



(19) 대한민국특허청(KR)  
(12) 등록특허공보(B1)

(45) 공고일자 2013년10월23일  
(11) 등록번호 10-1321487  
(24) 등록일자 2013년10월15일

<p>(51) 국제특허분류(Int. Cl.) G06F 11/10 (2006.01)</p> <p>(21) 출원번호 10-2010-7027398</p> <p>(22) 출원일자(국제) 2009년04월21일 심사청구일자 2013년05월31일</p> <p>(85) 번역문제출일자 2010년12월06일</p> <p>(65) 공개번호 10-2012-0011310</p> <p>(43) 공개일자 2012년02월07일</p> <p>(86) 국제출원번호 PCT/US2009/041215</p> <p>(87) 국제공개번호 WO 2010/123493 국제공개일자 2010년10월28일</p> <p>(56) 선행기술조사문헌 US20080104485 A1 US20070234184 A1</p> <p>전체 청구항 수 : 총 10 항</p>	<p>(73) 특허권자 에이저 시스템즈 엘엘시 미합중국 펜실베이니아 18109 알렌타운 노스이스트 아메리칸 파크웨이 1110</p> <p>(72) 발명자 그래프 닐스 미국 캘리포니아 950.5 밀피타스 알레그라 테라스 436 군만 키란 미국 캘리포니아 95134 산 호세 에이퍼티. 226 엘 란 빌리지 레인 371</p> <p>(74) 대리인 장훈</p>
--	---

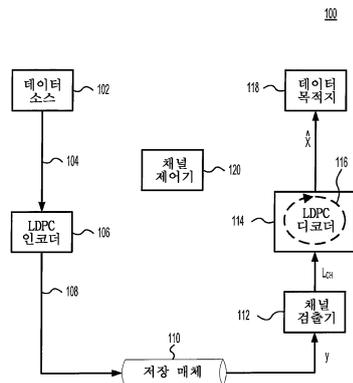
심사관 : 이정은

(54) 발명의 명칭 기입 검증에 사용한 코드들의 에러-플로어 완화

**(57) 요약**

채널 입력 (예를 들어, LDPC) 코드워드가 저장 매체에 기입될 때 실행되는, 기입-검증 방법은, (i) 채널 입력 코드워드를 기입된 코드워드와 비교하고, (ii) 어떤 에러 비트들을 식별하고, (iii) 에러-비트 인덱스들을 테이블 내 레코드에 저장한다. 나중에, 기입된 코드워드는 판독되어 디코더에 보내진다. 디코더가 근접 코드워드에 실패했을 경우, 기입-에러 복구 프로세스는 테이블을 탐색하여 에러-비트 정보를 가져온다. 이들 인덱스들에서 코드워드 비트들은 조절되고, 수정된 코드워드는 추가의 처리가 제공된다.

**대표도 - 도1**



## 특허청구의 범위

### 청구항 1

원 인코딩된 코드워드(original encoded codeword)를 저장 매체에 저장하는, 기계로 구현되는 방법에 있어서,

- (a) 상기 저장 매체에 상기 원 인코딩된 코드워드를 기입(written) 인코딩된 코드워드로 기입하는 단계;
- (b) 상기 저장 매체로부터 상기 기입 인코딩된 코드워드를 판독함으로써 채널 출력 코드워드를 생성하는 단계;
- (c) 상기 기입 인코딩된 코드워드 내 제 1 세트의 하나 이상의 에러 비트들을 식별하기 위해서 상기 원 인코딩된 코드워드를 상기 채널 출력 코드워드에 기초하는 도출된 코드워드와 비교하는 단계;
- (d) 상기 제 1 세트의 하나 이상의 에러 비트들을 선택함으로써 상기 기입 인코딩된 코드워드 내 제 2 세트의 하나 이상의 에러 비트들을 생성하는 단계; 및
- (e) 상기 제 2 세트의 상기 하나 이상의 에러 비트들에 대응하는 에러-비트 정보를 상기 저장 매체에 기입하는 단계를 포함하는, 원 인코딩된 코드워드를 저장매체에 저장하는, 기계로 구현되는 방법.

### 청구항 2

제 1 항에 있어서,

상기 원 인코딩된 코드워드는 저밀도 패리티 체크(low-density parity-check) 코드워드인, 원 인코딩된 코드워드를 저장매체에 저장하는, 기계로 구현되는 방법.

### 청구항 3

제 1 항에 있어서,

단계 (c)는 상기 도출된 코드워드를 생성하기 위해 상기 채널 출력 코드워드에 디코딩을 수행하는 단계를 추가로 포함하는, 원 인코딩된 코드워드를 저장매체에 저장하는, 기계로 구현되는 방법.

### 청구항 4

제 1 항에 있어서,

(f) 상기 저장 매체로부터 상기 기입 인코딩된 코드워드를 판독함으로써 디코더 입력 코드워드를 생성하는 단계;

(g) 디코딩된 코드워드를 생성하기 위해 상기 디코더 입력 코드워드에 디코딩을 수행하는 단계; 및

(h) 상기 디코딩된 코드워드가 디코딩된 정확한 코드워드(decoded correct codeword)가 아니라면:

(h1) 상기 저장 매체로부터 상기 에러-비트 정보를 판독하는 단계;

(h2) 상기 에러-비트 정보에 기초하여 수정된 코드워드를 생성하는 단계; 및

(h3) 상기 수정된 코드워드에 (i) 디코딩, (ii) 신드롬 체크(syndrome check), 및 (iii) 주기적 리던던시 체크(CRC) 중 적어도 하나를 수행하는 단계를 추가로 포함하는, 원 인코딩된 코드워드를 저장매체에 저장하는, 기계로 구현되는 방법.

### 청구항 5

제 1 항에 있어서,

단계 (d)는:

(d1) 상기 제 1 세트의 하나 이상의 에러 비트들을 선택하는 단계;

(d2) 하나 이상의 선택된 에러 비트들에 기초하여 수정된 코드워드를 생성하는 단계;

(d3) 후보 디코딩된 코드워드(candidate decoded codeword)를 생성하기 위해 상기 수정된 코드워드에 디코딩을

수행하는 단계;

(d4) 상기 후보 디코딩된 코드워드가 디코딩된 정확한 코드워드인지를 결정하는 단계;

(d5) 상기 후보 디코딩된 코드워드가 상기 디코딩된 정확한 코드워드이면, 상기 하나 이상의 선택된 에러 비트들에 기초하여 상기 제 2 세트를 생성하는 단계; 및

(d6) 상기 후보 디코딩된 코드워드가 상기 디코딩된 정확한 코드워드가 아니라면, 상기 제 1 세트의 또 다른 하나 이상의 선택된 에러 비트들에 대해 단계들 (d1) 내지 (d4)를 반복하는 단계를 포함하는, 원 인코딩된 코드워드를 저장매체에 저장하는, 기계로 구현되는 방법.

**청구항 6**

저장 매체에 저장된 기입 인코딩된 코드워드에 대해 디코딩된 코드워드를 생성하는, 기계로 구현되는 방법으로서, 상기 기입 인코딩된 코드워드는 하나 이상의 에러 비트들을 갖는, 상기 방법에 있어서:

(a) 상기 저장 매체로부터 상기 기입 인코딩된 코드워드를 판독함으로써 채널 출력 코드워드를 생성하는 단계;

(b) 상기 채널 출력 코드워드에 기초하여 도출된 코드워드를 생성하는 단계;

(c) 상기 저장 매체로부터 에러-비트 정보를 판독하는 단계로서, 상기 에러-비트 정보는 상기 기입 인코딩된 코드워드 내 하나 이상의 에러 비트들에 대응하는, 상기 판독 단계;

(d) 수정된 코드워드를 생성하기 위해 상기 에러-비트 정보에 기초하여 상기 도출된 코드워드의 하나 이상의 비트들에 플립핑(flipping)과 소거(erasing) 중 적어도 하나를 수행하는 단계; 및

(e) 상기 디코딩된 코드워드를 생성하기 위해 상기 수정된 코드워드에 (i) 디코딩, (ii) 신드롬 체크(syndrome check), 및 (iii) 주기적 리던던시 체크(CRC) 중 적어도 하나를 수행하는 단계를 포함하는, 디코딩된 코드워드를 생성하는, 기계로 구현되는 방법.

**청구항 7**

제 6 항에 있어서,

상기 기입 인코딩된 코드워드는 저밀도 패리티 체크 코드워드인, 디코딩된 코드워드를 생성하는, 기계로 구현되는 방법.

**청구항 8**

제 6 항에 있어서,

단계 (b)는:

(b1) 상기 도출된 코드워드를 생성하기 위해 상기 채널 출력 코드워드에 디코딩을 수행하는 단계; 및

(b2) 상기 도출된 코드워드가 디코딩된 정확한 코드워드가 아닌 것으로 결정하는 단계를 포함하는, 디코딩된 코드워드를 생성하는, 기계로 구현되는 방법.

**청구항 9**

제 6 항에 있어서,

단계 (e)는:

(e1) 상기 에러-비트 정보가 상기 기입 인코딩된 코드워드의 모든 에러 비트들에 대응하는지의 여부를 결정하는 단계;

(e2) 상기 에러-비트 정보가 상기 기입 인코딩된 코드워드의 모든 에러 비트들에 대응한다면, 신드롬 체크(syndrome check) 및 주기적 리던던시 체크(CRC) 중 하나 이상을 상기 수정된 코드워드에 수행하는 단계; 및

(e3) 상기 에러-비트 정보가 상기 기입 인코딩된 코드워드의 모든 에러 비트들에 대응하지 않는다면, 상기 수정된 코드워드에 디코딩을 수행하는 단계를 포함하는, 디코딩된 코드워드를 생성하는, 기계로 구현되는 방법.

**청구항 10**

제 6 항에 있어서,

상기 에러-비트 정보는:

상기 기입 인코딩된 코드워드에 대한 고유 식별자;

상기 기입 인코딩된 코드워드의 각각의 에러 비트에 대한 인덱스 값; 및

상기 에러-비트 정보가 상기 기입 인코딩된 코드워드의 모든 에러 비트들에 대응하는지의 표시를 포함하는, 디코딩된 코드워드를 생성하는, 기계로 구현되는 방법.

**청구항 11**

삭제

**청구항 12**

삭제

**청구항 13**

삭제

**청구항 14**

삭제

**청구항 15**

삭제

**청구항 16**

삭제

**명세서**

**기술분야**

[0001] 관련출원들에 대한 상호 참조

[0002] 이 출원의 요지는 (1) 12/12/08에 출원된 PCT 출원번호 PCT/US08/86523의 요지(attorney docket no. 08-0241), (2) 12/12/08에 출원된 PCT 출원번호 PCT/US08/86537의 요지(attorney docket no. 08-1293), (3) 3/10/09에 출원된 미국출원번호 12/401,116 의 요지(attorney docket no. 08-0248), (4) 04/02/09에 출원된 PCT 출원번호 PCT/US09/39279의 요지(attorney docket no. 08-1057), (5) 04/08/09에 출원된 미국출원번호 12/420,535의 요지(attorney docket no. 08-0242), (6) 05/01/2008에 출원된 미국특허출원번호 12/113,729의 요지, (7) 05/01/2008에 출원된 미국특허출원번호 12/113,755의 요지, (8) 04/08/09에 출원된 PCT 출원번호 PCT/US09/39918의 요지(attorney docket no. 08-0243)에 관계된 것으로, 이들 모두의 교시된 바들은 전체를 참조로 여기에 포함시킨다.

[0003] 본 발명은 디지털 신호 처리에 관한 것으로, 특히 저밀도 패리티 체크(low-density parity check; LDPC) 코딩과 같은 데이터-인코딩 방법들에 관한 것이다.

**배경 기술**

[0004] 통신은 통신 채널을 통해 송신기에서 수신기로의 정보의 송신이다. 실세계에서, 통신 채널은 잡음성 채널이어서, 송신기로부터 송신된 정보의 왜곡된 버전이 수신기에 제공된다. 저장 디바이스(예를 들어, 하드디스크(HD) 드라이브, 플래시 드라이브)는 하나의 이러한 잡음성 채널이며, 송신기로부터 정보를 받아들이며, 이 정보를 저장하고 이 정보의 다소 왜곡된 버전을 수신기에 제공한다.

**발명의 내용**

**해결하려는 과제**

[0005] 저장 디바이스와 같은 통신 채널에 의해 도입되는 왜곡은 채널 에러, 즉 채널 입력 신호가 0이었을 때 또는 1이었을 때 채널 출력 신호를 1로서 또는 0으로서 수신기가 해석하는 채널 에러를 일으킬 만큼 클 수도 있다. 채널 에러들은 스루풋을 감소시키며 이에 따라 바람직하지 못하다. 따라서, 채널 에러들을 검출 및/또는 정정하는 도구들에 대한 지속적인 필요성이 있다. 저밀도 패리티 체크(LDPC) 코딩은 채널 에러들의 검출 및 정정을 위한 하나의 방법이다.

[0006] LDPC 코드들은 저 신호 대 잡음비 (SNR) 애플리케이션들에 대해 매우 낮은 비트 에러율들(BER)을 달성할 수 있는 공지된 근-샤논-리미트 코드들(near-Shannon-limit codes) 중 하나이다. LDPC 디코딩은 고 SNR들에서, 병행화(parallelization), 저 구현 복잡성, 낮은 디코딩 레이턴시 또는 덜 심한 에러-플로어들(less-severe error floors)라는 잠재성에 의해 두드러진다. LDPC 코드들은 사실상 모든 차세대 통신 표준들용으로 간주된다.

**과제의 해결 수단**

[0007] 일 실시예에서, 본 발명은 원 인코딩된 코드워드(original encoded codeword)를 저장 매체에 저장하는, 기계로 구현되는 방법이다. 원 인코딩된 코드워드는 상기 저장 매체에 기입 인코딩된 코드워드로서 기입된다. 채널 출력 코드워드는 상기 기입 인코딩된 코드워드를 상기 저장 매체로부터 판독함으로써 생성된다. 원 인코딩된 코드워드는 상기 기입 인코딩된 코드워드 내 제 1 세트의 하나 이상의 에러 비트들을 식별하기 위해서 상기 채널 출력 코드워드에 기초하여 도출된 코드워드와 비교된다. 상기 제 1 세트에 하나 이상의 에러 비트들을 선택함으로써 상기 기입 인코딩된 코드워드 내 제 2 세트의 하나 이상의 에러 비트들이 생성된다. 제 2 세트에 하나 이상의 에러 비트들에 대응하는 에러-비트 정보가 상기 저장 매체에 기입된다.

[0008] 또 다른 실시예에서, 저장 매체에 저장된 기입 인코딩된 코드워드에 대해 디코딩된 코드워드를 생성하는, 기계로 구현되는 방법이며, 상기 기입 인코딩된 코드워드는 하나 이상의 에러 비트들을 갖는다. 상기 저장 매체로부터 기입 인코딩된 코드워드를 판독함으로써 채널 출력 코드워드가 생성된다. 도출된 코드워드는 채널 출력 코드워드에 기초하여 생성된다. 상기 저장 매체로부터 에러-비트 정보가 판독되고, 에러-비트 정보는 기입 인코딩된 코드워드 내 하나 이상의 에러 비트들에 대응한다. 수정된 코드워드를 생성하기 위해 에러-비트 정보가 사용된다. 다른 프로세싱이, 수정된 코드워드에 수행되어 상기 디코딩된 코드워드를 생성한다.

**발명의 효과**

[0009] 본 발명은 채널 에러들의 검출 및 정정을 위한 저밀도 패리티 체크(LDPC) 코딩 방법을 제공한다.

**도면의 간단한 설명**

- [0010] 도 1은 LDPC 코딩을 이용하는 통신 시스템(100)의 블록도.
- 도 2(a)는 LDPC H 매트릭스(200)를 도시한 도면이며, 도 2(b)는 H 매트릭스(200)의 태너 그래프를 도시한 도면.
- 도 3은 도 A의 디코더 AC에 의해 사용되는 LDPC 디코딩 방법(300)의 흐름도.
- 도 4는 본 발명의 일 실시예에 따라 채널 제어기(120)에 의해 제어되는 도 1의 통신 시스템(100)에 의해 구현되는 LDPC 인코딩/디코딩 프로세스(400)의 흐름도.
- 도 5는 에러-비트 테이블의 일 실시예의 다이어그램을 도시한 도면.
- 도 6은 본 발명의 일 실시예에 따라, 도 4의 단계(410), 즉 기입-에러 검증 프로세스의 흐름도.
- 도 7은 도 6의 단계(612)의 흐름도.
- 도 8은 본 발명의 일 실시예에 따라 도 4의 기입-에러 복구 프로세스(418)의 흐름도.

**발명을 실시하기 위한 구체적인 내용**

[0011] 본 발명의 다른 양태들, 특징들, 및 잇점들은 다음의 상세한 설명, 첨부한 청구항들, 유사 또는 동일 구성요소들에 동일 참조부호를 사용한 첨부한 도면들로부터 더 완전히 명백하게 될 것이다.

[0012] 도 1은 LDPC 코딩을 이용하는 통신 시스템(100)의 블록도이다. 데이터 소스(102)는 원 정보 워드(104)로서 알려진 한 세트의 비트들을 생성한다. LDPC 인코더(106)는 원 정보 워드(104)를 인코딩하며 원 인코딩된 코드워드

(108)를 생성한다. LDPC 인코딩은 이하 더 상세히 논의된다. 원 인코딩된 코드워드(108)(채널 입력 코드워드로서도 또한 알려진)은 저장 매체(110)(예를 들어, 플래시 드라이브, 하드-드라이브 플래터, 등)에, 기입 인코딩된 코드워드로서 기입된다.

[0013] 나중에, 저장 매체(110)는 기입 인코딩된 코드워드를 판독하고 한 세트의 값들( $y$ )(즉, 채널 출력 코드워드)을 채널 검출기(112)에 출력한다. 채널 출력 코드워드 및 채널 출력 코드워드로부터 도출된 임의의 코드워드는 도출된 코드워드들로서 알려진다. 채널 검출기(112)는 수신된 값들( $y$ )을 한 세트의 LLR(log-likelihood ratio) 값들( $L_{ch}$ )로 변환한다. LLR 값은 ( $i$ ) 대응하는 값( $y$ )에 의해 표시된 1비트 경-결정 값(hard-decision value)에 관한 디코더의 최선의 추측을 나타내는 부호 비트, 및 ( $ii$ ) 경-판정에서 디코더의 신뢰도를 나타내는 하나 이상의 크기 비트들을 포함한다. 예를 들어, 채널 검출기(112)는 각각의 LLR 값( $L_{ch}$ )을 5-비트 값으로서 출력할 수도 있는데, 최상위 비트는 경-판정을 나타내는 부호 비트이며 4개의 크기 비트들의 값은 경-판정의 신뢰도를 나타낸다. 이에 따라, 하나의 가능한 LLR 수법에서, 이진수 00000의 LLR 값은 최소 신뢰도를 가진 0의 경-판정을 나타내고, 이진수 01111의 LLR 값은 최대 신뢰도를 가진 0의 경-판정을 나타내고, 이진수 10001의 LLR 값은 최소 신뢰도를 가진 1의 경-판정을 나타내고, 이진수 11111의 LLR 값은 최대 신뢰도를 가진 1의 경-판정을 나타낼 것이며, 이진수 10000은 사용되지 않는다.

[0014] 채널 검출기(112)는  $L_{ch}$  값들을 LDPC 디코더(114)에 보내며, 여기서 이들은 디코더 입력 코드워드가 된다. LDPC 디코더(114)는 한 세트의  $L_{ch}$  값들에 하나 이상의 디코딩 반복들(116)("국부적 반복들")을 수행하여 디코딩된 코드워드( $\hat{x}$ )를 생성한다. LDPC 디코더(AC)는 ( $i$ ) LDPC 디코더(114)가 디코딩된 정확한 코드워드(DCCW)에 도달할 때, 즉  $\hat{x}$ 가 채널 입력 코드워드(108)와 동일할 때, 또는 ( $ii$ ) LDPC 디코더(AC)가 DCCW에 도달함이 없이 최대 허용가능한 회수의 국부적 반복들을 수행할 때, 즉 LDPC 디코더(114)가 실패하였을 때, 종료한다. 디코더(114)가 종료하였을 때, 이것은 디코딩된 코드워드  $\hat{x}$ 를 데이터 목적지(118)에 출력한다. LDPC 디코딩은 이하 더 상세히 기술된다.

[0015] 채널 제어기(120)는 적어도 LDPC 인코더(106), 채널 검출기(112), 및 LDPC 디코더(114)의 동작들을 제어한다. 채널 제어기는 ARM(Advanced RISC(reduced instruction-set code) Machine) 프로세서이다.

[0016] **LDPC 인코딩**

[0017] 코드워드(108)를 생성하기 위해서, LDPC 인코더(106)는 LDPC 코드에 의해 명시된 다수의 패리티 비트들을 정보워드(104)의 비트들에 첨부한다. 정보워드(104)에 비트들의 수는  $K$ 로 표기한다. 인코딩된 코드워드에 비트들은 가변 비트들로서 알려져 있으며, 이들 가변 비트들의 수는  $N$ 으로 표기한다. 이에 따라, 패리티 비트들의 수는  $N - K$ 에 의해 주어진다.

[0018] LDPC 코드워드에 각각의 패리티 비트는 특별한 LDPC 코드에 의해 명시된 바와 같은 특별한 방법으로 이 코드워드 내 하나 이상의 다른 비트들에 연관되며, 패리티 비트에 할당된 값은 LDPC 코드를 만족시키도록 설정된다. 전형적인 LDPC 코드들은 패리티 비트 및 이의 연관된 비트들이 패리티-체크 제약을 만족시킬 것들을 특정하고 있는데, 예를 들어, 비트들의 합은 우수, 즉 합 모듈로 2 = 0이다.

[0019] **LDPC 코드**

[0020] 하나의 특정한 LDPC 코드는 패리티-체크 매트릭스, 또는  $H$  매트릭스, 또는 간단히  $H$ 로서 알려진 1들 및 0들의 2차원 매트릭스로 정의된다.  $H$ 는 LDPC 인코더 및 디코더 둘 다에 선형적으로 알려진다.  $H$ 는  $N$  열들 및  $N - K$  행들, 즉, 코드워드의 매 비트마다 한 열, 및 매 패리티 비트마다 한 행을 포함한다.  $H$ 에서 각각의 1은 열의 코드워드 비트와 행의 패리티 비트 간에 연관을 나타낸다. 예를 들어,  $H$ 의 제 3 행, 제 7 열의 1은 코드워드의 제 7 비트에 연관된다는 것을 의미한다. 체크 비트의 값과 이 체크 비트에 연관된 모든 가변 비트들의 합 모듈로 2는 0이어야 한다. 전형적인 LDPC 코드들의 정의된 특징은  $H$ 가 "스파스(sparse)", 즉  $H$ 의 원소들이 비교적 몇 개의 1들을 가진 대부분 0들이라는 것이다.

[0021] 도 2(a)는 LDPC  $H$  매트릭스(200)를 도시한 것이다.  $H$  매트릭스(200)는  $N = 9$  열들 및  $N - K = 6$  행들을 포함한다. 이에 따라,  $H$  매트릭스(200)는 3-비트 정보 워드를 받고 6개의 패리티 비트들을 첨부하고, 9비트 코드워드를 출력하는 LDPC 코드를 정의한다. 저장 매체가 하드디스크 드라이브 또는 플래시 드라이브인 일 구현에서, 각각의 정보 워드는 길이가 4,096 비트이고, 각각의 코드워드는 길이가 4,552 비트이다. 다른 구현들은 다른 비트

길이들을 갖는 정보 워드들 및/또는 코드워드들을 가질 수 있다.

[0022] LDPC 디코딩: 신뢰 전파

[0023] 도 3은 도 1의 디코더(114)에 의해 사용되는 LDPC 디코딩 방법(300)의 흐름도이다. 디코딩 방법(300)의 핵심은 신뢰 전파라고 하는 반복성의 2-국면 메시지-전달 알고리즘이다. 신뢰 전파를 태너 그래프(Tanner graph)를 사용하여 설명될 수 있다.

[0024] 도 2(b)는  $H$  매트릭스(200)에 대한 태너 그래프이다. 일반적으로, 태너 그래프는  $H$ 에 열들의 수와 동일한(및 이에 따라 가변 비트들의 수( $N$ )과 같다) 다수의 비트 노드들(가변 노드들이라고도 함)( $n$ ), 2)  $H$ 에 행들의 수와 동일한(및 이에 따라 패리티 비트들의 수와 동일한) 다수의 체크 노드들( $m$ ), 3) 각각이 단일 비트 노드( $n_i$ )를 단일 체크 노드( $m_j$ )에 연결하는 에지들(202), 4) 각각의 비트 노드( $n_i$ )에 대해, 원래의  $L_{ch}$  값, 및 5) 각각의 비트 노드( $n_i$ )에 대해, 계산된 경-판정 출력 값  $\hat{x}_n$  을 포함한다. 도 2(b)의 태너 그래프는 9 비트 노드들( $n_0$  내지  $n_8$ ), 6 체크 노드들( $m_0$  내지  $m_5$ ), 비트 노드들을 체크 노드들에 연결하는 18 에지들(202), 9개의  $L_{ch}$  값들, 및 9 개의  $\hat{x}_n$  값들을 포함한다.

[0025] 태너 그래프에서 에지들은 비트 노드들( $n$ )과 체크 노드들( $m$ ) 간에 관계를 나타내며, 에지들은  $H$ 에서 1들을 나타낸다. 예를 들어, 도 2(b)에서, 에지(202)는 도 2(a)에서  $H$  매트릭스(200)의 제 1 열, 제 4 행에 1이 있기 때문에, 제 1 비트 노드( $n_0$ )를 제 4 체크 노드( $m_3$ )에 연결한다.

[0026] 태너 그래프는 이분 그래프이다. 즉, 에지는 비트 노드를 체크 노드에만 연결할 수 있고, 비트 노드를 다른 비트 노드에, 또는 체크 노드를 다른 체크 노드에 연결할 수 없다. 에지들에 의해 하나의 특정한 체크 노드( $m$ )에 연결된 한 세트의 모든 비트 노드들( $n$ )을  $N(m)$ 으로 표기한다. 에지들에 의해 하나의 특정한 비트 노드( $n$ )에 연결된 한 세트의 모든 체크 노드들( $m$ )을  $M(n)$ 으로 표기한다. 하나의 특정한 (비트 또는 체크) 노드의 인덱스는 그래프 내 이의 서수 열이다.

[0027] 도 3으로 돌아가서, 처리는 단계(302)에서 시작하며 단계(304)의 디코더 초기화로 진행한다. 디코더 초기화(304)는 각각의 비트 노드( $n$ )에 연결된 모든 에지들(예를 들어, 도 2(b)의 에지들(202))을, 비트 노드( $n$ )에 연관된 대응하는  $L_{ch}$  값 으로 설정하고, 비트 노드( $n$ )의  $\hat{x}_n$  값을, 비트 노드( $n$ )의  $L_{ch}$ 의 경-판정 값 (즉, MSB) 으로 설정하는 것을 포함한다. 이에 따라, 예를 들어, 도 2(b)에서, 비트 노드( $n_0$ )에 연관된  $L_{ch}$  값이 십진수 값 +5이라면, 단계(304)에서, 비트 노드( $n_0$ )를 체크 노드들( $m_0, m_3$ )에 연결하는 2개의 에지들(202)은 +5로 설정되고, 비트 노드( $n_0$ )의  $\hat{x}_n$  값은 1로 설정된다. 이 단계의 제 1 부분을 나타내는 대안적 방법은 비트 노드( $n_0$ )가 세트  $M(n_0)$ 에서 각각의 체크 노드( $m$ )에 +5의 메시지를 보내는 것이다. 비트 노드( $n$ )에서 체크 노드( $m$ )로 보내진 메시지는 비트-노드 또는  $Q$  메시지라 하고,  $Q_{nm}$ 으로 표기한다.

[0028] 단계(304)는  $N \hat{x}_n$  값들을 포함하는 후보 디코딩된 코드워드 벡터  $\hat{x}$  를 신드롬 체크 단계(306)에 보낸다. 신드롬 체크 단계(306)는 다음 식(1)을 사용하여 신드롬 벡터( $z$ )를 계산한다:

[0029] 
$$z = \hat{x}H^T \tag{1}$$

[0030]  $H^T$ 는  $H$ 의 전치 매트릭스이다. 신드롬 벡터( $z$ )가 0 벡터이면, 벡터( $\hat{x}$ )는  $H$ 에 의해 정의된 모든 패리티-체크 제약들을 만족시킨 것으로, 즉  $\hat{x}$  는 유효 디코딩된 코드워드이다. 이 경우, 처리는 CRC(cyclic-redundancy check) 체크(318)로 진행한다.

[0031] 대신에, 신드롬 벡터( $z$ )가 0 벡터가 아니면, 벡터( $\hat{x}$ )는 하나 이상의 패리티-체크 제약들을 실패한다. 신드롬 벡터( $z$ )에 각각의 비-제로 원소는 만족되지 않은 체크 노드(USC)라고도 하는 실패된 패리티-체크 제약을 나타낸다. 신드롬 벡터( $z$ )에서 비-제로 원소들의 수는 벡터( $\hat{x}$ )에서 USC들의 수( $b$ )이다. 또한, 신드롬 벡터( $z$ )의 비-

제로 원소들의 인덱스들은 벡터( $\hat{x}$ )에서 USC들의 인덱스들이다.

[0032] 벡터( $\hat{x}$ )가 신드롬 체크(306)를 실패하면, 처리는 하나 이상의 디코딩 반복들(308)("국부적 반복들"이라 함) 중 첫번째로 계속된다. 디코딩 반복(308)은 3개의 단계들로서, 1) 신뢰 전과 체크-노드 업데이트 단계(310), 2) 신뢰 전과 비트-노드 업데이트 단계(312), 및 3) 단계(306)와 동일한 신드롬 체크 단계(314)를 포함한다.

[0033] 신뢰 전과 체크-노드 업데이트 단계(310)에서, 각각의 체크 노드(m)는 다음 식들(2), (3), (4)에 따라,  $R_{mn}$ 으로 표기한 하나 이상의 체크-노드 또는 R 메시지들을 계산하기 위해서 세트  $N(m)$  내 모든 비트 노드들(n)로부터 수신된  $Q_{nm}$  메시지들을 사용한다.

[0034] 
$$R_{mn}^{(i)} = \delta_{mn}^{(i)} \max(\kappa_{mn}^{(i)} - \beta, 0) \quad (2)$$

[0035] 
$$\kappa_{mn}^{(i)} = |R_{mn}^{(i)}| = \min_{n' \in N(m) \setminus n} |Q_{n'm}^{(i-1)}| \quad (3)$$

[0036] 
$$\delta_{mn}^{(i)} = \left( \prod_{n' \in N(m) \setminus n} \text{sgn}(Q_{n'm}^{(i-1)}) \right) \quad (4)$$

[0037] 여기에서 i는 디코딩 반복이며,  $N(m) \setminus n$ 은 비트 노드(n)을 제외한 세트  $N(m)$ 이며, 함수 sgn는 그의 피연산자(operand)에 부호를 리턴하며,  $\beta$ 는 양의 상수이며, 이 값은 코드 파라미터들에 따른다. 각각의 체크 노드(m)는 계산된  $R_{mn}$  메시지들을 동일 에지들을 따라 세트  $N(m)$  내 모든 비트 노드들(n)에 보낸다.

[0038] 다음에, 신뢰 전과 비트-노드 업데이트 단계(312)에서, 각각의 비트 노드(n)는 다음 식(5)에 따라 하나 이상의  $Q_{nm}$  메시지들을 계산한다.

[0039] 
$$Q_{nm}^{(i)} = L_n^{(0)} + \sum_{m' \in M(n) \setminus m} R_{m'n}^{(i)} \quad (5)$$

[0040] 여기에서  $L_n^{(0)}$ 은 비트 노드(n)에 대한 원래의  $L_{ch}$  값이며,  $M(n) \setminus m$ 은 체크 노드(m)를 제외한 세트  $M(n)$ 이다. 각각의 비트 노드(n)는 계산된  $Q_{nm}$  메시지들을 세트  $M(n)$ 의 모든 체크 노드들(m)에 보낸다.

[0041] 또한, 비트-노드 업데이트 단계(312) 동안에, 각각의 비트 노드(n)는 그의  $\hat{x}_n$ 값을 다음 식들 (6), (7)에 따라 업데이트한다.

[0042] 
$$E_n^{(i)} = \sum_{m' \in M(n)} R_{m'n}^{(i)} \quad (6)$$

[0043] 
$$P_n = L_n^{(0)} + E_n^{(i)} \quad (7)$$

[0044]  $P_n \geq 0$ 이라면,  $\hat{x}_n = 0$ 이며,  $P_n < 0$ 이면,  $\hat{x}_n = 1$ 이다. 식(6)에 의해 생성된 값들은 외래 또는 E 값들이라고도 하며,  $E_{LDPC}$ 로 표기한다. 식(7)에 의해 생성된 값들을 P값들이라 한다. 식들(2) 내지 (7)로 표현되는 특정의 신뢰 전과 알고리즘은 최소-합 알고리즘으로서 알려져 있다.  $\hat{x}_n$  값들은 각각의 디코딩 반복(308) 동안 업데이트되며 마지막에 디코딩 프로세스(300)에 의해 출력됨에 유의한다. 원 LLR 값들  $L_{ch}$ 은 디코딩 프로세스(300) 동안 변경되지 않은 채로 있다.

[0045] 비트-노드 업데이트 단계(312)는 디코더의 현  $\hat{x}_n$  값들로부터 구축된 벡터( $\hat{x}$ )를 신드롬 체크 단계(314)에 보낸다. 단계(314)의 신드롬 체크는 위에 논의된 단계(306)의 신드롬 체크와 동일하다. 벡터( $\hat{x}$ )가 신드롬 체크

(314)를 통과하면, 벡터( $\hat{x}$ )는 CRC 단계(318)에 보내진다.

- [0046] **LDPC 디코딩: 주기적 리턴던시 체크 및 오만족 체크 노드들**
- [0047] 신드롬 체크(306 또는 314)를 통과한 것은 벡터( $\hat{x}$ )가 유효 디코딩된 코드워드임을 의미하나, 반드시 디코딩된 정확한 코드워드(DCCW)인 것은 아니다. LDPC 디코더가 DCCW가 아닌 유효 디코딩된 코드워드를 생성하는 것이 가능하다. 이 경우, 벡터( $\hat{x}$ )엔 USC들이 없으나, 오만족 체크 노드들(MSC)이 있다. 오만족 체크 노드는 짝수 개의 에러 비트-노드들에 연관되는 체크 노드이다.
- [0048] 이에 따라, 유효 벡터( $\hat{x}$ )가 DCCW임을 보증하기 위해서, 프로세스(300)는 벡터( $\hat{x}$ )를 주기적 리턴던시 체크(CRC)(318)에 보낸다. CRC 체크는 송신 또는 저장 동안 데이터의 변경(alteration)을 검출할 수 있는 체크섬 동작(checksum operation)이다. 구체적으로, 인코더는 보내질 코드워드의 제 1 CRC 체크섬을 계산하고, 제 1 CRC 체크섬 및 코드워드 둘 모두를 디코더에 보낸다. 디코더는 제 1 CRC 체크섬 및 코드워드 둘 모두를 수신하였을 때, 수신된 코드워드를 사용하여 제 2 CRC 체크섬을 계산하고, 이를 제 1 CRC 체크섬과 비교한다. 두 CRC 체크섬들이 일치하지 않는다면, 수신된 코드워드에 에러들이 있을 높은 가능성이 있다.
- [0049] 벡터( $\hat{x}$ )가 CRC 체크를 통과한다면, 벡터( $\hat{x}$ )는 DCCW이며, 프로세스(300)는 전역 변수 DCCW를 트루(true)로 설정하고, 벡터( $\hat{x}$ )를 출력하며, 단계(320)에서 종료한다. 그렇지 않다면, 벡터( $\hat{x}$ )는 DCCW가 아니며, 프로세스(300)는 전역 변수 DCCW를 폴스(false)로 설정하고, 벡터( $\hat{x}$ )를 출력하며, 단계(320)에서 종료한다. 전역 변수 DCCW는 DCCW가 생성되었는지를 다른 디코딩 프로세스들에 알린다.
- [0050] 단계(314)로 돌아가서, 벡터( $\hat{x}$ )가 신드롬 체크를 실패한다면, 벡터( $\hat{x}$ )에 하나 이상의 USC들이 존재한다. USC들을 해결하는 전형적인 방법은 또 다른 디코딩 반복(308)을 수행하는 것이다. 그러나, 하나의 특정한 디코딩 세션에서, 상당한 시간량 내에 결코 만족되지 않을 하나 이상의 USC들이 존재할 수도 있을 것이다(이하, 트래핑 세트들(trapping sets)의 논의를 참조). 이에 따라, LDPC 디코더들은 전형적으로 이들이 얼마나 많은 디코딩 반복들을 수행할 수 있는가로 제한된다. 최대수의 반복들에 대한 전형적인 값들은 50 내지 200의 범위이다.
- [0051] 도 3에서, 단계(316)는 특정된 최대수의 반복들에 도달되었는지 결정한다. 그렇지 않다면, 또 다른 디코딩 반복(308)이 수행된다. 대신에, 최대수의 반복들에 도달하였다면, 디코더 프로세스(300)는 실패된다. 이 경우, 프로세스(300)는 전역 변수 DCCW를 폴스로 설정하고, 벡터( $\hat{x}$ )를 출력하고, 단계(320)에서 종료한다.
- [0052] 프로세스(300)의 완전한 실행(하나 이상의 국부적 디코딩 반복들(308)을 사용한)은 디코딩 세션으로서 알려진다.
- [0053] **BER, SNR, 및 에러 플로어들**
- [0054] LDPC 디코더의 비트 에러율(BER)은 디코딩된 비트가 틀린 값을 가질 확률을 나타낸다. 이에 따라, 예를 들어,  $10^{-9}$ 의 BER을 가진 디코더는 평균으로, 매 10억의 디코딩된 비트들마다 하나의 에러 비트를 발생할 것이다. LDPC 디코딩 세션 이 DCCW 수렴에 실패가 디코더의 BER에 기여한다.
- [0055] LDPC 디코더의 BER은 디코더의 입력 신호의 신호 대 잡음비(SNR)에 의해 강하게 영향을 받는다. SNR의 함수로서 BER의 그래프는 전형적으로 2개의 서로 구별되는 영역들로서, SNR에 단위 증가가 주어졌을 때 BER이 급속히 개선되는(감소하는) 초기 "위터폴" 영역과, SNR에서 BER에 완만한 개선들만을 증가시키는 후속하는 "에러-플로어" 영역을 포함한다. 이에 따라, 에러-플로어 영역에서 현저한 BER 개선을 달성하는 것은 SNR 증가 이외의 방법들을 요구한다.
- [0056] LDPC 디코딩의 에러-플로어 특징들을 개선하는 하나의 방법은 코드워드 길이를 증가시키는 것이다. 그러나, 코드워드 길이를 증가시키는 것은 LDPC 디코딩을 위해 요구되는 메모리 및 그외 컴퓨팅 자원들도 증가시킨다. 이에 따라, 전형적으로 저장 디바이스들 상에 관독-채널 디바이스들의 경우와 같이, 이러한 자원들이 엄격히 제한된다면, 필요한 에러-플로어 개선을 얻기 위해 다른 방법들이 발견되어야 한다.
- [0057] 또 다른 부족한 자원은 처리 사이클들이다. 전형적으로, 명시된 스루풋을 달성하기 위해서, 저장 디바이스는 코

드워드를 디코딩하기 위해 고정된 수의 관독-채널 처리 사이클들의 예산을 세운다. 예산을 초과하는 방법들(예를 들어, 오프-더-플라이(off-the-fly) 방법들)은 스루풋을 감소시킨다. 더 바람직한 것은 클럭-사이클 할당 내에서 DCCW를 복구하고 이에 따라 스루풋을 감소시키지 않는 온-더-플라이(on-the-fly) 방법들이다.

- [0058] LDPC 디코더의 에러-플로어 특징들을 개선하는 또 다른 방법은 하나 이상의 후처리 방법들을 사용하는 것이다. 후처리 방법은 디코딩 세션이 허용된 최대수의 반복들 내에서 DCCW에 수렴하지 못할 때 인보크된다(invoked). 후처리 방법은 디코딩 프로세스에 연관된 하나 이상의 변수들(예를 들어,  $y$  값들,  $L_{ch}$  값들, 및/또는 디코더의 동작 파라미터들)을 조절하여 디코딩을 다시 시작한다. 디코딩 자체처럼, 후처리 방법은 흔히 반복적이어서, 디코딩 프로세스로의 입력들에 복수의 순차적 변화들을 야기한다.
- [0059] 전형적인 LDPC-디코딩 세션에서, 디코더는 첫번째 몇번의 디코딩 반복들 내에서 DCCW에 수렴한다. 대신에, 명시된 최대수의 반복들 내에서 LDPC 디코더가 DCCW에 수렴하지 못할 때, 이것은 실패된 디코더로서 알려지고, 실패된 디코더에 의해 생성된 디코딩된 코드워드는 실패된 코드워드이다.
- [0060] 실패된 코드워드들은 전형적으로 이들이 내포하는 USC 노드들의 수에 의해 구별된다. 무효한 디코딩된 코드워드(ICW)는 많은 수(예를 들어, 대략 5,000-비트 코드워드에 대해 16보다 큰)의 USC 노드들을 가진 실패된 코드워드이다. ICW들은 전형적으로 디코더가 모든 비트 에러들을 정정할 수 없을 정도로 너무 많은 비트 에러들, 즉 몇개의 정확한 값들을 내포하는 디코더 입력 코드워드로부터 비롯된다. ICW를 취급하기 위한 전형적인 후처리 방법은 입력 코드워드의 재송신을 요청하는 것이다. 재송신이 오프-더-플라이 방법이고 이에 따라 바람직하지 않을지라도, 전형적으로 ICW를 정정하기 위한 유일한 신뢰성있는 선택이다.
- [0061] 근접(near) 코드워드(NCW)는 적은 수의 USC들(예를 들어, 대략 5,000-비트 코드워드에 대해 16 또는 그 미만)을 소유하는 실패된 코드워드이다. 때때로, NCW에 USC들은 트래핑 세트로서 알려진 안전한 구성을 형성하며, 이 트래핑 세트에 대해서 추가의 디코딩 반복들은 DCCW를 생성하지 않을 것이다. 트래핑 세트들은 LDPC 디코더의 에러-플로어 특징들에 현저한 영향을 미친다.
- [0062] 트래핑 세트들은  $(a, b)$ 로 표기하고,  $b$ 는 트래핑 세트 내 USC들의 개수이며,  $a$ 는 이들 USC들에 연관된 에러 비트 노드들의 개수이다. 이에 따라,  $(8, 2)$  트래핑 세트는 2개의 USC들과 이들 2개의 USC들에 연관된 8개의 에러 비트 노드들(EBNs)을 포함한다. 대부분의 트래핑 세트들은 5 미만의 USCE들과 10 미만의 EBN들을 포함한다.
- [0063] 비트 노드의 플립핑(flipping)은 비트 노드에 연관된 하나 이상의 값들을 변경하기 위한 특정의 프로세스를 말한다. 플립핑 동안 어느 값들이 변경되는가는 LDPC 디코더의 상태에 따른다. 하나의 가능한 구현에서, LDPC 디코더가 이제 초기화되었다면, 비트 노드를 플립핑하는 것은 (i) 이 비트 노드의  $L_{ch}$  값의 경-관정 값을 반전시키는 것으로, 즉, 1은 0이 되고 0은 1이 되는 것과, (ii) 동일  $L_{ch}$  값의 크기 비트들, 즉, 신뢰도를 최대로 설정하는 것과, (iii) 이외 모든  $L_{ch}$  값들의 크기 비트들을 최대 허용할 수 있는 크기 값의 기껏해야 15%로 제한시키는 것을 포함한다.
- [0064] 예를 들어, 4-비트  $L_{ch}$  크기 값들을 가진 시스템을 가정하고, 여기서 최대 허용가능한 양의 크기는 +15이며 최대 허용가능한 음의 크기는 -16이며, 최대 허용가능한 값들의 15%는 각각 +2 및 -2이 될 것이다. 또한, 4개의 비트 노드들에 대응하는 4개의  $L_{ch}$  값들로서 +2, -11, +1, +13을 가정한다. 이 예에서, 제 1 비트 노드를 플립핑하는 것은 (i) 제 1 비트 노드의  $L_{ch}$  값의 부호를 반전시키는 것으로서, 즉 +2가 -2로 되는 것과, (ii) 제 1 비트 노드의  $L_{ch}$  값의 크기를 최대 허용가능한 값으로 설정하는 것으로서, 즉, -2는 -16이 되는 것과, (iii) 다른 3개의 비트 노드들의  $L_{ch}$  값들의 크기를 기껏해야 15%의 최대 허용가능한 값으로 제한시키는 것으로, 즉, -11, +1, 및 +13이 각각 -2, +1, 및 +2이 되는 것을 포함한다.  $L_{ch}$  크기들을 낮은 값들로 설정하는 것은 반복적 디코딩 프로세스가 DCCW에 더 신속하게 수렴할 수 있게 하며, 디코더가 실패될 기회들을 낮춘다.
- [0065] 이 동일 구현에 따라서, 디코더가 초기화 이외의 어떤 상태에 있다면, 비트 노드를 플립핑하는 것은, (i) 비트 노드의  $P$  값(위에 식(7)에 의해 정의됨)의 경-관정 값을 결정하는 것과, (ii) 이 비트 노드의  $L_{ch}$  값,  $P$  값, 및 모든 연관된  $Q_{nm}$  메시지들의 경-관정 값들을 상기  $P$  값 경-관정 값의 반대로 설정하는 것과, (iii) 이 비트 노드의  $L_{ch}$  값,  $P$  값, 및 모든 연관된  $Q_{nm}$  메시지들의 크기 비트들을 최대로 설정하는 것과, (iv) 그외 모든 다른 비트들의  $L_{ch}$ ,  $P$ ,  $Q_{nm}$  메시지 값들의 크기를 최대 허용가능한 값의 15%로 제한시키는 것을 포함한다. 단지 초기 크

기들만이 제한됨에 유의한다. 디코딩 세션이 진전됨에 따라, P 및  $Q_{\text{min}}$  메시지 값들은 업데이트되고 임의의 허용 가능한 값을 취할 수 있다. 한편,  $L_{\text{ch}}$  값들은 읽을 수만 있으며 이에 따라 디코딩 세션 기간 동안 이들의 제한된 크기 값들을 유지할 것이다.

[0066] 소거는 비트-노드 값들을 변경하는 또 다른 특정한 프로세스이다. 비트 노드를 소거하는 것은 (i) 이 비트 노드의  $L_{\text{ch}}$  값의 경-관정 값을 0으로 설정하는 것과, (ii) 이 동일  $L_{\text{ch}}$  값의 크기 비트들, 즉, 신뢰도를 0으로, 즉 무 신뢰도로 설정하는 것을 포함한다.

[0067] 트래핑 세트에 대응하는 근접 코드워드에 대한 하나 이상의 EBN들이 조절된다면(예를 들어, 플립핑 또는 소거), 결과적인 수정된 코드워드에 LDPC 디코딩을 재수행하는 것은 DCCW에 수렴시킬 수 있다. 성공적일 때, 이 프로세스를 트래핑 세트 와해(breaking)라 한다. 이에 따라, LDPC 디코더의 에러-플로어 특징들을 개선하는 또 다른 방법은 실패된 디코더의 근접 코드워드(NCW)를 취하고, NCW에서 잠재적 EBN들을 확인하고, 하나 이상의 이들 EBN들을 플립핑 또는 소거하고, 추가의 LDPC 처리를 위해 이 수정된 NCW를 제공하는 것이다.

[0068] 일부 트래핑 세트들은 단일 EBN을 플립핑 또는 소거함으로써 와해될 수 있다. 다른 트래핑 세트들에서, 단일의 EBN을 플립핑 또는 소거하는 것은 USC들의 수를 감소시킬 수 있으나, 트래핑 세트를 완전히 와해하지 못하여, 또 다른 EBN을 플립핑 또는 소거함으로써 와해될 수도 있을 제 2의 다른 트래핑 세트를 산출한다. 또 다른 트래핑 세트들은 2 이상의 EBN들을 동시에 플립핑 또는 소거함으로써만 와해될 수 있다.

[0069] 트래핑 세트들은 (i) 디코더의 동작 상태들(예를 들어, 디코더 알파벳, 디코더 알고리즘, 디코더 체크-노드 업데이트 방법), (ii) 기입 에러들, 및 (iii) 판독 에러들에 기초하여 크게 달라진다. 통신 채널이 저장 디바이스 일 때, 기입 에러는 코드워드의 비트가 저장 디바이스에 부정확하게 기입될 때, 즉, 기입 인코딩된 코드워드 비트의 값이 대응하는 채널 입력 코드워드 내 대응하는 비트의 값과 일치하지 않을 때이다. 판독 에러는 코드워드의 정확하게 기입된 비트가 저장 디바이스로부터 부정확하게 판독될 때, 즉, 채널 출력 코드워드의 비트의 값이 기입 인코딩된 코드워드 내 대응 비트의 값과 일치하지 않을 때이다. 판독 에러들로부터 복구는 예를 들어, 섹터를 복수회 재 판독하고 이어서 복수의 판독 샘플들을 평균함으로써 가능하다. 그러나, 기입 에러들로부터 복구는 섹터를 다시 판독함으로써 가능하지 않다.

[0070] 본 발명의 실시예들은 기입 에러들에 기인한 LDPC 디코더 입력 코드워드에 이들 EBN들을 추정하여 정정하는 방법들이다. 방법들은 전형적으로 한 쌍의 프로세스로서, 기입-에러 검증 프로세스 및 기입-에러 복구 프로세스를 포함한다. 기입-에러 검증 프로세스는 코드워드가 저장 매체에 기입될 때 수행된다. 기입-에러 복구 프로세스는 코드워드가 저장 매체로부터 판독되고 디코딩될 때 나중에 수행된다.

[0071] 도 4는 본 발명의 일 실시예에 따라 채널 제어기(120)에 의해 제어되는 도 1의 통신 시스템(100)에 의해 구현되는 LDPC 인코딩/디코딩 프로세스(400)의 흐름도이다. 처리는 단계(402)에서 시작하며 단계(404)로 계속되어 여기서 데이터 소스(102)는 원 정보 워드(104)를 LDPC 인코더(106)에 제공한다. 다음에, 단계(406)에서, LDPC 인코더(106)는 원 정보 워드(104)를 인코딩하여 채널 입력 코드워드(108)를 생성한다. 다음에, 단계(408)에서, 채널 입력 코드워드(108)가 저장 매체(110)(예를 들어, 플래시 드라이브)에 기입된다. 다음에, 단계(410)에서, 기입-에러 검증 프로세스가 수행되는데, 이 프로세스는 이하 도 6 및 도 7의 논의에서 더 상세히 기술된다.

[0072] 나중에, 단계(412)에서, 저장된 코드워드가 저장 매체(110)로부터 판독되고, 채널 검출기(112)에 의해 검출되며, 대응하는 디코더 입력 코드워드에 대한  $L_{\text{ch}}$  값들이 LDPC 디코더(114)에 보내진다. 다음에, 단계(414)에서, LDPC 디코더(114)는 디코더 입력 코드워드를 처리하여, 디코딩된 코드워드를 생성한다. 처리는 LDPC 디코딩 및 다양한 판독-이벤트 방법들로서, 이를테면 에러-플로어 완화 방법들, 미디어-결함 검출 방법들, 및 표준 채널-재시도 방법들을 포함한다. 단계(414)의 처리의 순 효과는 판독 에러들에 기인한 모든 EBN들이 정정되고, 디코딩된 코드워드에 남아 있는 임의의 EBN들이 기입 에러들이라는 것이다.

[0073] 다음에, 단계(416)에서, 디코딩된 코드워드가 어떤 에러 비트들을 갖고 있는지 결정된다. 그렇지 않다면, 디코딩된 코드워드는 DCCW이며, 프로세스는 단계(420)에서 종료한다. 대신에, 디코딩된 코드워드가 하나 이상의 에러 비트들을 갖고 있다면, 단계(418)에서, 기입-에러 복구 프로세스는 디코딩된 코드워드에 수행되고, 이 프로세스는 이하 도 8의 논의에서 더 상세히 기술된다.

[0074] 기입-에러 검증 프로세스(410)는 채널 입력 코드워드가 저장 매체, 예를 들어 플래시 드라이브에, 기입 인코딩된 코드워드로서 초기에 기입될 때 실행된다. 기입-에러 검증 프로세스는 기입 인코딩된 코드워드를 다시 판독하고 대응하는 디코더 입력 코드워드를 생성한다. 프로세스는 디코더 입력 코드워드에 LDPC 디코딩을 수행하여

디코딩된 코드워드를 생성한다. 프로세스는 채널 입력 코드워드를 디코딩된 코드워드와 비교하고 임의의 에러 비트들의 수와 인덱스들을 결정한다. 에러 비트들의 수가 0보다 크고 명시된 임계값 미만이면, 프로세스는 에러-비트 테이블이라고 하는 데이터 구조에 에러-비트 정보를 기입한다. 본 발명의 일 실시예에서, 에러-비트 정보는 코드워드 식별자(예를 들어, 플래시-드라이브 페이지 ID) 및 하나 이상의 에러 비트들의 인덱스들을 포함한다. 디코딩된 코드워드에서 에러 비트의 인덱스는 에러 비트의 위치를 식별한다. 에러-비트 테이블은 전형적으로 저장 매체에 저장된다.

[0075] 도 5는 에러-비트 테이블의 일 실시예의 다이어그램이다. 이동도 테이블(500)은 3개의 열들(필드들) 및 임의의 수의 행들(레코드들)을 포함한다. 열 CODEWORD\_IDENTIFIER는 코드워드 식별자(예를 들어, 플래시-드라이브 페이지 ID)를 내포한다. 열 EB\_INDICES는 기입-에러 검증 프로세스에 의해 식별된 하나 이상의 에러 비트들의 인덱스들을 내포한다. 열 COMPLETE\_SET는 열 EB\_INDICES에서 식별된 에러-비트 인덱스들이 기입-에러 검증 프로세스에 의해 검출된 모든 에러-비트 인덱스들(즉, 트루 또는 1) 또는 적합한 서브세트(즉, 폴스 또는 0)을 나타내는 지를 나타내는 1-비트 트루/폴스 필드이다.

[0076] 도 6은 발명의 일 실시예에 따라 도 4의 단계(410), 즉 기입-에러 검증 프로세스의 흐름도이다. 처리는 단계(602)에서 시작하며 단계(604)로 가서, 기입 인코딩된 코드워드가 저장 매체로부터 판독되고 디코더 입력 코드워드가 생성된다.

[0077] 다음에, 단계(606)에서, LDPC 디코딩은 여러 판독-이벤트 방법들(예를 들어, 에러-플로어 완화 방법들, 미디어-결함 검출 방법들, 표준 채널-재시도 방법들)에 따라 디코더 입력 코드워드에 수행되며, 디코딩된 코드워드가 생성된다. 단계(606)의 처리의 순 효과는 판독 에러들에 기인한 임의의 및 모든 에러 비트들은 정정되고, 디코딩된 코드워드에 남은 임의의 에러 비트들은 기입 에러들에 기인한다는 것이다. 다음에, 단계(608)에서, 디코딩된 코드워드는 비트씩 채널 입력 코드워드와 비교되고, 임의 에러 비트들(비트 미스매칭들에 대응하는)의 수와 인덱스들이 결정된다.

[0078] 단계(610)에서, 에러 비트들의 수가 0이면(즉, 디코더가 단계(606)에서 DCCW에 수렴되면), 기입-에러 검증 프로세스(410)는 단계(616)에서 종료한다. 대신에, 에러 비트들의 수가 명시된 임계값(예를 들어, 32)을 초과하면, 단계(614)에서, 에러들을 정정하기 위해 자신의 방법들을 실행할 수도 있을 도 1의 채널 제어기(120)에 제어를 넘긴다. 예를 들어, 채널 제어기(120)는 저장 매체(예를 들어, 플래시-드라이브 페이지)에 현재의 위치를 배드(bad)로서 표시하고, 채널 입력 코드워드(108)를 다른 위치에 기입하는 것을 시도하기로 결정할 수도 있을 것이다. 채널-제어기 방법들은 특정의 채널 제어기 및 채널-제어기 제조자에 특정한 경향이 있다. 대신에, 에러 비트들의 수가 0보다 크지만 명시된 임계값 이하이면, 단계(612)에서, 하나 이상의 에러-비트 인덱스들은 에러 비트 테이블에 기입된다.

[0079] 테이블 제한값 L은 에러-비트 테이블에 임의의 코드워드에 대해 저장될 수 있는 최대수의 에러-비트 인덱스들을 명시한다. 식별된 모든 에러 비트들에 관한 정보를 저장하는 것이 항상 필요하진 않다. 연구에 따르면, 5,000-비트 LDPC 코드워드들에 대해서, 3이상의 EBN들을 가진 트래핑 세트에 대해서, 이들 EBN들중 임의의 3개의 EBN들을 플립핑하는 것은 트래핑 세트를 와해할 것임을 보였다. 이에 따라, 전형적으로 에러-비트 테이블에 코드워드당 3 이상의 에러 비트들을 저장할 필요가 없다. 이 경우, 3이 테이블 제한값 L이다.

[0080] 또한, 대부분의 트래핑 세트들은 하나 또는 2개의 EBN들을 플립핑함으로써 와해될 수 있다. 단지 하나 또는 2개의 에러-비트 인덱스들만을 저장할 것으로 판단되면, 즉, L이 1 또는 2이고, 식별된 에러 비트들의 수가 L보다 크다면, 이들 인덱스들에서 비트들을 플립핑하는 것이 디코더가 DCCW에 수렴하게 한다는 것을 확실히 하기 위해 L 선택된 인덱스들이 테스트된다. 그렇지 않다면, (i) DCCW에 디코더가 수렴하거나, 또는 (ii) 더 이상의 서브 세트들이 가용하지 않을 때까지, 다른 L 인덱스들의 세트들이 선택되고 테스트된다.

[0081] 도 7은 도 6의 단계(612)의 흐름도이다. 프로세스(612)는 -있다고 한다면- 도 6의 단계(607)에서 검출된 에러-비트 인덱스들 중 어느 것이 에러-비트 테이블에 기입될 것인가를 선택한다.

[0082] 처리는 단계(702)에서 시작하며 단계(704)로 가서 에러 비트들의 수가 테이블 제한 L 이하인지가 결정된다.

[0083] 단계(704)에서, 에러 비트들의 수가 L 이하이면, 모든 에러 비트 인덱스들이 단계(706)에서 선택되고, 변수 VARCOMPLETESET의 값은 1(트루)로 설정된다. 다음에, 단계(708)에서, 레코드는 에러-비트 테이블(예를 들어, 도 5의 500)에 첨부된다. 첨부된 레코드의 필드들은 다음과 같이 설정된다. 필드 CODEWORD\_IDENTIFIER는 고유 코드워드 식별자(예를 들어, 플래시-드라이브 페이지 ID)와 같게 설정된다. 필드 COMPLETE\_SET는 VARCOMPLETESET의 값으로 설정된다. 선택된 에러-비트 인덱스들은 EB\_INDICES들에 저장된다.

- [0084] 대신에, 단계(704)에서, 에러 비트들의 수가 L보다 크다면, 단계(712)에서, L이 3 이상인지가 결정된다. 그러하다면, 단계(714)에서, L 에러-비트 인덱스들은 무작위로 선택되고, VARCOMPLETESET는 0으로 설정된다. 이어서 처리는 단계 716로 계속되어, 변수 VARCOMPLETESET는 0으로 설정된다(즉, 모든 에러-비트 인덱스들 미만이 에러-비트 테이블에 저장되고 있다). 처리는 단계(708)(위에 기술된)로 진행하고 단계(710)에서 종료한다.
- [0085] 대신에, 단계(712)에서, L이 1 또는 2이면, 처리는 단계(718)로 계속되어, L 에러-비트 인덱스들이 선택된다. 다음에, 단계(720)에서, 선택된 인덱스들에서 원 디코더 입력 코드워드(original decoder input codeword)에 이들 비트들은 반전되어 수정된 디코더 입력 코드워드를 생성한다. 다음에, 단계(722)에서, LDPC 디코딩이 수정된 디코더 입력 코드워드에 수행되어 디코딩된 코드워드를 발생한다. 이어서, 단계(724)에서, 디코딩된 코드워드는 채널 입력 코드워드와 비교된다. 디코더 출력 코드워드에서 에러 비트들이 없다면(단계 726), 단계(716)에서, VARCOMPLETESET는 0으로 설정되고, 처리는 단계들(708, 710)(위에 기술된)로 진행한다.
- [0086] 대신에, 단계(726)에서, 디코딩된 코드워드가 임의의 에러 비트들을 내포한 것으로 결정되면, 단계(728)에서, 또 다른 한 세트의 L 에러-비트 인덱스들이 존재하는 것으로 결정된다. 그러하다면, 단계(732)에서, 다른 세트가 선택된다. 이어서, 단계(734)에서, 디코더 입력 코드워드는 원 디코더 입력 코드워드로 재설정되고(즉, 단계(720)에서 행해진 변경들은 원래대로 된다), 처리는 단계(720)로 루프한다.
- [0087] 대신에, 단계(728)에서, 다른 L 에러-비트 인덱스들의 세트들이 존재하지 않는다면, 프로세스(612)는 단계(730)에서 채널 제어기로 제어를 넘기고, 단계(710)에서 종료한다. 채널 제어기의 논의에 대해서, 도 6의 단계(614)의 설명을 참조한다.
- [0088] 시스템이 저장 매체로부터 기입 인코딩된 코드워드를 판독하고, LDPC 디코더가 명시된 최대횟수의 디코딩 반복들 내에서 DCCW에 수렴하기를 실패하였을 때, 도 4의 기입-에러 복구 프로세스(418)는 나중에 실행된다. 기입-에러 복구 프로세스는 고유 코드워드 식별자(예를 들어, 플래시-드라이브 페이지 ID)를 사용하여 실패된 디코딩된 코드워드에 매칭하는 레코드에 대해 에러-비트 테이블을 탐색한다. 매칭이 발견된다면, 에러-비트 인덱스들은 매칭하는 레코드로부터 가져오고, 가져온 인덱스들에 실패된 디코딩된 코드워드 비트들의 값들은 플립핑되어 수정된 코드워드를 산출한다.
- [0089] 저장된 에러-비트 인덱스들이 기입-검증 프로세스 동안 검출된 모든 에러-비트 인덱스들을 나타낸다면(즉, COMPLETE\_SET = 1), 수정된 코드워드는 전형적으로 신드롬 체크 및 주기적 리던던시 체크에만 제공된다. 신드롬 체크는 수정된 디코더 입력 코드워드가 유효 LDPC 코드워드인지를 결정하며, 주기적 리던던시 체크는 유효 LDPC 코드워드가 DCCW인지 결정한다.
- [0090] 대신에, 저장된 에러-비트 인덱스들이 기입-검증 프로세스 동안 검출된 모든 에러-비트 인덱스들 미만을 나타낸다면(즉, COMPLETE\_SET = 0), 수정된 디코딩된 코드워드는 추가의 LDPC 디코딩에 제공된다(예를 들어, 초기화 단계(304)없이 도 3의 프로세스(300)). 기입-에러 복구 프로세스는 (i) 추가의 디코딩이 DCCW에 수렴할 때, 또는 (ii) 명시된 최대횟수의 디코딩 반복들 내에 디코더가 DCCW에 수렴을 실패할 때 종료한다.
- [0091] 도 8은 본 발명의 일 실시예에 따른 도 4의 기입-에러 복구 프로세스(418)의 흐름도이다. 처리는 단계(802)에서 시작하며 단계(804)로 진행하여 실패된 디코딩된 코드워드가 LDPC 디코더로부터 수신된다. 다음에, 단계(806)에서, 프로세스(418)는 디코딩된 코드워드에 대한 매칭에 대해, 예를 들어, 플래시-드라이브 페이지 ID에 의해 탐색하여, 에러-비트 테이블을 탐색한다(예를 들어, 도 5의 500). 매칭이 발견되지 않는다면, 단계(808)에서 프로세스(418)는 채널 제어기에 제어를 넘기고 단계(810)에서 종료하며; 그렇지 않다면, 단계(812)에서, EB\_INDICES 및 COMPLETE\_SET의 값들은 비트-에러 테이블로부터 가져온다.
- [0092] 다음에, 단계(814)에서, 프로세스(418)는 가져온 에러-비트 인덱스들에서 디코딩된 코드워드 비트들의 값들을 플립핑하여 수정된 코드워드를 산출한다. 플립핑은 위에 단락들에서 기술되었다. 다음에, 단계(816)에서, COMPLETE\_SET이 0이면(즉, 에러-비트 인덱스들의 적합한 서브세트만이 에러-비트 테이블에 저장되어 있다면), 수정된 코드워드가 단계(818)에서 LDPC 디코딩에 제공된다. 단계(818)가 DCCW를 산출한다면(단계 820), 프로세스(418)는 단계(810)에서 종료한다. 단계(818)가 DCCW를 산출하지 않는다면, 단계(808)에서, 제어는 채널 제어기에 넘겨지고, 프로세스(418)는 단계(810)에서 종료한다.
- [0093] 한편으로, 단계(816)에서, COMPLETE\_SET가 1이면(즉, 모든 에러-비트 인덱스들이 에러-비트 테이블에 저장되었다면), 수정된 코드워드가 단계(822)에서 신드롬 체크에 제공된다. 수정된 코드워드가 신드롬 체크에 실패하면, 프로세스(418)는 채널 제어기에 제어를 넘기고(단계(808)), 단계(810)에서 종료한다. 대신에 수정된 디코더 입력 코드워드가 단계(822)에서 신드롬 체크를 통과하면, CRC 체크가 단계(824)에서 수행된다. 수정된 디코더 입

력 코드워드가 CRC 체크를 통과하면, 처리는 단계(810)에서 종료한다. 대신에 단계(824)에서, 수정된 코드워드가 CRC 체크를 통과하지 못하면, 제어는 채널 제어기에 넘겨지고(단계(808)), 프로세스(418)는 단계(810)에서 종료한다.

[0094] 기입-복구 프로세스(410)의 다른 실시예에서, 단계(814)는 디코딩된 코드워드가 아니라 디코더 입력 코드워드에 비트들을 플립핑한다.

[0095] 요약하여, 본 발명의 실시예들은 기입 에러들을 내포하는 NCW로부터 DCCW를 생성하는 방법들이다. 방법들은 전형적으로 기입-검증 프로세스 및 기입-에러 복구 프로세스를 포함한다. 채널 입력 코드워드가 저장 매체에 기입될 때 실행되는, 기입-검증 프로세스는 기입된 코드워드에 기입 에러들, 예를 들어, 에러 비트들을 식별하고, 하나 이상의 에러 비트들의 인덱스들을 에러-비트 테이블에 저장한다. 디코더가 코드워드를 디코딩하려 하였으나 NCW엔 실패하였을 때 나중에 실행되는, 기입-복구 프로세스는 NCW에 연관된 에러-비트 인덱스들에 대해 에러-비트 테이블을 탐색한다. 연관된 에러-비트 인덱스들이 발견된다면, NCW 내 대응하는 비트들은 플립핑되고, NCW은 추가의 LDPC 디코딩에 제공된다.

[0096] 본 발명이 하드디스크 드라이브 및 플래시 드라이브들의 맥락에서 기술되었을지라도, 본 발명은 이것으로 제한되지 않는다. 일반적으로, 본 발명은 어떤 적합한 저장 매체로 구현될 수 있다.

[0097] 또한, 본 발명의 실시예들이 LDPC 코드들의 맥락에서 기술되었을지라도, 본 발명은 이것으로 제한되지 않는다. 본 발명의 실시예들은 트래핑 세트들이 야기되는 그래프로 정의되는 코드들이기 때문에, 그래프에 의해 정의될 수 있는 임의의 코드, 예를 들어, 토네이도 코드들, 구조형 IRA 코드들에 대해 구현될 수도 있을 것이다.

[0098] 본 발명은 이들 방법들을 실시하기 위한 방법들 및 장치들 형태로 구현될 수 있다. 또한, 본 발명은 자기 기록 매체, 광학 기록 매체, 고체상태 메모리, 플로피 디스켓들, CD-ROM들, 하드 드라이브들, 또는 이외 어떤 다른 기계-판독가능 저장 매체와 같은 실제 미디어에 구현된 프로그램 코드 형태로 구현될 수 있고, 프로그램 코드가 컴퓨터와 같은 기계에 로딩되어 이에 의해 실행될 때, 기계는 본 발명을 실시하는 장치가 된다. 본 발명은 또한 예를 들어, 저장 매체에 저장되든 아니면 기계에 로딩되고 및/또는 이에 의해 실행되든 프로그램 코드 형태로 구현될 수 있고, 프로그램 코드가 컴퓨터와 같은 기계에 로딩되어 실행될 때, 기계는 발명을 실시하는 장치가 된다. 범용 프로세서에 구현될 때, 프로그램 코드 세그먼트들은 프로세서와 결합하여 특정 논리회로들과 유사하게 동작하는 유일무이의 디바이스를 제공한다.

[0099] 달리 분명히 언급되지 않는 한, 각각의 숫자 값 및 범위는 "약" 또는 "대략"이라는 어구가 값 또는 범위의 값 앞에 놓인 것으로 하여 근사화되는 것으로 해석되어야 한다.

[0100] 이 발명의 특징을 설명하기 위해 기술되고 도시된 상세, 재료들, 및 부품들의 배열들에서 다양한 변경들은 다음 청구항들에 표현된 본 발명의 범위 내에서 당업자들에 의해 행해질 수 있음을 또한 알 것이다.

[0101] 청구항들에서 도면 부호들 및/또는 도면 참조부호들의 사용은 청구항들의 해석을 용이하게 하기 위해 청구된 요지의 하나 이상의 가능한 실시예들을 식별하게 하려는 것이다. 이러한 사용은 반드시 대응하는 도면들에 도시된 실시예들로 이들 청구항들의 범위를 제한하는 것으로 해석되지 않아야 한다.

[0102] 여기에 개시된 예시적인 방법들의 단계들은 반드시 기술된 순서로 수행되어야 하는 것은 아니며, 이러한 방법들의 단계들의 순서는 단지 예시적인 것으로 이해되어야 한다는 것이 이해되어야 한다. 마찬가지로, 추가의 단계들이 이러한 방법들에 포함될 수 있고, 어떤 단계들은 본 발명의 여러 실시예들에 일관된 방법들에서 생략되거나 결합될 수 있다.

[0103] 다음 방법 청구항들에서 요소들이 -만약 있다면- 대응하는 라벨링에 따라 특별한 순서로 인용될지라도, 청구항 인용들이 달리 이들 일부 또는 모든 요소들을 구현하기 위한 특별한 순서를 의미하지 않는 한, 이들 요소들은 반드시 이 특별한 순서로 구현되는 것을 한정되도록 하려는 것은 아니다.

[0104] 여기에서 "일 실시예" 또는 "실시예"라는 언급은 실시예에 관련하여 기술된 특별한 특징, 구조, 또는 특성이 본 발명의 적어도 일 실시예에 포함될 수 있음을 의미한다. 명세서에 여러 곳에서 "일 실시예에서"라는 어구의 출현들은 반드시 이들 모두가 동일 실시예를 언급하는 것도 아니고, 다른 실시예들을 반드시 상호 배제하는 별도의 또는 대안적 실시예들을 언급하는 것도 아니다. 상기와 동일한 것들이 "구현"이라는 용어에도 적용된다.

**부호의 설명**

[0105] 100 : 통신 시스템

102 : 데이터 소스

106 : LDPC 인코더

110 : 저장 매체

112 : 채널 검출기

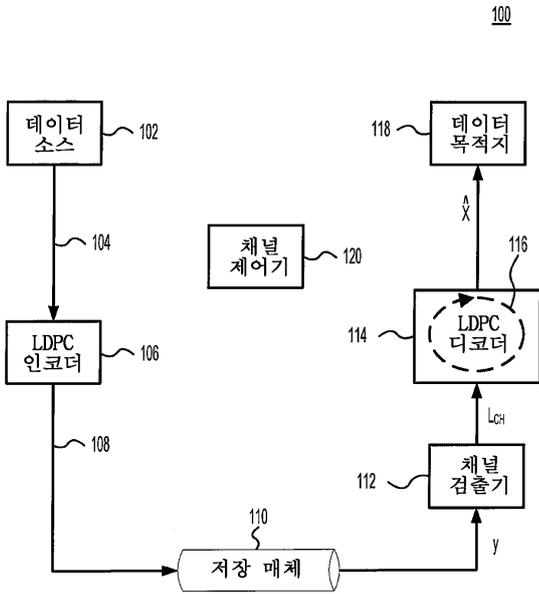
114 : LDPC 디코더

118 : 데이터 목적지

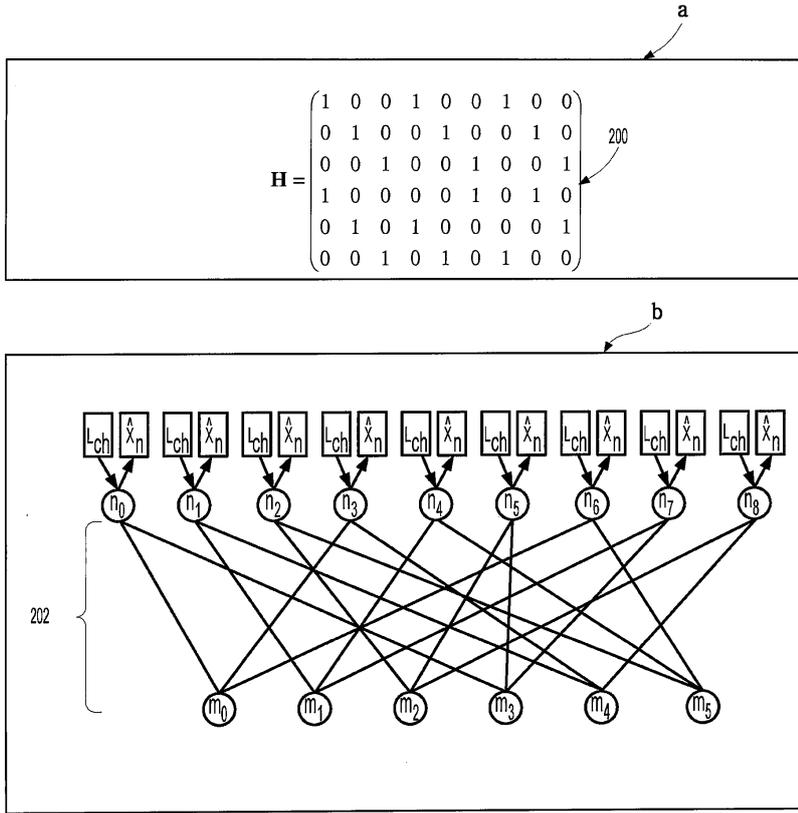
120 : 채널 제어기

도면

도면1

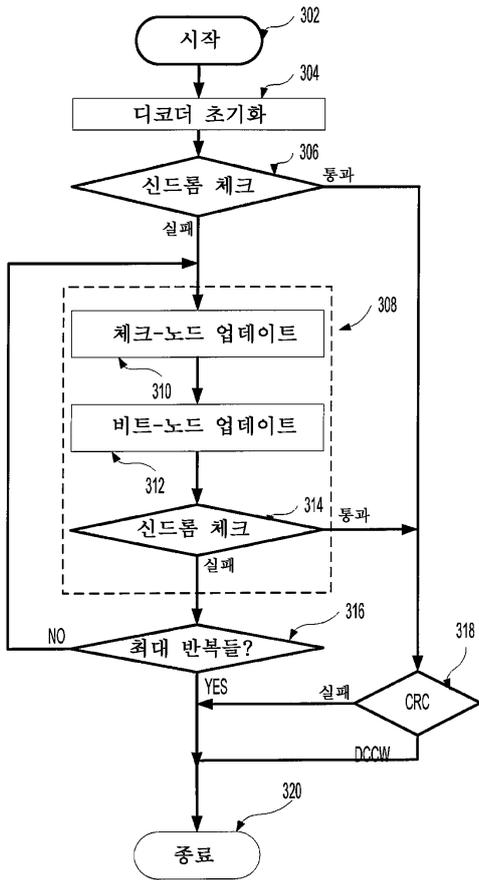


도면2

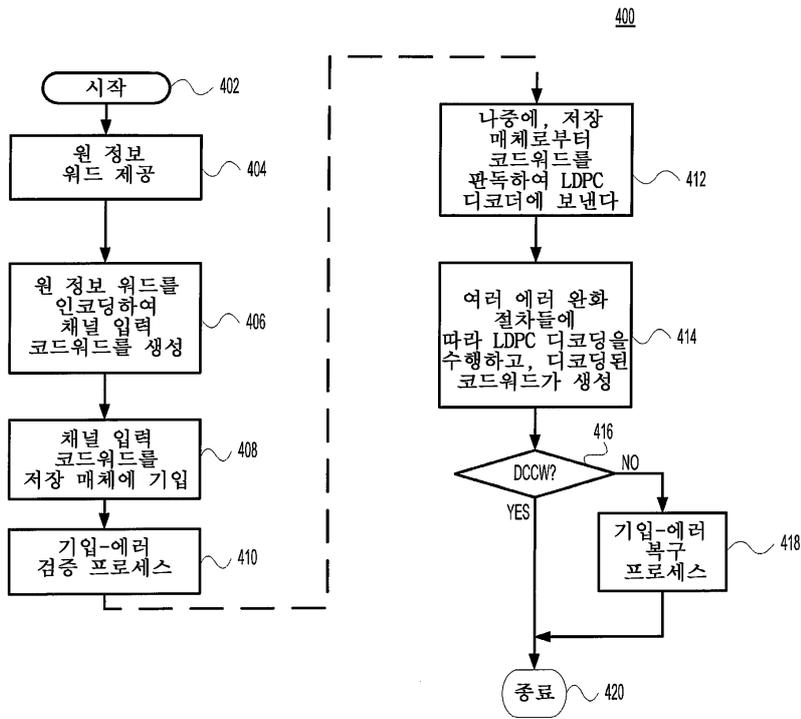


도면3

300



도면4



도면5

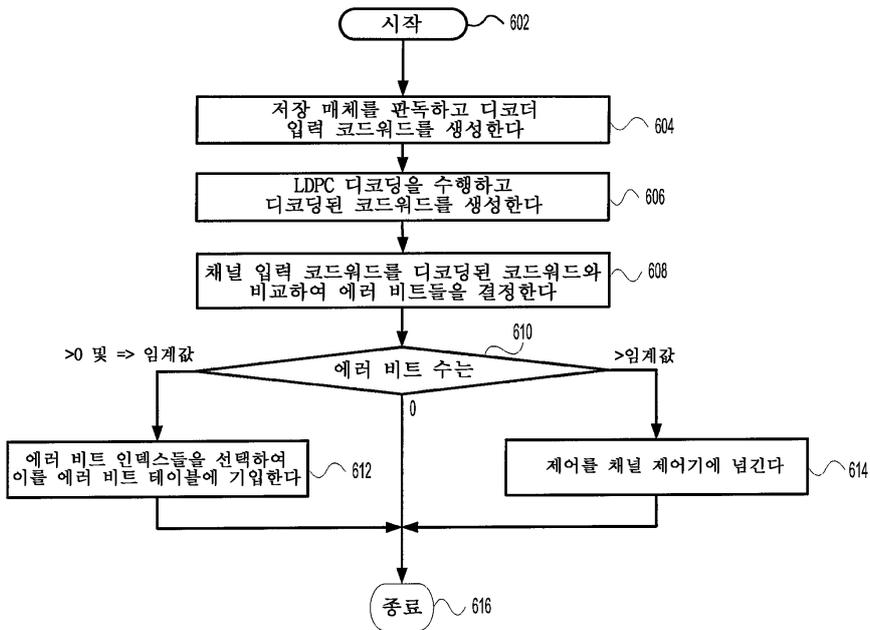
500

CODEWORD_IDENTIFIER	EB_INDICES	COMPLETE_SET
4002	3, 42, 101	0
391	30, 1007	1

·  
·  
·

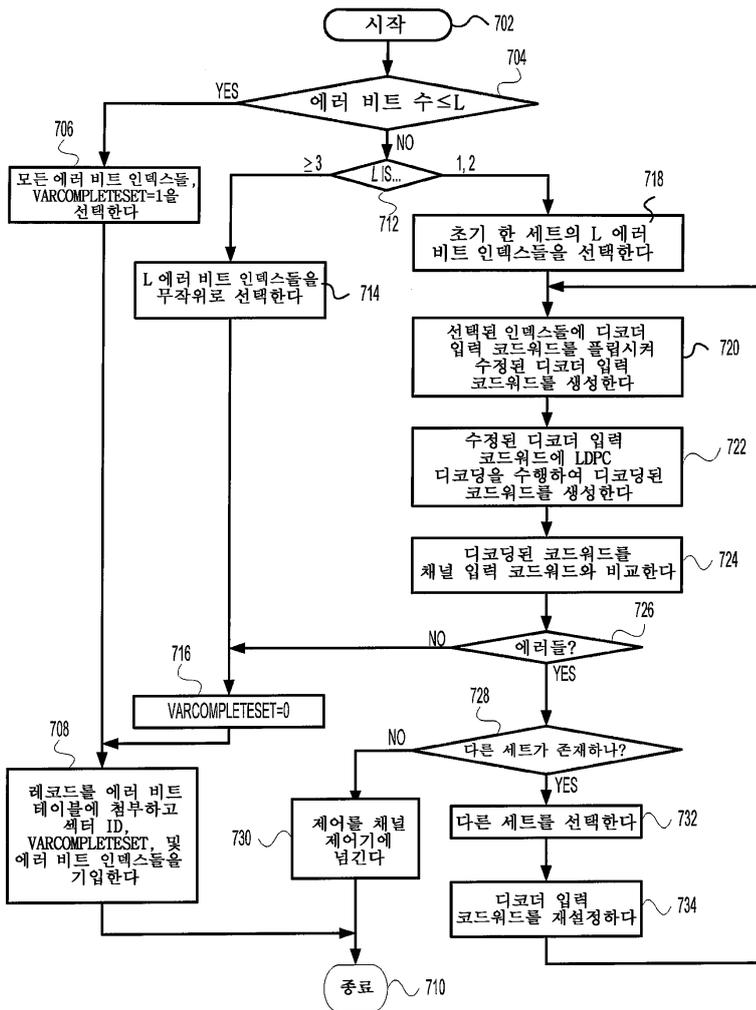
도면6

410



도면7

612



도면8

418

