通信情報保護學會論文誌
第 6 卷 第 1 號, 1996. 3

대수적 복호에 의한 Golay 부호의 고속 복호기 설계

김 창 규*

High-Speed Algebraic Decoding of the Golay Codes

Chang-Kyu Kim

요 약

오증 요소로부터 오류위치다항식의 계수를 계산함으로서 (23,12) Golay 부호를 복호할 수 있는 대수적 복호법이 최근 증명되었다. GF(2^{11}) 상에서의 3중 오류정정 BCH 부호의 복호법을 이 부호에 완벽하게 적용하여 해석하는 것을 소개한다. 그리고 GF(2^{11})에 대한 최적의 정규기저를 구하여 이를 유한체 연산에 적용하며 단계별로 복호 회로의 구성을 제시한다. 이는 기존의 복호기보다 논리회로적으로 간단하며, 복호된 정보를 얻기까지 35번의 치환이 필요하다.

Abstract

An algebraic method for decoding the (23,12) Golay code by computing the relation between the coefficients of the error-locator polynomial and the syndrome components has been proved recently. A complete analysis of the decoding procedure over GF(2^{11}) for triple-error correcting BCH codes applicable to decoding of this code is introduced. A step-by-step complete decoding circuit is proposed, which is composed of the circuit for calculation of the optimal normal basis elements of GF(2^{11}). This decoder provides less complexity than existing decoder and requires 35 shift-operations for decoding one word.

1. 서 론

2원 Golay 부호는 오류정정능력이 3인 완전 부호(perfect code)로 여러 가지 복호법이 연구되어왔다. 이들 중 가장 잘 알려진 것이 오류포착(error trapping)기법을 이용한 Kasami 복호기[1]이며 다수결논리를 이용한 임계복호법이 Goethals[2,5]에 의해 제시되기도 하였다. 한편

* 동의대학교 전자통신공학과

이 논문은 1994학년도 한국학술진흥재단의 공모과제 연구비에 의하여 연구되었음.

BCH 부호의 단계별(step-by-step) 복호 알고리듬[3]을 이용한 Golay 부호의 복호가 Chien과 Lum[4]에 의해 최초로 제시되었으며, 최근에는 3중 오류정정 BCH 부호의 대수적 복호법을 적용하여 (23,12) Golay 부호를 완전하게 복호할 수 있음을 Elia[5]는 증명하였다. 이는 유한체(finite field) $GF(2^{11})$ 상에서의 연산으로 복호가 가능하므로 유한체 원소들의 연산에 따르는 회로의 복잡성을 지니고 있다. Wei[6]는 이 결과를 수정하여 이 변형된 결과와 오증(syndrome)으로부터 얻어지는 판정벡터(decision vector)를 이용하여 Kasami의 오류포착 복호기보다 치환회수를 줄일 수 있는 고속 복호법을 제시하였다. Kasami의 복호법은 부호의 순회성을 이용하여 엄호다항식(covering polynomial)을 생성다항식(generator polynomial)으로 나눈 나머지와 오증으로부터 오류의 포착 상태를 파악하는 기법으로 오류가 정정된 정보를 얻기까지 최대 58번의 치환이 필요하다. 반면 Wei의 복호법은 판정회로에 의해 바로 오류의 위치를 발견할 수 있어서 정보를 얻기까지 모두 35번의 치환이 필요하다. 하지만 이 복호법에서는 수신된 계열을 역순으로 바꾸어야 하는 단점이 있다. 본 논문에서는 Elia가 증명한 대수적 복호법을 기반으로 유한체 상에서의 연산의 복잡도를 최소화할 수 있는 $GF(2^{11})$의 정규기저[7]를 구하고 이 정규기저로 원소를 표현함으로서 Wei의 복호기와 같은 회수의 치환이 필요하며 대수적 연산 과정에서 복잡성이 덜한 회로의 구성 방법을 제시한다.

2. 2원 Golay 부호

순회부호(cyclic code)에 속하는 (23,12) Golay 부호는

$$g(x) = 1 + x + x^5 + x^6 + x^7 + x^9 + x^{11} \quad (1)$$

에 의해 생성되며 $GF(2)$ 상에서 $1 + x^{23}$의 인수임은 쉽게 알 수 있다. 생성다항식 $g(x)$는 $GF(2)$의 확대체 $GF(2^{11})$의 임의의 원소에 대한 최소다항식(minimal polynomial)의 계산으로 얻어진다. α를 유한체 $GF(2^{11})$의 원시원소(primitive element)라 하고 $\beta = \alpha^{89}$라 하면 β의 공액(conjugate)은 다음과 같이 $GF(2^{11})$의 원소 11개의 집합으로 나타나며

$$R = \{\beta, \beta^2, \beta^3, \beta^4, \beta^6, \beta^8, \beta^9, \beta^{12}, \beta^{13}, \beta^{16}, \beta^{18}\} \quad (2)$$

이들이 모두 근이 되는 다항식을 $GF(2)$ 상에서 구하면 식 (1)과 같다. (23,12) Golay 부호의 부호다항식 $c(x)$는 12비트의 정보다항식 $d(x)$에 의해 생성되는데 조직(systematic) 형태로 생성되려면

$$c(x) = x^{11}d(x) + v(x) \quad (3)$$

로 되어야 하며, 이때 $v(x)$는 $x^{11}d(x)$를 생성다항식 $g(x)$로 나눈 나머지이고 $c(x)$는 $g(x)$의 곱의 형태가 된다.

3. 정규기저

일반적으로 유한체 $GF(2^m)$의 원소들은 다항식표현, 벡터표현 또는 지수표현으로 나타낸다. $GF(2^m)$의 선형독립인 m개의 원소를 사용하여 임의의 원소를 다항식으로 표현할 수 있을 때 이 m개의 원소를 $GF(2^m)$의 기저(basis)라 한다. 유한체 $GF(2^m)$의 원시원소를 α라 하면 $1, \alpha, \alpha^2, \alpha^3, \cdots, \alpha^{m-1}$는 $GF(2^m)$의 기저가 되며 임의의 원소 β를 $GF(2)$ 상의 다항식

$$\beta = a_0 + a_1\alpha + a_2\alpha^2 + \cdots + a_{m-1}\alpha^{m-1} \quad (4)$$

로 표현할 수 있다. 이를 유한체의 표준기저(standard basis)라 한다.

유한체 상에서의 연산에서 덧셈은 용이하지

만 곱셈 또는 나눗셈은 구현이 복잡하다. 그러나 정규기저(normal basis)를 사용하면 연속적인 순회치환(cyclic shift)에 의해 구현의 복잡함을 해결할 수 있다. $GF(2^m)$의 원소 중 m개의 원소 $N = \{\delta, \delta^2, \delta^4, \cdots, \delta^{2^{m-1}}\}$은 임의의 원시원소 δ에 대해 $GF(2^m)$의 기저가 될 수 있으며 임의의 원소 γ는 다음과 같이 $GF(2)$ 상의 다항식으로 유일하게 표현된다.

$$\gamma = b_0\delta + b_1\delta^2 + b_2\delta^4 + \cdots + b_{m-1}\delta^{2^{m-1}} \quad (5)$$

이 m개의 원소를 유한체 $GF(2^m)$의 정규기저라 한다. (5)식과 같이 표현된 $GF(2)$ 상의 다항식을 자승하면

$$\gamma^2 = b_{m-1}\delta + b_0\delta^2 + b_1\delta^4 + \cdots + b_{m-2}\delta^{2^{m-1}} \quad (6)$$

가 된다. 그리고 γ를 m차원 벡터 $\gamma = (b_0, b_1, b_2, \cdots, b_{m-1})$로 표현하면 그것의 자승은 $\gamma^2 = (b_{m-1}, b_0, b_1, \cdots, b_{m-2})$로 표현된다. 즉, 정규기저 표현에서는 γ^2을 얻기 위해서는 γ를 한번 순회치환하면 된다. 따라서 정규기저를 사용하면 $GF(2^m)$의 연산에서 임의의 원소의 자승은 간단한 논리회로에 의해 구현된다.

$\alpha = (a_0, a_1, \cdots, a_{m-1})$와 $\beta = (b_0, b_1, \cdots, b_{m-1})$가 정규기저로 표현된 $GF(2^m)$의 두 원소라 하면 두 원소의 곱 $\gamma = (c_0, c_1, \cdots, c_{m-1})$의 마지막 요소 c_{m-1}은 α와 β의 요소들의 2진함수

$$c_{m-1} = f(a_0, a_1, \cdots, a_{m-1}; b_0, b_1, \cdots, b_{m-1}) \quad (7)$$

의 결과이다. 그리고 정규기저 표현에서 임의의 원소의 자승은 한번의 순회치환으로 이루어지므로 γ^2의 마지막 요소 c_{m-2}는 α^2와 β^2를 수행한 후에 나타나는 요소들을 사용하여 (7)식과 같은 논리함수에 의해 구해진다. 즉,

$$c_{m-2} = f(a_{m-1}, a_0, \cdots, a_{m-2}; b_{m-1}, b_0, \cdots, b_{m-2}) \quad (8)$$

가 된다. 따라서 α와 β를 순회치환하면서 그것의 요소를 함수 f의 입력으로 삼으면 두 원소의 곱을 그림 1과 같은 방법으로 구할 수 있으며 이를 Massey-Omura 승산기[8]라 한다.

유한체 $GF(2^m)$의 임의의 원소에 어떤 원소를 나눈다는 것은 그 원소의 역원을 곱하는 것과 같다. $GF(2^m)$의 원소 γ에 대해 이것의 역원은 $\gamma^{-1} = \gamma^{2^m-2}$이므로

$$\gamma^{-1} = (\gamma^2)(\gamma^{2^2})(\gamma^{2^3})\cdots(\gamma^{2^{m-1}}) \quad (9)$$

로 표현할 수 있다. 따라서 γ가 정규기저로 표현되면 순회치환에 의해 자승한 값을 얻을 수

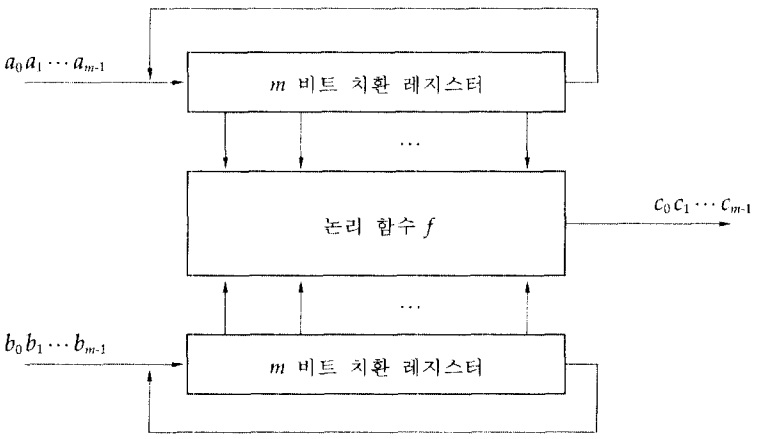

그림 1 GF(2^m) 상의 Massey-Omura 승산기

있으므로 어떤 원소의 역원은 최대 m번의 연속적인 순회치환과 앞에서 언급한 Massey-Omura 승산기를 이용하여 구해질 수 있다[8].

이상에서 보듯이 유한체의 원소를 정규기저로 표현하면 표준기저로 표현할 때보다 하드웨어 구현시 훨씬 간단하게 곱셈과 나눗셈이 수행된다. 그러나 정규기저의 발견이 쉽지 않으며 정규기저 중에서도 최소의 곱셈항을 갖는 논리함수에 의해 승산이 수행될 수 있는 최적의 정규기저를 발견하여야 한다. 본 논문에서는 원시다항식(primitive polynomial)을 $p(x) = 1 + x^2 + x^{11}$로 했을 때 존재하는 여러 가지 $GF(2^{11})$의 정규기저를 구하고 이 중 연산의 복잡도를 최소화할 수 있는 정규기저를 찾아서 유한체의 연산에 적용하였다. 보통의 정규기저는 논리함수가 50개 이상의 곱셈항을 가진다. 그러나 $\delta = \alpha^{139}$인 정규기저를 사용하면 최소의 곱셈항을 가지는 논리함수가 구해진다.

$$N = \{\alpha^{139},\ \alpha^{878},\ \alpha^{1756},\ \alpha^{1465},\ \alpha^{883},\ \alpha^{1766},\ \alpha^{1485},\ \alpha^{923},\ \alpha^{1846},\ \alpha^{1645},\ \alpha^{1243}\} \tag{10}$$

를 $GF(2^{11})$의 정규기저로 사용하여 원소를 표현하고 임의의 두 원소의 곱셈을 그림 1과 같이 수행할 수 있는 논리함수를 (7)식으로 표현하면 아래와 같이 곱셈항이 21개 존재하는 최적의 경우가 된다.

$$\begin{aligned} c_{10} = {} & a_0b_7 + a_0b_{10} + a_1b_5 + a_1b_7 + a_2b_3 \\ & + a_2b_4 + a_3b_2 + a_3b_8 + a_4b_2 + a_4b_6 \\ & + a_5b_1 + a_5b_8 + a_6b_4 + a_6b_9 + a_7b_0 \\ & + a_7b_1 + a_8b_3 + a_8b_5 + a_9b_6 + a_9b_9 \\ & + a_{10}b_6 \end{aligned} \tag{11}$$

즉, 식 (11)과 같이 논리함수를 구성하고 두 원소를 순회치환시켜면서 논리함수의 입력이 되게 하면 두 원소를 곱한 결과가 구해진다.

4. Golay 부호의 복호

4.1 오증과 오류위치다항식

일반적으로 BCH 부호의 복호는 1)수신계열로부터 오증을 계산하고 2)계산된 오증으로부터 오류위치다항식(error-locator polynomial)의 계수를 구하며 3)오류위치다항식의 근을 구함으로서 오류의 위치를 알아내는 3단계로 이루어진다. Golay 부호의 대수적 복호는 BCH 부호의 복호와 같은 단계별로 복호하는 것이다. 먼저 (1)식으로 표현되는 생성다항식 $g(x)$의 근 중 β, β^3, β^9을 수신다항식 $r(x)$에 대입하여 3개의 오증 요소를 구한다.

$$s_1 = r(\beta),\quad s_3 = r(\beta^3),\quad s_9 = r(\beta^9) \tag{12}$$

(3)식으로 표현되는 조직형 (23,12) Golay 부호는 부호다항식 $c(x)$가 $g(x)$의 곱이므로 오증은 $r(x)$를 $g(x)$로 나눈 나머지 $w(x)$에 $g(x)$의 근을 대입시킨 결과와 동일하다.

$$s_1 = w(\beta),\quad s_3 = w(\beta^3),\quad s_9 = w(\beta^9) \tag{13}$$

각 오증 요소는 $GF(2^{11})$의 원소 중 하나로 나타나며 11비트의 벡터로 표현할 수 있다. 이를 3절에서 언급한 최적의 정규기저에 의한 벡터 표현을 이용하면 (14)식과 같은 행렬로 나타낼 수 있다.

$$\mathbf{s}_1 = \mathbf{w} \cdot \begin{bmatrix} 11111111111 \\ 11011001010 \\ 01101100101 \\ 11001010110 \\ 10110110010 \\ 00101011001 \\ 01100101011 \\ 10101100100 \\ 01011011001 \\ 01010110110 \\ 10010101100 \end{bmatrix} \tag{14}$$

$$\mathbf{s}_2 = \mathbf{w} \cdot \begin{bmatrix} 11111111111 \\ 11001010110 \\ 01100101011 \\ 01010110110 \\ 10110010101 \\ 01011001001 \\ 00101011011 \\ 01100100101 \\ 11011001010 \\ 10110110010 \\ 10101100100 \end{bmatrix} \quad (15)$$

$$\mathbf{s}_3 = \mathbf{w} \cdot \begin{bmatrix} 11111111111 \\ 01010110110 \\ 00101011011 \\ 10110110010 \\ 10010101101 \\ 11001001010 \\ 01011011001 \\ 00100101011 \\ 11001010110 \\ 10110010101 \\ 01100100101 \end{bmatrix} \quad (16)$$

여기서, $\mathbf{w}$는 수신다항식 $r(x)$를 생성다항식 $g(x)$로 나눈 나머지 $w(x)$를 벡터로 표현한 것이다. $GF(2^{11})$의 한 원소로 나타나는 오증요소는 수신계열을 생성다항식으로 나눈 나머지인 $\mathbf{w}$의 각 비트를 (14), (15), (16)식에 따라 비트별 2원합으로 얻어지는 것이다. 예를 들어 $\mathbf{s}_1$의 마지막 비트는 $\mathbf{w}$의 첫번째, 세번째, 여섯번째, 일곱번째 그리고 아홉번째 비트를 2원합한 것이다.

오증이 계산되면 복호의 다음 단계는 오류위치다항식 $\sigma(x)$를 구하는 것이다. j가 오류의 위치를 나타낼 때 오류위치다항식은 오류위치번호 $y = \beta^j$의 역수가 근이 되도록 정의한다. 오류가 없는 경우라면 $s_1 = s_3 = s_9 = 0$가 되고 $\sigma(x) = 1$일 것이며 수신계열에 하나의 오류가 발생하였다면 틀림없이 $s_1^3 = s_3$, $s_3^3 = s_9$이며 $\sigma(x) = 1 + yx$가 된다. 그리고, 둘 또는 세개의 오류가 발생하였다면

$$\begin{aligned} \sigma(x) &= (1+y_1x)(1+y_2x)(1+y_3x) \\ &= 1 + \sigma_1x + \sigma_2x^2 + \sigma_3x^3 \end{aligned} \quad (17)$$

와 같이 정의되고 오류위치다항식의 계수와 오류위치번호와의 관계는

$$\begin{aligned} \sigma_1 &= y_1 + y_2 + y_3 \\ \sigma_2 &= y_1y_2 + y_2y_3 + y_3y_1 \\ \sigma_3 &= y_1y_2y_3 \end{aligned} \quad (18)$$

로 된다. (13)식을 보면 오증 요소는

$$s_1 = y_1^i + y_2^i + y_3^i, \quad i = 1, 3, 9 \quad (19)$$

로 계산된다.

$$K = (s_1^3 + s_3)^2 + (s_1^9 + s_9)/(s_1^3 + s_3) \quad (20)$$

라 정의하면 $K = (\sigma_2 + s_1^2)^3$이고 (18), (19)식으로부터 $\sigma_3 + s_3 = \sigma_2s_1 + s_1^3$이므로 (21)식이 얻어진다[5]. 그리고 $\sigma_1 = s_1$이 됨은 틀림없다.

$$\begin{aligned} \sigma_2 &= s_1^2 + K^{1/3} \\ \sigma_3 &= s_3 + s_1K^{1/3} \end{aligned} \quad (21)$$

여기서 $K^{1/3}$의 연산은 $GF(2^{11})$ 상에서는 K^{1365}로 계산이 가능하다.

이상에서 기술한 (23,12) Golay 부호의 복호과정을 구성하면 그림2와 같다. 여기서 $(\ )^2$은 입력을 한번 순회치환하면 되고, $(\ )^3$은 한번 순회치환된 내용과 원래의 내용을 입력으로 (11)식의 논리함수로 구현되는 승산기에 의해 얻어질 수 있다. 또 $(\ )^{-1}$은 Massey-Omura 역원기에 의해 수행이 가능하며 $K^{1/3}$은 다음의 알고리듬으로 계산될 수 있다. γ가 정규기저로 표현된 $GF(2^{11})$의 한 원소라면

$$\gamma^{1365} = \gamma^1\gamma^{2^2}\gamma^{2^4}\gamma^{2^6}\gamma^{2^8}\gamma^{2^{10}} \quad (22)$$

이므로 정규기저로 원소를 표현하게 되면 아래와 같이 수차례의 순회치환과 승산으로 연산이 가능하다.

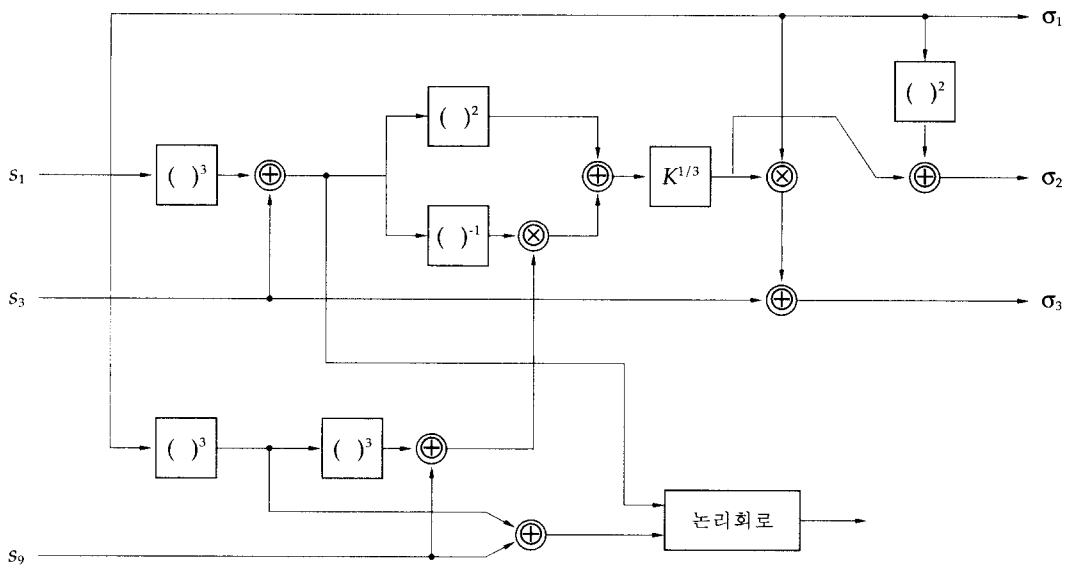

그림 2 (23,12) Golay 부호의 오류위치다항식 계수 계산 회로

(1) γ를 두번 순회치환하여 $A = \gamma^{2^2}$를 구한다. 그리고 $B = \gamma$, $l = 0$라 놓는다.

(2) A와 B를 곱하여 $C = AB$를 구하고 $l = l + 1$로 놓는다.

(3) i) 만약 $l = 5$이면 $C = \gamma^{1/3}$가 되고 연산을 멈춘다.

ii) 만약 $l < 5$이면 A를 두번 순회치환하고 $B = C$로 대치하여 (2)를 수행한다.

그리고 그림에서 논리회로는 11입력 NOR 게이트 2개와 AND 게이트로 구성되고 그 값이 '1'이면 단일 오류가 발생한 경우이므로 $\sigma_2 = \sigma_3 = 0$이 되도록 하여야 하며 ⊕와 ⊗는 $GF(2^{11})$ 상의 가산과 승산을 각각 표시한다..

4.2 오류정정

오류위치의 발견은 Chien의 탐지회로[9,10]를 사용하면 된다. 오류위치다항식 $\sigma(x)$가 계산되면 $\sigma(x)$에 $GF(2^{11})$의 원소 $\beta^i = (\alpha^{89})^i$, $1 \le i \le 23$를 대입하여 $\sigma(x)$의 근을 구함으로서 오류위치를 알 수 있다. β^i가 근이라면 3개 이하의 오류가 발생한 경우

$$\sigma(\beta^i) = 1 + \sigma_1\beta^i + \sigma_2\beta^{2i} + \sigma_3\beta^{3i} = 0 \qquad (23)$$

또는

$$\sum_{k=1}^{3} \sigma_k(\beta^i)^k = 1, \quad 1 \le i \le 23 \qquad (24)$$

이다. (23), (24)식에서 보는 바와 같이 오류위치다항식의 계수에 $\beta^k(k = 1, 2, 3)$를 각각 곱하면서 이들의 합이 '1'인지를 알아본다. 만약 i번 곱했을 때 합이 '1'이라면 β^i는 $\sigma(x)$의 근이며 오류위치번호는

$$\beta^j = 1/\beta^i = \beta^{23-i}$$

가 된다. 이와 같이 (23,12) Golay 부호의 오류위치를 알 수 있는 장치는 σ_k에 β^k를 곱하는

$GF(2^{11})$ 상에서의 승산회로가 필요하며 이는 2원 가산기와 귀환선(feedback line), 그리고 합이 '1'인지 알아보기 위한 11-입력 NOR 게이트로 구성된다. $GF(2^{11})$ 상에서 원소를 (10)식에서 제시한 최적의 정규기저로 표현하면 σ_k에 β^k를 곱하는 식은 다음과 같이 행렬로 표현할 수 있다.

$$\sigma_1\beta = (\sigma_{1,0}\delta + \sigma_{1,1}\delta^2 + \sigma_{1,2}\delta^4 + \cdots + \sigma_{1,10}\delta^{2^{10}})\beta$$

$$= \sigma_1 \cdot \begin{bmatrix} 11010010111 \\ 10000111111 \\ 00100111000 \\ 10010011010 \\ 00000011100 \\ 01100001111 \\ 11111011010 \\ 01111110010 \\ 11001100101 \\ 11010111001 \\ 11000100111 \end{bmatrix} \quad (25)$$

$$\sigma_2\beta^2 = (\sigma_{2,0}\delta + \sigma_{2,1}\delta^2 + \sigma_{2,2}\delta^4 + \cdots + \sigma_{2,10}\delta^{2^{10}})\beta^2$$

$$= \sigma_2 \cdot \begin{bmatrix} 11100010011 \\ 11101001011 \\ 11000011111 \\ 00010011100 \\ 01001001101 \\ 00000001110 \\ 10110000111 \\ 01111101101 \\ 00111111001 \\ 11100110010 \\ 11101011100 \end{bmatrix} \quad (26)$$

$$\sigma_3\beta^3 = (\sigma_{3,0}\delta + \sigma_{3,1}\delta^2 + \sigma_{3,2}\delta^4 + \cdots + \sigma_{3,10}\delta^{2^{10}})\beta^3$$

$$= \sigma_3 \cdot \begin{bmatrix} 10011010100 \\ 00011100000 \\ 00001111011 \\ 11011010111 \\ 11110010011 \\ 01100101110 \\ 10111001110 \\ 00100111110 \\ 10010111110 \\ 00111111100 \\ 00111000001 \end{bmatrix} \quad (27)$$

이들은 각각 11비트로 표현되는 $GF(2^{11})$의 한 원소이며 귀환선은 (25), (26), (27)식에 따라 구성된다. 예를 들어 σ_3의 두번째 비트는 네번째, 다섯번째, 그리고 여섯번째 비트의 2원합이 이것의 입력으로 귀환되도록 Chien의 탐지회로를 구성하면 된다.

5. 결 론

BCH 부호의 대수적 복호와 같은 방법으로 계산된 오증으로부터 오류위치다항식을 구하고 오류위치를 알아내어 오류를 정정하는 Golay 부호의 대수적 복호 과정에서 대두되는 것은 유한체 상의 연산을 얼마나 간단하게 구현할 수 있느냐 하는 것이다. 본 논문에서는 $GF(2^{11})$의 정규기저 중 승산기에서 논리함수의 곱셈항이 21개인 최적의 정규기저를 구하고 이를 바탕으로 Elia가 증명한 대수적 복호법에 따라 (23,12) Golay 부호의 복호에 대해 단계별로 회로의 구성을 제시하였다. 오류가 정정된 정보를 얻기까지 Kasami의 오류포착 복호기는 수신계열을 최대 58번 치환하여야 하지만 논문에 제시된 복호기는 Wei의 복호기와 같이 35번의 치환이 필요하며, 수신계열을 역순으로 바꾸고 판증벡터를 구하여 이를 복호에 이용하는 Wei의 복호기에 비해 논리회로면에서 간단하게 구현할 수 있다.

참 고 문 헌

[1] T. Kasami, "A decoding procedure for multiple-error-correction cyclic codes," IEEE Trans. Inf. Theory, vol. IT-10, pp. 134-139, 1964.

[2] F. J. MacWilliams and N. J. A. Sloane, The Theory of Error-Correcting Codes, Amsterdam, The Netherlands: North-Holland, 1977.

[3] J. L. Massey, "Step-by-step decoding of the Bose-Chaudhuri-Hocquenghem codes," IEEE Trans. Inf. Theory vol. IT-11, pp. 580-585, 1965.

[4] R. T. Chien and V. Lum, "On Golay's perfect codes and step-by-step decoding,"IEEE Trans. Inf. Theory, vol. IT-12, pp. 403-404, 1966.

[5] M. Elia, "Algebraic decoding of the (23,12,7) Golay codes," IEEE Trans. Inf. Theory, vol. IT-33, pp. 150-151, 1987.

[6] S. W. Wei and C. H. Wei, "On high-speed decoding of the (23,12,7) Golay codes," IEEE Trans. Inf. Theory, vol. IT-36, pp. 692-695, 1990.

[7] A. J. Menezes, Application of Finite Fields, Kluwer Academic Publishers, Norwell, Mass. 1993.

[8] C. C. Wang, T. K. Truong, H. M. Shao, J. K. Omura and I. S. Reed, "VLSI architectures for computing multiplications and inverses in GF(2^m)," IEEE Trans. Computers, vol. C-34, pp. 709-716, 1985.

[9] R. T. Chien, "Cyclic decoding Procedure for the BCH codes," IEEE Trans. Inf. Theory, vol. IT-10, pp. 357-363, 1964.

[10] M. Y. Rhee, Error Correcting Coding Theory, McGraw-Hill, 1989.

□ 著者紹介

김 창 규(金 彰 圭)

1981년 한양대학교 전자통신공학과(학사)
1984년 한양대학교 대학원 전자통신공학과(석사)
1989년 한양대학교 대학원 전자통신공학과(박사)
1988년 3월 ~ 현재 동의대학교 전자통신공학과 부교수

※ 주관심분야 : 암호이론, 부호이론