논문 2008-45SD-6-14

새로운 저전력 및 저면적 리드-솔로몬 복호기

(New Low-Power and Small-Area Reed-Solomon Decoder)

백 재 현*, 선우 명 훈**

(Jae Hyun Baek and Myung Hoon Sunwoo)

요 약

본 논문에서는 새로운 저전력 및 저면적 리드-솔로몬 (Reed-Solomon) 복호기를 제안한다. 제안하는 리드-솔로몬 복호기는 새로운 단순화된 수정 유클리드 알고리즘을 사용하여 낮은 하드웨어 복잡도 및 저전력 리드-솔로몬 복호가 가능하다. 새로운 단순화된 수정 유클리드 알고리즘은 하드웨어 복잡도를 줄이기 위해서 새로운 초기 조건 및 다항식 연산 방식을 사용한다. 따라서 $3t$개의 기본 셀로 구성된 새로운 단순화된 수정 유클리드 구조는 기존 수정 유클리드 구조는 물론 베르캠프-메세이 구조들에 비해 가장 낮은 하드웨어 복잡도를 갖는다. 0.18㎛ 삼성 라이브러리를 사용하여 논리합성을 수행한 리드-솔로몬 복호기는 370MHz의 동작 주파수 및 2..9Gbps의 데이터 처리 속도를 갖는다. (255, 239, 8) 리드-솔로몬 코드 복호를 수행하는 단순화된 수정 유클리드 구조와 전체 리드-솔로몬 복호기의 게이트 수는 각각 20,166개와 40,136개이다. 따라서 구현한 리드-솔로몬 복호기는 기존 DCME 복호기에 비해 5%의 게이트 수 절감 효과를 갖는다.

Abstract

This paper proposes a new low-power and small-area Reed-Solomon decoder. The proposed Reed-Solomon decoder using a novel simplified form of the modified Euclid's algorithm can support low-hardware complexity and low-power consumption for Reed-Solomon decoding. The simplified modified Euclid's algorithm uses new initial conditions and polynomial computations to reduce hardware complexity, and thus, the implemented architecture consisting of $3t$ basic cells has the lowest hardware complexity compared with existing modified Euclid's and Berlekamp-Massey architectures. The Reed-Solomon decoder has been synthesized using the 0.18㎛ Samsung standard cell library and operates at 370MHz and its data rate supports up to 2.9Gbps. For the (255, 239, 8) RS code, the gate counts of the simplified modified Euclid's architecture and the whole decoder excluding FIFO memory are only 20,166 and 40,136, respectively. Therefore, the proposed decoder can reduce the total gate count at least 5% compared with the conventional DCME decoder.

Keywords : Reed-Solomon, Key equation, Modified Euclid's algorithm, Error control codes

Ⅰ. 서 론

고속 데이터 통신 및 휴대 인터넷 서비스의 수요 증가로 인해 UWB (Ultra Wideband)[1], WiBro (Wireless Broadband)[2], DVB-RCS (Digital Video Broadcasting-Return Channel via Satellite)[3], 차세대 무선 랜 통신 표준 (IEEE 802.11n) 등 다양한 통신 표준이 개발되고 있다. 통신 시스템의 다양한 구성 요소 가운데 오류 정정 부호는 통신 표준에 관계없이 사용되는 필수 요소로서 실시간 데이터 복호 및 낮은 하드웨어 복잡도를 갖는 고속 통신 시스템을 위한 오류 정정 부호의 연구는 매우 중요하며 현재 다양한 연구가 진행되고 있다.

전송된 데이터의 오류 수정을 위해서 사용되는 오류 정정 부호는 컨볼루션 (Convolutional) 부호와 블록 (Block) 부호로 구분되며, 리드-솔로몬 (Reed- Solomon) 부호, 비터비 (Viterbi) 부호, 터보 (Turbo) 부호, BCH (Bose-Chaudhuri- Hocquenheim) 부호, LDPC (Low-

* 정회원, 서울대학교 BK21 정보기술사업단
(Seoul National University)
** 정회원, 아주대학교 전자공학부
(Ajou University)
※ 본 연구는 교육과학기술부 2단계 BK (Brain Korea) 21 과제, 반도체설계교육센터 (IDEC)의 지원을 받아서 이루어졌습니다.
접수일자: 2008년2월21일, 수정완료일: 2008년5월26일

Density Parity-Check) 부호 등이 사용된다. 이와 같은 다양한 오류 정정 부호 중 리드-솔로몬 부호는 랜덤 오류뿐만 아니라 연집 오류에 우수한 정정 능력을 갖고 있어 통신 시스템 및 디지털 저장 장치에 폭넓게 사용되고 있다.

일반적인 (n, k, t) 리드-솔로몬 부호에서 k는 m 비트 정보 심벌의 개수를 나타내며 리드-솔로몬 부호화 후 $n = 2^m - 1$개의 심벌을 갖는다. $t\ (= (n-k)/2)$는 리드-솔로몬 부호의 오류 정정 능력을 나타낸다. 리드-솔로몬 복호기는 신드롬 (syndrome) 연산, 키 방정식 연산, Chien search 알고리즘, Forney 알고리즘, 오류 정정의 5개 블록으로 구성되며, 이들 블록 중 오류 위치 다항식 (error locator polynomial) 및 오류 크기 다항식 (error value polynomial) 연산을 위한 키 방정식 블록은 가장 큰 하드웨어 복잡도 및 연산 시간을 필요로 하여 다양한 구조의 키 방정식 연산 회로에 대한 연구가 진행되고 있다. 베르캠프-메세이 (Berlekamp-Massey) 알고리즘,[4~8] 유클리드 (Euclid) 알고리즘,[7~8] 수정 유클리드 (modified Euclid) 알고리즘[9~12]가 키 방정식 연산을 위해 사용된다.

수정 유클리드 알고리즘은 베르캠프-메세이 알고리즘에 비해 높은 하드웨어 복잡도를 갖는 반면 오류정정능력 t에 관계없이 일정한 임계경로를 가지므로 고속의 키 방정식 연산이 가능하였다. 반면에 베르캠프-메세이 알고리즘은 낮은 하드웨어 복잡도를 갖는 대신 오류정정능력 t에 따라 임계경로가 증가하는 단점을 갖는다. 그러나 2001년 베르캠프-메세이 알고리즘을 기반으로 오류정정능력에 관계없이 일정한 임계경로를 갖는 것은 물론 수정 유클리드 알고리즘에 비해서도 낮은 하드웨어 복잡도를 갖는 RiBM (Reformulated inversionless Berlekamp-Massy)[8] 알고리즘이 개발되었으며, 현존하는 가장 우수한 키 방정식 연산 알고리즘이라 할 수 있다. 따라서 본 저자는 2006년 및 2007년에 다항식의 차수 연산을 제거한 수정 유클리드 (DCME) 알고리즘 및 이를 개선한 E-DCME 알고리즘의 하드웨어 구조를 각각 제안한바 있다.

DCME[15] 및 E-DCME 알고리즘[16]은 시프트 연산을 통해서 다항식의 차수가 항상 동일해진다는 가정을 기존 다항식 연산 방법에 적용 기존 수정 유클리드 알고리즘에 비해 단순한 키 방정식 연산이 가능하다. 그러나 DCME[15] 및 E-DCME 알고리즘[16]은 하드웨어 구현시 RiBM 알고리즘에 비해 여전히 높은 하드웨어 복잡도를 갖는다. 따라서 본 논문에서는 RiBM 알고리즘에 비해 낮은 하드웨어 복잡도를 갖는 새로운 단순화된 수정 유클리드 알고리즘을 사용하여 저전력 및 저면적 리드-솔로몬 복호기를 제안한다. 새로운 단순화된 수정 유클리드 알고리즘[17]은 기존 수정 유클리드 알고리즘을 하드웨어 구현시 복잡도가 높다는 단점을 개선하기 위해서 새로운 초기 조건 및 다항식 연산 방식을 사용한다.

$3t$ 개의 기본 셀로 구성된 단순화된 수정 유클리드 구조는 전체 $6t$개 곱셈기, $6t+2$개 레지스터, $3t$개 덧셈기, $2t-1$개 멀티플렉서로 구성되는 반면 $3t+1$개의 기본 셀로 구성된 RiBM 구조는 $6t+2$개 곱셈기, $6t+2$개 레지스터, $3t+1$개 덧셈기, $3t+1$개 멀티플렉서를 필요로 한다. 따라서 제안하는 단순화된 수정 유클리드 구조는 2개 곱셈기, 1개 덧셈기, $t+2$개 멀티플렉서를 절약할 수 있다. 삼성 0.18㎛ 셀 라이브러리를 사용하여 합성한 결과 단순화된 수정 유클리드 구조의 전체 게이트 수는 20,166개이며 메모리를 제외한 전체 리드-솔로몬 복호기는 40,136개 게이트로 구성된다.

본 논문의 구성은 다음과 같다. Ⅱ장에서는 기존 키 방정식 연산기들을 소개한다. Ⅲ장에서는 제안하는 단순화된 수정 유클리드 키 방정식 연산기를 사용하는 저면적 리드-솔로몬 복호기를 소개하고 Ⅳ장에서 구현 결과 및 성능 비교를 보인다. 마지막으로 Ⅴ장에서 결론을 맺는다.

Ⅱ. 기존 키 방정식 알고리즘

1. 수정 유클리드 알고리즘

그림 1은 수정 유클리드 알고리즘[11]의 연산 흐름을 나타낸다. 수정 유클리드 알고리즘은 $R_i(x)$, $Q_i(x)$, $\lambda_i(x)$, $\mu_i(x)$, $deg(R_i(x))$, $deg(Q_i(x))$을 입력받아 다음과 같은 키 방정식 연산을 수행한다.

식 (1)과 (2)는 그림 2의 수정 유클리드 알고리즘을 사용하여 키 방정식 연산을 위한 초기 조건 (initial condition)을 나타낸다.

$$R_0(x) = x^{2t},\ Q_0(x) = S(x) = \sum_{i=0}^{2t-1} S_i x^i \qquad (1)$$

$$\lambda_0(x) = 0,\ \mu_0(x) = 1 \qquad (2)$$

수정 유클리드 알고리즘은 식 (3)~(6)의 다항식 연산을 반복적으로 수행한다.

$$R_{i+1}(x) = [\sigma_i b_i R_i(x) + \overline{\sigma_i} a_i Q_i(x)] - x^{|l_i|}[\sigma_i a_i Q_i(x) + \overline{\sigma_i} b_i R_i(x)] \quad (3)$$

$$\lambda_{i+1}(x) = [\sigma_i b_i \lambda_i(x) + \overline{\sigma_i} a_i \mu_i(x)] - x^{|l_i|}[\sigma_i a_i \mu_i(x) + \overline{\sigma_i} b_i \lambda_i(x)] \quad (4)$$

$$Q_{i+1}(x) = \sigma_i Q_i(x) + \overline{\sigma_i} R_i(x) \quad (5)$$

$$\mu_{i+1}(x) = \sigma_i \mu_i(x) + \overline{\sigma_i} \lambda_i(x) \quad (6)$$

a_i와 b_i는 각각 $R_i(x)$와 $Q_i(x)$의 최고차항 계수이고, i는 다항식 연산의 반복 횟수를 나타낸다. 식 (3)~(6)의 σ_i와 l_i는 식 (7)에 의해 주어진다.

$$l_i = deg(R_i(x)) - deg(Q_i(x)) \quad (7)$$
$$\sigma_i = 1 \quad \text{if} \quad l_i \geq 0$$
$$\sigma_i = 0 \quad \text{otherwise}$$

식 (7)의 $deg(R_i(x))$와 $deg(Q_i(x))$는 다항식 $R_i(x)$와 $Q_i(x)$의 차수를 나타낸다. 식 (3)~(6)의 다항식 연산은 $deg(R_i(x)) < t$를 만족할 때까지 반복적으로 수행된다. 연산이 종료되면 다항식 오류 크기 다항식 $R(x)$와 오류 위치 다항식 $\lambda(x)$를 얻는다.

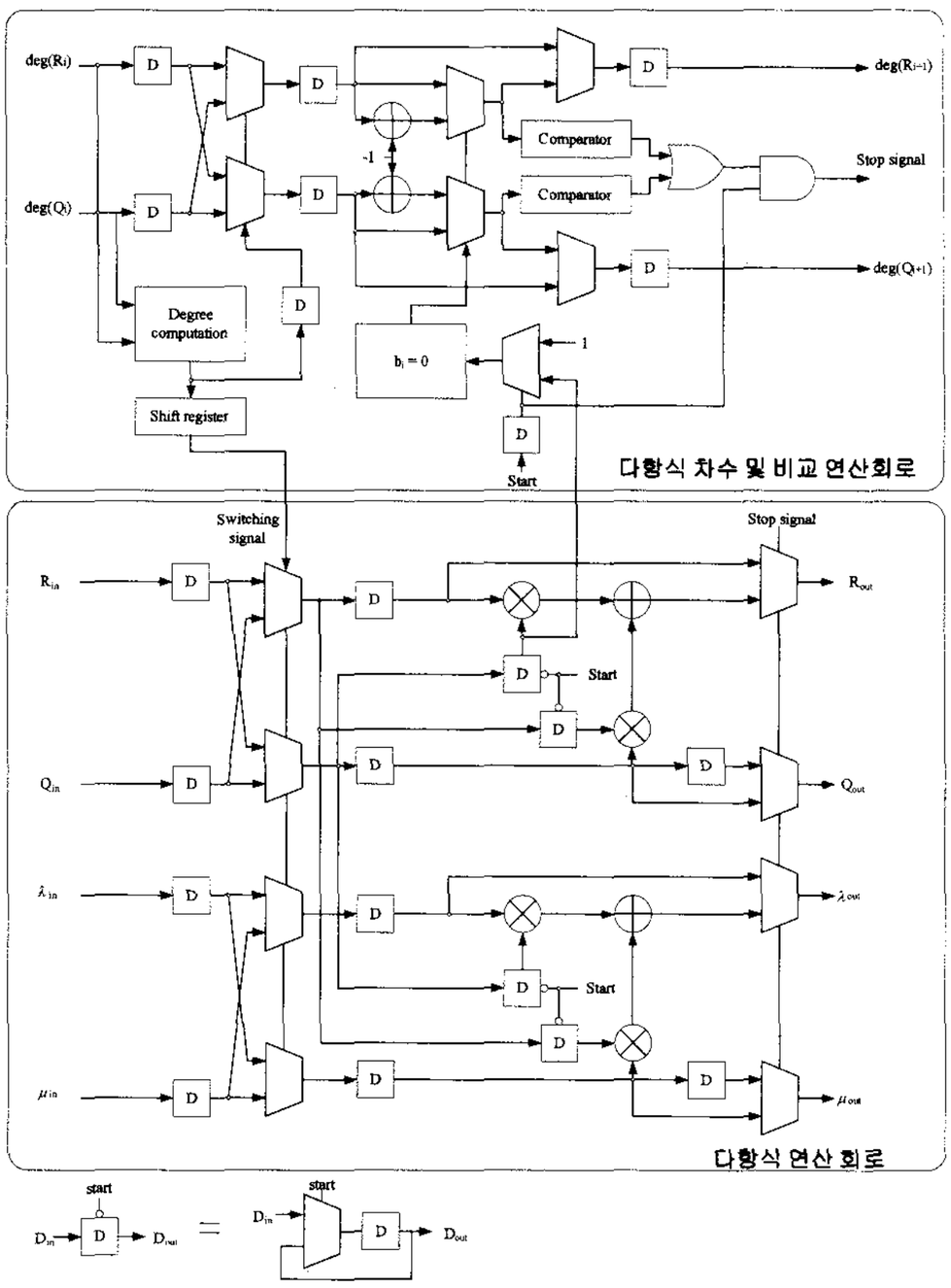


그림 2. 수정 유클리드 알고리즘 연산 회로.
Fig. 2. Basic cell for the modified Euclid's algorithm.

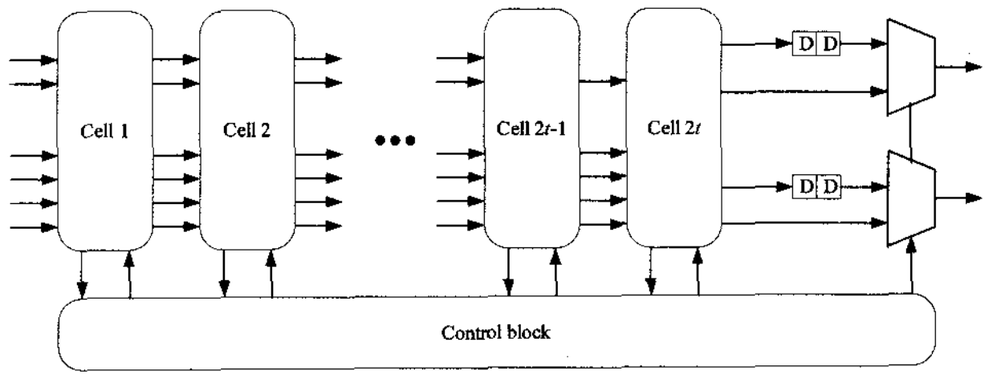


그림 3. 수정 유클리드 키 방정식 연산기
Fig. 3. Modified Euclid's algorithm architecture.

따라서 수정 유클리드 알고리즘은 하드웨어로 구현 시 그림 2에서 보인 것처럼 다항식 식 (7)의 다항식 차수 연산 및 비교 연산 회로가 필요하며 이로 인해 수정 유클리드 알고리즘 연산 회로의 하드웨어 복잡도 및 임계경로가 증가하게 된다. 그림 3은 $2t$개의 수정 유클리드 알고리즘을 사용하는 키 방정식 연산기의 전체 구조를 나타낸다.

2. RiBM 알고리즘

베르캠프-메세이 알고리즘은 크게 Discrepancy 연산 회로와 오류 위치 갱신 회로로 구성되며, 별개의 하드웨어 구조를 갖는다. 두 연산 회로의 임계 경로는 각각

Initialization:
$\delta_{3t}(0) = 1$; $\delta_i(0) = 0$; for $i = 2t, 2t+1, \ldots, 3t-1$.
$k(0) = 0$. $\gamma(0) = 1$.
Input : S_i, $i = 0, 1, \ldots, 2t-1$.
  $\delta_i(0) = \theta_i(0) = S_i$, $(i = 0, 1, \ldots, 2t-1)$
  for $r = 0$ step 1 until $2t$ -1 do
    begin
      Step RiBM.1 $\delta_i(r+1) = \gamma(r) \cdot \delta_{i+1}(r) - \delta_0(r) \cdot \theta_i(r)$,
      $(i = 0, 1, \ldots, 3t)$
      Step RiBM.2
        if $\delta_0(r) \neq 0$ and $k(r) \geq 0$ then
          begin
            $\theta_i(r+1) = \delta_{i+1}(r)$, $(i = 0, 1, \ldots, 3t)$
            $\gamma(r+1) = \delta_0(r)$
            $k(r+1) = -k(r) - 1$;
          end
        else
          begin
            $\theta_i(r+1) = \delta_i(r)$, $(i = 0, 1, \ldots, 3t)$
            $\gamma(r+1) = \gamma(r)$
            $k(r+1) = k(r) + 1$;
          end
    end
Output :
  $\lambda_i(2t) = \delta_{t+i}(2t)$, $(i = 0, 1, \ldots, t)$
  $R_i(2t) = \delta_i(2t)$, $(i = 0, 1, \ldots, t-1)$

그림 4. RiBM 알고리즘.
Fig. 4. RiBM algorithm.

$T_{Mul}+(1+\lceil log_2(t+1)\rceil)\cdot T_{add}+\lceil log_2 m\rceil\cdot T_{or}$와 $T_{Mul}+T_{add}$이다. 따라서 Discrepancy 연산 회로의 임계 경로가 오류 정정 능력 t에 가변적임을 알 수 있다.

이와 같이 오류 정정 능력에 따라 임계 경로가 증가하는 베르캠프-메세이 알고리즘의 단점을 개선하기 위해서 2001년에 RiBM 알고리즘이 제안되었다. 그림 4는 나눗셈 연산을 제거한 베르캠프-메세이 (inverse-free Berlekamp-Massey, iBM) 알고리즘의 성능을 개선한 RiBM 알고리즘을 나타낸다.

RiBM 알고리즘은 기존 iBM 알고리즘의 Discrepancy 연산 방식을 그림 4의 Step RiBM.1로 변경하였다. 그 결과 RiBM 알고리즘은 그림 5의 Processing Element (PE)를 사용하여 Discrepancy 연산 및 오류 위치 갱신을 수행할 수 있어 수정 유클리드 알고리즘과 같이 오류 정정 능력에 관계없이 규칙적인 하드웨어 구조를 사용하여 키 방정식 연산이 가능하다. 그러나 그림 5에서 보인 것처럼 2개의 곱셈기, 2개의 레지스터, 1개의 덧셈기, 1개의 멀티플렉서로 구성된 PE 회로는 그림 2의 수정 유클리드 알고리즘에 비해 매우 낮은 하드웨어 복잡도를 갖는다.

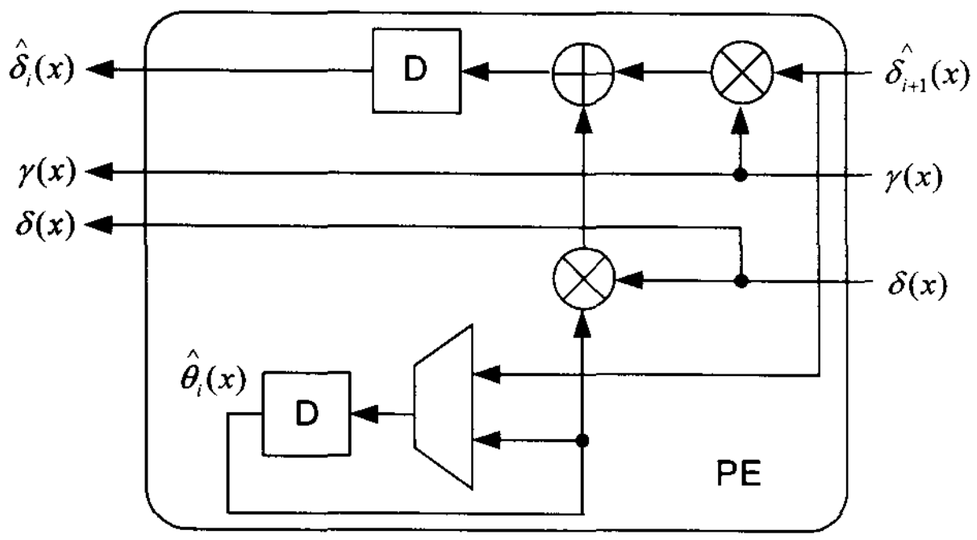


그림 5. RiBM 알고리즘 연산 회로.
Fig. 5. Processing Element for the RiBM algorithm.

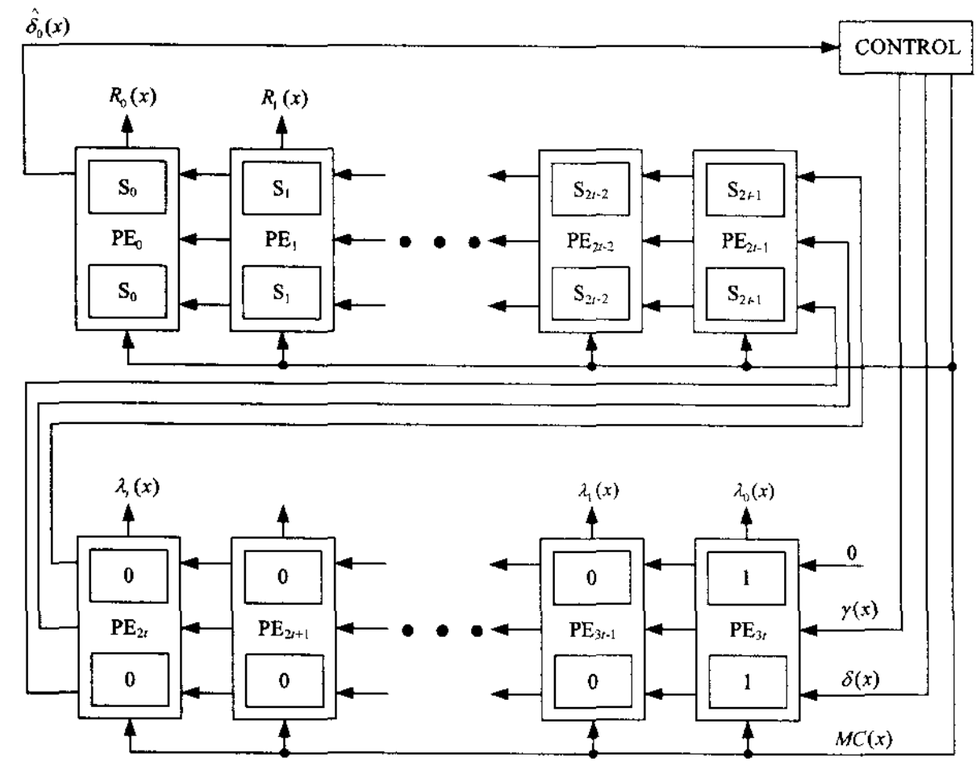


그림 6. RiBM 키 방정식 연산기.
Fig. 6. RiBM algorithm architecture.

그림 6은 $3t+1$개의 PE로 구성된 RiBM 키 방정식 연산기를 나타내며, 전체 $6t+2$개의 곱셈기, $6t+2$개의 레지스터, $3t+1$개의 덧셈기, $3t+1$개의 멀티플렉서로 구성된다. 또한, 그림 6에서 보인 것처럼 RiBM 구조는 기존 키 방정식 회로에 매우 단순한 데이터 패스를 갖는다. 그러나 그림 6에는 그림 5에서 보인 것처럼 각 PE 회로의 레지스터 및 멀티플렉서 제어 신호는 생략되어 있다.

Ⅲ. 새로운 저면적 리드-솔로몬 복호기

1. 구현한 전체 리드-솔로몬 복호기

복호기에 입력되는 수신 다항식 (received polynomial) $r(x)$는 식 (8)과 같다.

$$r(x)=c(x)+e(x) \tag{8}$$

$c(x)$와 $e(x)$는 각각 송신 다항식 (transmitted polynomial) 및 오류 다항식 (error polynomial)이다. $c(x)$는 정보 다항식 (message polynomial)과 생성 다항식에 의해서 식 (9)과 같이 표현된다.

$$\begin{aligned} c(x) &= x^{n-k}m(x)+[x^{n-k}m(x) \bmod g(x)] \\ &= q(x)g(x) \end{aligned} \tag{9}$$

$q(x)$는 몫 다항식 (quotient polynomial)이다. RS 복호기는 수신 다항식 $r(x)$로부터 $e(x)$을 찾고 오류를 정정한다.

그림 7은 제안하는 RS 복호기이다. 신드롬 연산 블록은 $g(x)$의 근을 사용하여 신드롬 다항식 $S(x)$을 구한다. 키 방정식 연산 블록은 $R(x)$와 $\lambda(x)$을 얻기 위해 다항식 연산을 수행한다. Chien Search 블록 및 Forney's 블록은 $R(x)$와 $\lambda(x)$을 사용하여 오류 위치 및 크기를 연산한다. FIFO 메모리는 RS 복호 시간동안 수신 다항식을 지연시킨다. RS 복호기는 FIFO 메모리의 출력 값에 오류를 더하여 수신된 오류를 정정한다.

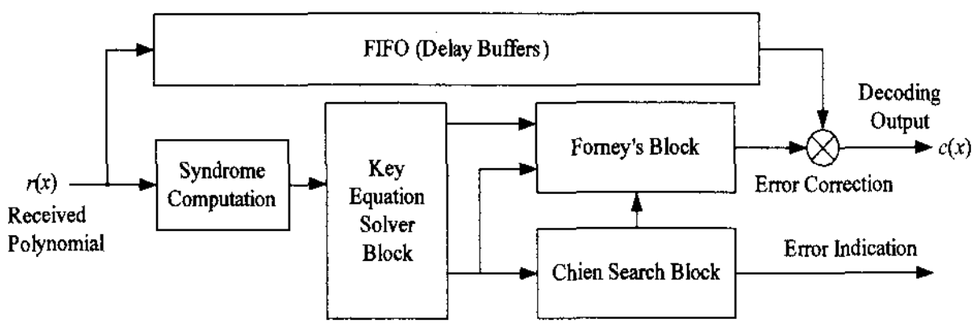


그림 7. 제안하는 저면적 리드-솔로몬 복호기
Fig. 7. Proposed small-area Reed-Solomon decoder.

2. 새로운 단순화된 수정 유클리드 알고리즘 연산기

기존 수정 유클리드 알고리즘은 다항식 차수 연산 및 비교 연산으로 인해 하드웨어 복잡도 및 임계경로가 증가하게 된다. 따라서 본 논문에서는 다항식 차수 연산 및 비교 연산을 하지 않는 단순화된 수정 유클리드 알고리즘[17]을 사용하는 새로운 키 방정식 연산 회로를 제안한다. 새로운 단순화된 수정 유클리드 알고리즘은 식 (10)과 (11)를 초기 조건으로 사용한다.

$$R_0(x) = S(x) = \sum_{i=0}^{2t-1} S_i x^i,\ Q_0(x) = x^{2t-1} \quad (10)$$

$$\lambda_0(x) = 1,\ \mu_0(x) = 0 \quad (11)$$

식 (1)과 (2)의 기존 초기 조건에서 최고 차수는 R 다항식의 $2t$인 반면 새로운 초기 조건은 R 다항식 및 Q 다항식 모두 최고 차수로 $2t-1$을 갖는다. 결과적으로 새로운 초기 조건은 기존 초기 조건에 비해서 다항식을 구성하는 계수의 개수가 적으므로 키 방정식 연산 회로 구현시 적은 기본 셀을 필요로 하여 낮은 하드웨어 복잡도를 달성할 수 있다.

식 (12)~(15)는 새로운 단순화된 수정 유클리드 알고리즘이 사용하는 다항식 연산 방법을 나타낸다.

$$R_{i+1}(x) = [b_i R_i(x) + a_i Q_i(x)]x^{\omega_i} \quad (12)$$

$$\lambda_{i+1}(x) = [b_i \lambda_i(x) + a_i \mu_i(x)]x^{\omega_i} \quad (13)$$

$$Q_{i+1}(x) = \overline{\omega_i} Q_i(x) + \omega_i R_i(x) \quad (14)$$

$$\mu_{i+1}(x) = \overline{\omega_i} \mu_i(x) + \omega_i \lambda_i(x) \quad (15)$$

제어 신호 ω_i와 $Q_i(x)$의 최고차항 계수 b_i는 그림 8에 의해서 결정된다.

기존 수정 유클리드 알고리즘은 식 (7)에 의해서 다항식 연산 및 교환 연산을 제어하였다. 그러나 앞서 설명한 것처럼 식 (7)의 다항식 차수 연산 및 비교 연산은 수정 유클리드 알고리즘의 하드웨어 설 계시 하드웨어 복잡도 및 임계 경로를 증가시키는 주된 원인이다. 따라서 단순화된 수정 유클리드 알고리즘은 제어 신호 sw를 사용하여 다항식 연산 및 교환 연산을 단순화한다.

sw 값이 0이면 현재 R 다항식의 차수가 Q 다항식의 차수보다 낮다는 것을 나타내므로 식 (12)과 (13)의 다항식 연산뿐만 아니라 식 (14)과 (15)의 교환 연산을 함께 수행한다. 반면에 sw 값이 0이 아니면 현재 R 다항식의 차수가 Q 다항식의 차수보다 크거나 같다는 것을 의미하므로 식 (12)과 (13)의 다항식 연산만을 수행하게 된다. 결과적으로 제어 신호 sw는 식 (7)의 l_i와 같은 역할을 수행하는 반면 그림 8의 간단한 순차도로 표현 가능하다. 따라서 하드웨어 구현시 기존 방식에 비해 쉽게 구현이 가능하다.

그림 9는 단순화된 수정 유클리드 알고리즘을 사용하는 키 방정식 연산기를 나타낸다. 그림 9의 상위 셀은 식 (12)과 (14)의 오류 크기 다항식 연산을 수행하고 하위 셀은 식 (13)과 (15)의 오류 위치 다항식 연산을 반복적으로 수행한다. 앞서 설명한 것처럼 제안하는 키 방정식 연산기는 식 (10)과 (11)의 새로운 초기 조건을 사용하므로 오류 크기 다항식 연산을 위해서 $2t$ 개의 기본 셀을 필요로 한다. 그러나 제안하는 알고리즘은 기존 수정 유클리드

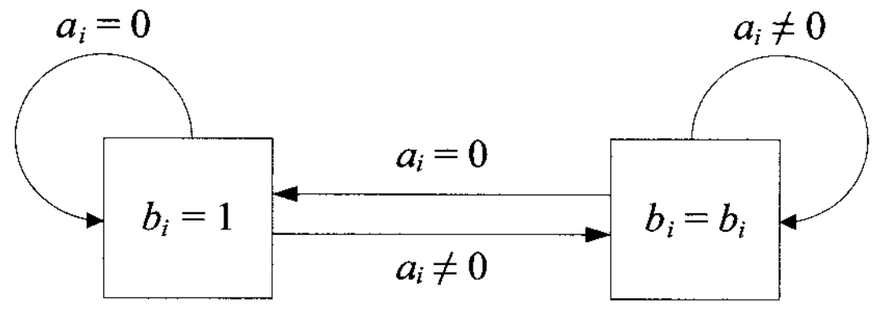


그림 8. 다항식 연산 제어 신호.
Fig. 8. State diagram for the new control method.

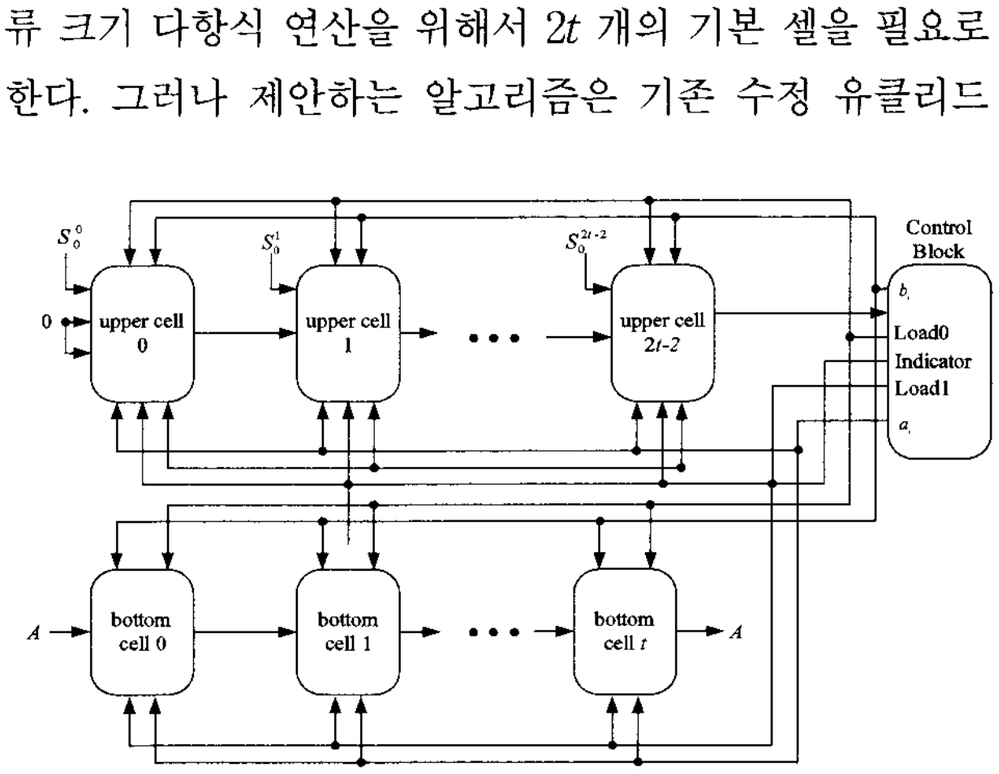


그림 9. 새로운 단순화된 수정 유클리드 키 방정식 연산기.
Fig. 9. New simplified Euclid's algorithm architecture.

알고리즘과 같이 오류 크기 다항식 $R(x)$의 최고차항 계수를 반복적인 연산을 통해서 지속적으로 제거하므로 최상위 기본 셀의 출력은 항상 0이다[16]. 따라서 제안하는 키 방정식 연산기는 최상위 셀의 유한체 곱셈기 및 덧셈기를 제거하여 최상위 셀과 제어 블록을 통합하였고 그 결과 $2t$ - 1개의 기본 셀만을 필요로 한다.

또한, 오류 위치 다항식 연산을 수행하는 하위 셀 역시 $2t$ + 1개를 구성되어야 하지만 하위 셀은 최상위 셀의 출력을 하위 셀에 입력하여 두 번 사용하는 방식을 적용하여 낮은 하드웨어 복잡도를 달성할 수 있다[15]. 따라서 제안하는 키 방정식 연산기는 $2t$ - 1개의 상위 셀과, t + 1개의 하위 셀, 제어 블록으로 구성된다.

그러나 그림 9에서 보인 제안하는 새로운 단순화된 수정 유클리드 연산기의 데이터 패스는 그림 6의 기존 RiBM 연산기에 비해 상대적으로 복잡해 보인다. 그러나 앞서 설명한 것처럼 그림 6에는 레지스터 및 멀티플렉서 제어 신호가 생략되어 있다. 따라서 그림 9에서 레지스터 및 멀티플렉서 제어 신호인 Load0, Load1, Indicator 신호를 제외하면 기존 RiBM 연산기와 제안하는 연산기의 데이터 패스는 유사한 복잡도를 갖는 것을 확인할 수 있다.

그림 10은 2개의 곱셈기, 2개의 레지스터, 1개의 덧셈기, 1개의 멀티플렉서로 구성된 제안하는 상위 셀을 나타낸다. 식 (10)과 (11)의 새로운 초기 조건 가운데 $R_0(x)$만 오류 형태에 따라 달라지는 신드롬 다항식에 의해 다른 값을 가지며 다른 초기 조건들은 오류 형태에 관계없이 항상 동일한 값을 갖는다. 따라서 제안하는 키 방정식 연산기는 $R_0(x)$의 초기 조건을 입력받기 위해 한 개의 멀티플렉서를 사용한다. 그러나 $Q_0(x), \lambda_0(x), \mu_0(x)$ 초기 조건은 레지스터에 1 또는 0의 초기 값으로 저장하고 이를 사용하므로 별도로의 초기 조건 입력을 위해 데

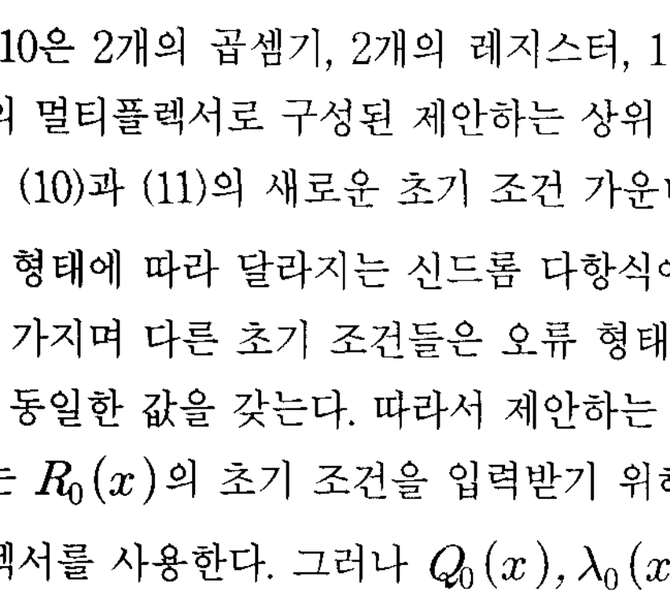


그림 10. 제안하는 상위 셀.
Fig. 10. Proposed upper cell.

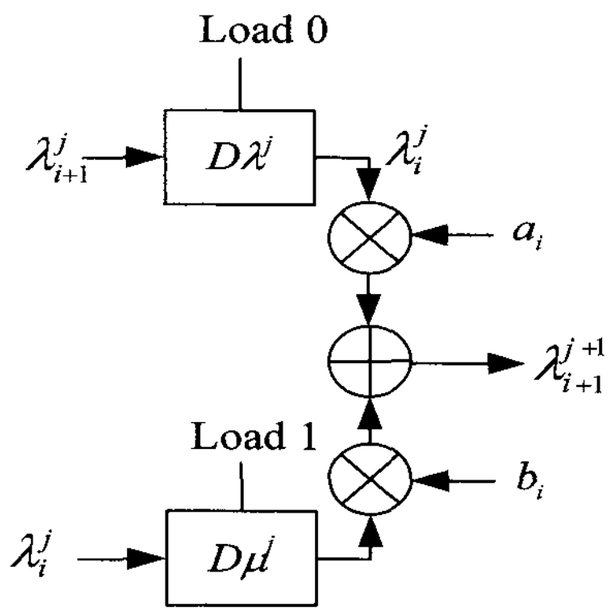


그림 11. 제안하는 하위 셀.
Fig. 11. Proposed bottom cell.

이터 패스를 필요로 하지 않는다.

그림 11은 2개의 곱셈기, 2개의 레지스터, 1개의 덧셈기만으로 구성된 제안하는 하위 셀을 나타낸다. 앞서 설명한 것처럼 식 (11)의 $\lambda_0(x)$ 및 $\mu_0(x)$는 항상 고정된 값을 가지므로 $D\lambda$ 레지스터와 $D\mu$ 레지스터는 초기 조건 로드 신호가 입력되면 항상 1 또는 0으로 초기화 된다. 따라서 제안하는 하위 셀은 별도의 초기 조건 입력을 위한 데이터 패스가 필요 없고 그 결과 RiBM 키 방정식 연산기의 기본 셀인 그림 5의 PE에 비해 낮은 하드웨어 복잡도를 갖는다.

Ⅳ. 성능 비교

새로운 단순화된 수정 유클리드 알고리즘을 사용하는 리드-솔로몬 복호기는 VHDL을 사용하여 설계하였으며 삼성 0.18㎛ 라이브러리와 SYNOPSYS™을 사용하여 논리 합성을 수행하였다. 구현한 리드-솔로몬 복호기는 370MHz의 동작 속도와 메모리를 제외하고 40,136개의 게이트로 구성된다. 표 1은 (255, 239, 8) 리드-솔로몬 부호에서 기존 DCME 복호기[15], RiBM 복호기[8]와의 성능 비교를 나타낸다.

RS 복호기에 사용되는 FIFO 메모리는 앞서 설명한 것처럼 수신된 데이터 심벌을 저장하기 위한 것으로 지원하는 RS 부호가 동일하다면 동일한 크기의 메모리를 사용한다. 결과적으로 본 논문에서 사용한 (255, 239, 8) RS 부호를 지원하는 RS 복호기에 사용되는 FIFO 메모리 크기는 하드웨어 구조에 관계없이 255개의 데이터 심벌을 저장하기 위해서 256Byte로 동일하다. 따라서 메모리 크기는 성능비교에서 제외하였다.

$3t$ + 2개의 기본 셀로 구성된 DCME 키 방정식 연산기[15]는 $6t$ + 4개의 곱셈기, $6t$ + 4개의 레지스터, $3t$ + 2개의

표 1. (255, 239, 8) 리드 솔로몬 복호기 성능 비교
Table 1. Performance comparisons.

		DCME	RiBM	제안하는 구조
공정		0.25㎛	0.25㎛	0.18㎛
키 방정식 연산기	레지스터	$6t + 4$	$6t + 2$	$6t + 2$
	멀티플렉서	$6t + 4$	$3t + 1$	$2t - 1$
	곱셈기	$18t + 12$	$6t + 2$	$6t$
	덧셈기	$3t + 2$	$3t + 1$	$3t$
키 방정식 연산기 게이트 수		21,760	-	20,166
전체 게이트 수		42,213	-	40,136
키 방정식 연산 Latency		$2t$	$2t$	$2t$

덧셈기, $18t + 12$개의 2-입력 멀티플렉서로 구성된다. 따라서 DCME 키 방정식 연산기는 21,760개의 게이트로 구성되며, DCME 연산기를 사용하는 리드-솔로몬 복호기는 메모리를 제외하고 42,220 게이트를 사용한다. 또한 DCME 연산기는 키 방정식 연산을 위해 $2t$ 클록 사이클을 필요로 한다.

$3t + 1$개의 PE로 구성된 RiBM 키 방정식 연산기[8]은 $6t + 2$개의 곱셈기, $6t + 2$개의 레지스터, $3t + 1$개의 덧셈기, $3t + 1$개의 2-입력 멀티플렉서로 구성된다. 따라서 RiBM 키 방정식 연산기는 DCME 연산기에 비해 낮은 하드웨어 복잡도를 갖는다. 실제로 제어 블록을 제외한 RiBM 키 방정식 연산 회로의 게이트수를 계산해보면 약 18,600 게이트로 구성된다. RiBM 연산기 역시 키 방정식 연산을 위해 $2t$ 클록 사이클을 필요로 한다.

반면에 $3t$개의 기본 셀로 구성된 제안하는 단순화된 수정 유클리드 키 방정식 연산기는 전체 $6t$개의 곱셈기, $6t + 2$개의 레지스터, $3t$ 개의 덧셈기, $2t - 1$개의 2-입력 멀티플렉서로 구성된다. 따라서 현재까지 가장 낮은 하드웨어 복잡도를 갖는 RiBM 키 방정식 연산기에 비해서 곱셈기 1개, 덧셈기 1개, 2-입력 멀티플렉서 $t + 2$개 낮은 하드웨어 복잡도를 갖는다. 제안하는 단순화된 수정 유클리드 연산기의 게이트 수는 20,166개이며, RiBM 연산기와의 공정한 비교를 위해 제어 블록을 제외하면 17,800 게이트만을 필요로 한다. 전체 리드-솔로몬 복호기의 게이트 수는 40,136개이다. 따라서 단순화된 수정 유클리드 키 방정식 연산기는 기존 연산기들과 동일한 성능을 갖으면서도 DCME 연산기에 비해 약 5%, RiBM 연산기에 비해 약 2.5%의 면적 감소 효과가 예상된다.

V. 결 론

본 논문에서는 새로운 단순화된 수정 유클리드 알고리즘을 사용하는 저전력 및 저면적 리드-솔로몬 복호기를 제안하였다. 새로운 단순화된 수정 유클리드 키 방정식 연산기는 새로운 초기 조건 및 새로운 다항식 연산 방식을 사용 기존 키 방정식 연산기들에 비해 낮은 하드웨어 복잡도를 갖는다. 20,166 게이트로 구성된 단순화된 수정 유클리드 키 방정식 연산기는 기존 키 방정식 연산기들에 비해 최소 2.5% 이상의 면적 감소 효과를 갖는다. 따라서 제안한 리드-솔로몬 복호기는 짧은 지연 시간 및 낮은 하드웨어 복잡도를 가지므로 DVB-T, DVB-RCS, T-DMB, DVD, Blu-ray disk 등 다양한 디지털 통신 시스템 및 디지털 저장 장치에 활용 가능하다.

참 고 문 헌

[1] A. Batra *etal.*, "Multi-Band OFDM Physical Layer Proposal for IEEE 802.15 Task Group 3a," *IEEE* P802.15-03/268r3, Mar.2004.

[2] TTAS.KO-06.0082/R1, "Specifications for 2.3GHz band portable internet service: physical & medium access control layer," Dec. 2005.

[3] European Telecommunications Standards Institute (ETSI), "interaction channel for satellite distribution systems," EN 301 790 v1.4.1, Sep. 2005.

[4] E. R. Berlekamp, *Algebraic Coding Theory*. New York : McGraw-Hill, 1968. (revised - Laguna Hills, CA : Aegean Park, 1984)

[5] R. E. Blahut, *Theory and Practice of Error-Control Codes*. Reading, MA : Addison-Wesley,1983.

[6] J. H. Jeng and T. K. Truong, "On decoding of both errors and erasures of a Reed-Solomon code using an inverse-free Berlekamp-Massey algorithm," *IEEE Trans. Commun.*, vol. 47, pp. 1488-1494, Oct. 1999.

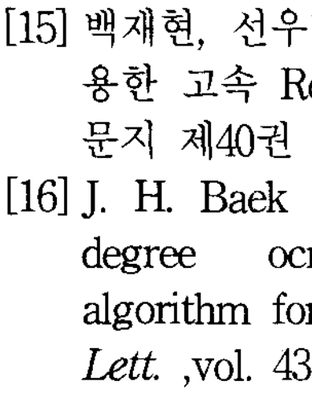
[7] A. Raghupathy and K. J. R. Liu, "Algorithm-based low-power/high-speed Reed-Solomon decoder design," *IEEE Trans. Circuit Syst. II*, vol. 47, pp. 1254-1270, Nov. 2000.

[8] D. V. Sarwate and N. R. Shanbhag, "High-speed architectures for Reed- Solomon decoders," *IEEE Trans. VLSI Syst.*, vol. 9, pp. 641-655, Oct. 2001.

[9] M. A. A.Ali, A. Abou-El-Azm, and M. F. Marie, "Error rates for non-coherent demodulation

FCMA with Reed-Solomon codes in fading satellite channel," in *Proc. IEEE Vehicular Techn. Conf. (VTC'99)*, vol. 1, 1999, pp. 92-96.
[10] T. K. Matsushima, T. Matsushima, and S. Hirasawa, "Parallel architecture for high-speed Reed-Solomon codec," in *Proc. IEEE Int. Telecommun. Symp. (ITS'98)*, vol. 2, 1998, pp. 468-473.
[11] H. M. Shao, T. K. Truong, L. J. Deutsch, J. H. Yuen and I. S. Reed, "A VLSI design of a pipeline Reed-Solomon decoder," *IEEE Trans. Comput.*, vol. C-34, pp. 393-403, May 1985.
[12] H. H. Lee, M. L. Yu and L. Song, "VLSI design of Reed-Solomon decoder architectures," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS' 2000)*, vol. 5, May 2000, pp. 705-708.
[13] H. H. Lee, "Modified Euclidean algorithm block for high-speed Reed-Solomon decoder," *Electron. Lett.*, vol. 37, pp. 903-904, July 2001.
[14] J. H. Baek and Myung H. Sunwoo, "New degree computationless modified Euclid's algorithm and architecture for Reed-Solomon decoder," *IEEE Trans. VLSI Syst.*, vol. 14, pp. 915-920, Aug. 2006.
[15] 백재현, 선우명훈, "새로운 DCME 알고리즘을 사용한 고속 Reed-Solomon 복호기," 전자공학회 논문지 제40권 SD편, 6호, 81-90쪽, 2003.
[16] J. H. Baek and Myung H. Sunwoo, "Enhanced degree ocmputationless modified Euclid's algorithm for Reed-Solomon decoders," *Electron. Lett.* ,vol. 43, pp. 175-177, Feb. 2007.
[17] J. H. Baek and M. H. Sunwoo. "Simplified Degree Computationless Modified Euclid's Algorithm and its Architecture," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS'2007)*, May 2007, pp. 905-908.

저 자 소 개



백 재 현(정회원)

2002년 아주대학교 전자공학부 학사 졸업.
2008년 아주대학교 전자공학부 박사 졸업(석박사 통합).
2008년 현재 서울대학교 BK21 정보기술사업단 박사후 연구원.

<주관심분야 : 통신 및 신호처리 SOC 설계>

선우 명 훈(정회원)

1980년 서강대학교 전자공학과 학사 졸업.
1982년 KAIST 전자공학과 석사 졸업.
1990년 Univ. of Texas at Austin 전자공학과 박사 졸업.
2008년 현재 아주대학교 전자공학부 교수

<주관심분야 : VLSI 및 Parallel Architecture, 통신 멀티미디어용 DSP 칩 및 SOC 설계>