論 文

대칭행렬을 이용한 2원 BCH 부호의 복호알고리즘

正會員 廉 興 烈* 正會員 李 晩 榮**

The Decoding Algorithm of Binary BCH Codes using Symmetric Matrix

Heung Youl YOUM* Man Young RHEE** *Regular Members*

要 約 대칭행렬식계산에 의한 2원 BCH부호의 복호방법을 제안한다. 이 복호법은 오류위치번호와 미지주 X의 판별식인 대칭행렬식을 오류위치다항식으로 이용한 것으로 오류위치다항식 계수를 구하는 방법으로 기존의 어느 방법보다 간단하고 복호기 구성도 간단하다. 본 논문에서는 대칭행렬을 이용한 복호알고리즘을 설명하고 일반적인 복호기를 구성한 후 시뮬레이션을 통해 그 정당함을 입증하였으며 (63,45)BCH 부호에 적용하여 복호기를 구성하였다. 그리고 Peterson- Gorenstein-Zierler 알고리즘과 유한체 연산의 횟수와 하드웨어의 복잡도를 비교하여 본 복호방법이 효율적임을 보였다.

ABSTRACT The decoding method of Binary BCH Codes using symmetric matrix is proposed in this paper. With this method, the error-locator. polynomial is composed by symmetric matrix which consists of the powers of the unknown X plus the syndromes as its elements. The symmetric matrix can also be represented in terms of the unknown X. But the each coefficients of the error-locator polynomial represents the matrix with the syndromes as its entries. By utilizing this proposed alogrithm, the device for decoding circuit of the (63,45) BCH Code for t=3 has been implemented for demonstration.

I. 서 론

정보전달에 따르는 오류를 제어하는 방식에 FEC(Forward Error-Control)와 ARQ(Automatic Repeat Request)가 있는데 이중에서 FEC는 오류정정부호를 사용하여 통신정보의 신뢰도를 높이는 방법이다. Shannon이 1948년에 부호화이론을 제시한이후에 부호이론은 블럭부호와 길쌈부호로 나뉘어 발전하여 왔다. 블럭부호는 선형부호와 순회부호로 나눌수 있고 부호화된 블럭부

*韓國電子通信研究所
Transmission Systems Section Electronics and Telecommunications Research Institute
**漢陽大學校 電子通信工學科
Dept. of Elec. Comm. Eng., Hanyang Univ.
論文番號 : 89-36(接受 1989. 4. 11)

호의 복호과정은 일반적으로 1.오증계산 2.오증으로부터 오류형태탐지, 3.오류정정의 3단계로 이루어진다. 순회부호중에서도 가장 강력한 산발오류정정능력을 갖는 BCH 부호의 복호방법으로는 table look-up 방식과 오류위치다항식을 구하여 정정하는 방식이있는데 table look-up 방식은 ROM의 용량문제로 실현성이 없지만 오류위치다항식법은 오증비트수 증가에 따른 실현성의 급속한 저하는 없으나 복잡한 구성을 갖는다. 오류위치다항식법으로는 오증과 오류위치다항식계수와의 관계를 Newton의 항등식으로 푸는 Peterson 방법, 이를 하드웨어적으로 나타낸 Berlekamp-Massey 방법이 대표적이다.

본 논문에서 제안한 복호법은 오류위치다항식법에 속하는 것으로 오증에 미지수를 더한 성분을 대칭행렬식으로 표현하여 이것을 오류위치다항식으로 이용한다. 이것은 계산값을 대칭행렬요소로 나타내 이용하고 또한 X의 다항식으로 나타내어 오증을 계수로 갖는 행렬식으로 표현된다. 오류위치다항식계수를 구하는 방법으로는 이 방법이 어느 방법보다도 간단하고 복호기구성도 단순하다. 본 논문에서는 대칭행렬을 이용한 복호알고리즘을 상세히 설명하고 일반적인 복호기를 구성한 후 시뮬레이션을 통하여 본 알고리즘이 정당함을 입증하였으며 (63,45) BCH부호에 적용하여 복호기를 구성하였다. 그리고 Peterson- Gorenstein-Zierler 알고리즘과 유한체 연산의 횟수와 하드웨어의 복잡도를 비교하여 본 복호기법이 효율적임을 보였다.

Ⅱ. 대칭행렬을 이용한 BCH 부호의 복호방법

1. 오증의 계산

부호장이 $n=2^m-1$인 t중 오류정정 BCH 부호의 생성다항식 $g(x)$는 다음과 같다.

$$g(x)=\mathrm{LCM}\{m_1(x), m_3(x), \cdots, m_{2t-1}(x)\} \quad (1)$$

여기서 $m_i(x)$, $i=1,3,\cdots,2t-1$는 최소다항식이다. $c(x)$가 t중 오류정정 BCH부호의 부호어면

$$\bar{c}\cdot\bar{H}^T=0 \quad (2)$$

이 성립하고, $GF(2^m)$ 상의 원소 α^i를 근으로 갖는 BCH부호의 부호어와 검사행렬$\bar{H}$의 행은 영공간관계에 있음을 의미한다. 그리고, 검사행렬$\bar{H}$는

$$H=[(\alpha^i)^0\ (\alpha^i)1\ (\alpha^i)^2\cdots(\alpha^i)^{n-1}] \quad (3)$$

이다. 또한, 부호어$c(x)$를 전송했을 때 수신계열$r(x)$는 다음과 같다.

$$r(x)=c(x)+e(x) \quad (4)$$

수신어$\bar{r}$에 대하여 오증$\bar{S}(s_1, s_2, \cdots, s_{2t})$은

$$\bar{S}=\bar{r}\cdot\bar{H}^T=(\bar{c}+\bar{e})\bar{H}^T=\bar{e}\bar{H}^T \quad (5)$$

로 된다. ν개의 오류가 발생했을 때 수신어를 최소다항식$m_i(x)$, $i=1,3,\cdots 2t-1$로 나누어 얻는 오증성분은

$$s_k=e(\alpha^k)=\sum_{\lambda=1}^{\nu}(\alpha^{j\lambda})^k=\sum_{\lambda=1}^{\nu}X_\lambda^k\ (1\leq k\leq 2t) \quad (6)$$

이다. 여기서, $X_\lambda\ (=\alpha^{j\lambda})$는 오류위치번호를 나타내고, $s_{2k}=s_k^2$인 관계를 갖는다.[1]

2. 오류위치다항식

ν개의 오류가 발생한 경우, 모든 오류위치부호를 근으로 갖는 오류위치다항식을 유도한다. 오증성분에 미지수 X를 더한 것으로 M_k를 다음과 같이 정의한다.

$$M_k=S_k+X^k \quad (k=1,2,\cdots,2t) \quad (7)$$

$$M_0=\sum_{i=1}^{\nu}X_i^0+X^0$$

이 M_k를 대칭으로 배열한 행렬D_v는 다음과 같이 쓸 수 있으며 Vandermonde[10] 행렬V와 V^T의 적으로 나타낼 수 있다. $X=X_{v+1}$로 놓으면

$$D_v=\begin{bmatrix} M_0 & M_1 & \cdots M_{v-1} & M_v \\ M_1 & M_2 & \cdots M_v & M_{v+1} \\ \bullet & \bullet & \cdots \bullet & \bullet \\ M_{v-1} & M_v & \cdots M_{2v-2} & M_{2v-1} \\ M_v & M_{v+1} & \cdots M_{2v-1} & M_{2v} \end{bmatrix}$$

$$=\begin{bmatrix} 1 & 1 & \cdots 1 & 1 \\ X_1 & X_2 & \cdots X_v & X_{v+1} \\ X_1^2 & X_2^2 & \cdots X_v^2 & X_{v+1}^2 \\ \bullet & \bullet & \cdots \bullet & \bullet \\ X_1^v & X_2^v & \cdots X_v^v & X_{v+1}^v \end{bmatrix} \begin{bmatrix} 1 & X_1 & X_1^2 & \cdots X_1^v \\ 1 & X_2 & X_2^2 & \cdots X_2^v \\ \bullet & \bullet & \bullet & \cdots \bullet \\ \bullet & \bullet & \bullet & \cdots \bullet \\ 1 & X_{v+1} & X_{v+1}^2 & \cdots X_{v+1}^v \end{bmatrix} \quad (8)$$

이다. 행렬 A, B, C는 A=BC가 성립한다면, $\det(A)=\det(B)\det(C)$가 성립한다.
그런데, Vandermonde 행렬식V는 다음과 같다[10].

$$V=\det(V)=\det(V^T)=\begin{vmatrix} 1 & 1 & \cdots 1 & 1 \\ X_1 & X_2 & \cdots X_v & X_{v+1} \\ X_1^2 & \cdots & \cdots\cdots & X_v^2 \\ \bullet & \bullet & \cdots & \bullet \\ X_1^v & X_2^v & \cdots X_v^v & X_{v+1}^v \end{vmatrix}$$

$$=\begin{vmatrix} 1 & X_1 & X_1^2 & \cdots X_1^v \\ 1 & X_2 & X_2^2 & \cdots X_2^v \\ \bullet & \bullet & \cdots & \bullet \\ 1 & X_v & X_v^2 & \cdots X_v^v \\ X_1^v X_{v+1} & X_{v+1}^2 & \cdots X_{v+1}^v \end{vmatrix} = \prod_{i>j}^{v+1}(X_i+X_j) \quad (9)$$

식 (8)과 (9)로부터

$$D_v=\det(D_v)=\det(V)\det(V^T)=\prod_{i>j}^{v+1}(X_i+X_j)^2 \quad (10)$$

이고, 식(10)에서 X_{v+1}항에 대하여 다시 쓰면

$$D_v=\prod_{i>j}^{v+1}(X_i+X_j)^2=\prod_{i>j}^{v}(X_i+X_j)^2\prod_{k=1}^{v}(X_{v+1}+X_k)^2$$
$$=\prod_{i>j}^{v}(X_i+X_j)^2\prod_{k=1}^{v}(X+X_k)^2 \quad (11)$$

이다. 그런데 윗식에서 $X=X_{v+1}$ 이므로 미지수 X가 오류위치번호를 나타내는 $X_1, X_2, \cdots, X_v$ 중의 하나와 일치하면 D_v가 영이 된다. 따라서, 식 (11)을 오류 위치다항식으로 이용할 수 있다[9].

한편, 식(9)에서 Vandermonde 행렬식은 다음과 같이 쓸수 있다.

$$\begin{vmatrix} 1 & 1 & \cdots 1 \\ X_1 & X_2 & \cdots X_{v+1} \\ \bullet & \bullet & \cdots \bullet \\ X_1^v & X_2^v & \cdots X_{v+1}^v \end{vmatrix} = \begin{vmatrix} X_2-X_1 & X_3-X_1 & \cdots X_{v+1}-X_1 \\ X_2^2-X_1^2 & X_3^2-X_1^2 & \cdots X_{v+1}^2-X_1^2 \\ \bullet & \bullet & \cdots \bullet \\ X_2^v-X_1^v & X_3^v-X_1^v & \cdots X_{v+1}^v-X_1^v \end{vmatrix} \quad (12)$$

식(12)에서 행렬식의 차수가 하나 줄어든 우변은

$$V=\begin{vmatrix} X_2 & X_3 & \cdots X_{v+1} \\ X_2^2 & X_3^2 & \cdots X_{v+1}^2 \\ \bullet & \bullet & \cdots \bullet \\ X_2^v & X_3^v & \cdots X_{v+1}^v \end{vmatrix} + \begin{vmatrix} X_1 & X_3 & \cdots X_{v+1} \\ X_1^2 & X_3^2 & \cdots X_{v+1}^2 \\ \bullet & \bullet & \cdots \bullet \\ X_1^v & X_3^v & \cdots X_{v+1}^v \end{vmatrix}$$

$$+ \begin{vmatrix} X_1 & X_2 & X_4 & \cdots X_{v+1} \\ X_1^2 & X_2^2 & X_4^2 & \cdots X_{v+1}^2 \\ \bullet & \bullet & \bullet & \cdots \bullet \\ X_1^v & X_2^v & X_4^v & \cdots X_{v+1}^v \end{vmatrix} + \cdots +$$

$$\begin{vmatrix} X_1 \cdots X_{v-1} & X_{v+1} \\ X_1^2 \cdots X_{v-1}^2 & X_{v+1}^2 \\ \bullet \quad \bullet \quad \cdots & \bullet \\ X_1^v \cdots X_{v-1}^v & X_{v+1}^v \end{vmatrix} + \begin{vmatrix} X_1 \cdots X_{v-1} & X_v \\ X_1^2 \cdots X_{v-1}^2 & X_v^2 \\ \bullet \quad \bullet \quad \cdots & \bullet \\ X_1^v \cdots X_{v-1}^v & X_v^v \end{vmatrix} + R$$

$$=\sum_{n=1}^{v+1}\left[\prod_{\substack{m=1 \\ m\neq n}}^{v+1} X_m \prod_{\substack{i>j \\ i,j\neq n}}^{v+1}(X_i+X_j)\right] \quad (13)$$

이다. 식(13)에서 R은 두열이상이 같은 열인 경우로 행렬식은 영이 된다. 그러므로, 식(10)은 2

원 BCH부호의 경우에

$$D_v=\begin{vmatrix}1 & 1 & \cdots 1\\ X_1 & X_2 & \cdots X_{v+1}\\ \cdot & \cdot & \cdots \cdot\\ X_1^v & X_2^v & \cdots X_{v+1}^v\end{vmatrix}\begin{vmatrix}1 & X_1 & \cdots X_1^v\\ 1 & X_2 & \cdots X_2^v\\ \cdot & \cdot & \cdots \cdot\\ 1 & X_{v+1} & \cdots X_{v+1}^v\end{vmatrix}$$

$$=\begin{vmatrix}X_2-X_1 & X_3-X_1 & \cdots & X_{v+1}-X_1\\ X_2^2-X_1^2 & X_3^2-X_1^2 & \cdots & X_{v+1}^2-X_1^2\\ \cdot & \cdot & \cdots & \cdot\\ X_2^v-X_1^v & X_3^v-X_1^v & \cdots & X_{v+1}^v-X_1^v\end{vmatrix}$$

$$\begin{vmatrix}X_2 & -X_1 & X_2^2 & -X_1^2 & \cdots & X_2^v & -X_1^v\\ X_3 & -X_1 & X_3^2 & -X_1^2 & \cdots & X_3^v & -X_1^v\\ \cdot & & \cdot & & \cdots & \cdot & \\ X_{v+1} & -X_1 & X_{v+1}^2 & -X_1^2 & \cdots & X_{v+1}^v & -X_1^v\end{vmatrix}$$

$$=\sum_{n=1}^{v+1}\left[\prod_{\substack{m\neq n\\ m=1}}^{v+1}X_m\prod_{\substack{i>j\\ i\neq n\\ j\neq n}}^{v+1}(X_i+X_j)^2\right] \tag{14}$$

이 된다. 그런데 행렬 E_v를 아래와 같이 나타낸 경우를 생각해 보자.

$$E_v=\begin{bmatrix}X_1 & X_2 & \cdots & X_{v+1}\\ X_1^2 & X_2^2 & \cdots & X_{v+1}^2\\ \cdot & \cdot & \cdots & \cdot\\ X_1^v & X_2^v & \cdots & X_{v+1}^v\end{bmatrix}\cdot\begin{bmatrix}X_1 & X_1^2 & \cdots X_1^v\\ X_2 & X_2^2 & \cdots X_2^v\\ \cdot & \cdot & \cdots \cdot\\ X_{v+1} & X_{v+1}^2 & \cdots X_{v+1}^v\end{bmatrix}$$

$$=\begin{bmatrix}\sum_{i=1}^{v+1}X_i^2 & \sum_{i=1}^{v+1}X_i^3 & \cdots & \sum_{i=1}^{v+1}X_i^{v+1}\\ \sum_{i=1}^{v+1}X_i^3 & \sum_{i=1}^{v+1}X_i^4 & \cdots & \sum_{i=1}^{v+1}X_i^{v+2}\\ \cdot & \cdot & \cdots & \cdot\\ \sum_{i=1}^{v+1}X_i^{v+1} & \sum_{i=1}^{v+1}X_i^{v+2} & \cdots & \sum_{i=1}^{v+1}X_i^{2v}\end{bmatrix} \tag{15}$$

식(15)의 행렬식은 아래와 같다.

$$E_v=\det(E_v)=\begin{vmatrix}X_2 & X_3 & \cdots X_{v+1}\\ X_2^2 & X_3^2 & \cdots X_{v+1}^2\\ \cdot & \cdot & \cdots \cdot\\ X_2^v & X_3^v & \cdots X_{v+1}^v\end{vmatrix}\begin{vmatrix}X_2 & X_2^2 & \cdots X_2^v\\ X_3 & X_3^2 & \cdots X_3^v\\ \cdot & \cdot & \cdots \cdot\\ X_{v+1} & X_{v+1}^2 & \cdots X_{v+1}^v\end{vmatrix}+$$

$$\begin{vmatrix}X_1 & X_3 & \cdots X_{v+1}\\ X_1^2 & X_3^2 & \cdots X_{v+1}^2\\ \cdot & \cdots\cdots & \cdot\\ X_1^v & X_3^v & \cdots X_{v+1}^v\end{vmatrix}\begin{vmatrix}X_1 & X_1^2 & \cdots X_1^v\\ X_2 & X_2^2 & \cdots X_2^v\\ \cdot & \cdots\cdots & \cdot\\ X_{v+1} & X_{v+1}^2 & \cdots X_{v+1}^v\end{vmatrix}+\cdots+$$

$$\begin{vmatrix}X_1 & \cdots & X_{v-1} & X_{v+1}\\ X_1^2 & \cdots & X_{v-1}^2 & X_{v+1}^2\\ \cdot & \cdots & \cdot & \cdot\\ X_1^v & \cdots & X_{v-1}^v & X_{v+1}^v\end{vmatrix}\begin{vmatrix}X_1 & X_1^2 & \cdots X_1^v\\ \cdot & \cdot & \cdots \cdot\\ X_{v-1} & X_{v-1}^2 & \cdots X_{v-1}^v\\ X_{v+1} & X_{v+1}^2 & \cdots X_{v+1}^v\end{vmatrix}$$

$$+\begin{vmatrix}X_1 & X_2 & \cdots X_v\\ X_1^2 & X_2^2 & \cdots X_v^2\\ \cdot & \cdot & \cdots \cdot\\ X_1^v & X_2^v & \cdots X_v^v\end{vmatrix}\begin{vmatrix}X_1 & X_1^2 & \cdots X_1^v\\ X_2 & X_2^2 & \cdots X_2^v\\ \cdot & \cdot & \cdots \cdot\\ X_v & X_v^2 & \cdots X_v^v\end{vmatrix}$$

$$=\sum_{n=1}^{v+1}\left[\prod_{\substack{m\neq n\\ m=1}}^{v+1}X_m\prod_{\substack{i>j\\ i\neq n\\ j\neq n}}^{v+1}(X_i+X_j)^2\right] \tag{16}$$

그리고, E_v를 M_k 형태로 표현하면 다음과 같다. 또한 이것은 짝수차의 항을 갖고 최고차수는 2v인 X의 다항식으로 쓸 수 있다.

$$E_v=\begin{vmatrix}M_2 & M_3 & \cdots & M_{v+1}\\ M_3 & M_4 & \cdots & M_{v+2}\\ \cdot & \cdot & \cdots & \cdot\\ M_v & M_{v+1} & \cdots & M_{2v-1}\\ M_{v+1} & M_{v+2} & \cdots & M_{2v}\end{vmatrix} \tag{17}$$

$$= \sum_{i=0}^{v} B_i X^{2i} \tag{18}$$

따라서, 식(14)와 (16)으로 부터 $D_v = E_v$이므로 D_v보다 차수가 하나 작은 행렬식 E_v를 오류위치 다항식으로 이용할 수 있다. 식(11)과 (14)에서

$$E_v = D_v = \prod_{i>j}^{v+1} (X_i + X_j)^2 = \prod_{i>j}^{v} (X_i + X_j)^2$$

$$\cdot \prod_{k=1}^{v} (X_{v+1} + X_k)^2$$

$$= \sum_{n=1}^{v+1} \left[\prod_{\substack{m=1 \\ m \neq n}}^{v+1} X_m \prod_{\substack{i>j \\ i \neq n \\ j \neq n}}^{v+1} (X_i + X_j)^2 \right]$$

이다. D_v와 E_v는 오류위치번호 X_i, i = 1, 2, …, v중에서 임의의 한 X_μ 가 영일 경우에

$$E_v = D_v = \prod_{\substack{i>j \\ i \neq \mu \\ j \neq \mu}}^{v} (X_i + X_j)^2 \prod_{\substack{k \neq \mu \\ k=1}}^{v} (X + X_k)^2$$

$$= X^2 \prod_{\substack{m \neq \mu \\ m=1}}^{v} X_m^2 \prod_{\substack{i>j \\ i \neq \mu \\ j \neq \mu}}^{v} (X_i + X_j)^2 \prod_{k \neq \mu}^{v} (X + X_k)^2 \tag{19}$$

이다. 식(19)의 우변을 살펴보면 미지수 X가 $X_k (k \neq \mu)$중의 하나와 일치할 경우에만 E_v가 영이 됨을 알수 있다. 식(11)과 (19)로부터 오류위치다항식 E_v는 오류가 v개, v-1개 발생한 경우의 근을 모두 구할 수 있다.

식(17)에서 미지수X를 영으로 놓고, 오증만으로 표현되는 이식을 Z_v로 정의하자. 이것은 식(18)의 B_0값과 같다.

$$Z_v = B_0 = \begin{vmatrix} S_2 & S_3 & \cdots & S_{v+1} \\ S_3 & S_4 & \cdots & S_{v+2} \\ \bullet & \bullet & \cdots & \bullet \\ S_{v+1} & S_{v+2} & \cdots & S_{2v} \end{vmatrix} \tag{20}$$

$v > \nu$ (실제로 발생한 오류비트수)에 대하여 $X_v = 0$으로 놓고 A를 Vandermonde행렬로 놓자.

$$A = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ X_1 & X_2 & \cdots & X_v \\ \bullet & \bullet & \cdots & \bullet \\ X_1^{v-1} & X_2^{v-1} & \cdots & X_v^{v-1} \end{bmatrix}, \text{ 여기서 원소 } A_{ij} = X_j^{i-1}$$

그리고, B를 대각행렬로 놓으면

$$B = \begin{bmatrix} Y_1 X_1^2 & 0 & \cdots & 0 \\ 0 & Y_2 X_2^2 & \cdots & 0 \\ \bullet & \bullet & \cdots & \bullet \\ 0 & 0 & \cdots & Y_v X_v^2 \end{bmatrix}, \text{ 여기서 } B_{ij} = Y_i X_i \delta_{ij}, \ \delta_{ij} = 1, (i=j) \ \delta_{ij} = 0, (i \neq j)$$

그러면 행렬의적 ABA^T는

$$ABA^T = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ X_1 & X_2 & \cdots & X_v \\ \bullet & \bullet & \cdots & \bullet \\ X_1^{v-1} & X_2^{v-1} & \cdots & X_v^{v-1} \end{bmatrix} \begin{bmatrix} Y_1 X_1^2 & 0 & \cdots & 0 \\ 0 & Y_2 X_2^2 & \cdots & 0 \\ \bullet & \bullet & \cdots & \bullet \\ 0 & 0 & \cdots & Y_v X_v^2 \end{bmatrix}$$

$$\begin{bmatrix} 1 & X_1 & \cdots & X_1^{v-1} \\ 1 & X_2 & \cdots & X_2^{v-1} \\ \bullet & \bullet & \cdots & \bullet \\ 1 & X_v & \cdots & X_v^{v-1} \end{bmatrix} = Z_v \tag{21}$$

여기서 $Y_1 = Y_2 = \cdots = Y_v = 1$ 이다. 식(21)에서

$$(ABA^T)_{ij} = \sum_{n=1}^{v} X_n^{i-1} \sum_{k=1}^{v} Y_n X_n^2 \delta_{nk} X_k^{j-1}$$

$$= \sum_{n=1}^{v} X_n^{i-1} Y_n X_n^2 X_n^{j-1} = \sum_{n=1}^{v} Y_n X_n^{i+j-1}$$

로 쓸수 있으며, 행렬 Z_v의 ij 원소이다. 그러므로, $Z_v = ABA^T$이다. 따라서, $Z_v(\det(Z_v))$는 $\det(Z_v) = \det(A)\det(B)\det(A^T)$를 만족하므로 만약 v가 실제 발생한 오류 ν보다 큰 경우, 즉 $v > \nu$일 때 $\det(B) = 0$ 이 되어 $\det(Z_v) = 0$이다. 또한

v가 ν와 같은 경우(v=ν), det(B)≠0 이고, det(A)≠0 이므로 det(Z_v)≠0 이다.

위에 서술한 바에 의하여 Z_v를 가지고 v를 줄여 가면서 det(Z_v)≠0 으로 오류가 몇개 발생했는지를 알 수 있다.[7]

식(17)은 X에 대하여 나타내면 오류위치다항식 E_v는 $M_k=S_k+X^k$ 이므로

$$E_v=\begin{vmatrix} S_2+X^2 & S_3+X^3 & \cdots & S_{v+1}+X^{v+1} \\ S_3+X^3 & S_4+X^4 & \cdots & S_{v+2}+X^{v+2} \\ \bullet & \bullet & \cdots & \bullet \\ S_{v+1}+X^{v+1} & S_{v+2}+X^{v+2} & \cdots & S_{2v}+X^{2v} \end{vmatrix}$$

$$=\sum_{i=0}^{v} B_i X^{2i}$$

$$=\begin{vmatrix} S_2 & S_3 & \cdots & S_{v+1} \\ S_3 & S_4 & \cdots & S_{v+2} \\ \bullet & \bullet & \cdots & \bullet \\ S_{v+1} & S_{v+2} & \cdots & S_{2v} \end{vmatrix}+\begin{vmatrix} X^2 & S_3 & \cdots & S_{v+1} \\ X^3 & S_4 & \cdots & S_{v+2} \\ \bullet & \bullet & \cdots & \bullet \\ X^{v+1} & S_{v+2} & \cdots & S_{2v} \end{vmatrix}$$

$$+\cdots+$$

$$\begin{vmatrix} S_2 & \cdots & X^v & S_{v+1} \\ S_3 & \cdots & X^{v+1} & S_{v+2} \\ \bullet & \cdots & \bullet & \bullet \\ S_{v+1} & \cdots & X^{2v-1} & S_{2v} \end{vmatrix}+\begin{vmatrix} S_2 & \cdots & S_v & X^{v+1} \\ S_3 & \cdots & S_{v+1} & X^{v+2} \\ \bullet & \cdots & \bullet & \bullet \\ S_{v+1} & \cdots & S_{2v-1} & X^{2v} \end{vmatrix}$$

$$+R \qquad (22)$$

여기서, R 은두열이상이 X의 멱 형태로 나타나는 행렬식으로

$$\begin{vmatrix} S_2 & \cdots X^i & \cdots X^j & \cdots S_{v+1} \\ S_3 & \cdots X^{i+1} & \cdots X^{j+1} & \cdots S_{v+1} \\ \bullet & \cdots \bullet & \cdots \bullet & \cdots \bullet \\ S_{v+1} & \cdots X^{i+v} & \cdots X^{j+v} & \cdots S_{2v} \end{vmatrix}=X^iX^j$$

$$\begin{vmatrix} S_2 & \cdots 1 & \cdots 1 & \cdots S_{v+1} \\ S_3 & \cdots X & \cdots X & \cdots S_{v+2} \\ \bullet & \cdots \bullet & \cdots \bullet & \cdots \bullet \\ S_{v+1} & \cdots X^v & \cdots X^v & \cdots S_{2v} \end{vmatrix}=0 \qquad (23)$$

이므로, 두열이상이 X의 멱승으로 나타난 행렬식 R은 영이다. 식(22)는 Z_v의 한 열에 대하여 S_k를 X_k로 놓은 각각의 모든 행렬식을 그대로 놓고 X를 포함하고 있는 한열에 대하여 전개하고난 다음에 X의 다항식으로 나타내면 X의 계수 B_i는 Z_v의 대각요소중 S_{2i}를 포함한 행과 열을 소거한 행렬식과 같다. 따라서, 오류위치다항식 $E_v(X)$를 X에 대한 식으로 계수를 구할 경우에 B_0는 v차의 행렬식계산을 하지만 B_i, i=1,2,⋯,v는 v-1차 행렬식 계산을 해서 계수를 구할수 있다. 이것은 오류위치다항식법에 속하는 Peterson 방법이 각계수를 구하는데 있어서 모두 v가 행렬식을 계산함에 비하여 계산수가 훨씬 줄어듬을 의미한다. 위에서 서술한 알고리즘은 오류위치다항식의 제곱형태로 주어지고 근으로 얻어진 값은 바로 오류위치를 나타낸다. 따라서, 오류위치다항식으로는 M_k로 나타나는 식(17)과 X의 다항식으로 표현되는 식(22)을 이용한다. v=3일 경우 식(17)을 이용해서 M_k로 나타낸 오류위치다항식을 구하면

$$E_3=\begin{vmatrix} M_2 & M_3 & M_4 \\ M_3 & M_4 & M_5 \\ M_4 & M_5 & M_6 \end{vmatrix}=M_2M_4M_6+M_4^3+M_3^2M_6+$$

$$M_2M_5^2=M_1^{12}+M_3^4+M_1^6M_3^2+M_1^2M_5^2 \qquad (24)$$

식(22)를 이용하여 X의 다항식으로 나타낸 오류위치다항식은

$$B_0=Z_3=\begin{vmatrix} S_2 & S_3 & S_4 \\ S_3 & S_4 & S_5 \\ S_4 & S_5 & S_6 \end{vmatrix}=S_1^{12}+S_3^4+S_1^6S_3^2+S_1^2S_5^2$$

$$B_1 = \begin{vmatrix} s_4 & s_5 \\ s_5 & s_6 \end{vmatrix} = s_1^4 s_3^2 + s_5^2$$

$$B_2 = \begin{vmatrix} s_2 & s_4 \\ s_4 & s_6 \end{vmatrix} = s_1^2 s_3^2 + s_1^6$$

$$B_3 = \begin{vmatrix} s_2 & s_3 \\ s_3 & s_4 \end{vmatrix} = s_1^6 + s_3^2$$

이고,

$$E_3 = [s_1^6 + s_3^2 + s_1^3 s_3 + s_1 s_5 + (s_1^2 s_3 + s_6) X + (s_1 s_3 + s_1^4) X^2 + (s_1^3 + s_3) X^3]^2 \quad (25)$$

3. 오류위치계산 및 오류정정

M_k로 나타낸 오류위치다항식은 X로 α^{n-i}, (i=1,2 … ,n)를 발생하고, 이때 대응하는 각 M_k값을 결정하고, $X=\alpha^{n-i}$에 대하여 계산한 행렬식값이 영이면, 수신어의 제i번째 비트를 오류로 판정하여 정정한다. t가 작은 경우는 고속복호가 가능하고 구성도 간단하여 실용적이다. 한편, X의 다항식으로 오류위치다항식을 이용하면, 먼저 계수를 계산한 다음 Chien의 오류위치탐지회로를 이용하여 오류위치다항식의 근을 구하여 오류를 정정한다. 그림1과 그림2는 복호기를 나타낸다.

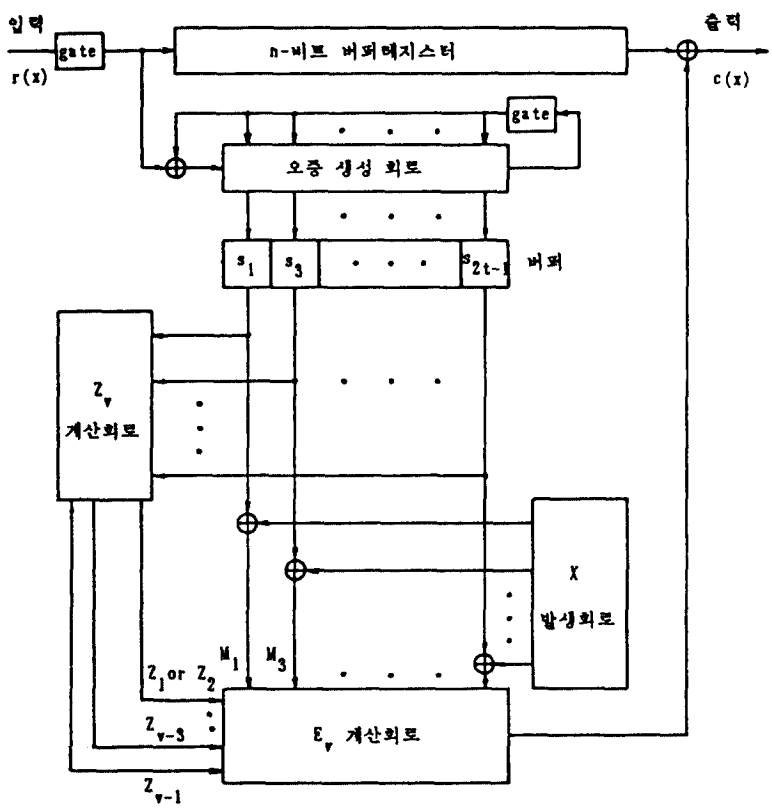


그림1. M_k로 나타낸 오류위치다항식을 이용한 복호기

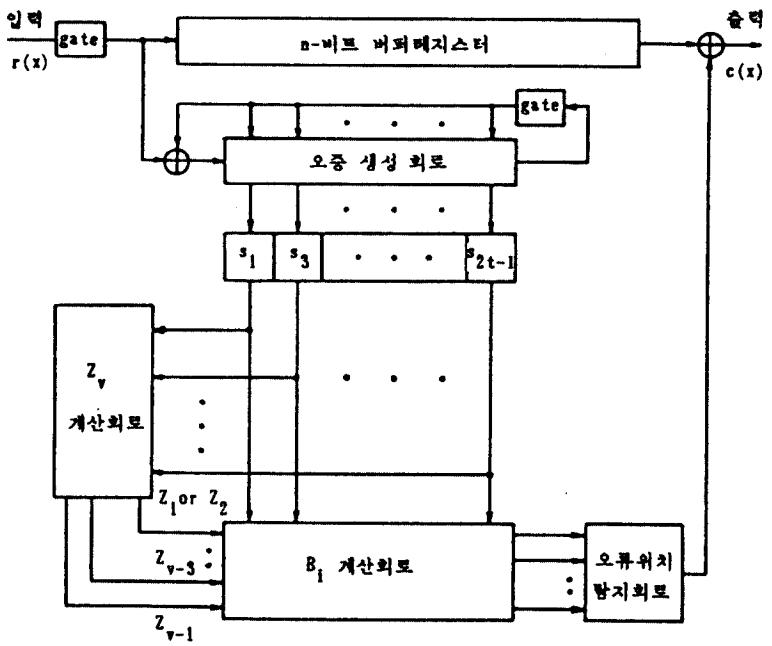


그림2. X의 다항식으로 나타낸 오류위치다항식을 이용한 복호기

Ⅲ. (63,45) BCH 부호의 복호

(63,45) BCH 부호는 3중 오류정정 BCH 부호이고, 부호장 $n=2^m-1=63$ 이므로 $m=6$이다. 이 부호의 생성다항식 $g(x)$는

$$g(x)=1+x+x^2+x^3+x^6+x^7+x^9+x^{15}+x^{16}+x^{17}+x^{18}$$

이 부호의 복호기는 수신어로부터 오증을 구하는 오증회로와 구해진 오증에서 오류비트수를 판정하는 Z_v 계산회로 및 비교기, 오류비트수에 따라 오류위치다항식을 계산하는 E_v계산회로, 그리고 행렬식의 요소 M_k를 만들어 주기위한 미지수발생회로와 오류위치를 찾아내는 오류위치탐지회로로 구성할 수 있다.

행렬식 Z_2값이 영이 아니면 (63,45) BCH부호의 경우, 오류가 2개 또는 3개 발생했음을 의미하고 복호에 이용할 오류위치다항식으로 E_3를 선택한다. 그렇지만 Z_2값이 영일 경우에는 오류가 한개이하로 발생한 경우로 오류위치다항식은 E_1을 선택한다. 그러므로 행렬식계산에 따른 오류위치다항식은

$E_1=M_1^2=s_1^2+X^2$ 이고, E_3는 식(25)와 같다.

M_k로 나타낸 오류위치다항식을 이용한 (63,45) BCH부호의 복호기는 그림3과 같다. 복호기의 동작은 데이타버퍼에 수신어 63비트가 모두 입력되면 오증요소 s_1,s_3,s_5가 동시에 생성되고, Z_2값이 계산되고, 이값이 영이 아닐경우에 E_3를 택하여 영인지 아닌지를 검사한다. Z_2값이 영일 경우에는 E_1을 계산한 값이 영인지를 검사하게 된다. 따라서 데이타버퍼의 수신어가 한 비트씩 출력되고 미지수 X값이 α^{n-i} 일때 계산된 E_v값이 영이면 이때의 출력비트를 오류로 판정하여 정정하게 된다. X발생회로는 그림4와 같다.

한편, X의 다항식 형태의 오류위치다항식을 이용한 복호기는 M_k로 나타낸 오류위치다항식의 경우와 같이 오증과 오류비트수 판정회로를 구성하고, 이때의 복호기 구성은 그림5와 같고 동작은 오류위치를 구할때에 Chien의 오류탐지회로를 사용한 점이 다르다. Chien의 오류탐지회로를 설계하면 그림6과 같다.

이 방법은 행렬식의 특성으로 인하여 계산수의 감소가 발생하므로 t가 큰 경우의 고속복호에 본 복호방법이 유용하게 이용될 수 있다.

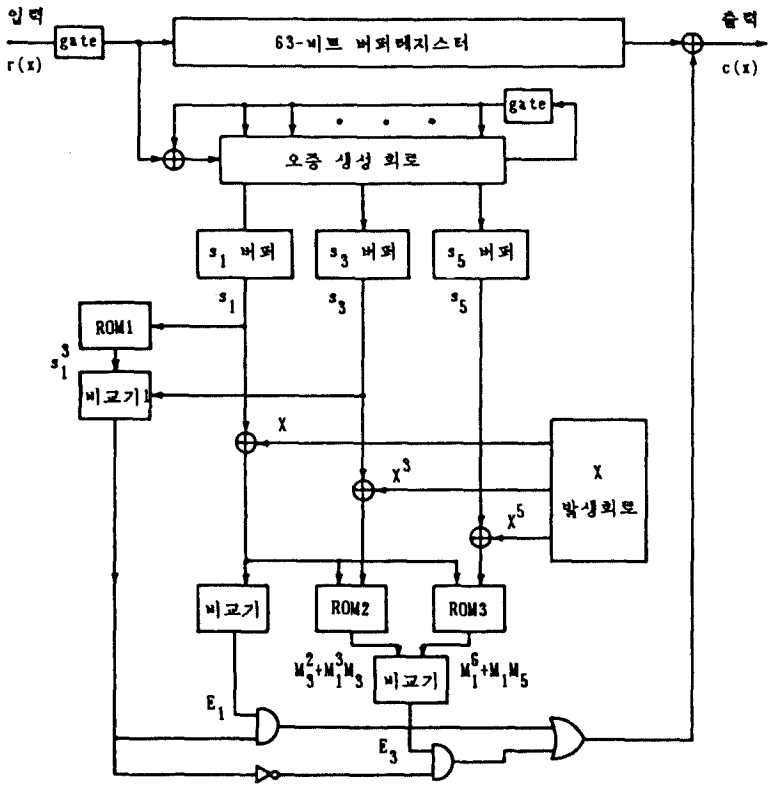


그림3. M_k로 나타낸 오류위치다항식을 이용한 (63,45) BCH 복호기

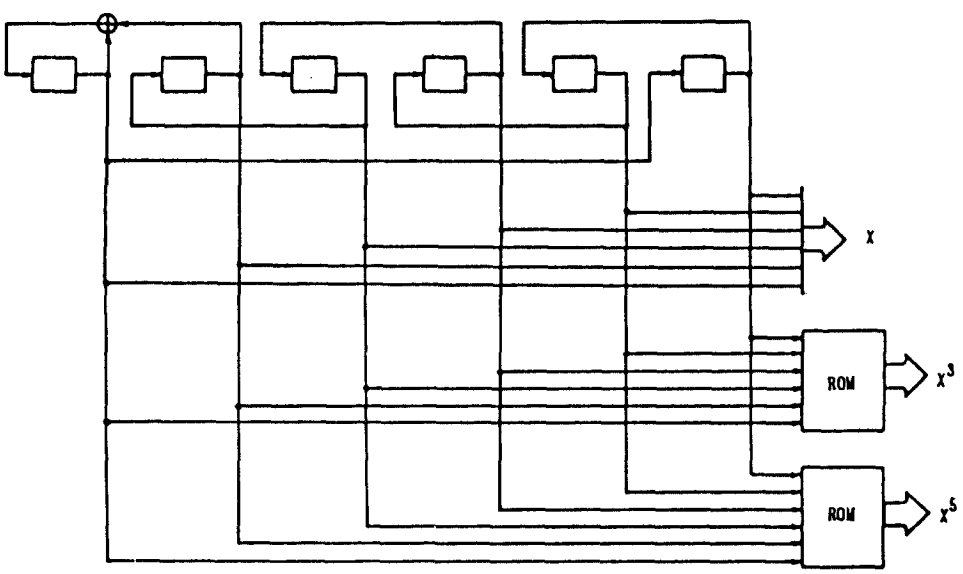


그림4. 미지수 X 발생회로

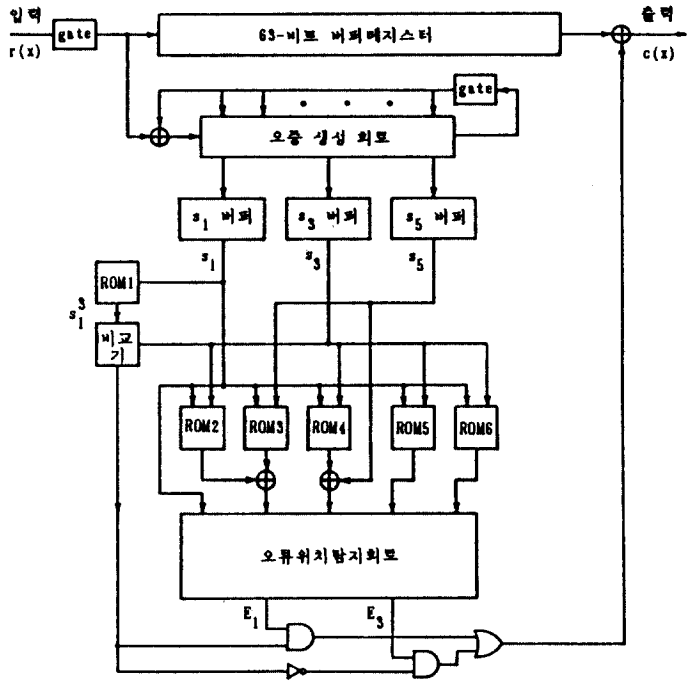


그림5. X의 다항식으로 나타낸 오류위치다항식을 이용한 (63,45) BCH 복호기

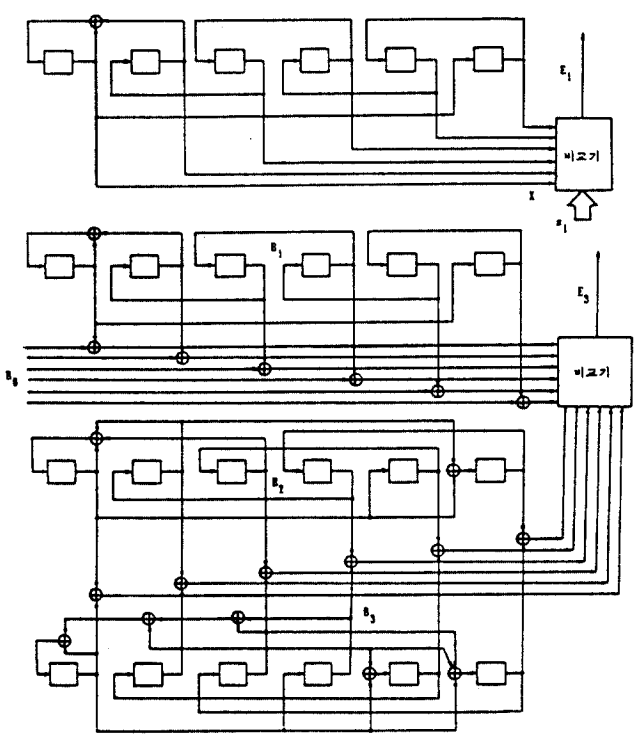


그림6. Chien의 오류탐지회로

Ⅳ. 시뮬레이션 및 결과고찰

(63,45) BCH 부호의 대칭행렬을 이용한 복호기 구성을 위해, 복호알고리즘을 컴퓨터 시뮬레이션 하여 본 복호기법이 정당함을 입증하였다. 시뮬레이션을 위한 프로그램의 흐름도는 그림7과 같다.

(63,45) BCH 부호의 부호어는 난수발생기를 이용하여 45 비트의 정보어를 구한후 이를 부호화하여 얻었으며, 통신로 상에서 발생하는 3개 이하의 오류도 역시 난수를 발생시켜 이를 부호어에 더하여 수신어를 생성하였다.

한편 실제로 하드웨어로 구현할때 수신어 저장 버퍼에서 제일 먼저 출력되는 비트가 α^{62}의 위치에 해당하는 비트이므로 X를 α^{62}로 초기화한다. 그리고 X^3,X^5을 구한 후 (63,45) BCH 부호의 수신어로부터 구한 오증을 이용하여 실제로 발생한 오류의 갯수를 $S_1^3+S_3\neq 0$ 식을 이용하여 판단한다.

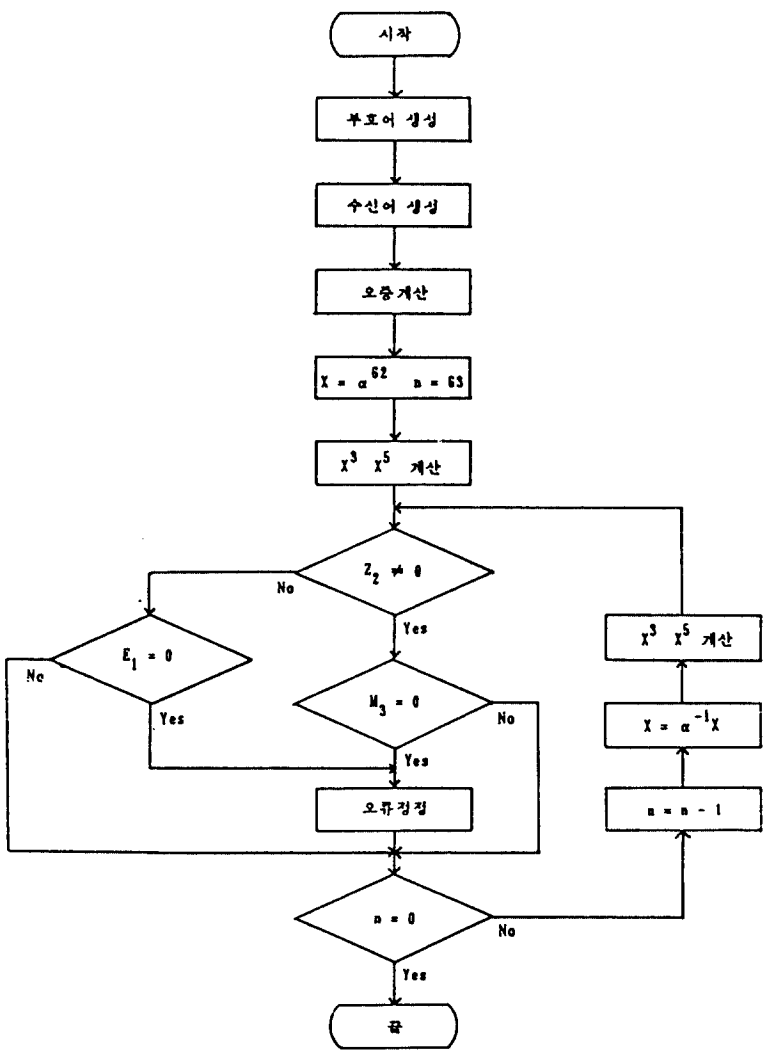


그림7. (63,45) BCH 부호의 M_k를 이용한 복호과정 흐름도

오류가 2개 또는 3개 발생한 경우 $M_3^2+M_1^3$ $M_3=M_1^6+M_1M_5$의 성립여부를 판단하여 성립하면 해당위치 X에서 오류가 발생한 것으로 간주하여 오류를 정정하고 만일 성립하지 않으면 오류의 위치가 아닌 것으로 간주한다.

만일 오류가 1개 발생하였다면 $S_1+X=0$ 여부를 판단하여 만족하면 해당위치 X가 오류위치로 간주하고 오류를 정정하며, 성립하지 않으면 오류위치가 아닌 것으로 판단한다.

수신어 α^{62}의 위치가 대한 오류에 정정된 후 $X\rightarrow X\alpha^{-1}$로 변환시켜 다음 비트위치에 대해서도 위에서와 같은 단계를 거쳐 오류를 정정한다. 이런식으로 수신어의 모든 위치에 대하여 순차적으로 오류위치 여부를 판단하여 오류를 정정한다. 위와 같은 알고리즘으로 (63,45) BCH 부호의 복호에 대한 시뮬레이션 하면 $t\leqq 3$ 이하의 어떤 오류의 형태가 발생하더라도 모두 정정됨을 알 수 있었다.

일반적으로 Peterson-Gorenstein-Zierler는 먼저 Vandermonde 행렬로 부터 오류위치다항식을 구한후 Chien의 오류탐지회로를 이용하여 오류위치를 찾아내어 오류를 정정한다.

오류위치다항식의 계수들은 다음과 같다.

Ⅰ)오류가 3개 발생한 경우 오류위치다항식 $\sigma(x)$의 계수는

$$\sigma_1=\frac{1}{D_3}[S_1S_3S_6+S_1S_4S_5+S_2^2S_6+S_2S_3S_5+S_2S_4^2+S_3^2S_4] \quad (26.a)$$

$$\sigma_2=\frac{1}{D_3}[S_1S_4S_6+S_1^4S_5^2+S_2S_3S_6+S_2S_4S_5+S_3^2S_5+S_3S_4^4] \quad (26.b)$$

$$\sigma_3=\frac{1}{D_3}[S_2S_4S_6+S_2S_5^2+S_3^2S_6+S_4^3] \quad (26.c)$$

$$D_3=S_1S_3S_5+S_3^3+S_1S_4^2+S_2^2S_6\neq 0 \quad (26.d)$$

Ⅱ)오류가 2개 발생한 경우

$$\sigma_1=\frac{1}{D_2}[S_1S_4+S_2S_3] \quad (27.a)$$

$$\sigma_2=\frac{1}{D_2}[S_2S_4+S_3^2] \quad (27.b)$$

$$D_2=S_1S_3+S_2^2\neq 0 \quad (27.c)$$

Ⅲ)오류가 1개 발생한 경우

$$\sigma_1=\frac{S_2}{S_1} \quad (28)$$

식(26), (27)을 이용한 Peterson-Gorenstein-Zierler의 복호기법과 본 연구에서 제안한 알고리즘을 승산(제산 포함)횟수를 서로 비교하면〈표1〉과 같다.

표1. 승산 횟수 비교

알고리즘 / 오류정정능력	Peterson Gorenstein Zierler	M_k	X
t = 2	8	2	1
t = 3	30	10	10

〈표1〉에서 보는 바와 같이 오류정정능력이 커질수록 승산 횟수면에서 본 연구의 복호기법이 훨씬 경제적임을 알수있다. 그러므로 유한체내의 승산 연산을 하는 데 k클럭이 소요된다고 하면 본 연구의 알고리즘을 이용하면 기존의 복호기 보다 빠른 처리속도를 갖는 복호기를 만들 수 있다.

표 2 복호기의 복잡도 비교

복호과정 / 알고리즘	오증계산 (시프트레지스트)	$\sigma(x)$ 계산 (ROM)	오류위치판정 (시프트레지스트)
Peterson Gorenstein Zierler	3	30	3
M_k	3	5	·
X	3	6	4

〈표2〉는 M_k를 이용한 복호기를 실제로 구성하여 Peterson-Gorenstein-Zierler복호기와 X를 이용한 복호기를 비교한 것이다. 하드웨어의 구현시 복잡한 유한체 연산이 필요한 M_k는 ROM으로 처리하여 하드웨어의 복잡도를 줄였으며 또한 처리속도의 향상을 기대할 수 있다.
블록부호의 복호과정 중에서 가장 어렵고 복잡한 오류위치다항식을 구하는 과정을 ROM을 이용하여 구현할때 본 연구에서 제안한 M_k를 이용한 복호방법이 가장 효율적임을 알 수 있다.

실제로 (63,45) BCH 부호의 복호기를 구현하는데 소요된 주요 TTL IC 칩은 아래와 같다.

74LS174 (Hex D f/f): 7
74LS688 (Comparator): 3
82321 (32K PROM) : 2
6308 (2K PROM) : 3

V. 결 론

대칭행렬식계산에 의한 2원 BCH 부호의 복호법은 오류위치번호와 미지수 X의 판별식인 대칭행렬식을 오류위치다항식으로 사용할 수 있음을 상세하게 유도하고 복호기구성은 M_k로 나타낸 오류위치다항식을 이용한 것과 X의 다항식으로 나타낸 오류위치다항식을 이용한 것으로 구성한다.

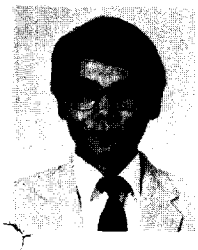
(63,45) BCH 부호에 적용하여 실제로 이 방법이 오류위치다항식의 계수를 구함에 있어 원리적으로 쉽고 이를 시뮬레이션해본 결과 이 방법을 만족한다.

복호기구성은 t가 크면 X의 다항식으로 나타낸 오류위치다항식을 써서 설계하고 t가 작은경우는 M_k로 나타낸 오류위치다항식을 이용한 복호기가 적당하다.

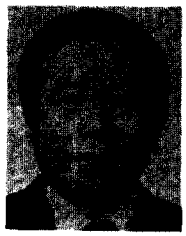
M_k로 나타낸 오류위치다항식을 이용한 복호기는 기존의 Peterson 방법과 소자수가 비슷하고 X의 다항식으로 나타낸 오류위치다항식을 이용한 복호기는 t가 큰 경우 계산수가 Peterson 방법보다 많이 적어지므로 고속복호에 유용하다.

參 考 文 獻

1. 이만영, 부호이론, 희중당, 1984.
2. E.R. Berlekamp. Algebraic Coding Theory, McGraw-Hill, 1968.
3. W.W. Peterson and E.J. Weldon, Jr., Error-Correcting Codes, 2nd ed., MIT Press, Cambridge, Mass., 1972.
4. M. Williams, The Theory of Error-Correcting Codes, North-Holland, 1978.
5. R.C. Bose and C.R. Ray-Chaudhuri, "On a Class of Error-Correcting Binary Group codes", Inf. and Control, Vol.3, pp. 68-79, 1960.
6. R.T. Chien, "Cyclic Decoding Procedures for Bose-Chaudhuri- Hocquenghem Codes", IEEE Trans. Inf. Theory, IT-10, pp. 357-363, 1964.
7. R.E. Blahut, Theory and Practice of Error Control Codes, Addison Wesley, 1983.
8. S. Lin and D.J. Costello, Jr., Error Control Coding Fundamentals and Applications, Prentice-Hall, 1983.
9. 게이찌로 고가, "2원 BCH 부호의 한 복호방식", 일본전자통신학회 논문지, Vol. J66-A No.10 Oct. 1983.
10. R.M. Thrall and L. Tornheim, Vector Spaces and Matrices, Wiley, 1957.



廉 興 烈(Heung Youl YOUM) 正會員
1959年 2月10日生
1981年：漢陽大學校 電子工學科 卒業
1983年：漢陽大學校 大學院 電子工學科 工學碩士
1983年～現在：漢陽大學校 大學院 電子工學科 博士課程中
1982年12月～現在：韓國電子通信研究所 전송시스템研究室 先任研究員



李 晩 榮(Man Young RHEE) 正會員
1924年11月30日生
서울大學校電氣工學科 卒業
美國Colorado大學院卒業(工學博士 1958年)
美國Boeing會社研究員
美國Virginia工大教授
美國California工大 JPL NASA研究員
國防科學研究所副所長
韓國電子通信(株)代表理事社長
漢陽大學校工科大學電子通信工學科 教授