드롬 $s(X)$와 모든 a에 대한 $u_a(X)$를 계산 한다. 만약 $w[s(X)]<t$ 이면 $e(X)=s(X)$이고 error를 정정한다.

[제2단계] $w[s(X)]>t$ 일때 어떤 a에 대해서 $u_a(X)$의 $n-k-t+1$개 이상의 상수가 ε와 동일하다면 $e(X)=\varepsilon X^{n-k+a}+s(X)-\varepsilon\rho_a(X)$가 되고 에러를 정정한다.

[제3단계] 위의 [제1단계], [제2단계]가 성립하지 않는다면 $1<i<n$에 대해 $s^{(i)}(X)=Xs^{(i-1)}(X) \bmod g(X)$를 계산한 후 위의 과정을 되풀이 한다.

만약 n번 순환 이동 후 에러가 정정되지 않는다면 정정 불가능한 에러 패턴이 검출된 것이다.

3. (15,11) Reed-Solomon 코드의 디코더 설계

(1) 인코더 설계

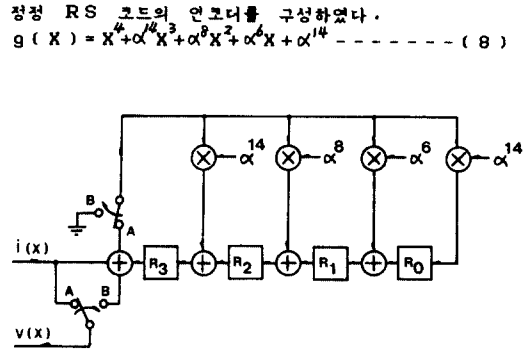
통신 채널상에 코드워드를 발생시키기 위해 인코더를 설계한다. GF(2)의 primitive element를 α라 두면 primitive irreducible 다항식이 $f(X)=X^4+X+1$ 이므로 $\alpha^4=\alpha+1$이 성립한다. 이를 바탕으로 GF(2)의 Field element를 표현하면 표(1)과 같이 된다.

GF element	Notation	Binary Notation
0	0	0000
1	1	0001
α^1	2	0010
α^2	4	0100
α^3	8	1000
α^4	3	0011
α^5	6	0110
α^6	12	1100
α^7	11	1011
α^8	5	0101
α^9	10	1010
α^{10}	7	0111
α^{11}	14	1110
α^{12}	15	1111
α^{13}	13	1101
α^{14}	9	1001

표(1) GF(16)의 element

(15,11) RS 코드의 생성 다항식은 식(8)과 같이되며, 이를 바탕으로 그림(2)에 (15,11)-2-에러 정정 RS 코드의 인코더를 구성하였다.

$$g(X)=X^4+\alpha^{14}X^3+\alpha^8X^2+\alpha^6X+\alpha^{14} \quad \text{-------(8)}$$



그림(2) (15,11) RS 코드의 인코더

(2) 디코더 설계

2-에러 정정 (15,11) RS 코드의 covering set는 $\{0, X^7\}$으로 주어진다. 어떤 2개의 심볼 에러 패턴 $e_iX^i+e_jX^j (i<j)$를 생각하면 다음과 같이 covering 된다.

1 $j-i \leq 3$ zero 다항식에 의해 covering

2 $4 \leq j-i \leq 7$ X에 의해 covering

3 $j-i \geq 8$ 순환 이동에 의해 covering

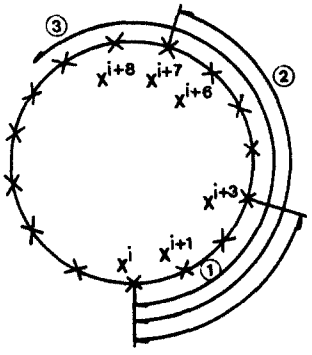
그림(3)에 위의 covering 범위를 도시하였다.

X을 생성 다항식 g(X)로 나눈 나머지는 다음과 같다.

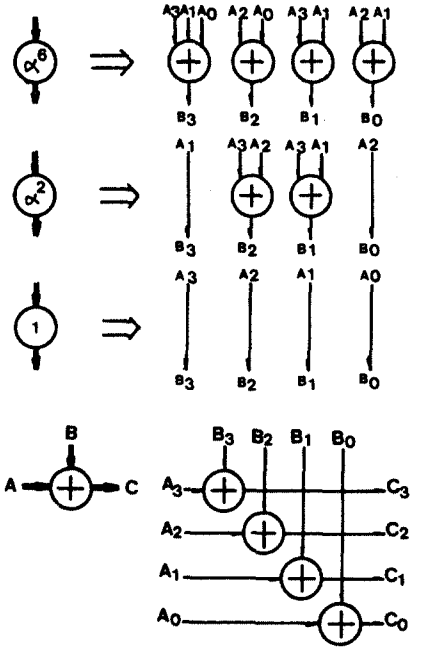
$$\rho(X)=\alpha^{10}X^3+\alpha^{13}X^2+\alpha^{13}X+\alpha^{11} \quad \cdots\cdots\cdots (9)$$

$n-k-t+1=3$ 이어서, $U_i=e_i+\rho_i\rho_{i+1}^{-1}e_{i+1}$ $(0 \leq i \leq 3)$으로 두면 3개의 U_i가 zero열 때 두개의 연속적인 U_i가 zero이어서 디코딩 과정을 단순화 시킨다.

GF(2)의 곱과 덧셈의 실현은 그림(4)에 나타낸 예와 같이 된다.



그림(3) Covering 범위



그림(4) GF(16) 의 곱과 덧셈의 실현 예

위의 이론을 바탕으로 하여 설계한 디코더를 그림(5)에 도시하였으며 디코딩 과정은 다음과 같다.

[제1단계] 채널로 부터 들어오는 15개의 심볼은 버퍼 레지스터와 신드롬 레지스터로 동시에 들어오며 이 수행 과정 동안 G0, G1의 A port를 'ON' 시킨다.

[제2단계] 16번째 shift 부터 45번째 shift 까지 G0이 'OFF' 되고, G1의 B port를 'ON' 시킨다.

1) 만약 적어도 두개의 e_i가 '0' 이면 T1=1이 되고 G2를 'ON' 시켜 에러를 정정한다.

2) 만약 T2=1이면(연속적인 두개의 U_i가 zero) G3, G4, G5, G6를 'ON' 시켜 4번의 cycle을 수행하여 에러를 정정한다.

4번의 cycle 후, 위 과정을 재개한다.

[제3단계] 45번의 shift가 끝난 뒤, 신드롬 레지스터의 값이 '0' 이 되지 않으면 에러를 정정하지 못한 것으로 수신 코드워드를 버리게 된다. 위의 과정이 끝나면 15 cycle 동안 G7이 'ON' 되고 나머지 gate는 'OFF' 되어 정정된 코드워드가 빠져 나간다.

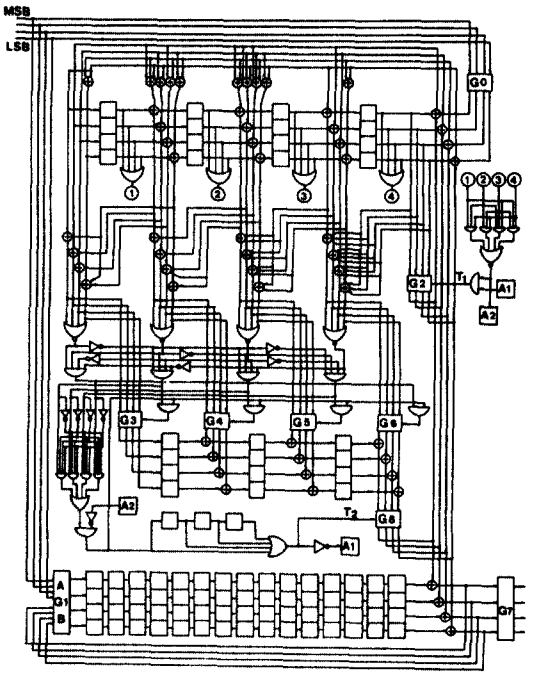
하나의 수신 코드워드를 디코딩하는 데는 60 cycle이걸린다. 그러나 [제1단계]와 [제2단계]는 동시에 수행 가능하므로 하나의 코드워드를 디코딩하는 데는 45 cycle이 필요하다.

(3) 웨이트 분포 및 Performance

BCH 채널을 따라 전송하는 에러는 랜덤하고, 서로 무관하게 발생할 때, (15,11) RS 코드의 웨이트 분포를 표(2)에, Performance를 그림(6)에 구했다.

(4) 실험 및 고찰

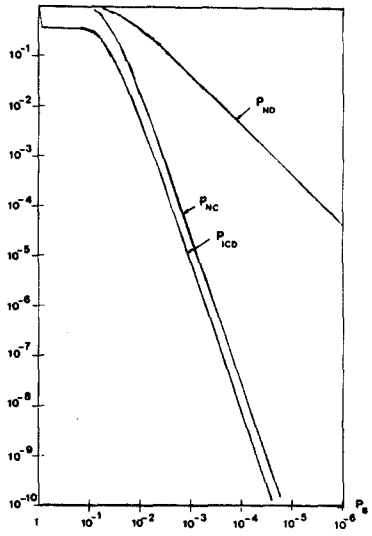
디코더를 구성한 후, 그림(7)와 같이 마이크로 컴퓨터(Apple-II)와 연결하여 동작시켰다.



그림(5) (15,11) RS 코드의 디코더

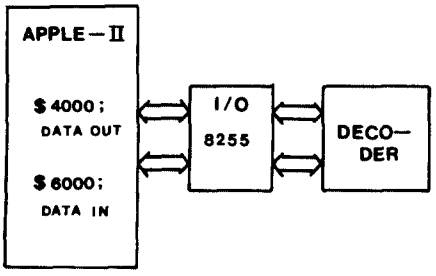
Weight	Number
0	1
1	0
2	0
3	0
4	0
5	45045
6	825825
7	1.689188E+07
8	2.511476E+08
9	2.936183E+09
10	2.642313E+10
11	1.801594E+11
12	9.007961E+11
13	3.118141E+12
14	6.68173E+12
15	6.681731E+12

표(2) (15,11) RS 코드의 웨이트 분포



그림(6) Performce 분석

P_B : 1bit의 에러가 날 확률
P_{ND} : not decoding 에러
P_{NC} : 정정 않될 확률
P_{ICD} : 다른 코드워드로 잘못 디코딩될 에러 확률



그림(7) 실험 장치도

레지스터의 clock과 clear는 Apple-II의 clock(1MHz)를 사용하여 만들었고 각 gate들은 software로 제어했다. I/O는 8255를 사용하여 구성했으며 입력 data는 컴퓨터 시뮬레이션에 의해 구한 코드워드에 에러를 발생시켜 한심볼씩(4bit) parallel로 전송했다.

1개의 코드워드를 디코딩하는 데는 1.5㎳sec가 소요되었으며 실험 결과 2개 이하의 에러가 발생한 경우 에러가 모두 정정되었다. 또한 8개의 연속적인 binary burst 에러도 정정되었다.

4. 결론

(15,11)-2-에러 정정 Reed-Solomon 코드의 디코더를 Covering 단항식 개념을 도입하여 Error-Trapping 방법으로 구성했다. 그 결과 에러의 확율을 줄일 수 있었으며, 2개 이하의 에러와 8bit의 binary burst 에러가 발생한 경우에 완전히 에러가 정정됨을 알 수 있었다.

* 참 고 문 헌 *

1. Shu Lin, Error Control Coding, Prentice-Hall, 1983.
2. Peterson and Weldon, Error-Correcting Codes, 2nd ed., MIT Press,1972.
3. R.E.Blahut, Theory and Practice of Error Control Codes, 1983.
4. T.Kasami,"An decoding procedure for multiple-error correcting cyclic codes," IEEE Trans. Inform. Theory, vol.IT-10, pp.134-138, 1964.
5. V.K.Wei,"An Error-Trapping Decoder for Nonbinary cyclic codes," IEEE Trans. on Inform. Theory, vol.IT-5, pp.114-123, 1959.
6. 정 연호, BCH 코드를 사용한 에러 정정에 관한 연구, 연세대학교 대학원, 1980.
7. 홍 대식, Error-Trapping 방법을 이용한 Reed-Solomon 코드의 인코더 및 디코더 설계, 연세대학교 대학원, 1985.