



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2020-0084687
(43) 공개일자 2020년07월13일

(51) 국제특허분류(Int. Cl.)
H03M 13/39 (2006.01) H03M 13/13 (2006.01)
(52) CPC특허분류
H03M 13/3927 (2013.01)
H03M 13/13 (2013.01)
(21) 출원번호 10-2019-0000853
(22) 출원일자 2019년01월03일
심사청구일자 없음

(71) 출원인
삼성전자주식회사
경기도 수원시 영통구 삼성로 129 (매탄동)
(72) 발명자
손재용
경기도 수원시 영통구 덕영대로 1487, 308호 (망포동)
설창규
경기도 오산시 수목원로 430, 1306동 901호 (수청동, 물향기마을휴먼시아꿈에그린13단지아파트)
(74) 대리인
리엔목특허법인

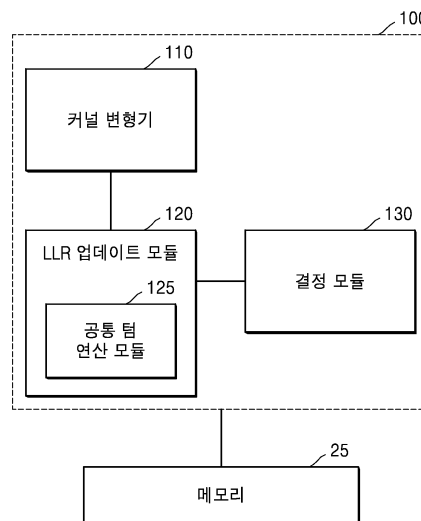
전체 청구항 수 : 총 10 항

(54) 발명의 명칭 비이진 폴라 코드를 수신하는 장치 및 이의 디코딩 방법

(57) 요약

본 개시의 기술적 사상에 따른 채널을 통해 비이진 폴라 코드(non binary polar code)를 수신하는 장치는, 상기 채널 특성이 반영되어 생성된 상기 비이진 폴라 코드에 포함된 입력 심볼들의 LLR(Log Likelihood Ratio)들 및 로어 트라이앵글러 커널(lower triangular kernel)을 이용하여 제1 대상 출력 심볼에 대응하는 제1 입력 심볼을 제외한 나머지 상기 입력 심볼들에 대한 제1 공통 텀(common term)들을 우선적으로 연산하고, 상기 제1 공통 텀들을 이용하여 상기 제1 대상 출력 심볼의 LLR들을 연산하며, 상기 제1 대상 출력 심볼의 LLR들을 기반으로 상기 제1 대상 출력 심볼의 값을 결정하도록 구성된 저복잡도 디코더 및 상기 저복잡도 디코더에 의해 접근 가능하고, 상기 제1 공통 텀들이 저장되도록 구성된 메모리를 포함한다.

대표도 - 도2



(52) CPC특허분류

H03M 13/6502 (2013.01)

(72) 발명자

손홍락

경기도 안양시 동안구 시민대로159번길 62, 204동
501호 (비산동, 은하수벽산아파트)

윤필상

경기도 화성시 동탄반석로 172, 102동 2304호 (반
송동, 동탄 파라곤)

명세서

청구범위

청구항 1

채널을 통해 비이진 폴라 코드(non binary polar code)를 수신하는 장치에 있어서,

상기 채널 특성이 반영되어 생성된 상기 비이진 폴라 코드에 포함된 입력 심볼들의 LLR(Log Likelihood Ratio)들 및 로어 트라이앵글러 커널(lower triangular kernel)을 이용하여 제1 대상 출력 심볼에 대응하는 제1 입력 심볼을 제외한 나머지 상기 입력 심볼들에 대한 제1 공통 텀(common term)들을 우선적으로 연산하고, 상기 제1 공통 텀들을 이용하여 상기 제1 대상 출력 심볼의 LLR들을 연산하며, 상기 제1 대상 출력 심볼의 LLR들을 기반으로 상기 제1 대상 출력 심볼의 값을 결정하도록 구성된 저복잡도 디코더; 및

상기 저복잡도 디코더에 의해 접근 가능하고, 상기 제1 공통 텀들이 저장되도록 구성된 메모리를 포함하는 장치.

청구항 2

제1항에 있어서,

상기 제1 공통 텀들은,

상기 제1 대상 출력 심볼이 어느 하나의 값을 가지는 것을 가정한 때에, 상기 제1 입력 심볼을 제외한 상기 나머지 입력 심볼들의 조합들을 기반으로 상기 입력 심볼들의 LLR들을 이용하여 합산된 값들을 포함하는 것을 특징으로 하는 장치.

청구항 3

채널을 통해 비이진 폴라 코드를 수신하는 장치의 디코딩 방법에 있어서,

상기 채널 특성을 기반으로 상기 비이진 폴라 코드에 포함된 입력 심볼들의 LLR들을 생성하는 단계;

로어 트라이앵글러 커널을 이용하여 제1 대상 출력 심볼에 대응하는 제1 입력 심볼을 제외한 나머지 상기 입력 심볼들에 대한 제1 공통 텀들을 우선적으로 연산하는 단계;

상기 제1 공통 텀들을 이용하여 상기 제1 대상 출력 심볼의 LLR들을 연산하는 단계; 및

상기 제1 대상 출력 심볼의 LLR들을 기반으로 상기 제1 대상 출력 심볼의 값을 결정하는 단계를 포함하는 디코딩 방법.

청구항 4

제3항에 있어서,

상기 장치는, 메모리를 더 포함하고,

상기 디코딩 방법은,

생성된 상기 제1 공통 텀들을 상기 메모리에 저장하는 단계를 상기 메모리에 저장하는 단계를 더 포함하는 것을 특징으로 하는 디코딩 방법.

청구항 5

제3항에 있어서,

상기 디코딩 방법은,

상기 입력 심볼들의 LLR들 및 상기 로어 트라이앵글러 커널을 이용하여 제2 대상 출력 심볼에 대응하는 제2 입력 심볼 및 값이 결정된 상기 제1 입력 심볼을 제외한 나머지 상기 입력 심볼들에 대한 제2 공통 텀들을 우선적으로 연산하는 단계;

상기 제2 공통 텀들을 이용하여 상기 제2 대상 출력 심볼의 LLR을 연산하는 단계; 및

상기 제2 대상 출력 심볼의 LLR을 기반으로 상기 제2 대상 출력 심볼의 값을 결정하는 단계를 더 포함하는 것을 특징으로 하는 디코딩 방법.

청구항 6

제3항에 있어서,

상기 디코딩 방법은,

상기 폴라 코드를 생성하기 위해 이용된 최초 커널에 대한 가우시안 엘리미네이션을 수행함으로써 상기 로어 트라이앵글러 커널을 생성하는 단계를 더 포함하는 것을 특징으로 하는 디코딩 방법.

청구항 7

제6항에 있어서,

상기 로어 트라이앵글러 커널을 생성하는 단계는,

상기 로어 트라이앵글러 커널의 대각선 아래의 텀들을 원시 요소들로 구성하는 단계; 및

상기 로어 트라이앵글러 커널의 지수 값이 상기 최초 커널의 지수 값과 동일하게 변형하는 단계를 더 포함하는 것을 특징으로 하는 디코딩 방법.

청구항 8

제3항에 있어서,

상기 제1 대상 출력 심볼의 LLR들을 연산하는 단계는,

상기 입력 심볼들의 LLR들로부터 상기 제1 대상 출력 심볼이 가질 수 있는 값들에 따른 상기 제1 입력 심볼의 LLR들을 획득하는 단계; 및

상기 제1 입력 심볼의 LLR들과 상기 제1 공통 텀들을 각각 대응하는 성분끼리 합산함으로써 상기 제1 대상 출력 심볼의 LLR들을 생성하는 단계를 더 포함하는 것을 특징으로 하는 디코딩 방법.

청구항 9

제3항에 있어서,

상기 제1 대상 출력 심볼의 값을 결정하는 단계는,

상기 제1 대상 출력 심볼의 LLR들 중 가장 큰 값을 갖는 LLR에 대응하는 값을 상기 제1 대상 출력 심볼의 값으로 결정하는 단계를 더 포함하는 것을 특징으로 하는 디코딩 방법.

청구항 10

제3항에 있어서,

상기 제1 공통 텀들을 우선적으로 연산하는 단계는,

상기 제1 대상 출력 심볼이 어느 하나의 값을 가지는 것을 가정한 때에, 상기 제1 입력 심볼의 값들에 따른 상기 나머지 입력 심볼들의 조합들을 생성하는 단계; 및

상기 입력 심볼들의 LLR들을 이용하여 상기 나머지 입력 심볼들의 조합들에 대한 LLR들을 연산함으로써 상기 제1 공통 텀들을 생성하는 단계를 더 포함하는 것을 특징으로 하는 디코딩 방법.

발명의 설명

기술 분야

본 개시의 기술적 사상은 비이진 폴라 코드를 수신하여 이를 처리하는 장치 및 이의 디코딩 방법에 관한 것이다.

[0001]

배경 기술

[0002] 폴라 코드(polar code)는 ECC(error correction code) 자체로써 뿐만 아니라 다양한 메모리 신호 처리 기술에 적용할 수 있을만큼 활용도가 크다. 특히, 대수부호 기반 비이진 폴라 코드를 이용한 연속 제거(successive cancellation) 디코딩 기법은 이진 폴라 코드를 이용한 연속 제거 리스트(list) 디코딩 기법에 필적하는 정정 능력을 가지면서 동시에 향상된 디코딩 쓰루풋(throughput)을 가질 수 있다. 그러나, 비이진 폴라 코드의 기수가 커질수록 디코딩에서의 연산 복잡도가 기하급수적으로 증가하는 문제가 발생할 수 있다. 따라서, 비이진 폴라 코드를 이용한 효율적인 디코딩 동작을 위해서는 우선적으로 연산 복잡도를 낮추는게 고려되어야 한다.

발명의 내용

해결하려는 과제

[0003] 본 개시의 기술적 사상이 해결하려는 과제는 비이진 폴라 코드를 이용한 디코딩 동작을 수행할 때에, 디코딩 동작에 포함된 다양한 연산의 복잡도를 효과적으로 낮춤으로써, 디코딩 성능의 열화 없이 효율적으로 디코딩을 수행할 수 있는 장치 및 이의 디코딩 방법을 제공하는 데에 있다.

과제의 해결 수단

[0004] 상기와 같은 목적을 달성하기 위하여, 본 개시의 일 측면에 따른 채널을 통해 비이진 폴라 코드(non binary polar code)를 수신하는 장치는, 상기 채널 특성이 반영되어 생성된 상기 비이진 폴라 코드에 포함된 입력 심볼들의 LLR(Log Likelihood Ratio)들 및 로어 트라이앵글러 커널(lower triangular kernel)을 이용하여 제1 대상 출력 심볼에 대응하는 제1 입력 심볼을 제외한 나머지 상기 입력 심볼들에 대한 제1 공통 텀(common term)들을 우선적으로 연산하고, 상기 제1 공통 텀들을 이용하여 상기 제1 대상 출력 심볼의 LLR들을 연산하며, 상기 제1 대상 출력 심볼의 LLR들을 기반으로 상기 제1 대상 출력 심볼의 값을 결정하도록 구성된 저복잡도 디코더 및 상기 저복잡도 디코더에 의해 접근 가능하고, 상기 제1 공통 텀들이 저장되도록 구성된 메모리를 포함한다.

[0005] 본 개시의 다른 측면에 따른 채널을 통해 비이진 폴라 코드를 수신하는 장치의 디코딩 방법은, 상기 채널 특성을 기반으로 상기 비이진 폴라 코드에 포함된 입력 심볼들의 LLR들을 생성하는 단계, 로어 트라이앵글러 커널을 이용하여 제1 대상 출력 심볼에 대응하는 제1 입력 심볼을 제외한 나머지 상기 입력 심볼들에 대한 제1 공통 텀들을 우선적으로 연산하는 단계, 상기 제1 공통 텀들을 이용하여 상기 제1 대상 출력 심볼의 LLR들을 연산하는 단계 및 상기 제1 대상 출력 심볼의 LLR들을 기반으로 상기 제1 대상 출력 심볼의 값을 결정하는 단계를 포함한다.

[0006] 본 개시의 또 다른 측면에 따른 채널을 통해 비이진 폴라 코드를 수신하는 장치의 디코딩 방법은, 상기 채널 특성을 기반으로 상기 비이진 폴라 코드에 포함된 입력 심볼들의 LLR들을 생성하는 단계, 로어 트라이앵글러 커널을 이용하여 제1 대상 출력 심볼에 대응하는 제1 입력 심볼 및 상기 대상 출력 심볼 다음에 디코딩 대상이 되는 제2 대상 출력 심볼에 대응하는 제2 입력 심볼을 제외한 나머지 상기 입력 심볼들에 대한 공통 텀들을 우선적으로 연산하는 단계, 상기 공통 텀들을 이용하여 상기 대상 출력 심볼의 LLR들을 연산하는 단계 및 상기 대상 출력 심볼의 LLR들을 기반으로 상기 대상 출력 심볼의 값을 결정하는 단계를 포함한다. 해당 디코딩 방법은 마지막 출력 심볼의 LLR들을 연산하는 단계까지 연속해서 계산된다.

발명의 효과

[0007] 본 개시의 실시 예들에 따른 저복잡도 디코더는 디코딩을 수행할 때에, 대상 출력 심볼에 따른 공통 텀들을 미리 연산하고, 대상 출력 심볼의 LLR들을 생성하기 위한 연산 동작시에 공통 텀들을 필요할 때마다 손쉽게 이용함으로써 모든 입력 심볼들의 조합들을 반복하여 연산하는 과정을 생략할 수 있고, 이에 따라, 디코딩의 연산 복잡도를 낮출 수 있는 효과가 있다. 결과적으로, 저복잡도 디코더를 구비하는 수신 장치는 기존 수신 장치와 비교하여 디코딩 성능의 열화 없이 효율적으로 디코딩을 수행할 수 있다.

도면의 간단한 설명

[0008] 도 1은 본 개시의 일 실시 예에 따른 데이터 송수신 시스템의 개략적인 블록도이다.

도 2는 본 개시의 일 실시 예에 따른 저복잡도 디코더를 나타내는 블록도이다.

도 3은 도 2의 커널 변형기의 동작을 설명하기 위한 순서도이다.

도 4는 도 2의 단계 S110을 구체적으로 설명하기 위한 순서도이다. 이하에서는, 이해를 돕기 위해 도 2를 참조하여 서술한다.

도 5a는 최초 커널에 대한 설명을 위한 도면이고, 도 5b는 로어 트라이앵글러 커널에 대한 설명을 위한 도면이다.

도 6은 도 1의 저복잡도 디코더의 디코딩 방법을 설명하기 위한 도면이다.

도 7a는 도 2의 저복잡도 디코더의 디코딩 방법을 구체적으로 설명하기 위한 도면이고, 도 7b는 도 7a의 디코딩 방법으로 생성된 공통 텀들을 구체적으로 설명하기 위한 테이블이다.

도 8a는 도 2의 저복잡도 디코딩의 디코딩 방법의 일 실시 예를 설명하기 위한 도면이고, 도 8b는 도 8a의 디코딩 방법으로 생성된 공통 텀들을 설명하기 위한 테이블이다.

도 9는 본 개시의 일 실시 예에 따른 저복잡도 디코더를 나타내는 블록도이다.

도 10은 본 개시의 일 실시예에 따른 전자 장치를 나타내는 블록도이다.

도 11은 본 개시의 일 실시예에 따른 컴퓨팅 시스템을 나타내는 블록도이다.

발명을 실시하기 위한 구체적인 내용

[0009] 이하에서 설명되는 본 개시의 다양한 실시 예들에서는 하드웨어적인 접근 방법을 예시로서 설명한다. 하지만, 본 개시의 다양한 실시 예들에서는 하드웨어와 소프트웨어를 모두 사용하는 기술을 포함하고 있으므로, 본 개시의 다양한 실시 예들이 소프트웨어 기반의 접근 방법을 제외하는 것은 아니다.

[0010] 또한, 본 개시는 정보 비트들을 포함하는 폴라 코드를 디코딩하기 위한 장치 및 방법에 대하여 설명할 것이다. 다만, 이하 사용되는 용어들은 설명의 편의를 위해 예시된 것으로, 본 개시가 후술되는 용어들에 한정되는 것은 아니며, 동등한 기술적 의미를 가지는 다른 용어가 사용될 수 있다.

[0011] 이하, 첨부한 도면을 참조하여 본 발명의 실시 예에 대해 상세히 설명한다.

[0012] 도 1는 본 개시의 일 실시 예에 따른 데이터 송수신 시스템(1)의 개략적인 블록도이다.

[0013] 도 1을 참조하면, 데이터 송수신 시스템(1)은 송신 장치(10) 및 수신 장치(20)를 포함할 수 있다. 본 개시의 사상에 따른 데이터 송수신 시스템(1)은 소정의 데이터를 채널(CH)을 통해 각각 송수신하는 둘 이상의 장치들을 포함하는 다양한 시스템들에 적용될 수 있다. 예를 들어, 데이터 송수신 시스템(1)은 무선 통신 시스템인 때에, 송신 장치(10) 및 수신 장치(20)는 각각 단말 또는 기지국에 해당될 수 있으며, 채널(CH)은 무선 통신에 필요하고, 충분한 대역을 갖는 데이터 전송 영역을 의미할 수 있다. 또한, 데이터 송수신 시스템(1)은 컴퓨팅 시스템인 때에, 송신 장치(10) 및 수신 장치(20)는 각각 호스트, 메모리 컨트롤러(또는, 메모리 컨트롤러를 포함하는 메모리 시스템)에 해당될 수 있으며, 채널(CH)은 송신 장치(10) 및 수신 장치(20) 사이에 연결된 데이터 전송 라인(일 예로, 데이터 버스)을 의미할 수 있다.

[0014] 송신 장치(10)는 채널(CH)을 통하여 송신할 수 있는 데이터의 에러 레이트(error rate)에 관한 성능을 향상시키기 위해 다양한 인코딩 기법을 이용할 수 있다. 일 실시 예로, 송신 장치(10)는 비이진 폴라 코드를 기반으로 한 인코딩을 수행하여, 소정의 데이터를 비이진 폴라 코드로 변환한 후에 채널(CH)을 통해 수신 장치(20)로 송신할 수 있다. 본 개시의 일 실시 예에 따른 수신 장치(20)는 저복잡도 디코더(100)를 포함할 수 있으며, 저복잡도 디코더(100)는 채널(CH)을 통해 수신한 비이진 폴라 코드에 대하여 연산 복잡도를 최소화하여 효율적인 디코딩을 수행할 수 있다. 구체적으로, 저복잡도 디코더(100)는 연속 제거 디코딩 알고리즘을 이용하여 비이진 폴라 코드에 대한 디코딩을 수행할 수 있다. 연속 제거 디코딩 알고리즘에 따르면, 저복잡도 디코더(100)는 폴라 코드를 구성하는 입력 심볼(input symbol)들 및 소정의 커널을 이용하여 출력 심볼들에 대한 순차적인 디코딩을 수행함으로써, 디코딩 결과로 값이 정해진 출력 심볼들을 출력할 수 있다. 출력 심볼들은 송신 장치(10)에서 송신하고자 하는 정보 비트들에 대응될 수 있으며, 수신 장치(20)는 디코딩 결과로 출력된 출력 심볼들을 이용하여 디코딩 다음의 데이터 처리 동작(예를 들면, 에러 정정, 메모리, 신호 처리 등)을 수행할 수 있다.

[0015] 일 실시 예에 따른 저복잡도 디코더(100)는 폴라 코드를 디코딩하기 위한 최초 커널(kernel)을 변형하고, 변형된 커널을 이용하여 폴라 코드를 디코딩할 수 있다. 변형된 커널은 로어 트라이앵글러 커널(lower triangular kernel)로 지칭될 수 있다. 최초 커널(kernel)은 송신 장치(10)가 소정의 정보 비트들을 폴라 코드로 인코딩할

때에, 이용하는 폴라 코드 생성 행렬을 형성하는 기본 단위이며, 이러한 최초 커널에 대한 정보는 폴라 코드를 수신하기 전에 수신 장치(20)는 미리 인지할 수 있다. 출력 심볼들은 $1 \times N$ (N 은 1 이상의 정수) 크기의 비이진 u 벡터에 포함될 수 있으며, 입력 심볼들은 $1 \times N$ 크기의 비이진 x 벡터에 포함될 수 있다. u 벡터 및 x 벡터는 각각 N 개의 구성요소들을 포함할 수 있다. u 벡터와 x 벡터 사이에는 하기의 수학적 1의 관계를 가질 수 있으며, 이러한 관계를 이용하여 저복잡도 디코더(100)는 디코딩을 수행할 수 있다.

[0016] [수학적 1]

[0017]
$$uG_N = x$$

[0018] 상기 수학적 1에서 G_N 은 $N \times N$ 크기의 폴라 코드 생성 행렬을 의미하고, G_N 은 로어 트라이앵글러 커널의 크로네커 곱(Kronecker product)에 의해 형성될 수 있으며, 이에 따라, u 벡터에 포함된 출력 심볼들과 x 벡터에 포함된 입력 심볼들은 G_N 을 기반으로 각각 대응될 수 있다. 로어 트라이앵글러 커널에 관련된 내용은 도 2, 도 5a 및 도 5b에서 구체적으로 서술한다.

[0019] 저복잡도 디코더(100)는 채널(CH) 특성을 기반으로 비이진 폴라 코드에 포함된 입력 심볼들 각각에 대한 LLR(Log Likelihood Ratio)들을 생성할 수 있다. 저복잡도 디코더(100)는 로어 트라이앵글러 커널 및 입력 심볼들의 LLR들을 이용하여 출력 심볼들이 각각 가질 수 있는 값들에 대응하는 LLR들을 연산할 수 있다. 저복잡도 디코더(100)는 디코딩을 통해 출력 심볼들에 대하여 순차적으로 값을 결정할 수 있으며, 값이 결정될 차례에 해당하는 출력 심볼은 대상 출력 심볼로 정의될 수 있다.

[0020] 일 실시 예로, 저복잡도 디코더(100)는 입력 심볼들의 LLR들 및 로어 트라이앵글러 커널을 이용하여 대상 출력 심볼에 대응하는 입력 심볼을 제외한 나머지 입력 심볼들에 대한 공통 텀들을 우선적으로 연산할 수 있다. 일 실시 예로, 공통 텀들은 대상 출력 심볼이 어느 하나의 값을 가지는 것을 가정한 때에, 대상 출력 심볼에 대응하는 입력 심볼을 제외한 나머지 입력 심볼들의 조합들을 기반으로 입력 심볼들의 LLR들을 이용하여 합산된 값들을 포함할 수 있다. 나머지 입력 심볼들의 조합들은 나머지 입력 심볼들이 가질 수 있는 값들로 구성된 패턴들을 의미할 수 있다. 공통 텀들에 대한 내용은 도 7a 및 도 7b에서 구체적으로 서술한다. 이후, 저복잡도 디코더(100)는 공통 텀들을 이용하여 대상 출력 심볼의 LLR들을 연산할 수 있으며, 대상 출력 심볼의 LLR들을 기반으로 대상 출력 심볼의 값을 결정할 수 있다. 저복잡도 디코더(100)는 대상 출력 심볼의 값이 결정될 때에, 다음 대상 출력 심볼에 대한 값을 결정하기 위한 디코딩을 수행할 수 있다.

[0021] 본 개시의 실시 예들에 따른 저복잡도 디코더(100)는 디코딩을 수행할 때에, 대상 출력 심볼에 따른 공통 텀들을 미리 연산하고, 대상 출력 심볼의 LLR들을 생성하기 위한 연산 동작시에 공통 텀들을 필요할 때마다 손쉽게 이용함으로써 모든 입력 심볼들의 조합들을 반복하여 연산하는 과정을 생략할 수 있고, 이에 따라, 디코딩의 연산 복잡도를 낮출 수 있는 효과가 있다. 결과적으로, 수신 장치(20)는 비이진 폴라 코드에 대한 디코딩 성능의 열화 없이 효율적으로 디코딩을 수행할 수 있다.

[0022] 도 2는 본 개시의 일 실시 예에 따른 저복잡도 디코더(100)를 나타내는 블록도이다.

[0023] 도 2를 참조하면, 저복잡도 디코더(100)는 커널 변형기(110), LLR 업데이트 모듈(120) 및 결정 모듈(130)을 포함할 수 있다. 이하에서, 저복잡도 디코더(100)에 포함된 블록들은 각각 하드웨어 블록, 소프트웨어 블록 및 하드웨어/소프트웨어가 결합된 블록들로 구현될 수 있다. 더 나아가, 저복잡도 디코더(100)는 하드웨어 로직으로서 직접 비이진 폴라 코드를 수신하여 이에 대한 디코딩을 수행하여 디코딩 결과를 출력하거나, 저복잡도 디코더(100)는 명령 코드 형식으로 메모리(25)에 저장되어 소정의 프로세서의 의해 실행됨으로써 기능을 수행할 수 있는 로직으로 구현될 수 있다. 즉, 공통 텀들을 미리 연산하고, 공통 텀들을 이용하여 비이진 폴라 코드의 디코딩 연산 복잡도를 낮추는 저복잡도 디코더(100)의 구현 예는 다양하며, 특정 구현 예로 제한되지 않음은 분명하다.

[0024] 커널 변형기(110)는 폴라 코드를 생성하기 위해 이용된 최초 커널에 대한 가우시안 엘리미네이션(Gaussian elimination)을 수행하여 최초 커널로부터 로어 트라이앵글러 커널을 생성할 수 있다. 가우시안 엘리미네이션을 통해 생성된 로어 트라이앵글러 커널의 대각선 아래(off-diagonal)의 텀들이 원시 요소(primitive element)들로 구성되고, 로어 트라이앵글러 커널의 지수 값이 최초 커널의 지수 값과 동일한 조건을 만족할 수 있다.

[0025] 일 예로서, 커널 변형기(110)는 참고 문헌 1인 R. Mori and T. Tanaka, "Channel polarization on q-ary discrete memoryless channels by arbitrary kernels," in *Proc. IEEE ISIT 2010* 를 기반으로 로어 트라이앵글러 커널의 대각선 아래(off-diagonal)의 텀들이 원시 요소(primitive)들로 구성되도록 할 수 있다. 이를 통해,

로어 트라이앵글러 커널은 폴라라이징(polarizing)이 가능할 수 있다.

- [0026] 일 예로서, 커널 변형기(110)는 참고 문헌 2인 S. B. Korada, E. Sasoglu, and R. Urbanke, "Polar codes: characterization of exponent, bounds, and constructions," *IEEE Trans. Inform. Theory*, vol. 56, no. 12, Dec. 2010 를 기반으로 로어 트라이앵글러 커널의 부분 거리(partial distance)를 연산하여 이로부터 로어 트라이앵글러 커널의 지수(exponent) 값을 연산할 수 있으며, 최초 커널의 부분 거리를 연산하여, 이로부터 최초 커널의 지수 값을 연산할 수 있다. 커널 변형기(110)는 로어 트라이앵글러 커널의 지수 값과 최초 커널의 지수 값이 동일한지 여부를 확인하고, 동일한 것이 확인되면 로어 트라이앵글러 커널의 생성을 완료할 수 있다.
- [0027] LLR 업데이트 모듈(120)은 송신 장치와 수신 장치간 채널의 특성을 기반으로 비이진 폴라 코드에 포함된 입력 심볼들의 LLR들을 생성할 수 있다. LLR 업데이트 모듈(120)은 로어 트라이앵글러 커널 및 입력 심볼들의 LLR들을 이용하여 출력 심볼들의 LLR들을 생성할 수 있다. 출력 심볼들의 LLR들을 생성하는 동작은 LLR 업데이트 모듈(120)로 정의될 수 있다.
- [0028] 일 실시 예로, LLR 업데이트 모듈(120)은 공통 텀 연산 모듈(125)을 포함할 수 있으며, 공통 텀 연산 모듈(125)은 입력 심볼들의 LLR들 및 로어 트라이앵글러 커널을 이용하여 대상 출력 심볼에 대응하는 입력 심볼을 제외한 나머지 입력 심볼들에 대한 공통 텀들을 우선적으로 연산할 수 있다. 구체적으로, 공통 텀 연산 모듈(125)은 대상 출력 심볼이 어느 하나의 값을 갖는 것을 가정한 때에, 대상 출력 심볼에 대응하는 입력 심볼의 값들에 따른 나머지 입력 심볼들의 조합들을 생성하고, 채널 특성을 기반으로 생성된 입력 심볼들의 LLR들을 이용하여 나머지 입력 심볼들의 조합들에 대한 LLR들을 연산함으로써 공통 텀들을 생성할 수 있다. 예를 들어, 4-ary 폴라 코드의 디코딩을 가정한 때에, 각 입력 심볼은 4개의 값을 가질 수 있으므로, 공통 텀 연산 모듈(125)은 대상 출력 심볼에 대응하는 입력 심볼의 4개의 값들 각각에 따른 나머지 입력 심볼들의 조합들에 대한 LLR들을 연산하여 4개의 공통 텀들을 생성할 수 있다. 이에 대한 내용은 도 7b에서 구체적으로 서술한다.
- [0029] 다른 실시 예로, 공통 텀 연산 모듈(125)은 대상 출력 심볼에 대응하는 입력 심볼 및 적어도 하나의 추가 입력 심볼을 제외한 나머지 입력 심볼들에 대한 공통 텀들을 우선적으로 연산할 수 있다. 추가 입력 심볼은 다음 대상 출력 심볼에 대응하는 입력 심볼일 수 있다.
- [0030] 구체적으로, 공통 텀 연산 모듈(125)은 대상 출력 심볼이 어느 하나의 값(예를 들면, 제1 값)을 갖고, 대상 출력 심볼에 대응하는 입력 심볼이 어느 하나의 값(예를 들면, 제1 값)을 갖는 것을 가정한 때에, 추가 입력 심볼의 값들에 따른 나머지 입력 심볼들의 조합들을 생성하고, 채널 특성을 기반으로 생성된 입력 심볼들의 LLR들을 이용하여 나머지 입력 심볼들의 조합들에 대한 LLR들을 연산함으로써 공통 텀들을 생성할 수 있다. 이와 같은 방식으로, 공통 텀 연산 모듈(125)은 대상 출력 심볼이 어느 하나의 값(예를 들면, 제1 값)을 갖고, 대상 출력 심볼에 대응하는 입력 심볼이 다른 하나의 값(예를 들면, 제2 값)을 갖는 것을 가정하여, 위와 같은 방식으로 다른 공통 텀들을 생성할 수 있다. 이에 대한 내용은 도 8a 및 도 8b에서 구체적으로 서술한다. 공통 텀 연산 모듈(125)은 생성된 공통 텀들을 메모리(25)에 저장할 수 있다. 공통 텀 연산 모듈(125)은 수신 장치(20, 도 1)의 성능 또는 동작 환경을 고려하여 추가 입력 심볼의 개수를 가변적으로 조절할 수 있다. 예를 들어, 수신 장치(20, 도 1)의 성능 또는 동작 환경을 고려할 때, 연산 로직이 단순하여 복잡한 계산을 빠르게 수행하지 못하거나, 낮은 레이턴시가 필요하지 않은 경우에는 공통 텀 연산 모듈(125)은 추가 입력 심볼의 개수를 많게 조절할 수 있다. 또한, 수신 장치(20, 도 1)의 성능 또는 동작 환경을 고려할 때, 연산 로직이 복잡하여 복잡한 계산을 빠르게 수행할 수 있거나, 낮은 레이턴시가 필요한 경우에는 공통 텀 연산 모듈(125)은 추가 입력 심볼의 개수를 적게 조절할 수 있다.
- [0031] LLR 업데이트 모듈(120)은 메모리(25)에 저장된 공통 텀들을 리드(read)하여 대상 출력 심볼의 LLR들을 생성하기 위한 연산 동작시에 공통 텀들을 이용할 수 있다. 구체적으로, LLR 업데이트 모듈(120)은 채널 특성을 기반으로 생성된 입력 심볼들의 LLR들로부터 대상 출력 심볼이 가질 수 있는 값들에 따른 대상 출력 심볼에 대응하는 입력 심볼의 LLR들을 획득할 수 있다. 이후, LLR 업데이트 모듈(120)은 획득한 입력 심볼의 LLR들과 공통 텀들을 각각 대응하는 성분끼리 합산함으로써 대상 출력 심볼의 LLR들을 생성할 수 있다.
- [0032] LLR 업데이트 모듈(120)은 대상 출력 심볼의 LLR들을 생성하여 결정 모듈(130)에 제공할 수 있다. 결정 모듈(130)은 대상 출력 심볼의 LLR들 중 가장 큰 값을 갖는 LLR에 대응하는 값을 대상 출력 심볼의 값으로 결정할 수 있다. 이후, LLR 업데이트 모듈(120)은 전술한 방식으로 다음 대상 출력 심볼의 LLR들을 생성할 수 있다.
- [0033] 저복잡도 디코더(100)는 모든 대상 출력 심볼들의 값을 결정함으로써 디코딩을 완료할 수 있으며, 수신 장치(20, 도 1) 내의 프로세서, AP(Application Processor), SoC(System On Chip), 모뎀(modem) 등의 데이터 처리

를 수행하는 블록에 제공할 수 있다.

- [0034] 일 실시 예로, 메모리(25)는 공통 텀 연산 모듈(125)에 의해 생성된 공통 텀들을 저장할 수 있으며, 공통 텀들에 대응하는 대상 출력 심볼의 LLR들이 생성된 이후에 공통 텀들을 삭제하여 다른 공통 텀들을 저장하기 위한 메모리 공간을 확보할 수 있다.
- [0035] 도 3은 도 2의 커널 변형기(110)의 동작을 설명하기 위한 순서도이고, 도 4는 도 2의 단계 S110을 구체적으로 설명하기 위한 순서도이다. 이하에서는, 이해를 돕기 위해 도 2를 참조하여 서술한다.
- [0036] 도 3을 참조하면, 커널 변형기(110)는 최초 커널로부터 로어 트라이앵글러 커널을 생성할 수 있다(S100). 최초 커널은 비이진 폴라 코드를 생성하기 위해 이용된 폴라 코드 생성 행렬을 형성하는 기본 단위이며, 커널 변형기(110)는 미리 저장된 최초 커널을 변형하여 디코딩시에 연산 복잡도를 낮출 수 있는 로어 트라이앵글러 커널을 생성할 수 있다. 커널 변형기(110)는 로어 트라이앵글러 커널을 LLR 업데이트 모듈(120)에 제공할 수 있다(S110). LLR 업데이트 모듈(120)은 로어 트라이앵글러 커널 및 채널 특성을 기반으로 생성된 비이진 폴라 코드에 포함된 입력 심볼들의 LLR들을 이용하여 디코딩을 수행할 수 있다.
- [0037] 도 4를 참조하면, 커널 변형기(110)는 최초 커널에 대한 가우시안 엘리미네이션을 수행할 수 있다(S102). 커널 변형기(110)는 가우시안 엘리미네이션을 수행하면서 로어 트라이앵글러 커널의 대각선 아래 텀들이 원시 요소들인지 여부를 판별할 수 있다(S104). 커널 변형기(110)는 로어 트라이앵글러 커널의 대각선 아래 텀들이 원시 요소들로 구성된 때에(S104, Yes), 로어 트라이앵글러 커널의 지수 값과 최초 커널의 지수 값이 동일한지 여부를 판별할 수 있다(S106). 커널 변형기(110)는 로어 트라이앵글러 커널의 지수 값과 최초 커널의 지수 값이 동일한 때에(S106, Yes), 단계 S110을 수행할 수 있다. 또한, 커널 변형기(110)는 로어 트라이앵글러 커널의 대각선 아래 텀들이 원시 요소들로 구성되지 않은 때(S104, No), 또는 로어 트라이앵글러 커널의 지수 값과 최초 커널의 지수 값이 동일하지 않은 때(S106, No), 단계 S102로 돌아가 로어 트라이앵글러 커널의 조건을 만족할 수 있도록 최초 커널에 대한 가우시안 엘리미네이션을 재수행할 수 있다. 도 4에서는 단계 S104와 단계 S106가 순차적으로 수행되는 것으로 도시되어 있으나, 이에 국한되지 않고, 단계 S102와 동시에 수행될 수 있다.
- [0038] 도 5a는 최초 커널(G_{KN})에 대한 설명을 위한 도면이고, 도 5b는 로어 트라이앵글러 커널(G_{LTKN})에 대한 설명을 위한 도면이다.
- [0039] 이하에서는, 비이진 폴라 코드가 4-ary 폴라 코드인 것을 가정하여 서술하나, 이는 설명의 편의를 위해 예시적으로 서술한 것에 불과하므로, 본 개시의 사상은 더 높은 기수를 갖는 폴라 코드에 대한 디코딩에도 적용될 수 있음은 분명하다. u 벡터는 1×4 크기를 가지며, 4개의 출력 심볼들(u_0-u_3)을 포함하고, x 벡터는 1×4 크기를 가지며, 4개의 입력 심볼들(x_0-x_3)을 포함할 수 있다. 최초 커널(G_{KN}) 및 로어 트라이앵글러 커널(G_{LTKN})은 4×4 크기를 가지는 폴라 코드 생성 행렬의 기본 단위로서, 각각 도 1에서 서술된 수식 1을 만족할 수 있다. 출력 심볼들(u_0-u_3) 및 입력 심볼들(x_0-x_3)은 각각 '0', '1', 'a', 'a²'의 값을 가질 수 있다.
- [0040] 도 5a를 참조하면, 최초 커널(G_{KN})에 의할 때, 제1 케이스(Case_1)에서 u 벡터는 $[0 \ 0 \ 1 \ 1]$ 일 때에, x 벡터는 $[a \ a^2 \ 0 \ a]$ 이고, 제2 케이스(Case_2)에서 u 벡터는 $[1 \ 0 \ 1 \ 1]$ 일 때에, x 벡터는 $[a^2 \ a \ 1 \ a]$ 이고, 제3 케이스(Case_3)에서 u 벡터는 $[a \ 0 \ 1 \ 1]$ 일 때에, x 벡터는 $[0 \ 1 \ a \ a]$ 이며, 제4 케이스(Case_4)에서 u 벡터는 $[a^2 \ 0 \ 1 \ 1]$ 일 때에, x 벡터는 $[1 \ 0 \ a^2 \ a]$ 이다. 이와 같이, 제1 내지 제4 케이스(Case_1-Case_4)를 살펴보면, 최초 커널(G_{KN})에 의한 x 벡터와 u 벡터와의 관계에서 제1 출력 심볼(u_0)를 제외한 제2 내지 제4 출력 심볼(u_1-u_3)의 값이 동일하여도 x 벡터의 제2 내지 제4 입력 심볼(x_1-x_3)의 값의 패턴은 케이스(Case_1-Case_4) 별로 각각 상이함을 확인할 수 있다. 이에 따라, 최초 커널(G_{KN})을 이용한 비이진 폴라 코드의 디코딩은 연산 복잡도가 높을 수 밖에 없는 문제가 있었다.
- [0041] 도 5b를 참조하면, 로어 트라이앵글러 커널(G_{LTKN})에 의할 때에, 제1 케이스(Case_1)에서 u 벡터는 $[0 \ 0 \ 1 \ 1]$ 일 때에, x 벡터는 $[a \ a^2 \ 0 \ a]$ 이고, 제2 케이스(Case_2)에서 u 벡터는 $[1 \ 0 \ 1 \ 1]$ 일 때에, x 벡터는 $[a^2 \ a^2 \ 0 \ a]$ 이고, 제3 케이스(Case_3)에서 u 벡터는 $[a \ 0 \ 1 \ 1]$ 일 때에, x 벡터는 $[0 \ a^2 \ 0 \ a]$ 이며, 제4 케이스(Case_4)에서 u 벡터는 $[a^2 \ 0 \ 1 \ 1]$ 일 때에, x 벡터는 $[1 \ a^2 \ 0 \ a]$ 이다. 이와 같이, 제1 내지 제4 케이스(Case_1-Case_4)를 살펴보면, 로어 트라이앵글러 커널(G_{LTKN})에 의한 x 벡터와 u 벡터와의 관계에서 제1 출력 심

볼(u_0)를 제외한 제2 내지 제4 출력 심볼(u_1-u_3)의 값이 동일할 때에, x 벡터의 제2 내지 제4 입력 심볼(x_1-x_3)의 값의 패턴은 케이스(Case_1-Case_4)에서 모두 동일함을 확인할 수 있다. 도 5b에서처럼 제1 출력 심볼(u_0)의 값을 결정할 때에, 제2 내지 제4 출력 심볼(u_1-u_3)의 값의 패턴이 동일한 경우, x 벡터의 제2 내지 제4 입력 심볼(x_1-x_3)의 값의 패턴도 동일하게 나타나기 때문에 이러한 점은 공통 텀들을 미리 연산할 수 있는 근거가 될 수 있다. 즉, 본 개시의 실시 예에 따른 디코딩 방법은 대상 출력 심볼(예를 들면, 제1 출력 심볼(u_0))에 대응하는 입력 심볼(예를 들면, 제1 입력 심볼(x_1))을 제외한 나머지 입력 심볼들(예를 들면, 제2 내지 제4 입력 심볼(x_1-x_3))의 조합들에 관련한 공통 텀들을 미리 생성할 수 있으며, 생성된 공통 텀들은 대상 출력 심볼(예를 들면, 제1 출력 심볼(u_0))의 LLR들을 연산하는 데에 이용될 수 있다.

[0042] 또한, 로어 트라이 앵글러 커널(G_{LTKN})을 통해 ' $x_0=u_0+u_1+a^2 u_2+u_3$, $x_1=u_1+a u_2+u_3$, $x_2=u_2+u_3$, $x_0=a u_3$ '의 관계식을 얻을 수 있는 바, 정리하면, 제2 입력 심볼(x_1)은 제1 출력 심볼(u_0)에 의존하지 않고, 제3 입력 심볼(x_2)은 제1 출력 심볼(u_1) 및 제2 출력 심볼(u_2)에 의존하지 않고, 제4 입력 심볼(x_3)은 제1 내지 제3 출력 심볼(u_0-u_2)에 의존하지 않을 수 있도록 로어 트라이 앵글러 커널(G_{LTKN})이 구성될 수 있다.

[0043] 도 6은 도 1의 저복잡도 디코더(100)의 디코딩 방법을 설명하기 위한 도면이다. 이하에서는, 설명을 돕기 위해 도 1을 참조하여 설명된다.

[0044] 도 6을 참조하면, 저복잡도 디코더(100)는 수신 장치(20)가 수신한 비이진 폴라 코드에 포함된 입력 심볼들에 대한 LLR들을 채널(CH) 특성을 기반으로 생성할 수 있다. 저복잡도 디코더(100)는 최초 커널에 대한 가우시안 엘리미네이션을 통해 로어 트라이앵글러 커널을 생성할 수 있다. 저복잡도 디코더(100)는 로어 트라이앵글러 커널을 이용하여 대상 출력 심볼에 대응하는 공통 텀들을 우선적으로 연산할 수 있다(S200). 대상 출력 심볼에 대응하는 공통 텀들은 대상 출력 심볼의 값을 결정하기 위해 미리 연산되는 공통 텀들을 의미할 수 있다. 구체적으로, 저복잡도 디코더(100)는 대상 출력 심볼이 어느 하나의 값을 갖는 것을 가정하고, 대상 출력 심볼에 대응하는 입력 심볼의 값에 따른 나머지 입력 심볼들의 조합들을 생성하고, 조합들에 대한 LLR들을 연산함으로써 공통 텀들을 생성할 수 있다. 저복잡도 디코더(100)는 생성한 공통 텀들을 소정의 메모리에 저장할 수 있다.

[0045] 저복잡도 디코더(100)는 메모리로부터 공통 텀들을 리드하여 공통 텀들과 입력 심볼들의 LLR들을 이용하여 대상 출력 심볼의 LLR들을 연산할 수 있다(S210). 저복잡도 디코더(100)는 대상 출력 심볼의 LLR들을 이용하여 대상 출력 심볼의 값을 결정할 수 있다(S220).

[0046] 도 7a는 도 2의 저복잡도 디코더(100)의 디코딩 방법을 구체적으로 설명하기 위한 도면이고, 도 7b는 도 7a의 디코딩 방법으로 생성된 공통 텀들을 구체적으로 설명하기 위한 테이블(TB1)이다. 이하에서는, 이해를 돕기 위해 도 7a 및 도 7b는 도 2를 참조하여 설명된다. 또한, 이하에서 도시된 ' $b+1$, $b+a$, $b+a^2$ '의 연산은 일반 사칙연산이 아닌 소정의 갈루아 필드 기반 연산임을 가정한다.

[0047] 도 7a를 참조하면, LLR 업데이트 모듈(120)은 송신 장치와 수신 장치간 채널의 특성을 기반으로 4-ary 폴라 코드에 포함된 입력 심볼들(x_0-x_3)의 LLR들을 생성할 수 있다. 입력 심볼들(x_0-x_3)의 LLR들은 제1 행렬(ISM)로 정리될 수 있다. ' b '는 입력 심볼들(x_0-x_3) 및 출력 심볼들(u_0-u_3) 각각이 가질 수 있는 값을 나타내는 것으로 ' b '는 '0', '1', ' a ', ' a^2 ' 중 어느 하나의 값에 해당할 수 있다. 예를 들

[0048] 어, ' $\lambda_0^{(0)}$ '은 제1 입력 심볼(x_0)이 '0'의 값을 가질 수 있는 확률을 나타내는 LLR이고, ' $\lambda_0^{(1)}$ '은 제1 입력 심볼(x_0)이 '1'의 값을 가질 수 있는 확률을 나타내는 LLR이고, ' $\lambda_0^{(a)}$ '은 제1 입력 심볼(x_0)이 ' a '의 값을 가질 수 있는 확률을 나타내는 LLR이며, ' $\lambda_0^{(a^2)}$ '은 제1 입력 심볼(x_0)이 ' a^2 '의 값을 가질 수 있는 확률을 나타내는 LLR이다. 이와 같은 방식으로, 제2 내지 제4 입력 심볼(x_1-x_3)의 LLR들이 정의될 수 있다.

[0049] LLR 업데이트 모듈(120)은 로어 트라이앵글러 커널(G_{LTKN}) 및 제1 행렬(ISM)을 이용한 연산을 통해 출력 심볼들(u_0-u_3)의 LLR들 생성할 수 있다. 출력 심볼들(u_0-u_3)의 LLR들은 제2 행렬(OSM)로 정리될 수 있다. 예를 들어,

$\Lambda_0^{(0)}$ '은 제1 출력 심볼(u_0)이 '0'의 값을 가질 수 있는 확률을 나타내는 LLR이고, $\Lambda_0^{(1)}$ '은 제1 출력 심볼(u_0)이 '1'의 값을 가질 수 있는 확률을 나타내는 LLR이고, $\Lambda_0^{(\alpha)}$ '은 제1 출력 심볼(u_0)이 'a'의 값을 가질 수 있는 확률을 나타내는 LLR이며, $\Lambda_0^{(a^2)}$ '은 제1 출력 심볼(u_0)이 'a²'의 값을 가질 수 있는 확률을 나타내는 LLR이다. 이와 같은 방식으로, 제2 내지 제4 출력 심볼(u_1 - u_3)의 LLR들이 정의될 수 있다.

[0050] 구체적으로, LLR 업데이트 모듈(120)은 제1 수식(F1)을 기반으로 로어 트라이앵글러 커널(G_{LTKN}) 및 제1 행렬(ISM)을 이용한 연산을 수행하여 출력 심볼들(u_0 - u_3)의 LLR들을 순차적으로 생성할 수 있다. 일 예로, 제1 출력 심볼(u_0), 제2 출력 심볼(u_1), 제3 출력 심볼(u_2), 제4 출력 심볼(u_3) 순으로 LLR들을 생성할 수 있다.

[0051] 먼저, LLR 업데이트 모듈(120)은 제1 출력 심볼(u_0)의 LLR들($\Lambda_0^{(b)}$)을 생성할 수 있으며, 이 때에, 공통 텀 연산 모듈(125)은 우선적으로 제1 출력 심볼(u_0)이 어느 하나의 값을 가지는 것을 가정한 후에, 제1 출력 심볼(u_0)에 대응하는 제1 입력 심볼(x_0)을 제외한 나머지 입력 심볼들(x_1 - x_3)의 조합들, 즉, 나머지 입력 심볼들(x_1 - x_3)이 가질 수 있는 값들로 구성된 패턴들을 생성할 수 있다. 예를 들어, 제1 입력 심볼(x_0)이 '0' 값을 가질 때에, 나머지 입력 심볼들(x_1 - x_3)은 제1 조합들을 가질 수 있고, 제1 입력 심볼(x_0)이 '1' 값을 가질 때에, 나머지 입력 심볼들(x_1 - x_3)은 제2 조합들을 가질 수 있고, 제1 입력 심볼(x_0)이 'a' 값을 가질 때에, 나머지 입력 심볼들(x_1 - x_3)은 제3 조합들을 가질 수 있으며, 제1 입력 심볼(x_0)이 'a²' 값을 가질 때에, 나머지 입력 심볼들(x_1 - x_3)은 제4 조합들을 가질 수 있다. 공통 텀 연산 모듈(125)은 제1 행렬(ISM)을 이용하여 제1 내지 제4 조합들에 대한 LLR들(A, B, C, D)을 각각 생성할 수 있다. 제1 내지 제4 조합들에 대한 LLR들(A, B, C, D)은 공통 텀들로 지칭될 수 있다. 공통 텀 연산 모듈(125)은 생성된 공통 텀들(A, B, C, D)을 메모리(25)에 저장할 수 있다.

[0052] LLR 업데이트 모듈(120)은 제2 수식(F2)을 기반으로 제1 행렬(ISM) 및 공통 텀들(A, B, C, D)을 이용하여 제1 출력 심볼(u_0)의 LLR들($\Lambda_0^{(b)}$)을 생성할 수 있다. 구체적으로, LLR 업데이트 모듈(120)은 구비된 복수의 에더들(Adder_1-Adder_4)을 이용하여 제1 입력 심볼(x_0)이 '0'일 때의 LLR($\lambda_0^{(0)}$)과 'A' 공통 텀의 합산 값, 제1 입력 심볼(x_0)이 '1'일 때의 LLR($\lambda_0^{(1)}$)과 'B' 공통 텀의 합산 값, 제1 입력 심볼(x_0)이 'a'일 때의 LLR($\lambda_0^{(\alpha)}$)과 'C' 공통 텀의 합산 값 및 제1 입력 심볼(x_0)이 'a²'일 때의 LLR($\lambda_0^{(a^2)}$)과 'D' 공통 텀의 합산 값을 생성할 수 있으며, 구비된 연산 블록(CB)을 이용하여 합산 값들 중 가장 큰 값으로 제1 출력 심볼(u_0)이 '0'일 때의 LLR($\Lambda_0^{(0)}$)을 결정할 수 있다. 이와 같은 방식으로, LLR 업데이트 모듈(120)은 공통 텀들(A, B, C, D)을 이용하여 빠르게 제1 출력 심볼(u_0)의 나머지 LLR들($\Lambda_0^{(1)}, \Lambda_0^{(\alpha)}, \Lambda_0^{(a^2)}$)을 구할 수 있으며, 구체적인 내용은 생략한다.

[0053] LLR 업데이트 모듈(120)은 생성된 제1 출력 심볼(u_0)의 LLR들($\Lambda_0^{(0)}, \Lambda_0^{(1)}, \Lambda_0^{(\alpha)}, \Lambda_0^{(a^2)}$)을 결정 모듈(130)에 제공할 수 있으며, 결정 모듈(130)은 제1 출력 심볼(u_0)의 LLR들($\Lambda_0^{(0)}, \Lambda_0^{(1)}, \Lambda_0^{(\alpha)}, \Lambda_0^{(a^2)}$) 중 가장 큰 LLR에 대응하는 값을 갖는 것으로 결정할 수 있다.

[0054] 이후, LLR 업데이트 모듈(120)은 제2 출력 심볼(u_1)의 LLR들($\Lambda_1^{(b)}$)을 생성할 수 있으며, 이 때에, 공통 텀 연산 모듈(125)은 우선적으로 제2 출력 심볼(u_1)이 어느 하나의 값을 가지는 것을 가정한 후에, 제2 출력 심볼(u_1)에 대응하는 제2 입력 심볼(x_1) 및 값이 결정된 제1 출력 심볼(u_0)에 대응하는 제1 입력 심볼(x_0)을 제외한 나머지

입력 심볼들(x_2, x_3)의 조합들, 즉, 나머지 입력 심볼들(x_2, x_3)이 가질 수 있는 값들로 구성된 패턴들을 생성할 수 있다. 공통 텀 연산 모듈(125)은 제1 행렬(ISM)을 이용하여 조합들 대한 LLR들을 각각 생성할 수 있다. 이후의 공통 텀 연산 모듈(125), LLR 업데이트 모듈(120) 및 결정 모듈(130)의 동작은 제1 출력 심볼(u_0)의 값을 결정할 때와 유사한 바, 구체적인 내용은 생략한다.

[0055] 도 7b를 참조하면, 테이블(TB1)에 도시된 바와 같이, 공통 텀 연산 모듈(125)은 대상 출력 심볼이 제1 출력 심볼(u_0)일 때, 제1 입력 심볼(u_0)이 제1 값(b)인 경우 이에 대응하는 'A' 공통 텀을 생성하고, 제1 입력 심볼(u_0)이 제2 값(b+1)인 경우 이에 대응하는 'B' 공통 텀을 생성하고, 제1 입력 심볼(u_0)이 제3 값(b+a)인 경우 이에 대응하는 'C' 공통 텀을 생성하며, 제1 입력 심볼(u_0)이 제4 값(b+a²)인 경우 이에 대응하는 'D' 공통 텀을 생성할 수 있다. 또한, 공통 텀 연산 모듈(125)은 대상 출력 심볼이 제2 출력 심볼(u_1)일 때, 제2 입력 심볼(u_1)이 제1 값(b)인 경우 이에 대응하는 'A' 공통 텀을 생성하고, 제2 입력 심볼(u_1)이 제2 값(b+1)인 경우 이에 대응하는 'B' 공통 텀을 생성하고, 제2 입력 심볼(u_1)이 제3 값(b+a)인 경우 이에 대응하는 'C' 공통 텀을 생성하며, 제2 입력 심볼(u_1)이 제4 값(b+a²)인 경우 이에 대응하는 'D' 공통 텀을 생성할 수 있다. 이와 같은 방식으로, 공통 텀 연산 모듈(125)은 대상 출력 심볼이 제3 출력 심볼(u_2) 및 제4 출력 심볼(u_3) 중 어느 하나일 때에 이에 대응하는 공통 텀들을 생성할 수 있다.

[0056] 도 8a는 도 2의 저복잡도 디코더(100)의 디코딩 방법의 일 실시 예를 설명하기 위한 도면이고, 도 8b는 도 8a의 디코딩 방법으로 생성된 공통 텀들을 설명하기 위한 테이블(TB2)이다. 이하에서는, 이해를 돕기 위해 도 8a 및 도 8b는 도 2 및 도 7a를 참조하여 설명된다.

[0057] 도 8a를 참조하면, 공통 텀 연산 모듈(125)은 전술한 바와 같이, 대상 출력 심볼에 대응하는 입력 심볼 및 적어도 하나의 추가 입력 심볼을 제외한 나머지 입력 심볼들에 대한 공통 텀들을 우선적으로 연산할 수 있다. 일 예로, 공통 텀 연산 모듈(125)은 제1 출력 심볼(u_0)의 LLR들($\Lambda_0^{(b)}$)을 생성할 때에, 추가 입력 심볼을 제2 입력 심볼(x_1)로 설정하여, 제1 입력 심볼(x_0) 및 제2 입력 심볼(x_1)을 제외한 나머지 입력 심볼들(x_2, x_3)의 조합들을 생성하고, 조합들에 대한 LLR들을 연산함으로써 공통 텀들을 생성할 수 있다.

[0058] 구체적으로, 공통 텀 연산 모듈(125)은 우선적으로 제1 출력 심볼(u_0)이 어느 하나의 값을 갖고, 제1 입력 심볼(x_1)이 어느 하나의 값을 갖는 것을 가정한 후에, 제1 입력 심볼(x_0) 및 제2 입력 심볼(x_1)을 제외한 나머지 입력 심볼들(x_2, x_3)의 조합들, 즉, 나머지 입력 심볼들(x_2, x_3)이 가질 수 있는 값들로 구성된 패턴들을 생성할 수 있다. 예를 들어, 제2 입력 심볼(x_1)이 '0' 값을 가질 때에, 나머지 입력 심볼들(x_2, x_3)은 제1 조합들을 가질 수 있고, 제2 입력 심볼(x_1)이 '1' 값을 가질 때에, 나머지 입력 심볼들(x_2, x_3)은 제2 조합들을 가질 수 있고, 제2 입력 심볼(x_1)이 'a' 값을 가질 때에, 나머지 입력 심볼들(x_2, x_3)은 제3 조합들을 가질 수 있으며, 제2 입력 심볼(x_1)이 'a²' 값을 가질 때에, 나머지 입력 심볼들(x_2, x_3)은 제4 조합들을 가질 수 있다. 공통 텀 연산 모듈(125)은 제1 행렬(ISM)을 이용하여 제1 내지 제4 조합들에 대한 LLR들(A_0, A_1, A_2, A_3)을 각각 생성할 수 있다. 제1 내지 제4 조합들에 대한 LLR들(A_0, A_1, A_2, A_3)은 공통 텀들로 지칭될 수 있다. 공통 텀 연산 모듈(125)은 생성된 공통 텀들(A_0, A_1, A_2, A_3)을 메모리(25)에 저장할 수 있다.

[0059] 도 8b를 참조하면, 테이블(TB2)에 도시된 바와 같이, 공통 텀 연산 모듈(125)은 대상 출력 심볼이 제1 출력 심볼(u_0)이고, 제1 입력 심볼(u_0)이 제1 값(b)인 때에, 제2 입력 심볼(u_1)이 제1 값(b)인 경우 이에 대응하는 'A₀' 공통 텀을 생성하고, 제2 입력 심볼(u_1)이 제2 값(b+1)인 경우 이에 대응하는 'A₁' 공통 텀을 생성하고, 제2 입력 심볼(u_1)이 제3 값(b+a)인 경우 이에 대응하는 'A₂' 공통 텀을 생성하며, 제2 입력 심볼(u_1)이 제4 값(b+a²)인 경우 이에 대응하는 'A₃' 공통 텀을 생성할 수 있다.

[0060] 또한, 공통 텀 연산 모듈(125)은 대상 출력 심볼이 제1 출력 심볼(u_0)이고, 제1 입력 심볼(u_0)이 제2 값(b+1)인

때에, 제2 입력 심볼(u_1)이 제1 값(b)인 경우 이에 대응하는 ' B_0 ' 공통 텀을 생성하고, 제2 입력 심볼(u_1)이 제2 값($b+1$)인 경우 이에 대응하는 ' B_1 ' 공통 텀을 생성하고, 제2 입력 심볼(u_1)이 제3 값($b+a$)인 경우 이에 대응하는 ' B_2 ' 공통 텀을 생성하며, 제2 입력 심볼(u_1)이 제4 값($b+a^2$)인 경우 이에 대응하는 ' B_3 ' 공통 텀을 생성할 수 있다. 이와 같은 방식으로, 공통 텀 연산 모듈(125)은 제2 입력 심볼(u_0)이 제3 값 또는 제4 값인 때의 공통 텀들을 구할 수 있으며, 더 나아가, 대상 출력 심볼이 제2 내지 제4 출력 심볼(u_1-u_3) 중 어느 하나일 때에 이에 대응하는 공통 텀들을 생성할 수 있다.

[0061] 도 9는 본 개시의 일 실시 예에 따른 저복잡도 디코더(100')를 나타내는 블록도이다.

[0062] 저복잡도 디코더(100')는 도 2의 저복잡도 디코더(100)와 비교하여 모드 제어기(140)를 더 포함할 수 있으며, 다른 구성들은 도 2에서 구체적으로 서술된 바, 모드 제어기(140)를 중심으로 서술하도록 한다.

[0063] 도 9를 참조하면, 모드 제어기(140)는 저복잡도 디코더(100')가 포함된 수신 장치에 대한 성능 정보 및 동작 환경 등의 다양한 정보들을 포함하는 상태 정보(SI)를 수신하여, 상태 정보(SI)를 기반으로 공통 텀 연산 모듈(125)에 공통 텀들을 연산할 때의 추가 입력 심볼의 개수를 조절하기 위한 모드 제어신호(M_CS)를 생성하여 LLR 업데이트 모듈(120)(또는, 공통 텀 연산 모듈(125))에 제공할 수 있다. 예를 들어, 수신 장치의 연산 로직이 단순하여 복잡한 계산을 빠르게 수행하지 못하거나, 낮은 레이턴시가 필요하지 않은 경우에는, 모드 제어기(140)는 LLR 업데이트 모듈(120)이 연산 복잡도의 일정 정도만 낮출 수 있는 제1 모드로 디코딩을 수행하도록 LLR 업데이트 모듈(120)에 모드 제어신호(M_CS)를 제공할 수 있다. 또한, 수신 장치의 연산 로직이 복잡하여 복잡한 계산을 빠르게 수행할 수 있거나, 낮은 레이턴시가 필요한 경우에는 모드 제어기(140)는 LLR 업데이트 모듈(120)이 연산 복잡도의 일정 정도 이상을 낮출 수 있는 제2 모드로 디코딩을 수행하도록 LLR 업데이트 모듈(120)에 모드 제어신호(M_CS)를 제공할 수 있다.

[0064] 공통 텀 연산 모듈(125)은 모드 제어신호(M_CS)를 기반으로 제1 모드일 때에는 추가 입력 심볼의 개수를 제1 개수로 설정하고, 제2 모드일 때에는 추가 입력 심볼의 개수를 제2 개수로 설정할 수 있으며, 제2 개수는 제1 개수보다 작은 값일 수 있다.

[0065] 도 10은 본 개시의 일 실시예에 따른 전자 장치(1000)를 나타내는 블록도이다.

[0066] 도 10을 참조하면, 전자 장치(1000)는 메모리(1010), 프로세서 유닛(Processor Unit)(1020), 입출력 제어부(1040), 표시부(1050), 입력 장치(1060) 및 통신 처리부(1090)를 포함할 수 있다. 여기서, 메모리(1010)는 다수 개 존재할 수도 있다. 각 구성요소에 대해 살펴보면 다음과 같다.

[0067] 메모리(1010)는 전자 장치의 동작을 제어하기 위한 프로그램을 저장하는 프로그램 저장부(1011) 및 프로그램 수행 중에 발생하는 데이터를 저장하는 데이터 저장부(1012)를 포함할 수 있다. 데이터 저장부(1012)는 애플리케이션 프로그램(1013), 저복잡도 디코딩 프로그램(1014)의 동작에 필요한 데이터를 저장할 수 있다. 프로그램 저장부(1011)는 애플리케이션 프로그램(1013), 저복잡도 디코딩 프로그램(1014)을 포함할 수 있다. 여기서, 프로그램 저장부(1011)에 포함되는 프로그램은 명령어들의 집합으로 명령어 세트(instruction set)로 표현할 수도 있다.

[0068] 애플리케이션 프로그램(1013)은 전자 장치에서 동작하는 애플리케이션 프로그램을 포함한다. 즉, 애플리케이션 프로그램(1013)은 프로세서(1022)에 의해 구동되는 애플리케이션의 명령어를 포함할 수 있다. 저복잡도 디코딩 프로그램(1014)은 본 개시의 실시 예들에 따른 디코딩 방법에서 공통 텀들을 미리 연산하여 출력 심볼들의 LLR들을 생성할 때에 빠르게 이용할 수 있는 동작을 제어할 수 있다.

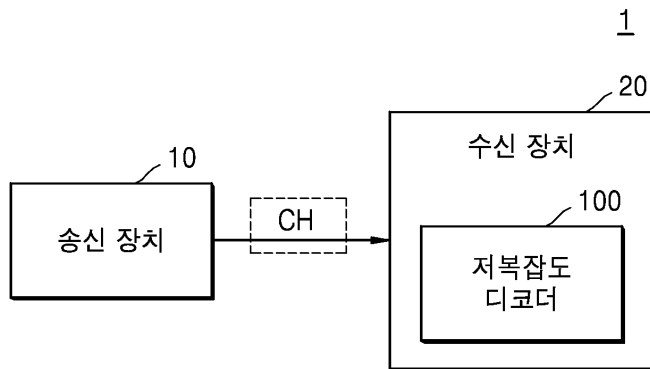
[0069] 주변 장치 인터페이스(1023)는 기지국의 입출력 주변 장치와 프로세서(1022) 및 메모리 인터페이스(1021)의 연결을 제어할 수 있다. 프로세서(1022)는 적어도 하나의 소프트웨어 프로그램을 사용하여 기지국이 해당 서비스를 제공하도록 제어한다. 이때, 프로세서(1022)는 메모리(1010)에 저장되어 있는 적어도 하나의 프로그램을 실행하여 해당 프로그램에 대응하는 서비스를 제공할 수 있다.

[0070] 입출력 제어부(1040)는 표시부(1050) 및 입력 장치(1060) 등의 입출력 장치와 주변 장치 인터페이스(1023) 사이에 인터페이스를 제공할 수 있다. 표시부(1050)는 상태 정보, 입력되는 문자, 동영상(moving picture) 및 정지 영상(still picture) 등을 표시한다. 예를 들어, 표시부(1050)는 프로세서(1022)에 의해 구동되는 응용프로그램 정보를 표시할 수 있다.

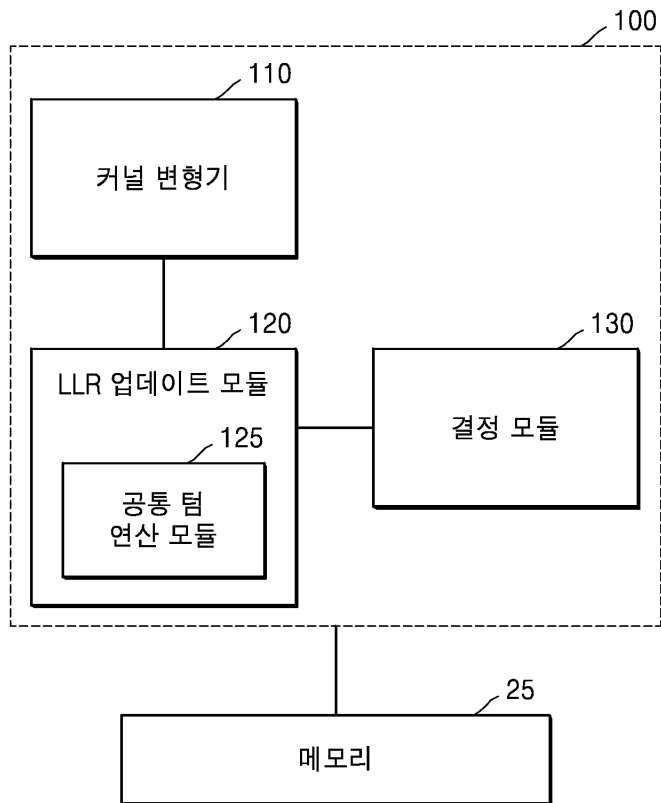
- [0071] 입력 장치(1060)는 전자 장치의 선택에 의해 발생하는 입력 데이터를 입출력 제어부(1040)를 통해 프로세서 유닛(1020)으로 제공할 수 있다. 이때, 입력 장치(1060)는 적어도 하나의 하드웨어 버튼을 포함하는 키패드 및 터치 정보를 감지하는 터치 패드 등을 포함할 수 있다. 예를 들어, 입력 장치(1060)는 터치 패드를 통해 감지한 터치, 터치 움직임, 터치 해제 등의 터치 정보를 입출력 제어부(1040)를 통해 프로세서(1022)로 제공할 수 있다.
- [0072] 전자 장치(1000)는 음성 통신 및 데이터 통신을 위한 통신 기능을 수행하는 통신 처리부(1090)를 포함하고, 통신 처리부(1090)는 다양한 통신 환경(5G 등의 차세대 통신 환경 포함)에서 비이진 폴라 코드를 수신할 수 있으며, 이러한 비이진 폴라 코드를 주변 장치 인터페이스(1020)를 통해 프로세서(1422)에 전달할 수 있다.
- [0073] 도 11은 본 개시의 일 실시예에 따른 컴퓨팅 시스템(1100)을 나타내는 블록도이다.
- [0074] 도 11을 참조하면, 컴퓨팅 시스템(1100)은 호스트(1200) 및 메모리 시스템(1300)을 포함할 수 있다. 메모리 시스템(1300)은 메모리 컨트롤러(1310), 메모리 장치(1320)를 포함할 수 있다. 메모리 컨트롤러(1310)는 본 개시의 실시 예들에 따른 저복잡도 디코더(1315)를 포함할 수 있다. 메모리 컨트롤러(1310)는 호스트(1200)로부터 비이진 폴라 코드 형태의 신호들을 수신할 수 있으며, 저복잡도 디코더(1315)는 비이진 폴라 코드 형태의 신호들을 본 개시의 실시 예들에 따른 디코딩 방법으로 디코딩을 수행할 수 있다. 메모리 컨트롤러(1310)는 디코딩 결과를 이용하여 ECC를 수행하거나 메모리 장치(1320)와의 메모리 동작을 수행할 수 있다.
- [0075] 이상에서와 같이 도면과 명세서에서 예시적인 실시 예들이 개시되었다. 본 명세서에서 특정한 용어를 사용하여 실시 예들을 설명되었으나, 이는 단지 본 개시의 기술적 사상을 설명하기 위한 목적에서 사용된 것이지 의미 한정이나 특허청구범위에 기재된 본 개시의 범위를 제한하기 위하여 사용된 것은 아니다. 그러므로 본 기술분야의 통상의 지식을 가진 자라면 이로부터 다양한 변형 및 균등한 타 실시 예가 가능하다는 점을 이해할 것이다. 따라서, 본 개시의 진정한 기술적 보호범위는 첨부된 특허청구범위의 기술적 사상에 의해 정해져야 할 것이다.

도면

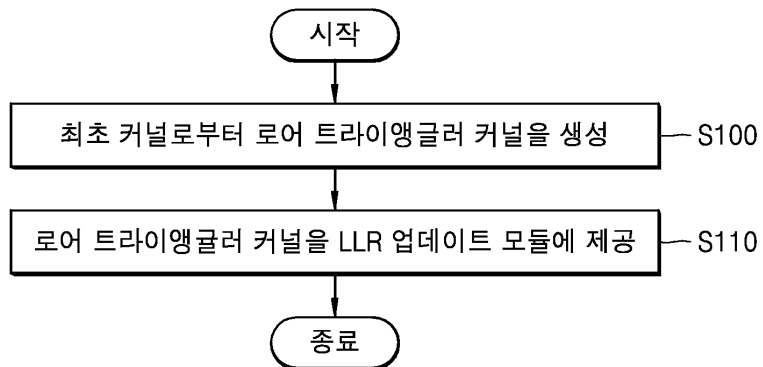
도면1



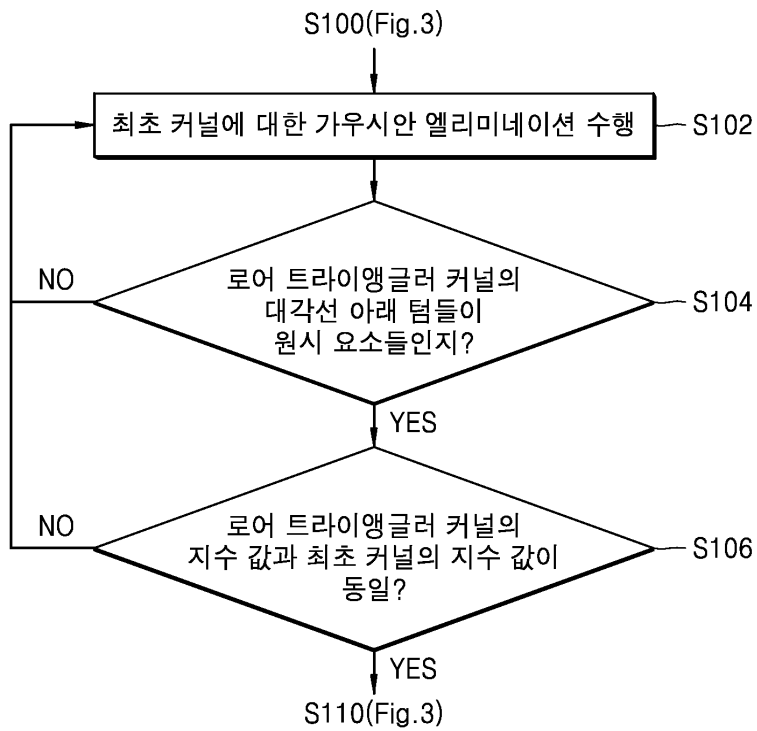
도면2



도면3



도면4



도면5a

$$\begin{array}{c}
 \text{G}_{KN} \\
 u \leftarrow \begin{bmatrix} 1 & 1 & 1 & 0 \\ \alpha & \alpha^2 & 1 & 0 \\ \alpha^2 & \alpha & 1 & 0 \\ 1 & 1 & 1 & \alpha \end{bmatrix} \leftarrow x
 \end{array}$$

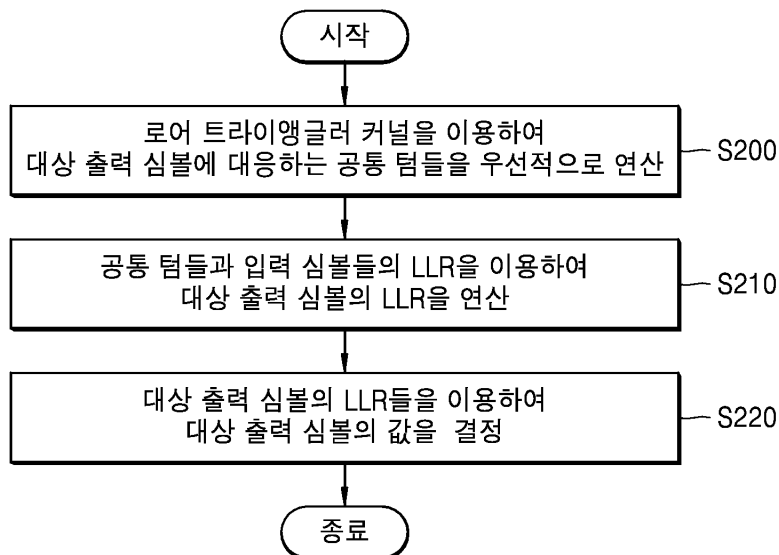
$u_0 u_1 u_2 u_3$ $x_0 x_1 x_2 x_3$
 Case_1: $u = [0 \ 0 \ 1 \ 1] \leftrightarrow x [\alpha \ \alpha^2 \ 0 \ \alpha]$
 Case_2: $u = [1 \ 0 \ 1 \ 1] \leftrightarrow x [\alpha^2 \ \alpha \ 1 \ \alpha]$
 Case_3: $u = [\alpha \ 0 \ 1 \ 1] \leftrightarrow x [0 \ 1 \ \alpha \ \alpha]$
 Case_4: $u = [\alpha^2 \ 0 \ 1 \ 1] \leftrightarrow x [1 \ 0 \ \alpha^2 \ \alpha]$

도면5b

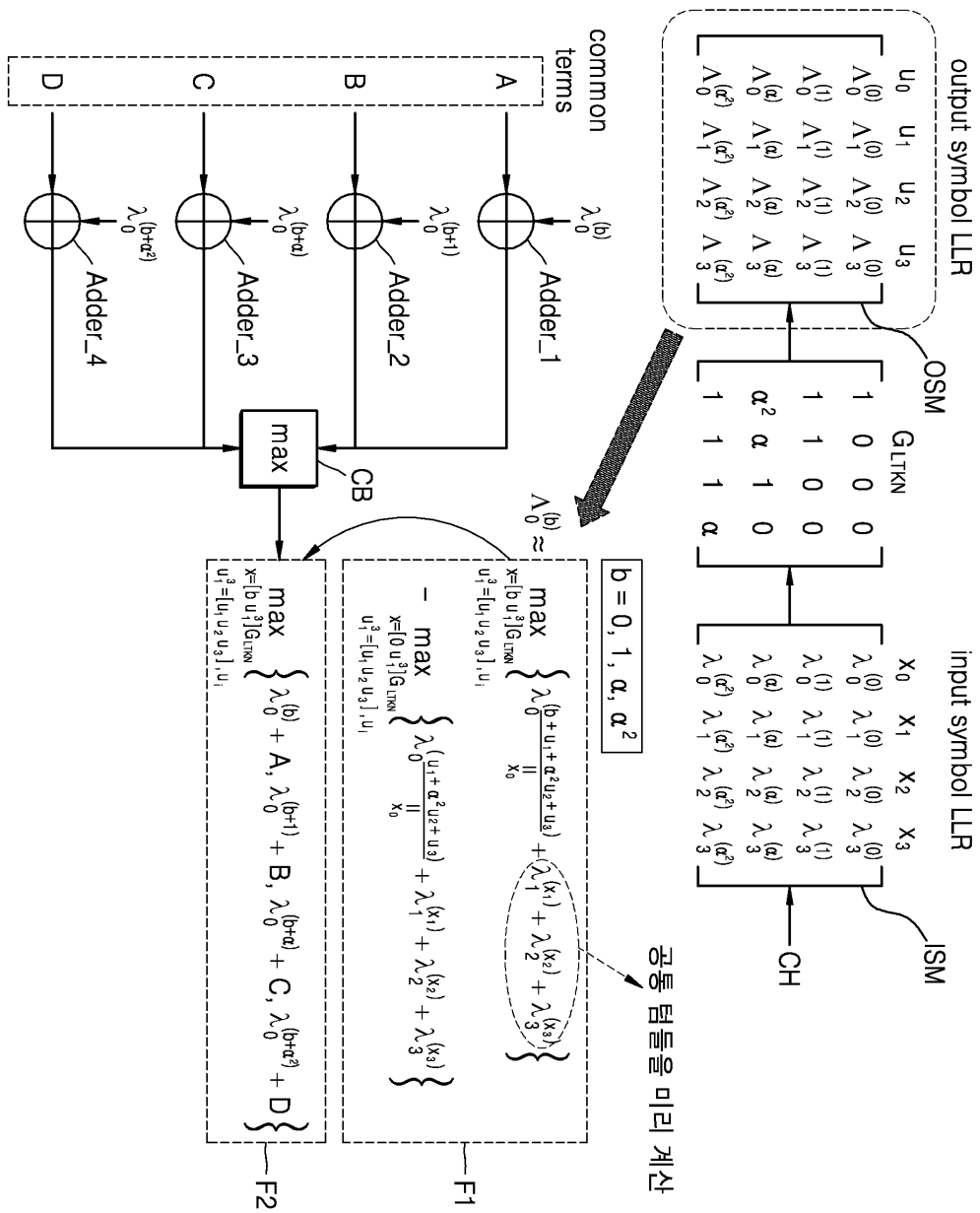
$$u \leftarrow \begin{matrix} & G_{LTKN} \\ \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ \alpha^2 & \alpha & 1 & 0 \\ 1 & 1 & 1 & \alpha \end{bmatrix} & \leftarrow x \end{matrix}$$

$u_0 \ u_1 \ u_2 \ u_3$ $x_0 \ x_1 \ x_2 \ x_3$
 Case_1: $u = [0 \ \underline{0 \ 1 \ 1}] \leftrightarrow x [\alpha \ \underline{\alpha^2 \ 0 \ \alpha}]$
 Case_2: $u = [1 \ \underline{0 \ 1 \ 1}] \leftrightarrow x [\alpha^2 \ \underline{\alpha^2 \ 0 \ \alpha}]$
 Case_3: $u = [\alpha \ \underline{0 \ 1 \ 1}] \leftrightarrow x [0 \ \underline{\alpha^2 \ 0 \ \alpha}]$
 Case_4: $u = [\alpha^2 \ \underline{0 \ 1 \ 1}] \leftrightarrow x [1 \ \underline{\alpha^2 \ 0 \ \alpha}]$

도면6



도면 7a



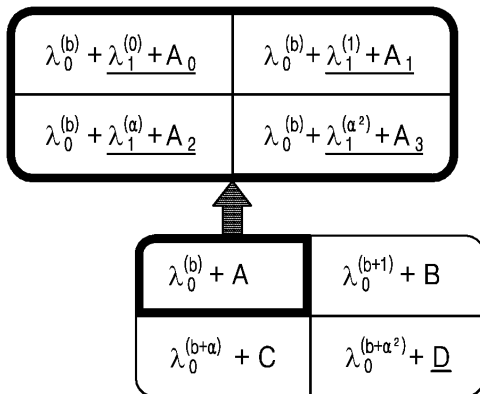
도면7b

TB1

대상 출력 심볼	입력 심볼의 LLR	공통 템
u_0	$\lambda_0^{(b)} \ (b = 0, 1, \alpha, \alpha^2)$	A
	$\lambda_0^{(b+1)}$	B
	$\lambda_0^{(b+\alpha)}$	C
	$\lambda_0^{(b+\alpha^2)}$	D
u_1	$\lambda_1^{(b)}$	A'
	$\lambda_1^{(b+1)}$	B'
	$\lambda_1^{(b+\alpha)}$	C'
	$\lambda_1^{(b+\alpha^2)}$	D'

⋮

도면8a



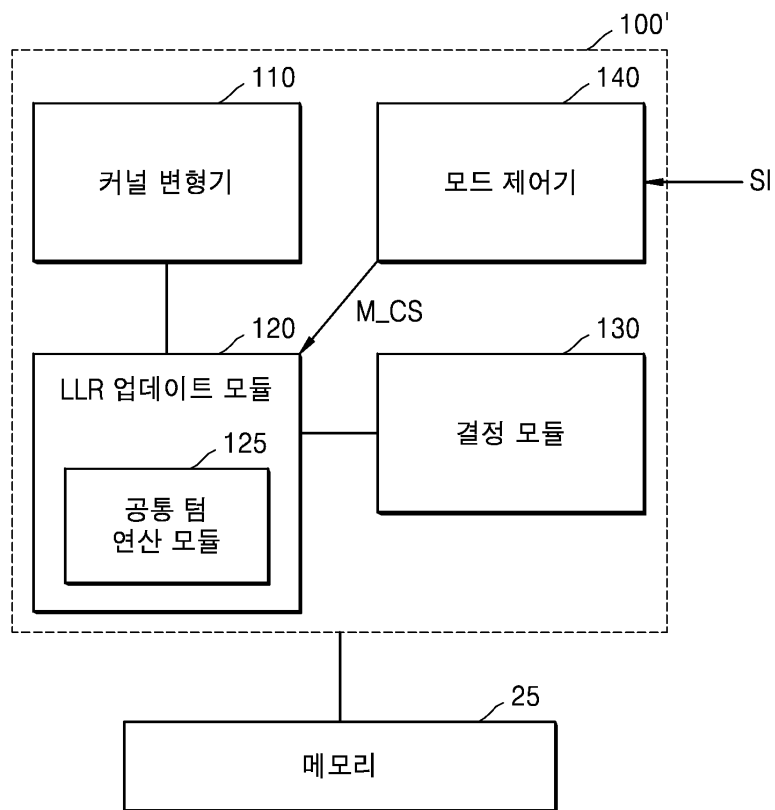
도면8b

대상 출력 심볼	입력 심볼의 제1 LLR	입력 심볼의 제2 LLR	공통 텀
u_0	$\lambda_0^{(b)} (b = 0, 1, \alpha, \alpha^2)$	$\lambda_1^{(b)}$	A_0
		$\lambda_1^{(b+1)}$	A_1
		$\lambda_1^{(b+\alpha)}$	A_2
		$\lambda_1^{(b+\alpha^2)}$	A_3
	$\lambda_0^{(b+1)}$	$\lambda_1^{(b)}$	B_0
		$\lambda_1^{(b+1)}$	B_1
		$\lambda_1^{(b+\alpha)}$	B_2
		$\lambda_1^{(b+\alpha^2)}$	B_3

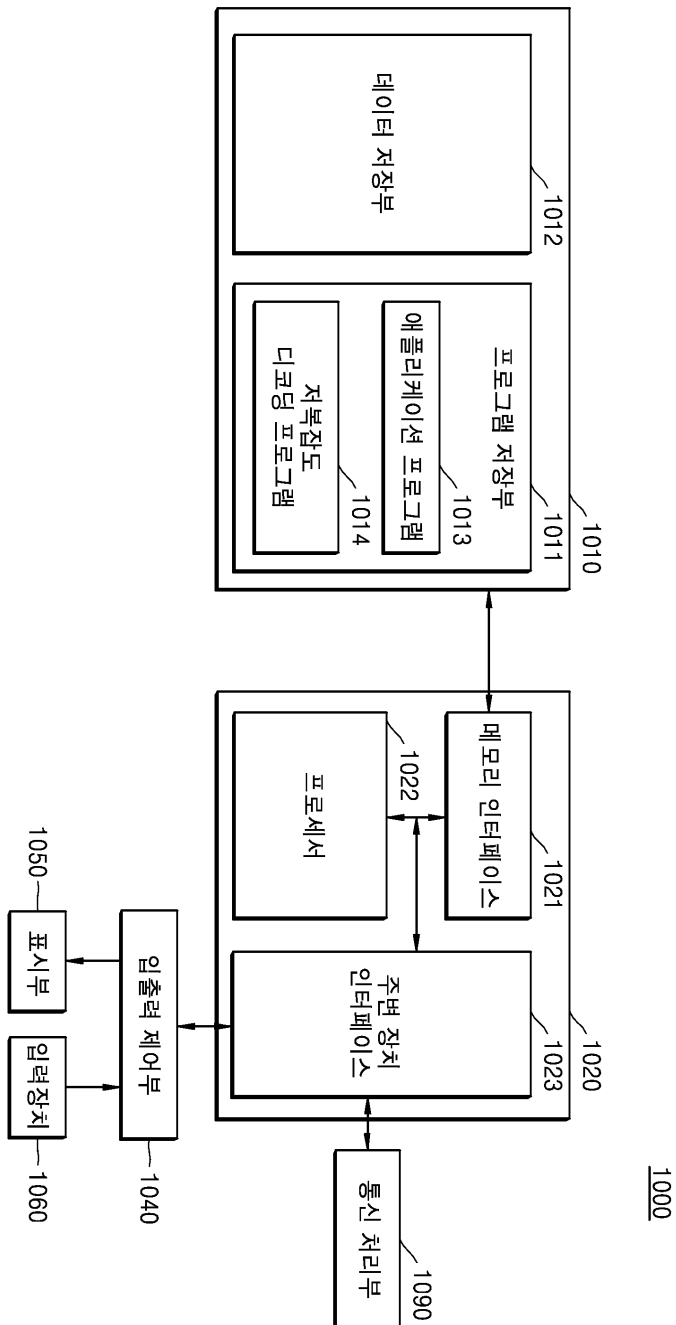
⋮

IB2

도면9



도면10



도면11

