1999년도 대한전기학회 하계학술대회 논문집 1999. 7.19-21

디지털 TV용 Reed-Solomon 복호기의 구현

박찬일, 김종태
성균관대학교 전기전자 및 컴퓨터 공학부 시스템 합성 연구실

Reed-Solomon Decoder using Berlekamp-Massey Algorithm for Digital TV

Chan-Il Park, Jong-Tae Kim
School of Electrical&Computer Engineering, SungKyunKwan University

Abstract - RS(Reed-Solomon)부호는 오류 정정을 위한 채널 코딩기법중의 하나로 특히 연집 오류에 대해 강한 특성을 갖고 있으며, CD-P(Compact Disc Player), DAT(Digital Audio Tape), VTR, DVD(Digital Video Disc), 디지탈 TV 디코더등에서 사용되고 있다. 본 논문은 Galois Field, GF[2^8]상에서 (204, 188, 8)의 규격을 갖는 디지탈 TV용 RS 복호기의 구현에 관한 연구로 8개의 심볼 오류까지 정정 가능하다. 오증 계산은 16개의 오증 계산셀로 구성되어지며, 오류 위치 다항식을 계산하는데 있어서는 Berlekamp-Massey 알고리즘을 사용한다. VHDL로 설계되어 Synopsys를 이용하여 검증 및 합성하였다.

1. 서 론

디지털 통신계의 경우 오류의 형태가 크게 Random 오류와 연집 오류로 분류되는데, RS 부호의 경우 한 부분의 대량의 정보가 한꺼번에 오류가 되는 집중적, 대량적 오류의 형태를 갖는 연집 오류에 강한 특성을 갖고 있다. 이런 특성이 RS 부호가 각광받는 이유이며, 이는 RS 부호가 블럭 부호로써 정보를 하나의 bit단위가 아닌 부호 Word단위로 처리를 하는데 기인한다. 또한 RS 부호는 그 부호과정과 복호 과정에서 통상적인 4칙 연산과는 다른 Galois Field내에서의 다항식 연산과 심볼 단위의 연산으로 이루어진다.

본 연구에서의 RS 부호의 경우 GF[2^8]를 사용하는데 이때의 Galois Field와 원시 다항식은 다음과 같다[1].

$GF[2^8] = \{ 0, \alpha^0, \alpha^1, \ldots\ldots\ldots, \alpha^{254} \}$

$P(x) = X^8+X^4+X^3+X^2+1$

일반적인 RS 부호는 (n, k, t)로 표현된다. 여기서

n : 부호 Word의 크기
k : 정보 Word의 크기
t : 오류 정정 능력 이다.

부호의 규격을 결정하는 최소거리와 정정 능력은 다음과 같다.

최소거리 : $d_{\min} = (parity+1) = (n-k+1)$

정정 능력 : $t \le \frac{d_{\min}-1}{2} \left(= \frac{parity수}{2} = \frac{n-k}{2}\right)$

2.1장에서는 RS 부호의 부호화 과정을 설명하고, 2.2에서는 복호 과정인 오증 계산, Berlekamp-Massey 고리즘을 사용한 오류 위치 다항식의 계산, 오류값 계산 과정과 각각의 블럭도를 설명하였다.

2. 본 론

2.1 RS 부호의 부호화

부호기는 정보원으로부터 정보를 받아서 규정된 형태의 부호 word를 생성하는 연산회로이며, parity는 parity 생성 다항식(Generator Polynomial)에 의해 구해진다.

2.1.1. 생성 다항식(Generator Polynomial)

생성 다항식 g(x)는 부호화 과정에서 Parity를 생성하며, 복호 과정에서 수신된 정보(Received Word)에 오류가 발생하였는가를 판단하고 오류를 정정하는데 유일한 단서로 제공되는 식으로 다음과 같이 정의된다.

$$g(x) = \prod_{i=0}^{2t-1}(x+\alpha_i)$$

본 연구의 경우, 정정 능력 t는 8이므로, 생성 다항식은 17개의 계수를 갖는 16차 다항식이 된다.

2.1.2. 부호화의 원리

RS부호의 부호 Word다항식은 다음과 같이 표현된다.

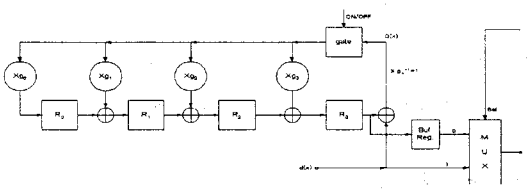
$$C(x) = C_{n-1}X^{n-1}+C_{n-2}X^{n-2}+\cdots\cdots C_2X^2+C_1X^1+C_0X^0$$
$$= d(x)\cdot X^{n-k}+p(x)$$

위의 c(x)를 생성 다항식 g(x)가 결합된 형태로 표현하면 다음과 같으며, 이 식은 위의 $d(x)\cdot X^{n-k}+p(x)$식과 항등식으로 정의된다[2].

$$C(x) = g(x)\cdot Q(x)$$

부호 Word 다항식 C(x)가 g(x)와 Q(x)의 곱셈 결과로 정의되었다는 것에서 "g(x)의 모든 근은 C(x)의 근이 된다."는 것이 복호 과정에서의 유일한 단서가 된다.

부호화 회로는 X^{n-k}을 곱하는 곱셈회로와 g(x)로 나누는 나눗셈회로로 구성되어지며 그 정정 능력이 2인 부호기의 블럭도는 〈그림 1〉과 같다.



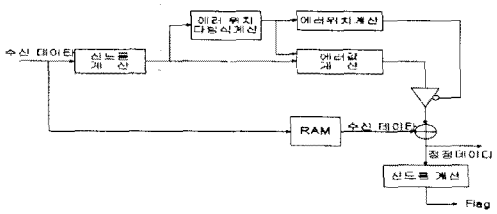
〈그림 1〉 t=2인 RS 부호기

〈그림 1〉 Register들의 초기값은 '0'이며 정보원으로부터의 정보 d(x)의 최저차항 심볼인 d_0가 입력된 직후 Parity 정보 p(x)가 Register에 남게 된다. 본 연구에서는 정정 능력 t가 8인 부호기를 설계하여 이를 통해 생성된 부호 Word를 가지고 RS 복호기를 검증하였다. t가 8인 부호기의 경우 g(x)의 계수값은 16개이며 따라서 사용되는 곱셈기와 Register도 16개로 늘어난다.

2.2 RS 부호의 복호화

일반적인 RS 부호의 복호 과정과 복호기의 블록도는 다음과 같다[3].

1. 오증 계산
2. 오류 위치 다항식 $\sigma(X)$의 계산
3. 오류 위치 판별
4. 오류값 계산
5. 정정 및 검증



〈그림 2〉 RS 부호기의 블럭도

2.2.1. 오증 계산

Received 다항식 R(x)는 원래의 부호 Word 다항식에 오류가 더해진 값으로 볼 수 있다.

$$R(x)=c(x)+e(x)$$

즉, 부호기에서의 생성 다항식의 근이 부호 Word의 근이라는 것을 감안할 때 위 식에 x대신 근 $\alpha^0, \alpha^1, \alpha^2, \ldots, \alpha^{n-k-1}$을 대입해 보면 $c(x)$의 성분은 모두 0이 되므로 아래와 같이 전개된다.

$$R(\alpha^j)=e(\alpha^j) \qquad (0\le j\le n-k-1)$$

만약 오류가 발생하지 않았다면 $e(x)=0$이 될 것이므로 $R(\alpha^j)=0$이 된다. 그러므로 $R(x)$에 오류가 발생하였는가의 판단은 생성 다항식의 모든 근을 $R(x)$에 대입한 값들, 오증들이 모두 0이 되는가를 보면 된다. 즉, 오증값이란 Received Word 다항식 $R(x)$에 생성 다항식의 근을 대입한 값으로 오류 발생 여부의 판별과 오류 위치 다항식 계산, 오류값 계산의 근거를 제공하는 중요한 값이다. 일반적으로 $c(n,k)$ RS 부호의 경우 $c(\alpha^j)=0$ $(0\le j\le n-k-1)$이므로 그 오증은 다음과 같이 정의된다[4].

$$S_j=R(\alpha^j)=e(\alpha^j) \qquad (0\le j\le n-k-1)$$

$$S_j=e(\alpha^j)=\sum_{i=1}^{n-1} e_i(\alpha^j)^i \qquad (0\le j\le n-k-1)$$

n개의 Received Word로 위의 식을 전개해보면,

$S_j=R(\alpha^j)$

$=r_{n-1}(\alpha^j)^{n-1}+r_{n-2}(\alpha^j)^{n-2}+\ldots.+r_1(\alpha^j)^1+r_0(\alpha^j)^0=$

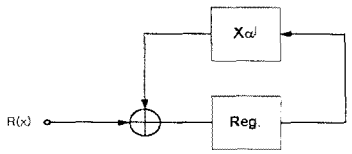
$((..((r_{n-1}*\alpha^j+r_{n-2})*\alpha^j+r_{n-2})*\alpha^j+..)*\alpha^j+r_1)*\alpha^j+r_0$

본 연구에서의 오증은 다음과 같이 계산된다.

$$S_j=R(\alpha^j)=e(\alpha^j) \qquad (0\le j\le 15)$$

$$S_j=e(\alpha^j)=\sum_{i=1}^{n-1} e_i(\alpha^j)^i \qquad (0\le j\le 15)$$

즉, 오증의 개수는 생성 다항식의 근의 개수와 같은 16개가 계산되며, 앞의 식의 특징을 살려 오증값을 계산하는 회로는 다음과 같다.



〈그림 3〉 오증 계산 회로

〈그림 3〉에서 Register의 초기값이 0인 상태에서 $R(x)$가 r_{n-1} 심볼로부터 입력되어 r_0까지 입력된 직후 Register의 값이 오증 S_j가 된다. α^j는 생성 다항식의 근을 의미하며, 본 연구에서의 오증 회로는 16개의 위와 같은 오증 계산셀로 구성된다.

2.2.2. 오류 위치 다항식 $\sigma(x)$의 계산

오류의 위치를 알기 위해서 오증을 이용하여 오류 위치 다항식을 구한다. 즉 오류 위치 다항식은 오류의 위치를 근으로 하는 다항식으로 다음과 같은 식으로 표현 가능하다.

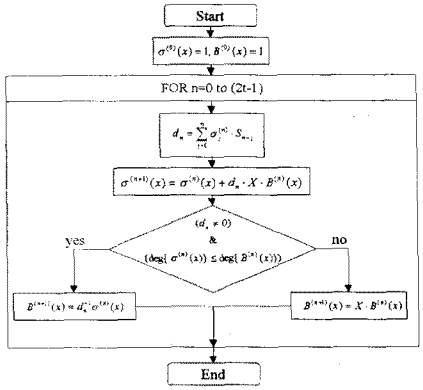
$$\sigma(x)=(x+X_1)(x+X_2)\ldots\ldots(x+X_v) \qquad (v\le t)$$
$$=\sigma_{0x}^{v}+\sigma_1 x^{v-1}+\sigma_2 x^{v-2}+\ldots.+\sigma_{v-1}x+\sigma_v$$

여기서 x_i는 오류가 발생한 위치이다. 그러므로 오류 위치 x_i를 x에 대입하면 그 결과는 항상 0이 된다. 이를 이용하여 다음 블록인 오류 위치 판별에서 오류의 위치를 찾게 되는 것이다. 그런데 오증의 정의에서 S_j는 모든 오류 심볼에 대해 Y_i(오류 값)$\times X_i^j$(오류 위치)와 같으므로 다음과 같은 Key 방정식을 얻을 수 있다[2].

$$S_j\sigma_v+S_{j+1}\sigma_{v-1}+\cdots\cdots+S_{j-v-1}\sigma_1+S_{j+v}=0$$

Key 방정식으로부터 오류 위치 다항식을 구하는 대표적인 알고리즘으로는 Berlekamp-Massey 알고리즘과 정 Euclid 알고리즘이 있으며 본 연구에서의 경우에는 Berlekamp-Massey 알고리즘으로 구현되었다.

Berlekamp-Messay 알고리즘은 σ_i를 찾는데 있어서, Key 방정식에 S_n까지 대입하였을때 식을 만족하는 $\sigma(x)$의 계수들을 찾아 이 계수들이 S_{n+1}, S_{n+2}, S_{n+3} 순으로 식이 만족되는 가를 살피는 과정을 수행한다. 만약 S_n까지는 만족하였지만 S_{n+1}에 대해서는 Key 방정식을 만족하지 않았다면 새로운 $\sigma(x)$를 찾아야 하는데 이와 같이 S_n까지 Key 방정식을 만족하는 중간 과정의 오류위치 다항식을 $\sigma^{(n)}(x)$, n번째 단계에서의 Minimal Solution이라 하며, $\sigma^{(n)}(x)$에 S_{n-1}까지를 대입시켜 본 값을 d_n, Discrepancy라 한다. 즉, Berlekamp-Massey 알고리즘은 0번째 단계에서부터 (2t-1)까지의 Minimal Solution을 구함으로써 오류 위치 다항식의 계수를 구하는 알고리즘이며 〈그림 4〉과 같은 단계를 거친다[2][4].



〈그림 4〉 Berlekamp - Massey 알고리즘

본 연구의 경우 정정 능력이 8이므로 오류 위치 다항식은 최대 8차 다항식으로 계산된다.

2.2.3. 오류 위치의 판별

오류 위치를 판별하기 위해서 오류 위치 다항식, $\sigma(x)$의 근이 오류가 발생한 위치임을 이용한다. v개의 오류에 대해 $\sigma(x)$는 아래와 같다.

$$\sigma(X)=\sigma_0X^v+\sigma_1X^{v-1}+\cdots\cdots+\sigma_{v-1}X^1+\sigma_vX^0$$

여기에 오류 위치 X_i를 대입하면 $\sigma(X_i)=0$이다.

위의 식을 변형하여 구현하면,

$$\sigma_0+\sigma_1X_i^{-1}+\sigma_2X_i^{-2}+\cdots\cdots+\sigma_{v-1}X_i^{-(v-1)}+\sigma_vX_i^{-v}=0$$

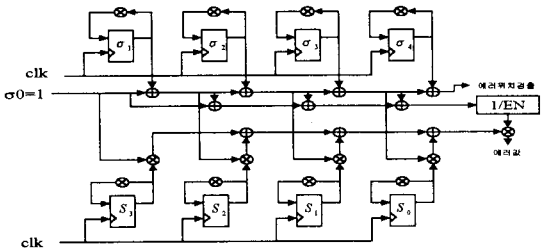
$X_i=\alpha^k$일 때 k의 범위는 $0\le k\le n-1$ 이므로 $\alpha^{(n-1)}$에서부터 α^0을 위 식에 대입하여 0이 되는 위치를 오류 위치로 결정한다.

2.2.4. 오류값 계산

오류의 크기 계산은 오증값과 오류 위치 다항식으로 구해지며, 오류의 크기 Y_j는 다음과 같다.

$$Y_j = \frac{\sum_{j=0}^{v-1} \sigma_{ji} S_{(v-1)-i}}{\sum_{j=0}^{v-1} \sigma_{ji} X_j^{(v-1)-i}}$$

위 식을 이용하여 오류 값을 계산하는 회로를 구성하면 〈그림 7〉과 같다. 이 연산회로는 오류 위치를 계산하는 회로를 포함한다. 이는 위 식에서의 분모항이 오류 위치를 판별하는 $\sigma(\alpha^i)$를 계산하는 회로를 포함하기 때문이다.



〈그림 5〉 t=2 RS 부호기의 오류 위치 계산 및 오류값 계산 회로

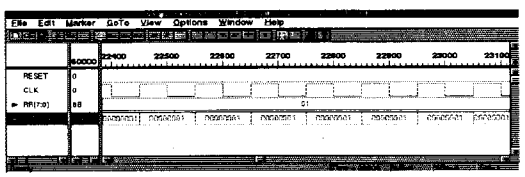
〈그림 5〉회로의 초기 조건으로는 Register σ에 오류 위치 다항식 계수 σ_i가 입력되고 Register S에는 오증 S_i가 입력된다. 첫 번째 오증이 입력된 직후의 값, 즉 j=0 일때가 α^0위치의 오류 Y_0이며 순차적으로 Register에 clock을 인가하면 Y_i의 값이 계산된다. 본 연구의 경우 위치 다항식 계수 σ_i 와 오증 S_i는 16개이지만 실제적으로 정정되는 오류는 최대 8개이므로 오류값을 계산하는데 σ_i는 8개만이 사용된다.

2.2.4. 오류의 정정

오류의 위치가 판별되고 오류값이 계산되면 해당하는 위치의 심볼 정보에 오류값을 더해줌으로써 오류의 정정이 이루어진다. 연산은 Galois Field에서의 덧셈 연산, 즉 XOR 연산으로 구현된다.

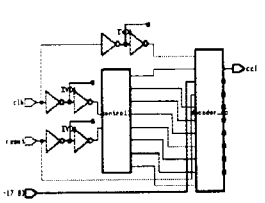
2.2.5 검증 및 합성결과

본 연구에서 설계된 RS 복호기의 검증을 위해 function 레벨 시뮬레이션이 수행되었다. 시뮬레이션은 Sun Ultra Sparc60환경에서 Synopsys^{tm}의 VHDL System Simulator(VSS)를 사용하여 시뮬레이션 하였다. 8개 오류를 정정하는 시뮬레이션은 〈그림 6〉과 같다.

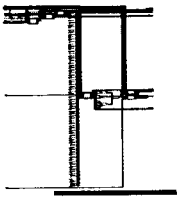


〈그림 6〉 Synopsys 시뮬레이션 결과

〈그림 7〉은 설계된 RS 복호기의 논리 합성한 결과이다. 전체적으로 Control 부분과 Data-Path 부분으로 나뉘어지며, 합성은 Synopsys^{tm}의 Design Analyzer를 사용하였고 lsi-10K 라이브러리를 사용하였다. 주 연산이 이루어지는 Data path부분의 합성 결과는 〈그림 8〉과 같다.



〈그림 7〉 RS 복호기 합성 결과



〈그림 8〉 Data-Path 합성결과

설계된 RS 복호기의 주요 부분의 크기 및 전체 크기는 〈표 1〉과 같다.

단위 : Gate

	Combi. logic	Noncombi. logic	Total logic
Buffer	2790	16056	18846
오증계산	7984	1152	9136
오류위치	12469	2303	14772
오류값 계산	39664	2304	41968
Control	162	281	443
전체 복호기	63881	23248	87129

〈표 2〉 RS 복호기의 논리합성결과

3. 결 론

본 연구에서는 Berlekamp-Massey 알고리즘을 사용하여 디지털TV디코더에 사용되는 RS(Reed-solomon) 복호기를 설계하였다. Berlekamp-Massey 알고리즘은 수정 Euclid 알고리즘에 비해 이해하기 어려우며, 그 계산 과정이 복잡하여 구현하기 어렵다는 단점이 있으나 실질적으로는 크기와 속도에 있어 구현하는데 장점이 있다. 본 연구의 RS 복호기는 디지털 TV의 규격에 맞는 (204, 188, 8)로 VHDL로 설계되어 최대 8개의 심볼, 64비트까지 정정할 수 있으며 I/O Data Rate는 80 Mbit/s이다.

〔참 고 문 헌〕

〔1〕 Shu Lin and Daniel J. Costello, "Error Control Coding : Fundamentals and Application", Prentice-Hall inc.

〔2〕 Moonho Lee, Seungbae Choi, and Jinsu Chang, "A High Speed Reed-Solomon Decoder", IEEE trans. on Consumer Electronics, Vol.41, No4, Nov.1995

〔3〕 JooSeon Kim, ByungGook Chung, YoungHwan Kim, KiWon Lee, "A High Speed RS Decoder using Berlekamp-Massey Algorithm for DVD/CD", MicroSystems R&D Lab. Corporate Technical Operations, SAMSUNG Electronics Co.

〔4〕 Kuang Yung Liu, "Architecture for VLSI Design of Reed Solomon Decoders", Ieee trans. on Computer, Vol.c-33, No.2, Feb.1984