



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2017-0112645
(43) 공개일자 2017년10월12일

(51) 국제특허분류(Int. Cl.)
H03M 13/15 (2015.01) G11C 29/42 (2015.01)
(52) CPC특허분류
H03M 13/15 (2013.01)
G11C 29/42 (2013.01)
(21) 출원번호 10-2016-0039975
(22) 출원일자 2016년04월01일
심사청구일자 2016년04월01일

(71) 출원인
고려대학교 산학협력단
서울특별시 성북구 안암로 145, 고려대학교 (안암동5가)
(72) 발명자
박중선
서울특별시 서초구 신반포로 9, 85동 403호 (반포동, 반포아파트)
당호영
서울특별시 송파구 오금로35길 17, 33동 904호 (오금동, 현대아파트)
(74) 대리인
특허법인엠에이피에스

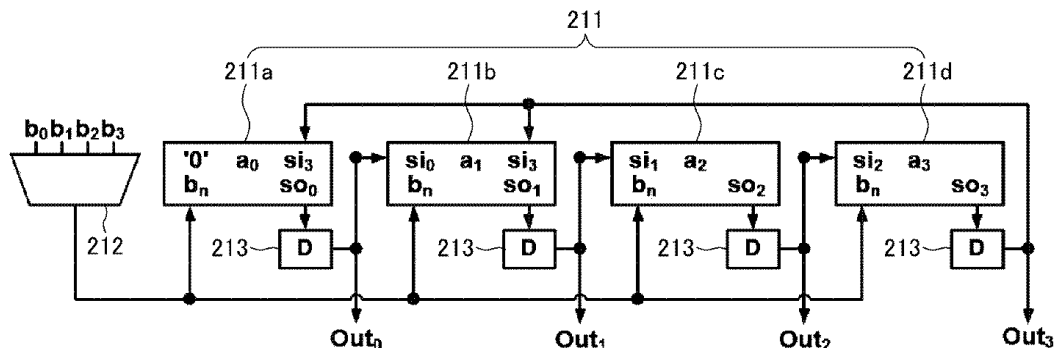
전체 청구항 수 : 총 12 항

(54) 발명의 명칭 폴딩된 곱셈기가 적용된 BCH 디코더

(57) 요약

본 발명의 일 실시예에 따르는, 폴딩된 곱셈기가 적용된 BCH 디코더는, 복수의 곱셈기를 포함하는 키 방정식 연산기를 포함하며, 상기 곱셈기는, 연산 동작을 수행하는 복수의 연산 블록을 포함한다. 이때, 각 연산 블록은, 복수의 연산 스테이지 동안 각 연산 스테이지마다 반복적으로 연산 동작을 수행하되, 각 연산 스테이지마다 적어도 하나의 입력값을 기반으로 하나의 출력값을 출력하고, 현재 연산 스테이지의 출력값을 다음 연산 스테이지에서의 적어도 하나의 다른 연산 블록의 입력값으로 전달하도록 다른 연산 블록과 연결된다.

대표도



이 발명을 지원한 국가연구개발사업

과제고유번호 1345159833

부처명 미래창조과학부

연구관리전문기관 한국연구재단

연구사업명 기초연구실육성사업

연구과제명 그린 스마트패치 및 응용 소프트웨어 연구

기 여 율 1/20

주관기관 고려대학교 산학협력단

연구기간 2015.09.01 ~ 2016.08.31이 발명을 지원한 국가연구개발사업

과제고유번호 10052716

부처명 산업통상자원부

연구관리전문기관 한국연구재단

연구사업명 산업기술혁신사업

연구과제명 스마트 센서 soc용 초저전압 회로 및 IP 설계 기술 개발

기 여 율 9/20

주관기관 고려대학교 산학협력단

연구기간 2015.06.01 ~ 2020.05.31이 발명을 지원한 국가연구개발사업

과제고유번호 2016R1A2B4015329

부처명 교육과학기술부

연구관리전문기관 한국연구재단

연구사업명 중견연구자지원사업

연구과제명 사물인터넷용 저면적/저전력 전용 프로세서 설계를 위한 맞춤형 임베디드 메모리 개발

기 여 율 10/20

주관기관 고려대학교 산학협력단

연구기간 2016.06.01 ~ 2019.05.31

명세서

청구범위

청구항 1

폴딩된 곱셈기가 적용된 BCH(Bose-Chaudhuri-Hoc quenghem) 디코더에 있어서,

복수의 곱셈기를 포함하는 키 방정식 연산기를 포함하며,

상기 곱셈기는,

연산 동작을 수행하는 복수의 연산 블록을 포함하되,

각 연산 블록은,

복수의 연산 스테이지 동안 각 연산 스테이지마다 반복적으로 연산 동작을 수행하되, 각 연산 스테이지마다 적어도 하나의 입력값을 기반으로 하나의 출력값을 출력하고, 현재 연산 스테이지의 출력값을 다음 연산 스테이지에서의 적어도 하나의 다른 연산 블록의 입력값으로 전달하도록 다른 연산 블록과 연결되는 것인, 폴딩된 곱셈기가 적용된 BCH 디코더.

청구항 2

제 1 항에 있어서,

상기 곱셈기는,

원시 다항식의 계수 값(coefficient of primitive polynomial element)이 서로 상이한 연산 블록으로 구성된 것인, 폴딩된 곱셈기가 적용된 BCH 디코더.

청구항 3

제 2 항에 있어서,

상기 복수의 연산 블록은,

상기 원시 다항식 계수 값이 0인 적어도 하나의 연산 블록과 상기 원시 다항식의 계수 값이 1인 적어도 하나의 연산 블록으로 구성되는 것인, 폴딩된 곱셈기가 적용된 BCH 디코더.

청구항 4

제 3 항에 있어서,

상기 원시 다항식의 계수 값이 0인 연산 블록과 상기 원시 다항식의 계수 값이 1인 연산 블록의 개수는 동일한 것인, 폴딩된 곱셈기가 적용된 BCH 디코더.

청구항 5

제 1 항에 있어서,

상기 각 연산 블록은

AND 게이트와 XOR 게이트의 조합으로 구성되는 것인, 폴딩된 곱셈기가 적용된 BCH 디코더.

청구항 6

제 3 항에 있어서,

상기 원시 다항식의 계수 값이 0인 연산 블록은 1개의 AND 게이트와 1개의 XOR 게이트로 구성되고,

상기 원시 다항식의 계수 값이 1인 연산 블록은 1개의 AND 게이트와 2개의 XOR 게이트로 구성되는 것인, 폴딩된 곱셈기가 적용된 BCH 디코더.

청구항 7

제 6 항에 있어서,

상기 원시 다항식의 계수 값이 0인 연산 블록은, XOR 게이트의 출력값이 AND 게이트의 출력값에 종속되도록 구성되며,

상기 원시 다항식의 계수 값이 1인 연산 블록은, 어느 한 개의 XOR 게이트의 출력값이 나머지 한 개의 XOR 게이트의 출력값과 AND 게이트의 출력값에 종속되도록 구성되는 것인, 폴딩된 곱셈기가 적용된 BCH 디코더.

청구항 8

제 1 항에 있어서,

상기 곱셈기는,

상기 각 연산 블록과 병렬로 연결되어, 상기 각 연산 스테이지마다 복수의 연산 블록에 제 1입력값을 제공하는 믹스(MUX)를 더 포함하는, 폴딩된 곱셈기가 적용된 BCH 디코더.

청구항 9

제 8 항에 있어서,

상기 복수의 연산 블록 중,

원시 다항식의 계수 값이 0인 어느 하나의 연산 블록은 출력값을 두 개 이상의 다른 연산 블록의 제 2 입력값으로 입력하도록 구성되며,

나머지 연산 블록들은 출력값을 한 개의 다른 연산 블록의 제 3 입력값으로 입력하도록 구성되는, 폴딩된 곱셈기가 적용된 BCH 디코더.

청구항 10

제 9 항에 있어서,

상기 각 연산 블록은,

상기 각 연산 스테이지마다 전달받은 입력값과 미리 설정된 제 4 입력값을 기초로 연산을 수행하는, 폴딩된 곱셈기가 적용된 BCH 디코더.

청구항 11

제 8 항에 있어서,

상기 곱셈기는,

두 개의 연산 블록 사이마다 연결되어, 클럭 신호에 동기화하여 이전 연산 스테이지의 연산 블록의 출력값을 다음 연산 스테이지의 다른 연산 블록의 입력값으로 전달하는 플립플롭(flip-flop)을 더 포함하는, 폴딩된 곱셈기가 적용된 BCH 디코더.

청구항 12

제 8 항에 있어서,

상기 각 연산 블록은,

상기 믹스로부터 출력되는 상기 제 1 입력값이 마지막 연산 스테이지에 대한 값인 경우,

연산 결과에 따르는 출력값을 상기 곱셈기의 최종 출력값으로 제공하는 것인, 폴딩된 곱셈기가 적용된 BCH 디코더.

발명의 설명

기술 분야

[0001] 본 발명은 폴딩된 곱셈기가 적용된 BCH 디코더에 관한 것으로서, 보다 상세하게는, BCH 디코더의 동작속도를 개선하기 위해 KES(Key Equation Solver) 내에 폴딩 구조를 적용한 곱셈기가 적용된 BCH 디코더에 관한 것이다.

배경 기술

[0002] NAND 플래시 메모리 등의 스토리지 장치 및 디지털 전송을 처리하는 통신 시스템 등에서는 처리하는 메시지에 대한 오류 정정 처리가 필요하며, 이를 위해 오류정정부호인 BCH(Bose-Chaudhuri-Hoc quenghem) 부호가 이용되고 있다. 예를 들어, NAND 플래시 메모리는 저장할 메시지를 BCH 인코더를 통해 BCH 코드로 부호화한 후 메모리에 저장하고, 메모리로부터 독출된 BCH 코드를 BCH 디코더를 통해 복호화하여 원본 메시지를 출력한다.

[0003] 일반적으로 BCH 알고리즘을 이용한 에러 정정은 BCH 인코더를 통해 BCH 코드 데이터를 생성한 후, BCH 디코더를 통해 BCH 코드 데이터의 복호화를 통해 이루어진다. 이때, BCH 코드 데이터의 복호화는 다음과 같은 과정을 거친다. BCH 코드 데이터가 BCH 디코더에 수신되면, 신드롬(syndrome)을 계산하고, 신드롬을 이용하여 에러 위치 다항식(error locator polynomial)을 구성한 다음, 에러 위치 다항식의 근(root)을 구하여 에러 비트들(error bits)의 위치를 계산한다. 이진 BCH 코드 데이터에 대한 복호화의 경우, 에러 비트의 비트 값을 반전시킴으로써 에러를 정정할 수 있다.

[0004] 이러한 BCH 디코더는 KES(Key Equation Solver)를 포함한다. 그러나, KES의 동작 속도에 한계가 있기 때문에, BCH 디코더가 높은 동작 주파수와 빠른 연산 속도로 동작하는 데에 많은 문제가 발생하고 있다. 기존의 KES는 도1과 같은 GF(Galois Field) multiplier로 구성된다. 이러한 종래의 GF multiplier는 그 특성상 중복되는 연산 과정을 많이 포함하며 계산 경로 딜레이(critical path delay)가 크므로 낮은 동작 주파수로 인한 느린 연산 속도를 갖는다. 이에 따라, 종래의 KES는 BCH 디코더의 높은 동작 속도 구현에 걸림돌이 되어 왔다. 또한, GF multiplier는 폴딩이 불가능한 구조로 구성되기 때문에 불가피하게 KES의 대부분의 면적을 차지하여 왔다.

선행기술문헌

특허문헌

[0005] (특허문헌 0001) 한국 등록특허 제1307792호 (등록일 : 2013.09.06)

발명의 내용

해결하려는 과제

[0006] 본 발명은 전술한 종래 기술의 문제점을 해결하기 위한 것으로서, BCH 디코더 내의 KES를 구성하는 곱셈기(multiplier)들을 폴딩 방식으로 구현함으로써, BCH 디코더의 동작속도 개선과 동시에 KES 구조의 저면적화를 달성할 수 있도록 하는 것을 목적으로 한다.

과제의 해결 수단

[0007] 본 발명의 일 실시예에 따르는, 폴딩된 곱셈기가 적용된 BCH 디코더는, 복수의 곱셈기를 포함하는 키 방정식 연산기(Key Equation solver)를 포함하며, 상기 곱셈기는, 연산 동작을 수행하는 복수의 연산 블록을 포함한다. 이때, 각 연산 블록은, 복수의 연산 스테이지 동안 각 연산 스테이지마다 반복적으로 연산 동작을 수행하되, 각 연산 스테이지마다 적어도 하나의 입력값을 기반으로 하나의 출력값을 출력하고, 현재 연산 스테이지의 출력값을 다음 연산 스테이지에서의 적어도 하나의 다른 연산 블록의 입력값으로 전달하도록 다른 연산 블록과 연결된다.

[0008] 종래기술의 경우, 곱셈기 자체의 회로 구조가 매우 불규칙적으로 배열되어 있어 곱셈기의 폴딩 구조 적용이 불가능하였다. 그러나, 본 발명의 일 실시예는 곱셈기를 구성하는 연산 블록들을 규칙적인 조합의 구조로 변환하여 연산이 수행되도록 하여 종래 곱셈기에서는 불가능하였던 곱셈기의 폴딩을 가능하게 하였다. 그에 따라, 본 발명의 일 실시예는 불규칙한 구조와 복잡한 연산방식에 기인한 종래의 KES의 동작 속도 한계를 해결할 수 있게 되었으며, 그 결과 KES의 계산 경로(critical path)를 짧게 구현할 수 있다.

발명의 효과

[0009] 본 발명의 일 실시예는 곱셈기를 구성하는 연산 블록들을 규칙적인 조합의 구조로 구성하여 폴딩이 가능하도록 구현함으로써 종래의 KES의 동작 속도 한계를 해결할 수 있게 되었으며, 그 결과 KES의 계산 경로(critical path)를 짧게 구현할 수 있다. 즉, 동작 주파수를 높여 KES의 연산 속도 병목 현상을 해결할 수 있게 되어 전체 BCH 디코더의 연산 속도를 상승시킬 수 있다. 또한, GF dimension이 커져도 동일한 동작속도를 유지할 수 있어 고속 BCH 디코더 동작을 구현하기에 매우 유리하다. 또한, 연산 블록의 개수를 최소화하는 구조이기 때문에 곱셈기의 저면적화를 달성함으로써 전체적인 BCH 디코더의 면적을 최소화할 수 있다.

도면의 간단한 설명

[0010] 도 1은 종래기술에 따르는 KES(Key Equation Solver) 내의 곱셈기의 회로구조에 대한 개념도이다.

도 2는 본 발명의 일 실시예에 따르는 BCH 디코더의 구조도이다.

도 3은 본 발명의 일 실시예에 따르는 BCH 디코더의 키 방정식 연산기의 회로구조에 관한 개념도이다.

도 4는 본 발명의 일 실시예에 따르는 BCH 디코더의 키 방정식 연산기를 구성하는 프로세스 엘리먼트(PE : Process Element)의 회로구조에 관한 개념도이다.

도 5는 도 6의 회로구조를 폴딩(folding)한 회로구조를 나타내는 구조도이다.

도 6은 본 발명의 일 실시예에 따르는 연산 블록으로 구성된 곱셈기의 회로구조를 나타내는 구조도이다.

도 7a 및 도 7b는 본 발명의 일 실시예에 따르는 프로세스 엘리먼트에 포함되는 곱셈기를 구성하는 연산 블록의 회로구조에 관한 개념도로서, 도 7a는 원시 다항식의 계수의 값 (coefficient of primitive polynomial element)이 0일 때의 회로구조에 관한 것이고, 도 7b는 원시 다항식의 계수의 값이 1일 때의 회로구조에 관한 것이다.

발명을 실시하기 위한 구체적인 내용

[0011] 아래에서는 첨부한 도면을 참조하여 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자가 용이하게 실시할 수 있도록 본 발명의 실시예를 상세히 설명한다. 그러나 본 발명은 여러 가지 상이한 형태로 구현될 수 있으며 여기에서 설명하는 실시예에 한정되지 않는다. 그리고 도면에서 본 발명을 명확하게 설명하기 위해서 설명과 관계없는 부분은 생략하였으며, 명세서 전체를 통하여 유사한 부분에 대해서는 유사한 도면 부호를 붙였다.

[0012] 명세서 전체에서, 어떤 부분이 다른 부분과 "연결"되어 있다고 할 때, 이는 "직접적으로 연결"되어 있는 경우뿐 아니라, 그 중간에 다른 소자를 사이에 두고 "전기적으로 연결"되어 있는 경우도 포함한다. 또한 어떤 부분이 어떤 구성요소를 "포함"한다고 할 때, 이는 특별히 반대되는 기재가 없는 한 다른 구성요소를 제외하는 것이 아니라 다른 구성요소를 더 포함할 수 있는 것을 의미한다.

[0013] 본 명세서에 있어서 '부(部)'란, 하드웨어에 의해 실현되는 유닛(unit), 소프트웨어에 의해 실현되는 유닛, 양방을 이용하여 실현되는 유닛을 포함한다. 또한, 1 개의 유닛이 2 개 이상의 하드웨어를 이용하여 실현되어도 되고, 2 개 이상의 유닛이 1 개의 하드웨어에 의해 실현되어도 된다. 한편, '~부'는 소프트웨어 또는 하드웨어에 한정되는 의미는 아니며, '~부'는 어드레싱 할 수 있는 저장 매체에 있도록 구성될 수도 있고 하나 또는 그 이상의 프로세서들을 재생시키도록 구성될 수도 있다. 따라서, 일 예로서 '~부'는 소프트웨어 구성요소들, 객체지향 소프트웨어 구성요소들, 클래스 구성요소들 및 태스크 구성요소들과 같은 구성요소들과, 프로세스들, 함수들, 속성들, 프로시저들, 서브루틴들, 프로그램 코드의 세그먼트들, 드라이버들, 펌웨어, 마이크로코드, 회로, 데이터, 데이터베이스, 데이터 구조들, 테이블들, 어레이들 및 변수들을 포함한다. 구성요소들과 '~부'들 안에서 제공되는 기능은 더 작은 수의 구성요소들 및 '~부'들로 결합되거나 추가적인 구성요소들과 '~부'들로 더 분리될 수 있다. 뿐만 아니라, 구성요소들 및 '~부'들은 디바이스 또는 보안 멀티미디어카드 내의 하나 또는 그 이상의 CPU들을 재생시키도록 구현될 수도 있다.

[0014] 이하, 도 2를 참조하여, 본 발명의 일 실시예에 따르는 BCH 디코더의 구조에 대하여 구체적으로 설명한다.

[0015] 본 발명의 일 실시예에 따르는 BCH 디코더(10)는 신드롬 연산기(100)(Syndrome Calculator), 키 방정식 연산기(200)(Key Equation Solver), 치엔 탐색기(300)(Chien Search Unit), 에러 정정 처리기(400)(Error Correction Unit)를 포함한다.

[0016] 신드롬 연산기(100)는 수신된 데이터에 대해 신드롬 계산을 통해 에러 발생 유무를 확인하는 모듈이다.

- [0017] 키 방정식 연산기(200)는 신드롬 생성기를 통해 수신된 데이터에 에러가 발견될 경우, 키 방정식(Key Equation)을 계산하는 모듈이다. 여기서 키 방정식은 에러 위치 다항식(error location polynomial)이라고 호칭될 수도 있다.
- [0018] 치엔 탐색기(300)는 키 방정식의 근을 구하여 에러 위치를 검색하는 모듈이다. 이때, 치엔 탐색기(300)는 치엔 탐색 알고리즘을 이용할 수 있다.
- [0019] 에러 정정 처리기(400)는 치엔 탐색기(300)를 통하여 도출된 에러 위치를 참고하여 수신된 데이터의 에러를 정정 처리하는 모듈이다.
- [0020] 본 발명의 일 실시예는 키 방정식 연산기(200)를 구성하는 곱셈기의 구조를 폴딩된 곱셈기 형태로 구성함으로써 종래의 갈로아 필드 곱셈기(GF multiplier)를 이용하던 방식보다 더욱 회로 복잡도가 감소되며 동작 속도가 개선된 BCH 디코더(10)를 제공하는 것을 특징으로 한다.
- [0021] 도 3 및 도 4를 참조하면, 본 발명의 일 실시예에 따르는 키 방정식 연산기(200)는 도 3과 같은 구조로 구성될 수 있다.
- [0022] 즉, 키 방정식 연산기(200)는 복수의 프로세스 엘리먼트(PE : Process Element) (210)로 구성될 수 있다. 각각의 프로세스 엘리먼트(210)들은 키 방정식을 계산하기 위한 모듈 또는 칩으로서, 신드롬 연산기(100)로부터 입력된 값을 처리하기 위해 서로 병렬 또는 직렬로 연결되어 있다.
- [0023] 이때, 각각의 프로세스 엘리먼트(210)는 도 4와 같은 회로 구조로 구성된다. 구체적으로, 두 개의 곱셈기와 하나의 덧셈기 및 하나의 믹스(MUX)(212)로 구성되어, 입력된 값들을 내부적인 연산과정을 거쳐 처리한 후 출력값으로 반환하여 연결되어 있는 다른 프로세스 엘리먼트(210)로 출력한다.
- [0024] 이때, 각각의 곱셈기들은 도 5와 같이 복수의 연산 블록(211), 복수의 플립플롭(213) 및 믹스(212)를 포함할 수 있다.
- [0025] 도 5의 회로구조는 도 6의 회로를 폴딩한 구조로서, 먼저, 도 6의 회로의 연산과정을 설명하면 다음과 같다.
- [0026] 도 6의 회로는 ①~④의 총 4개의 연산 스테이지를 거쳐서 출력값을 반환하는 구조이다. 여기서 연산 스테이지란 연산이 수행되는 단계를 의미한다. 각각의 연산 스테이지는 이전 연산스테이지의 출력값을 입력으로 받아 이를 이용하여 연산을 수행하고, 다음 연산 스테이지로 출력값을 반환하는 형태로 구성되어 있다.
- [0027] 이때, 각각의 연산 스테이지는 4개의 연산 블록(211)으로 구성되는데, 각 연산 블록(211)의 출력값은 도 6에 표시된 하나의 연산 블록(211)에서 2번째 컬럼(column), 3번째 로(row)에 기재된 $S_{i,j}$ ($i, j = 1, 2, 3, \dots$) 형태의 값이 되며, 나머지 컬럼과 로에 기재된 값들은 모두 입력값을 의미한다. 여기서 연산 블록(211)은 곱셈 연산을 수행하기 위하여 AND 게이트 및 XOR 게이트로 구성된 회로를 의미한다.
- [0028] 여기서, 입력값이 3개로 구성되는 연산 블록(211)은 원시 다항식의 계수 값 (coefficient of primitive polynomial element, 이하, 'p'라고 지칭함)이 0이 되는 경우의 연산회로에 관한 것이며, 입력값이 4개로 구성되는 연산 블록(211)은 원시 다항식의 계수 값이 1이되는 경우의 연산회로에 관한 것이다.
- [0029] 도 6에서는 각 연산 스테이지마다 p가 서로 다른 연산 블록(211)들이 구성되며, 구체적으로, $p = 0$ 인 연산 블록(211)과 $p = 1$ 인 연산 블록(211)이 각각 2개로 구성되어 있다.
- [0030] 여기서 $p = 0$ 인 연산 블록(211)은 도 7a와 같이 하나의 AND 게이트와 하나의 XOR 게이트로 구성된다. 구체적으로, 연산 블록(211)에 구비되는 3개의 입력포트 중 2개의 입력포트에 입력되는 2개의 입력값(a_{m-j} , b_{m-i})은 AND 게이트로 입력된다. 그리고, 나머지 1개의 입력포트에 입력되는 입력값($s_{i-1, j+1}$)과 AND 게이트의 출력값은 XOR 게이트로 입력되고, 연산 블록(211)의 출력포트에는 XOR 게이트의 출력값이 출력된다. 즉, XOR 게이트의 출력값이 AND 게이트의 출력값이 종속되도록 구성된다.
- [0031] $p = 1$ 인 연산 블록(211)은 도 7b와 같이 두 개의 XOR 게이트와 한 개의 AND 게이트로 구성된다. 구체적으로, 연산 블록(211)에 구비되는 4개의 입력포트 중 2개의 입력포트에 입력되는 입력값들(a_{m-j} , b_{m-i})은 AND 게이트로 입력된다. 그리고, 나머지 2개의 입력포트에 입력되는 입력값들($s_{i-1, j+1}$, $s_{i-1, j}$)은 제 1 XOR 게이트(G1)로 입력된다. 이어서, 제 2 XOR 게이트(G2)에는 AND 게이트와 제 1 XOR 게이트(G1)의 출력값들이 입력되고, 연산 블록(211)의 출력포트에는 제 2 XOR 게이트(G2)의 출력값이 출력된다. 즉, 제 2 XOR 게이트(G2)의 출력값이 제 1

XOR 게이트(G1)와 AND 게이트의 출력값에 종속되도록 구성된다.

[0032] 이하, 도 6의 회로구조의 연산과정을 구체적으로 설명하면, 먼저, 연산 스테이지에서, 모든 연산 블록(211)들에 대하여 $b_{m-i} = b_3$, $s_{i-1, j+1} = 0$ 으로 설정된다. 그리고, 모든 연산 블록(211)들에 대하여 a_{m-j} 은, a_0 , a_1 , a_2 , a_3 중 각 연산 블록(211)에 대응하는 어느 하나의 값을 가지도록 설정된다. 이어서, 연산 스테이지의 연산 결과로 출력된 값들 ($s_{1,4}$, $s_{1,3}$, $s_{1,2}$, $s_{1,1}$)은 연산 스테이지에 포함된 연산 블록(211)으로 입력된다. 이러한 출력값들 중 $s_{1,1}$ 은 $p=1$ 인 연산 블록(211)의 $s_{i-1, 1}$ 로 입력되되, 나머지 출력값들($s_{1,4}$, $s_{1,3}$, $s_{1,2}$)은 $s_{i-1, j+1}$ 로 입력된다. 연산 스테이지에서는 입력된 값들을 기준으로 연산을 수행하며, 그에 따르는 출력값들은 연산 스테이지로 입력된다. 이와 동일한 방식으로 각 연산 스테이지의 b_{m-i} 값만 달라진 채 및 연산 스테이지의 연산이 수행된다. 그리고, 연산 스테이지에서는 최종 출력값들($s_{4,4}$, $s_{4,3}$, $s_{4,2}$, $s_{4,1}$)이 곱셈기의 출력포트로 출력된다.

[0033] 이어서, 도 5를 참조하여, 본 발명의 일 실시예에 따르는 곱셈기의 연산 과정을 구체적으로 설명한다.

[0034] 도 5에서 믹스(212)는 b_0 , b_1 , b_2 , b_3 값들 중 어느 하나의 값들만을 출력하며, 믹스(212)는 모든 연산 블록(211)과 병렬적으로 연결되어 있기 때문에, 믹스(212)의 출력하는 제 1 입력값은 모든 연산 블록(211a~211d)에 동시에 입력된다. 이때, 믹스(212)에서 어떤 값을 출력하느냐에 따라 연산 스테이지가 결정된다. 구체적으로, 믹스(212)가 b_0 을 출력하는 경우, 도 6에서 설명한 연산 스테이지의 연산이 실행된다. 즉, 믹스(212)는 b_0 을 출력하여 연산 스테이지의 연산이 종료되면, b_1 을 출력하여 그 다음 연산 스테이지가 진행되도록 한다. 이러한 방식으로, 믹스(212)는 각 연산 스테이지에 맞는 입력값을 출력한다.

[0035] 연산 블록들(211)은 제 1 연산 블록(211a), 제 2 연산 블록(211b), 제 3 연산 블록(211c), 제 4 연산 블록(211d)으로 구성된다. 도면에서는 4개의 연산 블록(211a~211d)으로 구성된 예가 도시되었으나, 연산 블록의 개수는 다른 개수로도 구성될 수 있다.

[0036] 여기서, 제 1 및 제 2 연산 블록(211a, 211b)은 $p=1$ 인 연산 블록이며, 제 3 및 제 4 연산 블록(211c, 211d)은 $p=0$ 인 연산 블록이다.

[0037] 각 연산 블록(211a, 211b, 211c 또는 211d)은 출력값이 다른 연산 블록의 입력값으로 입력되도록, 다른 연산 블록과 연결된다. 예를 들어, 제 1 연산 블록(211a)의 출력값(so_0)은 제 2 연산 블록(211b)의 제 3 입력값(si_0)으로 입력되며, 제 2 연산 블록(211b)의 출력값(so_1)은 제 3 연산 블록(211c)의 제 3 입력값(si_1)으로 입력되고, 제 3 연산 블록(211c)의 출력값(so_2)은 제 4 연산 블록(211d)의 제 3 입력값(si_2)으로 입력되고, 제 4 연산 블록(211d)의 출력값(so_3)은 제 1 및 제 2 연산 블록(211b)의 제 2 입력값(si_3)으로 입력되도록 구성된다. 한편, 다른 연산 블록(211)들과 달리 제 4 연산 블록(211d)의 경우 출력값(so_3)이 제 1 및 제 2 연산 블록(211b)의 si_3 입력 포트에 입력되도록 구성된다. 그리고, 각 연산 블록(211a, 211b, 211c 또는 211d)에는 제 4 입력값으로서, a_0 , a_1 , a_2 , a_3 중 어느 하나에 해당하는 값이 미리 설정되어 있다.

[0038] 한편, 믹스(212)가 출력하는 제 1 입력값이 마지막 값인 경우(즉, $b=3$), 각 연산 블록(211)은 연산 결과에 따르는 출력값을 곱셈기의 최종 출력값으로 제공한다.

[0039] 플립플롭(213)은 클럭 신호에 동기화 하여 입력값을 출력하는 구성으로서, 플립플롭(213)은 각 연산 블록(211a, 211b, 211c 또는 211d)의 출력포트와 다른 연산 블록(211a, 211b, 211c 또는 211d)의 입력포트 사이에 연결된다. 플립플롭(213)은 모든 연산 블록(211)의 출력값들을 클럭 신호에 동기화하여 한번에 다른 연산 블록(211)의 입력포트로 입력함으로써 각각의 연산 스테이지의 연산이 원활히 이루어지도록 할 수 있다. 플립플롭(213)의 종류는 바람직하게 D 플립 플롭일 수 있다.

[0040] 곱셈기의 회로구조가 도 6의 회로구조를 도 5와 같이 폴딩하여 구성되는 경우, 4개의 연산 블록(211a~211d)만으로 연산 스테이지의 연산이 종료된 후, 각 연산 블록(211)들은 연산 스테이지로 진입하여 연산을 수행할 수 있게 된다. 즉, 이와 같은 형태의 연산 블록 연결 구조를 통해, 단 4개의 연산 블록(211a~211d)과 믹스(212) 및 플립플롭(213)을 구성하는 것만으로도 도 6과 같이 16개의 연산 블록(211)을 구성하여 회로를 구성하는 경우와 동일한 연산을 수행할 수 있다.

[0041] 도 1과 같은 종래기술의 경우, 곱셈기의 회로구조가 매우 불규칙적으로 배열되어 있어, 폴딩이 어렵다. 그러나, 본 발명의 일 실시예의 곱셈기는 각각의 연산 블록(211)이 도 6과 같은 규칙적인 조합으로 곱셈 연산을 수행할

수 있도록 하되, 도 6의 구조를 폴딩하여 도 5와 같은 구조를 달성함으로써, 계산 경로(critical path)를 짧게 하여 연산 속도를 상승시킬 수 있다. 또한, GF dimension이 커져도 동일한 동작속도를 유지할 수 있어 고속 BCH 동작에 매우 유리하다. 또한, 연산 블록(211)의 개수를 최소화하는 구조이기 때문에 곱셈기의 저면적화를 달성함으로써 BCH 디코더(10)의 면적을 최소화할 수 있다.

[0042] 본 발명의 일 실시예는 컴퓨터에 의해 실행되는 프로그램 모듈과 같은 컴퓨터에 의해 실행가능한 명령어를 포함하는 기록 매체의 형태로도 구현될 수 있다. 컴퓨터 판독 가능 매체는 컴퓨터에 의해 액세스될 수 있는 임의의 가용 매체일 수 있고, 휘발성 및 비휘발성 매체, 분리형 및 비분리형 매체를 모두 포함한다. 또한, 컴퓨터 판독 가능 매체는 컴퓨터 저장 매체 및 통신 매체를 모두 포함할 수 있다. 컴퓨터 저장 매체는 컴퓨터 판독가능 명령어, 데이터 구조, 프로그램 모듈 또는 기타 데이터와 같은 정보의 저장을 위한 임의의 방법 또는 기술로 구현된 휘발성 및 비휘발성, 분리형 및 비분리형 매체를 모두 포함한다. 통신 매체는 전형적으로 컴퓨터 판독가능 명령어, 데이터 구조, 프로그램 모듈, 또는 반송파와 같은 변조된 데이터 신호의 기타 데이터, 또는 기타 전송 매커니즘을 포함하며, 임의의 정보 전달 매체를 포함한다.

[0043] 본 발명의 방법 및 시스템은 특정 실시예와 관련하여 설명되었지만, 그것들의 구성 요소 또는 동작의 일부 또는 전부는 범용 하드웨어 아키텍처를 갖는 컴퓨터 시스템을 사용하여 구현될 수 있다.

[0044] 전술한 본 발명의 설명은 예시를 위한 것이며, 본 발명이 속하는 기술분야의 통상의 지식을 가진 자는 본 발명의 기술적 사상이나 필수적인 특징을 변경하지 않고서 다른 구체적인 형태로 쉽게 변형이 가능하다는 것을 이해할 수 있을 것이다. 그러므로 이상에서 기술한 실시예들은 모든 면에서 예시적인 것이며 한정적이 아닌 것으로 이해해야만 한다. 예를 들어, 단일형으로 설명되어 있는 각 구성 요소는 분산되어 실시될 수도 있으며, 마찬가지로 분산된 것으로 설명되어 있는 구성 요소들도 결합된 형태로 실시될 수 있다.

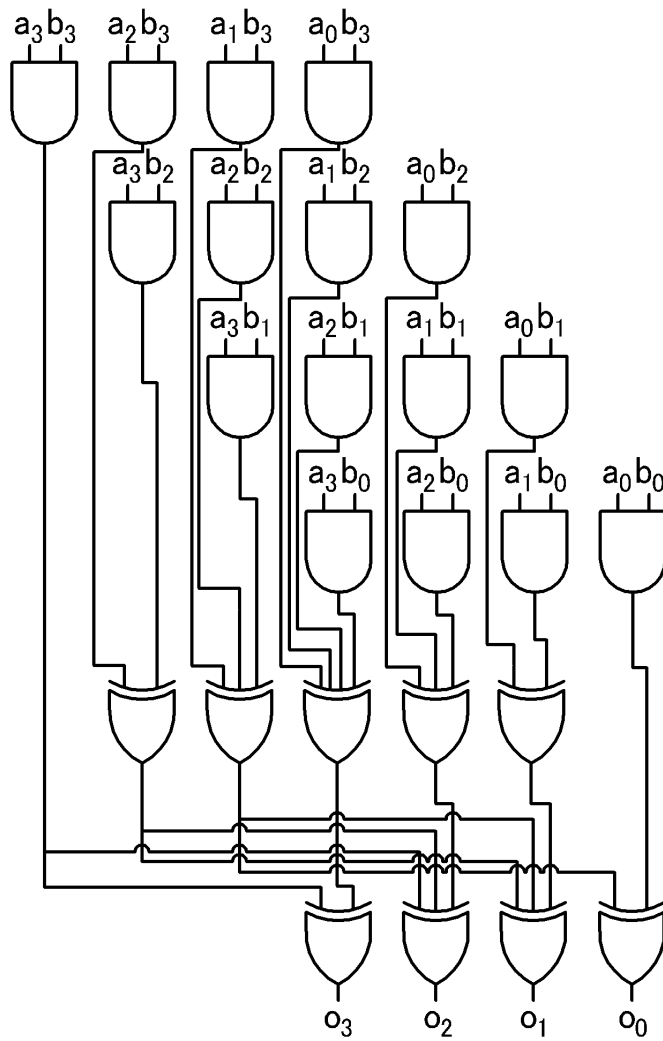
[0045] 본 발명의 범위는 상기 상세한 설명보다는 후술하는 특허청구범위에 의하여 나타내어지며, 특허청구범위의 의미 및 범위 그리고 그 균등 개념으로부터 도출되는 모든 변경 또는 변형된 형태가 본 발명의 범위에 포함되는 것으로 해석되어야 한다.

부호의 설명

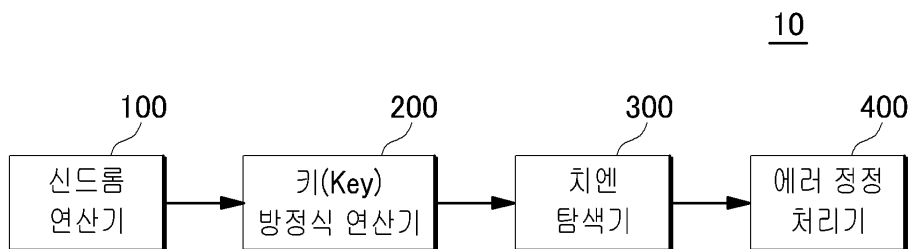
[0046] 10: BCH 디코더 200: 키 방정식 연산기
210: 프로세스 엘리먼트 211: 연산 블록
212: 믹스 213: 플립플롭

도면

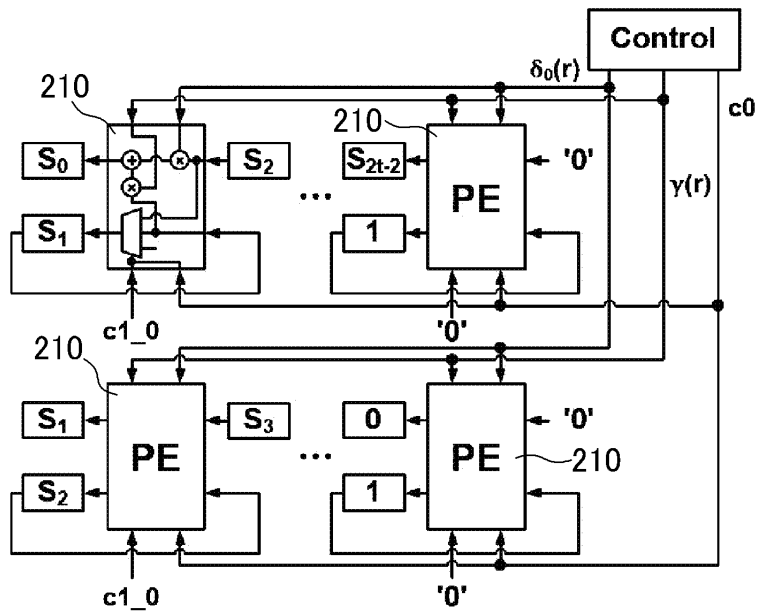
도면1



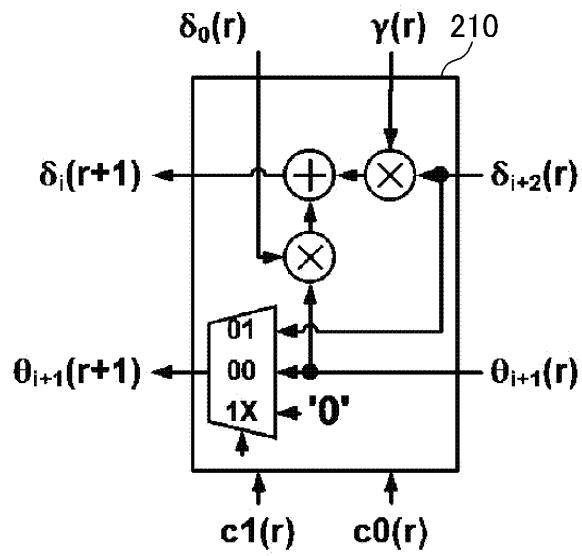
도면2



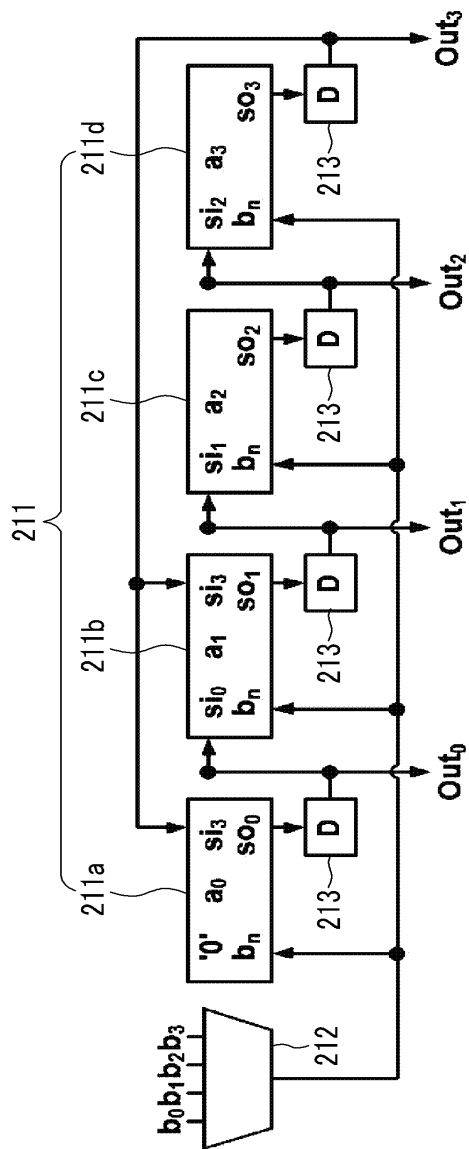
도면3



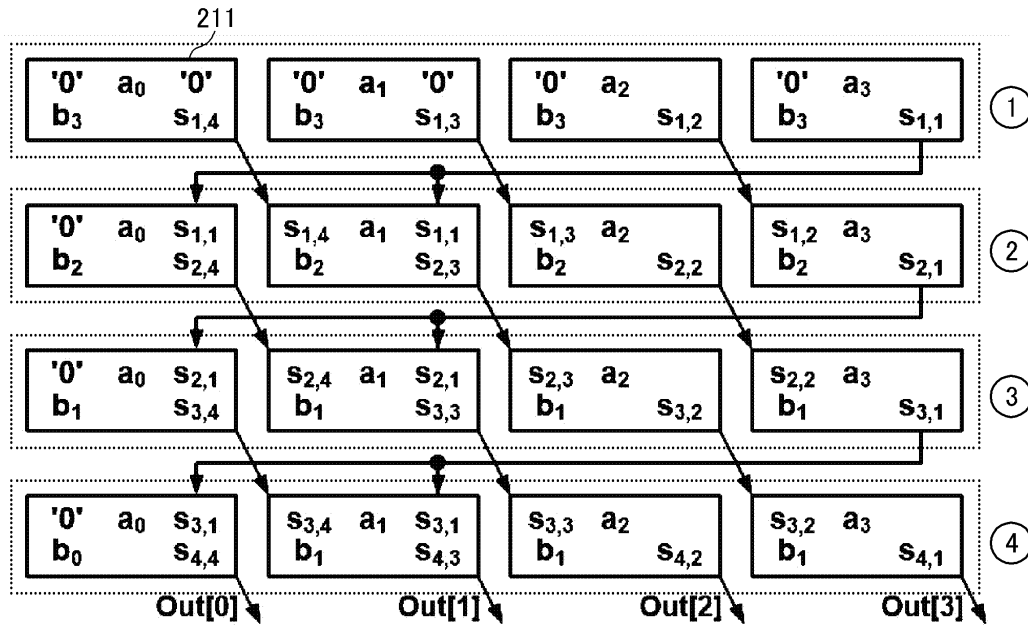
도면4



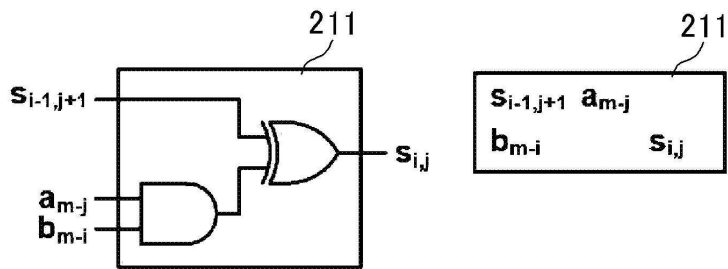
도면5



도면6



도면7a



도면7b

