

論文99-36C-3-2

# 하드웨어 공유 극대화에 의한 $GF(2^8)$ Reed-Solomon Decoder의 VLSI 설계

## (VLSI Design of Reed-Solomon Decoder over $GF(2^8)$ with Extreme Use of Resource Sharing)

李柱泰\*, 李昇優\*, 趙仲彙\*

(Jootae Lee, Seungwoo Lee, and Joonghwee Cho)

### 요 약

본 논문에서는 변형된 유클리드(Modified Euclid) 알고리즘을 이용한  $GF(2^8)$ 의 Reed-Solomon(RS) 복호기에 대하여 VLSI로 설계하였다. 면적의 관점에서 효율적인 설계를 위하여 레지스터와 유클리드 ALU를 최대로 공유하는 여러 가지의 새로운 구조를 제안하였다. 에러 위치 다항식( $\sigma(x)$ )과 에러 평가 다항식( $\omega(x)$ )을 계산하기 위하여 16개의 ALU 대신에 1개의 ALU를 사용하였으며, 이들 다항식의 계수를 저장하기 위한 레지스터를 24개 대신에 18개를 사용하였다. VHDL을 이용하여 시뮬레이션을 행하고 FLEX™ FPGA를 이용하여 구현을 행함으로써 제안한 구조에 대한 정확성을 검증하였으며, DVD(Digital Versatile Disc) 시스템을 위하여 (208,192,17) RS 부호와 (182,172,11) RS 부호에 대한 복호 기능을 갖는 RS 복호기를  $0.6\mu\text{m}$ 의 CMOS TLM Compass™ 라이브러리를 사용하여 게이트 숫자가 약17,000이고, 코어 면적이  $2.299 \times 2.284$  ( $5.25 \text{ mm}^2$ )인 VLSI로 설계함으로써 효율성을 검증하였다. 한편, 설계한 칩은 20MHz로 동작함을 확인하여 DVD의 요구 조건인 3.74MHz를 만족함을 확인하였다.

### Abstract

This paper describes a VLSI design of Reed-Solomon(RS) decoder using the modified Euclid algorithm, with the main theme focused on the  $GF(2^8)$ . To get area-efficient design, a number of new architectures have been devised with maximal register and Euclidean ALU unit sharing. One ALU is shared to replace 18 ALUs which computes an error locator polynomial and an error evaluation polynomial. Also, 18 registers are shared to replace 24 registers which stores coefficients of those polynomials. The validity and efficiency of the proposed architecture have been verified by simulation and by FLEX™ FPGA implementation in hardware description language VHDL. The proposed Reed-Solomon decoder, which has the capability of decoding RS(208,192,17) and RS(182,172,11) for Digital Versatile Disc(DVD), has been designed by using  $0.6\mu\text{m}$  CMOS TLM Compass™ technology library, which contains totally 17k gates with a core area of  $2.299 \times 2.284$  ( $5.25 \text{ mm}^2$ ). The chip can run at 20MHz while the DVD requirement is 3.74MHz.

### I. 서 론

\* 正會員, 仁川大學校 電子工學科

(Dept. Electronic Engineering, Univ. of Incheon)

※ 본 연구는 한국과학재단 지정 인천대학교 멀티미디어 연구센터와 정보통신연구관리원 산학연 협력과제의 부분적 지원에 의한 것입니다.

接受日字:1999年1月12日, 수정완료일:1999年3月2日

디지털 통신 및 디지털 저장 기술의 발전에 따라 여러 형태의 잡음, 왜곡, 및 간섭 등에 의하여 발생하는 오류를 수신 및 데이터 복원 장치에서 제거하는 기능을 수행하는 오류 정정 부호에 대한 연구와 이를 VLSI로 구현하는 연구가 활발하게 진행되고 있다.

오류 정정 부호는 크게 길쌈 부호(Convolutional Code)와 블록 부호(Block Code)로 나뉘며<sup>[1]</sup>, 최근에 등장한 터보 코드(Turbo Code)<sup>[2]</sup>가 있다. 이러한 부호들 가운데 블록 부호는 전송하고자 하는 정보와는 독립적으로 약간의 검색정보를 추가시키고 이를 분석함으로써 수신되는 정보 신호에 발생하는 오류를 검출하고 정정한다. 이와 같은 부호어의 구성에 의하여 오류 정정 능력에 제한을 받지만 오류 발생의 형태에 무관하게 원 신호를 복원할 수 있는 장점을 가지고 있기 때문에 주로 연접 에러(Burst Error) 정정 능력이 길쌈부호보다 우수하다. 특히, Reed-Solomon (RS) 부호는 비이진(Non-Binary) 연산을 행하기 때문에 연접 에러 발생에 대한 정정 능력의 관점에서 가장 효율적인 부호로 알려져 있어 이에 대한 연구가 가장 활발하다.

RS 부호의 응용 예로 DVD(Digital Versatile Disc)의 Inner Product 부호로 (182,172,11) RS 부호와 Outer Product 부호로 (208,192,17) RS 부호<sup>[3-5]</sup>가 사용된다. 그리고 위성 방송 송/수신기 및 LMDS(Local Multi-point Distribution Services) 등과 같은 무선 CATV 송/수신기에는 길쌈 부호와 결합된 연접 오류 정정 부호(Concatenated Forward Error Correction Code)<sup>[6]</sup> 형태로 사용되며 (204, 188, 17) RS부호가 사용된다.

RS 부호의 부호화 및 복호 과정은  $m$ -비트의 심벌 단위로 이루어지며, 복호에서 가장 중요한 단계는 오류 위치 다항식 및 오류 평가 다항식을 추출하는 과정이다. 이 과정은 다른 블록부호에 비해 구현이 복잡하며, 이를 하드웨어로 구현 할 시에는 칩 면적이 커지는 단점을 가지고 있다. 이러한 단점을 극복하기 위하여 여러 가지 복호 알고리즘<sup>[7-8]</sup>이 제안되었고, 각 알고리즘에 대해서도 복호기의 면적 복잡도(Area Complexity)를 줄이기 위하여 여러 가지 하드웨어 구조들이 연구되어 왔다<sup>[3-6] [9-11]</sup>. 그러나, 기존의 제안된 구조를 분석한 결과 에러 위치와 평가 다항식의 추출 연산에 있어서 효율적인 레지스터의 공유가 이루어져 있지 않았으며<sup>[3] [9]</sup>,  $GF(2^8)$ 상에서의 곱셈기가 최소 3개<sup>[4] [5]</sup> 이상이 사용되고 있음을 발견하였다.

따라서, 본 논문에서 제안된 RS 부호의 복호기는 오류 위치 다항식( $\alpha(x)$ )과 오류 평가 다항식( $\omega(x)$ )을 추출하는 블록에 대해서 여러 가지 복호 알고리즘 중

두 다항식의 최대공약수를 구하여 오류 위치 다항식( $\alpha(x)$ )과 오류 평가 다항식( $\omega(x)$ )을 추출 할 수 있는 변형된 유클리드 알고리즘(Modified Euclid Algorithm)을 택한다. 이 알고리즘으로부터 연산 과정에서 핵심이 되는 유클리드 코어(Core) 블록을 서로 공유함으로써  $GF(2^8)$ 상에서의 곱셈기의 수를 2개로 줄이며, 다항식의 계수를 저장하는 레지스터의 효율적인 공유로 인하여 하드웨어의 복잡도를 최대한 줄일 수 있는 마이크로 아키텍처를 제안한다.

제안한 구조를 이용하여 DVD에 사용하기 위한 (182,172,11) RS 부호와 (208,192, 17) RS 부호의 복호기를 설계하며, VLSI로 구현한다.

II장에서는 RS 부호 및 복호 알고리즘에 대한 소개를 하고, III장에서는 II장에서 선택된 알고리즘에 따른 마이크로 아키텍처 설계에 대하여 기술하며, IV장에서는 이에 대한 실험 결과 및 RS 복호기의 각 블록에 대한 성능 비교를 하여 본 논문의 우수성을 보이며, V장에서는 결론과 앞으로의 연구 과제에 대하여 기술한다.

## II. RS 부호 및 복호 알고리즘

### 1. 부 호

RS 부호는 다른 블록 부호와는 달리  $m$  비트의 심벌단위로 부호 및 복호를 수행한다. 따라서,  $(n,k,d)$  RS부호의 각 파라미터는 다음과 같이 정의된다.

$$\text{부호어의 길이} : n = 2^m - 1$$

$$\text{정보어의 길이} : k = n - 2t$$

$$\text{부호어의 최소거리} : d = n - k + 1$$

하나의 심벌이  $m$  개의 비트로 표현되었다면, 이에 대한 원시 다항식(primitive polynomial  $p(x)$ )은 식 (1)과 같으며, 부호화 과정에서 검색 심벌을 생성하기 위하여 사용되는 생성 다항식(generate polynomial  $g(x)$ )은 오류 정정이 가능한 심벌의 수가  $t$ 일 때, 식 (2)와 같이 표현된다.

$$p(x) = x^8 + x^4 + x^3 + x^2 + 1 \quad (1)$$

$$g(x) = \sum_{i=0}^{2t-1} (x + a^i) = (x + a^0)(x + a^1) \cdots (x + a^{2t-1}) \quad (2)$$

부호의 생성은 식(3)과 같이 systematic의 성질을 이용하여 부호화 한다. 즉, 부호어에 정보 심벌의 형태를 그대로 유지하고 검색 심벌을 정보 심벌의 앞이나

뒤에 첨가시킴으로써 부호어를 생성한다.

$$c(x) = m(x)x^{2t} + (m(x)x^{2t} \bmod g(x)) = q(x)g(x) \quad (3)$$

2. 복호

RS 부호의 복호 과정은 그림 1에서의 같이 4단계에 의해서 이루어진다.

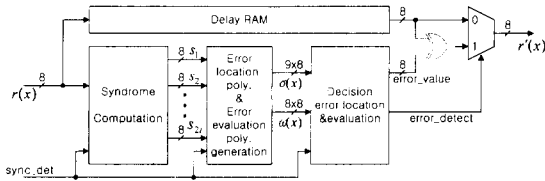


그림 1. (n,k,d)RS 디코더의 블록도  
Fig. 1. Block diagram of (n,k,d) RS decoder.

1-단계 : 수신된 신호의 다항식으로부터 신드롬을 구한다.

2-단계 : 수신다항식에서 얻어진 신드롬으로부터 알고리즘을 이용하여 오류 위치 다항식(α(x)) 및 오류 평가 다항식(ω(x))을 추출한다.

3-단계 : 2-단계로부터 구해진 오류 위치 다항식과 오류 평가 다항식으로부터 오류의 값과 위치를 결정한다.

4-단계 : 결정된 오류 위치와 오류 값을 delay RAM(FIFO)을 통해 나온 수신신호와 비교하여 오류를 정정한다.

위의 4 단계에서 2-단계는 RS 복호기의 전체 면적에서 가장 큰 비중을 차지하며, 연산량도 많다. 따라서, 이 부분에 대한 면적 복잡도나 연산량을 줄이기 위하여 여러 가지의 알고리즘이 소개되어 왔다. 그 대표적인 알고리즘으로는 다음의 3가지로 나뉘어진다.

- Peterson-Gorenstein-Zeirlar 알고리즘<sup>[7] [8]</sup>
- Berlekamp-Massey 알고리즘<sup>[4-5] [7-8]</sup>
- Euclid 알고리즘<sup>[3] [6-9]</sup>

일반적으로, 정정해야 할 오류의 개수가 적을 때는 Peterson-Gorenstein-Zeirlar 알고리즘 같이 신드롬으로부터 직접 오류의 위치와 값을 추출하는 알고리즘도 적합하지만, 정정해야 할 오류의 개수가 많아짐에 따라 이러한 알고리즘은 구현상 어려운 점이 많다. 따라서, 최근 오류 정정 능력이 커짐에 따라 복호 알고리즘으로 Berlekamp-Massey 알고리즘과 변형된 유클리드 알고리즘이 보편적으로 적용된다. 특히, 변형된 유클리드 알고리즘은 임의의 두 다항식의 최대공약수

를 구함으로써 오류 위치 다항식과 오류 평가 다항식을 추출한다. 이러한 변형된 유클리드 알고리즘의 연산구조는 반복적인 연산구조를 가지기 때문에 하드웨어 구현 시 핵심블록의 공유가 용이하다는 장점을 가진다.

변형된 유클리드 알고리즘의 계산과정은 다음과 같으며, mod 2의 연산을 한다.

① 초기값 설정

$$A_0 = x^{2t}, \quad B_0 = s(x), \quad M_0 = 1, \quad L_0 = 0$$

② A와 B의 차수를 계산하여 A,B값을 변형한다.

$$\Delta = 1 \quad \text{if } \deg A_i - \deg B_i \geq 0$$

$$\Delta = 0 \quad \text{if } \deg A_i - \deg B_i < 0$$

$$E_i = B_i, \text{ 의 최고차 항의 계수}$$

$$F_i = A_i, \text{ 의 최고차 항의 계수}$$

$$\Delta = 1 \quad A_{i+1} = E_i A_i(x) - F_i B_i(x) x^{\deg A_i - \deg B_i}$$

$$B_{i+1} = B_i$$

$$L_{i+1} = E_i L_i(x) - F_i M_i(x) x^{\deg A_i - \deg B_i}$$

$$M_{i+1} = M_i$$

$$\Delta = 0 \quad A_{i+1} = E_i A_i(x) x^{\deg B_i - \deg A_i} - F_i B_i(x)$$

$$B_{i+1} = A_i$$

$$L_{i+1} = E_i L_i(x) x^{\deg B_i - \deg A_i} - F_i M_i(x)$$

$$M_{i+1} = L_i$$

③ A<sub>i</sub>의 차수가 오류 정정 능력 t 보다 크거나 같으면 i의 값을 1 증가시켜 ①의 과정으로 되돌아간다. 만약 t 보다 작으면 그때의 A<sub>i</sub>의 값을 오류 평가 다항식(ω(x)), L<sub>i</sub>의 값을 오류 위치 다항식(α(x))으로 결정을 하고 반복 과정을 중단한다.

$$\alpha(x) = L_i, \quad \omega(x) = A_i, \quad (\deg[A_i] < t)$$

본 논문에서는 변형된 유클리드 알고리즘을 이용한 하나의 (208,192,17) RS 복호기를 사용하여 DVD에 적용되는 (208,192,17) Shortened RS 부호와 (182,172,11) Shortened RS 부호의 복호가 모두 이루어 질 수 있도록 설계한다.

Ⅲ. 제안하는 마이크로아키텍처 구조 및 설계

본 장에서는 II장에서 기술한 변형된 유클리드 알고리즘을 이용하여 GF(2<sup>8</sup>)상의 RS 부호의 복호기를 설계하였다. 특히, 변형된 유클리드 알고리즘을 이용하여 오류 위치 다항식과 오류 평가 다항식을 추출하는 블

록에서는 곱셈기의 수를 줄임과 동시에, 쉬프트 레지스터를 사용함으로써 다항식 연산에 있어서 레지스터를 최대한 공유하여 사용할 수 있도록 효율적인 하드웨어 구조를 제안하여 면적을 최소화하는데 중점을 두었다. 또한, 그림 2와 같이 DVD에 적용하기 위하여 하나의 RS 복호기를 이용하여 (208,192,17) RS 부호와 (182,172,11) RS 부호를 모두 복호 할 수 있도록 설계하였다.

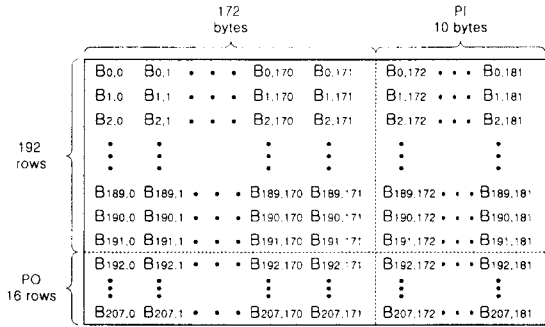


그림 2. DVD 시스템의 데이터 형식  
Fig. 2. Data format for DVD system.

1. Syndrome 계산 블록

본 논문의 (208,192,17) RS 복호기에서는 모두 16개의 신드롬이 생성되며, (182,172,11) RS 복호기에서는 모두 10개의 신드롬이 생성된다.

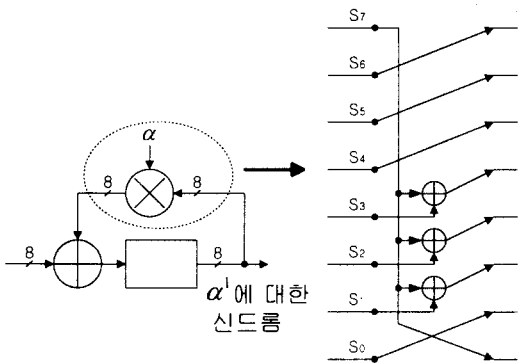


그림 3.  $\alpha^1$ 에 대한 신드롬 계산회로  
Fig. 3. Syndrome calculation circuit for  $\alpha^1$ .

신드롬 계산 회로에서는 곱셈회로와 덧셈회로, 그리고 8 비트 레지스터를 각각 하나씩을 필요로 한다. 일반적으로  $GF(2^8)$ 상의 곱셈기는 차지하는 면적이 클 뿐만 아니라, 신호의 지연에 의한 복호기의 동작 속도에도 큰 영향을 미친다. 그러나, 신드롬계산 블록에서

의 곱셈회로는  $GF(2^8)$ 상의 일반적인 곱셈기와는 달리 상수 값의 곱셈기가 필요로 하기 때문에  $\alpha$ 를 곱하는 곱셈회로는 단지 XOR 게이트 3 개로 구현 될 수 있다.

나머지  $\alpha^2 \sim \alpha^{15}$ 에 대한 신드롬 계산 회로도 그림 3과 같이 곱셈기를 간략화하여 설계할 수 있다.

2. 변형된 유클리드 알고리즘을 이용한 오류 위치 다항식( $\sigma(x)$ )과 오류 평가 다항식( $\omega(x)$ )의 추출 블록

본 논문에서는 변형된 유클리드 알고리즘을 이용하여 다항식 연산의 레지스터 공유를 극대화 시켜 모두 18개의 레지스터 쌍을 이루어 연산을 수행하였으며, 연산 코어의 공유를 통한 1개의 유클리드 코어를 이용하여 18번의 변형된 유클리드 연산을 수행하였다. 이러한 레지스터 및 연산 코어의 공유로 인해 [9]에서 제안된 아키텍처에서의 유클리드 모듈에서 사용되었던 Inverse ROM을 제거 할 수 있었으며, 곱셈기의 수도 줄일 수 있었다.

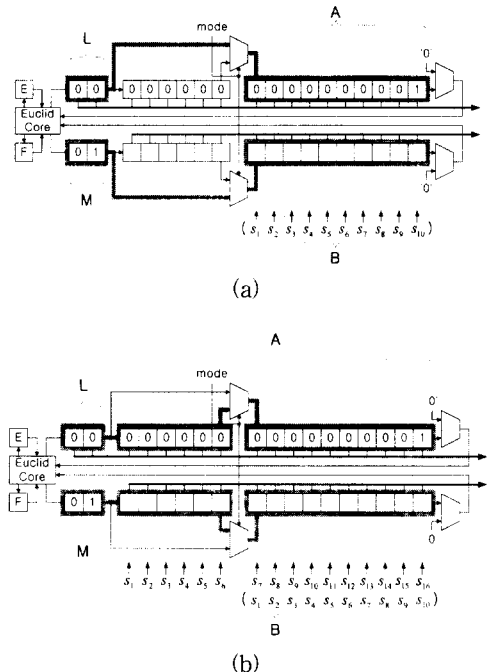


그림 4. 변형된 Euclid알고리즘을 이용한 에러위치다항식 및 에러평가다항식의 추출 회로 (a) (182, 172, 11) (b) (208, 192, 17) RS 디코더  
Fig. 4. Presumption block of error location polynomial( $\sigma(x)$ ) and error evaluation polynomial( $\omega(x)$ ) using modified Euclid algorithm for (a) (182,172,11), (b) (208,192,17)RS decoder

II장의 변형된 유클리드 알고리즘의 계산과정에서 A, B, L, M이라는 4개의 다항식의 계수 값을 기억하는 레지스터가 필요함을 알 수 있다. 그리고 이러한 레지스터들의 계산과정을 살펴보면 서로의 공통점을 찾을 수 있다. 즉, A와 B 레지스터는 차수가 점점 감소하는 형태를 가지고, L과 M의 레지스터는 차수가 점점 증가하는 것을 알 수 있다. 또한 차수의 증감의 크기는 서로 같은 비율을 갖는다.

이 블록에 대해서 본 논문에서는 그림 4와 같이 18쌍의 쉬프트 레지스터와 1개의 유클리드 코어를 사용함으로써 다항식의 공통점에 따른 효율적인 레지스터와, 유클리드 코어 공유를 이용하여 연산을 행하였다. 또한, 심벌 클럭의 2배 빠른 클럭을 사용함으로써 delay RAM의 사용을 줄일 수 있었다.

그림 4에서 (a)의 블록은 (182,172,11) RS 부호의 복호시 사용되는 레지스터를 나타낸다. (182,172,11) RS 부호는 오류 정정 능력이 5이므로 이 블록에서는 모두 12쌍의 레지스터만 사용하면 된다. 따라서, mode신호의 값에 따라 A, B 레지스터의 가운데 6쌍은 사용되지 않는다.

그림 4에서 AL 레지스터와 BM 레지스터는 오른쪽이 MSB이고, 오른쪽으로 이동하는 쉬프트 레지스터를 사용하며, 유클리드 코어 블록의 입력 계수 값을 항상 같은 차수의 계수로 만들어주기 위하여 각 레지스터는 enable 신호를 이용하여 제어한다. 유클리드 코어 블록은 A와 B의 최상위 계수 값을 소거하는 기능을 하며 그 블록도는 그림 5과 같다.

이 회로의 동작 과정을 살펴보면 다음과 같다.

신드롬 생성 블록에서 신드롬이 구해지면 B 레지스터에는 초기적으로 신드롬의 값이 저장되고, 나머지 A 레지스터에는  $x^{16}$ 의 계수 값, L 레지스터에는 '0' 그리고 M 레지스터에는 '1'의 값이 각각 저장된다. 단, A 레지스터에는 처음 계산과정에서 B 레지스터와 차수를 맞출 필요가 있기 때문에 MSB에 '1'의 값을 저장하고 나머지는 '0'으로 초기화한다.

각 레지스터의 마지막까지 쉬프트 연산이 이루어지고 나면 제어 블록은 새로운 값을 저장하고 있는 A와 B 레지스터의 값으로부터 차수와  $\Delta$ 의 값을 계산한다. 이와 같은 반복 과정을 거쳐 최종적으로 A 레지스터의 차수가 오류 정정 능력  $t$  보다 작을 때는 각 레지스터의 쉬프트가 이루어지지 않도록 하며, 그 때의 AL 레지스터 값을 다음절에서 기술 될 Chien

Search 블록과 Forney 알고리즘 블록으로 넘겨준다.

이 때 A 레지스터의 상위 레지스터 값은  $\omega(x)$ 를 의미하고, A 레지스터의 하위 레지스터와 L 레지스터의 값은  $\alpha(x)$ 를 의미한다.

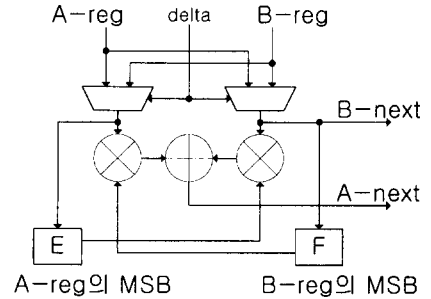


그림 5. Euclid 핵심 연산 블록

Fig. 5. Euclid core block.

(182,172,11) RS 부호의 복호시 연산이 종료된 후의  $\omega(x)$ 의 값은 그림 4(a)의 빗금 친 부분의 A 레지스터 상위 5개에 표현되고,  $\alpha(x)$ 의 값은 A 레지스터 하위 4개와 L 레지스터에 의해 표현되며, (208, 192, 17) RS 부호의 복호시에는  $\omega(x)$ 의 값은 그림 4(b)의 빗금 친 부분의 A 레지스터 상위 8개에,  $\alpha(x)$ 의 값은 A 레지스터의 하위 7개와 L 레지스터에 의해 표현된다.

그림 4와 같은 구조는 유클리드 알고리즘을 이용한 다항식 연산에 있어서 하나의 유클리드 코어를 공유할 수 있다는 장점을 가지고 있다. 따라서, 이 블록에서 오류 위치 다항식 및 오류 평가 다항식을 추출하기 위해서 필요한 곱셈기의 수는 2개이다.

### 3. 오류 위치 및 오류 값의 결정 블록

이 블록은 그림 6과 같이 4개의 다항식의 합을 구하는 블록과 GF(2<sup>8</sup>)의 모든 원소를 생성시키는 블록 그리고, 오류 위치와 오류 값을 결정하는 블록으로 이루어져 있다. 이때 4개의 다항식의 합을 구하는 블록에 대해서 본 논문에서는 RAM의 방식으로 레지터를 이용하여 구현하였다.

일반적으로 타이밍의 안정 조건이 만족이 되면 레지스터보다 레치의 크기가 더 작기 때문에 본 논문에서는 레치를 이용하였으며, 사용되는 클럭의 속도를 심벌 클럭의 4배 빠른 클럭으로 사용함으로써 반복구조의 연산을 통한 하드웨어의 공유와 delay RAM의 사용을 줄일 수 있었다.

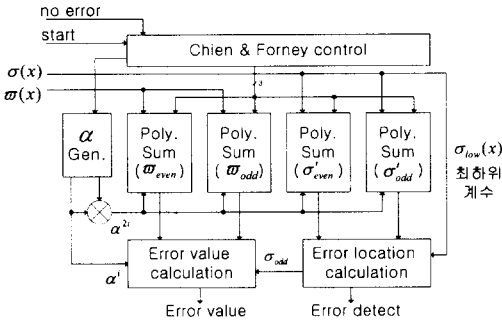


그림 6. 에러위치 및 에러 값의 결정 블록도  
Fig. 6. Decision block of error location and error value.

그림 6에서 오류의 위치와 오류 값을 구하기 위하여  $\alpha(\alpha^i)$ 와  $w(\alpha^i)$ 의 연산이 필요하다. 이 두 연산은 그림 7과 같은 구조를 갖는 각 다항식의 짝수 항과 홀수 항으로 나누어서 이루어진다.

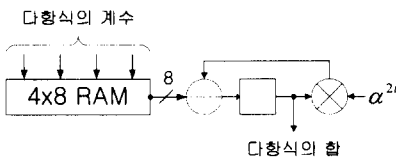


그림 7. 다항식 연산의 합 블록  
Fig. 7. Polynomial summation block.

각각의  $w'_{even}(x)$ ,  $w'_{odd}(x)$ ,  $\sigma'_{even}(x)$ ,  $\sigma'_{odd}(x)$ 의 계수 값은 그림 7과 같은 다항식의 합 블록으로 입력된다. 각 블록에서 각 다항식의 합이 구해지면 그림 8과 같이 오류의 위치와 값을 결정하는 블록으로 전달된다.

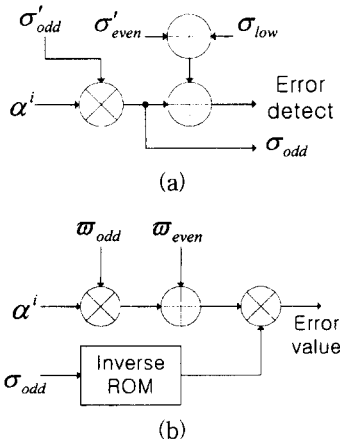


그림 8. (a) 에러 위치 결정 회로 (b) 에러 값 결정 회로  
Fig. 8. (a) Decision of error location, (b) Decision of error value.

오류의 위치를 결정하는 블록에서는  $\sigma'_{even}(x)$ ,  $\sigma'_{odd}(x)x$ ,  $\sigma_{low}(x)$ 의 합을 구한다. 결과의 값이 '0'이면 이때의  $x$ 에 해당하는  $\alpha^i$ 의 값의 역수가 오류의 위치가 된다. 또한, 오류의 값을 결정하는 블록에서는  $w'_{even}(x)$ ,  $w'_{odd}(x)x$ 의 합과 오류 위치 다항식에서의 홀수 차수항 계수의 합인  $\sigma_{odd}$ 의 역수를 곱한다. 이에 대한 결과 값은 오류의 위치가  $\alpha^{-i}$ 일 때의 오류 값을 나타낸다.

4. 오류의 정정

오류의 위치와 값이 결정이 되면 delay RAM을 거쳐서 나오는 지연된 수신 신호와 XOR 연산을 하여 출력한다. 즉, 그림 9의 에러 검출 신호가 '0'이면 출력 값은 'error value' 신호의 값과 delay RAM의 출력 값과 XOR 연산이 이루어지고, '1'이면 delay RAM 값을 그대로 출력한다.

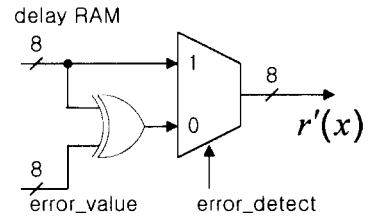


그림 9. 에러 정정 블록  
Fig. 9. Error correction block.

IV. 실험 및 고찰

본 논문에서 설계된 RS 복호기는 2가지 방법으로 하드웨어 검증은 시도하였는데, 첫째 방법은 VHDL 모델링을 이용한 FPGA의 구현 및 검증이며, 둘째 방법은 0.6 $\mu$ m LG CMOS TLM Compass 라이브러리를 이용한 Layout 및 검증이다. 한편, RS 복호기의 기능 검증을 위한 테스트 벡터는 RS 부호기를 C-언어로 프로그래밍 하여 추출하였다.

1. FPGA 구현 및 검증

선택된 알고리즘을 기반으로 복호기의 각 블록에 해당하는 마이크로 아키텍처를 추출한다. 추출된 마이크로 아키텍처로부터 VHSIC Hardware Description Language (VHDL)을 이용하여 모델링을 하였으며, VHDL로 설계된 RS 복호기는 Modeltech의 시뮬레이터를 사용하여 기능적인 동작을 검증하였다. 검증이 완료된 VHDL 모델링을 합성 툴인 Synopsys를 사

용하여 ALTERA 라이브러리 FLEX10K로 회로를 합성하였다. 합성된 회로로부터 .edif 파일을 생성하여 MAXPLUS II로 컴파일을 한 후, FPGA에 프로그래밍을 하여 하드웨어 동작을 검증하였다.

(208,192,17) RS 복호기의 기능 검증을 위해서 그림 10에서와 같은 입력 데이터가 사용되었다. 그림 10은 입력 데이터의 2~9번째 심벌에 “31, 32, 33, 34, 35, 36, 37, 38”의 에러가 발생하여 원래의 정보 “31, 32, 33, 34, 35, 36, 37, 38”이 모두 “00”으로 수신되었음을 나타낸다.

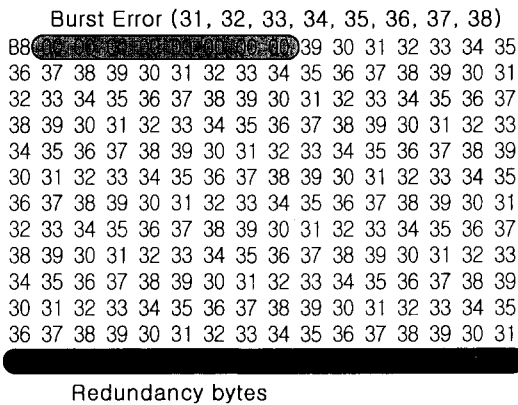


그림 10. (208, 192, 17) RS 디코더의 입력 데이터  
Fig. 10. Input data of (208, 192, 17) RS decoder.

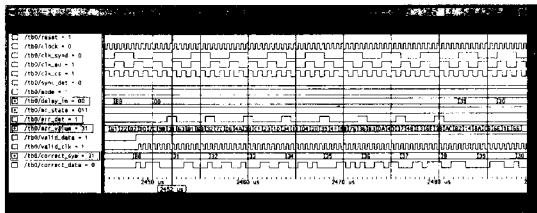


그림 11.(208, 192, 17)RS 디코더에 대한 시뮬레이션 파형  
Fig. 11.Simulation for (208, 192, 17)RS decoder.

그림 11에서 'delay\_in'의 신호는 그림 10과 같이 에러가 발생한 정보데이터가 입력된다. 그리고, 'err\_det'신호는 각 입력 데이터에 따라 에러의 발생 여부를 알려준다. 따라서 'err\_det'의 신호가 '1'의 값을 가질 때 최종적으로 에러가 정정된 정보신호인 'correct\_sym'의 값은 'delay\_in'과 'err\_value'의 값이 더해져서 출력된다. 그림 11에서 에러가 발생한 심벌에 대해 정확히 에러를 정정하고 있음을 알 수 있다.

본 논문에서 설계된 RS 복호기의 FPGA 구현 및

검증을 위하여 그림 12와 같은 test board를 구성하였다.

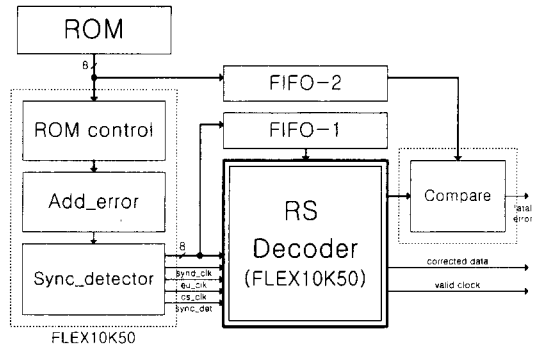


그림 12. RS 디코더의 FPGA 테스트 보드에 대한 블록도  
Fig. 12. Block diagram of FPGA test board for RS decoder.

RS 복호기를 위해 로직셀의 이용율이 86%인 한 개의 FLEX10K50 FPGA가 사용되었으며, 입력 데이터를 받기 위한 ROM 컨트롤 블록, 에러를 추가하는 블록, 수신된 데이터의 동기를 맞추는 블록과 복호 후의 데이터가 올바르게 복호되었는지를 비교하는 블록이 또 다른 하나의 FLEX10K50 FPGA로 구현되었다.

2. VLSI 구현 및 검증

VLSI 구현을 위하여 그림 13과 같이 Synopsys와 Compass 툴을 이용하여 0.6 $\mu$ m 3V LG CMOS TLM Compass 라이브러리로 합성 및 Auto P&R을 행하였다. 이러한, Layout으로부터 SDF(Standard Delay File)을 추출하고, 이를 Verilog-XL을 이용하여 타이밍 시뮬레이션을 통한 기능 검증을 행하였다.

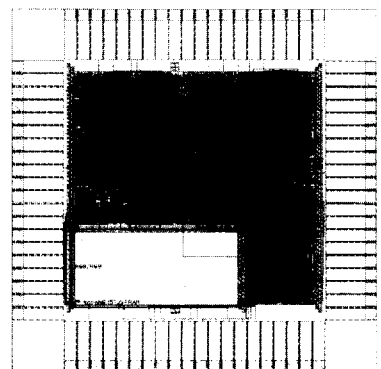


그림 13. LG 라이브러리를 이용한 RS 디코더의 Layout  
Fig. 13. Layout of RS decoder using LG library.

LG 라이브러리를 이용한 VLSI의 전체 칩 면적은  $3.49 \times 3.52$  ( $12.31\text{mm}^2$ ) 이며, dealy RAM을 포함한 RS 복호기의 코어 면적은  $2.299 \times 2.284$  ( $5.25 \text{mm}^2$ )으로 구현이 되었고, SDF를 이용한 타이밍 시뮬레이션으로부터 20 MHz까지 동작함을 확인하였다.

그리고, 본 논문에서 제시한 구조에 대한 우수성을 보이고자 칩 면적 및 게이트 수, 동작 속도의 관점에서 표 1과 같이 DVD에 응용하기 위하여 제시된 [5]와 비교하여 우수함을 확인하였다.

표 1. DVD RS 디코더에 대한 [5]와의 성능 비교

Table 1. Complexity comparison for DVD RS decoder with reference [5].

	제안된 RS 복호기	[5]
적용된 알고리즘	변형된 유클리드 알고리즘	Berlekamp-Massey
(n,k,d)	(208,192,17) & (182,172,11)	(208,192,17) & (182,172,11)
응용분야	DVD	DVD
라이브러리	0.6 $\mu\text{m}$ LG CMOS TLM	0.6 $\mu\text{m}$ SPDM CMOS
게이트 수	17,071	21,190
코어면적	$2.299 \times 2.284$ ( $5.25 \text{mm}^2$ )	$2.70 \times 2.39$ ( $6.45 \text{mm}^2$ )
동작속도	20 MHz	16.67 MHz

또한, [3]에서는 변형된 유클리드 알고리즘을 사용하고, [4]에서는 Berlekamp-Massey 알고리즘을 이용하여 DVD에 사용되는 RS 복호기의 구조를 각각 제안하였는데 각각 약35,000 게이트 및 약30,000 게이트로 발표되어 본 논문이 보다 우수함을 알 수 있다.

한편, [6]과의 비교를 위하여 본 논문을 위성 방송 수신기 및 LMDS 블럭에 포함되는 (204,188,17) RS 부호를 복호할 수 있도록 수정하고, [6]과 동일한 라이브러리를 이용하여 합성하였는데 표 2와 같이 [6]에서 제시된 것 보다 우수함을 알 수 있다.

표 2. Euclid 알고리즘에 대한 [6]과의 성능 비교

Table 2. Complexity comparison with reference [6].

	제안된 RS 복호기	[6]
적용된 알고리즘	변형된 유클리드 알고리즘	변형된 유클리드 알고리즘
(n,k,d)	(204,188,17)	(204,188,17)
응용분야	LMDS	위성방송수신기
라이브러리	0.5 $\mu\text{m}$ 현대 Simbios	0.5 $\mu\text{m}$ CMOS TLM
게이트 수	12,661	약 14,000

## V. 결 론

본 논문에서는 저장 매체인 DVD에서의 오류 정정을 위하여 사용되는 RS 복호기를 변형된 유클리드 알고리즘을 이용하면서 면적의 관점에서 효율적인 구조를 갖도록 제안하였고, 이를 VLSI로 구현하였다.

오류 위치 다항식과 오류 평가 다항식을 추출하기 위하여 두 다항식의 최고차 항 계수를 소거하여 최대 공약수를 구하는 반복적인 연산구조의 변형된 유클리드 알고리즘을 이용하였다. 이 알고리즘으로부터 계수 값의 소거 기능을 하는 유클리드 코어에 대해서 기존의 18개 유클리드 코어의 사용을 1개의 유클리드 코어만을 이용하여 서로 공유하도록 설계하였다. 또한, 다항식의 연산 과정에 있어서 오류 위치 다항식과 오류 평가 다항식의 차수가 비례적으로 증감하는 형태를 이용하여 연산과정에 있어서 임시적으로 사용하는 레지스터를 서로 공유하도록 설계하였으며, 그 결과 두 다항식의 계수 값을 저장하는데 필요한 24쌍의 레지스터를 18쌍만으로 저장하여 레지스터의 사용을 줄일 수 있었다.

제안한 RS 복호기를 VHDL로 표현하고 FPGA로 시험 제작하였으며, LG 0.6 $\mu\text{m}$  3.3V 라이브러리를 이용하여 코어 면적이  $2.299 \times 2.284$  ( $5.25 \text{mm}^2$ )이면서 20MHz 까지 동작하는 칩을 VLSI로 구현하였다.

본 논문에서 제안된 RS 복호기에 대한 향후의 연구 과제로는 GF(2<sup>8</sup>)상에서의 곱셈기와 Forney 알고리즘에 사용되는 역원 추출회로에 대한 보다 효율적인 개선이 요구된다.

## 참 고 문 헌

- [1] 陳年綱, *아날로그와 디지털 통신*, 淸文閣, pp. 593-638, 1996
- [2] Nicolas Ibrahim and Ghassan Kawas Kaleh, "A Turbo Coding Scheme for Spread-spectrum Signalling," *Proceedings of the International Conference on Telecommunications, vol. 1, pp. 50-54, 1998.*
- [3] GoangSeog Choi, HoonSoon Choi, and YoungHwan Kim, "RS Decoder using Modified Euclidean Algorithm for DVD/CD," *ISCPAT Conference Pro-*



- ceedings, vol. 1, pp. 208-212, 1997.*
- [ 4 ] JooSeon Kim, ByungGook Chung, YoungHwan Kim, and Kiwon Lee, "A High Speed RS Decoder using Berlekamp-Massey Algorithm for DVD/CD," *ISCPAT Conference Proceedings, vol. 2, pp. 1430-1434, 1998.*
- [ 5 ] Hsie-Chia Chang, C. Bernard Shung, "A (208,192,8) REED-SOLOMON DECODER FOR DVD APPLICATION," *IEEE International Conference on Communications, vol. 2, pp. 957-960, 1998.*
- [ 6 ] Seung-Jun Lee, Jong-Seob Baek, and Young-Shig Choi, "A Single Chip DVB Receiver for Variable Rate QPSK Demodulation and Forward Error Correction," *ASIC DESIGN WORKSHOP 신진 박사 논문 발표대회 및 공개토론회 논문집, pp. 47-50, 1997*
- [ 7 ] 이 만 영, *BCH 부호와 Reed-Solomon 부호*, 민음사, 1990
- [ 8 ] Stephen B. Wicker, *Error Control Systems for Digital Communication and Storage*, Prentice Hall, 1995.
- [ 9 ] Po Tong, "A 40-MHz ENCODER/DECODER CHIP GENERATED BY A REED-SOLOMON CODE COMPILER," *IEEE Custom Integrated Circuits Conference, pp. 13.5.1-4, 1990.*
- [ 10 ] HOWARD M. SHAO and T. K. TRUONG, "A VLSI Design of a Pipeline Reed-Solomon Decoder," *IEEE Transaction on Computer, vol. C-34, no. 5, pp. 393-402, 1985.*
- [ 11 ] HOWARD M. SHAO and IRVING S. REED, "On the VLSI Design of a Pipeline Reed-Solomon Decoder Using Systolic Arrays," *IEEE Transaction on Computer, vol. 37, no. 10, pp. 1273-1280, 1988.*

---

 저 자 소 개
 

---



李柱泰(正會員)

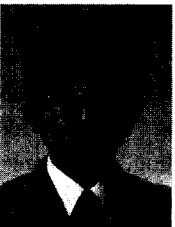
1997년 2월 인천대학교 전자공학과 공학사. 1999년 2월 인천대학교 대학원 전자공학과 공학석사. 1999년 2월 ~ 현재 (주)기라정보통신 연구원. 주관심분야는 VHDL을 이용한 ASIC 설계, 이동 통신용 ASIC 설계 등



李昇(正會員)

1993년 2월 인천대학교 전자공학과 공학사. 1996년 2월 인천대학교 대학원 전자공학과 공학석사. 1996년 10월 ~ 현재 (주)주흥정보통신 연구원, 1999년 2월 ~ 현재 인천대학교 전자공학과 박사 과정. 주관심분야는

VHDL을 이용한 ASIC 설계, 이동 통신용 ASIC 설계 등



趙仲彙(正會員)

1981년 2월 한양대학교 전자공학과 공학사. 1983년 2월 한양대학교 대학원 전자공학과 공학석사. 1986년 8월 한양대학교 대학원 전자공학과 공학박사. 1989년 8월 ~ 1990년 7월 미국 Univ. of California(Irvine)

Post Doctor. 1986년 9월 ~ 현재 인천대학교 전자공학과 교수. 주관심분야는 VHDL을 이용한 ASIC 설계, 이동 멀티미디어용 ASIC 설계 등