직접복호법을 이용한 (255,239) BCH 부호의 복호기

°조 용 석 박 차 생 이 만 영
한양대학교

Hardware Implementation of (255,239) BCH decoder using Direct Decoding Method

Cho Yong Suk Park Cha Sang Rhee Man Young
Hanyang University

* ABSTRACT *

Direct Decoding Method for binary BCH codes which directly can find error location number from syndrome without calculating error locator polynomial is presented in this paper

The (255,239) BCH decoder is implemented using TTL logics. It is shown from our results that this decoder can be implemented with relatively simple hardware.

Ⅰ. 서 론

BCH 부호는 무작위적(random)으로 발생하는 여러개의 산발오류를 정정할 수 있는 다중오류정정 부호로써 많은 디지탈 통신망 및 컴퓨터 기억장치 등에 널리 사용되고 있는 실용상 매우 중요한 부호이다.

BCH 부호의 복호과정은 일반적으로 다음과 같이 4단계로 나눌 수 있다. [1]

(1) 오증(syndrome)의 계산

(2) 오류위치다항식(error location polynomial)을 구함

(3) 오류위치(error location number)를 구함

(4) 오류정정(error correction)

Chien[2]은 위 4단계중 (2)와(3)을 생략할 수 있는,즉 오류위치다항식을 구하지 않고 오증으로 부터 직접 오류위치를 찾아내어 오류를 정정할 수 있는 직접복호법을 제안하였다.

본 논문에서는 이 직접복호법을 연구 분석하고 이를 이용하여 2중 오류정정 (255,239)BCH 부호의 복호기를 직접 TTL IC로 장치화 함으로써 이 복호법이 기존의 복호법 보다 훨씬 간단한 Hardware로 장치화 될 수 있음을 보였다. 또한 2중 오류생성기를 제작하여 시험해 봄으로써 이 복호기가 2개 이하의 모든 오류를 정정할 수 있음을 입증하였다.

Ⅱ. 2원 BCH 부호의 복호

2원 BCH 부호의 복호는 오류가 발생한 위치를 찾아 내는 것 이다. 즉 오류위치만 알게되면 그 위치의 비트를 0이면 1로 1이면 0으로 바꿈으로써 오류를 정정할 수 있다.

부호다항식을 c(x) , 오류다항식을 e(x) , 수신다항식을 r(x)라 하면

$$r(x) = c(x) + e(x) \quad \text{---------- (1)}$$

가 되고 α, α^3, ... , α^{2t-1} 이 생성다항식 g(x)의 근 일 때 오증 S_j 는 $c(\alpha^j) = 0$, $j = 1,3,....2t-1$ 이므로

$$S_j = r(\alpha^j) = e(\alpha^j) , \quad j = 1,3,....,2t-1 \quad \text{------ (2)}$$

이다. $u(1 \leq t \leq)$개의 오류가 $i_1, i_2, ..., i_u$ 위치에서 발생하였다고 가정하면 오류다항식 e(x)는

$$e(x) = x^{i_1} + x^{i_2} + ... + x^{i_u} \quad \text{---------- (3)}$$

가 되며 따라서 오증은

$$S_j = r(\alpha^j) = e(\alpha^j)$$
$$= (\alpha^j)^{i_1} + (\alpha^j)^{i_2} + ... + (\alpha^j)^{i_u}$$
$$, \quad j = 1, 3, ... , 2t-1 \quad \text{---------- (4)}$$

가 된다. 여기에서 오류위치번호 $\alpha^{i_k} = X_k$, $1 \leq k \leq u$라 놓고 식 (4)를 풀어쓰면 다음과 같은 t개의 방정식이 된다.

$$\begin{aligned} S_1 &= X_1 + X_2 + ... + X_u \\ S_3 &= X_1^3 + X_2^3 + ... + X_u^3 \\ S_5 &= X_1^5 + X_2^5 + ... + X_u^5 \quad \text{--------- (5)} \\ &\vdots \\ S_{2t-1} &= X_1^{2t-1} + X_2^{2t-1} + ... + X_u^{2t-1} \end{aligned}$$

2원 BCH 부호의 복호는 이 t개의 방정식으로부터 미지수 $X_1, X_2, \ldots, X_u$ $(1 \le u \le t)$를 찾아내는 것이다. 그러나 이 방정식들은 비선형(nonlinear)방정식이기 때문에 직접 해를 구하는 것은 매우 어렵다.

Peterson[3] 은 2원 BCH 부호의 경우에 다음과 같은 오류위치다항식(error location polynomial)을 도입하여 해를 구하는 방법을 처음 제안하였다.

$$\sigma(x) = (1+X_1x)(1+X_2x)\ldots\ldots(1+X_ux) = \sigma_0 + \sigma_1x + \sigma_2x^2 + \ldots\ldots + \sigma_ux^u \quad \text{----- (6)}$$

위 식의 σ_i , $0 \le i \le u$ 와 오증 S_j와의 관계는 Newton의 항등식에 의하여 다음과 같이 표현된다. [1]

$$\begin{aligned} &S_1 + \sigma_1 = 0 \\ &S_3 + \sigma_1S_2 + \sigma_2S_1 + \sigma_3 = 0 \\ &S_5 + \sigma_1S_4 + \sigma_2S_3 + \sigma_3S_2 + \sigma_4S_1 + \sigma_5 = 0 \\ &\quad\vdots \end{aligned} \quad \text{------ (7)}$$

위의 방정식들은 선형(linear)방정식들 이므로 비교적 쉽게 해를 구할 수 있다. 이 해를 구하는 방법으로는 Peterson-Gorenstein-Zierler의 방법, Berlekamp-Massey의 반복 알고리즘, Euclid알고리즘 등 여러 방법이 연구되어져 왔다.[4]

$\sigma(x)$의 계수를 구한 다음에는 이 $\sigma(x)$의 근을 구하여야 한다. Chien[2]은 이 $\sigma(x)$의 근을 구하는 데 있어서 방정식 $\sigma(x)=0$를 직접 풀지않고 순회성질을 이용하여 근을 구하는 간단한 방법을 제안하였다. 오류위치다항식의 근을 구하기 위하여는

$$\sigma(x) = 1 + \sigma_1x + \sigma_2x^2 + \ldots\ldots + \sigma_tx^t \quad \text{------ (8)}$$

에 $GF(2^m)$상의 모든 원소를 대입하여 그 값이 0이 되는지 검사하면된다.

$$\begin{aligned} &\sigma(1) = 1 + \sigma_1 + \sigma_2 + \ldots\ldots + \sigma_t = 0 \\ &\sigma(\alpha) = 1 + \sigma_1\alpha + \sigma_2\alpha^2 + \ldots\ldots + \sigma_t\alpha^t = 0 \\ &\sigma(\alpha^2) = 1 + \sigma_1\alpha^2 + \sigma_2(\alpha^2)^2 + \ldots\ldots + \sigma_t(\alpha^t)^t = 0 \\ &\quad\vdots \end{aligned}$$

위 식을 살펴 보면

$$\sigma(1) = 1 + \sigma_1 + \sigma_2 + \ldots\ldots + \sigma_t \quad \text{---------- (9)}$$

에 σ_1대신 $\sigma_1\alpha$를, σ_2대신 $\sigma_2\alpha^2$을, ... , σ_t대신 $\sigma_t\alpha^t$를 계속 대입해 나가면서 그 값이 0인지를 검사하는 것 이다. 즉 다시쓰면

$$\sum_{k=1}^{t} \sigma_k = 1 \quad \text{---------- (10)}$$

를 만족하는지 검사하는 것이 된다. 이상과 같은 알고리즘을 그림으로 나타내면 그림.1과같다.

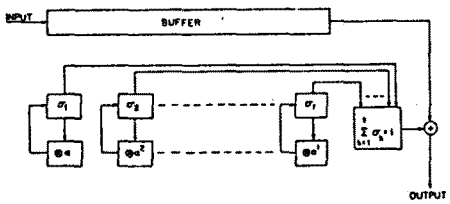


그림.1 Chien의 오류 탐지회로

Ⅲ. 2원 BCH 부호의 직접복호법

2원 BCH 부호의 경우 식(7)의 Newton의 항등식을 행렬로 표현하면 다음과 같다.

$$\begin{bmatrix} 1 & 0 & 0 & \ldots\ldots & 0 \\ S_2 & S_1 & 1 & \ldots\ldots & 0 \\ S_4 & S_3 & S_2 & \ldots\ldots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ S_{2t-2} & S_{2t-3} & S_{2t-4} & \ldots\ldots & S_{t-1} \end{bmatrix} \cdot \begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \vdots \\ \sigma_t \end{bmatrix} = \begin{bmatrix} S_1 \\ S_3 \\ S_5 \\ \vdots \\ S_{2t-1} \end{bmatrix}$$

$$A \cdot \sigma = B \quad \text{---------- (11)}$$

Peterson[3]은 오류가 t혹은 t-1개 발생하였을 경우 행렬식 $|A| \neq 0$ 이고 t-2개 이하가 발생하였을 경우 $|A| \neq 0$ 임을 증명하였다. 이 성질을 이용하면 실제 발생한 오류의 갯수를 판정할 수 있다.

행렬식 $|A|$가 0이 아니라면 σ_k, $k = 1,2,\ldots,t$ 는 다음과 같이 쓸 수 있다.

$$\sigma_k = \frac{1}{|A|} \sum_{i=1}^{t} S_{2i-1} A_{i,k}, \quad k = 1,2,\ldots,t \quad \text{---(12)}$$

여기에서 $A_{i,k}$ $(k=1,2,\ldots,t)$는 $|A|$의 cofactor이다. 위 식을 식(10)에 대입하면 다음과 같다.

$$\sum_{k=1}^{t} \{\frac{1}{|A|} \sum_{i=1}^{t} S_{2i-1}A_{i,k}\} = 1 \quad \text{------- (13)}$$

$|A|$는 k와 무관하므로 식(13)을 정리하면

$$\sum_{k=1}^{t} \sum_{i=1}^{t} S_{2i-1}A_{i,k} + |A| = 0 \quad \text{------ (14)}$$

가 되고 이것을 행렬식으로 바꾸면 다음과 같이 된다.

$$\Delta = \begin{vmatrix} 1 & 1 & 1 & \cdots\cdots & 1 \\ S_1 & 1 & 0 & \cdots\cdots & 0 \\ S_3 & S_2 & S_1 & \cdots\cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ S_{2t-1} & S_{2t-2} & S_{2t-3} & \cdots\cdots & S_{t-1} \end{vmatrix} = 0 \quad \text{----(15)}$$

이상에서와 같이 식(10)을 만족하는 오류위치는 식(15)를 만족하게 된다. 식(15)는 오증(syndrome)만의 함수이므로 복호는 오류위치다항식과 그 근을 구하는 단계를 거치지 않고 오증으로 부터 직접 오류위치를 구하여 오류를 정정할 수 있다.

Ⅳ. 2중 오류정정 (255,239) BCH 부호의 복호기

2중 오류정정인 경우 식(15)는

$$\Delta = \begin{vmatrix} 1 & 1 & 1 \\ S_1 & 1 & 0 \\ S_3 & S_2 & S_1 \end{vmatrix} = S_1 + S_2^2 + S_1^3 + S_3 = 0 \quad \text{--- (16)}$$

가 된다. 이 식을 이용하여 2중 오류정정 BCH부호의 복호기를 설계하면 그림.2와 같다. $S_1 + S_1^2 + S_1^3$은 유한체 내에서의 연산이므로 상당히 복잡하게 된다. 본 논문에서는 회로소자를 줄이기 위하여 이 계산을 ROM 으로 처리하였다.

비교기에서는 S_3 와 $S_1 + S_1^2 + S_1^3$ 이 같을 때에만 1을 출력하여 그 위치의 수신비트와 더하여져 오류를 정정하게 된다. 그림.3에 TTL IC를 이용하여 장치화한 (255,239) BCH부호의 복호기를 사진으로 나타내었다.

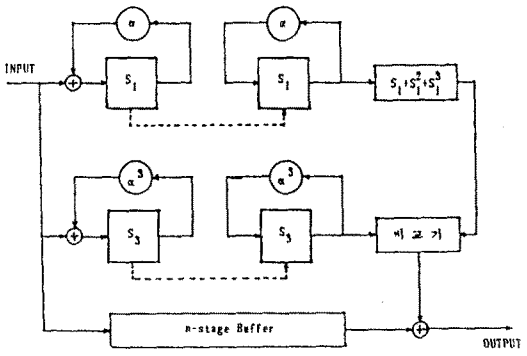


그림.2 2중 오류정정 BCH 부호의 복호기

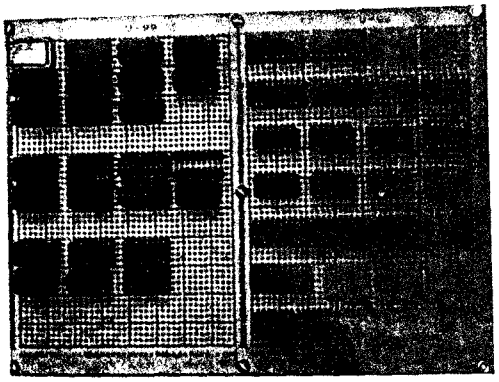


그림.3 (255,239) BCH 부호의 복호기

기존의 방법으로 장치화한 복호기[5]와 본 논문의 복호기에 소요된 IC gate 들의 수를 비교해 보면 표.1과 같다.

	기존의 복호기	본 논문의 복호기
D-FF	48	48
AND	16	16
OR	7	7
NOT	1	1
Buffer	2 (128 bit)	2 (128 bit)
EX-OR	59	49
ROM	1 (512K bit)	1 (2K bit)

표.1 IC gate 수 비교

위의 표에서 보듯이 거의 비슷한 수의 게이트들이 사용되었으나 본 논문의 복호기에서는 ROM 의 용량이 크게 줄어들어 Hardware 적으로 매우 간단해졌음을 알 수 있다.

시험 예로써 c(x)=0 를 전송하였다고 가정하였을 때 단일오류와 2중 오류가 발생하였을 경우를 각각 생각해보자.

예) 1 $e(x) = x^{252} = r(x)$

$S_1 = r(\alpha) = \alpha^{252}$

$S_3 = r(\alpha^3) = \alpha^{246}$

	255	254	253	252	251 .....
S_1	α^{252}	α^{253}	α^{254}	1	α .....
S_1	α^{246}	α^{249}	α^{252}	1	α^3 .....

252 위치에서 보면 $\Delta = S_1 + S_1^2 + S_1^3 = 1 = S_3$이므로 비교기의 출력이 1이되어 이 위치의 오류를 정정한다.

예) 2 $e(x) = x^{250} + x^{248} = r(x)$

$S_1 = r(\alpha) = \alpha^{250} + \alpha^{248} = \alpha^{43}$

$S_3 = r(\alpha^3) = \alpha^{240} + \alpha^{234} = \alpha^{170}$

	255	251	250	249	248	247 ...
S_1	α^{43}	α^{47}	α^{48}	α^{49}	α^{50}	α^{51} ..
S_3	α^{170}	α^{182}	α^{185}	α^{188}	α^{191}	α^{194} ..

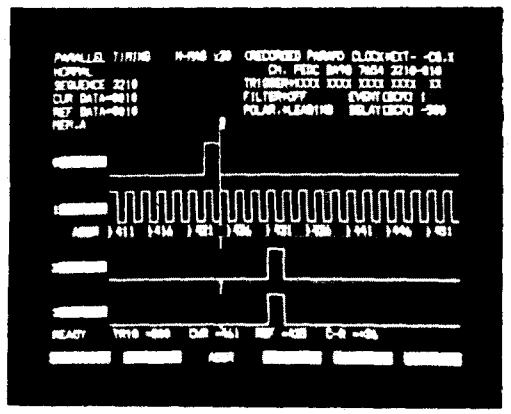
250 위치에서와 248 위치에서 보면

$\Delta = \alpha^{48} + (\alpha^{48})^2 + (\alpha^{48})^3 = \alpha^{185} = S_3$

$\Delta = \alpha^{50} + (\alpha^{50})^2 + (\alpha^{50})^3 = \alpha^{191} = S_3$

가 되어 2개의 오류가 정정된다.

이 시험동작의 결과를 Logic Analyzer를 이용하여 그림.4에 사진으로 나타내었다.



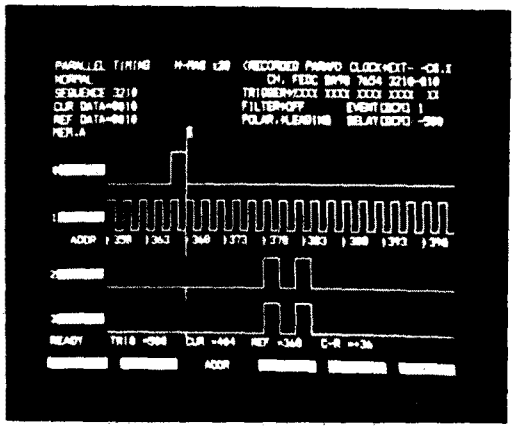


그림.4 시 험 결 과

V. 결 론

본 논문에서는 2원 BCH 부호의 복호에 있어서 오류위치다항식과 그 근을 구하지 않고 오증으로 부터 직접 오류위치를 찾아 오류를 정정할 수 있는 직접복호법을 연구 분석하고 이를 이용한 2중 오류정정 BCH 부호의 복호기를 설계하였다. 또한 (255,239)BCH 부호를 택하여 이 복호기를 TTL로 직접 장치화 함으로써 이 복호법의 효율성과 타당성을 입증하였다.

* 참 고 문 헌 *

1. 이 만 영 , 부 호 이 론 , 희 중 당 ,1984
2. R.T.Chien "Cyclic Decoding Procedures for BCH codes" IEEE Trans. on Inf. Theory, IT-10, pp357-363, 1964
3. W.W.Peterson "Encoding and error-correction procedures for the Bose-Chaudhuri codes", IRE Trans. on Inf Theory, IT-6, pp459-470, September, 1960
4. R.E.Blahut , Theory and practice of Error Control Codes , Addison-wesley , 1983
5. 이 만영, 김 창규, 임 채증, "디지탈 통신망에 BCH 부호의 응용에 관한 연구", 한양대학교, 최종연구보고서, 한국전자통신연구소, 1984