(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2012년02월28일
(11) 등록번호 10-1115088
(24) 등록일자 2012년02월03일

(51) Int. Cl.
H03M 13/15 (2006.01)
(21) 출원번호 10-2010-0109754
(22) 출원일자 2010년11월05일
심사청구일자 2010년11월05일
(56) 선행기술조사문헌
KR1020060098269 A
KR1020100060930 A

(73) 특허권자
인하대학교 산학협력단
인천광역시 남구 인하로 100, 인하대학교 (용현동)
(72) 발명자
이한호
인천광역시 남구 인하로 100, 인하대학교 정보통신공학부 (용현동)
박정인
인천광역시 남구 인하로 100, 인하대학교 하이테크센터 528호 (용현동)
(74) 대리인
특허법인무한

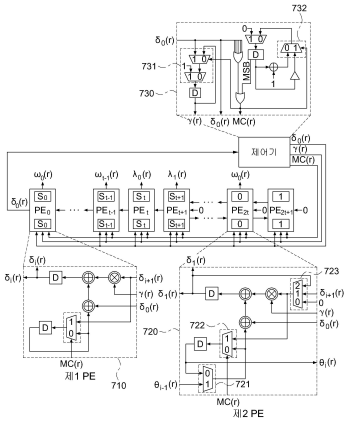
전체 청구항 수 : 총 2 항 심사관 : 장석환

(54) 절단된 RiBM 구조를 이용한 고속 저면적 리드-솔로몬 복호 장치

(57) 요 약

절단된 RiBM 구조를 이용한 고속 저면적 리드-솔로몬 복호 장치가 개시된다. 리드-솔로몬 복호 장치는 TiBM 구조를 사용하여 신드롬 다항식을 기초로 오류 위치 다항식과 오류 값 다항식을 구하기 위한 KES 블록을 포함하는 것으로, KES 블록은 t+1개의 제1 PE와 t+1개의 제2 PE 및 제어기를 포함하고, 제어기에서 전달된 제어 신호에 의해 제1 PE와 제2 PE에서 연산된 다항식을 선택하기 위하여, 제1 PE는 한 개의 선택기를 구성하고 제2 PE는 세 개의 선택기를 구성한다. 이때, 제1 PE에 구성된 선택기는 이전 값과 다항식 δ_{i+1} 중 한 가지를 선택하는 역할을 수행하고, 제2 PE에 구성된 선택기 중에서 제1 선택기는 제1 PE에 구성된 선택기와 동일한 역할을 수행하고, 제2 선택기는 이전 값과 다항식 Θ_{i-1} 중 한 가지의 값을 선택하는 역할을 수행하고, 제3 선택기는 다항식 δ_i과 δ_{i+1}, 그리고 0 값 중 한 가지의 값을 선택하는 역할을 수행한다.

대 표 도 - 도8



이 발명을 지원한 국가연구개발사업
과제고유번호 KI002145
부처명 지식경제부
연구관리전문기관 한국산업기술평가관리원
연구사업명 산업원천기술개발사업
연구과제명 차세대 광통신용디지털신호처리 기반 초고속 CMOS 회로 설계 기술
기여율 1/1
주관기관 인하대학교 산학협력단
연구기간 2009.03.01 ~ 2014.02.28

특허청구의 범위

청구항 1

TiBM 구조(Truncated inversionless Berlekamp-Massey)를 사용하여 신드롬 다항식을 기초로 오류 위치 다항식과 오류 값 다항식을 구하기 위한 KES(Key-Equation Solver) 블록을 포함하는 리드-솔로몬(Reed-Solomon) 복호 장치에 있어서,

상기 KES 블록은,

t+1개의 제1 PE(processing element)와 t+1개의 제2 PE 및 제어기를 포함하고,

상기 제어기에서 전달된 제어 신호에 의해 상기 제1 PE와 제2 PE에서 연산된 다항식을 선택하기 위하여, 상기 제1 PE는 한 개의 선택기를 구성하고 상기 제2 PE는 세 개의 선택기를 구성하며,

상기 제1 PE에 구성된 선택기는 이전 값과 다항식 δ_{i+1} 중 한 가지를 선택하는 역할을 수행하고,

상기 제2 PE에 구성된 선택기 중에서 제1 선택기는 상기 제1 PE에 구성된 선택기와 동일한 역할을 수행하고, 제2 선택기는 이전 값과 다항식 Θ_{i-1} 중 한 가지의 값을 선택하는 역할을 수행하고, 제3 선택기는 다항식 δ_i과 δ_{i+1}, 그리고 0(zero) 값 중 한 가지의 값을 선택하는 역할을 수행하고,

상기 다항식 δ_i과 δ_{i+1}는 불합도(discrepancy)를 나타내는 불합도 다항식이고, 상기 다항식 Θ_{i-1}는 상기 불합도 다항식을 임시 저장하는 다항식인 것

을 특징으로 하는 리드-솔로몬 복호 장치.

청구항 2

제1항에 있어서,

상기 KES 블록은,

상기 제2 선택기 및 상기 제3 선택기에 의해 연산 시간 동안 항상 고정된 0(zero)의 값을 가지는 다항식 δ_i 값을 제거하는 TiBM 구조로 구성되는 것

을 특징으로 하는 리드-솔로몬 복호 장치.

명 세 서

기 술 분 야

[0001] 본 발명의 실시예들은 고속 저면적 리드-솔로몬 복호 장치에 관한 것으로써, 보다 상세하게는 절단된 RiBM(Reformulated inversionless Berlekamp-Massey) 구조를 KES(Key Equation Solver) 블록에 적용함으로써 저면적의 리드-솔로몬 복호기를 구현하는 기술에 관한 것이다.

배 경 기 술

[0002] 리드-솔로몬 코드는 마그네틱, 메모리, 광 저장매체, 유선 및 위성 통신 등 다양한 응용분야에 널리 쓰이는 순방향 오류정정(Forward Error Correction; FEC) 기술로써 통신로 상에서 발생하는 랜덤오류 및 연집오류까지 정정 가능한 채널부호화방식이다.

[0003] 이와 같은 광범위한 분야에서 사용되고 있는 리드-솔로몬 코드를 하드웨어로 구현함에 있어서, 플래시 메모리와 같은 저장매체 및 무선통신 시스템은 더 작은 하드웨어 크기를 요구하기 때문에 종래의 리드-솔로몬 복호기보다 저면적으로 구현할 수 있는 장치 및 방법이 요구된다.

발명의 내용

해결하려는 과제

[0004] RiBM 알고리즘을 사용한 리드-솔로몬 복호기를 수정하여 저면적의 리드-솔로몬 복호기를 구현하는 리드-솔로몬 복호 장치가 제공된다.

[0005] 리드-솔로몬 복호기를 이루는 하위 블록 중 가장 큰 KES 블록에서, 불필요한 데이터 연산 부분을 제거하는 TiBM(Truncated inversionless Berlekamp-Massey)구조를 KES블록에 적용시키는 리드-솔로몬 복호 장치가 제공된다.

과제의 해결 수단

[0006] TiBM 구조를 사용하여 신드롬 다항식을 기초로 오류 위치 다항식과 오류 값 다항식을 구하기 위한 KES 블록을 포함하는 리드-솔로몬 복호 장치에 있어서, KES 블록은 t+1개의 제1 PE와 t+1개의 제2 PE 및 제어기를 포함하고, 제어기에서 전달된 제어 신호에 의해 제1 PE와 제2 PE에서 연산된 다항식을 선택하기 위하여, 제1 PE는 한 개의 선택기를 구성하고 제2 PE는 세 개의 선택기를 구성한다. 이때, 제1 PE에 구성된 선택기는 이전 값과 다항식 δ_{i+1} 중 한 가지를 선택하는 역할을 수행하고, 제2 PE에 구성된 선택기 중에서 제1 선택기는 제1 PE에 구성된 선택기와 동일한 역할을 수행하고, 제2 선택기는 이전 값과 다항식 Θ_{i-1} 중 한 가지의 값을 선택하는 역할을 수행하고, 제3 선택기는 다항식 δ_i과 δ_{i+1}, 그리고 0 값 중 한 가지의 값을 선택하는 역할을 수행한다. 여기서, 다항식 δ_i과 δ_{i+1}는 불합도를 나타내는 불합도 다항식이고, 다항식 Θ_{i-1}는 상기 불합도 다항식을 임시 저장하는 다항식이다.

[0007] 일측에 따르면, 리드-솔로몬 복호 장치의 KES 블록은 제2 선택기 및 제3 선택기에 의해 연산 시간 동안 항상 고정된 0의 값을 가지는 다항식 δ_i 값을 제거하는 TiBM 구조로 구성된다.

발명의 효과

[0008] RiBM 알고리즘을 사용한 리드-솔로몬 복호기에서 불필요한 데이터 연산을 제거하는 TiBM 구조를 적용하여 하드웨어 크기를 감소시키는 연산 회로 및 그 방법을 제시함으로써 저면적의 리드-솔로몬 복호기를 구현할 수 있다. 또한, 하드웨어 크기가 감소하였기 때문에 플래시 메모리 및 무선통신시스템과 같은 적은 하드웨어 크기가 요구되는 시스템에 바로 적용될 수 있다.

도면의 간단한 설명

[0009] 도 1은 단일 채널 리드-솔로몬 복호기의 구조를 설명하기 위한 도면이다.

도 2는 단일 채널 리드-솔로몬 복호기의 연산과정을 설명하기 위한 도면이다.

도 3은 신드롬 계산 블록의 연산회로를 도시한 도면이다.

도 4는 RiBM 알고리즘을 이용한 KES 블록의 연산과정을 순서도로 도시한 도면이다.

도 5는 친서치 블록과 포니 알고리즘 계산블록의 구조를 도시한 도면이다.

도 6은 본 발명의 일실시예에 있어서, KES 블록의 데이터 흐름을 설명하기 위한 도면이다.

도 7은 RiBM 구조를 사용한 KES 블록과 TiBM 구조를 사용한 KES 블록 내부에 사용되는 PE(processing element) 개수의 차이를 도시한 도면이다.

도 8은 본 발명의 일실시예에 있어서, TiBM 구조를 사용한 KES 블록의 회로 구조를 도시한 도면이다.

도 9는 본 발명의 일실시예에 있어서, KES 블록의 PE에 추가된 선택기의 반복 회수에 따른 선택신호를 설명하기 위한 도면이다.

발명을 실시하기 위한 구체적인 내용

[0010] 이하, 본 발명의 실시예를 첨부된 도면을 참조하여 상세하게 설명한다.

[0011] 도 1은 단일 채널 리드-솔로몬 복호기의 구조를 설명하기 위한 도면이고, 도 2는 단일 채널 리드-솔로몬 복호기의 연산과정을 설명하기 위한 도면이다.

[0012] 도 1과 도 2를 참조하여, 리드-솔로몬 복호기의 연산 과정을 설명한다.

[0013] 신드롬 다항식 계산(Syndrome Polynomial Computation)(이하, 'SC'라 칭함) 블록(101)은 신드롬 다항식 S(x)를 생성하고 수신된 코드워드(codeword)의 오류 패턴을 표현한다. 신드롬 다항식 S(x)는 리드-솔로몬 복호기의 두 번째 부분인 KES(Key-Equation Solver) 블록(102)에서 사용된다. KES 블록(102)은 키 등식(Key equation) $S(x)\sigma(x)=\omega(x) \bmod x^{2t}$을 해결하기 위해 유클리드 알고리즘, 수정 유클리드 알고리즘, 또는 벨르캄프-메시 알고리즘 등을 이용한다. 이러한 알고리즘은 오류 위치 다항식(error locator polynomial) $\sigma(x)$와 오류 값 다항식(error value polynomial) $\omega(x)$을 구하기 위하여 사용된다.

[0014] 벨르캄프-메시 알고리즘은 일반적으로 수정 유클리드 알고리즘보다 적은 하드웨어 복잡도를 가지고 있기 때문에 KES 블록을 구현함에 있어서 수정 유클리드 알고리즘보다 유리하다. 이러한 벨르캄프-메시 알고리즘을 분석하여 데이터의 흐름을 살펴보면, 불필요한 하드웨어 부분을 제거하는 것이 가능하다. 따라서, 기존의 벨르캄프-메시 알고리즘보다 더 하드웨어 크기가 줄어든 KES 블록을 구현할 수 있다.

[0015] 오류 위치 다항식 $\sigma(x)$와 오류 값 다항식 $\omega(x)$의 두 다항식은 친서치(Chien Search) 블록(103)과 포니(Formey) 알고리즘 연산 블록(104)을 이용하여 오류의 위치에 대응하는 오류들의 크기 값을 구하기 위해 사용된다.

[0016] 일반적인 직렬(serial) 구조에서 사용되는 SC 블록(101)의 연산은 수학식 1과 같이 나타낼 수 있다. 리드-솔로몬 부호기에서 사용했던 연속된 16개의 근(root)들을 수학식 1에 각각 대입하여 16개의 신드롬 값을 얻을 수 있다.(도 2에서 201)

수학식 1

$$S(x) = S_{15}x^{15}+S_{14}x^{14}+\dots +S_1x+S_0$$

[0017]

$$S_i = R(\alpha^i) = R_{254}\alpha^{254i} + R_{253}\alpha^{253i} + \dots + R_1\alpha^i + R_0, \ (i = 0,1,2, \dots ,15)$$

[0018] 도 3은 SC 블록(101)의 연산회로를 도시한 도면이다. 도 3은 수학식 1을 회로로 구현한 것으로서, 이로부터 16개의 신드롬 값을 생성한다. 여기서, 각각의 곱셈기는 상수 곱셈기(constant multiplier)로서 일반 곱셈기 보다 하드웨어 복잡도가 적다. 총 255개의 수신 메시지가 각각의 기본 셀(cell)들로 들어오게 되면, 각각의 기본 셀들은 재귀적으로 순환 하면서 누적 덧셈 값을 가지게 된다. 이러한 연산이 255 클럭 동안 이루어지고, 연산이 끝나게 되면 16개의 신드롬 값이 병렬로 KES 블록(102)으로 들어가게 된다.

[0019] KES 블록(102)에서 사용되는 기본적인 RiBM 알고리즘은 오류 위치 및 오류 값 다항식을 구하기 위해 다음 수학식 2 및 수학식 3과 같이 초기 값을 정한다. 여기서, 수학식 2의 $\delta_i(x)$는 불합도(discrepancy)를 나타내는 다항식이고, k와 γ는 제어신호를 나타낸다. 또한, 수학식 3의 $\Theta_i(x)$는 불합도 다항식을 임시로 저장하는 다항식을 나타내고, S_i는 앞서 언급한 신드롬 다항식이다.

수학식 2

$$\delta_{3t}(0) = 1, \ \delta_i(0) = 0 \ (i = 2t, 2t+1, \dots , 3t-1), \ k(0) = 0, \ \gamma(0) = 1$$

[0020]

수학식 3

$$\delta_i(0) = \Theta_i(0) = S_i, \ (i = 0, \dots , 2t-1)$$

[0021]

[0022] 수학식 4는 반복적으로 2t번 수행되고, 동시에 불합도 다항식을 업데이트(update)하는 $\delta_i(x)$, $\Theta_i(x)$, k, γ에

대해 나타낸 것이다. 여기서, r값은 0부터 2t-1까지 증가하면서 불합도 다항식을 업데이트 하게 된다.

수학식 4

$$\delta_i(r+1) = \gamma(r) \cdot \delta_{i+1}(r) - \delta_0(r) \cdot \Theta_i(r), (i = 0, \ldots, 3t)$$

[0023]

수학식 5

if $\delta_0(r) \neq 0$ and $k(r) \geq 0$

then $\Theta_i(r+1) = \delta_{i+1}(r), (i = 0, 1, \ldots, 3t)$

$\gamma(r+1) = \delta_0(r)$

$k(r+1) = -k(r) - 1$

else $\Theta_i(r+1) = \Theta_i(r), (i = 0, 1, \ldots, 3t)$

$\gamma(r+1) = \gamma(r)$

$k(r+1) = k(r) + 1$

[0024]

[0025] RiBM 알고리즘은 위의 수학식 5를 2t번만큼 반복적으로 연산하는 것이고, 2t번의 연산이 끝나면 오류위치 다항식 $\sigma(x)$와 오류 값 다항식 $\omega(x)$가 수학식 6과 같이 출력된다. (도 2에서 202, 203)

수학식 6

$$\sigma_i(2t) = \delta_{t+1}(2t), (i = 0, 1, \ldots, t)$$

$$\omega_i(2t) = \delta_i(2t), (i = 0, 1, \ldots, t-1)$$

[0026]

[0027] RiBM 알고리즘을 적용한 KES 블록(102)은 총 3t+1개의 처리소자(processing element)(이하, 'PE'라 칭함)로 구성되고, 각각의 PE는 2개의 곱셈기, 레지스터(register)와 1개의 덧셈기(adder), 선택기(multiplexer)로 구성된다. 각각의 PE는 전단의 출력 값 $\delta_{i+1}(r)$와 제어 유닛(control unit)으로부터의 출력 값 MC, δ_0, γ를 입력으로 받아서 연산을 수행한다. 이러한 연산을 2t번 수행한 후 PE_0부터 PE_7까지 $\omega(x)$의 값이 출력되고, PE_8부터 PE_{16}까지 $\sigma(x)$의 값이 출력된다.

[0028] 도 4는 RiBM 알고리즘을 이용한 KES 블록(102)의 연산과정을 순서도로 도시한 도면이다.

[0029] RiBM 알고리즘을 적용한 KES 블록(102)의 데이터 흐름을 분석하면, 불필요한 부분을 제거할 수 있고, 따라서 리드-솔로몬 복호기의 전체적인 하드웨어 크기를 감소시킬 수 있다.

[0030] 수학식 7은 친서치 블록(103)과 포니 알고리즘 계산블록(104)의 연산을 나타낸다. 가능한 모든 255개의 근들을 수학식 7에 대입하여 오류의 위치 $\sigma(x)$와 오류의 값 $\omega(x)$들을 찾을 수 있다.(도 2에서 204,205)

수학식 7

$$\sigma(x) = \sigma_t x^t + \sigma_{t-1} x^{t-1} + \sigma_{t-2} x^{t-2} + \ldots + \sigma_2 x^2 + \sigma_1 x^1 + \sigma_0$$

$$\omega(x) = \sigma_{t-1} x^{t-1} + \omega_{t-2} x^{t-2} + \ldots + \omega_2 x^2 + \omega_1 x^1 + \omega_0$$

[0031]

[0032] 도 5는 친서치 블록(103)과 포니 알고리즘 계산블록(104)의 구조를 도시한 도면이다.

[0033] 친서치 블록(103)은 총 8개의 기본 셀들로 이루어져 있고, 각각의 기본 셀들은 각각 1개씩의 상수 곱셈기와 레지스터, 선택기로 이루어져 있다. 기본 셀에서 선택기는 첫 번째 클럭에서만 1을 선택하고, 나머지 255클럭까지는 0의 값을 선택한다. 이후, 레지스터 값이 나머지 클럭 동안 재귀적으로 곱해짐으로써 오류 위치를 찾게 된다. 기본 셀들로부터 나온 값들은 덧셈 트리(tree)를 거치고 최종적으로 모두 더해 줌으로써 수학식 8을 완성하게 된다. 또한, 홀수 번째 기본 셀들은 따로 모아져 더해짐으로써 오류위치 다항식을 미분한 값($x\sigma'(x)$)을 얻을 수 있고, 이 값은 최종적으로 오류를 정정할 때 쓰이게 된다.

[0034] 포니 알고리즘 계산블록(104)은 친서치 블록의 C8이 제거됨으로써 포니 알고리즘 계산블록을 구성하게 된다. 여기서, 친서치 블록(103)과 포니 알고리즘 계산블록(104)의 기본 셀들은 동일하다. 친서치 블록(103)과 마찬가지로 255개의 모든 근들을 차례대로 대입함으로써 오류 값을 찾을 수 있다.

[0035] 도 6은 본 발명의 일실시예에 있어서, KES 블록(102)의 데이터 흐름을 설명하기 위한 도면이다.

[0036] 도 6의 상단도면을 참조하면, δ_i값이 16클럭 동안 변함에 있어서 점선으로 표시된 부분의 δ_i값은 오류의 개수와 관계없이 항상 0의 값을 가지게 됨을 알 수 있다. 이 0값은 고정되었고, 미리 알 수 있기 때문에 제거될 수 있다. 이것은 하드웨어 크기가 감소될 수 있음을 의미한다. 즉, 도 6의 하단도면을 참조하면, 고정된 0의 값을 제거되고 남은 δ_i값들이 좌측으로 병합된 것을 도시한 것이다.

[0037] 도 7은 RiBM 구조를 사용한 KES 블록과 TiBM 구조를 사용한 KES 블록 내부에 사용되는 PE 개수의 차이를 도시한 도면이다.

[0038] 도 7을 참조하면, RiBM 구조(701)에서의 PE 개수와 TiBM 구조(702)에서의 PE 개수의 차이를 나타내고 있다. RiBM 구조(701)에서의 PE 개수는 총 3t+1개의 PE가 사용되고, TiBM 구조(702)에서의 PE 개수는 총 2t+2개의 PE가 사용된다. TiBM 구조(702)가 RiBM 구조(701)에 비해 t-1개의 PE가 더 적게 사용되는 것을 알 수 있다. 이것은 전체적인 KES 블록의 하드웨어 크기를 감소시키므로 저면적의 리드-솔로몬 복호기를 구현할 수 있음을 나타낸다. 또한, 오류 정정 능력인 t값이 증가할수록 하드웨어 감소 효과는 더 증가한다.

[0039] 도 8은 본 발명의 일실시예에 있어서, TiBM 구조를 사용한 KES 블록의 회로 구조를 도시한 도면이다.

[0040] 도 8을 참조하면, TiBM 구조를 사용한 KES 블록은 RiBM 구조에서 사용했던 기존 t+1개의 제1 PE(710)와, 기존의 제1 PE(710)에 선택기(721)(723)를 추가한 수정된 t+1개의 제2 PE(720)와, 그리고 1개의 제어기(730)로 구성된다.

[0041] 제1 PE(710)은 앞단의 δ_{i+1}값과 선두의 δ_0값을 γ 값과 선택기(711)에 의해 선택된 값으로 각각 곱하고 이를 더하여 다음 단으로 δ 값을 전달해 주는 역할을 수행한다. 이것은 수학식 5에 나타난 δ 값과 Θ값을 조건에 따라 갱신하는 과정을 나타낸다. 여기서, 선택기(711)은 제어기(730)에서 전달된 MC신호에 의해 이전 값과 δ_{i+1}의 값 중에서 한 가지를 선택한다.

[0042] 제2 PE(720)은 제1 PE(710)에서 선택기(721)(723)가 추가된 것이고, 추가된 선택기(721)(723)은 도 6에 나타난 고정된 0의 값을 제거하기 위해 구성된 것이다. 추가된 선택기(721)은 Θ값을 갱신하는데 있어서 이전 값과 Θ_{i-1}의 값 중에서 한 가지의 값을 선택한다. 그리고, 선택기(722)의 역할은 선택기(711)의 역할과 동일하고, 추가된 선택기(723)은 δ_i값과 δ_{i+1}, 그리고 0의 값 중에서 한 가지의 값을 선택하는 역할을 수행한다.

[0043] 제어기(730)은 KES 블록의 전체적인 제어신호를 생성해주는 역할을 수행한다. 이때, 선택기(731)은 첫 클럭일 때는 초기 γ 값을 1로 설정하고 이후 16클럭까지는 현재의 δ_0값과 이전의 δ_0값 중 한 가지를 선택하여 γ 신호를 생성한다. 그리고, 선택기(732)는 수학식 5에 나타난 제어신호 k값을 생성하는 역할을 수행한다.

[0044] 이와 같이 제1 PE(710)에 선택기 두 개를 추가하는 대신, t-1개의 제1 PE(710)를 제거할 수 있으므로 저면적의 KES 블록을 구현할 수 있다.

[0045] 도 9는 본 발명의 일실시예에 있어서, KES 블록의 제2 PE에 추가된 선택기의 반복 회수에 따른 선택신호를 설명하기 위한 도면이다. 즉, 제2 PE(720) 내에 추가된 선택기(721)(723)의 반복횟수에 따른 선택신호를 도시한 것이다.

[0046] 도 9를 참조하면, KES 블록은 16클럭의 연산 시간을 갖기 때문에 총 16번의 반복을 하게 된다. 이때, 각각의 반복횟수에 따라 선택기(721)(723)의 선택신호는 도 9에 나타난 표와 같이 선택되고, 이것은 TiBM 구조의 값이 기존 RiBM 구조에서의 값과 동일하게 제어되는 역할을 수행한다.

[0047] 상기한 바와 같이, 리드-솔로몬 복호기는 크게 SC 블록과 KES 블록, 그리고 오류의 위치와 오류의 값을 알려주는 친서치 블록과 포니 알고리즘 계산블록으로 구성된다. 리드-솔로몬 복호기의 구성요소 중 KES 블록은 하드웨어 크기에 있어서 가장 큰 비중을 차지한다. 일반적으로, KES 블록은 RiBM 알고리즘을 사용하여 구현되므로 상대적으로 큰 하드웨어 크기를 차지한다. 이러한 RiBM 알고리즘을 사용한 기존 KES 블록은 불필요한 연산 부분을 포함하기 때문에 개선할 필요가 있다.

[0048] 따라서, 본 발명의 실시예들에 따르면, TiBM 구조를 사용하여 KES 블록을 구성하여 기존 RiBM 구조보다 t-1개의 PE를 줄일 수 있으므로 약 25%의 하드웨어 크기를 감소시킬 수 있다. 이는 작은 하드웨어 크기가 요구되는 플래시 메모리와 같은 저장장치 시스템에 적용될 수 있다.

[0049] 이상과 같이 본 발명은 비록 한정된 실시예와 도면에 의해 설명되었으나, 본 발명은 상기의 실시예에 한정되는 것은 아니며, 본 발명이 속하는 분야에서 통상의 지식을 가진 자라면 이러한 기재로부터 다양한 수정 및 변형이 가능하다.

[0050] 그러므로, 본 발명의 범위는 설명된 실시예에 국한되어 정해져서는 아니 되며, 후술하는 특허청구범위뿐 아니라 이 특허청구범위와 균등한 것들에 의해 정해져야 한다.

부호의 설명

[0051] 710: 제1 PE

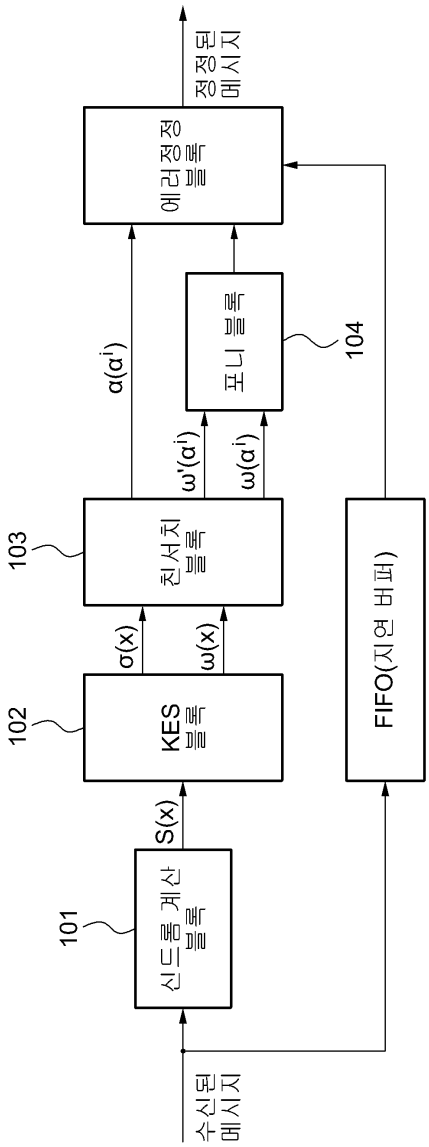
720: 제2 PE

721, 722, 723: 선택기

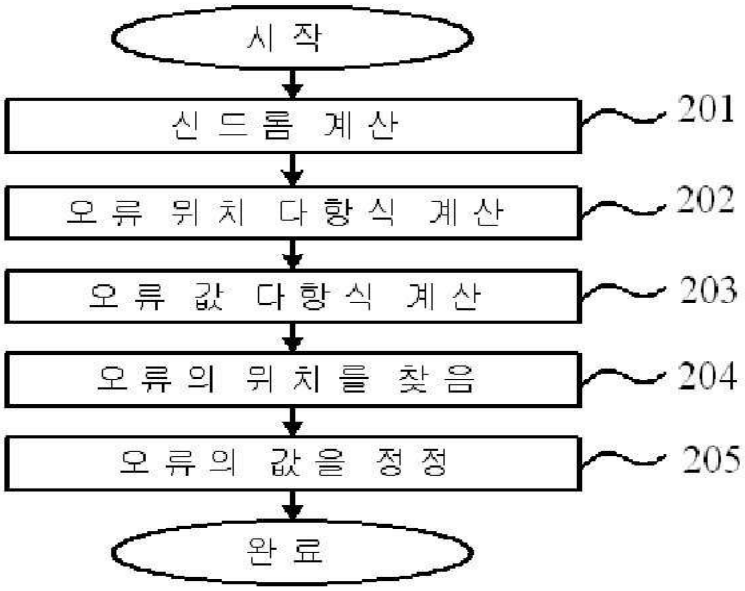
730: 제어기

도면

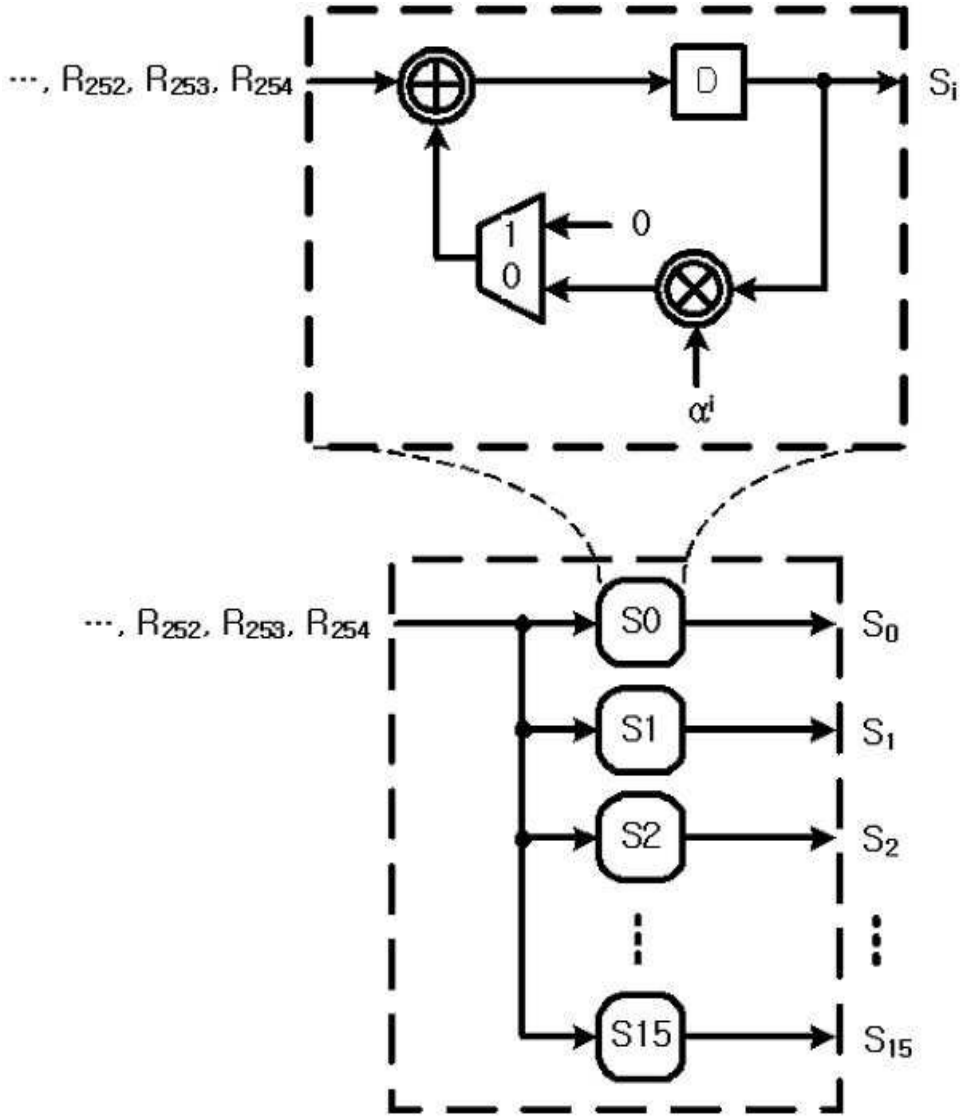
도면1


수신된 메시지
신드롬 계산 블록
101
S(x)
KES 블록
102
σ(x)
ω(x)
천서치 블록
103
α(α^i)
ω'(α^i)
ω(α^i)
포니 블록
104
에러정정 블록
정정된 메시지
FIFO(지연 버퍼)

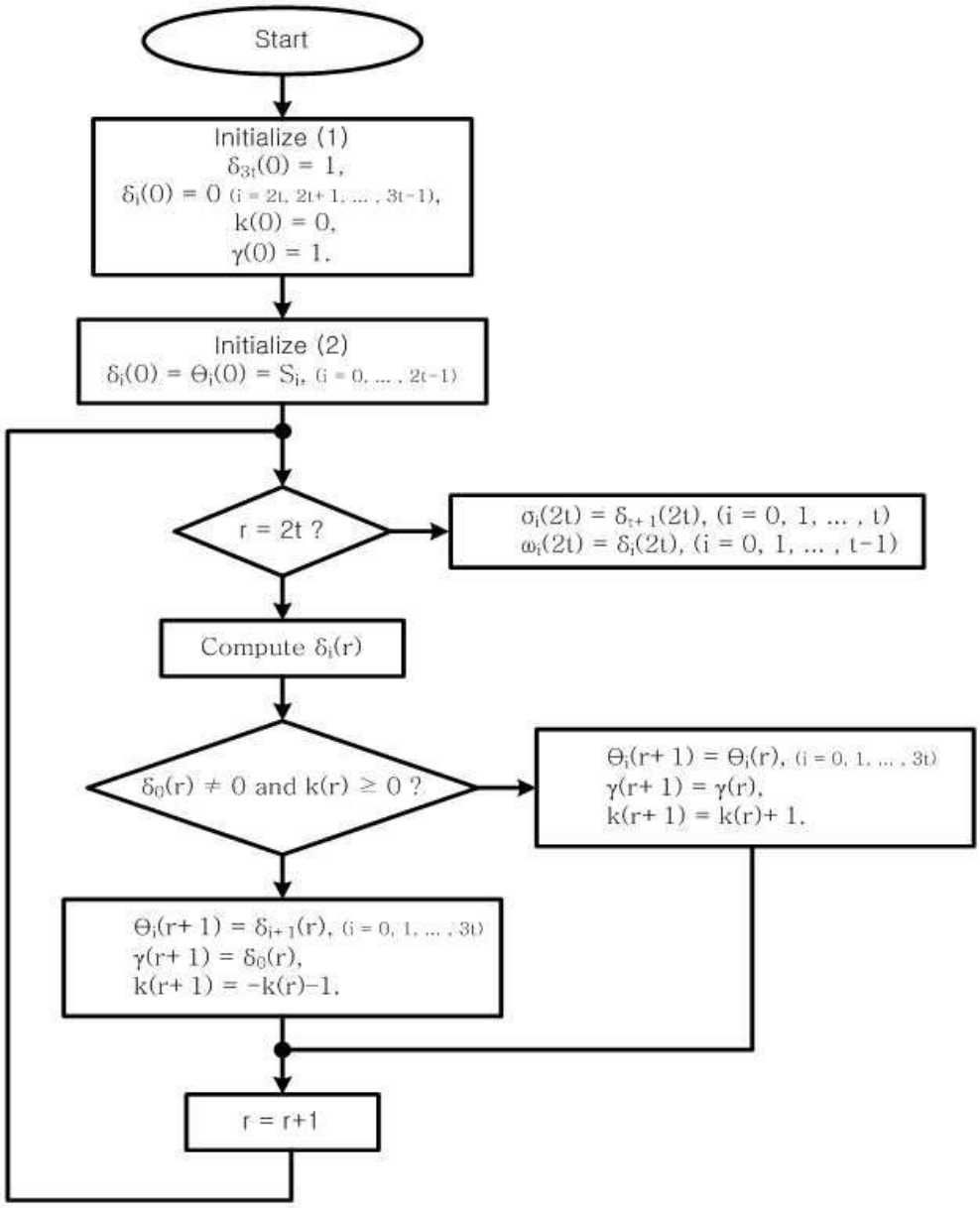
도면2


시 작
신 드 롬 계 산
201
오 류 위 치 다 항 식 계 산
202
오 류 값 다 항 식 계 산
203
오 류 의 위 치 를 찾 음
204
오 류 의 값 을 정 정
205
완 료

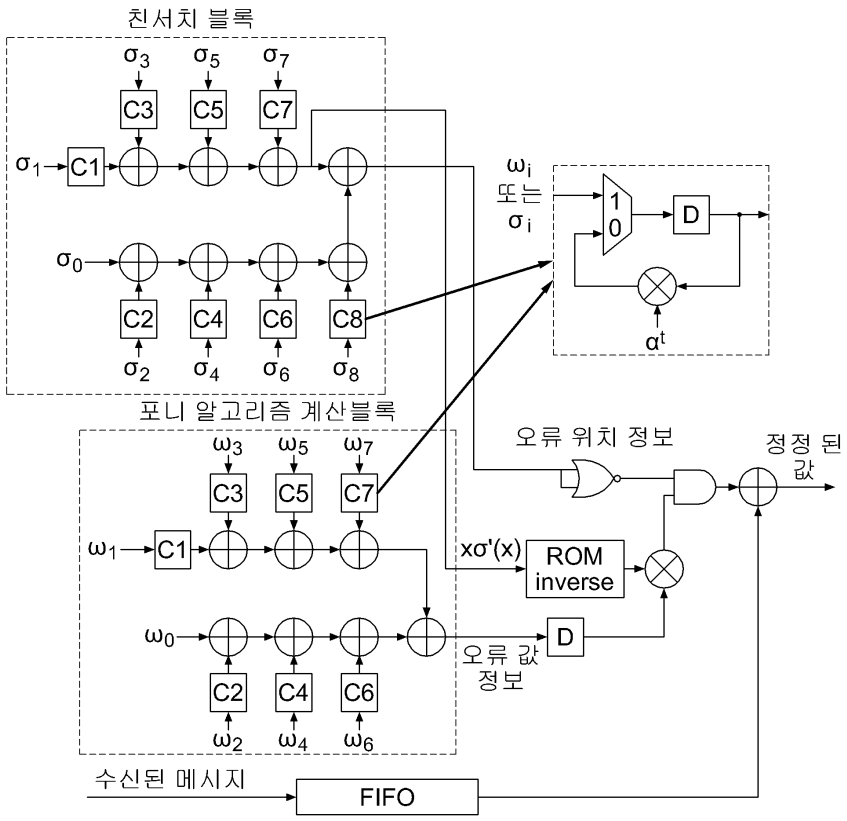
도면3


…, R252, R253, R254
D
Si
1
0
0
αi
…, R252, R253, R254
S0
S1
S2
S15
S0
S1
S2
S15

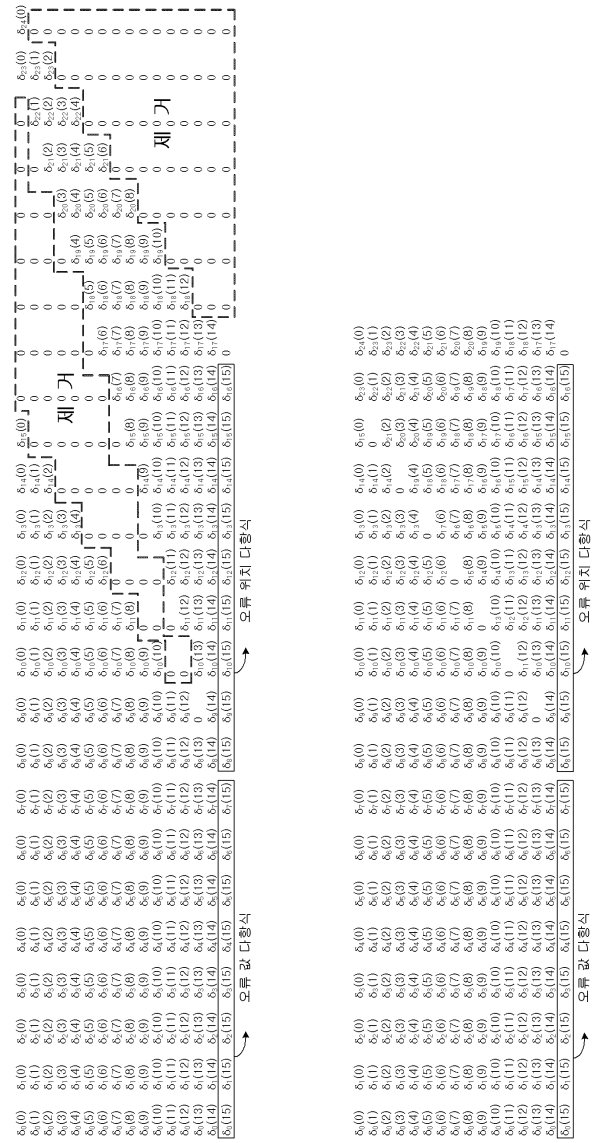
도면4


Start
Initialize (1)
$\delta_{3t}(0) = 1$,
$\delta_i(0) = 0$ (i = 2t, 2t+1, ... , 3t-1),
k(0) = 0,
γ(0) = 1.
Initialize (2)
$\delta_i(0) = \Theta_i(0) = S_i$, (i = 0, ... , 2t-1)
r = 2t ?
$\sigma_i(2t) = \delta_{t+1}(2t)$, (i = 0, 1, ... , t)
$\omega_i(2t) = \delta_i(2t)$, (i = 0, 1, ... , t-1)
Compute $\delta_i(r)$
$\delta_0(r) \neq 0$ and $k(r) \geq 0$?
$\Theta_i(r+1) = \Theta_i(r)$, (i = 0, 1, ... , 3t)
γ(r+1) = γ(r),
k(r+1) = k(r)+1.
$\Theta_i(r+1) = \delta_{i+1}(r)$, (i = 0, 1, ... , 3t)
γ(r+1) = $\delta_0(r)$,
k(r+1) = -k(r)-1.
r = r+1

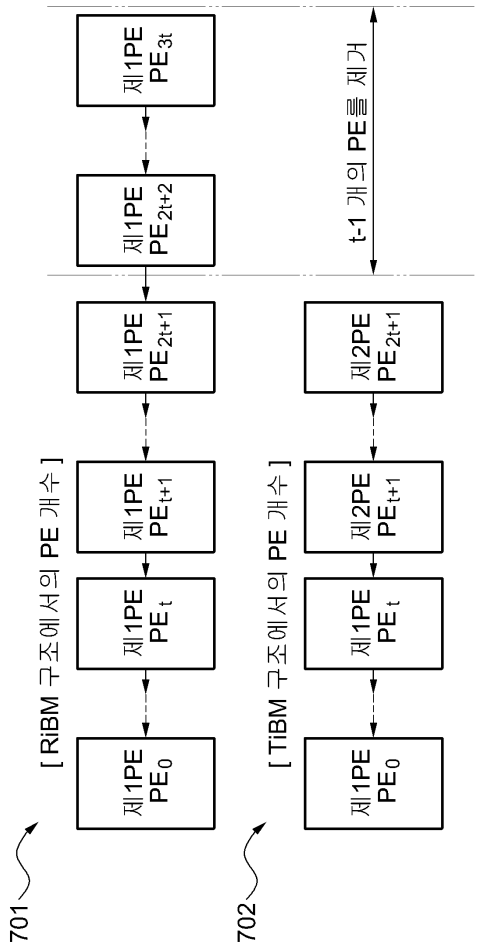
도면5


신서치 블록
σ3
σ5
σ7
C3
C5
C7
σ1
C1
σ0
C2
C4
C6
C8
σ2
σ4
σ6
σ8
ωi
또는
σi
1
0
D
αt
포니 알고리즘 계산블록
ω3
ω5
ω7
C3
C5
C7
ω1
C1
ω0
C2
C4
C6
ω2
ω4
ω6
오류 위치 정보
정정 된
값
xσ'(x)
ROM
inverse
D
오류 값
정보
수신된 메시지
FIFO

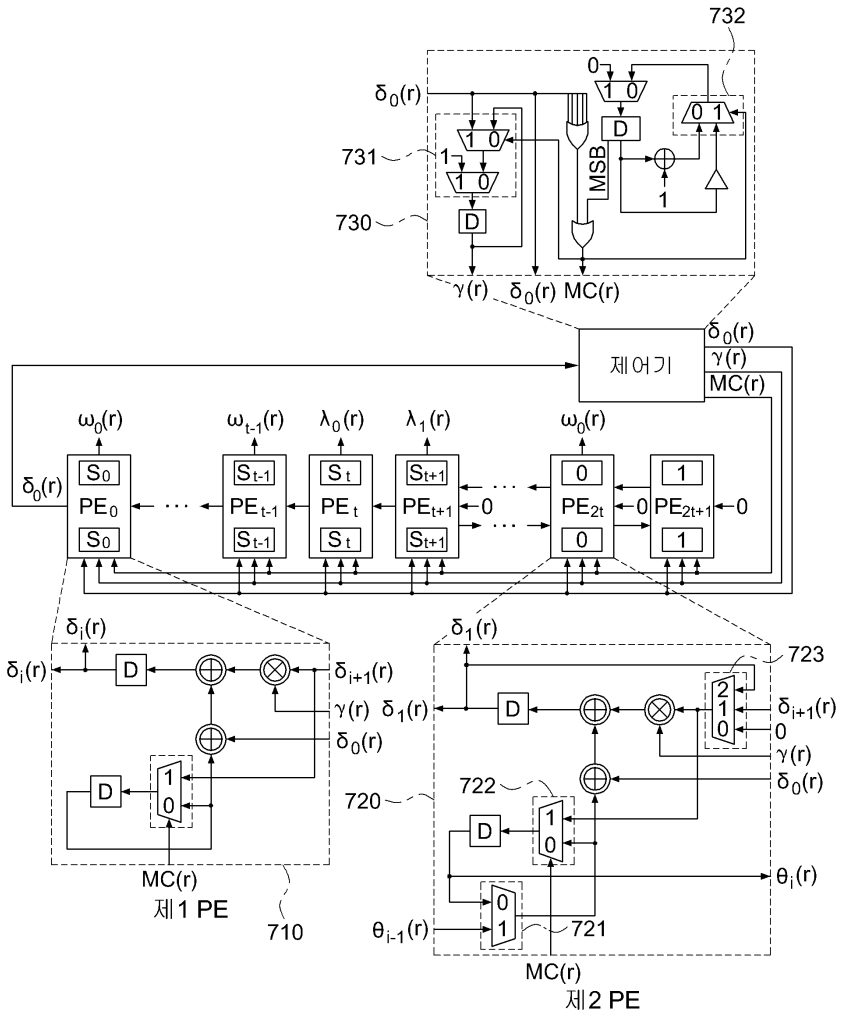
도면6



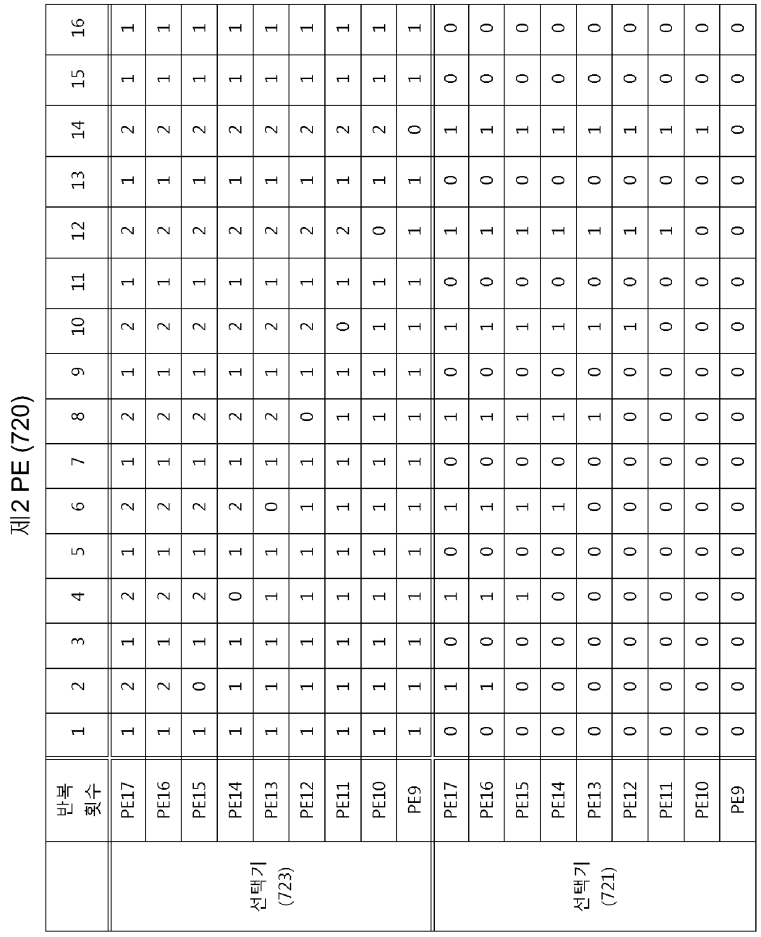
도면7


[RiBM 구조에서의 PE 개수]
제1PE PE0
제1PE PEt
제1PE PEt+1
제1PE PE2t+1
제1PE PE2t+2
제1PE PE3t
701
[TiBM 구조에서의 PE 개수]
제1PE PE0
제1PE PEt
제2PE PEt+1
제2PE PE2t+1
702
t-1 개의 PE를 제거

도면8


732
731
730
MSB
제어기
PE0
PEt-1
PEt
PEt+1
PE2t
PE2t+1
제1 PE
제2 PE
710
720
721
722
723
MC(r)

도면9


제2 PE (720)
선택기 (723)
선택기 (721)
반복 횟수
PE17 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2
PE16 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2
PE15 1 0 1 2 1 2 1 2 1 2 1 2 1 2 1 2
PE14 1 1 1 0 1 2 1 2 1 2 1 2 1 2 1 2
PE13 1 1 1 1 1 0 1 2 1 2 1 2 1 2 1 2
PE12 1 1 1 1 1 1 1 0 1 2 1 2 1 2 1 2
PE11 1 1 1 1 1 1 1 1 1 0 1 2 1 2 1 2
PE10 1 1 1 1 1 1 1 1 1 1 1 0 1 2 1 2
PE9 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 2
PE17 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
PE16 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
PE15 0 0 0 1 0 1 0 1 0 1 0 1 0 1 0 1
PE14 0 0 0 0 0 1 0 1 0 1 0 1 0 1 0 1
PE13 0 0 0 0 0 0 0 1 0 1 0 1 0 1 0 1
PE12 0 0 0 0 0 0 0 0 0 1 0 1 0 1 0 1
PE11 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1
PE10 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
PE9 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0