

# Berlekamp-Massey 알고리즘을 이용한 소형 Reed-Solomon 디코우더의 아키텍처 설계

전 우 형<sup>†</sup> · 송 낙 운<sup>††</sup>

## 요 약

본 논문에서는 소형 RS(Reed-Solomon) 디코우더의 효율적인 하드웨어 아키텍처를 제안하였다. 전체 아키텍처는 3단 파이프라인 구조를 택하였으며, 디코우딩 연산시, 에러위치다항식은 BMA(Berlekamp-Massey algorithm)에 의한 fast-iteration 방식으로 구하였으며, 계산의 복잡성이 요구되는 신드롬연산 부분은 ROM 테이블을 이용해서 병렬로 수행하고, 에러위치 다항식의 근을 구하는 부분은 Chien search 알고리즘을 응용한 방법을 ROM을 채택하여 계산하였다. 제안된 디코우더로 3심볼 랜덤에러정정을 수행하며, 시스템클록 25MHz를 사용하여 124Mbps의 디코우딩 데이터율을 가짐을 확인할수 있었다.

## Architecture design of small Reed-Solomon decoder by Berlekamp-Massey algorithm

Woo-Hyung Chun<sup>†</sup> · Nag-Un Song<sup>††</sup>

## ABSTRACT

In this paper, the efficient architecture of small Reed-solomon architecture is suggested. Here, 3-stage pipeline is adopted. In decoding, error-location polynomials are obtained by BMA using fast iteration method, and syndrome polynomials, where calculation complexity is required, are obtained by parallel calculation using ROM table, and the roots of error location polynomial are calculated by ROM table using Chien search algorithm. In the suggested decoder, it is confirmed that 3 symbol random errors can be corrected and 124Mbps decoding rate is obtained using 25 Mhz system clock.

### 1. 서 론

현재 무선통신의 상용화가 급속히 이루어짐에 따라 이에 관련된 이동통신 및 네트워크에 관한 연구가 활발히 이루어지고 있다. 한편 이를 통한 각종 정보화 데이터, 즉, 오디오, 비디오 등의 각종 데이터의 신뢰성 있는 처리가 매우 중요하게 되었다. 이중 비디오서비스를 하기 위해서는 효율적인 소스코딩 뿐만이 아니라

연집에러(burst error)에 강하며, 비트율이 높은 채널 코딩이 필요하게 된다. 이러한 연집에러에 강한 채널 코딩으로서 대표적인 것이 RS 코우드이며 이는 각종 CD, DAT, VTR, TV 등 가전제품과 위성 및 이동통신 등에 널리 쓰이고 있다[1-3].

RS 코우드의 디코우딩방식은 크게 주파수영역(FD : frequency domain)과 시간영역(TD : time domain)에서의 디코우딩으로 나눌수 있다. 이에 관하여 Clark[1]에 의한 정의가 있으며, 이후 Blahut[2, 4]는 시간영역에서의 디코우딩을 'Transform decoding without transform'으로 칭하여 앞서와는 달리 정의하여 진행하였다. 일

† 정 회 원 : 홍익대학교 대학원 전자공학과  
†† 성 회 원 : 홍익대학교 전자공학과 교수  
논문접수 : 1999년 8월 19일, 심사완료 : 1999년 12월 9일

반적으로 주파수 영역에서의 디코우딩방식은 낮은 연산의 복잡성이 요구되는 곳에 유리하고, 시간영역에서의 디코우딩방식은 낮은 제어/통신의 복잡성이 요구되는 곳에 보다 유리한 것으로 알려져 있다. 그러나 속도와 면적의 측면에서 TD 디코우딩 방식이 보다 유리한 것으로 알려지고 있다[5,6].

RS 코우드의 디코우딩 단계는 신드롬을 계산하는 부분, 에러위치 다항식을 구하는 부분, 이의 근과 에러값을 구하고 부분, 그리고 이를 정정하는 부분으로 나눌 수 있다. Clark[1]에 의하면 시간영역에서의 디코우딩의 경우, 에러크기의 연산이 더 필요하게 된다. 한편, 이중 에러위치 다항식을 구하는 방법에는 BMA[7], MEA(Modified Euclid's algorithm)[8], Reed의 연속된 fraction에 근거한 방법[9] 등이 있으며 이들은 거의 같은 정도의 계산상 복잡성을 가진 것으로 알려져 있다.

에러다항식을 구하는 방법중에서, 표준적인 매트릭스 역변환방법이외에 다양한 방법이 제안되었다. BMA 방법의 경우, 효율적인 고속 반복방법이 적용되었으며, 연산상의 나눗셈을 줄이는 방법 등이 제안되었다[10, 11]. 한편, 이 BMA 적용에서 연산량 및 속도를 줄이는 것이 중요한 연구점이 되었으며, Blahut의 시간영역 디코우딩의 정의를 이용한 가변적인 디코우딩이 가능한 구조[12], 변형된 BMA와 파이프라인 구조[13], 면적과 속도 측면에서 개선된 구조[14, 15] 등이 제안되었다. BMA보다 후에 제안된 EA(Euclid algorithm)의 경우, GCD(greatest common division) 개념을 통하여 관련 식을 구하며 이 과정 중에서의 나눗셈 등을 개선하는 변형된 MEA 알고리즘이 제안되었으며, 아울러 이의 아키텍처 설계에 관한 많은 연구가 있다[4, 16-19].

본 논문에서는 비교적 적은 비트를 사용하여 이의 속도개선을 위한 RS 디코우더의 구조를 제안하였다. 이에서 Clark[1]에 의한 시간영역에서의 연산방법을 사용하였으며, 에러위치 다항식을 구하는데 있어서 정정 가능한 에러의 수가 많은 경우에 비교적 효율이 높으며 고속반복 연산에 유리한 알고리즘인 BMA 방식을 사용하였다. 또한, 디코우딩 데이터처리속도를 높이기 위하여, 파이프라인단을 사용하여 쓰루풋을 향상시켰으며, 신드롬과 에러의 연산부위에 속도와 회로의 복잡도 등을 고려한 ROM을 채택한 구조를 제안하였다.

이를 위한 본 논문은 다음과 같이 구성되었다. 2장에서 일반적인 RS 인코우딩 알고리즘과 디코우딩 알고리즘에 대한 배경을 서술하였고, 3장에서 제안된 하

드웨어 아키텍처에 관해서 서술하였으며, 4장에서 모의실험 및 결과를 통해서 제안된 하드웨어 아키텍처의 성능을 실험하였고, 5장에서 결론을 기술하였다.

## 2. 기본 이론

### 2.1 RS 인코우딩 알고리즘

RS 코우드심볼은 Galois field,  $GF(2^m)$ 에 속하는 원소이다. 일반적인 RS 코우드는  $(n,k,t)$ ( $n$ : 코우드워드 길이,  $k$ : 정보길이,  $t$ : 에러정정능력)로 표현된다. 코우드는 최대  $t$  심볼에러를 정정할 수 있으며 여기서  $t=(n-k)/2$ 가 되게 된다. RS 코우드는 코우드워드 다항식  $c(x)$ (차수:  $n-1$ ), 정보다항식  $d(x)$ (차수:  $k-1$ ), 생성다항식  $g(x)$ (차수:  $n-k$ )와 패리티체크용  $x^{2t}$ (차수:  $2t$ )로 구성되어 있다. 일반적으로 생성다항식  $g(x)$ 는 다음과 같다.

$$g(x) = (x + \beta^{j_0})(x + \beta^{j_0+1}) \cdots (x + \beta^{j_0+2t-1}) \quad (1)$$

여기에서,  $j_0$ 는 오프셋항이고  $t$ 는 에러정정능력이다. 이제 주된 정보다항식과 생성다항식을 위한 보조방정식은 다음과 같다.

$$d(x) = d_{k-1} x^{k-1} + d_{k-2} x^{k-2} + \cdots + d_1 x + d_0 \quad (2)$$

$$\begin{aligned} r(x) &= R_{g(x)}[x^{j_0} d(x)] \\ &= c_{n-k-1} x^{n-k-1} + \cdots + c_1 x^1 + c_0 \\ &= r_{n-k-1} x^{n-k-1} + \cdots + r_1 x^1 + r_0 \end{aligned} \quad (3)$$

이를 아래 식에 대입하여 전달코우드를 생성한다. 즉, 정보다항식  $d(x)$ 가 RS 인코우더를 통해서 전송된 후 패리티체크 심볼들이 전송되어져 코우드워드 다항식  $c(x)$ 가 생성된다.

$$\begin{aligned} c(x) &= x^{2t} d(x) - r(x) \\ &= c_{n-1} x^{n-1} + c_{n-2} x^{n-2} \\ &\quad + \cdots + c_1 x^1 + c_0 \end{aligned} \quad (4)$$

### 2.2 RS 디코우딩 알고리즘

전송된 코우드블록  $c(x)$ 는 채널에 의한 잡음의 간섭을 받게 된다. 전송받은 다항식  $R(x)$ 는  $c(x)$ 와 에러다항식  $e(x)$ 에 의해서 다음과 같이 표현될 수 있다.

$$\begin{aligned} R(x) &= c(x) + e(x) \\ &= R_{n-1} x^{n-1} + \cdots + R_1 x + R_0 \end{aligned} \quad (5-1)$$

여기에서 에러는 에러위치  $x^i$ 와 에러크기  $e_i$ 로 다음 식과 같이 얻어진다.

$$e(x) = \sum_{i=1}^t e_i x^i \quad (5-2)$$

이들로부터 RS 디코딩의 각 과정, 즉, 신드롬, 에러위치 다항식, 에러의 근과 크기, 에러정정의 연산이 연속적으로 이루어진다[2, 3].

2.2.1 신드롬 계수와 다항식을 계산하는 부분

첫번째인 신드롬다항식  $S(x)$ 의 연산에서는 먼저 신드롬계수  $S_j$ 를 구한후 이를 다음 식에 대입하여 구한다.

$$S_j = R(a^j) \quad (6-1)$$

$$S(x) = \sum_{j=0}^{2t-1} S_j x^j \quad (6-2)$$

2.2.2 에러위치 다항식과 에러의 근과 크기의 계산

이제  $r = 1, \dots, 2t$ 에 대하여 신드롬이 주어졌을 때, 에러위치 다항식  $\sigma^{(2t)}(x)$ 는 다음 조건을 만족하는 최소차수의 다항식이 된다.

$$S_r + \sum_{j=1}^{r-1} \sigma_j^{(2t)} S_{r-j} = 0, \quad (7)$$

$$r = l_{2t+1}, \dots, 2t \ \& \ \sigma^{(2t)} = 1$$

이  $\sigma^{(2t)}(x)$ 를 구하기 위해 채택한 BMA 연산은 다음과 같다[2].

Initialization :  $\sigma^{(0)}(x) = 1, \ b^{(0)}(x) = 1, \ l_0 = 0$

For  $r = 1 : 2t,$

$$d_r = \sum_{i=0}^{l_r-1} \sigma_i^{(r-1)} S_{r-i}$$

$$\delta_r = \begin{cases} 1 & \text{if } d_r \neq 0, \ 2l_{r-1} \leq r \\ 0 & \text{otherwise} \end{cases}$$

$$l_r = \delta_r(r - l_{r-1}) + (1 - \delta_r) l_{r-1}$$

$$\begin{pmatrix} \sigma^{(r)}(x) \\ b^{(r)}(x) \end{pmatrix} = \begin{pmatrix} 1 & -d_r x \\ d_r^{-1} \delta_r & (1 - \delta_r)x \end{pmatrix} \begin{pmatrix} \sigma^{(r-1)}(x) \\ b^{(r-1)}(x) \end{pmatrix} \quad (8)$$

이제 이를 이진 BCH 코우드에 한정적으로 적용하여 고속반복법을 적용할수 있다[2, 3]. 요약하면 이는, 앞의 식 (8)에서,  $r$ 의 값이 우수인 경우  $d_r=0$ 가 되고  $r$ 의 값이 기수인 경우 다음식으로 얻어진다.

$$\begin{pmatrix} \sigma^{(r)}(x) \\ b^{(r)}(x) \end{pmatrix} = \begin{pmatrix} 1 & -d_r x^2 \\ d_r^{-1} \delta_r & (1 - \delta_r)x^2 \end{pmatrix} \begin{pmatrix} \sigma^{(r-2)}(x) \\ b^{(r-2)}(x) \end{pmatrix} \quad (9)$$

이제 에러를 정정하기 위하여 위치와 크기값을 결정해야한다. 이를 위한 에러연산 방정식은 다음과 같다.

$$\alpha(x)S(x) \equiv \Omega(x) \pmod{x^{2t}} \quad (10)$$

$\Omega(x)$  : 에러연산 다항식

이 경우 앞의 식 (10)의  $\alpha(x)=0$ 를 만족하는 에러위치 다항식의 근을 Chien search방식에 의하여 구하며 이의 역수로부터 에러위치를 구한다. 한편, 에러크기는 Forney 알고리즘에 의하여 구하며, 에러위치 다항식의 근이  $x = \beta_i^{-1}$ 일 때, 이는 다음 식과 같다.

$$e_i = \beta_i^{-1} \Omega(\beta_i^{-1}) / \prod_{j \neq i} (1 - \beta_j \beta_i^{-1}) \quad (11)$$

2.2.3 에러의 정정

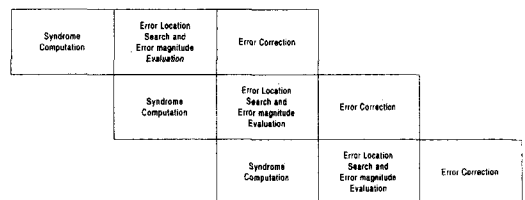
이제 식 (10), (11)을 통해 구해진 에러위치와 에러크기를 구해 앞의 식 (5-2)에 의하여 에러를 구하고 식 (5-1)에 의해 이를 정정한다.

3. 제안된 RS-디코더의 아키텍처

본 논문에서 구현한 RS 코우드는 (31,25,3)으로 최대 에러정정심볼이 3(15비트)이며, 코우드율은  $25/31=0.806$ 이다. 여기에서 생성다항식은 식 (1)의 오프셋항  $j_0=1$ , 에러정정능력  $t=3$ 을 대입해 얻어진다. 한편, 각각의 심볼들은 다음과 같은 primitive 다항식에 의해서 생성된다.

$$p(x) = x^5 + x^2 + 1 \quad (12)$$

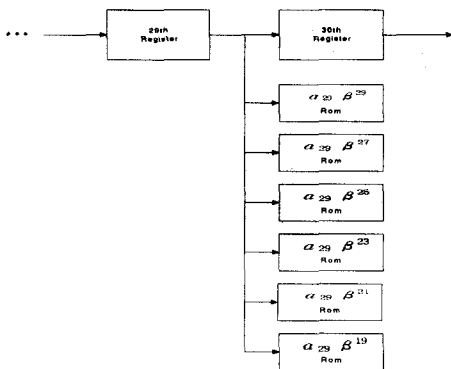
여기서  $p(x)$ 는  $GF(2^5)$ 상에 있다. 디코딩 알고리즘으로는 시간영역에서의 연산방법을 사용하였으며, 에러위치 다항식은 BMA에 의한 fast-iteration 방법을 사용하였다. 제안된 RS 디코더의 아키텍처는 다음 그림과 같이 3단 파이프라인을 사용하였다.



(그림 1) 3단 파이프라인 아키텍처

3.1 신드롬 연산회로의 구현

본 논문에서는 6개의 신드롬을 동시에 계산해서 연산속도를 올리는데 주안점을 두어서 설계하였다. 기본적인 구조는 문헌[2-ch.6]와 같으나, 이때 나눗셈 연산 시 Galois field에서의 곱셈결과를 미리 계산된 값을 입력한 ROM을 사용하여 빠른 시스템 클럭에서도 동작할수 있도록 하였다. 이는 Galois field의 연산을 위해 곱셈기, 나눗셈기 등을 구현하기보다는 곱셈 또는 나눗셈의 결과가 같은 field안의 한 element가 되는 것을 이용하여 ROM table로 구현한 것으로, 빠른 클럭속도에 적합하며, 필요한 ROM의 크기도 4Kbytes로 비교적 작다. 이의 회로의 단순화측면에서는 논리회로방식보다, 속도면에서는 cellular-array 방식보다 장점이 있게 된다. 이제 다음 (그림 2)는 29번째 블록을 나타내며 이러한 블록이 30개 연결되어 있는 구조이다.

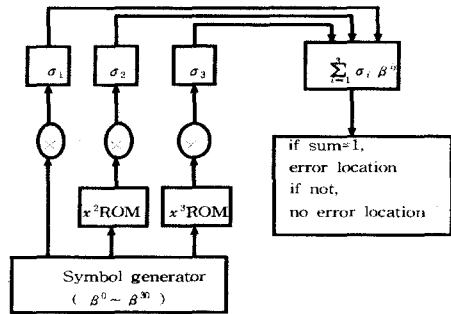


(그림 2) 신드롬연산의 병렬연산

3.2 에러위치다항식 및 에러의 근과 크기 계산회로의 구현

에러위치다항식은 BMA 방식으로 구하였다. 에러위치 다항식의 근은 GF(2^5)을 순환하는 카운터로 심볼 생성기를 설계하여 Chein search 유닛에 적용시켰으며 이를 통해서 신드롬연산블록의 시스템 클럭수와 같이 맞추며 파이프라인 구조를 가능하도록 하였으며, (그림 3)에 이를 나타내었다.

에러크기 계산회로는 Forney 알고리즘을 사용하여 구현하였으며, 5비트 어드레스와 5비트 워드를 가지는 ROM으로 나누기 연산을 단순화하여 조합회로만으로 구현하였다.

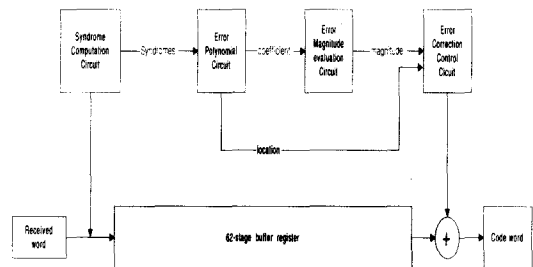


(그림 3) Chein search를 응용한 에러위치다항식의 근 연산

3.3 에러정정회로의 구현

버퍼에 저장되어 있던 전송된 다항식을 출력시킴과 동시에 에러가 발생한 위치에 계산된 에러의 크기를 mod-2 덧셈을 하여 정정하도록 구현하였다. 파이프라인으로 인하여 버퍼의 수는 전송된 다항식의 두배인 62개가 되며, 이 부분도 카운터를 통한 제어신호발생으로 구현이 가능하도록 하였다.

제안된 RS-디코더의 전체적인 하드웨어 아키텍처를 다음 (그림 4)에 보였다.

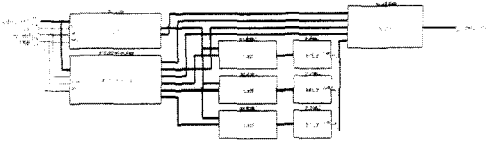


(그림 4) RS-디코더의 하드웨어 아키텍처

4. 시뮬레이션 및 결과 검토

제안된 구조의 전체적인 RS 디코더를 우선 Microsoft사의 Visual C++을 사용하여 검증하였다. RS 디코더의 하드웨어는 VHDL로 설계하였는데, 이의 로직합성과 검증은 Mentor Graphics Inc.의 Quick HDL과 Autologic을 통해서 수행하였으며, 동사의 default library를 사용하였다. 설계는 상위단계에서 신드롬연산블록, 에러위치 및 크기연산 블록, 에러정정블록, 버퍼레지스터 블록으로 나누고, 각각의 블록을 다시 여러개의 하위

단계로 나눈 다음, 최하위 component들을 behavioral과 RTL로 코우딩한 후, 상위 레벨에서 하위 component들을 불러들여 structural로 코우딩하였다. 이를 통한 로직합성결과를 (그림 5)에 나타내었으며, 이는 (그림 4)와 대응되어 버퍼, 에러정정연산, 에러정정의 블록부위로 이루어졌다.



(그림 5) 제안된 RS 디코우더의 VHDL을 이용한 로직합성 결과

이제, 에러정정을 확인하기 위하여 주어진 영상에 노이즈에러를 주어 이를 디코우더를 통하여 정정되는 과정을 Visual C++로 시뮬레이션하였다. 이때 영상에 들어간 에러는 Visual C++ 5.0에서 제공되는 rand 함수를 통해서 발생시켰으며, 결과적으로 RS 디코우딩을 통하여 주어진 에러가 정상적으로 복원이 됨을 확인할 수 있었다. C 언어로 일단 확인을 한 후, 이를 제안된 아키텍처를 갖는 RS 디코우더를 통하여 VHDL을 이용하여 시뮬레이션한 결과를 다음 (그림 6)에 보였으며 이때 시스템클록은 40ns(25Mhz)로 수행하였다.



(a) 에러가 들어간 이미지

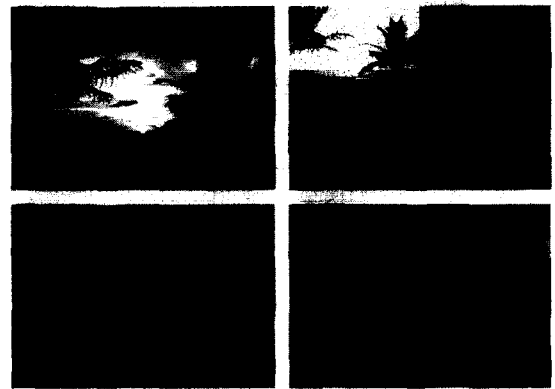


(b) 제안된 RS 디코우더로부터 복원된 이미지

(그림 6) VHDL을 이용한 시뮬레이션 결과

Lenna의 영상에서 BER의 값은 (복호전 : 0.0457, 복호후 : 0.0007)으로 얻어졌다. 한편 영상결과에 의하면, 전반적으로 에러정정이 이루어졌으나, 부분적으로 에러정정이 안 된 부분이 있는데 이것은 한 코우드에 3심볼보다 많은 수의 에러가 들어있을 경우이며 이것은 interleaving을 함으로써 해결될수 있으리라 본다.

다음으로 이를 이용하여 SBF(sub-band filtering)를 이용한 웨이블릿 영상처리에 적용하여 에러정정 신호처리를 C언어로 시뮬레이션하였다. 이의 결과를 다음 그림에 보였으며, 이는 각각 ((a)원영상, (b)웨이블릿변환과 에러합성영상, (c)웨이블릿 복원영상(에러정정 생략), (d)웨이블릿 복원영상(에러정정 수행))을 나타낸다.



(a) 원영상 (b) 웨이블릿변환과 에러합성영상  
(c) 복원영상(에러정정 생략) (d) 복원영상(에러정정 수행)

(그림 7) SBF를 이용한 영상신호처리에서의 에러정정 수행 결과의 비교

이를 통하여 SBF에도 정상적으로 정정이 됨을 확인하였다. 이때 BER은 (복호전 : 0.9226, 전체 ECC 복호 후 : 0.8278)로 얻어졌는데 이의 미흡함은 데이터 처리시 truncation 에러에 기인함으로 발생하였다. SBF의 경우, 향후 이는 밴드별로 가변적인 코우드(소스 및 채널)의 분배와 이에 따른 보다 효율적인 정정이 가능하리라 본다[20].

본 시뮬레이션에서 파이프라인을 통한 쓰루풋의 증가를 180x125 8비트 gray scale 영상을 처리하는데 걸리는 시간을 통해서 분석해 보면 3배정도의 쓰루풋을 증가시켰다. 다음, 표 1은 기 문헌에서 설계한 RS 디

코더를 본 논문에서 제안한 RS 디코더와 비교한 것이다. 참고문헌 [18]에서의 sequential한 구조를 이용했을 때 보다 파이프라인 구조를 이용했을 때, I/O 데이터율이 증가했다.

〈표 1〉 기존에 제안된 방법과의 비교

	제안된 아키텍처	문헌[17]	문헌[18]
I/O Data rate	15.6M Byte/s (124 Mbps)	23M Byte/s (184 Mbps)	5M Byte/s (80 Mbps)
structure	3-stage Pipeline	3, 4-stage Pipeline	Sequential
Syndrome generator	parallel	parallel	Sequential
Error locator Search 알고리즘	Berlekamp-Massey 알고리즘	Euclid's 알고리즘	Euclid's 알고리즘

### 5. 결론

본 논문은 BMA 방식을 이용한 (31, 25) RS 디코우더의 아키텍처를 제안하고 이를 Visual C++과 VHDL을 통한 시뮬레이션으로 검증하였다.

제안된 RS 디코우더의 하드웨어 설계시 3단 파이프라인을 사용함으로써 쓰루풋을 증가시켰다. 다량의 연산량이 요구되는 신드롬연산회로는 동시에 모든 신드롬을 발생시키는 병렬구조의 ROM을 사용하여 구현하였고, 에러연산에서는 BMA를 이용한 에러위치다항식 회로를 구현하였으며 이때 에러 위치, 크기 계산회로는 ROM을 이용하여 Chien search, Forney 알고리즘을 적용하여 구현하였다.

본 논문에서 제안한 RS 디코우더는 25Mhz의시스템클럭에서 124Mbps로 동작하므로써 지상파 디지털 HDTV에서 요구되는 20Mbps이상의 디코우딩 데이터율을 만족시켰다. 또한, RAM를 사용하지 않고, Galois field에서의 ALU를 사용하는 대신 ROM을 사용함으로써, 데이터패스를 단순화시켰으므로, 레이아웃시 칩면적이 작아질 것이 예상된다. 아울러 비교적 낮은 코우드비트를 이용하여 주어진 영상에 대하여 정상적인 에러정정을 확인할수 있었다.

향후 이는 SBF 소스코우딩과 연계하여 가변적인 설계를 하여 이동비디오 장비와 같이 크기가 작고 연결에 강한 에러정정 코덱이 필요한 곳에서의 응용이 기대된다. 실제 HDTV 등에 응용되기 위해서는 더욱 높은 코우드율과 에러정정율을 가지는 코우드에도 적

합하도록, 향후 아키텍처의 개선이 필요하며, RS 코우드와 컨벌루션코우드를 결합시킨 concatenated 코우드에의 연구도 기대된다.

### 참 고 문 헌

- [1] G. C. Clark, J. B. Cain, 'Error-correction coding for digital communications,' ch.5, Plenum press 1981.
- [2] Richard E. Blahut, 'Theory and Practice of Error Control Codes,' Addison-Wesley, 1984.
- [3] M. Rhee, 'Error Correcting Coding Theory,' MH, 1989.
- [4] R. E. Blahut, "A universal Reed-Solomon decoder," IBM J. R&D, Vol.28, No.2, pp.150-158, March 1984
- [5] H. M. Shao, I. S. Reed, "On the VLSI design of a pipeline R-S decodes using systolic arrays," IEEE T-Comp. Vol.37, No.10, pp.1273-1280, Oct. 1988.
- [6] Y. Jeong, W. Burleson, "High-level estimation of high-performance architectures for R-S decoding, 1995 IEEE, pp.720-723.
- [7] Berlekamp, 'Algebraic coding theory,' MH, NY, 1968.
- [8] Y. Sugiyamo et al., "A method for solving equation for decoding Goppa codes," IEEE T-Cont., Vol.27, pp.87-99. 1975.
- [9] L. Welch, R. Scholtz, "Continued fraction and Berlekamp's algorithm," IEEE T-IT, Vol.25, pp.19-27, Jan. 1979.
- [10] I. S. Reed et al., "VLSI design of inverse-free BMA," IEE proc., Vol.138, No.5, pp.295-298, Sept. 1991.
- [11] H. Chang, C. B. Shung, "A (208,192,8) R-S decoder for DVD applications," IEEE Int. Conf. Comm., Vol.2, pp.957-960, 1998.
- [12] Y. R. Shayan et al., "A versatile time-domain Reed-Solomon decoder," IEEE J. Selec. in Areas Comm., Vol.8, No.8, pp.1535-1542, Oct. 1990.
- [13] S. Choomchuay et al., "Time-domain algorithms and architectures and architectures for R-S decoding," Proc.-1, Inst. Electron. Eng., Vol.140, pp.189-196, June 1993.
- [14] J. M. Hsu et al., "An area-efficient pipelined VLSI architecture for decoding R-S codes based on a time-domain algorithm," IEEE T-CAS-VT, Vol.7, No.6, pp.864-871, Dec. 1997.

- [15] J. Kim et al., "A high-speed RS decoder using BMA for DVD/CD," IEEE conf. proc., Vol.2, pp.1430-1434, 1998.
- [16] S. R. Whitaker et al., "Reed Solomon VLSI Codec for Advanced Television," IEEE T-CAS-VT, Vol.1, No.2, pp.230-236, June 1991.
- [17] M. Lee et al., "A High Speed Reed-Solomon Decoder," Proc. VLSI Sig proc. Vol.8, pp.362-367, 1995.
- [18] Y. Park, S. Park, "Reed Solomon VLSI Codec for HDTV," Asic Center Goldstar Central Research Lab., 1994.
- [19] 이 주태, 이 승우, 조 중휘, "하드웨어 공유 극대화에 의한 RS decoder의 VLSI 설계," 전자공학회지, Vol.86, No.3, pp.8-16, 1993.
- [20] R. Stedman, H. Gharavi et al., "Transmission of subband-coded images via mobile channels," IEEE T-CASVT, Vol.3, No.1, pp.15-26, Feb., 1993.



### 전 우 형

e-mail : wjluv@hitel.net

1997년 홍익대학교 전자공학과 졸업  
(학사)

1999년 홍익대학교 대학원 전자공  
학과 졸업(석사)

1999년~현재 콤팩트 시스템



### 송 낙 운

e-mail : snukeh@wow.hongik.ac.kr

1975년 서울대학교 전자공학과 졸업  
(학사)

1986년 Univ. Texas Austin(Ph.D)

1986년~1989년 금성반도체 근무

1989년 홍익대 전자공학과 부교수

관심분야 : VLSI시스템 자동화 설계