通信情報保護學會論文誌
第 7 卷 第 1 號, 1997. 3

5중 오류정정 (255, 215) BCH 부호의 효율적인 복호 알고리즘과 이의 VHDL 시뮬레이션

강 경 식*, 박 진 수**

Efficient Decoding Algorithm of 5-error-correcting (255, 215) BCH Code And Its Simulation with VHDL

Kyung-Sik Kang, Jin-Soo Park

요 약

본 논문에서는, 무선 통신시스템에 적용 가능한 (255, 215) BCH 부호의 효율적인 복호 알고리즘을 제안하고, 이를 이용하여 5중 에러 정정 부호기 및 복호기를 설계하였다. peterson의 복호기보다 곱셈기, X-OR 게이트의 수가 현저히 줄어들었을 뿐만 아니라 역원계산기가 필요 없음이 입증되었고, VHDL을 사용한 컴퓨터 시뮬레이션을 통해서 그 타당성을 검증하였다.

Abstract

In this paper, efficient decoding algorithm of (255, 215) BCH codes which is applicable to the wireless communication system is proposed. By using the proposed decoding algorithm, encoder and decoder of (255, 215) BCH codes with 5-error-correcting capability are designed. As a result , we have shown that the number of multipliers and EX-OR gates are considerably reduced compared with the Peterson's decoder, and complex inversion circuits in GF(28) are avoidable. And to show the validity of the presented algorithm, computer simulations are performed with VHDL. As a result of simulation, the validity of decoding algorithm is demonstrated.

1. 서 론

디지탈 통신 방식과 컴퓨터의 응용기술이 발달함에 따라 정보의 전송, 교환 처리를 위하여 신속하고 신뢰성 높은 대용량의 통신망이 요구되고 있다. 그러나 정보를 전달하는데 있

* 주성전문대학 전자과
* * 청주대학교 전자 · 반도체 · 정보통신공학부

어서 피할 수 없는 문제의 하나는 통신로상의 잡음으로 인한 신뢰성의 저하이다. 따라서 정보를 송수신하는 과정에서 발생하는 오류를 효과적으로 제어하기 위한 오류정정부호이론은 디지탈 통신 시스템의 신뢰도를 보장할 수 있게 되었다.

1948년 Shannon에 의하여 부호화 이론이 처음 제기된 이후에 부호이론은 블록부호(Block Code)와 길쌈부호(Convolutional Code)로 크게 나누어 발전하였으며 특히 BCH부호의 복호알고리즘은 Peterson에 의해 처음 제안되었고 Chien은 오류위치 다항식의 $\sigma(x)$의 근을 구하는 효율적인 방법을 연구하였다. 또한 Berlekamp는 복호시 가장 어려운 과정인 오류위치 다항식을 구하는 방법을 제안하였고, Berlekamp의 반복알고리즘을 Massey가 LFSR(Linear Feedback Shift Register)로 설계하면서 H/W적으로 실현하였다. 또한 Burton은 이 Berlekamp-Massey 알고리즘을 제산이 필요없는 이원 BCH부호에 적용시켰다. BCH 부호는 블럭 부호중에서 산발 오류 정정능력이 뛰어난 부호로서 부호의 대수학적 정교함을 이용한 복호 알고리즘이 Peterson, Berlekamp, Chein등에 의하여 개발되었다.

본 논문에서는 오증으로부터 직접 오류의 갯수 및 오류 값을 결정하고, 오류를 정정하는 방법을 이용하여 t=5인 (255, 215) BCH 부호의 부호기 및 복호기를 설계하고, 복호알고리즘의 타당성을 VHDL로 검증하였다.

및 통신 상대방의 확인을 행할 수 있는 기능이다.

2. 일반적인 BCH 부호의 복호방법

부호장이 $n=2^m-1$인 2원 t중 오류 정정 BCH부호의 생성다항식 $q(x)$는 근 $\alpha, \alpha^2 \cdots\cdots \alpha^{2t}$에 대응하는 최소다항식 $m_i(x)$ $\le i \le 2t$의 최소공배수이다.

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

따라서, BCH 부호의 부호어는 생성행렬의 근 $\alpha, \alpha^2 \cdots\cdots \alpha^{2t}$을 근으로 갖는다.

이를 행렬식으로 표시하면 식(2)와 같다.

$$(c_0, c_1, c_2 \cdots c_{n-1}) \cdot \begin{bmatrix} (\alpha^i)^0 \\ (\alpha^i)^1 \\ \cdot \\ \cdot \\ (\alpha^i)^{n-1} \end{bmatrix} = 0,\ 1 \le i \le 2t \quad (2)$$

식 (2)로부터 BCH 부호의 검사다항식은 식(3)과 같음을 알 수 있다.

$$H = [(a^i)^0 \cdot (a^i)^1 \cdot (a^i)^2 \cdot \cdot \cdot (a^i)^{n-1}],\ 1 \le i \le 2t \quad (3)$$

그러므로 BCH 부호에서는 식(4)의 관계가 만족된다.

$$\overline{c} \cdot H^T = \overline{0} \quad (4)$$

부호의 최소거리 $d \ge 2t+1$인 경우에 t개 이하의 임의의 산발 오류를 모두 정정할 수 있는데 부호어 C(x)를 전송했을 때 수신계열 r(x)는 다음과 같다.

$$r(x) = c(x) + e(x) \quad (5)$$

수신어 $\overline{0}$에 대하여 오증 $S(S_1, S_2, \cdots S_{2t})$은

$$S = \overline{r} \cdot H^T = (\overline{c} + \overline{e}) H^T = \overline{e} \cdot H^T \quad (6)$$

로 정의된다.

$v(1 \le v \le t)$개의 오류가 미지의 위치 $j_1, j_2, \cdots j_v$에서 발생했다면 오류 형태는

$$e(x) = \sum_{\lambda=0}^{v} x^{j_\lambda} \quad 0 \le j_\lambda \le n-1 \tag{7}$$

오증 요소는

$$S_k = \sum_{\lambda=1}^{v} (\alpha^k)^{j_\lambda} = \sum_{\lambda=1}^{v} (\alpha^{j_\lambda})^k = \sum_{\lambda=1}^{v} (\beta_\lambda)^k \tag{8}$$

이 되며, 오증요소 $S_k : 1 \le k \le 2t$와 오류 위치 번호 $\beta_\lambda = \alpha^{j_\lambda} (1 \le \lambda \le v)$간의 결합된 방정식 조합을 만든다. BCH복호기법은 식(8)의 해를 구하는 방법과 오증 요소로부터 오류 위치 번호 α^{j_λ}, $\lambda = 1, 2, \cdots, v$를 찾아내는 것이다.

3. BCH부호의 효율적인 복호알고리즘

본 논문에서 제시한 복호알고리즘은 다음과 같다.

오류위치 다항식은 식 (9)과 같다.

여기서, v는 실제로 오류가 발생한 갯수이다($v \le t$).

$$\begin{aligned} \sigma(x) &= \sigma_0 + \sigma_1 x + \sigma_2 x^2 + \cdots + \sigma_v x^v \\ &= (1+\beta_1 x)(1+\beta_2 x) \cdots (1+\beta_v x) \end{aligned} \tag{9}$$

$\sigma(x)$의 근 β_i^{-1}의 역수인 β_i이 오류위치가 된다. BCH부호의 오류정정능력은 t이므로 이후부터는 t개의 오류가 발생했음으로 가정하고 식으로 전개한다.

오증 요소 S_i로부터 오류 위치 다항식 $\sigma(x)$의 계수 σ_k값을 구하기 위해서 Newton의 항등식이 필요하다.

$$\begin{aligned} &S_1 + \sigma_1 S_{2t-1} + \cdots = 0 \\ &S_2 + \sigma_1 S_1 + 2\sigma_2 = 0 \\ &S_3 + \sigma_1 S_2 + \sigma_2 S_1 + 3\sigma_3 = 0 \\ &\vdots \\ &S_{2t-1} + \sigma_1 S_{2t-2} + \cdots + \sigma_t S_{t-1} = 0 \end{aligned} \tag{10}$$

v가 홀수인 경우 $v\sigma_v = \sigma_v$이고 v가 짝수인 경우 $v\sigma_v = 0$이므로 식(10)을 다시 쓰면

$$\begin{aligned} &S_1 + \sigma_1 = 0 \\ &S_3 + \sigma_1 S_2 + \sigma_2 S_1 + \sigma_3 = 0 \\ &\vdots \\ &S_{2t-1} + \sigma_1 S_{2t-2} + \sigma_2 S_{2t-3} + \cdots + \sigma_{t-1} = 0 \end{aligned} \tag{11}$$

식 (11)를 행렬로 표시하면

$$\begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ S_2 & S_1 & 1 & \cdots & 0 \\ \vdots & & & & \\ S_{2t-2} & S_{2t-3} & \cdots & \cdots & S_{t-1} \end{bmatrix} \begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \\ \sigma_t \end{bmatrix} = \begin{bmatrix} S_1 \\ \\ S_3 \\ S_{2t-1} \end{bmatrix} \tag{12}$$

$$A \cdot \sigma = B \tag{13}$$

$$A \triangleq \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ S_2 & S_1 & 1 & \cdots & 0 \\ \vdots & & & & \\ S_{2t-2} & S_{2t-3} & \cdots & \cdots & S_{t-1} \end{bmatrix}$$

$$\sigma \triangleq \begin{bmatrix} \sigma_1 \\ \\ \end{bmatrix} \qquad B \triangleq \begin{bmatrix} S_1 \\ S_3 \\ \\ S_{2t-1} \end{bmatrix} \tag{14}$$

여기서 |A|는 실제 수신부호에서의 오류의 갯수를 판별하는데 사용될 수 있다.

행렬식 $|A| \neq 0$이면 식(13)을 이용해서 오류 위치 다항식을 구할수 있다. 식(13)는 식(15)와 등가이다.

$$\begin{bmatrix}\sigma_1\\ \sigma_2\\ \sigma_3\\ \vdots\\ \sigma_t\end{bmatrix}=\frac{1}{|A|}\begin{bmatrix}A_{1,1} & A_{1,2} & \cdots & A_{t,1}\\ A_{1,2} & A_{2,2} & \cdots & A_{t,2}\\ \vdots & & & \\ A_{1,t} & A_{2,t} & \cdots & A_{t,t}\end{bmatrix}\cdot\begin{bmatrix}S_1\\ S_3\\ \vdots\\ S_{2t-1}\end{bmatrix} \quad (15)$$

식(15)의 각 요소는 식(16)와 같다.

$$\sigma_1=\frac{1}{|A|}(A_{1,1}S_1+A_{2,1}S_3+\cdots+A_{t,1}S_{2t-1})$$
$$\sigma_1=\frac{1}{|A|}(A_{1,2}S_1+A_{2,2}S_3+\cdots+A_{t,2}S_{2t-1}) \quad (16)$$
$$\sigma_t=\frac{1}{|A|}(A_{1,t}S_1+A_{2,t}S_3+\cdots+A_{t,t}S_{2t-1})$$

이를 일반화하면 식(17)과 같다.

$$\sigma_k=\frac{1}{|A|}\sum_{i=1}^{t}A_{i,k}S_{2i-1} \qquad k=1,2,3,\cdots\cdots t \quad (17)$$

$A_{i,k}$는 $1\le k\le t$는 $|A|$의 여인자(Cofactor)이다. $\sigma(x)$의 근이 "1"이라면 오류는 α^0에 대응되는 위치에서 발생한 것이다. 이를 식으로 표현하면 다음과 같다.

$$1+\sum_{k=1}^{t}\sigma_k=0 \quad (18)$$

식(18)에 식(17)을 대입하면 식(19)을 구할 수 있다.

$$\sum_{k=1}^{t}\sigma_k=\sum_{k=1}^{t}\left\{\frac{1}{|A|}\cdot\sum_{i=1}^{t}A_{i,k}S_{2i-1}\right\}=1 \quad (19)$$

식(19)을 다시 쓰면

$$\sum_{k=1}^{t}\sum_{i=1}^{t}A_{i,k}S_{2i-1}+|A|=0 \quad (20)$$

$$\begin{aligned}\sum_{k=1}^{t}\sum_{i=1}^{t}A_{i,k}S_{2i-1}+|A| &= |A|+(A_{1,1}+A_{1,2}+\cdots+A_{1,t})S_1\\ &\quad+(A_{2,1}+A_{2,2}+\cdots+A_{2,t})S_3+\cdots \quad (21)\\ &\quad+(A_{t,1}+A_{t,2}+\cdots+A_{t,t})S_{2t-1}\\ &=0\end{aligned}$$

다시 식(21)은 식(22)와 등가임을 보인다.

$$\Delta=\begin{vmatrix}1 & 1 & 1 & \cdots & 1\\ S_1 & 1 & 0 & \cdots & 0\\ S_3 & S_2 & S_1 & \cdots & 0\\ \vdots & & & & \\ S_{2t-1} & S_{2t-2} & \cdots & & S_{t-1}\end{vmatrix}$$

$$=\Delta_0+S_1\Delta_1+\Delta_3S_3\cdots S_{2t-1}\Delta_{2t-1} \quad (22)$$
$$=0$$

여기서,

$$\Delta_0=|A|$$

$$\Delta_1=\begin{vmatrix}1 & 1 & 1 & \cdots & 1\\ S_2 & S_1 & 1 & \cdots & 0\\ S_4 & S_3 & S_2 & \cdots & 0\\ \vdots & & & & \\ S_{2t-2} & S_{2t-3} & \cdots & & S_{t-1}\end{vmatrix}$$

$$=\begin{vmatrix}S_1 & 1 & 0 & \cdots & 0\\ S_3 & S_2 & S_1 & \cdots & 0\\ S_4 & S_3 & S_2 & \cdots & 0\\ \vdots & & & & \\ S_{2t-3} & S_{2t-4} & \cdots & & S_{t-1}\end{vmatrix}$$

$$+\begin{vmatrix}S_2 & 1 & 0 & \cdots & 0\\ S_4 & S_2 & S_1 & \cdots & 0\\ \vdots & & & & \\ S_{2t-2} & S_{2t-3} & S_{3t-5} & \cdots & S_{t-1}\end{vmatrix}$$

$$\cdots+\begin{vmatrix}S_2 & S_1 & 1 & \cdots & 0\\ S_4 & S_3 & S_2 & \cdots & 0\\ \vdots & & & & \\ S_{2t-2} & S_{2t-3} & \cdots & & S_{t-1}\end{vmatrix}$$

$$=A_{1,1}+A_{1,2}+\cdots+A_{1,t} \quad (23)$$

$$\Delta_3 = A_{2,1}+A_{2,2}+\cdots+A_{2,4}$$
$$\vdots$$
$$\Delta_{2t-1} = A_{t,1}+A_{t,2}+\cdots+A_{t,4}$$

Δ_3 를 식(22)에 대입하면 식(24)을 구할수 있다.

$$\begin{aligned}\Delta_3 &= |A| - S_1(A_{1,1}+A_{1,1}\cdots A_{1,3}) + S_3(A_{2,1}+A_{2,2}\cdots A_{2,3}) \\ &\quad \cdots + S_{2t-1}(A_{t,1}+A_{t,2}\cdots A_{t,4}) \qquad (24) \\ &= 0\end{aligned}$$

식(24)과 식(21)은 식의 형태가 등가임을 알수 있다. 따라서 식(18)과 식(23)은 완전히 등가이다. 식(18)을 이용하여 오류 위치를 찾는 것은 식(24)을 이용하여 오류 위치를 찾는 것과 동일하다. 그러므로 본 논문에서는 식(24)를 이용하여 오류위치를 찾는 복호 알고리즘을 제시하였다.

4. 5중 오류정정 (255, 215) BCH 부호기 및 복호기 설계

4-1. (255, 215)BCH 부호기

부호기는 생성다항식을 구하면 설계할 수 있다. 생성다항식은 다음과 같이 구해진다.

원시다항식 $P(x) = 1+x^2+x^3+x^4+x^8$이고,최소다항식 $m_i(x)$는 다음과 같다.

① $m_1(x) = (x+\alpha)(x+\alpha^2)(x+\alpha^4)(x+\alpha^8)(x+\alpha^{16})(x+\alpha^{32})(x+\alpha^{64})(x+\alpha^{128})$
$= 1+x^2+x^3+x^4+x^8$

② $m_3(x) = (x+\alpha^3)(x+\alpha^6)(x+\alpha^{12})(x+\alpha^{24})(x+\alpha^{48})(x+\alpha^{96})(x+\alpha^{192})(x+\alpha^{129})$
$= 1+x+x^2+x^4+x^5+x^6+x^8$

③ $m_5(x) = (x+\alpha^5)(x+\alpha^{10})(x+\alpha^{20})(x+\alpha^{40})(x+\alpha^{80})(x+\alpha^{160})(x+\alpha^{165})(x+\alpha^{130})$
$= 1+x^4+x^5+x^6+x^7+x^8$

④ $m_7(x) = (x+\alpha^7)(x+\alpha^{14})(x+\alpha^{28})(x+\alpha^{56})(x+\alpha^{112})(x+\alpha^{224})(x+\alpha^{193})(x+\alpha^{131})$
$= 1+x^3+x^5+x^6+x^8$

⑤ $m_9(x) = (x+\alpha^9)(x+\alpha^{18})(x+\alpha^{36})(x+\alpha^{72})(x+\alpha^{144})(x+\alpha^{33})(x+\alpha^{66})(x+\alpha^{132})$
$= 1+x^2+x^3+x^5+x^7+x^8$

$\therefore\ g(x) = LCM\{m_1(x), m_3(x), m_5(x), m_7(x),$
$= (1+x^2+x^3+x^4+x^5+x^8)(1+x^2+x^3+x^5+x^6+x^8)$
$(1+x^1+x^4+x^5+x^6+x^7+x^8)(1+x^3+x^5+x^6+x^8)$
$(1+x^2+x^3+x^4+x^5+x^7+x^8)$
$= 1+x^4+x^8+x^{10}+x^{11}+x^{13}+x^{15}+x^{16}+x^{17}+x^{20}+x^{22}$
$+x^{23}+x^{24}+x^{26}+x^{27}+x^{28}+x^{29}+x^{30}+x^{32}+x^{33}+x^{36}$
$+x^{37}+x^{40}$

상기 식에 기초하여 설계된 (255, 215) BCH부호의 부호기는 그림1과 같다.

4-2. 오증요소

오증요소는 식(25)와 같다.

$$\begin{aligned}S_i &= r(\alpha^i) \\ &= r_0 + r_1\alpha^i + \cdots + r_{n-1}\alpha^{(n-1)i} \qquad (25) \\ &= (\cdots(((r_{n-1}\alpha^i + r_{n-2})\alpha^i + r_{n-3})\alpha^i + r_{n-4})\alpha^i + \cdots + r_1)\alpha^i + r_0)\end{aligned}$$

식 (25)는 다음과 같은 수식에 바탕을 두고 설계할 수 있다.

$GF(2^8)$상의 임의의 원소 β를 $\beta = \beta_0+\beta_1\alpha+\beta_2\alpha^2+\beta_3\alpha^3+\beta_4\alpha^4+\beta_5\alpha^5+\beta_6\alpha^6+\beta_7\alpha^7$ ($\beta_i \in \{0, 1\},\ 0 \le i \le 7$)이라면

①
$$\begin{aligned}\alpha\beta &= \alpha(\beta_0+\beta_1\alpha+\beta_2\alpha^2+\beta_3\alpha^3+\beta_4\alpha^4+\beta_5\alpha^5+\beta_6\alpha^6+\beta_7\alpha^7) \\ &= \beta_7+\alpha\beta_0+(\beta_1+\beta_7)\alpha^2+(\beta_2+\beta_7)\alpha^3+(\beta_3+\beta_7)\alpha^4+\beta_4\alpha^5 \\ &\quad +\beta_5\alpha^6+\beta_6\alpha^7\end{aligned}$$

②

$$\alpha^3\beta = \alpha^3(\beta_0+\beta_1\alpha+\beta_2\alpha^2+\beta_3\alpha^3+\beta_4\alpha^4+\beta_5\alpha^5+\beta_6\alpha^6+\beta_7\alpha^7)$$
$$= \beta_5+\beta_6\alpha+(\beta_5+\beta_7)\alpha^2+(\beta_0+\beta_5+\beta_6)\alpha^3+(\beta_1+\beta_5+\beta_6+\beta_7)\alpha^4$$
$$+(\beta_2+\beta_6+\beta_7)\alpha^5+(\beta_3+\beta_7)\alpha^6+\beta_4\alpha^7$$

③

$$\alpha^5\beta = \alpha^5(\beta_0+\beta_1\alpha+\beta_2\alpha^2+\beta_3\alpha^3+\beta_4\alpha^4+\beta_5\alpha^5+\beta_6\alpha^6+\beta_7\alpha^7)$$
$$= (\beta_3+\beta_7)+\beta_4\alpha+(\beta_3+\beta_5+\beta_7)\alpha^2+(\beta_3+\beta_4+\beta_6+\beta_7)\alpha^3$$
$$+(\beta_3+\beta_4+\beta_5)\alpha^4+(\beta_0+\beta_4+\beta_5+\beta_6)\alpha^5+(\beta_1+\beta_5+\beta_6+\beta_7)\alpha^6$$
$$+(\beta_2+\beta_6+\beta_7)\alpha^7$$

④

$$\alpha^7\beta = \alpha^7(\beta_0+\beta_1\alpha+\beta_2\alpha^2+\beta_3\alpha^3+\beta_4\alpha^4+\beta_5\alpha^5+\beta_6\alpha^6+\beta_7\alpha^7)$$
$$= (\beta_1+\beta_5+\beta_6+\beta_7)+(\beta_2+\beta_6+\beta_7)\alpha+(\beta_1+\beta_3+\beta_5+\beta_6)\alpha^2$$
$$+(\beta_1+\beta_2+\beta_4+\beta_5)\alpha^3+(\beta_1+\beta_2+\beta_3+\beta_7)\alpha^4$$
$$+(\beta_2+\beta_3+\beta_4)\alpha^5+(\beta_3+\beta_4+\beta_5)\alpha^6+(\beta_0+\beta_4+\beta_5+\beta_6)\alpha^7$$

$$\alpha^9\beta = \alpha^9(\beta_0+\beta_1\alpha+\beta_2\alpha^2+\beta_3\alpha^3+\beta_4\alpha^4+\beta_5\alpha^5+\beta_6\alpha^6+\beta_7\alpha^7)$$
$$= (\beta_3+\beta_4+\beta_5)+(\beta_0+\beta_4+\beta_5+\beta_6)\alpha+(\beta_1+\beta_3+\beta_4+\beta_6+\beta_7)\alpha^2$$
$$+(\beta_0+\beta_2+\beta_3+\beta_7)\alpha^3+(\beta_0+\beta_1+\beta_5)\alpha^4+(\beta_0+\beta_1+\beta_2+\beta_6)\alpha^5$$
$$+(\beta_1+\beta_2+\beta_3+\beta_7)\alpha^6+(\beta_2+\beta_3+\beta_4)\alpha^7$$

(255, 215) BCH 오증 요소 계산회로는 그림2와 같다.

4-3. 오증요소에서 계산된 A값 및 Δ값을 이용한 오류 판단

오류의 갯수를 판별하기 위한 A와 오류위치 판별하기 위한 Δ를 유도하면 다음과 같다. 오류의 갯수를 판단하기위한 A_3, A_5은 다음과 같다.

$$A_3 = \begin{bmatrix} 1 & 0 & 0 \\ S_2 & S_1 & 1 \\ S_4 & S_3 & S_2 \end{bmatrix} = S_1^3 + S_3$$

$$A_5 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ S_2 & S_1 & 1 & 0 & 0 \\ S_4 & S_3 & S_2 & S_1 & 1 \\ S_6 & S_5 & S_4 & S_3 & S_2 \\ S_8 & S_7 & S_6 & S_5 & S_4 \end{bmatrix} = S_1(S_3^3+S_1^9)$$
$$+S_3(S_1^7+S_7+S_1^2S_7)+S_5S_1^5+S_1^3S_7+S_5^2$$

오류값을 판별하기위한 Δ_1, Δ_3, Δ_5는 다음과 같다.

$$\Delta_1 = S+1$$

$$\Delta_3 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ S_1 & 1 & 0 & 0 \\ S_3 & S_2 & S_1 & 1 \\ S_5 & S_4 & S_3 & S_2 \end{bmatrix}$$

$$\Delta_3 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ S_1 & 1 & 0 & 0 & 0 & 0 \\ S_3 & S_2 & S_1 & 1 & 0 & 0 \\ S_5 & S_4 & S_3 & S_2 & S_1 & 1 \\ S_7 & S_6 & S_5 & S_4 & S_3 & S_2 \\ S_9 & S_8 & S_7 & S_6 & S_5 & S_4 \end{bmatrix}$$

$$= S_1(S_3^3+S_1^9+S_1^{10}+S_1^{12}+S_1^{13}+S_1^{14}+S_7^2+S_5^2)$$
$$+S_3(S_1^7+S_1^8+S_1^{12}+S_5^2+S_7+S_9)$$
$$+S_5(S_1^7+S_1^5+S_7+S_9+S_3^3+S_1^6)$$
$$+S_7(S_1^4+S_1^8+S_3^2+S_1^3+S_1^5)$$
$$+S_9(S_1^4+S_1^6+S_1^3+S_3^2)$$
$$+S_1(S_3^2S_5+S_3S_7+S_3S_9+S_3^2S_5+S_3S_9+S_3S_7)$$
$$+S_3(S_1^3S_5+S_1^5S_7+S_1^1S_9+S_1^7S_5+S_1^5S_5+S_1^6S_5+S_1^2S_5^2+S_1^4S_5+S_1^2S_9$$
$$+S_1^4S_7+S_1^2S_5)$$
$$+S_5^2(S_1^9+S_1^8+S_1^6)$$
$$+S_5^2(1+S_1^2+S_1^4+S_1^5)$$
$$+S_3^3(S_1^2+S_1^4+S_1^6)$$
$$+S_3^4+S_3^5+S_7^2+S_5^3+S_1^3S_1^2S_7$$

$A_3 = A_5 = 0$인 경우, 오류가 한 개 발생한 것으로 판단하고 그때의 오류위치는 $\Delta_1 = 0$이 될 때이고, $A_3 \neq 0$, $A_5 = 0$인 경우, 오류가 2, 3개 발생한 것으로 판단하고 그때의 오류위치는 $\Delta_3 = 0$이 될 때이고 $A_5 \neq 0$인 경우, 오류가 4, 5개 발생한 것으로 판단하고 그때의 오류 위치는 $\Delta_5 = 0$일때로 판별한다. 본 논문에서 사용한 효율적인 복호 알고리즘은 일반 복호과정에서 가장 어려운 오류위치다항식을 구한 후 오류위치를 판단하는 오류 위치 계산 과정을 거치

지 않고 수신 계열이 복호기에 입력된 후 이미 계산된 오증 요소들을 이용하여 오류의 갯수를 판단하고 수신계열의 오류가 난 위치에 오류값을 가산하여 오류정정을 수행하는 알고리즘이다. Peterson의 복호 알고리즘을 이용한 복호기는 식(26)에 근거하여 계산된 결과는 표 1과 같다.

$$\begin{bmatrix} S_1 & S_2 & S_3 & \cdots & S_{v-1} & S_v \\ S_2 & S_3 & S_4 & \cdots & S_v & S_{v+1} \\ & & & & & \\ S_v & S_{v+1} & S_{v+2} & \cdots & S_{2v-2} & S_{2v-1} \end{bmatrix} \begin{bmatrix} \sigma_v \\ \sigma_{v-1} \\ \\ \sigma_1 \end{bmatrix} \quad (26)$$

5중오류정정 (255, 215) BCH 부호의 복호기를 Peterson 알고리즘을 이용한 복호기의 소자수를 비교하면 표2와 같다.

5. 5중오류정정(255,215) BCH 부호의 오류정정 알고리즘의 C 및 VHDL 시뮬레이션

본장에서는 오류정정 능력이 5인 (255, 215) BCH 부호의 부호기 및 복호기를 C언어로 시뮬레이션한 결과를 보인다.

시뮬레이션을 위한 복호 알고리즘에서 제시된 오류정정 능력이 5인 (255, 215) BCH 부호의 부호화 및 복호 알고리즘을 이용하였다.

(255, 215) BCH 부호의 부호화 및 복호루틴은 부호기 시뮬레이션 루틴과 복호기 시뮬레이션 루틴으로 구성된다.

부호기 프로그램은 enc256.c이며 복호기 프로그램은 main 루틴에서 13개의 외부함수를 호출하며 실현하였다. main루틴에서 사용된 외부함수는 다음과 같다.

표1. Peterson의 복호 알고리즘을 이용한 복호기의 복잡도

비교항목 / 오류의 갯수	곱셈기	X-OR개수	역원계산기 수
t=1	0	0	1
t=2	6	3	1
t=3	44	18	1
t=4	260	96	1
t=5	1868	555	1
합	2178	672	5

표2. 5중(255, 215) BCH 복호기의 복잡도 비교

종류 / 비교항목	Peterson의 복호알고리즘	효율적인 복호알고리즘
곱셈기수	2178	73
X-OR 수	672	80
역원계산기수	5	없음

```
extern void SyndromeInit(int *, int *, int *, int *, int *);
extern void SyndromeRtn(int *,int *, int *, int *, int *,
int *);
extern int  QueueRead();
extern int  QueueWrite(int);
extern int  Delta1(int *);
extern int  Delta3(int *, int *, int *);
extern int  Delta5(int *, int *, int *, int *, int *);
extern int     A3(int *, int *);
extern int     A5(int *, int *, int *, int *);
extern void mult(int *, int *, int *);
extern void xor(int *, int *, int *);
extern void move(int *, int *);
extern int or(int *);
```

(255,215) BCH 부호의 시뮬레이션을 위한 외부함수 기능은 다음과 같다. SyndromeInit (int*, int*, int*, int*, int*, int*)은 오증요소를 초기화하는 프로그램이다.

SyndromeRtn(int*, int*, int*, int*, int*, int*, int*)은 수신된 부호 벡터로부터 오증요소를 구하는 프로그램이다.

Queue Read():부호화된 부호어을 읽는 프로그램이다.

QueueWrite(int):부호화된 부호어를 써넣는 프로그램이다.

Delta1(int *):△1 값을 구하는 프로그램이다.

Delta3(int *, int *, int *):△3 값을 구하는 프로그램이다.

Delta5(int *, int *, int *, int *, int *):△5 값을 구하는 프로그램이다.

A3(int *, int *):A3 값을 구하는 프로그램이다.

A5(int *, int *, int *, int *):A5 값을 구하는 프로그램이다.

mult(int *, int *, int *):원소 B와 C를 곱하여 A에 저장하는 프로그램이다.

xor(int *, int *, int *):B C하여 A에 저장하는 프로그램이다.

move(int *, int *):B의 내용을 A로 옮기는 프로그램이다.

or(int *):A의 내용이 하나라도 "1"이면 출력은 "1"이되고, 모두 "0"이면 출력은 "0"인 프로그램이다.

프로그램 시뮬레이션 결과 복호 알고리즘이 정상적으로 동작함을 확인하였다. 이 결과는 VHDL 로 부호한후 이를 논리레벨 시뮬레이션하고 back annotated 타이밍 레벨 시뮬레이션시 활용되었다.

본 소프트웨어 프로그램은 BCH 부호를 소프트웨어로 실현하는 경우 적극적으로 활용될 수 있다. 제안된 복호알고리즘을 VHDL로 실현하였다. VHDL 시뮬레이션 결과는 그림 4와 같다. 그림4의 rx는 수신데이타 s1, s3, s5, s7, s9는 오증 요소를 의미하고 복호된 결과는 c(x)이다. 오류는 그림 4 의 처음부분에 5개 발생했다고 가정하고, c(x)에 오류가 복구된 것을 확인하였다.

6. 결 론

BCH 부호의 복호 과정 중에서 오증으로부터 오류위치 다항식 $\sigma(x)$의 계수 σ_i를 결정하고 $\sigma(x)$의 근을 구하여 오류 위치를 찾아서 정정하는 과정은 매우 복잡하다. 본 논문에서는 이와 같은 문제를 해결할 수 있는 직접 복호 알고리즘을 제안한다.

본 논문에서는 오증으로부터 직접 오류의 위치를 찾아서 오류를 정정할 수 있는 효율적인 복호법을 이용하여 Peterson의 복호 알고리즘보다 곱셈기 및 EX-OR 게이트 수를 현저히 감소시키며 역원계산기가 필요없는 (255, 215) BCH 부호의 부호기 및 복호기를 설계하

였다. 그 결과 곱셈기는 1752개, EX-OR 게이트는 599개, 역원기는 5개가 줄었음을 확인하였고 복호알고리즘의 타당성을 VHDL로 검증하였다. 앞으로 이의 ASIC 또는 FPGA 실현에 관한 연구를 수행할 예정이다.

참 고 문 헌

[1] Rhee, M. Y., Error correcting coding Theory, McGraw-Hill, New York, 1989.

[2] 이 만 영 , BCH 부호와 Reed-Soloman 부호, 민음사, 1990.

[3] Berlekamp, E.R. , "On Decoding binary Base-Chaudhuri-Hocquenghem codes," IEEE Trans. on Inform. Theory, IT-11, pp.580-585, 1965.

[4] Polkinghorn, F. , "Decoding of double and Triple Error Correcting Base-Chaudhuri-Hocquenghem code. " IEEE Trans. Inf. Theory, vol. IT-12, pp. 480-481, 1966.

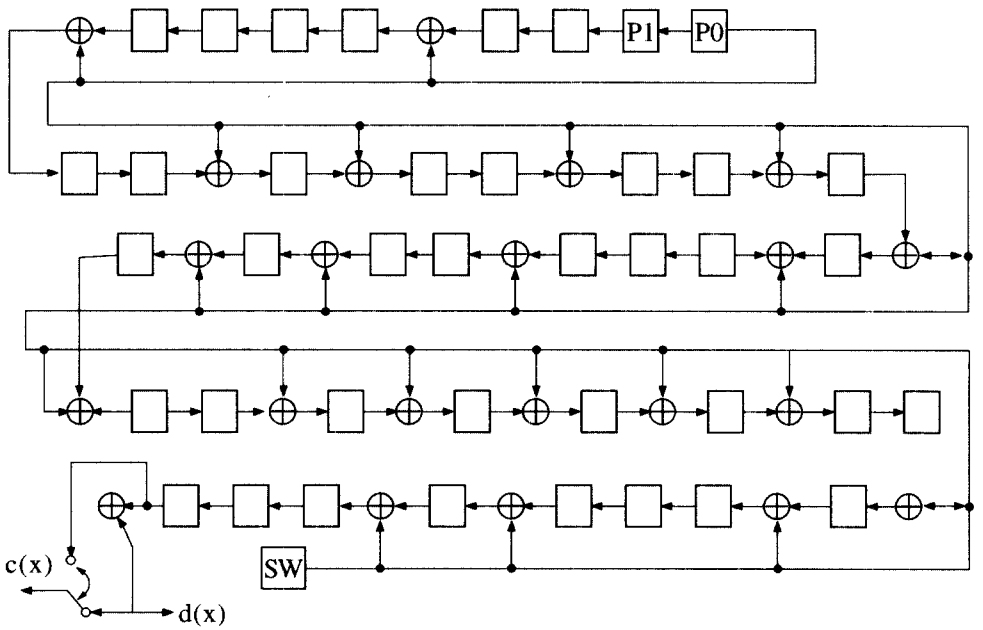


그림1. (255,215) BCH 부호기

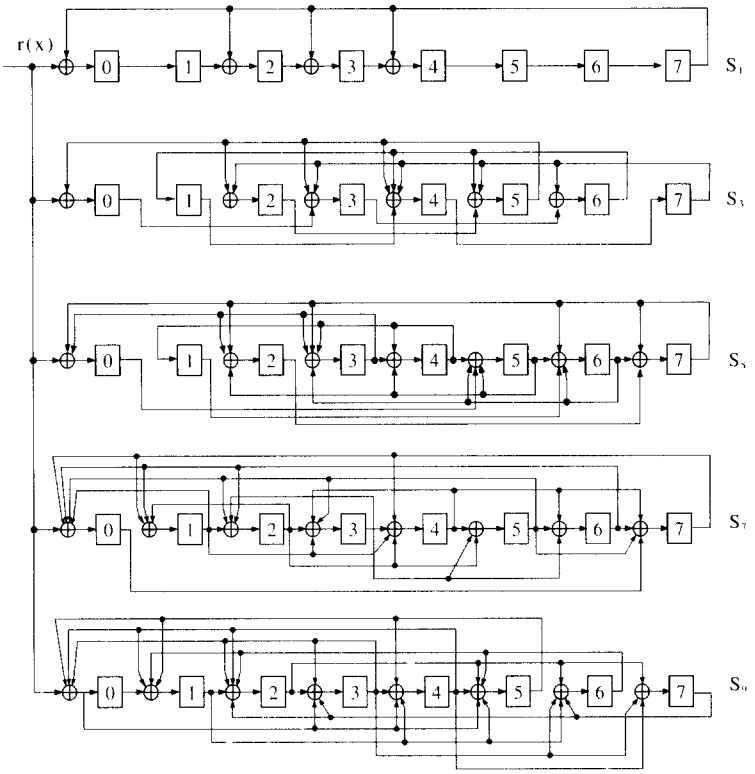


그림 2. (255,215) BCH 부호에 대한 오중요소 계산회로

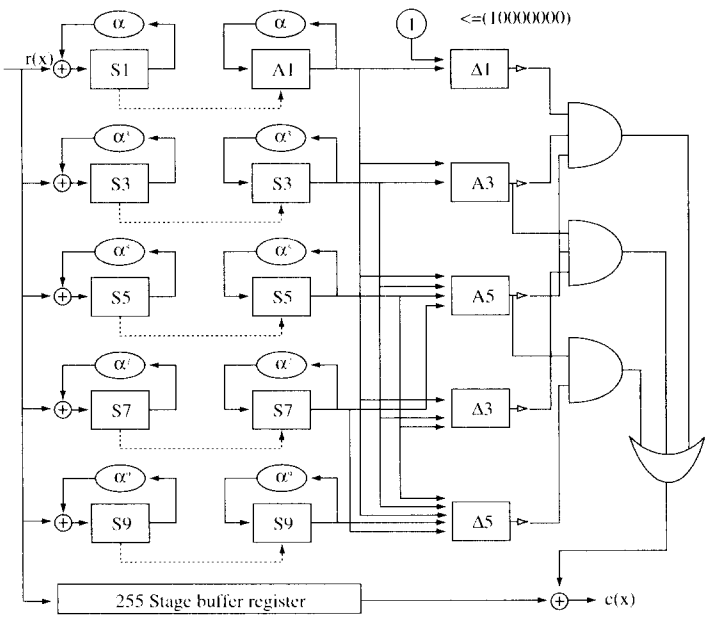


그림 3. (255, 215) BCH 복호기

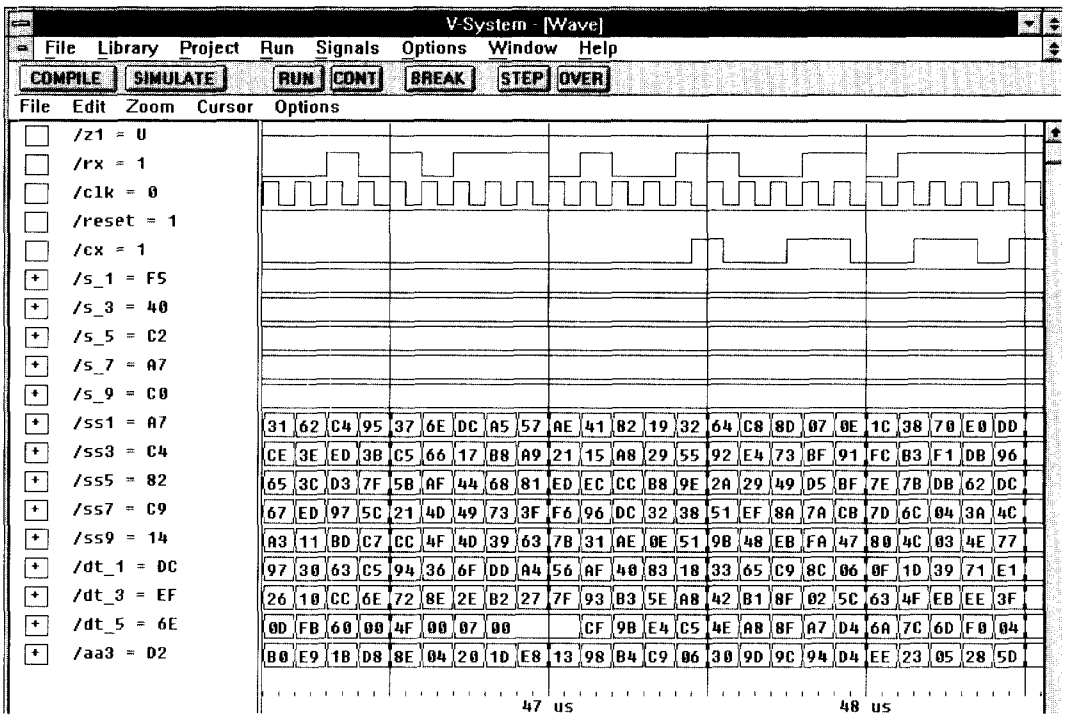
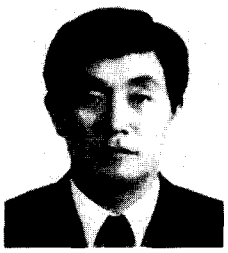


그림 4. 5중 오류정정 VHDL 시뮬레이션 결과

□ 著者紹介



朴 鎭 秀(Jin Soo Park) 회원

1948년 8월 30일생
1975년 2월 : 한양대학교 전자공학과(학사)
1977년 2월 : 한양대학교 대학원 전자통신공학과(석사)
1985년 2월 : 한양대학교 대학원 전자통신공학과(박사)
1987년 2월 ~ 1988년 2월 : Univ. Colorado at Colorado Spring(Post Doc.)
1978년 2월 ~ 현재 : 청주대학교 전자공학과 교수

※ 주관심분야 : 디지탈 이동통신, 부호이론, Spread Spectrum 통신, 무선 LAN등임.

姜 慶 植(Kyung Sik Kang) 회원

1961년 10월 25일생
1983년 2월 : 청주대학교 전자공학과(학사)
1989년 2월 : 한양대학교 대학원 전자통신공학과(석사)
1994년 3월 ~ 현재 : 청주대학교 대학원 전자공학과 박사과정
1993년 3월 ~ 현재 : 주성전문대 전자과 조교수

※ 주관심분야 : 부호이론, 디지탈 이동통신 등임.