



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2018년07월26일
(11) 등록번호 10-1882620
(24) 등록일자 2018년07월20일

(51) 국제특허분류(Int. Cl.)
H03M 13/15 (2015.01) H03M 13/00 (2017.01)
(52) CPC특허분류
H03M 13/152 (2013.01)
H03M 13/6502 (2013.01)
(21) 출원번호 10-2017-0097563
(22) 출원일자 2017년08월01일
심사청구일자 2017년08월01일
(56) 선행기술조사문헌
KR101569637 B1
Low-Complexity Non-Iterative Soft-Decision
BCH Decoder Architecture for WBAN
Applications (JOURNAL OF SEMICONDUCTOR
TECHNOLOGY AND SCIENCE, 2016년 8월)

(73) 특허권자
인하대학교 산학협력단
인천광역시 남구 인하로 100, 인하대학교 (용현동)
(72) 발명자
이한호
인천광역시 연수구 아트센터대로97번길 30, 1602동 (송도동, 더샵그린워크1차)
김태성
서울특별시 영등포구 여의대방로43나길 25, 102동403호(신길동, 삼환아파트)
(74) 대리인
양성보

전체 청구항 수 : 총 6 항

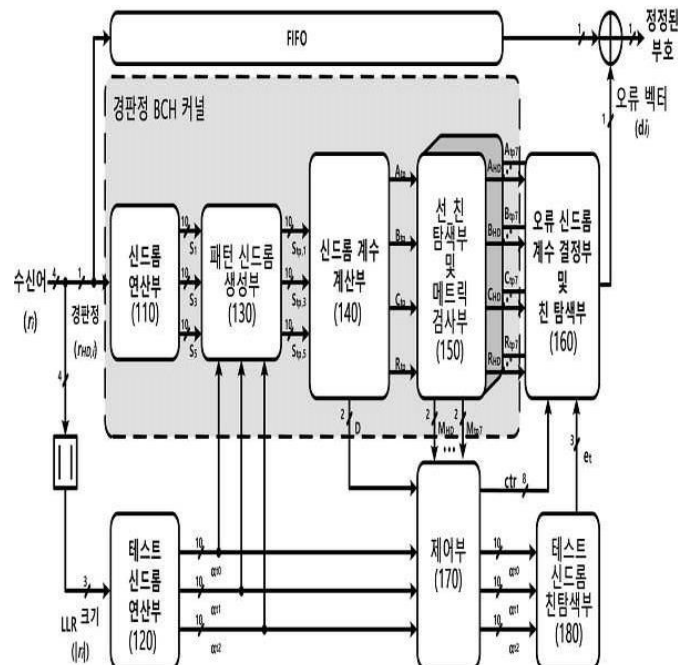
심사관 : 조춘근

(54) 발명의 명칭 신드롬 기반 연관정 BCH 복호 장치 및 그 방법

(57) 요약

신드롬 기반 연관정 BCH 복호 장치 및 그 방법이 제시된다. 본 발명에서 제안하는 테스트 신드롬을 이용한 연관정 BCH 복호 장치는 코드워드를 최소 신뢰 비트의 개수로 분할한 각 구간에 대하여 최소 신뢰 비트의 위치를 한 개씩 추출하고, 테스트 신드롬으로 변환하는 테스트 신드롬 연산부, 신드롬 연산부로부터의 신드롬 및 상기 변환 (뒷면에 계속)

대표도



된 테스트 신드롬을 입력 받아 패턴 신드롬을 생성하는 패턴 신드롬 생성부, 상기 생성된 패턴 신드롬을 입력 받아 후보 코드워드의 신드롬 계수를 연산하고, 신드롬 행렬의 각 차수에 대하여 신드롬 계수의 공통 부분을 공유하면서 모든 차수에 대한 신드롬 계수를 별도로 연산하는 신드롬 계수 계산부, 상기 패턴 신드롬을 입력 받아 오류의 수를 합 연산하는 선-친 탐색부 및 메트릭(Metric) 검사부 및 제어신호를 통해 상기 후보 코드워드의 신드롬 계수를 결정하고, 최소 신뢰 비트의 연산 결과를 선택하며, 오류를 정정하기 위해 벡터를 연산하는 오류 신드롬 계수 결정부 및 친 탐색부를 포함한다.

이 발명을 지원한 국가연구개발사업

과제고유번호 IITP-2017-2014-0-00729

부처명 미래창조과학부

연구관리전문기관 정보통신기술진흥센터

연구사업명 대학ICT연구센터육성 지원사업

연구과제명 스마트한 주파수 이용을 위한 스펙트럼 및 전파기술

기 여 율 1/1

주관기관 인하대학교 산학협력단

연구기간 2017.01.01 ~ 2017.12.31

공지예외적용 : 있음

명세서

청구범위

청구항 1

테스트 신드롬을 이용한 연관정 BCH 복호 장치에 있어서,

체이스 알고리즘에 기초하여 코드워드의 LLR의 크기를 비교하고, 상기 코드워드를 최소 신뢰 비트의 개수로 분할한 각 구역마다 한 개의 최소 신뢰 비트를 추출하여 최소 신뢰 비트의 위치를 테스트 신드롬으로 변환하는 테스트 신드롬 연산부;

신드롬 연산부로부터의 신드롬 및 상기 변환된 테스트 신드롬을 입력 받아 패턴 신드롬을 생성하는 패턴 신드롬 생성부;

상기 생성된 패턴 신드롬을 입력 받아 후보 코드워드의 신드롬 계수를 연산하고, 신드롬 행렬의 각 차수에 대하여 신드롬 계수의 공통 부분을 공유하면서 모든 차수에 대한 신드롬 계수를 별도로 연산하는 신드롬 계수 계산부;

상기 패턴 신드롬을 입력 받아 오류의 수를 합 연산하는 선-친 탐색부 및 메트릭(Metric) 검사부; 및

제어신호를 통해 상기 후보 코드워드의 신드롬 계수를 결정하고, 연관정을 위한 최소 신뢰 비트를 선택하며, 선택된 최소 신뢰 비트를 이용하여 오류를 정정하기 위해 벡터를 연산하는 오류 신드롬 계수 결정부 및 친 탐색부를 포함하는 테스트 신드롬을 이용한 연관정 BCH 복호 장치.

청구항 2

삭제

청구항 3

제 1항에 있어서,

상기 테스트 신드롬 연산부는,

별도의 정렬 연산 과정을 필요로 하지 않고 상기 테스트 신드롬을 쉬프트 레지스터(Shift register)로 입력함으로써 테스트 신드롬 연산을 수행하는

테스트 신드롬을 이용한 연관정 BCH 복호 장치.

청구항 4

제 1항에 있어서,

상기 패턴 신드롬 생성부는,

신드롬 연산부로부터의 신드롬 및 상기 변환된 테스트 신드롬을 입력 받아 모든 후보 코드워드에 대한 신드롬을 D-플립플롭과 피드백 회로를 이용하여 순차적으로 계산하는

테스트 신드롬을 이용한 연관정 BCH 복호 장치.

청구항 5

제 1항에 있어서,

상기 오류 신드롬 계수 결정부 및 친 탐색부는,

제어신호를 통해 후보 코드워드의 신드롬 계수를 결정하여 친 탐색부로 상기 결정된 신드롬 계수를 전달하고, 연관정을 위한 최소 신뢰 비트의 연산 결과를 선택하여 친 탐색부의 결과와 선택된 최소 신뢰 비트의 연산 결과를 GF 덧셈기를 이용하여 합 연산을 수행한 후, 코드워드의 오류를 정정하기 위한 오류 벡터를 생성하는

테스트 신드롬을 이용한 연관정 BCH 복호 장치.

청구항 6

제 1항에 있어서,

상기 선-친 탐색부 및 메트릭 검사부는,

오류의 발생 여부에 대한 출력없이 신드롬 계수로부터 발생하는 오류의 수를 합 연산하고, 후보 코드워드의 신드롬 계수를 입력 받아 상기 후보 코드워드의 신드롬 계수를 출력하는

테스트 신드롬을 이용한 연관정 BCH 복호 장치.

청구항 7

테스트 신드롬을 이용한 연관정 BCH 복호 방법에 있어서,

체이스 알고리즘에 기초하여 코드워드의 LLR의 크기를 비교하고, 상기 코드워드를 최소 신뢰 비트의 개수로 분할한 각 구역마다 한 개의 최소 신뢰 비트를 추출하여 최소 신뢰 비트의 위치를 테스트 신드롬으로 변환하는 단계;

신드롬 연산부로부터의 신드롬 및 상기 변환된 테스트 신드롬을 입력 받아 패턴 신드롬을 생성하는 단계;

상기 생성된 패턴 신드롬을 입력 받아 후보 코드워드의 신드롬 계수를 연산하고, 신드롬 행렬의 각 차수에 대하여 신드롬 계수의 공통 부분을 공유하면서 모든 차수에 대한 신드롬 계수를 별도로 연산하는 단계;

상기 패턴 신드롬을 입력 받아 오류의 수를 합 연산하는 단계; 및

제어신호를 통해 상기 후보 코드워드의 신드롬 계수를 결정하고, 연관정을 위한 최소 신뢰 비트를 선택하며, 선택된 최소 신뢰 비트를 이용하여 오류를 정정하기 위해 벡터를 연산하는 단계

를 포함하는 테스트 신드롬을 이용한 연관정 BCH 복호 방법.

발명의 설명

기술 분야

[0001] 본 발명은 신드롬(Syndromes)과 경판정 커널(hard-decision kernel)을 이용하고 복잡도가 낮은 연관정(soft-decision) BCH(Bose-Chaudhuri-Hocquenghem) 복호 방법 및 장치에 관한 것이다.

배경 기술

[0002] 최근 근거리 무선통신, 메모리시스템, 광통신시스템의 데이터 전송에서 송 수신 중에 발생하는 오류를 찾고 정정하는 순방향 오류 정정 기술에 BCH 부호가 널리 사용되고 있다.

[0003] BCH부호는 $BCH(n, k, t)$ 로 표현하는데 n 은 데이터 및 패리티가 포함된 부호어(codeword) 심볼(symbol)의 수이고, k 는 데이터의 심볼 수, t 는 부호어에서 정정 가능한 최대 오류 심볼 수를 나타낸다. 정정 가능한 최대 오류 심볼 수는 갈루아 체(Galois field)의 지수 m 으로부터 수식 $n-k \geq mt$ 의 관계로 정의된다. 보통, 경판정(hard-decision) BCH 부호는 각 심볼이 1비트(bit)인 바이너리(binary) BCH 부호를 의미한다.

[0004] 기본의 경판정 BCH 부호의 복호는 BM(Berlekamp-Massey) 알고리즘과 ME(Modified Euclidean) 알고리즘, 수정된 SBS(Modified Step-by-step) 알고리즘 등이 사용되어 왔다. 특히 수정된 SBS 알고리즘은 BM, ME 알고리즘에 비해 정정 가능한 오류의 수가 4개 이하일 때($t \leq 4$) 귀환(Feedback) 회로와 지연(Latency) 없이 신드롬 행렬식 연산이 가능하다. 이에 연관정 부호와 관련된 신드롬을 순차적으로 생성하고 이를 그대로 행렬식 계산에 적용 가능하다.

[0005] 연관정 BCH 부호는 채널 정보를 LLR(Log Likelihood Ratio)을 이용하여 측정한다. 크기가 가장 작은 p 개의 비트는 오류가 발생할 가능성이 제일 높다고 판단하며 이를 최소 신뢰 비트(Least Reliability Bit)라 한다. 최소 신뢰 비트는 $p = \lceil d_{\min}/2 \rceil$ 로 정의되며, 때문에 정정 가능한 최대 심볼의 수는 $t + p$ 가 된다. 따라서 연관정 BCH 부호는 경판정 BCH 부호보다 비트오류율(Bit Error Rate, BER)이 낮아 경판정 BCH 부호보다 높은 성능을 발휘한다.

다.

발명의 내용

해결하려는 과제

- [0006] 본 발명이 이루고자 하는 기술적 과제는 연관정 BCH 복호기의 오류 정정 능력을 강화하면서 면적 효율적인 설계를 위해, 경관정 BCH 복호를 커널로 이용하고 테스트 패턴 생성기(Test Pattern Generator)를 패턴 신드롬 생성부(Pattern Syndrome Generator)로 대체함으로써, 낮은 하드웨어 복잡도를 갖는 고성능 연관정 BCH 복호 장치 및 그 방법을 제공하는데 있다. 또한, 기존의 연관정 BCH 복호기의 차수(정정 오류 심볼의 수)를 증가시킴에 따라 필요 연산을 공유하면 복잡도가 지수적으로 증가하는 문제점을 개선하기 위해 경관정 부호어의 신드롬을 기준으로 후보 코드워드(Candidate codeword)의 신드롬 다시 말해, 패턴 신드롬을 생성하는 패턴 신드롬 생성부를 구성하여, 면적의 증가를 억제하면서 경관정 커널을 연관정 부호의 복호에 적용하고자 한다. 또한, 코드워드의 길이가 복호기의 복잡도에 영향을 주지 않는 연관정 BCH 복호 장치를 제공하기 위해, D-플립 플롭이 포함된 오류 위치 결정부를 오류 신드롬 계수 결정부로 대체한다.

과제의 해결 수단

- [0007] 일 측면에 있어서, 본 발명에서 제안하는 테스트 신드롬을 이용한 연관정 BCH 복호 장치는 코드워드를 최소 신뢰 비트의 개수로 분할한 각 구간에 대하여 최소 신뢰 비트의 위치를 한 개씩 추출하고, 테스트 신드롬으로 변환하는 테스트 신드롬 연산부, 신드롬 연산부로부터의 신드롬 및 상기 변환된 테스트 신드롬을 입력 받아 패턴 신드롬을 생성하는 패턴 신드롬 생성부, 상기 생성된 패턴 신드롬을 입력 받아 후보 코드워드의 신드롬 계수를 연산하고, 신드롬 행렬의 각 차수에 대하여 신드롬 계수의 공통 부분을 공유하면서 모든 차수에 대한 신드롬 계수를 별도로 연산하는 신드롬 계수 계산부, 상기 패턴 신드롬을 입력 받아 오류의 수를 합 연산하는 선-친 탐색부 및 메트릭(Metric) 검사부 및 제어신호를 통해 상기 후보 코드워드의 신드롬 계수를 결정하고, 최소 신뢰 비트의 연산 결과를 선택하며, 오류를 정정하기 위해 벡터를 연산하는 오류 신드롬 계수 결정부 및 친 탐색부를 포함한다.
- [0008] 상기 테스트 신드롬 연산부는 체이스 알고리즘에 기초하여 채널로부터 수신된 심볼의 LLR의 크기를 비교하고, 상기 심볼을 최소 신뢰 비트의 개수로 분할한 각 구역마다 한 개의 최소 신뢰 비트를 추출하여 최소 신뢰 비트의 위치를 테스트 신드롬으로 변환한다.
- [0009] 상기 테스트 신드롬 연산부는 별도의 정렬 연산 과정을 필요로 하지 않고 상기 테스트 신드롬 값을 쉬프트 레지스터(Shift register)로 입력함으로써 테스트 신드롬 연산을 수행한다.
- [0010] 상기 패턴 신드롬 생성부는 신드롬 연산부로부터의 신드롬 및 상기 변환된 테스트 신드롬을 입력 받아 모든 후보 코드워드에 대한 신드롬을 D-플립플롭과 피드백 회로를 이용하여 순차적으로 계산한다.
- [0011] 상기 오류 신드롬 계수 결정부 및 친 탐색부는 제어신호를 통해 후보 코드워드의 신드롬 계수를 결정하여 친 탐색부로 상기 결정된 신드롬 계수를 전달하고, 연관정을 위한 최소 신뢰 비트의 연산 결과를 선택하여 친 탐색부의 결과와 선택된 최소 신뢰 비트의 연산 결과를 GF(Galois field) 덧셈기를 이용하여 합 연산을 수행한 후, 수신어의 오류를 정정하기 위한 오류 벡터를 생성한다.
- [0012] 상기 선-친 탐색부 및 메트릭검사부는 오류의 발생 여부에 대한 출력없이 신드롬 계수로부터 발생하는 오류의 수를 합 연산하고, 후보 코드워드의 신드롬 계수를 입력 받아 상기 신드롬 계수를 출력한다.
- [0013] 또 다른 일 측면에 있어서, 본 발명에서 제안하는 테스트 신드롬을 이용한 연관정 BCH 복호 방법은 코드워드를 최소 신뢰 비트의 개수로 분할한 각 구간에 대하여 최소 신뢰 비트의 위치를 한 개씩 추출하고, 테스트 신드롬으로 변환하는 단계, 신드롬 연산부로부터의 신드롬 및 상기 변환된 테스트 신드롬을 입력 받아 패턴 신드롬을 생성하는 단계, 상기 생성된 패턴 신드롬을 입력 받아 후보 코드워드의 신드롬 계수를 연산하고, 신드롬 행렬의 각 차수에 대하여 신드롬 계수의 공통 부분을 공유하면서 모든 차수에 대한 신드롬 계수를 별도로 연산하는 단계, 상기 패턴 신드롬을 입력 받아 오류의 수를 합 연산하는 단계 및 제어신호를 통해 상기 후보 코드워드의 신드롬 계수를 결정하고, 최소 신뢰 비트의 연산 결과를 선택하며, 오류를 정정하기 위해 벡터를 연산하는 단계를 포함한다.

발명의 효과

[0014] 본 발명의 실시예들에 따르면 연관정 BCH 복호기 설계에 있어서 테스트 신드롬과 패턴 신드롬을 사용함으로써, 2^p 개의 후보 코드워드를 생성할 필요 없이 후보 코드워드에 대한 신드롬 다시 말해, 패턴 신드롬을 생성하고 경관정 BCH 복호기 구조를 사용하여 낮은 하드웨어 복잡도를 가지는 효율적인 연관정 BCH 복호 장치 및 그 방법을 제공할 수 있다. 또한, 선-친 검사부와 오류 신드롬 계수 결정부를 적용함으로써 모든 후보 코드워드의 경관정 오류 위치를 저장하지 않는다. 그리고, 정정 가능한 후보 코드워드의 오류 신드롬 계수만을 결정함으로써 코드워드의 길이와 후보 코드워드의 개수의 곱만큼 D-플립플롭을 제거하여 코드워드의 길이와 하드웨어 복잡도가 서로 독립적인 연관정 BCH 복호 장치 및 그 방법을 제공할 수 있다.

도면의 간단한 설명

[0015] 도 1은 본 발명의 일 실시예에 따른 테스트 신드롬을 이용한 연관정 BCH 복호 장치의 구성을 나타내는 도면이다.

도2는 본 발명의 일 실시예에 따른 최소 신뢰 비트의 위치를 추출하는 테스트 신드롬 연산부를 나타내는 도면이다.

도 3은 본 발명의 실시예에 따른 패턴 신드롬 생성부를 나타내는 도면이다.

도 4은 본 발명의 실시예에 따른 신드롬 계수 계산부를 나타내는 도면이다.

도 5은 본 발명의 실시예에 따른 선-친 탐색부 및 메트릭 검사부를 나타내는 도면이다.

도 6은 본 발명의 실시예에 따른 테스트 신드롬 친 탐색부를 나타내는 도면이다.

도 7은 본 발명의 실시예에 따른 오류 신드롬 계수 결정부 및 친 탐색부를 나타내는 도면이다.

도 8은 본 발명의 일 실시예에 따른 테스트 신드롬을 이용한 연관정 BCH 복호 방법을 설명하기 위한 흐름도이다.

도 9는 본 발명의 일 실시예에 따른 비트 오류율을 나타내는 도면이다.

발명을 실시하기 위한 구체적인 내용

[0016] 본 발명은 신드롬(Syndromes)과 경관정 커널(hard-decision kernel)을 이용해 복잡도가 낮은 연관정(soft-decision) BCH (Bose-Chaudhuri-Hocquenghem) 복호 방법 및 장치에 관한 것이다. 더욱 상세하게 연관정 BCH 부호를 경관정 BCH 부호 구조의 커널(kernel)을 이용해 복호하며 테스트 신드롬 연산부, 패턴 신드롬 생성부, 신드롬 계수 계산부, 선-친 탐색부와 오류 신드롬 계수 결정부를 통한 낮은 복잡도 BCH 복호 방법 및 장치에 관한 것이다. 이하, 본 발명의 실시 예를 첨부된 도면을 참조하여 상세하게 설명한다.

[0017] 도 1은 본 발명의 일 실시예에 따른 테스트 신드롬을 이용한 연관정 BCH 복호 장치의 구성을 나타내는 도면이다.

[0018] 제안하는 테스트 신드롬을 이용한 연관정 BCH 복호 장치는 신드롬 연산부(110), 코드워드를 최소 신뢰 비트의 개수로 분할한 각 구간에 대하여 최소 신뢰 비트의 위치를 한 개씩 추출하고, 테스트 신드롬으로 변환하는 테스트 신드롬 연산부(120), 신드롬 연산부로부터의 신드롬 및 상기 변환된 테스트 신드롬을 입력 받아 패턴 신드롬을 생성하는 패턴 신드롬 생성부(130), 상기 생성된 패턴 신드롬을 입력 받아 후보 코드워드의 신드롬 계수를 연산하고, 신드롬 행렬의 각 차수에 대하여 신드롬 계수의 공통 부분을 공유하면서 모든 차수에 대한 신드롬 계수를 별도로 연산하는 신드롬 계수 계산부(140), 상기 패턴 신드롬을 입력 받아 오류의 수를 합 연산하는 선-친 탐색부 및 메트릭(Metric) 검사부(150), 제어신호를 통해 상기 후보 코드워드의 신드롬 계수를 결정하고, 최소 신뢰 비트의 연산 결과를 선택하며, 오류를 정정하기 위해 벡터를 연산하는 오류 신드롬 계수 결정부 및 친 탐색부(160), 제어부(170) 및 테스트 신드롬 탐색부(180)를 포함한다.

[0019] 상기 테스트 신드롬 연산부(120)는 체이스 알고리즘에 기초하여 채널로부터 수신된 심볼의 LLR의 크기를 비교하고, 상기 심볼을 최소 신뢰 비트의 개수로 분할한 각 구역마다 한 개의 최소 신뢰 비트를 추출하여 최소 신뢰 비트의 위치를 테스트 신드롬으로 변환한다. 이때, 테스트 신드롬 연산부(120)는 별도의 정렬 연산 과정을 필요로 하지 않고 상기 테스트 신드롬 값을 쉬프트 레지스터(Shift register)로 입력함으로써 테스트 신드롬 연산을 수행한다.

- [0020] 상기 패턴 신드롬 생성부(130)는 신드롬 연산부로부터의 신드롬 및 상기 변환된 테스트 신드롬을 입력 받아 모든 후보 코드워드에 대한 신드롬을 D-플립플롭과 피드백 회로를 이용하여 순차적으로 계산한다.
- [0021] 상기 오류 신드롬 계수 결정부 및 친 탐색부(160)는 제어신호를 통해 후보 코드워드의 신드롬 계수를 결정하여 친 탐색부로 상기 결정된 신드롬 계수를 전달하고, 연관정을 위한 최소 신뢰 비트의 연산 결과를 선택하여 친 탐색부의 결과와 선택된 최소 신뢰 비트의 연산 결과를 GF 덧셈기를 이용하여 합 연산을 수행한 후, 수신어의 오류를 정정하기 위한 오류 벡터를 생성한다.
- [0022] 상기 선-친 탐색부 및 메트릭 검사부(150)는 오류의 발생 여부에 대한 출력없이 신드롬 계수로부터 발생하는 오류의 수를 합 연산하고, 후보 코드워드의 신드롬 계수를 입력 받아 상기 신드롬 계수를 출력한다. 테스트 신드롬을 이용한 연관정 BCH 복호 장치에 대하여 아래에서 더욱 상세히 설명한다.
- [0023] 본 발명의 실시예에 따른 테스트 신드롬을 이용한 연관정 BCH 복호 장치는 디지털 통신 시스템의 송신단에서 데이터 송신 과정 중 데이터에 발생한 오류를 수신단에서 정정하는 순방향 오류 정정(Forward Error Correction; FEC) 시스템에 관한 것으로, 신드롬 연산부(110), 크기가 가장 작은 LLR의 위치를 추출하는 테스트 신드롬 연산부(120), 신드롬과 테스트 신드롬을 입력으로 받아 패턴 신드롬을 생성하는 패턴 신드롬 생성부(130), 패턴 신드롬을 입력으로 받아 후보코드의 신드롬 계수를 연산하는 신드롬 계수 계산부(140), 후보코드의 신드롬 계수의 정보를 측정하는 선-친 탐색부 및 메트릭(Metric) 검사부(150), 제어부(170), 테스트 신드롬으로부터 LLR의 위치를 복원하는 테스트 신드롬 친 탐색부(180), 제어신호에 의해 정정 가능한 후보코드의 신드롬 계수를 결정하는 오류 신드롬 계수 결정부 및 친 탐색부(160)를 포함하는 연관정 BCH 복호 장치 및 방법에 관한 것이다.
- [0024] 본 발명의 실시예는 (1020, 990, 3) 부호를 기준으로 설명하는 것이며, 본 발명은 다양한 통신 시스템을 위한 연관정 BCH 복호기 구조의 구현에도 동일하게 적용될 수 있다. 또한, 본 발명의 실시예에 따르면 연관정 BCH 부호의 복호를 위한 알고리즘으로서 정정 가능한 오류의 개수가 4개 이하일 때 복잡도가 낮은 수정된 SBS 알고리즘이 사용될 수 있다.
- [0025] 본 발명에서 제안하는 연관정 BCH 복호 장치 및 방법은 후보 코드워드에 대한 연산을 신드롬을 기반으로 연산하며 모든 후보코드의 신드롬 다시 말해, 패턴 신드롬을 테스트 신드롬과 경관정 신드롬을 이용하여 생성한다. 그리고 제안하는 연관정 BCH 복호 장치는 후보 코드워드의 오류 위치를 저장하지 않고 후보 코드워드의 신드롬 계수를 선택하고 이를 이용해 오류의 위치를 생성함으로써 수신어의 오류를 정정하는 연산을 수행한다. 이러한 방법을 이용함으로써, 후보 코드워드의 오류 위치를 저장하기 위한 D-플립플롭을 제거할 수 있다.
- [0026] 그리고, 신드롬 계수 계산부(140)는 기존의 수정된 SBS 알고리즘을 응용하며, 연산에 필요한 계수들을 공유함으로써 하드웨어 복잡도를 낮추었고, 패턴 신드롬 생성부(130)로부터 입력을 받아 후보 코드워드의 신드롬 계수의 연산을 수행한다.
- [0027] 테스트 신드롬 친 탐색부(180)는 테스트 신드롬 연산부(120)로부터 입력을 받아 LLR의 위치를 복원해 연관정을 위한 오류 위치를 생성한다.
- [0028] 오류 신드롬 계수 결정부 및 친 탐색부(160)는 모든 후보 코드워드의 친 탐색 결과를 D-플립플롭에 저장하지 않아 코드워드의 길이와 독립적인 복잡도를 갖는다. 후보 코드워드의 친 탐색 결과는 오류 신드롬 계수를 선택하고 이를 친 검사하는 것으로 대체 가능하고, 선-친 탐색부로부터 후보 코드워드의 신드롬 계수와 제어부로부터 제어 신호를 입력으로 오류 신드롬 계수가 오류를 정정하기에 적합한지 판정하며, 선택된 하나의 후보 코드워드의 신드롬 계수만 이용해 후보 코드워드의 경관정 오류 위치를 생성한다.
- [0029] 하기의 표 1은 연관정 BCH 부호 복호 장치의 복호 과정의 의사코드를 도시한다.

[0030] <圖 1>

Soft-Decision (1020, 990, 3) BCH Decoding

Input: $r_i = \{r_{i,0}, r_{i,1}, r_{i,2}, r_{i,3}\}$, ($i = n-1, n-2, \dots, 1, 0$)

Start Soft-Decision BCH Decoding:

1) Separate out Hard-Decision and magnitude

Hard-Decision :

$$r_{HD,i} = r_{i,0} \quad (i = n-1, n-2, \dots, 1, 0)$$

Magnitude :

$$|r_i| = m_i = \{m_{i,0}, m_{i,1}, m_{i,2}\} \quad (i = n-1, n-2, \dots, 1, 0)$$

if ($r_{HD,i} == 0$)

$$m_i = \{r_{i,1}, r_{i,2}, r_{i,3}\}$$

else begin

$$\text{if } (\{r_{i,1}, r_{i,2}, r_{i,3}\} = \{1, 1, 1\}) \quad m_i = \{0, 0, 1\}$$

$$\text{else if } (\{r_{i,1}, r_{i,2}, r_{i,3}\} = \{1, 1, 0\}) \quad m_i = \{0, 1, 0\}$$

$$\text{else if } (\{r_{i,1}, r_{i,2}, r_{i,3}\} = \{1, 0, 1\}) \quad m_i = \{0, 1, 1\}$$

$$\text{else if } (\{r_{i,1}, r_{i,2}, r_{i,3}\} = \{1, 0, 0\}) \quad m_i = \{1, 0, 0\}$$

$$\text{else if } (\{r_{i,1}, r_{i,2}, r_{i,3}\} = \{0, 1, 1\}) \quad m_i = \{1, 0, 1\}$$

$$\text{else if } (\{r_{i,1}, r_{i,2}, r_{i,3}\} = \{0, 1, 0\}) \quad m_i = \{1, 1, 0\}$$

$$\text{else} \quad m_i = \{1, 1, 1\}$$

end

2) Syndrome and Test Syndrome Calculator

Syndrome Calculator :

for $i = n-1$ *until* $i = 0$

$$S_1 = \sum r_{HD,i} \cdot \alpha^i, S_2 = \sum r_{HD,i} \cdot \alpha^{2i}, S_3 = \sum r_{HD,i} \cdot \alpha^{3i}$$

[0034]

Test Syndrome Calculator :

Initial: $\min_0 = \{1, 1, 1\}$, $\min_1 = \{1, 1, 1\}$, $\min_2 = \{1, 1, 1\}$

for $i = n-1$ *until* $i = 0$

if ($i > 2(n-1) / 3$) *begin*

if ($m_i < \min_0$) $\min_0 = m_i$, $\alpha^0 = \alpha^i$

end

else if ($i > (n-1) / 3$) *begin*

if ($m_i < \min_1$) $\min_1 = m_i$, $\alpha^{2^1} = \alpha^i$

end

else begin

if ($m_i < \min_2$) $\min_2 = m_i$, $\alpha^{2^2} = \alpha^i$

end

3) Pattern Syndrome Generator

$G = \{G_2, G_1, G_0\}$, $C = \{C_1, C_0\}$

Initial: $S_{i,p,0} = S_b$, $C = \{0, 0\}$

for $j=1$ *until* $j=2^p-1$

begin

$C_0 = \overline{C_1}$

$C_1 = C_0$

$G_0 = \overline{C_0 \oplus C_1}$

$G_1 = C_0 \cdot \overline{C_1}$

$G_2 = \overline{C_0} \cdot C_1$

if ($\{G_2, G_1, G_0\} = \{0, 0, 1\}$) $S_{i,p,j} = S_{i,p(j-1)} + (\alpha^{2^0})^i$

else if ($\{G_2, G_1, G_0\} = \{0, 1, 0\}$) $S_{i,p,j} = S_{i,p(j-1)} + (\alpha^{2^1})^i$

else if ($\{G_2, G_1, G_0\} = \{1, 0, 0\}$) $S_{i,p,j} = S_{i,p(j-1)} + (\alpha^{2^2})^i$

else $S_{i,p,j} = S_{i,p(j-1)}$

end

4) Syndrome Factor Calculator

$deg = \{deg_{1,p,j}, deg_{3,p,j}\}$

for $j=0$ until $j=2^p-1$

begin

if $(S_{1, \eta j}^3 + S_{3, \eta j} == 0)$

begin

$R_{\eta j} = S_{1, \eta j}^3 + S_{3, \eta j}$

$A_{\eta j} = S_{1, \eta j}^2$

$B_{\eta j} = S_{1, \eta j}$

$C_{\eta j} = 0$

$deg_{2, \eta j} = 1$

if $(S_{1, \eta j} (S_{1, \eta j}^2 S_{3, \eta j} + S_{5, \eta j}) == 0)$ $deg_{3, \eta j} = 0$

else $deg_{3, \eta j} = 1$

end

end

else

begin

$R_{\eta j} = (S_{1, \eta j}^3 + S_{3, \eta j})^2 + S_{1, \eta j} (S_{1, \eta j}^2 S_{3, \eta j} + S_{5, \eta j})$

$A_{\eta j} = S_{1, \eta j}^2 S_{3, \eta j} + S_{5, \eta j}$

$B_{\eta j} = S_{1, \eta j}^4 + S_{1, \eta j} S_{3, \eta j}$

$C_{\eta j} = S_{1, \eta j}^3 + S_{3, \eta j}$

$deg_{2, \eta j} = 1$

if $((S_{1, \eta j}^3 + S_{3, \eta j})^2 + S_{1, \eta j} (S_{1, \eta j}^2 S_{3, \eta j} + S_{5, \eta j})) == 0$ $deg_{3, \eta j} = 0$

else $deg_{3, \eta j} = 1$

end

end

[0041]

[0042]

[0043]

5) Pre-Chien Search and Metric Check

Pre-Chien Search :

```

for j=0 until j=2p-1
begin
  for i = n-1 until i = 0
    begin
      if (  $R_{pj,i} = A_{pj,i} \alpha^i + B_{pj,i} \alpha^{2p} + C_{pj,i} \alpha^{3p}$  )  $M_{pj,i} = 1$ 
      else
         $M_{pj,i} = 0$ 
      end
    end
  end
end

```

Metric Check :

```

for j=0 until j=2p-1
begin
   $M_{tpj} = \sum M_{tpj,i}$ 
End

```

6) Test Syndrome Chien Search

```

for i = n-1 until i = 0
begin
  if (  $\alpha^i + \alpha^0 == 0$  )     $H_{\alpha^0} = 1$ 
  else
     $H_{\alpha^0} = 0$ 

  if (  $\alpha^i + \alpha^{t1} == 0$  )     $H_{\alpha^{t1}} = 1$ 
  else
     $H_{\alpha^{t1}} = 0$ 

  if (  $\alpha^i + \alpha^{t2} == 0$  )     $H_{\alpha^{t2}} = 1$ 
  else
     $H_{\alpha^{t2}} = 0$ 
  end
end

```

7) Error Syndrome Factor Decision and re-Chien Search

Error Syndrome Factor Decision :

```

for j=0 until j=2p-1
begin
    if(ctr == j)
        begin
            Rsel = Rtpj,i, Asel = Atpj,i, Bsel = Btpj,i, Csel = Ctpj,i
            esel = etpj
            break
        end
    end
end

```

Chien Search :

```

for i = n-1 until i = 0
begin
    if (Rsel = Asel α2p + Bsel α2p + Csel α2p)    di = 1
    else
        di = 0
    end
end

```

Merge Test Syndrome Chien search :

```

etz0 = 0

etz1 = Hαt0
etz2 = Hαt0 + Hαt1
etz3 = Hαt1
etz4 = Hαt1 + Hαt2
etz5 = Hαt0 + Hαt1 + Hαt2
etz6 = Hαt0 + Hαt2
etz7 = Hαt2

```

8) Corrected Codeword

$$C_i = r_{HD,i} + d_i + e_{sel}$$

Output : C_i

채널로부터 수신된 코드워드의 심볼은 2의 보수 체계의 LLR 값으로 연관정 BCH 복호기의 입력 값이며 크기 값으로 변환되어 테스트 신드롬 연산부로 입력된다. 그리고 최상위 비트는 경관정 부호로 신드롬 연산부에 입력된다.

신드롬 연산부의 신드롬과 테스트 신드롬 연산부 테스트 신드롬은 패턴 신드롬 생성부로 입력되며 패턴 신드롬이 출력된다. 패턴 신드롬 생성부의 출력은 신드롬 계수 계산부로 입력되며 순차적으로 모든 후보 코드워드에 대한 신드롬 계수가 출력된다. 패턴 신드롬을 연산할 경우에, 첫 번째 후보 코드워드에 해당하는 패턴 신드롬은 $S_{i,tp0} = S_i$, 두 번째 후보 코드워드에 해당하는 패턴 신드롬은 $S_{i,tp1} = S_{i,tp0} + (a^{t0})^i$, 두 번째 후보 코드워드에

해당하는 패턴 신드롬은 $S_{i, tp2} = S_{i, tp1} + (a^{t1})^i$, 세 번째 후보 코드워드에 해당하는 패턴 신드롬은 $S_{i, tp3} = S_{i, tp2} + (a^{t0})^i$, 네 번째 후보 코드워드에 해당하는 패턴 신드롬은 $S_{i, tp4} = S_{i, tp3} + (a^{t2})^i$, 다섯 번째 후보 코드워드에 해당하는 패턴 신드롬은 $S_{i, tp5} = S_{i, tp4} + (a^{t0})^i$, 여섯 번째 후보 코드워드에 해당하는 패턴 신드롬은 $S_{i, tp6} = S_{i, tp5} + (a^{t1})^i$, 일곱 번째 후보 코드워드에 해당하는 패턴 신드롬은 $S_{i, tp7} = S_{i, tp6} + (a^{t0})^i$ 로 계산된다.

[0054] 후보 코드워드에 해당하는 신드롬 계수를 계산할 경우, $S_{1, tpj}^3 + S_{3, tpj} = 0$ 가 성립되면 $R_{tpj} = S_{1, tpj}^3 + S_{3, tpj}$, $A_{tpj} = S_{1, tpj}^2$, $B_{tpj} = S_{1, tpj}$, $C_{tpj} = 0$, $deg_{2, tpj} = 1$ 의 값이 출력되며, 이때 $S_{1, tpj} (S_{1, tpj}^2 S_{3, tpj} + S_{5, tpj}) = 0$ 이 성립하면 $deg_{3, tpj} = 0$, 성립하지 않으면 $deg_{3, tpj} = 1$ 이 출력된다. 반면 $S_{1, tpj}^3 + S_{3, tpj} = 0$ 가 성립되지 않으면 $R_{tpj} = (S_{1, tpj}^3 + S_{3, tpj})^2 + S_{1, tpj} (S_{1, tpj}^2 S_{3, tpj} + S_{5, tpj})$, $A_{tpj} = S_{1, tpj}^2 S_{3, tpj} + S_{5, tpj}$, $B_{tpj} = S_{1, tpj}^4 + S_{1, tpj} S_{3, tpj}$, $C_{tpj} = S_{1, tpj}^3 + S_{3, tpj}$, $deg_{2, tpj} = 1$ 의 값이 출력되며, 이때 $(S_{1, tpj}^3 + S_{3, tpj})^2 + S_{1, tpj} (S_{1, tpj}^2 S_{3, tpj} + S_{5, tpj}) = 0$, 이 성립하면 $deg_{3, tpj} = 0$, 성립하지 않으면 $deg_{3, tpj} = 1$ 이 출력된다. 여기서 $j=2^p-1, \dots, 0$ 이다. 신드롬 계수의 차수 정보 $deg_{2, tp0}$, $deg_{3, tp0}$, $deg_{2, tp1}$, $deg_{3, tp1}$, $deg_{2, tp2}$, $deg_{3, tp2}$, $deg_{2, tp3}$, $deg_{3, tp3}$, $deg_{2, tp4}$, $deg_{3, tp4}$, $deg_{2, tp5}$, $deg_{3, tp5}$, $deg_{2, tp6}$, $deg_{3, tp6}$, $deg_{2, tp7}$, $deg_{3, tp7}$ 은 제어부(Controller)로 입력된다.

[0055] 선-친 탐색부 및 메트릭 검사부는 후보 코드워드의 신드롬 계수를 입력으로 받아 각 후보 코드워드의 오류 위치를 출력하는 선-친 탐색부와 오류 위치의 수에 대한 합을 연산하는 메트릭 검사부가 있다.

[0056] 선-친 탐색부 및 메트릭 검사부에서는 모든 후보 코드워드에 대한 신드롬 계수를 순차적으로 입력 받아 조건 $R_{tpj, i} = A_{tpj, i} a^p + B_{tpj, i} a^{2p} + C_{tpj, i} a^{3p}$ 를 검사하며 여기서 $j=N-1, \dots, 0$ 을 나타낸다. $R_{tpj, i} = A_{tpj, i} a^p + B_{tpj, i} a^{2p} + C_{tpj, i} a^{3p}$ 가 성립되면 해당하는 j번째 비트의 위치에 오류가 발생을 의미하며 메트릭 검사부에서는 이 값을 받아 M 값에 1을 더한다. 메트릭 검사부의 출력 M_{tp0} , M_{tp1} , M_{tp2} , M_{tp3} , M_{tp4} , M_{tp5} , M_{tp6} , M_{tp7} 는 제어부로 입력된다.

[0057] 제어부에서는 모든 후보 코드워드의 신드롬 계수의 차수 정보 deg_2 , deg_3 를 입력으로 받으며, 이후에 상기 위와 같이 친 탐색 및 메트릭 검사부의 Metric값 M도 입력으로 받음으로써, 오류 정정이 가능한 후보 코드워드 판정을 수행한다. 먼저 deg_2 가 0인지 판별하고, 0일 때 오류가 없거나 1개($t=1$)이기 때문에 M이 0이거나 1일 값인 후보 코드워드의 신드롬 계수를 선택한다. 반면 먼저 deg_2 가 1이라면 오류의 개수가 2개이거나 3개인 경우가 되며 deg_3 는 오류의 개수가 2일 때 항상 0이 되므로 만약 오류의 개수가 3이라면 1이 된다. 이러한 경우 M의 값과 $\{deg_2, deg_3\}$ 를 비교하고 상기 두 값이 같다면 신드롬 계수의 차수 정보와 오류의 개수가 동일하여 오류를 정정하기에 적합한 후보 코드워드의 신드롬 계수가 된다. 상기 조건에 따라 정정 가능한 후보 코드워드의 신드롬 계수를 판정하고 제어신호를 출력하게 된다.

[0058] 테스트 신드롬 친 탐색부에서는 테스트 신드롬으로부터 $a^i + a^{j0} = 0$ ($i=N-1, \dots, 0$, $j=p-1, \dots, 0$) 이 성립하는지 판단하며, 상기 조건이 만족할 경우 최소 신뢰 비트의 위치 $e_{t0}e_{t0}$, $e_{t1}e_{t1}$, $e_{t2}e_{t2}$, 에 1을 출력하고 만족하지 않을 경우 0을 출력한다.

[0059] 오류 신드롬 계수 결정 및 친 탐색부에서는 제어부에서 받은 제어신호와 선-친 탐색부 및 메트릭 검사부로부터 후보 코드워드의 신드롬 계수, 테스트 신드롬 친 탐색부의 복원된 최소 신뢰 비트의 위치를 입력으로 받는다. 제어신호에 따라 후보 코드워드의 신드롬 중 오류를 정정하기에 적합한 신드롬 계수와 연관된 오류의 위치 e_0 , e_1 , e_2 , e_3 , e_4 , e_5 , e_6 , e_7 을 표 1, 오류 신드롬 계수 결정 부분에서 나타낸 바와 같이 연산을 수행한다.

[0060] 도2는 본 발명의 일 실시예에 따른 최소 신뢰 비트의 위치를 추출하는 테스트 신드롬 연산부를 나타내는 도면이

다.

- [0061] 본 발명의 일 실시예에 따른 테스트 신드롬 연산부(200)는 비교기(210)를 포함한다. 비교기(210)를 통해 체이스 알고리즘에 기초하여 채널로부터 수신된 심볼의 LLR의 크기를 비교하고, 상기 심볼을 최소 신뢰 비트의 개수로 분할한 각 구역마다 한 개의 최소 신뢰 비트를 추출하여 최소 신뢰 비트의 위치를 테스트 신드롬으로 변환한다.
- [0062] 일 실시예에 따르면, 테스트 신드롬 연산부(200)는 표 1에 나타난 바와 같이 최소 신뢰 비트를 3개 ($p=3$)로 설정할 경우에 수신어의 구간을 3으로 나누어 각 구간에 존재하는 최소 신뢰 비트의 위치를 테스트 신드롬 값으로 변환한다. 변환된 테스트 신드롬을 이용하여 코드워드의 생성 없이 패턴 신드롬을 계산하게 된다.
- [0063] 도 3은 본 발명의 실시예에 따른 패턴 신드롬 생성부를 나타내는 도면이다.
- [0064] 본 발명의 실시예에 따른 패턴 신드롬 생성부(300)는 테스트 신드롬 선택기(310)를 포함한다. 테스트 신드롬 연산부로부터 테스트 신드롬 선택기(310)를 통해 α^{t_0} , α^{t_1} , α^{t_2} 을 입력으로 표 1에 나타난 바와 같이 1개의 테스트 신드롬을 선택해 합하는 방법으로 모든 후보 코드워드에 대한 패턴 신드롬 $S_{i,tp0}$, $S_{i,tp1}$, $S_{i,tp2}$, $S_{i,tp3}$, $S_{i,tp4}$, $S_{i,tp5}$, $S_{i,tp6}$, $S_{i,tp7}$ ($i=1,3,5$)을 순차적으로 생성한다.
- [0065] 도 4은 본 발명의 실시예에 따른 신드롬 계수 계산부를 나타내는 도면이다.
- [0066] 위에서 설명된 패턴 신드롬 생성부의 패턴 신드롬 $S_{1,tpj}$, $S_{3,tpj}$, $S_{5,tpj}$, ($j=2^p-1, \dots, 0$)은 신드롬 계수 계산부에 S_1 , S_3 , S_5 로 입력되며, 신드롬 계수 계산부는 공통되는 연산을 공유하여 오류 신드롬 계수의 차수가 2와 3일 때의 결과 표1에 나타난 바와 같이 연산을 최적화 하여 계산을 수행한다
- [0067] 도 5은 본 발명의 실시예에 따른 선-친 탐색부 및 메트릭 검사부를 나타내는 도면이다.
- [0068] 선-친 탐색부 및 메트릭 검사부(500)의 메트릭 검사부(510)는 후보 코드워드의 수에 따라 병렬 구조를 가지며, 하나의 블록 구조 내부에는 해당하는 후보 코드워드의 신드롬 계수가 입력되고, 입력된 신드롬 계수 R , A , B , C 와 메트릭 값 M 이 출력된다.
- [0069] 도 6은 본 발명의 실시예에 따른 테스트 신드롬 친 탐색부를 나타내는 도면이다.
- [0070] 테스트 신드롬 연산부로부터 테스트 신드롬을 입력 받아 최소 신뢰 비트의 위치를 복원하며, 첫 번째 코드워드 구간에서의 최소 신뢰 비트의 위치는 e_{t0} , 두 번째 위치는 e_{t1} , 세 번째 위치는 e_{t2} 에 복원된다.
- [0071] 도 7은 본 발명의 실시예에 따른 오류 신드롬 계수 결정부 및 친 탐색부를 나타내는 도면이다.
- [0072] 오류 신드롬 계수 결정부 및 친 탐색부(700)의 친 탐색부(710)는 제어부로부터 제어신호, 테스트 신드롬 친 탐색부로부터 p 개의 최소 신뢰 비트 위치, 선-친 탐색부 및 메트릭 검사부로부터 후보 코드워드의 신드롬 계수를 입력 받는다. 제어신호에 의해 후보 코드워드의 경관정을 위한 신드롬 계수를 선택하고 내부의 친 탐색부로 전달한다. 상기 제어신호에 따라 표 1에 나타난 바와 같이 연관정을 위한 최소 신뢰 비트의 연산 결과를 선택한다. 친 탐색 결과와 선택된 최소 신뢰 비트의 연산 결과를 갈루아 체(Galois field: GF) 덧셈기로 더하여 출력하고, 상기의 결과를 수신어의 경관정 부호와 GF 덧셈기로 더하며, 오류 벡터(Error vector)을 생성하여 수신어의 복원을 수행한다.
- [0073] 도 8은 본 발명의 일 실시예에 따른 테스트 신드롬을 이용한 연관정 BCH 복호 방법을 설명하기 위한 흐름도이다.
- [0074] 제안하는 테스트 신드롬을 이용한 연관정 BCH 복호 방법은 코드워드를 최소 신뢰 비트의 개수로 분할한 각 구간

에 대하여 최소 신뢰 비트의 위치를 한 개씩 추출하고, 테스트 신드롬으로 변환하는 단계(810), 신드롬 연산부로부터의 신드롬 및 상기 변환된 테스트 신드롬을 입력 받아 패턴 신드롬을 생성하는 단계(820), 상기 생성된 패턴 신드롬을 입력 받아 후보 코드워드의 신드롬 계수를 연산하고, 신드롬 행렬의 각 차수에 대하여 신드롬 계수의 공통 부분을 공유하면서 모든 차수에 대한 신드롬 계수를 별도로 연산하는 단계(830) 상기 패턴 신드롬을 입력 받아 오류의 수를 합 연산하는 단계(840) 및 제어신호를 통해 상기 후보 코드워드의 신드롬 계수를 결정하고, 최소 신뢰 비트의 연산 결과를 선택하며, 오류를 정정하기 위해 벡터를 연산하는 단계(850)를 포함한다.

[0075] 단계(810)에서, 코드워드를 최소 신뢰 비트의 개수로 분할한 각 구간에 대하여 최소 신뢰 비트의 위치를 한 개씩 추출하고, 테스트 신드롬으로 변환한다.

[0076] 단계(820)에서, 신드롬 연산부로부터의 신드롬 및 상기 변환된 테스트 신드롬을 입력 받아 패턴 신드롬을 생성한다. 단계(820)에서 체이스 알고리즘에 기초하여 채널로부터 수신된 심볼의 LLR의 크기를 비교하고, 상기 심볼을 최소 신뢰 비트의 개수로 분할한 각 구역마다 한 개의 최소 신뢰 비트를 추출하여 최소 신뢰 비트의 위치를 테스트 신드롬으로 변환한다. 이때, 별도의 정렬 연산 과정을 필요로 하지 않고 상기 테스트 신드롬 값을 쉬프트 레지스터(Shift register)로 입력함으로써 테스트 신드롬 연산을 수행한다.

[0077] 단계(830)에서 상기 생성된 패턴 신드롬을 입력 받아 후보 코드워드의 신드롬 계수를 연산하고, 신드롬 행렬의 각 차수에 대하여 신드롬 계수의 공통 부분을 공유하면서 모든 차수에 대한 신드롬 계수를 별도로 연산한다. 신드롬 연산부로부터의 신드롬 및 상기 변환된 테스트 신드롬을 입력 받아 모든 후보 코드워드에 대한 신드롬을 D-플립플롭과 피드백 회로를 이용하여 순차적으로 계산한다.

[0078] 이후 단계(840)에서, 상기 패턴 신드롬을 입력 받아 오류의 수를 합 연산한다.

[0079] 단계(850)에서, 제어신호를 통해 상기 후보 코드워드의 신드롬 계수를 결정하고, 최소 신뢰 비트의 연산 결과를 선택하며, 오류를 정정하기 위해 벡터를 연산한다. 제어신호를 통해 후보 코드워드의 신드롬 계수를 결정하여 친 탐색부로 상기 결정된 신드롬 계수를 전달하고, 연관정을 위한 최소 신뢰 비트의 연산 결과를 선택하여 친 탐색부의 결과와 선택된 최소 신뢰 비트의 연산 결과를 GF 덧셈기를 이용하여 합 연산을 수행한 후, 수신어의 오류를 정정하기 위한 오류 벡터를 생성한다. 이때, 오류의 발생 여부에 대한 출력없이 신드롬 계수로부터 발생하는 오류의 수를 합 연산하고, 후보 코드워드의 신드롬 계수를 입력 받아 상기 신드롬 계수를 출력한다.

[0080] 본 발명에서 제안하는 연관정 BCH 복호 방법은 후보 코드워드에 대한 연산을 신드롬을 기반으로 연산하며 모든 후보코드의 신드롬 다시 말해, 패턴 신드롬을 테스트 신드롬과 경관정 신드롬을 이용하여 생성한다. 그리고 제안하는 연관정 BCH 복호 장치는 후보 코드워드의 오류 위치를 저장하지 않고 후보 코드워드의 신드롬 계수를 선택하고 이를 이용해 오류의 위치를 생성함으로써 수신어의 오류를 정정하는 연산을 수행한다. 이러한 방법을 이용함으로써, 후보 코드워드의 오류 위치를 저장하기 위한 D-플립플롭을 제거할 수 있다.

[0081] 도 9는 본 발명의 일 실시예에 따른 비트 오류율을 나타내는 도면이다.

[0082] 도 9를 참조하면, (1020, 990, 3) 부호를 기준으로 경관정 BCH 부호의 비트오류율 성능, 연관정 BCH 부호의 비트오류율 성능을 비교할 수 있다. 이때, 오류 성능을 비트오류율 10⁻⁶을 기준하였을 경우 연관정 BCH 부호가 경관정 BCH 부호보다 약 0.65dB의 코딩 이득(coding gain)이 발생하며 비트오류율 성능이 향상된 것을 확인할 수 있다.

[0083] 따라서, 본 발명에서 제안하는 방법을 사용함으로써 기존에 경관정 BCH 복호 장치보다 비트오류율 성능이 더 좋은 연관정 BCH 복호 장치를 구현할 수 있다.

[0084] 이상에서 설명된 장치는 하드웨어 구성요소, 소프트웨어 구성요소, 및/또는 하드웨어 구성요소 및 소프트웨어 구성요소의 조합으로 구현될 수 있다. 예를 들어, 실시예들에서 설명된 장치 및 구성요소는, 예를 들어, 프로세서, 콘트롤러, ALU(arithmetic logic unit), 디지털 신호 프로세서(digital signal processor), 마이크로컴퓨터, FPGA(field programmable gate array), PLU(programmable logic unit), 마이크로프로세서, 또는 명령(instruction)을 실행하고 응답할 수 있는 다른 어떠한 장치와 같이, 하나 이상의 범용 컴퓨터 또는 특수 목적 컴퓨터를 이용하여 구현될 수 있다. 처리 장치는 운영 체제(OS) 및 상기 운영 체제 상에서 수행되는 하나 이상의 소프트웨어 애플리케이션을 수행할 수 있다. 또한, 처리 장치는 소프트웨어의 실행에 응답하여, 데이터를 접근, 저장, 조작, 처리 및 생성할 수도 있다. 이해의 편의를 위하여, 처리 장치는 하나가 사용되는 것으로

설명된 경우도 있지만, 해당 기술분야에서 통상의 지식을 가진 자는, 처리 장치가 복수 개의 처리 요소 (processing element) 및/또는 복수 유형의 처리 요소를 포함할 수 있음을 알 수 있다. 예를 들어, 처리 장치는 복수 개의 프로세서 또는 하나의 프로세서 및 하나의 컨트롤러를 포함할 수 있다. 또한, 병렬 프로세서 (parallel processor)와 같은, 다른 처리 구성 (processing configuration)도 가능하다.

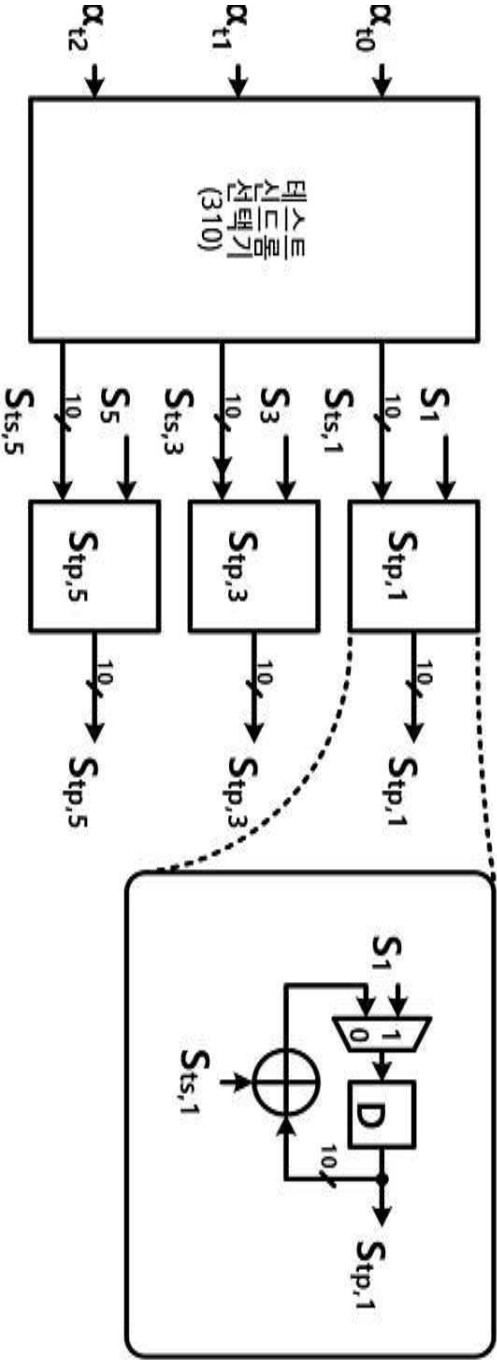
[0085] 소프트웨어는 컴퓨터 프로그램 (computer program), 코드 (code), 명령 (instruction), 또는 이들 중 하나 이상의 조합을 포함할 수 있으며, 원하는 대로 동작하도록 처리 장치를 구성하거나 독립적으로 또는 결합적으로 (collectively) 처리 장치를 명령할 수 있다. 소프트웨어 및/또는 데이터는, 처리 장치에 의하여 해석되거나 처리 장치에 명령 또는 데이터를 제공하기 위하여, 어떤 유형의 기계, 구성요소 (component), 물리적 장치, 가상 장치 (virtual equipment), 컴퓨터 저장 매체 또는 장치에 구체화 (embody)될 수 있다. 소프트웨어는 네트워크로 연결된 컴퓨터 시스템 상에 분산되어서, 분산된 방법으로 저장되거나 실행될 수도 있다. 소프트웨어 및 데이터는 하나 이상의 컴퓨터 판독 가능 기록 매체에 저장될 수 있다.

[0086] 실시예에 따른 방법은 다양한 컴퓨터 수단을 통하여 수행될 수 있는 프로그램 명령 형태로 구현되어 컴퓨터 판독 가능 매체에 기록될 수 있다. 상기 컴퓨터 판독 가능 매체는 프로그램 명령, 데이터 파일, 데이터 구조 등을 단독으로 또는 조합하여 포함할 수 있다. 상기 매체에 기록되는 프로그램 명령은 실시예를 위하여 특별히 설계되고 구성된 것들이거나 컴퓨터 소프트웨어 당업자에게 공지되어 사용 가능한 것일 수도 있다. 컴퓨터 판독 가능 기록 매체의 예에는 하드 디스크, 플로피 디스크 및 자기 테이프와 같은 자기 매체 (magnetic media), CD-ROM, DVD와 같은 광기록 매체 (optical media), 플롭티컬 디스크 (floptical disk)와 같은 자기-광 매체 (magneto-optical media), 및 롬 (ROM), 램 (RAM), 플래시 메모리 등과 같은 프로그램 명령을 저장하고 수행하도록 특별히 구성된 하드웨어 장치가 포함된다. 프로그램 명령의 예에는 컴파일러에 의해 만들어지는 것과 같은 기계어 코드뿐만 아니라 인터프리터 등을 사용해서 컴퓨터에 의해서 실행될 수 있는 고급 언어 코드를 포함한다.

[0087] 이상과 같이 실시예들이 비록 한정된 실시예와 도면에 의해 설명되었으나, 해당 기술분야에서 통상의 지식을 가진 자라면 상기의 기재로부터 다양한 수정 및 변형이 가능하다. 예를 들어, 설명된 기술들이 설명된 방법과 다른 순서로 수행되거나, 및/또는 설명된 시스템, 구조, 장치, 회로 등의 구성요소들이 설명된 방법과 다른 형태로 결합 또는 조합되거나, 다른 구성요소 또는 균등물에 의하여 대치되거나 치환되더라도 적절한 결과가 달성될 수 있다.

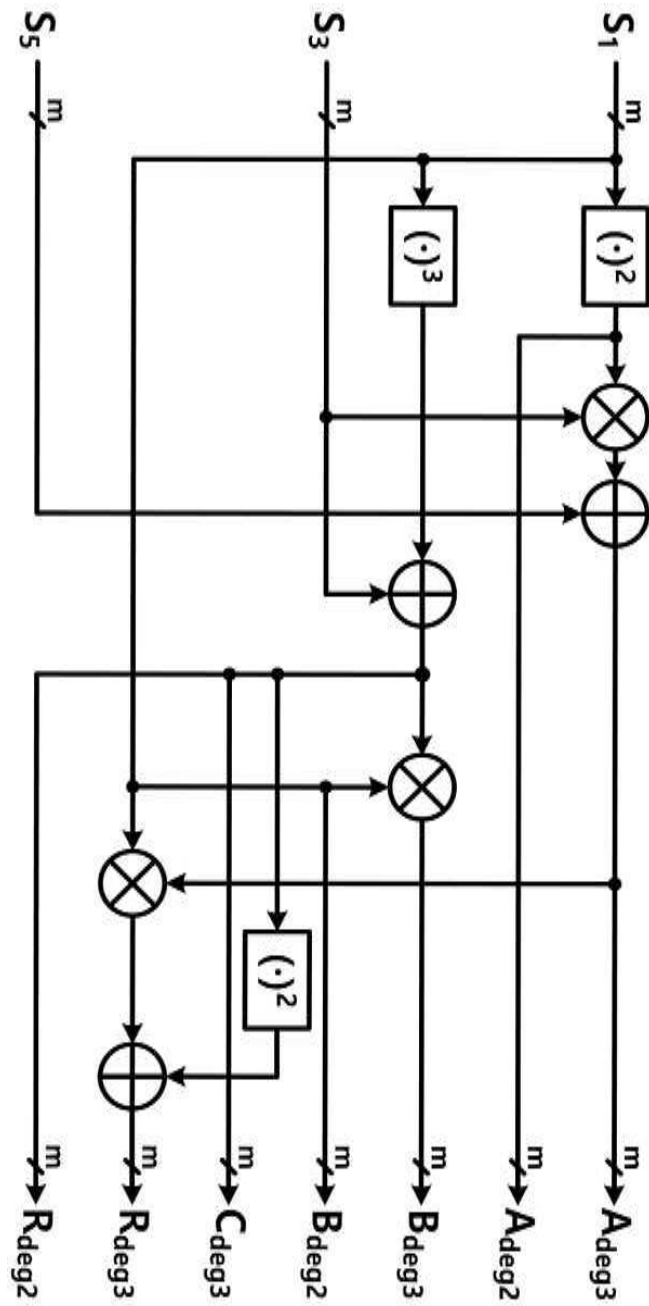
[0088] 그러므로, 다른 구현들, 다른 실시예들 및 특허청구범위와 균등한 것들도 후술하는 특허청구범위의 범위에 속한다.

도면3

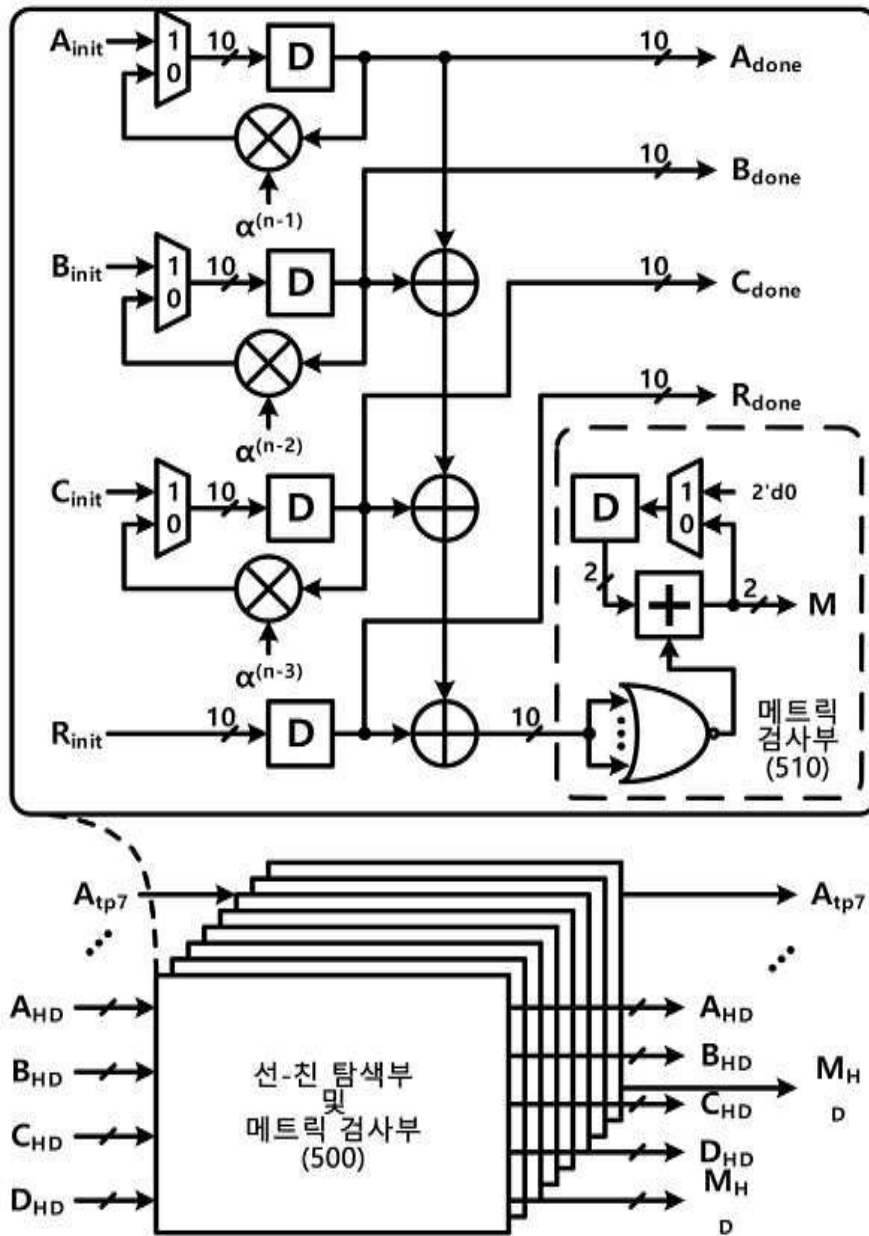


300

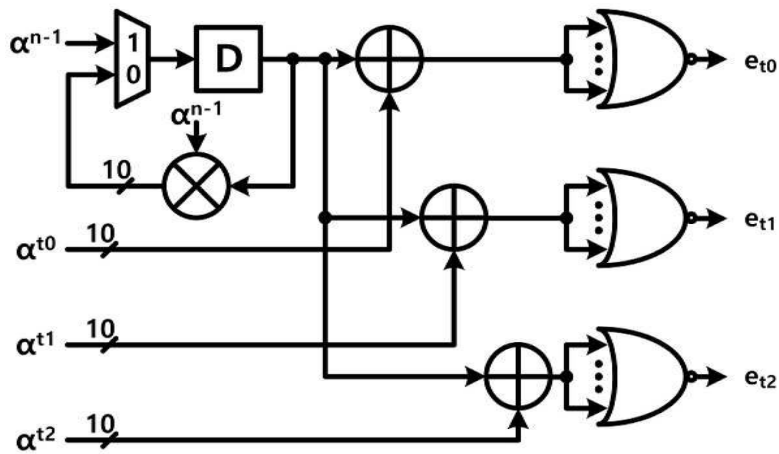
도면4



도면5

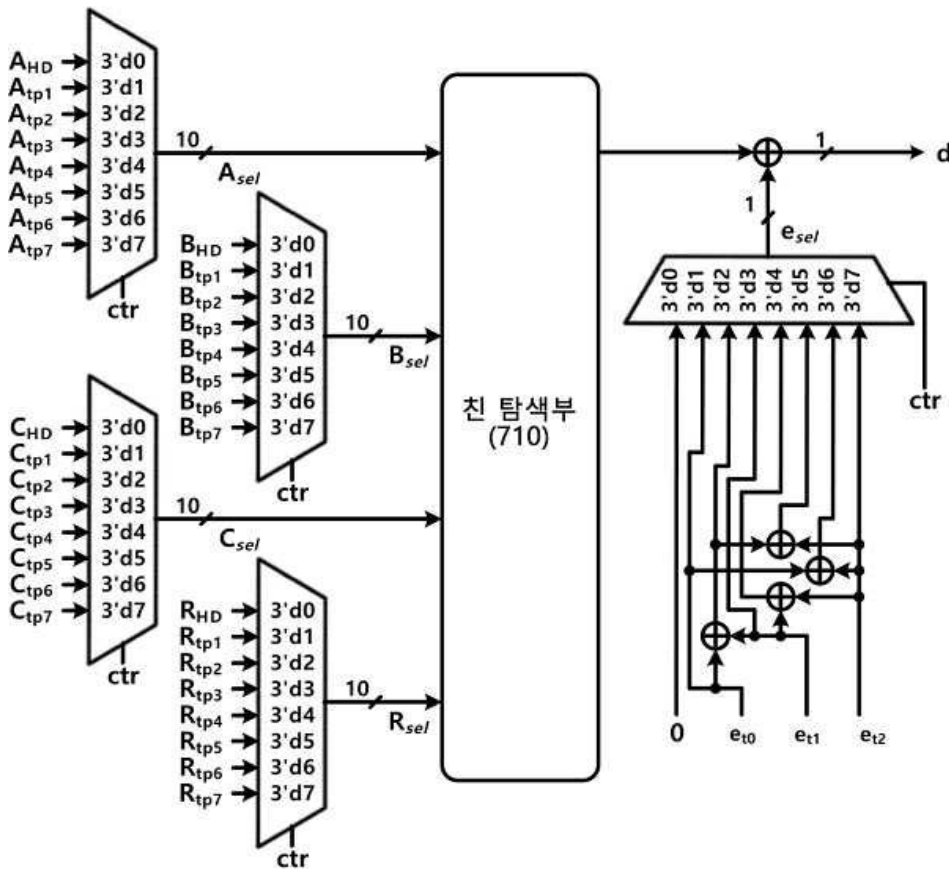


도면6

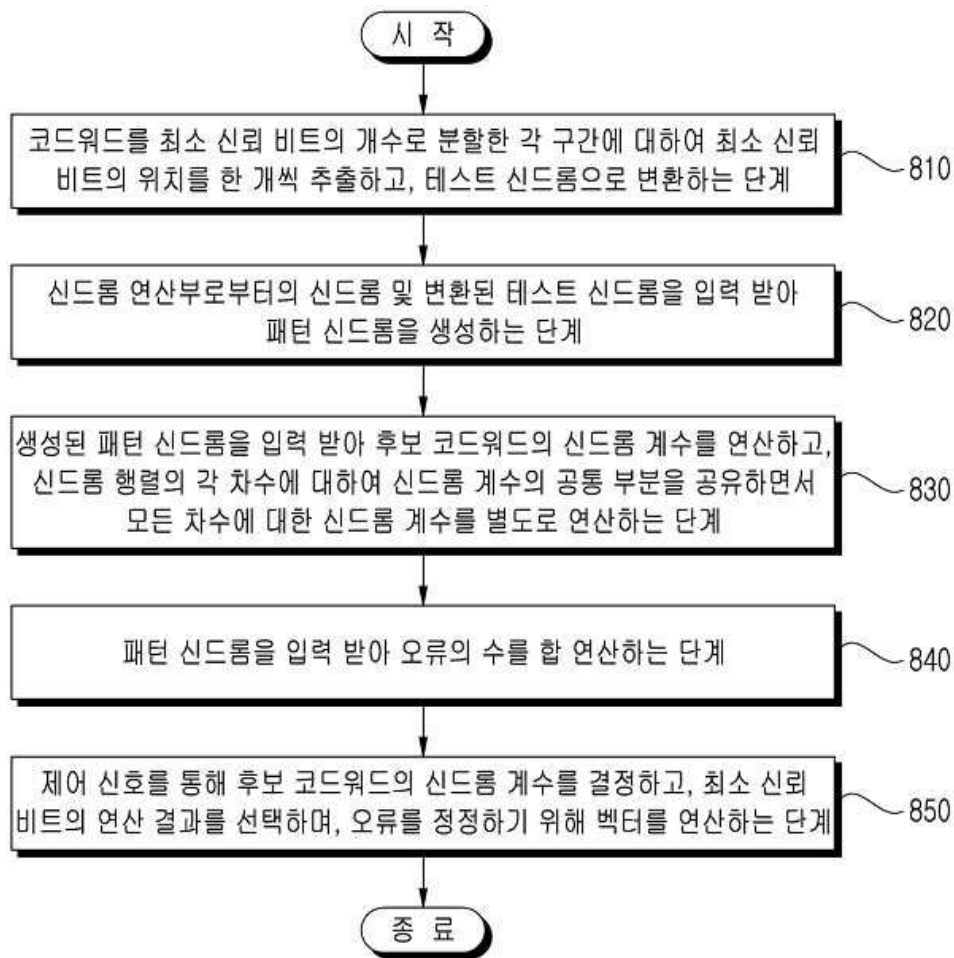


도면7

700



도면8



도면9

