



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2021년06월14일
(11) 등록번호 10-2265247
(24) 등록일자 2021년06월09일

- (51) 국제특허분류(Int. Cl.)
H03M 13/11 (2006.01) H03M 13/00 (2017.01)
- (52) CPC특허분류
H03M 13/1105 (2013.01)
H03M 13/6502 (2013.01)
- (21) 출원번호 10-2019-7012298
- (22) 출원일자(국제) 2017년09월23일
심사청구일자 2020년09월07일
- (85) 번역문제출일자 2019년04월26일
- (65) 공개번호 10-2019-0071728
- (43) 공개일자 2019년06월24일
- (86) 국제출원번호 PCT/US2017/053128
- (87) 국제공개번호 WO 2018/084956
국제공개일자 2018년05월11일
- (30) 우선권주장
62/416,584 2016년11월02일 미국(US)
15/712,845 2017년09월22일 미국(US)
- (56) 선행기술조사문헌
US20090113256 A1
US20050138519 A1

- (73) 특허권자
켈컴 인코포레이티드
미국 92121-1714 캘리포니아주 샌 디에고 모어하우스 드라이브 5775
- (72) 발명자
론크 빈센트
미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775
바라트카르 기리쉬
미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775
(뒷면에 계속)
- (74) 대리인
특허법인코리아나

전체 청구항 수 : 총 30 항

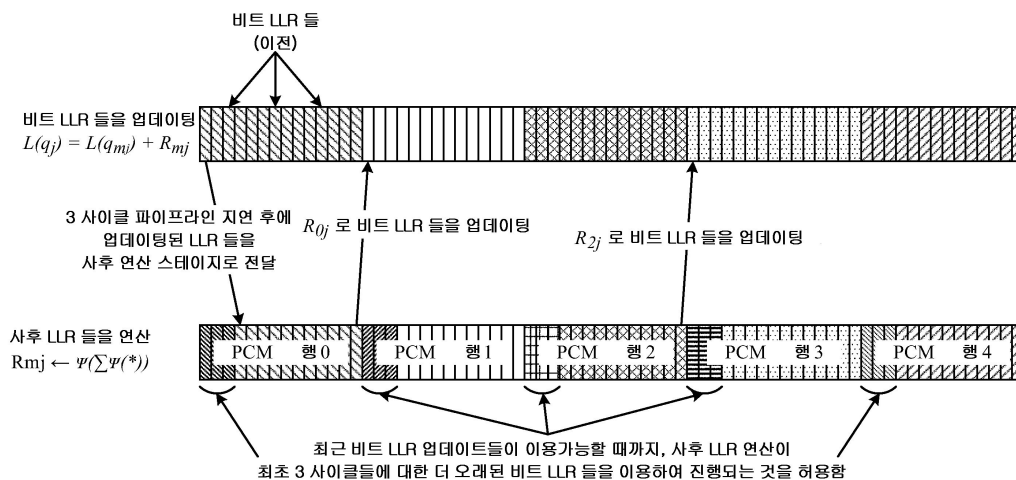
심사관 : 조춘근

(54) 발명의 명칭 **파이프라인 방식의 높은-스루풋의 계층화된 LDPC 디코더 아키텍처**

(57) 요약

본 개시물의 특정 양태들은 일반적으로, 저밀도 패리티 체크 (LDPC) 코드들을 디코딩하기 위한 방법들 및 장치, 그리고 더 상세하게는, 메모리 일관성 및 충돌 문제들을 해결하는, 높은 디코딩 스루풋들을 위한 심층적-파이프라인 방식의 계층화된 LDPC 디코더 아키텍처에 관한 것이다. 본 개시물의 양태들은 특정한 행에 대하여, 체 (뒷면에 계속)

대표도



크 노드 프로세싱이 최근의 업데이트들 (예컨대, 업데이트된 비트 LLR 들) 이 발생하는 것을 대기하는 것이 아니라, 최근에 이용가능한 비트 LLR 들 (예컨대, 아웃데이팅된 비트 LLR 들) 을 이용할 수 있도록, 예를 들어, 비트 LLR 들을 업데이트하는 것과 체크 노드 메시지들을 연산하는 것 사이의 중속성을 완화시킴으로써, 파이프라인 지연들을 저감시키기 위한 기법들을 제시한다. 메모리 일관성 충돌들은 최근의 이용가능한 비트 LLR 들을 LLR 메모리 내에 저장함으로써, 그리고 오래된 것과 새로운 체크 노드 메시지들 사이의 차이로 비트 LLR 들을 업데이트함으로써 회피된다. 또한, LLR 메모리를 듀얼 뱅크들로 논리적으로 분할하는 것은 디코더가 동시에 메모리의 뱅크들 양자 모두로부터 관독하거나 기록하는 것을 가능하게 함으로써, 관독/기록 대역폭을 증가시킨다. 유리하게도, 패리티 체크 행렬 행 (PCM) 계산 순서를 선택하는 것, 메모리에서의 체크 노드 메시지들 및 비트 LLR 업데이트들을 정렬하는 것, 및/또는 예를 들어, PCM 에서의 행들 사이의 중속성들에 기초하여 체크 노드 메시지들 및 비트 LLR 업데이트들을 저장하기 위한 메모리 뱅크를 선택하는 것은 메모리 충돌들/일관성 에러들을 경감시키고 파이프라인 프로세싱 지연들을 감소시킨다.

(72) 발명자

리차드슨 토마스 조셉

미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775

카오 이

미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775

명세서

청구범위

청구항 1

저밀도 패리티 체크 (low-density parity-check; LDPC) 디코딩을 수행하기 위한 방법으로서,

LDPC 인코딩을 이용하여 인코딩된 비트들에 대응하는 로그-우도 비율 (log-likelihood ratio; LLR) 들을 수신하는 단계;

패리티 체크 행렬 (parity check matrix; PCM) 의 열 (column) 들에 대응하는 상기 인코딩된 비트들의 각각에 대한 상기 LLR 들을 업데이트하는 단계로서, 상기 인코딩된 비트들의 각각에 대하여 업데이트된 상기 LLR 들은 상기 인코딩된 비트들 각각의 값의 확률을 표시하는, 상기 LLR 들을 업데이트하는 단계;

상기 인코딩된 비트들의 각각에 대하여, 행 (row) 내의 각각의 LLR 을 프로세싱함으로써 상기 PCM 의 상기 행에 대한 사후 LLR (a posteriori LLR) 을 결정하는 단계로서, 상기 인코딩된 비트들의 제 1 세트에 대하여, 상기 사후 LLR 을 결정하는 단계는 제 1 수의 아웃데이팅된 (outdated) LLR 들을 이용하는 단계를 포함하고, 상기 인코딩된 비트들의 제 2 세트에 대하여, 상기 사후 LLR 을 결정하는 단계는 제 2 수의 업데이트된 LLR 들을 이용하는 단계를 포함하고, 상기 인코딩된 비트들의 상기 제 1 세트 및 상기 인코딩된 비트들의 상기 제 2 세트는 상기 인코딩된 비트들의 전부를 포함하는, 상기 사후 LLR 을 결정하는 단계; 및

상기 인코딩된 비트들을 디코딩하기 위하여 상기 인코딩된 비트들의 각각에 대하여 결정된 상기 사후 LLR 을 포함하는 상기 LLR 들을 사용하는 단계를 포함하는, 저밀도 패리티 체크 (LDPC) 디코딩을 수행하기 위한 방법.

청구항 2

제 1 항에 있어서,

상기 LLR 들을 업데이트하는 단계는 대응하는 사후 LLR 에 기초하는, 저밀도 패리티 체크 (LDPC) 디코딩을 수행하기 위한 방법.

청구항 3

제 2 항에 있어서,

상기 대응하는 사후 LLR 에 기초하여 상기 LLR 들을 업데이트하는 단계는 대응하는 아웃데이팅된 사후 LLR 과 대응하는 업데이트된 사후 LLR 사이의 차이에 기초하여 상기 LLR 들을 업데이트하는 단계를 포함하는, 저밀도 패리티 체크 (LDPC) 디코딩을 수행하기 위한 방법.

청구항 4

제 1 항에 있어서,

상기 인코딩된 비트들의 제 3 세트에 대응하는 업데이트된 LLR 들의 제 1 세트를 제 1 메모리 뱅크 (memory bank) 내에 저장하는 단계, 및 상기 인코딩된 비트들의 제 4 세트에 대응하는 업데이트된 LLR 들의 제 2 세트를 제 2 메모리 뱅크 내에 저장하는 단계를 더 포함하고, 상기 인코딩된 비트들의 상기 제 3 세트 및 상기 인코딩된 비트들의 상기 제 4 세트는 상기 인코딩된 비트들의 전부를 포함하는, 저밀도 패리티 체크 (LDPC) 디코딩을 수행하기 위한 방법.

청구항 5

제 4 항에 있어서,

상기 제 1 메모리 뱅크로부터 판독하는 것과 동시에 상기 제 2 메모리 뱅크로부터 판독하는 단계를 더 포함하는, 저밀도 패리티 체크 (LDPC) 디코딩을 수행하기 위한 방법.

청구항 6

제 4 항에 있어서,

상기 PCM 의 행들 사이의 종속성들에 기초하여 상기 인코딩된 비트들의 상기 제 3 세트 또는 상기 인코딩된 비트들의 상기 제 4 세트에 대한 비트들을 선택하는 단계를 더 포함하는, 저밀도 패리티 체크 (LDPC) 디코딩을 수행하기 위한 방법.

청구항 7

제 6 항에 있어서,

상기 PCM 에서의 행들 사이의 종속성들에 기초하여 상기 LLR 들을 업데이트하기 위한 순서를 결정하는 단계를 더 포함하는, 저밀도 패리티 체크 (LDPC) 디코딩을 수행하기 위한 방법.

청구항 8

제 1 항에 있어서,

상기 인코딩된 비트들을 디코딩하기 위하여 상기 인코딩된 비트들의 각각에 대하여 결정된 상기 사후 LLR 을 포함하는 상기 LLR 들을 사용하는 단계는, 상기 LLR 들을 사용하여 상기 인코딩된 비트들에 대한 계층화된 디코딩 (layered decoding) 을 수행하는 단계를 포함하는, 저밀도 패리티 체크 (LDPC) 디코딩을 수행하기 위한 방법.

청구항 9

저밀도 패리티 체크 (LDPC) 디코딩을 수행하기 위한 장치로서,

적어도 하나의 프로세서로서,

LDPC 인코딩을 이용하여 인코딩된 비트들에 대응하는 로그-우도 비율 (LLR) 들을 수신하고;

패리티 체크 행렬 (PCM) 의 열들에 대응하는 상기 인코딩된 비트들의 각각에 대한 상기 LLR 들을 업데이트하는 것으로서, 상기 인코딩된 비트들의 각각에 대하여 업데이트된 상기 LLR 들은 상기 인코딩된 비트들 각각의 값의 확률을 표시하는, 상기 LLR 들을 업데이트하고;

상기 인코딩된 비트들의 각각에 대하여, 행 내의 각각의 LLR 을 프로세싱함으로써 상기 PCM 의 상기 행에 대한 사후 LLR 을 결정하는 것으로서, 상기 인코딩된 비트들의 제 1 세트에 대하여, 상기 사후 LLR 을 결정하는 것은 제 1 수의 아웃데이팅된 LLR 들을 이용하는 것을 포함하고, 상기 인코딩된 비트들의 제 2 세트에 대하여, 상기 사후 LLR 을 결정하는 것은 제 2 수의 업데이트된 LLR 들을 이용하는 것을 포함하고, 상기 인코딩된 비트들의 상기 제 1 세트 및 상기 인코딩된 비트들의 상기 제 2 세트는 상기 인코딩된 비트들의 전부를 포함하는, 상기 사후 LLR 을 결정하고; 그리고

상기 인코딩된 비트들을 디코딩하기 위하여 상기 인코딩된 비트들의 각각에 대하여 결정된 상기 사후 LLR 을 포함하는 상기 LLR 들을 사용하도록 구성되는, 상기 적어도 하나의 프로세서; 및

상기 적어도 하나의 프로세서와 결합된 메모리를 포함하는, 저밀도 패리티 체크 (LDPC) 디코딩을 수행하기 위한 장치.

청구항 10

제 9 항에 있어서,

상기 LLR 들을 업데이트하는 것은 대응하는 사후 LLR 에 기초하는, 저밀도 패리티 체크 (LDPC) 디코딩을 수행하기 위한 장치.

청구항 11

제 10 항에 있어서,

상기 대응하는 사후 LLR 에 기초하여 상기 LLR 들을 업데이트하는 것은 대응하는 아웃데이팅된 사후 LLR 과 대응하는 업데이트된 사후 LLR 사이의 차이에 기초하여 상기 LLR 들을 업데이트하는 것을 포함하는, 저밀도 패리티 체크 (LDPC) 디코딩을 수행하기 위한 장치.

청구항 12

제 9 항에 있어서,

상기 적어도 하나의 프로세서는 상기 인코딩된 비트들의 제 3 세트에 대응하는 업데이트된 LLR 들의 제 1 세트를 제 1 메모리 बैं크 내에 저장하고, 그리고 상기 인코딩된 비트들의 제 4 세트에 대응하는 업데이트된 LLR 들의 제 2 세트를 제 2 메모리 बैं크 내에 저장하도록 추가로 구성되고, 상기 인코딩된 비트들의 상기 제 3 세트 및 상기 인코딩된 비트들의 상기 제 4 세트는 상기 인코딩된 비트들의 전부를 포함하고, 상기 메모리는 상기 제 1 메모리 बैं크 및 상기 제 2 메모리 बैं크를 포함하는, 저밀도 패리티 체크 (LDPC) 디코딩을 수행하기 위한 장치.

청구항 13

제 12 항에 있어서,

상기 적어도 하나의 프로세서는 상기 제 1 메모리 बैं크로부터 판독하는 것과 동시에, 상기 제 2 메모리 बैं크로부터 판독하도록 추가로 구성되는, 저밀도 패리티 체크 (LDPC) 디코딩을 수행하기 위한 장치.

청구항 14

제 12 항에 있어서,

상기 적어도 하나의 프로세서는 상기 PCM 에서의 행들 사이의 종속성들에 기초하여 상기 인코딩된 비트들의 상기 제 3 세트 또는 상기 인코딩된 비트들의 상기 제 4 세트에 대한 비트들을 선택하도록 추가로 구성되는, 저밀도 패리티 체크 (LDPC) 디코딩을 수행하기 위한 장치.

청구항 15

제 14 항에 있어서,

상기 적어도 하나의 프로세서는 상기 PCM 에서의 행들 사이의 종속성들에 기초하여 상기 LLR 들을 업데이트하기 위한 순서를 결정하도록 추가로 구성되는, 저밀도 패리티 체크 (LDPC) 디코딩을 수행하기 위한 장치.

청구항 16

제 9 항에 있어서,

상기 인코딩된 비트들을 디코딩하기 위하여 상기 인코딩된 비트들의 각각에 대하여 결정된 상기 사후 LLR 을 포함하는 상기 LLR 들을 사용하는 것은, 상기 LLR 들을 사용하여 상기 인코딩된 비트들에 대한 계층화된 디코딩을 수행하는 것을 포함하는, 저밀도 패리티 체크 (LDPC) 디코딩을 수행하기 위한 장치.

청구항 17

저밀도 패리티 체크 (LDPC) 디코딩을 수행하기 위한 장치로서,

LDPC 인코딩을 이용하여 인코딩된 비트들에 대응하는 로그-우도 비율 (LLR) 들을 수신하기 위한 수단;

패리티 체크 행렬 (PCM) 의 열들에 대응하는 상기 인코딩된 비트들의 각각에 대한 상기 LLR 들을 업데이트하기 위한 수단으로서, 상기 인코딩된 비트들의 각각에 대하여 업데이트된 상기 LLR 들은 상기 인코딩된 비트들 각각의 값의 확률을 표시하는, 상기 LLR 들을 업데이트하기 위한 수단;

상기 인코딩된 비트들의 각각에 대하여, 행 내의 각각의 LLR 을 프로세싱함으로써 상기 PCM 의 상기 행에 대한 사후 LLR 을 결정하기 위한 수단으로서, 상기 인코딩된 비트들의 제 1 세트에 대하여, 상기 사후 LLR 을 결정하는 것은 제 1 수의 아웃데이팅된 LLR 들을 이용하는 것을 포함하고, 상기 인코딩된 비트들의 제 2 세트에 대하여, 상기 사후 LLR 을 결정하는 것은 제 2 수의 업데이트된 LLR 들을 이용하는 것을 포함하고, 상기 인코딩된 비트들의 상기 제 1 세트 및 상기 인코딩된 비트들의 상기 제 2 세트는 상기 인코딩된 비트들의 전부를 포함하는, 상기 사후 LLR 을 결정하기 위한 수단; 및

상기 인코딩된 비트들을 디코딩하기 위하여 상기 인코딩된 비트들의 각각에 대하여 결정된 상기 사후 LLR 을 포함하는 상기 LLR 들을 사용하기 위한 수단을 포함하는, 저밀도 패리티 체크 (LDPC) 디코딩을 수행하기 위한 장

치.

청구항 18

제 17 항에 있어서,

상기 업데이트하기 위한 수단은 대응하는 사후 LLR 에 기초하여 상기 LLR 들을 업데이트하기 위한 수단을 더 포함하는, 저밀도 패리티 체크 (LDPC) 디코딩을 수행하기 위한 장치.

청구항 19

제 18 항에 있어서,

상기 대응하는 사후 LLR 에 기초하여 상기 LLR 들을 업데이트하기 위한 수단은 대응하는 아웃데이팅된 사후 LLR 과 대응하는 업데이트된 사후 LLR 사이의 차이에 기초하여 상기 LLR 들을 업데이트하기 위한 수단을 더 포함하는, 저밀도 패리티 체크 (LDPC) 디코딩을 수행하기 위한 장치.

청구항 20

제 17 항에 있어서,

상기 인코딩된 비트들의 제 3 세트에 대응하는 업데이트된 LLR 들의 제 1 세트를 제 1 메모리 뱅크 내에 저장하기 위한 수단, 및 상기 인코딩된 비트들의 제 4 세트에 대응하는 업데이트된 LLR 들의 제 2 세트를 제 2 메모리 뱅크 내에 저장하기 위한 수단을 더 포함하고, 상기 인코딩된 비트들의 상기 제 3 세트 및 상기 인코딩된 비트들의 상기 제 4 세트는 상기 인코딩된 비트들의 전부를 포함하는, 저밀도 패리티 체크 (LDPC) 디코딩을 수행하기 위한 장치.

청구항 21

제 20 항에 있어서,

상기 제 1 메모리 뱅크로부터 판독하는 것과 동시에 상기 제 2 메모리 뱅크로부터 판독하기 위한 수단을 더 포함하는, 저밀도 패리티 체크 (LDPC) 디코딩을 수행하기 위한 장치.

청구항 22

제 20 항에 있어서,

상기 PCM 에서의 행들 사이의 종속성들에 기초하여 상기 인코딩된 비트들의 상기 제 3 세트 또는 상기 인코딩된 비트들의 상기 제 4 세트에 대한 비트들을 선택하기 위한 수단을 더 포함하는, 저밀도 패리티 체크 (LDPC) 디코딩을 수행하기 위한 장치.

청구항 23

제 22 항에 있어서,

상기 PCM 에서의 행들 사이의 종속성들에 기초하여 상기 LLR 들을 업데이트하기 위한 순서를 결정하기 위한 수단을 더 포함하는, 저밀도 패리티 체크 (LDPC) 디코딩을 수행하기 위한 장치.

청구항 24

제 17 항에 있어서,

상기 인코딩된 비트들을 디코딩하기 위하여 상기 인코딩된 비트들의 각각에 대하여 결정된 상기 사후 LLR 을 포함하는 상기 LLR 들을 사용하기 위한 수단은, 상기 LLR 들을 사용하여 상기 인코딩된 비트들에 대한 계층화된 디코딩을 수행하기 위한 수단을 포함하는, 저밀도 패리티 체크 (LDPC) 디코딩을 수행하기 위한 장치.

청구항 25

명령들을 포함하는, 저밀도 패리티 체크 (LDPC) 디코딩을 수행하기 위한 비-일시적 컴퓨터-판독가능 저장 매체로서,

상기 명령들은, 적어도 하나의 프로세서에 의해 실행될 경우, 상기 적어도 하나의 프로세서를,

LDPC 인코딩을 이용하여 인코딩된 비트들에 대응하는 로그-우도 비율 (LLR) 들을 수신하고;

패리티 체크 행렬 (PCM) 의 열들에 대응하는 상기 인코딩된 비트들의 각각에 대한 상기 LLR 들을 업데이트하는 것으로서, 상기 인코딩된 비트들의 각각에 대하여 업데이트된 상기 LLR 들은 상기 인코딩된 비트들 각각의 값의 확률을 표시하는, 상기 LLR 들을 업데이트하고;

상기 인코딩된 비트들의 각각에 대하여, 행 (row) 내의 각각의 LLR 을 프로세싱함으로써 상기 PCM 의 상기 행에 대한 사후 LLR (a posteriori LLR) 을 결정하는 것으로서, 상기 인코딩된 비트들의 제 1 세트에 대하여, 상기 사후 LLR 을 결정하는 것은 제 1 수의 아웃데이팅된 (outdated) LLR 들을 이용하는 것을 포함하고, 상기 인코딩된 비트들의 제 2 세트에 대하여, 상기 사후 LLR 을 결정하는 것은 제 2 수의 업데이트된 LLR 들을 이용하는 것을 포함하고, 상기 인코딩된 비트들의 상기 제 1 세트 및 상기 인코딩된 비트들의 상기 제 2 세트는 상기 인코딩된 비트들의 전부를 포함하는, 상기 사후 LLR 을 결정하고;

상기 인코딩된 비트들을 디코딩하기 위하여 상기 인코딩된 비트들의 각각에 대하여 결정된 상기 사후 LLR 을 포함하는 상기 LLR 들을 사용하도록 구성하는, 저밀도 패리티 체크 (LDPC) 디코딩을 수행하기 위한 비-일시적 컴퓨터-판독가능 저장 매체.

청구항 26

제 25 항에 있어서,

상기 LLR 들을 업데이트하는 것은 대응하는 사후 LLR 에 기초하고, 상기 대응하는 사후 LLR 에 기초하여 상기 LLR 들을 업데이트하는 것은 대응하는 아웃데이팅된 사후 LLR 과 대응하는 업데이트된 사후 LLR 사이의 차이에 기초하여 상기 LLR 들을 업데이트하는 것을 포함하는, 저밀도 패리티 체크 (LDPC) 디코딩을 수행하기 위한 비-일시적 컴퓨터-판독가능 저장 매체.

청구항 27

제 25 항에 있어서,

상기 적어도 하나의 프로세서를, 상기 인코딩된 비트들의 제 3 세트에 대응하는 업데이트된 LLR 들의 제 1 세트를 제 1 메모리 뱅크 내에 저장하고, 상기 인코딩된 비트들의 제 4 세트에 대응하는 업데이트된 LLR 들의 제 2 세트를 제 2 메모리 뱅크 내에 저장하도록 구성하는 명령들을 더 포함하고, 상기 인코딩된 비트들의 상기 제 3 세트 및 상기 인코딩된 비트들의 상기 제 4 세트는 상기 인코딩된 비트들의 전부를 포함하는, 저밀도 패리티 체크 (LDPC) 디코딩을 수행하기 위한 비-일시적 컴퓨터-판독가능 저장 매체.

청구항 28

제 27 항에 있어서,

상기 제 1 메모리 뱅크로부터 판독하는 것과 동시에 상기 제 2 메모리 뱅크로부터 판독하는 것을 더 포함하는, 저밀도 패리티 체크 (LDPC) 디코딩을 수행하기 위한 비-일시적 컴퓨터-판독가능 저장 매체.

청구항 29

제 27 항에 있어서,

상기 PCM 의 행들 사이의 종속성들에 기초하여 상기 인코딩된 비트들의 상기 제 3 세트 또는 상기 인코딩된 비트들의 상기 제 4 세트에 대한 비트들을 선택하는 것; 또는

상기 PCM 에서의 행들 사이의 종속성들에 기초하여 상기 LLR 들을 업데이트하기 위한 순서를 결정하는 것 중의 적어도 하나를 더 포함하는, 저밀도 패리티 체크 (LDPC) 디코딩을 수행하기 위한 비-일시적 컴퓨터-판독가능 저장 매체.

청구항 30

제 25 항에 있어서,

상기 인코딩된 비트들을 디코딩하기 위하여 상기 인코딩된 비트들의 각각에 대하여 결정된 상기 사후 LLR 을 포

합하는 상기 LLR 들을 사용하기 위한 상기 명령들은, 상기 적어도 하나의 프로세서로 하여금 상기 LLR 들을 사용하여 상기 인코딩된 비트들에 대한 계층화된 디코딩을 수행하게 하는 명령들을 포함하는, 저밀도 패리티 체크 (LDPC) 디코딩을 수행하기 위한 비-일시적 컴퓨터-판독가능 저장 매체.

발명의 설명

기술 분야

- [0001] 관련된 출원들에 대한 상호-참조
- [0002] 이 출원은 2016 년 11 월 2 일자로 출원된 미국 특허 가출원 제 62/416,584 호 및 2017 년 9 월 22 일자로 출원된 미국 특허 출원 제 15/712,845 호의 이익을 주장하고, 그 양자는 그 전체적으로 참조로 본원에 편입된다.
- [0003] 본 개시물의 특정 양태들은 일반적으로, 무선 통신들을 위한 방법들 및 장치에 관한 것으로, 더 상세하게는, 심층적-파이프라인 방식 (deeply-pipelined) 의 높은-스루풋 (high-throughput) 의 저밀도 패리티 체크 (low-density parity-check; LDPC) 디코더 아키텍처에 관한 것이다.

배경 기술

- [0004] 무선 통신 시스템들은 음성, 데이터 등과 같은 다양한 타입들의 통신 콘텐츠를 제공하기 위하여 폭넓게 전개된다. 이 시스템들은 이용가능한 시스템 리소스들 (예컨대, 대역폭 및 송신 전력) 을 공유함으로써 다수의 사용자들과의 통신을 지원할 수 있는 다중-액세스 시스템들일 수도 있다. 이러한 다중-액세스 시스템들의 예들은 롱텀 에볼루션 (Long Term Evolution; LTE) 시스템들, 코드 분할 다중 액세스 (Code Division Multiple Access; CDMA) 시스템들, 시간 분할 다중 액세스 (Time Division Multiple Access; TDMA) 시스템들, 주파수 분할 다중 액세스 (Frequency Division Multiple Access; FDMA) 시스템들, 3 세대 파트너십 프로젝트 (3rd Generation Partnership Project; 3GPP) 롱텀 에볼루션 (LTE) 시스템들, 롱텀 에볼루션 어드밴스드 (Long Term Evolution Advanced; LTE-A) 시스템들, 및 직교 주파수 분할 다중 액세스 (Orthogonal Frequency Division Multiple Access; OFDMA) 시스템들을 포함한다.
- [0005] 일반적으로, 무선 다중-액세스 통신 시스템은 다수의 무선 노드들을 위한 통신을 동시에 지원할 수 있다. 각각의 노드는 순방향 및 역방향 링크들 상에서의 송신들을 통해 하나 이상의 기지국들과 통신한다. 순방향 링크 (또는 다운링크) 는 기지국들로부터 노드들로의 통신 링크를 지칭하고, 역방향 링크 (또는 업링크) 는 노드들로부터 기지국들로의 통신 링크를 지칭한다. 통신 링크들은 단일-입력 단일-출력, 다중-입력 단일-출력, 또는 다중-입력 다중-출력 (multiple-input multiple-output; MIMO) 시스템을 통해 확립될 수도 있다.
- [0006] 현대의 정보화 시대에는, 2진 값들 (예컨대, 1 들 및 0 들) 은 비디오, 오디오, 통계적 정보 등과 같은 다양한 타입들의 정보를 표현하고 이를 통신하기 위하여 이용된다. 불행하게도, 2진 데이터의 저장, 송신, 및/또는 프로세싱 동안에, 에러들이 무의식적으로 도입될 수도 있고; 예를 들어, 1 이 0 으로 변경될 수도 있거나, 그 반대도 마찬가지일 수도 있다.
- [0007] 일반적으로, 데이터 송신의 경우에, 수신기는 잡음 또는 왜곡의 존재 시에 각각의 수신된 비트를 관측하고, 오직 비트의 값의 표시가 획득된다. 이 상황들 하에서, 관측된 값들은 "소프트 (soft)" 비트들의 소스 (source) 로서 획득된다. 소프트 비트는 비트의 값 (예컨대, 1 또는 0) 의 선호된 추정치를, 그 추정치의 신뢰성의 일부 표시와 함께 표시한다. 에러들의 수가 상대적으로 낮을 수도 있지만, 심지어 작은 수의 에러들 또는 왜곡의 레벨은 데이터가 이용불가능한 것으로 귀착될 수 있거나, 송신 에러들의 경우에는, 데이터의 재송신을 필요로 할 수도 있다.
- [0008] 에러들에 대하여 체크하기 위한 메커니즘을 제공하기 위하여, 그리고 일부 경우에는, 에러들을 정정하기 위하여, 2진 데이터는 주의 깊게 설계된 중복성 (redundancy) 을 도입하도록 코딩될 수 있다. 데이터의 유닛의 코딩은 코드 워드 (code word) 로서 통상적으로 지칭되는 것을 생성한다. 그 중복성으로 인해, 코드 워드는 종종, 코드 워드가 생성되었던 데이터의 입력 유닛보다 더 많은 비트들을 포함할 것이다.
- [0009] 중복적인 비트들은 코드 워드를 생성하기 위하여 인코더에 의해 송신된 비트 스트림에 추가된다. 송신된 코드 워드들로부터 발생하는 신호들이 수신되었거나 프로세싱되었을 때, 신호에서 관측된 바와 같은 코드 워드 내에 포함된 중복적인 정보는 원래의 데이터 유닛을 복원하기 위하여 수신된 신호에서 에러들을 식별하고 및/또는 정정하거나, 수신된 신호로부터 왜곡을 제거하기 위하여 이용될 수 있다. 이러한 에러 체크 및/또는 정정은

디코딩 프로세스의 일부로서 구현될 수 있다. 에러들의 부재 시에, 또는 정정가능한 에러들 또는 왜곡의 경우에는, 디코딩은 프로세싱되는 소스 데이터로부터, 인코딩되었던 원래의 데이터 유닛을 복원하기 위하여 이용될 수 있다. 복원불가능한 에러들의 경우에, 디코딩 프로세스는 원래의 데이터가 완전히 복원될 수 없다는 일부 표시를 생성할 수도 있다. 디코딩 실패의 이러한 표시들은 데이터의 재송신을 개시하기 위하여 이용될 수 있다.

[0010] 데이터 송신을 위한 광섬유 라인들의 증가된 이용, 및 데이터가 데이터 저장 디바이스들 (예컨대, 디스크 드라이브들, 테이프들 등)로부터 판독될 수 있고 데이터 저장 디바이스들로 저장될 수 있는 레이트에서의 증가들로, 데이터 저장 및 송신 용량의 효율적인 이용 뿐만 아니라, 높은 레이트들의 속도로 데이터를 인코딩하고 이를 디코딩하기 위한 능력에 대한 증가하는 필요성이 있다.

[0011] 인코딩 효율 및 높은 데이터 레이트들이 중요하지만, 인코딩 및/또는 디코딩 시스템이 광범위한 디바이스들 (예컨대, 소비자 디바이스들)에서의 이용을 위하여 실용적이기 위해서는, 인코더들 및/또는 디코더들이 합리적인 비용으로 구현될 수 있는 것이 중요하다.

[0012] 통신 시스템들은 몇몇 상이한 레이트들에서 동작할 필요가 종종 있다. 구현을 가능한 한 간단하게 유지하고 상이한 레이트들에서의 코딩 및 디코딩을 제공하기 위한 하나의 방법은 조절가능한 저밀도 패리티 체크 (LDPC) 코드들을 이용하는 것이다. 특히, 더 낮은 레이트의 코드들을 펀처링 (puncturing) 함으로써 더 높은 레이트의 LDPC 코드들이 생성될 수 있다.

[0013] 이 다중 액세스 기술들은, 상이한 무선 디바이스들이 도시, 국가, 지역, 및 심지어 글로벌 레벨 상에서 통신하는 것을 가능하게 하는 공통 프로토콜을 제공하기 위하여 다양한 통신 표준들에서 채택되었다. 부상하고 있는 통신 표준의 예는 NR (new radio) 이다. NR 은 3 세대 파트너십 프로젝트 (Third Generation Partnership Project; 3GPP) 에 의해 반포된 LTE 이동 표준 (예컨대, 5G 라디오 액세스) 에 대한 개량들의 세트이다. NR 은 스펙트럼 효율을 개선시킴으로써 이동 광대역 인터넷 액세스를 더 양호하게 지원하고, 비용들을 저하시키고, 서비스들을 개선시키고, 새로운 스펙트럼을 이용하고, 다운링크 (DL) 상에서 그리고 업링크 (UL) 상에서 순환 전치 (cyclic prefix; CP) 를 갖는 OFDMA 를 이용하여 다른 개방형 표준들과 더 양호하게 통합할 뿐만 아니라, 빔포밍 (beamforming), 다중-입력 다중-출력 (MIMO) 안테나 기술, 및 캐리어 어그리게이션 (carrier aggregation) 을 지원하도록 설계된다.

[0014] 이동 광대역 액세스에 대한 수요가 증가하는 것을 계속함에 따라, NR 기술에서의 추가의 개선들에 대한 필요성이 존재한다. 바람직하게는, 이 개선들은 다른 멀티-액세스 기술들과, 이 기술들을 채용하는 통신 표준들에 적용가능해야 한다. 개선들을 위한 하나의 분야는 NR 에 적용가능한, 인코딩/디코딩의 분야이다. 예를 들어, NR 을 위한 고성능 LDPC 코드들에 대한 기법들이 바람직하다.

발명의 내용

[0015] 개시물의 시스템들, 방법들, 및 디바이스들은 각각 몇몇 양태들을 가지고, 그 단일의 하나가 그 바람직한 속성 (attribute) 들을 전적으로 담당하지는 않는다. 뒤따르는 청구항들에 의해 표현된 바와 같은 이 개시물의 범위를 제한하지 않으면서, 일부 특징들이 간략하게 이제 논의될 것이다. 이 논의를 고려한 후에, 그리고 특히, "상세한 설명" 으로 제목이 붙여진 섹션을 읽은 후에, 이 개시물의 특징들이 무선 네트워크에서의 액세스 포인트 (access point) 들과 국 (station) 들 사이의 개선된 통신들을 포함하는 장점들을 어떻게 제공하는지를 이해할 것이다.

[0016] 본 개시물의 특정 양태들은 저밀도 패리티 체크 (LDPC) 디코딩을 수행하기 위한 방법을 제시한다. 예시적인 방법은 일반적으로, LDPC 인코딩을 이용하여 인코딩된 비트들에 대응하는 로그-우도 비율 (log-likelihood ratio; LLR) 들을 수신하는 단계, 패리티 체크 행렬 (parity check matrix; PCM) 의 열 (column) 들에 대응하는 인코딩된 비트들의 각각에 대한 LLR 들을 업데이트하는 단계로서, LLR 들은 인코딩된 비트들의 각각의 비트의 값의 확률을 표시하는, 상기 LLR 들을 업데이트하는 단계, 인코딩된 비트들의 각각에 대하여, 행 (row) 내의 각각의 LLR 을 프로세싱함으로써 PCM 의 행에 대한 사후 LLR (a posteriori LLR) 을 결정하는 단계로서, 인코딩된 비트들의 제 1 세트에 대하여, 사후 LLR 을 결정하는 단계는 제 1 수의 아웃데이팅된 (outdated) LLR 들을 이용하는 단계를 포함하고, 인코딩된 비트들의 제 2 세트에 대하여, 사후 LLR 을 결정하는 단계는 제 2 수의 업데이트된 LLR 들을 이용하는 단계를 포함하고, 인코딩된 비트들의 제 1 세트 및 인코딩된 비트들의 제 2 세트는 인코딩된 비트들의 전부를 포함하는, 상기 사후 LLR 을 결정하는 단계, 및 인코딩된 비트들을 디코딩하기 위하여 LLR 들을 사용하는 단계를 포함한다.

[0017] 본 개시물의 특정 양태들은 저밀도 패리티 체크 (LDPC) 디코딩을 수행하기 위한 장치를 제시한다. 장치는 일반적으로, LDPC 인코딩을 이용하여 인코딩된 비트들에 대응하는 로그-우도 비율 (LLR) 들을 수신하고, 패리티 체크 행렬 (PCM) 의 열들에 대응하는 인코딩된 비트들의 각각에 대한 LLR 들을 업데이트하는 것으로서, 비트 LLR 들은 인코딩된 비트들의 각각의 비트의 값의 확률을 표시하는, 상기 LLR 들을 업데이트하고, 인코딩된 비트들의 각각에 대하여, 행 내의 각각의 LLR 을 프로세싱함으로써, PCM 의 행에 대한 사후 LLR 을 결정하는 것으로서, 인코딩된 비트들의 제 1 세트에 대하여, 사후 LLR 을 결정하는 것은 제 1 수의 아웃데이팅된 LLR 들을 이용하는 것을 포함하고, 인코딩된 비트들의 제 2 세트에 대하여, 사후 LLR 을 결정하는 것은 제 2 수의 업데이트된 LLR 들을 이용하는 것을 포함하고, 인코딩된 비트들의 제 1 세트 및 인코딩된 비트들의 제 2 세트는 인코딩된 비트들의 전부를 포함하는, 상기 사후 LLR 을 결정하고, 그리고 인코딩된 비트들을 디코딩하기 위하여 LLR 들을 사용하도록 구성된 적어도 하나의 프로세서를 포함한다. 장치는 또한 일반적으로, 적어도 하나의 프로세서와 결합된 메모리를 포함한다.

[0018] 본 개시물의 특정 양태들은 저밀도 패리티 체크 (LDPC) 디코딩을 수행하기 위한 장치를 제시한다. 장치는 일반적으로, LDPC 인코딩을 이용하여 인코딩된 비트들에 대응하는 로그-우도 비율 (LLR) 들을 수신하기 위한 수단, 패리티 체크 행렬 (PCM) 의 열들에 대응하는 인코딩된 비트들의 각각에 대한 LLR 들을 업데이트하기 위한 수단으로서, LLR 들은 인코딩된 비트들의 각각의 비트의 값의 확률을 표시하는, 상기 LLR 들을 업데이트하기 위한 수단, 인코딩된 비트들의 각각에 대하여, 행 내의 각각의 LLR 을 프로세싱함으로써 PCM 의 행에 대한 사후 LLR 을 결정하기 위한 수단으로서, 인코딩된 비트들의 제 1 세트에 대하여, 사후 LLR 을 결정하는 것은 제 1 수의 아웃데이팅된 LLR 들을 이용하는 것을 포함하고, 인코딩된 비트들의 제 2 세트에 대하여, 사후 LLR 을 결정하는 것은 제 2 수의 업데이트된 LLR 들을 이용하는 것을 포함하고, 인코딩된 비트들의 제 1 세트 및 인코딩된 비트들의 제 2 세트는 인코딩된 비트들의 전부를 포함하는, 상기 사후 LLR 을 결정하기 위한 수단, 및 인코딩된 비트들을 디코딩하기 위하여 LLR 들을 사용하기 위한 수단을 포함한다.

[0019] 본 개시물의 특정 양태들은 저밀도 패리티 체크 (LDPC) 디코딩을 수행하기 위한 비-일시적 컴퓨터-판독가능 매체를 제시한다. 비-일시적 컴퓨터-판독가능 매체는 일반적으로, 적어도 하나의 프로세서에 의해 실행될 때, 적어도 하나의 프로세서를, LDPC 인코딩을 이용하여 인코딩된 비트들에 대응하는 로그-우도 비율 (LLR) 들을 수신하고, 패리티 체크 행렬 (PCM) 의 열들에 대응하는 인코딩된 비트들의 각각에 대한 LLR 들을 업데이트하는 것으로서, 비트 LLR 들은 인코딩된 비트들의 각각의 비트의 값의 확률을 표시하는, 상기 LLR 들을 업데이트하고, 인코딩된 비트들의 각각에 대하여, 행 내의 각각의 LLR 을 프로세싱함으로써 PCM 의 행에 대한 사후 LLR 을 결정하는 것으로서, 인코딩된 비트들의 제 1 세트에 대하여, 사후 LLR 을 결정하는 것은 제 1 수의 아웃데이팅된 LLR 들을 이용하는 것을 포함하고, 인코딩된 비트들의 제 2 세트에 대하여, 사후 LLR 을 결정하는 것은 제 2 수의 업데이트된 LLR 들을 이용하는 것을 포함하고, 인코딩된 비트들의 제 1 세트 및 인코딩된 비트들의 제 2 세트는 인코딩된 비트들의 전부를 포함하는, 상기 사후 LLR 을 결정하고, 그리고 인코딩된 비트들을 디코딩하기 위하여 LLR 들을 사용하도록 구성된 명령들을 포함한다.

[0020] 본 발명의 다른 양태들, 특징들, 및 실시형태들은 첨부된 도면들과 함께 본 발명의 특정 예시적인 실시형태들의 다음의 설명의 검토 시에 당해 분야의 당업자들에게 명백해질 것이다. 본 발명의 특징들은 이하의 어떤 실시형태들 및 도면들에 관하여 논의될 수도 있지만, 본 발명의 모든 실시형태들은 본원에서 논의된 유리한 특징들 중의 하나 이상을 포함할 수 있다. 다시 말해서, 하나 이상의 실시형태들은 어떤 유리한 특징들을 가지는 것으로서 논의될 수도 있지만, 이러한 특징들 중의 하나 이상은 또한, 본원에서 논의된 발명의 다양한 실시형태들에 따라 이용될 수도 있다. 유사한 방식으로, 예시적인 실시형태들은 디바이스, 시스템, 또는 방법 실시형태들로서 이하에서 논의될 수도 있지만, 이러한 예시적인 실시형태들은 다양한 디바이스들, 시스템들, 및 방법들에서 구현될 수 있다는 것을 이해해야 한다.

도면의 간단한 설명

[0021] 본 개시물의 위에서 인용된 특징들이 상세하게 이해될 수 있도록, 위에서 간략하게 요약된 더 특정한 설명은 양태들을 참조하여 행해질 수도 있고, 양태들의 일부는 첨부된 도면들에서 예시된다. 그러나, 첨부된 도면들은 오직 이 개시물의 어떤 전형적인 양태들을 예시하고, 그러므로, 설명이 다른 동일하게 효과적인 양태들을 인정할 수도 있으므로, 그 범위의 제한으로 고려되지 않아야 한다.

도 1 은 본 개시물의 특정 양태들에 따라, 일 예의 다중 액세스 무선 통신 시스템을 예시한다.

도 2 는 본 개시물의 특정 양태들에 따라, 기지국 및 무선 노드의 블록도를 예시한다.

- 도 3 은 본 개시물의 특정 양태들에 따라, 무선 디바이스에서 사용될 수도 있는 다양한 컴포넌트들을 예시한다.
 - 도 4 내지 도 4a 는 본 개시물의 특정 양태들에 따라, 예시적인 저밀도 패리티 체크 (LDPC) 코드의 그래픽 및 행렬 표현들을 도시한다.
 - 도 5 는 본 개시물의 특정 양태들에 따라, 도 4a 의 LDPC 코드의 리프팅 (lifting) 을 그래픽으로 예시한다.
 - 도 6 은 준순환 (quasi-cyclic) 802.11 LDPC 코드에 대한 행렬의 정수 표현이다.
 - 도 7 은 본 개시물의 특정 양태들에 따라, 펀처링 인코더 (puncturing encoder) 를 예시하는 단순화된 블록도이다.
 - 도 8 은 본 개시물의 특정 양태들에 따라, 디코더를 예시하는 단순화된 블록도이다.
 - 도 9 는 본 개시물의 특정 양태들에 따라, 일반적인 계층화된 LDPC 디코더의 하이 레벨 블록도를 예시한다.
 - 도 10 은 본 개시물의 특정 양태들에 따라, 패리티 체크 행렬에서의 비트 LLR 들 및 사후 LLR 들을 연산/업데이팅하기 위한 이 프로세스의 예를 예시한다.
 - 도 11a 및 도 11b 는 본 개시물의 특정 양태들에 따라, 일 예의 계층화된 디코더 파이프라인 프로세싱 타임라인 (layered decoder pipeline processing timeline) 을 예시한다.
 - 도 12 는 본 개시물의 특정 양태들에 따라, 저밀도 패리티 체크 (LDPC) 코드들을 디코딩하기 위한 일 예의 동작들을 예시하는 흐름도이다.
 - 도 13 은 본 개시물의 양태들에 따라, 비트 LLR 들을 업데이트하는 것과 사후 LLR 들을 연산하는 것 사이의 종속성이 완화된 일 예의 LLR 프로세싱 파이프라인을 예시한다.
 - 도 14a 는 본 개시물의 양태들에 따라, 메모리 일관성 (consistency) 및 충돌 (conflict) 이슈들을 입증하는 목적들을 위한 일 예의 PCM 을 예시한다.
 - 도 14b 는 본 개시물의 양태들에 따라, LLR 메모리를 기록하기 위한 단일 사이클 레이턴시 (latency) 및 3 사이클 파이프라인 심도 (pipeline depth) 를 가정하는 계층화된 LDPC 디코더 파이프라인을 예시한다.
 - 도 15 는 본 개시물의 양태들에 따라, 일 예의 패리티 체크 행렬 (PCM) 에 대응하는 비트 LLR 프로세싱 단계들을 예시한다.
 - 도 16 은 본 개시물의 양태들에 따라, 계층화된 디코더 프로세싱 파이프라인에서의 비트 LLR 들의 교체 (swapping) 를 예시한다.
 - 도 17 은 본 개시물의 양태들에 따라, LLR 들의 프로세싱 및 업데이트를 수용하기 위하여 메모리 대역폭을 증가시키기 위한 예시적인 해결책을 예시한다.
- 이해를 용이하게 하기 위하여, 가능한 경우, 도면들에 공통적인 동일한 엘리먼트들을 표기하기 위하여 동일한 참조 번호들이 이용되었다. 하나의 실시형태에서 개시된 엘리먼트들은 특정 인용 없이 다른 실시형태들에서 유익하게 사용될 수도 있다는 것이 고려된다.

발명을 실시하기 위한 구체적인 내용

[0022] 본 개시물의 양태들은 NR (new radio) (뉴 라디오 액세스 기술) 를 위한 인코딩을 위한 장치, 방법들, 프로세싱 시스템들, 및 컴퓨터 프로그램 제품들을 제공한다. NR (new radio) 은 새로운 무선 인터페이스 (air interface) 또는 고정된 전송 계층에 따라 동작하도록 구성된 라디오 (radio) 들을 지칭할 수도 있다. NR 은 개량된 이동 광대역 (Enhanced mobile broadband; eMBB) 타겟팅 광 대역폭 (예컨대, 80 MHz 초과), 밀리미터파 (millimeter wave; mmW) 타겟팅 높은 캐리어 주파수 (예컨대, 60 GHz), 대용량 MTC (massive MTC; mMTC) 타겟팅 비-역호환가능한 MTC 기법들, 및 미션 크리티컬 (mission critical) 타겟팅 초 신뢰성 저 레이턴시 통신들 (ultra reliable low latency communications; URLLC) 을 포함할 수도 있다. 이 일반적인 주제들에 대하여, 코딩, 저밀도 패리티 체크 (LDPC), 및 폴라 (polar) 와 같은 상이한 기법들이 고려된다. NR 셀은 새로운 무선 인터페이스 또는 고정된 전송 계층에 따라 동작하는 셀을 지칭할 수도 있다. NR 노드 B (예컨대, 5G 노드 B) 는 하나 또는 다수의 송신 수신 포인트 (transmission reception point; TRP) 들에 대응할 수도 있다.

- [0023] 본 개시물의 특정 양태들은 일반적으로, 저밀도 패리티 체크 (LDPC) 코드들을 디코딩하기 위한 방법들 및 장치, 그리고 더 상세하게는, 높은 디코딩 스루풋들을 위한 심층적-파이프라인 방식의 계층화된 LDPC 디코더 아키텍처에 관한 것이다. 예를 들어, 본 개시물의 양태들은 프로세싱 지연들의 감소로 귀착될 수도 있는, 비트 로그 우도 비율 (log likelihood ratio; LLR) 및 사후 LLR 종속성의 완화를 가능하게 하는 기법들을 제시한다. 추가적으로, 본 개시물의 양태들은 비트 LLR 들과 사후 LLR 들 사이의 이 종속성을 완화시킬 때에 발생할 수도 있는 메모리 충돌들 및 일관성 에러들을 해결하기 위한 기법들을 제시한다.
- [0024] 개시물의 다양한 양태들은 첨부된 도면들을 참조하여 이하에서 더 완전하게 설명된다. 그러나, 이 개시물은 많은 상이한 형태들로 구체화될 수도 있지만, 이 개시물의 전반에 걸쳐 제시된 임의의 특정 구조 또는 기능으로 제한되는 것으로 해석되지 않아야 한다. 오히려, 이 양태들은 이 개시물이 더 철저하고 완전하게 되도록, 그리고 이 개시물의 범위를 당해 분야의 당업자들에게 완전히 전달하도록 제공된다. 본원에서의 교시사항들에 기초하여, 당해 분야의 당업자는 개시물의 범위가, 개시물의 임의의 다른 양태에 관계 없이 또는 이와 조합하여 구현되든지 간에, 본원에서 개시된 개시물의 임의의 양태를 커버하도록 의도된다는 것을 인식해야 한다. 예를 들어, 본원에서 기재된 임의의 수의 양태들을 이용하여 장치가 구현될 수도 있거나 방법이 실시될 수도 있다. 추가적으로, 개시물의 범위는 본원에서 기재된 개시물의 다양한 양태들에 추가하여, 또는 이 다양한 양태들 이외에, 다른 구조, 기능성, 또는 구조 및 기능성을 이용하여 실시되는 이러한 장치 또는 방법을 커버하도록 의도된다. 본원에서 개시된 개시물의 임의의 양태는 청구항의 하나 이상의 엘리먼트들에 의해 구체화될 수도 있다는 것이 이해되어야 한다. 단어 "예시적" 은 "예, 사례, 또는 예시로서 작용함" 을 의미하기 위하여 본원에서 이용된다. "예시적" 으로서 본원에서 설명된 임의의 양태는 다른 양태들에 비해 바람직하거나 유익한 것으로 반드시 해석되어야 하는 것은 아니다.
- [0025] 특정한 양태들이 본원에서 설명되지만, 이 양태들의 많은 변형들 및 치환들은 개시물의 범위 내에 속한다. 바람직한 양태들의 일부 이득들 및 장점들이 언급되지만, 개시물의 범위는 특정한 이득들, 용도들, 또는 목적들에 제한되도록 의도된 것이 아니다. 오히려, 개시물의 양태들은 상이한 무선 기술들, 시스템 구성들, 네트워크들, 및 송신 프로토콜들에 폭넓게 적용가능하도록 의도되며, 이들의 일부는 바람직한 양태들의 도면들 및 다음의 설명에서 예로서 예시된다. 상세한 설명 및 도면들은 제한하는 것이 아니라 개시물의 단지 예시이고, 개시물의 범위는 첨부된 청구항들 및 그 등가물들에 의해 정의된다.
- [0026] 본원에서 설명된 기법들은 롱텀 에볼루션 (LTE), 코드 분할 다중 액세스 (CDMA) 네트워크들, 시간 분할 다중 액세스 (TDMA) 네트워크들, 주파수 분할 다중 액세스 (FDMA) 네트워크들, 직교 FDMA (OFDMA) 네트워크들, 단일-캐리어 FDMA (Single-Carrier FDMA; SC-FDMA) 네트워크들 등과 같은 다양한 무선 통신 네트워크들을 위하여 이용될 수도 있다. 용어들 "네트워크들" 및 "시스템들" 은 종종 상호 교환가능하게 이용된다. CDMA 네트워크는 유니버설 지상 라디오 액세스 (Universal Terrestrial Radio Access; UTRA), CDMA2000 등과 같은 라디오 기술 (radio technology) 을 구현할 수도 있다. UTRA 는 광대역-CDMA (Wideband-CDMA; W-CDMA) 및 낮은 칩레이트 (Low Chip Rate; LCR) 를 포함한다. CDMA2000 은 IS-2000, IS-95, 및 IS-856 표준들을 커버한다. TDMA 네트워크는 이동 통신을 위한 글로벌 시스템 (Global System for Mobile Communications; GSM) 과 같은 라디오 기술을 구현할 수도 있다. OFDMA 네트워크는 NR (예컨대, 5G RA), 진화형 UTRA (Evolved UTRA; E-UTRA), IEEE 802.11, IEEE 802.16, IEEE 802.20, 플래시-OFDM (Flash-OFDM) 등과 같은 라디오 기술을 구현할 수도 있다. UTRA, E-UTRA, 및 GSM 은 유니버설 이동 통신 시스템 (Universal Mobile Telecommunication System; UMTS) 의 일부이다. 롱텀 에볼루션 (LTE) 은 E-UTRA 를 이용하는 UMTS 의 릴리즈 (release) 이다. UTRA, E-UTRA, GSM, UMTS, 및 LTE 는 "3 세대 파트너십 프로젝트 (3rd Generation Partnership Project)" (3GPP) 로 명명된 기구로부터의 문서들에서 설명된다. CDMA2000 은 "3 세대 파트너십 프로젝트 2 (3rd Generation Partnership Project 2)" (3GPP2) 로 명명된 기구로부터의 문서들에서 설명된다. NR 은 5G 기술 포럼 (5G Technology Forum; 5GTF) 과 함께 개발 중인 부상하고 있는 무선 통신 기술이다. 이 통신 네트워크들은 이 개시물에서 설명된 기법들이 적용될 수도 있는 네트워크들의 예들로서 단지 열거되지만; 그러나, 이 개시물은 위에서 설명된 통신 네트워크로 제한되지는 않는다.
- [0027] 단일 캐리어 주파수 분할 다중 액세스 (SC-FDMA) 는 송신기 측에서 단일 캐리어 변조 (single carrier modulation) 를, 그리고 수신기 측에서 주파수 도메인 등화 (frequency domain equalization) 를 사용하는 송신 기법이다. SC-FDMA 는 OFDMA 시스템의 그것들과 유사한 성능 및 필수적으로 동일한 전체적인 복잡도를 가진다. 그러나, SC-FDMA 신호는 그 본래의 단일 캐리어 구조로 인해 더 낮은 피크-대-평균 전력 비율 (peak-to-average power ratio; PAPR) 을 가진다. SC-FDMA 는 특히, 더 낮은 PAPR 이 송신 전력 효율의 측면에서 무선 노드에 크게 유익한 업링크 (UL) 통신들에서 큰 주목을 끌었다.

[0028] 액세스 포인트 (access point) ("AP") 는 NodeB, 라디오 네트워크 제어기 (Radio Network Controller) ("RNC"), eNodeB (eNB), 노드 B (예컨대, 5G 노드 B), 송신 수신 포인트 (TRP), 기지국 제어기 (Base Station Controller) ("BSC"), 기지국 트랜시버 (Base Transceiver Station) ("BTS"), 기지국 (Base Station) ("BS"), 트랜시버 기능 (Transceiver Function) ("TF"), 라디오 라우터 (Radio Router), 라디오 트랜시버 (Radio Transceiver), 기본 서비스 세트 (Basic Service Set) ("BSS"), 확장된 서비스 세트 (Extended Service Set) ("ESS"), 라디오 기지국 (Radio Base Station) ("RBS"), 또는 기타 용어를 포함할 수도 있거나, 이들로서 구현될 수도 있거나, 이들로서 알려질 수도 있다.

[0029] 액세스 단말 (access terminal) ("AT") 은 액세스 단말 (access terminal), 가입자국 (subscriber station), 가입자 유닛 (subscriber unit), 이동국 (mobile station), 원격 국 (remote station), 원격 단말 (remote terminal), 사용자 단말 (user terminal), 사용자 에이전트 (user agent), 사용자 디바이스, 사용자 장비 (user equipment; UE), 사용자 국 (user station), 무선 노드 (wireless node), 또는 기타 용어를 포함할 수도 있거나, 이들로서 구현될 수도 있거나, 이들로서 알려질 수도 있다. 일부 구현예들에서, 액세스 단말은 셀룰러 전화, 스마트폰, 코드리스 전화 (cordless telephone), 세션 개시 프로토콜 (Session Initiation Protocol) ("SIP") 전화, 무선 로컬 루프 (wireless local loop) ("WLL") 국, 개인 정보 단말 (personal digital assistant) ("PDA"), 태블릿, 넷북, 스마트북, 울트라북, 무선 접속 능력을 가지는 핸드헬드 디바이스 (handheld device), 국 ("STA"), 또는 무선 모뎀에 접속된 기타 적당한 프로세싱 디바이스를 포함할 수도 있다.

따라서, 본원에서 교시된 하나 이상의 양태들은 전화 (예컨대, 셀룰러 전화, 스마트폰), 컴퓨터 (예컨대, 데스크톱), 휴대용 통신 디바이스, 휴대용 컴퓨팅 디바이스 (예컨대, 랩톱, 개인용 데이터 어시스턴트 (personal data assistant), 태블릿, 넷북, 스마트북, 울트라북), 의료용 디바이스들 또는 장비, 생체계측 센서들/디바이스들, 엔터테인먼트 디바이스 (예컨대, 음악 또는 비디오 디바이스, 또는 위성 라디오), 차량 컴포넌트 또는 센서, 스마트 계측기들/센서들, 산업용 제조 장비, 글로벌 위치결정 시스템 디바이스, 또는 무선 또는 유선 매체를 통해 통신하도록 구성되는 임의의 다른 적당한 디바이스 내로 편입될 수도 있다. 일부 양태들에서, 노드는 무선 노드이다. 무선 노드는 예를 들어, 유선 또는 무선 통신 링크를 통해 네트워크 (예컨대, 인터넷 또는 셀룰러 네트워크와 같은 광역 네트워크) 를 위한 또는 이러한 네트워크로의 접속성을 제공할 수도 있다.

[0030] 양태들은 3G 및/또는 4G 무선 기술들과 일반적으로 연관된 용어를 이용하여 본원에서 설명될 수도 있지만, 본 개시물의 양태들은 NR 기술들을 포함하는, 5G 및 그 이후와 같은 다른 세대-기반 통신 시스템들에서 적용될 수도 있다.

[0031] 일 예의 무선 통신 시스템

[0032] 도 1 은 본 개시물의 양태들이 수행될 수도 있는 일 예의 통신 네트워크 (100) 를 예시한다. 예시된 바와 같이, 노드 B (102) (예컨대, TRP 또는 5G 노드 B) 는 다수의 안테나 그룹들을 포함할 수도 있고, 하나의 그룹은 안테나들 (104 및 106) 을 포함할 수도 있고, 또 다른 그룹은 안테나들 (108 및 110) 을 포함할 수도 있고, 추가적인 그룹은 안테나들 (112 및 114) 을 포함할 수도 있다. 도 1 에서는, 오직 2 개의 안테나들이 각각의 안테나 그룹에 대하여 도시되지만, 그러나, 더 많거나 더 적은 안테나들이 각각의 안테나 그룹에 대하여 사용될 수도 있다. 무선 노드 (116) 는 안테나들 (112 및 114) 과 통신할 수도 있고, 여기서, 안테나들 (112 및 114) 은 정보를 순방향 링크 (120) 상에서 무선 노드 (116) 로 송신하고, 역방향 링크 (118) 상에서 무선 노드 (116) 로부터 정보를 수신한다. 무선 노드 (122) 는 안테나들 (106 및 108) 과 통신할 수도 있고, 여기서, 안테나들 (106 및 108) 은 정보를 순방향 링크 (126) 상에서 무선 노드 (122) 로 송신하고, 역방향 링크 (124) 상에서 무선 노드 (122) 로부터 정보를 수신한다. 노드 B (102) 는 또한, 예를 들어, 만물 인터넷 (Internet-of-Everything; IoE) 디바이스들일 수도 있는 다른 무선 노드들과 통신할 수도 있다. IoE 디바이스 (136) 는 노드 B (102) 의 하나 이상의 다른 안테나들과 통신할 수도 있고, 여기서, 안테나들은 정보를 순방향 링크 (140) 상에서 IoE 디바이스 (136) 로 송신하고, 역방향 링크 (138) 상에서 IoE 디바이스 (136) 로부터 정보를 수신한다. IoE 디바이스 (142) 는 노드 B (102) 의 하나 이상의 다른 안테나들과 통신할 수도 있고, 여기서, 안테나들은 정보를 순방향 링크 (146) 상에서 IoE 디바이스 (142) 로 송신하고, 역방향 링크 (144) 상에서 IoE 디바이스 (142) 로부터 정보를 수신한다. 주파수 분할 듀플렉스 (Frequency Division Duplex; FDD) 시스템에서, 통신 링크들 (118, 120, 124, 126, 138, 140, 144, 및 146) 은 통신을 위하여 상이한 주파수를 이용할 수도 있다. 예를 들어, 순방향 링크 (120) 는 역방향 링크 (118) 에 의해 이용된 것과는 상이한 주파수를 이용할 수도 있고, 순방향 링크 (140) 는 역방향 링크 (138) 에 의해 이용된 것과는 상이한 주파수를 이용할 수도 있다.

[0033] 안테나들의 각각의 그룹 및/또는 안테나들이 통신하도록 설계되는 에어리어 (area) 는 노드 B 의 섹터 (sector)

로서 종종 지칭된다. 본 개시물의 하나의 양태에서, 각각의 안테나 그룹은 노드 B (102) 에 의해 커버된 어리어들의 섹터에서 무선 노드들로 통신하도록 설계될 수도 있다.

[0034] 무선 노드 (130) 는 노드 B (102) 와 통신할 수도 있고, 여기서, 노드 B (102) 로부터의 안테나들은 정보를 순방향 링크 (132) 상에서 무선 노드 (130) 로 송신하고, 역방향 링크 (134) 상에서 무선 노드 (130) 로부터 정보를 수신한다.

[0035] 순방향 링크들 (120 및 126) 상에서의 통신에서, BS (102) 의 송신 안테나들은 상이한 무선 노드들 (116, 122, 136, 및 142) 에 대한 순방향 링크들의 신호-대-잡음 비율 (signal-to-noise ratio) 을 개선시키기 위하여 빔포밍을 사용할 수도 있다. 또한, 그 커버리지 (coverage) 를 통해 무작위적으로 산재된 무선 노드들로 송신하기 위하여 빔포밍을 이용하는 노드 B 는 단일 안테나를 통해 그 모든 무선 노드들로 송신하는 노드 B 보다는, 이웃하는 셀들에 있는 무선 노드들에 대한 더 적은 간섭을 야기시킨다.

[0036] 본원에서 설명된 예들의 양태들은 LTE 기술들과 연관될 수도 있지만, 본 개시물의 양태들은 NR 과 같은 다른 무선 통신 시스템들과 함께 적용가능할 수도 있다. NR 은 업링크 및 다운링크 상에서 CP 를 갖는 직교 주파수-분할 멀티플렉싱 (orthogonal frequency-division multiplexing; OFDM) 을 사용할 수도 있고, 시간 분할 듀플렉스 (time division duplex; TDD) 를 이용한 하프-듀플렉스 (half-duplex) 동작에 대한 지원을 포함할 수도 있다. 100 MHz 의 단일 컴포넌트 캐리어 대역폭이 지원될 수도 있다. NR 리소스 블록 (resource block) 들은 0.1 ms 지속시간 상에서 75 kHz 의 서브-캐리어 (sub-carrier) 대역폭을 갖는 12 개의 서브-캐리어들에 걸쳐 있을 수도 있다. 각각의 라디오 프레임 (radio frame) 은 10 ms 의 길이를 갖는 50 개의 서브프레임 (subframe) 들로 구성될 수도 있다. 결과적으로, 각각의 서브프레임은 0.2 ms 의 길이를 가질 수도 있다. 각각의 서브프레임은 데이터 송신을 위한 링크 방향 (즉, 다운링크 (DL) 또는 업링크 (UL)) 을 표시할 수도 있고, 각각의 서브프레임에 대한 링크 방향은 동적으로 스위칭될 수도 있다. 각각의 서브프레임은 DL/UL 데이터 뿐만 아니라, DL/UL 제어 데이터를 포함할 수도 있다. 빔포밍이 지원될 수도 있고, 빔 방향은 동적으로 구성될 수도 있다. 프리코딩 (precoding) 을 갖는 MIMO 송신들이 또한 지원될 수도 있다. DL 에서의 MIMO 구성들은 8 개에 이르는 스트림들을 갖는 멀티-계층 DL 송신들로 8 개에 이르는 송신 안테나들을 지원할 수도 있다. UE 당 2 개에 이르는 스트림들을 갖는 멀티-계층 송신들이 지원될 수도 있다. 다수의 셀들의 어그리게이션 (aggregation) 은 8 개에 이르는 서빙 셀 (serving cell) 들로 지원될 수도 있다. 대안적으로, NR 은 OFDM-기반 무선 인터페이스와 다른, 상이한 무선 인터페이스를 지원할 수도 있다. NR 네트워크들은 이러한 중앙 유닛들 또는 분산된 유닛들과 같은 엔티티 (entity) 들을 포함할 수도 있다.

[0037] 도 2 는 본 개시물의 양태들이 실시될 수도 있는 다중-입력 다중-출력 (MIMO) 시스템 (200) 에서의 송신기 시스템 (210) (예컨대, 기지국으로서 또한 알려짐) 및 수신기 시스템 (250) (예컨대, 무선 노드로서 또한 알려짐) 의 양태의 블록도를 예시한다. 시스템 (210) 및 시스템 (250) 의 각각은 송신뿐만 아니라 수신하기 위한 능력들을 가진다. 시스템 (210) 또는 시스템 (250) 이 송신하고 있거나, 수신하고 있고, 동시에 송신하고 수신하고 있는지 여부는 애플리케이션에 의존한다. 송신기 시스템 (210) 에서, 다수의 데이터 스트림들에 대한 트래픽 데이터는 데이터 소스 (212) 로부터 송신 (TX) 데이터 프로세서 (214) 로 제공된다.

[0038] 본 개시물의 하나의 양태에서, 각각의 데이터 스트림은 각각의 송신 안테나 상에서 송신될 수도 있다. TX 데이터 프로세서 (214) 는 코딩된 데이터를 제공하기 위하여 그 데이터 스트림에 대하여 선택된 특정한 코딩 방식 (예컨대, 저밀도 패리티 체크 (LDPC)) 에 기초하여 각각의 데이터 스트림에 대한 트래픽 데이터를 포맷팅하고, 코딩하고, 인터리빙 (interleaving) 한다.

[0039] 각각의 데이터 스트림에 대한 코딩된 데이터는 OFDM 기법들을 이용하여 파일럿 데이터 (pilot data) 와 멀티플렉싱될 수도 있다. 파일럿 데이터는 전형적으로, 알려진 방식으로 프로세싱되는 알려진 데이터 패턴이고, 채널 응답을 추정하기 위하여 수신기 시스템에서 이용될 수도 있다. 각각의 데이터 스트림에 대한 멀티플렉싱된 파일럿 및 코딩된 데이터는 그 다음으로, 변조 심볼 (modulation symbol) 들을 제공하기 위하여 그 데이터 스트림에 대하여 선택된 특정한 변조 방식 (예컨대, BPSK, QSPK, M-PSK, 또는 M-QAM) 에 기초하여 변조 (예컨대, 심볼 맵핑 (symbol mapping)) 된다. 각각의 데이터 스트림에 대한 데이터 레이트, 코딩, 및 변조는 프로세서 (230) 에 의해 수행된 명령들에 의해 결정될 수도 있다. 메모리 (232) 는 송신기 시스템 (210) 을 위한 데이터 및 소프트웨어/펌웨어를 저장할 수도 있다.

[0040] 모든 데이터 스트림들에 대한 변조 심볼들은 그 다음으로, (예컨대, OFDM 을 위하여) 변조 심볼들을 추가로 프로세싱할 수도 있는 TX MIMO 프로세서 (220) 에 제공된다. TX MIMO 프로세서 (220) 는 그 다음으로, N_T (예

컨대, 여기서, N_T 는 양수 정수임) 개의 변조 심볼 스트림들을 N_T 개의 송신기들 (TMTR) (222a 내지 222t) 에 제공한다. 본 개시물의 특정 양태들에서, TX MIMO 프로세서 (220) 는 빔포밍 가중치들을 데이터 스트림들의 심볼들과, 심볼이 송신되고 있는 안테나에 적용한다.

- [0041] 각각의 송신기 (222) 는 하나 이상의 아날로그 신호들을 제공하기 위하여 각각의 심볼 스트림을 수신 및 프로세싱하고, MIMO 채널 상에서의 송신을 위하여 적당한 변조된 신호를 제공하기 위하여 아날로그 신호들을 추가로 컨디셔닝 (예컨대, 증폭, 필터링, 및 업컨버팅 (upconverting)) 한다. 송신기들 (222a 내지 222t) 로부터의 N_T 개의 변조된 신호들은 그 다음으로, 각각 N_T 개의 안테나들 (224a 내지 224t) 로부터 송신된다.
- [0042] 수신기 시스템 (250) 에서, 송신된 변조된 신호들은 N_R (예컨대, 여기서, N_R 은 양수 정수임) 개의 안테나들 (252a 내지 252r) 에 의해 수신될 수도 있고, 각각의 안테나 (252) 로부터의 수신된 신호는 각각의 수신기 (RCVR) (254a 내지 254r) 에 제공될 수도 있다. 각각의 수신기 (254) 는 각각의 수신된 신호를 컨디셔닝 (예컨대, 필터링, 증폭, 및 다운컨버팅 (downconverting)) 할 수도 있고, 샘플들을 제공하기 위하여 컨디셔닝된 신호를 디지털화할 수도 있고, 대응하는 "수신된" 심볼 스트림을 제공하기 위하여 샘플들을 추가로 프로세싱할 수도 있다.
- [0043] 수신 (RX) 데이터 프로세서 (260) 는 그 다음으로, N_T 개의 "검출된" 심볼 스트림들을 제공하기 위한 특정한 수신기 프로세싱 기법에 기초하여 N_R 개의 수신기들 (254) 로부터 N_R 개의 수신된 심볼 스트림들을 수신 및 프로세싱한다. RX 데이터 프로세서 (260) 는 그 다음으로, 데이터 스트림에 대한 트래픽 데이터를 복원하기 위하여 각각의 검출된 심볼 스트림을 복조하고, 디인터리빙 (deinterleaving) 하고, 디코딩한다. RX 데이터 프로세서 (260) 에 의한 프로세싱은 송신기 시스템 (210) 에서의 TX MIMO 프로세서 (220) 및 TX 데이터 프로세서 (214) 에 의해 수행된 것과 상보적일 수도 있다.
- [0044] 프로세서 (270) 는 어느 프리-코딩 행렬 (pre-coding matrix) 을 이용할 것인지를 주기적으로 결정한다. 프로세서 (270) 는 행렬 인덱스 (matrix index) 부분 및 등급 값 (rank value) 부분을 포함하는 역방향 링크 메시지를 공식화 (formulate) 한다. 메모리 (272) 는 수신기 시스템 (250) 을 위한 데이터 및 소프트웨어/펌웨어를 저장할 수도 있다. 역방향 링크 메시지는 통신 링크 및/또는 수신된 데이터 스트림에 관한 다양한 타입들의 정보를 포함할 수도 있다. 역방향 링크 메시지는 그 다음으로, 데이터 소스 (236) 로부터 다수의 데이터 스트림들에 대한 트래픽 데이터를 또한 수신하는 TX 데이터 프로세서 (238) 에 의해 프로세싱되고, 변조기 (280) 에 의해 변조되고, 송신기들 (254a 내지 254r) 에 의해 컨디셔닝되고, 송신기 시스템 (210) 으로 다시 송신된다.
- [0045] 송신기 시스템 (210) 에서, 수신기 시스템 (250) 으로부터의 변조된 신호들은 안테나들 (224) 에 의해 수신되고, 수신기들 (222) 에 의해 컨디셔닝되고, 복조기 (240) 에 의해 복조되고, 수신기 시스템 (250) 에 의해 송신된 역방향 링크 메시지를 추출하기 위하여 RX 데이터 프로세서 (242) 에 의해 프로세싱된다. 프로세서 (230) 는 그 다음으로, 빔포밍 가중치들을 결정하기 위하여 어느 프리-코딩 행렬을 이용할 것인지를 결정하고, 그 다음으로, 추출된 메시지를 프로세싱한다.
- [0046] 수신기 시스템 (250) 의 프로세서 (270), RX 데이터 프로세서 (260), TX 데이터 프로세서 (238), 또는 다른 프로세서들/엘리먼트들, 또는 그 조합 중의 임의의 하나, 및/또는 송신기 시스템 (210) 의 프로세서 (230), TX MIMO 프로세서 (220), TX 데이터 프로세서 (214), RX 데이터 프로세서 (242), 또는 다른 프로세서들/엘리먼트들, 또는 그 조합 중의 임의의 하나는 이하에서 논의된 본 개시물의 특정 양태들에 따라, 무접속 액세스를 위한 절차들을 수행하도록 구성될 수도 있다. 양태에서, 프로세서 (270), RX 데이터 프로세서 (260), 및 TX 데이터 프로세서 (238) 중의 적어도 하나는 본원에서 설명된 무접속 액세스를 위한 랜덤 액세스 채널 (random-access channel; RACH) 절차들을 수행하기 위하여 메모리 (272) 내에 저장된 알고리즘들을 실행하도록 구성될 수도 있다. 또 다른 양태에서, 프로세서 (230), TX MIMO 프로세서 (220), TX 데이터 프로세서 (214), 및 RX 데이터 프로세서 (242) 중의 적어도 하나는 본원에서 설명된 무접속 액세스를 위한 랜덤 액세스 채널 (RACH) 절차들을 수행하기 위하여 메모리 (232) 내에 저장된 알고리즘들을 실행하도록 구성될 수도 있다.
- [0047] 도 3 은 도 1 에서 예시된 통신 네트워크 (100) 내에서 채용될 수도 있는 무선 디바이스 (302) 에서 사용될 수도 있는 다양한 컴포넌트들을 예시한다. 무선 디바이스 (302) 는 본원에서 설명된 다양한 방법들을 구현하도록 구성될 수도 있는 디바이스의 예이다. 무선 디바이스 (302) 는 노드 B (102) (예컨대, TRP), 또는 무선 노드들 (예컨대, 무선 노드들 (116, 122, 130), 또는 IoT 디바이스 (136 또는 142)) 중의 임의의 것일 수도

있다. 예를 들어, 무선 디바이스 (302) 는 도 12 에서 설명된 동작들 (1200) 뿐만 아니라, 본원에서 설명된 다른 동작들을 수행하도록 구성될 수도 있다.

[0048] 무선 디바이스 (302) 는 무선 디바이스 (302) 의 동작을 제어하는 프로세서 (304) 를 포함할 수도 있다. 프로세서 (304) 는 또한, 중앙 프로세싱 유닛 (central processing unit; CPU) 으로서 지칭될 수도 있다. 판독-전용 메모리 (read-only memory; ROM) 및 랜덤 액세스 메모리 (random access memory; RAM) 양자 모두를 포함할 수도 있는 메모리 (306) 는 명령들 및 데이터를 프로세서 (304) 에 제공한다. 메모리 (306) 의 부분은 또한, 비-휘발성 랜덤 액세스 메모리 (non-volatile random access memory; NVRAM) 를 포함할 수도 있다. 프로세서 (304) 는 전형적으로, 메모리 (306) 내에서 저장된 프로그램 명령들에 기초하여 논리적 또는 산술적 동작들을 수행한다. 메모리 (306) 에서의 명령들은 예를 들어, UE 가 무접속 액세스 동안에 데이터를 효율적으로 송신하는 것을 허용하기 위하여, 본원에서 설명된 방법들을 구현하도록 실행가능할 수도 있다. 프로세서 (304) 의 일부 비-제한적인 예들은 스냅드래곤 (Snapdragon) 프로세서, 주문형 집적 회로 (application specific integrated circuit; ASIC) 들, 프로그래밍가능 로직 등을 포함할 수도 있다.

[0049] 무선 디바이스 (302) 는 또한, 무선 디바이스 (302) 와 원격 위치 사이의 데이터의 송신 및 수신을 허용하기 위하여 송신기 (301) 및 수신기 (312) 를 포함할 수도 있는 하우징 (308) 을 포함할 수도 있다. 송신기 (310) 및 수신기 (312) 는 트랜시버 (314) 로 조합될 수도 있다. 단일의 또는 복수의 송신 안테나들 (316) 은 하우징 (308) 에 부착될 수도 있고, 트랜시버 (314) 에 전기적으로 결합될 수도 있다. 무선 디바이스 (302) 는 또한, (도시되지 않은) 다수의 송신기들, 다수의 수신기들, 및 다수의 트랜시버들을 포함할 수도 있다. 무선 디바이스 (302) 는 또한, 무선 배터리 충전 장비를 포함할 수 있다.

[0050] 무선 디바이스 (302) 는 또한, 트랜시버 (314) 에 의해 수신된 신호들의 레벨을 검출하고 정량화 (quantify) 하기 위한 노력에서 이용될 수도 있는 신호 검출기 (318) 를 포함할 수도 있다. 신호 검출기 (318) 는 이러한 신호들을 총 에너지, 심볼 당 서브캐리어 당 에너지, 전력 스펙트럼 밀도, 및 다른 신호들로서 검출할 수도 있다. 무선 디바이스 (302) 는 또한, 신호들을 프로세싱할 시의 이용을 위한 디지털 신호 프로세서 (digital signal processor; DSP) (320) 를 포함할 수도 있다.

[0051] 추가적으로, 무선 디바이스는 또한, 송신을 위하여 신호들을 인코딩할 시의 이용을 위한 인코더 (322), 및 수신된 신호들을 디코딩할 시의 이용을 위한 디코더 (324) 를 포함할 수도 있다. 특정 양태들에 따르면, 디코더 (324) 는 (예컨대, 도 12 에서 예시된 동작들 (1200) 을 구현함으로써) 본원에서 제시된 특정 양태들에 따라 디코딩을 수행할 수도 있다.

[0052] 무선 디바이스 (302) 의 다양한 컴포넌트들은 데이터 버스에 추가적으로, 전력 버스, 제어 신호 버스, 및 스테이터스 (status) 신호를 포함할 수도 있는 버스 시스템 (326) 에 의해 함께 결합될 수도 있다. 프로세서 (304) 는 이하에서 논의된 본 개시물의 양태들에 따라, 무접속 액세스를 수행하기 위하여 메모리 (306) 내에 저장된 명령들을 액세스하도록 구성될 수도 있다.

[0053] 일 예의 에러 정정 코딩

[0054] 많은 통신 시스템들은 에러-정정 코드들을 이용한다. 구체적으로, 에러-정정 코드들은 중복성을 데이터 스트림 내로 도입함으로써 이 시스템들에서의 정보 전송의 내재적인 비신뢰성 (intrinsic unreliability) 을 보상한다. 저밀도 패리티 체크 (LDPC) 코드들은 반복적인 코딩 시스템을 이용하는 특정한 타입의 에러 정정 코드들이다. 특히, 갤러거 코드 (Gallager code) 들은 규칙적 LDPC 코드들의 초기의 예이다. LDPC 코드들은 그 패리티 체크 행렬 H 의 엘리먼트들의 대부분이 '0' 으로 설정되는 선형 블록 코드 (linear block code) 들이다.

[0055] LDPC 코드들은 이분 그래프 (bipartite graph) 들 ("태너 그래프 (Tanner graph) 들" 로서 종종 지칭됨) 에 의해 표현될 수 있고, 여기서, 변수 노드들의 세트는 코드 워드의 비트들 (예컨대, 정보 비트들 또는 시스템 비트들) 에 대응하고, 체크 노드들의 세트는 코드를 정의하는 패리티-체크 제약 (parity-check constraint) 들의 세트에 대응한다. 그래프에서의 예지들은 변수 노드들을 체크 노드들에 접속한다. 이에 따라, 그래프의 노드들은 2 개의 상이한 타입들의 노드들을 접속하는 예지들을 갖는 2 개의 별개의 세트들, 변수 노드들 및 체크 노드들로 분리된다.

[0056] 리프팅된 그래프는 사진으로서 또한 알려질 수도 있는 이분 기본 그래프 (G) 를 다수의 횡수들 Z 만큼 복사함으로써 생성된다. 변수 노드 및 체크 노드는 그것들이 그래프에서 "에지 (edge)" (즉, 변수 노드 및 체크 노드를 접속하는 라인) 에 의해 접속될 경우에 "이웃 (neighbor) 들" 로 고려될 수도 있다. 추가적으로, 이분

기본 그래프 (G) 의 각각의 에지 (e) 에 대하여, 치환 (permutation) 은 G 의 Z 개의 복사본들을 상호접속하기 위하여 에지 (e) 의 Z 개의 복사본들에 적용된다. 변수 노드 시퀀스 (variable node sequence) 와의 일대일 (one-to-one) 연관성을 가지는 비트 시퀀스는, 각각의 체크 노드에 대하여, 모든 이웃하는 변수 노드들과 연관된 비트들이 제로 모듈로 2 (zero modulo 2) (즉, 그것들은 짝수 개의 1 들을 포함함) 로 합산될 경우에, 그리고 그러한 경우에만 유효한 코드워드 (codeword) 이다. 이용된 치환들이 순환적 (cyclic) 일 경우에 결과적인 LDPC 코드는 준순환 (quasi-cyclic; QC) 일 수도 있다.

[0057] 도 4 내지 도 4a 는 본 개시물의 특정 양태들에 따라, 예시적인 LDPC 코드의 그래픽 및 행렬 표현들을 도시한다. 예를 들어, 도 4 는 예시적인 LDPC 코드를 표현하는 이분 그래프 (400) 를 도시한다. 이분 그래프 (400) 는 4 개의 체크 노드들 (420) (정사각형들에 의해 표현됨) 에 접속된 5 개의 변수 노드들 (410) (원들에 의해 표현됨) 의 세트를 포함한다. 그래프 (400) 에서의 에지들은 변수 노드들 (410) 을 체크 노드들 (420) 에 접속한다 (변수 노드들 (410) 을 체크 노드들 (420) 에 접속하는 라인들에 의해 표현됨). 이 그래프는 $|E| = 12$ 개의 에지들에 의해 접속된, $|V| = 5$ 개의 변수 노드들 및 $|C| = 4$ 개의 체크 노드들로 구성된다.

[0058] 이분 그래프는 패리티 체크 행렬로서 또한 알려질 수도 있는 단순화된 인접 행렬 (adjacency matrix) 에 의해 표현될 수도 있다. 도 4a 는 이분 그래프 (400) 의 행렬 표현 (400A) 을 도시한다. 행렬 표현 (400A) 은 패리티 체크 행렬 H 및 코드 워드 벡터 x 를 포함하고, 여기서, x_1 내지 x_5 는 코드 워드 x 의 비트들을 표현한다. 패리티 행렬 H 는 수신된 신호가 정상적으로 디코딩되었는지 여부를 결정하기 위하여 이용된다. 패리티 체크 행렬 H 는 j 개의 체크 노드들에 대응하는 C 개의 행 (row) 들 및 i 개의 변수 노드들 (즉, 복조된 심볼) 에 대응하는 V 개의 열 (column) 들을 가지고, 여기서, 행들은 수학적식들을 표현하고, 열들은 코드 워드의 비트들을 표현한다. 도 4a 에서, 행렬 H 는 4 개의 체크 노드들 및 5 개의 변수 노드들에 각각 대응하는 4 개의 행들 및 5 개의 열들을 가진다. j 번째 체크 노드가 에지에 의해 i 번째 변수 노드에 접속될 경우, 즉, 2 개의 노드들이 이웃들일 경우, 패리티 체크 행렬 H 의 i 번째 열 및 j 번째 행에는 1 이 있다. 즉, i 번째 행 및 j 번째 열의 교차점은 에지가 대응하는 정점들을 합치는 "1" 및 에지가 없는 "0" 을 포함한다. 코드 워드 벡터 x 는 $Hx = 0$ 일 경우에 (예컨대, 각각의 제약 노드에 대하여, (변수 노드들과의 그 연관성을 통해) 제약과 이웃하는 비트들이 제로 모듈로 2 로 합산되고, 즉, 그것들이 짝수 개의 1 들을 포함할 경우에), 그리고 그러한 경우에만 유효한 코드 워드를 표현한다. 이에 따라, 코드 워드가 올바르게 수신될 경우에, $Hx = 0 \pmod{2}$ 이다. 코딩된 수신된 신호 및 패리티 체크 행렬 H 의 곱 (product) 이 '0' 이 될 때, 이것은 에러가 발생하지 않았다는 것을 나타낸다. 패리티 체크 행렬은 C 행 X V 열 2진 행렬이다. 행들은 수학적식들을 표현하고, 열들은 코드 워드에서의 디지털 (digit) 들을 표현한다.

[0059] 복조된 심볼들 또는 변수 노드들의 수는 LDPC 코드 길이이다. 행 (열) 에서의 비-제로 엘리먼트들의 수는 행 (열) 가중치 dc (dv) 로서 정의된다.

[0060] 노드의 차수 (degree) 는 그 노드에 접속된 에지들의 수를 지칭한다. 이 특징은 변수 노드 (410) 에 들어오는 에지들의 수가 대응하는 열에서의 1 들의 수와 동일한 도 4a 에서 도시된 H 행렬에서 예시되고, 변수 노드 차수 d(v) 로 칭해진다. 유사하게, 체크 노드 (420) 와 접속된 에지들의 수는 대응하는 행에서의 1 들의 수와 동일하고, 체크 노드 차수 d(c) 로 칭해진다.

[0061] 규칙적 그래프 또는 코드는 모든 변수 노드들이 동일한 차수 j 를 가지고 모든 제약 노드들이 동일한 차수 k 를 가지는 것이다. 이 경우에, 코드가 (j, k) 규칙적 코드이라고 한다. 다른 한편으로, 불규칙적 코드는 상이한 차수들의 제약 노드들 및/또는 변수 노드들을 가진다. 예를 들어, 일부 변수 노드들은 차수 4 일 수도 있고, 다른 것들은 차수 3 일 수도 있고, 또 다른 것들은 차수 2 일 수도 있다.

[0062] "리프팅" 은 LDPC 코드들이 큰 LDPC 코드들과 전형적으로 연관된 복잡도를 또한 감소시키면서, 병렬 인코딩 및/또는 디코딩 구현예들을 이용하여 구현되는 것을 가능하게 한다. 리프팅은 상대적으로 간결한 설명을 여전히 가지면서, LDPC 디코더들의 효율적인 병렬화를 가능하게 하는 것을 돕는다. 더 구체적으로, 리프팅은 더 작은 기본 코드의 다수의 복사본들로부터 상대적으로 큰 LDPC 코드를 생성하기 위한 기법이다. 예를 들어, 리프팅된 LDPC 코드는 기본 그래프 (예컨대, 사진) 의 Z 개의 병렬 복사본들을 생성함으로써, 그리고 그 다음으로, 기본 그래프의 각각의 복사본의 에지 번들 (edge bundle) 들의 치환들을 통해 병렬 복사본들을 상호접속함으로써 생성될 수도 있다. 기본 그래프는 코드의 (매크로 (macro)) 구조를 정의하고, 다수 (K) 의 정보 비트-열들 및 다수 (N) 의 코드 비트 열들로 구성된다. 기본 그래프를 횡수 (Z) 만큼 리프팅하는 것은 KZ 의 최종적인 블록 길이로 귀착된다.

- [0063] 이에 따라, 기본 그래프의 다수의 복사본들이 만들어지고 단일의 리프팅된 그래프를 형성하기 위하여 접속되는 "복사 및 치환" 동작에 의해, 더 큰 그래프가 획득될 수 있다. 다수의 복사본들에 대하여, 단일의 기본 에지의 복사본들의 세트인 같은 에지 (like edge) 들은 치환되고, 기본 그래프보다 Z 배 더 큰 접속된 그래프를 형성하기 위하여 접속된다.
- [0064] 도 5 는 도 4 의 그래프의 3 개의 복사본들을 만드는 효과를 그래픽으로 예시한다. 3 개의 복사본들은 복사본들 사이에서 같은 에지들을 치환함으로써 상호접속될 수도 있다. 치환들이 순환 치환들로 한정될 경우, 결과적인 그래프는 리프팅 Z = 3 을 갖는 준순환 LDPC 에 대응한다. 3 개의 복사본들이 만들어지는 원래의 그래프는 기본 그래프로서 본원에서 지칭된다. 상이한 크기들의 유도된 그래프들을 획득하기 위하여, "복사 및 치환" 동작을 기본 그래프에 적용할 수 있다.
- [0065] 리프팅된 그래프의 대응하는 패리티 체크 행렬은 기본 패리티 체크 행렬에서의 각각의 엔트리 (entry) 를 ZxZ 행렬로 대체함으로써 기본 그래프의 패리티 체크 행렬로부터 구축될 수 있다. 0 엔트리들 (기본 에지들을 가지지 않는 것들) 은 0 행렬로 대체되고, 1 엔트리들 (기본 에지들을 표시함) 은 ZxZ 치환 행렬로 대체된다. 순환 리프팅들의 경우에, 치환들은 순환 치환들이다.
- [0066] 순환적으로 리프팅된 LDPC 코드는 또한, 2진 다항식들 모듈로 x^Z+1 의 고리 상의 코드로서 해독될 수 있다. 이 해독에서, 2진 다항식, $(x) = b_0 + b_1x + b_2x^2 + \dots + b_{Z-1}x^{Z-1}$ 은 기본 그래프에서의 각각의 변수 노드에 연관될 수도 있다. 2진 벡터 $(b_0, b_1, b_2, \dots, b_{Z-1})$ 는 리프팅된 그래프에서의 Z 개의 대응하는 변수 노드들, 즉, 단일의 기본 변수 노드의 Z 개의 복사본들에 연관된 비트들에 대응한다. 2진 벡터의 k 에 의한 순환 치환은 대응하는 2진 다항식을 x^k 에 의해 승산함으로써 달성되고, 여기서, 승산은 모듈로 $x^Z + 1$ 로 취해진다. 기본 그래프에서의 d 패리티 체크 차수는 $x^{k_1}B_1(x) + x^{k_2}B_2(x) + \dots + x^{k_d}B_d(x)=0$ 로서 기록된 이웃하는 2진 다항식들 $B_1(x), \dots, B_d(x)$ 에 대한 선형 제약으로서 해독될 수 있고, 여기서, 값들 k_1, \dots, k_d 은 대응하는 에지들에 연관된 순환 리프팅 값들이다.
- [0067] 이 결과적인 수학식은 기본 그래프에서의 단일 연관된 패리티 체크에 대응하는 순환 리프팅된 태너 그래프에서 Z 개의 패리티 체크들과 동등하다. 이에 따라, 리프팅된 그래프에 대한 패리티 체크 행렬은, 1 엔트리들이 형태 x^k 의 다항식들로 대체되고 0 엔트리들이 0 으로서 리프팅되지만, 이제 0 은 0 2진 다항식 모듈로 $x^Z + 1$ 로서 해독되는 기본 그래프에 대한 행렬을 이용하여 표현될 수 있다. 이러한 행렬은 x^k 대신에 값 k 를 부여함으로써 쓰여질 수도 있다. 이 경우에, 0 다항식은 그것을 x^0 로부터 구별하기 위하여, 때때로 -1 로서, 그리고 때때로 또 다른 문자로서 표현된다.
- [0068] 전형적으로, 패리티 체크 행렬의 정사각형 하위행렬 (submatrix) 은 코드의 패리티 비트들을 표현한다. 상보적인 열들은 인코딩 시에, 인코딩되어야 할 정보 비트들과 동일하게 설정되는 정보 비트들에 대응한다. 인코딩은 패리티 체크 수학식들을 만족시키기 위하여 진술된 정사각형 하위행렬에서의 변수들에 대하여 구함으로써 달성될 수도 있다. 패리티 체크 행렬 H 는 2 개의 파트들 M 및 N 으로 파티셔닝될 수도 있고, 여기서, M 은 정사각형 부분이다. 이에 따라, 인코딩은 c 및 d 가 x 를 포함하는 $Mc = s = Nd$ 를 푸는 것으로 줄어든다. 준순환 코드들 또는 순환 리프팅된 코드들의 경우에, 위의 대수학은 2진 다항식들 모듈로 $x^Z + 1$ 의 고리 상에 있는 것으로서 해독될 수 있다. 준순환인 802.11 LDPC 코드들의 경우에, 인코딩 하위행렬 M 은 도 6 에서 도시된 바와 같은 정수 표현을 가진다.
- [0069] 수신된 LDPC 코드 워드는 원래의 코드 워드의 재구성된 버전을 생성하기 위하여 디코딩될 수 있다. 에러들의 부재 시에, 또는 정정가능한 에러들의 경우에, 디코딩은 인코딩되었던 원래의 데이터 유닛을 복원하기 위하여 이용될 수 있다. 중복적인 비트들은 비트 에러들을 검출하고 이를 정정하기 위하여 디코더들에 의해 이용될 수도 있다. LDPC 디코더 (들) 는 일반적으로, 국소적 계산들을 반복적으로 수행함으로써, 그리고 에지들을 따라, 이분 그래프 (400) 내에서 메시지들을 교환하는 것에 의해 그 결과들을 전달함으로써, 그리고 들어오는 메시지들에 기초하여 노드들에서의 연산들을 수행하는 것에 의해 이 메시지들을 업데이트함으로써 동작한다. 이 단계들은 전형적으로 여러 번 반복될 수도 있고, 메시지 전달 단계들로서 지칭될 수도 있다. 예를 들어, 그래프 (400) 에서의 각각의 변수 노드 (410) 는 초기에, 통신 채널로부터의 관측들에 의해 결정된 바와 같은 연관된 비트의 값의 추정치를 표시하는 "소프트 비트" (예컨대, 코드 워드의 수신된 비트를 표현함) 를

제공받을 수도 있다. 이 소프트 비트들을 이용하면, LDPC 디코더들은 메모리로부터 그것들 또는 그 일부 부분을 반복적으로 판독함으로써, 그리고 업데이트된 메시지 또는 그 일부 부분을 다시 메모리에 기록함으로써 메시지들을 업데이트할 수도 있다. 업데이트 동작들은 전형적으로 대응하는 LDPC 코드의 패리티 체크 제약들에 기초한다. 리프팅된 LDPC 코드들의 구현예들에서, 같은 에지들 상의 메시지들은 종종 병렬로 프로세싱된다.

[0070] 고속 애플리케이션들을 위하여 설계된 LDPC 코드들은 인코딩 및 디코딩 동작들에서의 높은 병렬화를 지원하기 위하여 큰 리프팅 인자들 및 상대적으로 작은 기본 그래프들을 갖는 준순환 구성들을 종종 이용한다. 더 높은 코드 레이트들 (예컨대, 코드 워드 길이에 대한 메시지 길이의 비율) 을 갖는 LDPC 코드들은 상대적으로 더 적은 패리티 체크들을 가지는 경향이 있다. 기본 패리티 체크들의 수가 변수 노드의 차수 (예컨대, 변수 노드에 접속된 에지들의 수) 보다 더 작을 경우, 기본 그래프에서는, 그 변수 노드는 2 개 이상의 에지들에 의해 기본 패리티 체크들 중의 적어도 하나에 접속된다 (예컨대, 변수 노드는 "이중 에지" 를 가질 수도 있음). 또는, 기본 패리티 체크들의 수가 변수 노드의 차수 (예컨대, 변수 노드에 접속된 에지들의 수) 보다 더 작을 경우, 기본 그래프에서는, 그 변수 노드가 2 개 이상의 에지들에 의해 기본 패리티 체크들 중의 적어도 하나에 접속된다. 2 개 이상의 에지들에 의해 접속된 기본 변수 노드 및 기본 체크 노드를 가지는 것은 일반적으로, 병렬 하드웨어 구현 목적을 위하여 바람직하지 않다. 예를 들어, 이러한 이중 에지들은 동일한 메모리 위치들로의 다수의 동시 판독 및 기록 동작들로 귀착될 수도 있고, 이것은 궁극적으로 데이터 코히어런스 (data coherency) 문제들을 생성할 수도 있다. 기본 LDPC 코드에서의 이중 에지는 단일 병렬 패리티 체크 업데이트 동안에 동일한 소프트 비트 값 메모리 위치의 병렬 판독을 2 번 트리거링할 수도 있다. 이에 따라, 추가적인 회로부는 양자의 업데이트들을 적절하게 편입시키기 위하여, 메모리로 다시 기록되는 소프트 비트 값들을 조합할 필요가 있다. 그러나, LDPC 코드에서 이중 에지들을 제거하는 것은 이 여분의 복잡도를 회피하는 것을 돕는다.

[0071] 순환 리프팅에 기초한 LDPC 코드 설계들은 2진 다항식들 모듈로 x^Z+1 일 수도 있는 다항식들 모듈로의 고리 상의 코드들로서 해독될 수 있고, 여기서, Z 은 리프팅 크기 (예컨대, 준순환 코드에서의 사이클의 크기) 이다. 이에 따라, 이러한 코드들을 인코딩하는 것은 이 링에서의 대수학 동작으로서 종종 해독될 수 있다.

[0072] 표준적인 불규칙적 LDPC 코드 앙상블 (code ensemble) 들 (차수 분포들) 의 정의에서는, 태너 그래프 표현에서의 모든 에지들이 통계적으로 상호 교환가능할 수도 있다. 다시 말해서, 에지들의 단일의 통계적인 등가 클래스 (statistical equivalence class) 가 존재한다. 리프팅된 LDPC 코드들의 더 상세한 논의는 예를 들어, Tom Richardson 및 Ruediger Urbanke 에 의해, 2008 년 3 월 17 일자로 출판된 "Modern Coding Theory" 라는 명칭의 서적에서 발견될 수도 있다. 멀티-에지 LDPC 코드들에 대하여, 에지들의 다수의 등가 클래스들이 가능할 수도 있다. 표준적인 불규칙적 LDPC 앙상블 정의에서는, 그래프에서의 노드들 (변수 및 제약 양자 모두) 이 그 차수, 즉, 노드들이 접속되는 에지들의 수에 의해 특정되지만, 멀티-에지 타입 설정에서는, 에지 차수가 벡터이고; 그것은 독립적으로 각각의 에지 등가 클래스 (타입) 로부터 노드에 접속된 에지들의 수를 특정한다. 멀티-에지 타입 앙상블은 유한한 수의 에지 타입들로 이루어진다. 제약 노드의 차수 타입은 (비-음수) 정수들의 벡터이고; 이 벡터의 i 번째 엔트리는 이러한 노드에 접속된 i 번째 타입의 소켓 (socket) 들의 수를 레코딩한다. 이 벡터는 에지 차수 (edge degree) 로서 지칭될 수도 있다. 변수 노드의 차수 타입은 그것이 (비-음수) 정수들의 벡터로서 간주될 수 있지만, 2 개의 파트들을 가진다. 제 1 파트는 수신된 분산에 관련되고, 수신된 차수로 칭해질 것이고, 제 2 파트는 에지 차수를 특정한다. 에지 차수는 제약 노드들에 대한 것과 동일한 역할을 행한다. 에지들은 그것들이 동일한 타입의 소켓들과 쌍을 이룰 때에 타입이 정해진다. 이 제약, 소켓들이 유사한 타입의 소켓들과 쌍을 이루어야 한다는 것은 멀티-에지 타입 개념을 특징짓는다. 멀티-에지 타입 설명에서, 상이한 노드 타입들은 상이한 수신된 분포들을 가질 수 있다 (예컨대, 연관된 비트들은 상이한 채널들을 거칠 수도 있음).

[0073] 도 7 은 무선 송신을 위한 인코딩된 메시지를 제공하도록 구성될 수도 있는 라디오 주파수 (RF) 모뎀 (704) 의 부분을 예시한다. 하나의 예에서, 기지국 (예컨대, 노드 B (102) 및/또는 송신기 시스템 (210)) (또는 역방향 경로 상의 무선 노드) 에서의 인코더 (706) 는 송신을 위한 메시지 (702) 를 수신한다. 메시지 (702) 는 수신 디바이스로 지향된 데이터 및/또는 인코딩된 음성 또는 다른 콘텐츠를 포함할 수도 있다. 인코더 (706) 는 기지국 또는 또 다른 네트워크 엔티티에 의해 정의된 구성에 기초하여 전형적으로 선택된 적당한 변조 및 코딩 방식 (modulation and coding scheme; MCS) 을 이용하여 메시지를 인코딩한다. 일부 경우에는, 인코더 (706) 가 (예컨대, LDPC 코드를 이용함으로써) 예를 들어, 위에서 설명된 기법들을 이용하여 메시지를 인코딩할 수도 있다. 인코더 (706) 에 의해 생성된 인코딩된 비트스트림 (708) 은 그 다음으로, 안테나

(718) 를 통한 송신을 위하여 RF 신호 (716) 를 생성하기 위하여 Tx 체인 (Tx chain) (714) 에 의해 변조되고, 증폭되고, 그리고 이와 다르게 프로세싱되는 Tx 심볼들 (712) 의 시퀀스를 생성하는 맵퍼 (mapper) (710) 에 제공될 수도 있다.

[0074] 도 8 은 인코딩된 메시지 (예컨대, 위에서 설명된 바와 같은 LDPC 코드를 이용하여 인코딩된 메시지) 를 포함하는 무선으로 송신된 신호를 수신하고 이를 디코딩하도록 구성될 수도 있는 RF 모뎀 (814) 의 부분을 예시한다.

다양한 예들에서, 신호를 수신하는 모뎀 (814) 은 무선 노드 (예컨대, 무선 노드 (116), 수신기 시스템 (250)) 에서, 기지국 (예컨대, 노드 B (102), 송신기 시스템 (210)) 에서, 또는 설명된 기능들을 수행하기 위한 임의의 다른 적당한 장치 또는 수단 (예컨대, 무선 디바이스 (302)) 에서 존재할 수도 있다. 안테나 (802) 는 무선 노드 (예컨대, 무선 노드 (116) 및/또는 수신기 시스템 (250)) 를 위한 RF 신호 (716) (즉, 도 7 에서 생성된 RF 신호 (716)) 를 수신한다. RF 체인 (804) 은 RF 신호 (716) 를 프로세싱하고 이를 복조하고, 인코딩된 메시지를 나타내는 비트스트림 (810) 을 생성하는, 디맵퍼 (demapper) (808) 에 복조된 심볼들 (806) 의 시퀀스를 제공할 수도 있다.

[0075] 디코더 (812) 는 그 다음으로, 코딩 방식 (예컨대, LDPC 코드) 을 이용하여 인코딩되었던 비트스트림으로부터 m-비트 정보 스트링 (information string) 들을 디코딩하기 위하여 이용될 수도 있다. 디코더 (812) 는 전체-병렬, 행-병렬, 또는 블록-병렬 아키텍처를 갖는 계층화된 LDPC 디코더를 포함할 수도 있다. LDPC 디코더 (들) 는 일반적으로, 국소적 계산들을 반복적으로 수행함으로써, 그리고 예지들을 따라, 이분 그래프 (400) 내에서 메시지들을 교환하는 것에 의해 그 결과들을 전달함으로써, 그리고 들어오는 메시지들에 기초하여 노드들에서의 연산들을 수행하는 것에 의해 이 메시지들을 업데이트함으로써 동작한다. 이 단계들은 전형적으로 여러 번 반복될 수도 있고, 메시지 전달 단계들로서 지칭될 수도 있다. 예를 들어, 그래프 (400) 에서의 각각의 변수 노드 (410) 는 초기에, 통신 채널로부터의 관측들에 의해 결정된 바와 같은 연관된 비트의 값의 추정치를 표시하는 "소프트 비트" (예컨대, 코드 워드의 수신된 비트를 표현함) 를 제공받을 수도 있다. "소프트 비트" 는 일부 양태들에서, 로그 (비트가 0 일 확률)/(비트가 1 일 확률) 로서 정의될 수도 있는 로그-우도 비율 (LLR) 에 의해 표현될 수도 있다. 이 LLR 들을 이용하면, LDPC 디코더들은 메모리로부터 그것들 또는 그 일부 부분을 반복적으로 판독함으로써, 그리고 업데이트된 메시지 또는 그 일부 부분을 다시 메모리에 기록함으로써 메시지들을 업데이트할 수도 있다. 업데이트 동작들은 전형적으로 대응하는 LDPC 코드의 패리티 체크 제약들에 기초한다. 리프팅된 LDPC 코드들의 구현예들에서, 유사한 예지들 상의 메시지들은 종종 병렬로 프로세싱된다. 양태들에 따르면, 이 디코딩 기법들 후에, 디코더 (812) 는 기지국 (예컨대, 노드 B (102) 및/또는 송신기 시스템 (210)) 으로부터 송신된 데이터 및/또는 인코딩된 음성 또는 다른 콘텐츠를 포함하는 메시지 (702) 를 결정하기 위하여 LLR 들에 기초하여 비트스트림 (810) 을 디코딩할 수도 있다. 디코더는 (예컨대, 도 12 에서 예시된 동작들 (1200) 을 구현함으로써) 이하에서 제시된 본 개시물의 양태들에 따라 비트스트림 (810) 을 디코딩할 수도 있다.

[0076] 일 예의 심층적-파이프라인 방식의 높은-스루풋 LDPC 디코더 아키텍처

[0077] 저밀도 패리티 체크 (LDPC) 는 무선 통신들, 스토리지 (storage), 및 이더넷 (Ethernet) 과 같은 몇몇 애플리케이션들에서 이용된 강력한 에러 정정 코딩 기술이다. LDPC 는 예를 들어, 위에서 설명되고 도 4 에서 예시된 바와 같이, 이분 그래프들 상에서 코드들을 설계하는 것에 기초한다. LDPC 디코딩은 위에서 설명된 신뢰 전파 기법 (belief propagation) 기법들을 이용하여 전형적으로 구현되고, 메시지들은 그래프의 예지들을 따라 전달되고, 그래프에서의 노드들은 소스 심볼들에 관한 판정들이 행해질 수 있는 그 주변 분포 (marginal distribution) 들을 연산한다. 준순환 (QC) 코드들은 기본 LDPC 패리티 체크 행렬 (PCM) 이 "리프팅" 되는 구조화된 LDPC 코드들의 인기 있는 클래스이다. 예를 들어, "리프팅" 은 각각의 기본 PCM 엔트리를 ZxZ 하위행렬로 대체하는 것을 수반한다. ZxZ 하위행렬은 '0' 기본 PCM 엔트리들에 대한 모든 제로들의 행렬, 또는 '1' 기본 PCM 엔트리들에 대한 순환 회전된 항등 행렬 (identity matrix) 일 수 있다. QC LDPC 코드들은 도 8 에서 예시된 디코더와 같은 디코더들이 메시지들을 교환하기 위하여 스위칭 네트워크들로 프로세싱을 Z 회 복제하는 것을 가능하게 함으로써 하드웨어에 의한 병렬 프로세싱을 가능하게 한다.

[0078] LDPC 디코더들은 반복적인 신뢰 전파 (Belief Propagation; BP) 알고리즘들을 구현하기 위하여 메시지 전달 아키텍처들을 구현한다. 로그 BP 알고리즘은 다음과 같이 쓰여질 수도 있다:

$$A_{mj} = \sum_{\substack{n \in N(m) \\ n \neq j}} \Psi(L(q_{mn})) \quad (\text{수학식 1})$$

$$s_{mj} = \prod_{\substack{n \in N(m) \\ n \neq j}} \text{Sign}(L(q_{mn})) \quad (\text{수학식 2})$$

$$R_{mj} = -s_{mj} \Psi(A_{mj}) \quad (\text{수학식 3})$$

$$L(q_j) = \sum_{m \in M(j)} R_{mj} + \frac{-2r_j}{\sigma^2} \quad (\text{수학식 4})$$

$$L(q_{mj}) = L(q_j) - R_{mj} \quad (\text{수학식 5})$$

[0079]

[0080] 여기서, $L(c)$ 는 $L(c) = \log \frac{\text{Probability } c=0}{\text{Probability } c=1}$ 으로서 정의된 로그-우도 비율이고, m 은 패리티 체크 노드 또는 PCM 행 인덱스이고, j 는 비트 노드 또는 PCM 열 인덱스이고, $N(m)$ 은 패리티 체크 노드 m 에 접속된 비트들에 대한 모든 비트 인덱스들의 세트이고, $M(j)$ 는 비트 j 에 접속된 모든 패리티 체크 노드들에 대한 모든 패리티 체크 노드 인덱스들의 세트이다. $L(q_{mj})$ 은 $\frac{-2r_j}{\sigma^2}$ 을 이용하는 것과 같이 통신 채널로부터의 관측들에 의해 결정된, 비트 LLR 들로서 또한 지칭된 코드 워드의 각각의 비트에 대한 LLR 들을 이용하여 초기화될 수도 있고, 여기서, r_j 은 페이딩 이득 (fading gain) 이고, σ^2 은 채널 분산 (channel variance) 이다.

[0081]

양태들에 따르면, 수학식 1 은 변환 Ψ 을 통해 (비트 j 에 대한 LLR 외의) 패리티 체크 노드 m 에 접속된 모든 비트들에 대한 들어오는 비트 LLR 들 $L(q_{mn})$ 을 합산하는 비트 j 에 대한 패리티 체크 매트릭 A_{mj} 을 연산한다. 이 동작은 수학식 3 과 함께, 패리티 체크 m 에 속하는 다른 비트들의 관측들에 기초하여 비트 j 에 대한 사후 LLR, R_{mj} 을 연산한다. 수학식 2 는 들어오는 비트 LLR 들 $L(q_{mn})$ 의 부호들에 기초하여 사후 LLR, R_{mj} 의 부호 s_{mj} 를 연산한다. 수학식 4 는 비트 j 에 대한 디코더로부터의 사후 LLR 들 R_{mj} (즉, 외재적인 (extrinsic) LLR 들) 의 전부를 채널로부터의 사전 LLR $\frac{-2r_j}{\sigma^2}$ (즉, 내재적인 LLR) 과 조합함으로써 업데이트된 비트 LLR 들, $L(q_j)$ 을 계산한다. 수학식 5 는 비트 LLR 합 $L(q_{mj})$ 이 다음 반복에서 업데이트된 사후/외재적인 LLR R_{mj} 의 연산을 위하여 패리티 체크 노드 m 로 다시 전달되기 전에, 비트 LLR 합 $L(q_j)$ 으로부터 패리티 체크 노드 m 에 대한 외재적인 LLR R_{mj} 을 감산한다. '플러딩 (flooding)' LDPC 디코더 반복에 대하여, 단계들 1 내지 3 (즉, 수학식들 1 내지 3 을 연산함) 은 모든 패리티 체크 노드들에 대하여 수행되고, 그 후에 비트 LLR 들 $L(q_j)$ 을 업데이트하기 위하여, 모든 비트 (변수) 노드들이 단계 4 (즉, 수학식 4 를 연산함) 를 수행한다.

[0082]

예를 들어, 본원에서 제시된 바와 같은 계층화된 LDPC 디코더들은 위의 수학식들 1 내지 5 와 유사하지만 일부

약간의 수정들을 갖는 단계들을 수행한다. 예를 들어, 계층화된 로그 BP 알고리즘은 다음처럼 쓰여질 수도 있다:

$$L(q_{mj}) = L(q_j) - R_{mj} \quad (\text{수학식 6})$$

$$A_{mj} = \sum_{\substack{n \in N(m) \\ n \neq j}} \Psi(L(q_{mn})) \quad (\text{수학식 7})$$

$$s_{mj} = \prod_{\substack{n \in N(m) \\ n \neq j}} \text{Sign}(L(q_{mn})) \quad (\text{수학식 8})$$

$$R_{mj} = -s_{mj} \Psi(A_{mj}) \quad (\text{수학식 9})$$

$$L(q_j) = L(q_{mj}) + R_{mj} \quad (\text{수학식 10})$$

[0083]

[0084]

위의 계층화된 디코딩 단계들 (즉, 수학식들 6 내지 10) 에서, 비트 LLR 들 $L(q_j)$ 은 채널 비트 LLR 들 $\frac{-2r_j}{\sigma^2}$ 로 초기화된다. 특정 양태들에 따르면, 계층화된 디코딩 (수학식들 6 내지 10) 과 플러딩 디코딩 (수학식들 1 내지 5) 사이의 핵심 차이는, 계층화된 디코딩 반복에서, 사후 LLR R_{mj} 이 수학식 9 에서 특정한 패리티 체크 노드 (PCM 행) 에 대하여 연산될 때, 비트 LLR 들 $L(q_j)$ 은 수학식들 6 내지 9 에서 다음 행의 사후 LLR 들 R_{mj} 을 연산하기 전에, 수학식 10 에서 새로운 사후 LLR 들 R_{mj} 로 즉시 업데이트된다는 것이다. 이것은 비트 LLR 들 $L(q_j)$ 의 전부가 수학식 4 에서 사후 LLR 들 R_{mj} 로 업데이트되기 전에, PCM 행들에 대응하는 사후 LLR 들 R_{mj} 의 전부가 연산 (모든 m 및 j 에 대한 수학식들 1 내지 3 루프) 되는 플러딩 디코더와 대조적이다. 그 결과, 계층화된 디코딩은 업데이트된 사후 LLR 들, R_{mj} 형태인 정보가 플러딩 디코더보다 더 빠르게 전달되는 신뢰 전파 메시지를 통해 전파하는 것을 허용하고, 이것은 더 빠른 디코더 컨버전스 (decoder convergence) 로 귀착된다.

[0085]

도 9 는 도 8 에서 예시된 디코더의 예일 수도 있는 일반적인 계층화된 LDPC 디코더 (900) 의 하이 레벨 블록도를 예시한다. 예시된 바와 같이, 계층화된 LDPC 디코더는, 궁극적으로, 사후 LLR 들 (예컨대, R_{mj}) 에 의해 업데이트되는 채널 비트 LLR 들 (예컨대, $\frac{-2r_j}{\sigma^2}$) 에 의해 초기화되는 비트 LLR 들 (예컨대, $L(q_j)$) (즉, 코드 워드의 비트 당 하나의 비트 LLR) 을 저장하기 위한 LLR 저장 메모리 (902) 를 포함한다. 계층화된 LDPC 디코더 (900) 는 또한, 사후 LLR 들을 연산하고 LLR 저장 메모리 (902) 에서의 저장된 비트 LLR 들을 업데이트하기 위하여 병렬로 동작하는 데이터 경로 프로세서들 (904) 을 포함한다. 계층화된 LDPC 디코더 (900) 는 LLR 들 (즉, 비트 LLR 들 및 사후 LLR 들) 을 메모리들 (902, 906) 과 데이터경로 프로세서들 (904) 사이에서 라우팅하기 위하여 데이터경로 프로세서들 (904) 및 치환 네트워크 (908) 에 의해 연산된 사후 LLR 들을 저장하기 위한 메트릭 저장 메모리 (metric storage memory) (906) 를 추가적으로 포함한다.

- [0086] 위에서 논의된 바와 같이, 계층화된 디코딩은 그 행에 대한 사후 LLR 들을 연산하기 위하여 PCM 에서의 행을 따라 PCM 열들 (비트 LLR 들) 을 횡단한다. 행에 대한 사후 LLR 들이 연산된 후에, 비트 LLR 들은 그것들이 다음 행에 대한 사후 LLR 들의 연산으로 공급될 때에, 그 대응하는 사후 LLR 로 각각 즉시 업데이트된다. 업데이트된 비트 LLR 의 열 인덱스가 다음 행에 접속될 경우, 업데이트된 비트 LLR 은 그 다음 행에 대한 사후 LLR 연산으로 전달된다. 접속이 없을 경우에, 업데이트된 비트 LLR 은 LLR 저장 메모리 (902) 내에 저장될 수 있다.
- [0087] 도 10 은 위에서 설명된 바와 같은 패리티 체크 행렬 (PCM) 에서의 비트 LLR 들 및 사후 LLR 들을 연산/업데이트하기 위한 이 프로세스의 예를 예시한다. 특히, PCM 의 각각의 셀은 계산된 사후 LLR 을 예시한다. 예를 들어, 도 10 에서 예시된 PCM 에 대하여, 일단 행 3 에 대한 사후 LLR 들이 연산되면, 열 5 가 열들 3 및 4 양자 모두에 접속되므로 (예컨대, PCM 엔트리들 (3, 5) 및 (4, 5) 는 비-제로임), 열 5 에 대한 비트 LLR 은 (예컨대, 수학적 10 을 이용하여) 업데이트될 수도 있고 (예컨대, 수학적들 6 내지 9 를 이용하는) 행 4 에 대한 사후 LLR 연산에서 이용될 수도 있다. 그러나, 열 6 에 대한 비트 LLR 이 행 3 으로부터 연산된 사후 LLR 로 업데이트될 때, 행 4 에 대한 사후 LLR 연산은 (4, 6) 이 비어 있을 경우에 열 6 을 포함하지 않으므로, 업데이트된 비트 LLR 은 메모리 (예컨대, LLR 저장 메모리 (902)) 내에 저장된다. 행 5 에 대한 사후 LLR 들이 연산되고 있을 때, 열 6 에 대한 비트 LLR 은 이전의 업데이트 연산으로부터 전달되는 것이 아니라, 메모리 (예컨대, LLR 저장 메모리 (902)) 로부터 관독된다. 수학적들 6 및 10 은 LLR 저장 메모리 (902) 로부터 관독될 수 있고 LLR 저장 메모리 (902) 로 기록될 수 있으므로, 기록 및 관독 충돌들이 가능하다는 것이 또한 주목되어야 한다. 이러한 충돌들은 LLR 저장 메모리 (902) 가 단지 단일의 관독 및 단일의 기록 포트를 가질 경우에, 프로세싱 파이프라인에서 "버블 (bubble) 들" (예컨대, 지연들) 을 생성할 수 있다.
- [0088] 행 (계층) 에 대한 비트 LLR 업데이트들은, 연산된 사후 LLR 들이 비트 LLR 들을 다시 업데이트하기 위하여 이용되는 다음 계층에 대한 사후 LLR 프로세싱으로 전달된 재귀적 프로세싱 (recursive processing) 으로 인해 도입된 버블들이 또한 있을 수 있다. 예를 들어, 비제로 프로세싱 파이프라인 심도가 주어질 경우, 사후 LLR 계산들이 완료될 수 있도록, 비트 LLR 업데이트 국면들 사이에는 갭 (gap) 이 있을 수도 있다.
- [0089] 예를 들어, 도 11a 는 사후 LLR 들을 연산하고 사후 LLR 들에 기초하여 비트 LLR 들을 업데이트하기 위한 이 행별 (row-by-row) 프로세싱을 예시한다. 도 11a 에서 예시된 바와 같이, 파이프라인 "버블들" (예컨대, 프로세싱에서의 갭들) 은 사후 연산 (예컨대, 수학적들 6 내지 9) 와 비트 LLR 업데이트 단계들 (예컨대, 수학적 10) 사이의 상호종속성 (interdependency) 을 갖는 재귀적 프로세싱으로 인해 존재한다. 파이프라인 버블들은 예를 들어, 도 11b 에서 예시된 바와 같이, 증가하는 파이프라인 심도 뿐만 아니라, 메모리 충돌들과 함께 커지고, 여기서, 메모리 충돌들과 함께, 3 사이클들까지의 파이프라인 심도에서의 증가는 파이프라인 버블들로 인해 낭비된 사이클들의 수를 증가시킨다는 것을 알 수 있다. 이에 따라, 본 개시물의 양태들은 예를 들어, 특정한 행에 대하여, 사후 프로세싱이 최근의 업데이트들이 발생하는 것을 대기하는 것이 아니라, 최근에 이용 가능한 비트 LLR 들을 이용할 수 있도록, 비트 LLR 들을 업데이트하는 것과 사후 LLR 들을 연산하는 것 사이의 종속성을 완화시킴으로써, 파이프라인 버블들을 저감시키기 위한 기법들을 제시한다.
- [0090] 도 12 는 예를 들어, LDPC 인코딩된 비트들을 디코딩할 때에 프로세싱 지연들 및 메모리 충돌들을 감소시키기 위한 무선 통신들을 위한 일 예의 동작들 (1200) 을 예시한다. 특정 양태들에 따르면, 동작들 (1200) 은 기지국 (예컨대, 노드 B (110) 및/또는 송신기 시스템 (210)), 사용자 장비 (예컨대, 무선 노드 (116) 및/또는 수신기 시스템 (250)), 및/또는 무선 디바이스 (302) 와 같은 무선 통신 디바이스에서의 디코더 (예컨대, 디코더 (800)) 에 의해 수행될 수도 있다.
- [0091] 동작들 (1200) 은 LDPC 인코딩을 이용하여 인코딩된 비트들에 대응하는 로그-우도 비율 (LLR) 들을 수신함으로써 1202 에서 시작된다. 1204 에서, 무선 통신 디바이스는 패리티 검사 행렬 (PCM) 의 열들에 대응하는 인코딩된 비트들의 각각에 대하여 LLR 들을 업데이트하고, LLR 들은 인코딩된 비트들의 각각의 비트의 값의 확률을 표시한다. 1206 에서, 무선 통신 디바이스는 인코딩된 비트들의 각각에 대하여, 행 내의 각각의 LLR 을 프로세싱함으로써 PCM 의 행에 대한 사후 LLR 을 결정하고, 여기서, 인코딩된 비트들의 제 1 세트에 대하여, 사후 LLR 을 결정하는 것은 제 1 수의 아웃데이팅된 LLR 들을 이용하는 것을 포함하고, 여기서, 인코딩된 비트들의 제 2 세트에 대하여, 사후 LLR 을 결정하는 것은 제 2 수의 업데이트된 LLR 들을 이용하는 것을 포함하고, 여기서, 인코딩된 비트들의 제 1 세트 및 인코딩된 비트들의 제 2 세트는 인코딩된 비트들의 전부를 포함한다. 1208 에서, 무선 통신 디바이스는 인코딩된 비트들을 디코딩하기 위하여 LLR 들을 사용한다.
- [0092] 위에서 언급된 바와 같이, 파이프라인 버블들을 저감시키기 위하여, 비트 LLR 들을 업데이트하는 것과 사후 LLR

들을 연산하는 것 사이의 종속성은 특정한 행에 대하여, 사후 프로세싱이 최근의 업데이트들 (예컨대, 업데이트된 비트 LLR 들) 이 발생하는 것을 대기하는 것이 아니라, 최근에 이용가능한 비트 LLR 들 (예컨대, 아웃데이팅된 비트 LLR 들) 을 이용할 수 있도록 완화될 수도 있다.

[0093] 도 13 은 비트 LLR 들을 업데이트하는 것과 사후 LLR 들을 연산하는 것 사이의 종속성이 완화된 일 예의 LLR 프로세싱 파이프라인을 예시한다. 예를 들어, 예시된 바와 같이, 사후 LLR 들의 연산 동안에는, 비트 LLR 들이 이전의 반복에서 업데이트되는 것과 (예컨대, 3 사이클 파이프라인 지연 후에) 현재의 사후 LLR 연산으로 전달되는 것을 대기하는 대신에, 사후 LLR 연산은 제 1 수의 비트들에 대응하는 제 1 수의 아웃데이팅된 비트 LLR 들을 이용하여, 그리고 제 2 수의 비트들에 대응하는 제 2 수의 업데이트된 비트 LLR 들을 이용하여, 업데이트된 비트 LLR 들 없이 진행될 수도 있다. 예를 들어, 도 13 에서 예시된 바와 같이, PCM 행 0 에 대하여, 사후 LLR 연산은 최근에 업데이트된 비트 LLR 들이 이용가능하게 될 때까지 (예컨대, 3 사이클 파이프라인 지연 후에 사후 LLR 연산 스테이지로 전달됨) 최초의 3 개의 사이클들에 대한 더 오래된 (예컨대, 아웃데이팅된/업데이트되지 않은) 비트 LLR 들을 이용하여 시작될 수도 있다.

[0094] 양태들에 따르면, 이러한 완화는 파이프라인 버블들을 감소시키지만, 이것은 메시지 전달에서 도입된 지연으로 인해 디코더 성능 열화를 대가로 한다. 그러나, (파이프라인 심도에 비례적인) 메시지 전달 지연의 양에 따라, 이러한 지연들로부터의 성능 열화는 작은 경향이 있고, 또한, 파이프라인 버블 오버헤드 (pipeline bubble overhead) 들의 제거로 인해 반복 당 사이클들에서의 감소에 의해 균형잡히는 경향이 있다. 양태들에 따르면, (예컨대, 이 메시지 전달 완화를 갖지 않는 계층화된 디코더와 비교하여) 반복 당 사이클들에서의 감소가 더 많은 PCM 행들이 일정한 시간 주기 내에서 프로세싱되는 것으로 귀착되므로, 메시지 전달 지연들로 인해 도입된 성능 열화는 작은 경향이 있고, 이것은 메시지 전달 지연으로 인한 성능 열화를 보상할 수 있는 성능 개선들로 귀착될 수도 있다. 추가적으로, 메시지 전달 지연을 허용하는 것은 디코딩 동작을 위한 더 높은 이용가능한 클럭 주파수로 옮겨지는 더 심층적 파이프라인을 제공한다.

[0095] 양태들에 따르면, 계층화된 디코딩을 위하여, 수학식 6 을 수행 (예컨대, 연산) 하여 마지막 사후 메시지를 비트 LLR, $L(q_j)$ 로부터 감산한 후, 그것을 사후 연산에 공급하여 새로운 사후 메시지를 연산하고 다음으로 이것을 이용하여 $L(q_{mj})$ 을 업데이트할 수도 있다. $L(q_{mj})$ 이 수학식 10 에서의 업데이트를 위하여 LLR 메모리 (예컨대, LLR 메모리 저장소 (902) 내에 저장될 경우, 그리고 메시지 전달에서 지연이 있을 경우에, 업데이트가 발생하기 전에 일어날 수 있는 비트 LLR 의 다수의 판독들이 잠재적으로 있을 수도 있고, 이것은 사후 업데이트들의 미싱 (missing) 으로 인해 신뢰 전파 메시지 전달을 손상시킬 수 있다. 추가적으로, 위에서 논의된 바와 같이, LLR 메모리를 동시에 액세스하는 수학식들 6 및 10 으로 인한 메모리 충돌들이 있을 수도 있고, 그 예는 도 14a 및 도 14b 에서 예시된다.

[0096] 도 14a 는 위에서 논의된 바와 같이, 계층화된 LDPC 디코더에서의 지연된 메시지 전달에 의해 야기될 수도 있는 메모리 일관성 및 충돌 이슈들을 입증하는 목적들을 위한 일 예의 PCM 을 예시한다. 도 14a 에서 예시된 일 예의 PCM 에서, 행 4 는 비트들 0, 2, 및 3 (예컨대, 비트들 (4, 0), (4, 2), 및 (4, 3) 은 비-제로임) 에 접속되고, 행 5 는 비트들 0, 1, 및 5 (예컨대, 비트들 (5, 0), (5, 1), 및 (5, 5) 는 비-제로임) 에 접속된다.

[0097] 도 14b 는 기록 LLR 메모리에 대한 단일 사이클 레이턴시 및 3 사이클 파이프라인 심도를 가정하는 계층화된 LDPC 디코더 파이프라인을 예시하고, 이것은 도 14a 에서 예시된 일 예의 PCM 을 참조하여 위에서 논의된 메모리 충돌들을 예시한다. 예시된 바와 같이, 사이클 0 동안에, 디코더는 도 14a 에서 예시된 PCM 의 열 0 에 대한 비트 LLR, $L(q_0)$ 을 취함으로써, 그리고 열 0 에 대한 비트 LLR 로부터 아웃데이팅된 사후 LLR, R_{mj} 을 감산함으로써, 행 4, 비트 0 에 대하여 수학식 6 을 연산한다. 이것은 행 4, 열 0 에 대한 비트 LLR (즉, $L(q_{40})$) 로 귀착된다. 결과적인 비트 LLR, $L(q_{40})$ 은 PCM 의 열 0 (즉, 비트 0) 에 대한 업데이트된 비트 LLR 을 연산하기 위하여 나중에 이용하기 위해 어딘가에 (예컨대, LLR 저장 메모리 (902) 에) 저장될 필요가 있고, 따라서, 디코더는 예를 들어, 도 14b 의 (사이클 1, LLR 메모리 열 0) 에서 예시된 바와 같이, 비트 LLR $L(q_{40})$ 에 대한 값으로 비트 LLR $L(q_0)$ 을 덮어쓰기 (overwrite) 한다. 그러나, 파이프라인 프로세싱 지연으로 인해, $L(q_{40})$ 으로 $L(q_0)$ 을 덮어쓰기 하는 것은 나중에 행 5 의 프로세싱 동안 이슈들을

제시한다.

[0098] 예를 들어, 도 14b 의 사이클 3 에서 예시된 바와 같이, 업데이트된 비트 LLR 들을 제공하기 위한 파이프라인 프로세싱 지연으로 인해, 수학적 식 6 에 대한 프로세싱은 업데이트된 $L(q_0)$ 대신에, 메모리로부터 $L(q_{40})$ 을 판독한다. 즉, 행 5 의 프로세싱 동안에는, ($L(q_{40})$ 으로부터 유도되는) 업데이트된 $L(q_0)$ 이 파이프라인 프로세싱 지연으로 인해 메모리 내에 아직 저장되지 않으므로, 수학적 식 6 의 연산은 행 4 에 대한 비트 LLR 정보 (즉, $L(q_{40})$) 를 이용하는데, 이것은 디코더 성능을 열화시킬 수도 있다. 추가적으로, 사이클 4 에 대하여, 수학적 식들 6 및 10 를 위한 프로세싱 모두는 동일한 사이클에서 메모리를 기록하기를 원하므로, 메모리 충돌이 있다. 일부 경우들에 있어서, 디코더는 충돌을 해결하기 위하여 수학적 식 6 기록 동작 또는 수학적 식 10 기록 동작 중 어느 하나를 지연시키는 것을 선택할 수도 있고, 이것은 도 14b 에서 도시된 괄호 내 단계들에서 예시된다. 그러나, 어느 쪽이든, 바람직하지 않은 거동이 있을 수도 있다. 예를 들어, 수학적 식 6 기록 동작이 먼저 발생할 경우에, 행 5 에 대한 업데이트가 발생할 때, $L(q_{50})$ 메시지는 덮어쓰기 되었을 것이다. 추가적으로, 수학적 식 10 기록 동작이 먼저 발생할 경우, 행 4 에 대한 업데이트는 덮어쓰기 될 것이다.

[0099] 특정 양태들에 따르면, 계층화된 수학적 식들 (예컨대, 수학적 식들 6 내지 10) 에 대한 약간의 수정이 이하에서 예시된 바와 같이, 메모리 일관성 및 충돌 문제들을 해결한다.

$$L(q_{mj}) = L(q_j) - R_{mj(ola)} \quad (\text{수학적 식 11})$$

$$A_{mj} = \sum_{n \in N(m), n \neq j} \Psi(L(q_{mn})) \quad (\text{수학적 식 12})$$

$$s_{mj} = \prod_{n \in N(m), n \neq j} \text{Sign}(L(q_{mn})) \quad (\text{수학적 식 13})$$

$$R_{mj(new)} = -s_{mj} \Psi(A_{mj}) \quad (\text{수학적 식 14})$$

$$L(q_j) = L(q_j) + (R_{mj(new)} - R_{mj(ola)}) \quad (\text{수학적 식 15})$$

[0100]

[0101] 수학적 식들 6 내지 9 에서와 같이, 수학적 식 11 에 의해 생성된 $L(q_{mj})$ 들은 수학적 식들 12 내지 14 에서의 사후 프로세싱으로 공급된다. 그러나, 수학적 식들 6 내지 9 와 달리, $L(q_{mj})$ 들은 도 14b 에서의 예에서 도시된 바와 같이 LLR 메모리 내에 저장되지 않는다. 그 대신에, $L(q_j)$ 은 수학적 식 15 에서 예시된 바와 같이, 오래된 (old) 것과 새로운 (new) 사후 LLR 들 (즉, $(R_{mj(new)} - R_{mj(ola)})$) 사이의 차이로 업데이트된다.

특정 양태들에 따르면, 이 동작은 비트 LLR 로부터 오래된 사후 정보를 효과적으로 차감하고, 그것을 최근의 정보로 대체하여, 특정한 행에 대한 연산이 도 14b 에서의 예에서 예시된 바와 같이, 이전의 행에 대해 맞추어진 정보를 더 이상 이용하지 않는다.

[0102] 도 15 는 도 14a 에서 예시된 일 예의 PCM 에 대응하는 수학적 식들 11 내지 15 에 대한 프로세싱 단계들을 예시한다. 도 15 에서 알 수 있는 바와 같이, $L(q_{mj})$ 들은 메모리로 기록되지 않으므로, 일관성 충돌들이 없다. 이에 따라, 메모리는 최근의 이용가능한 $L(q_j)$ 값 (즉, 아웃데이팅된 또는 업데이트된 비트 LLR) 을 항상 유지한다.

- [0103] 추가적으로, 도 15 에서의 사이클 3 에서 예시된 바와 같이, 수학식 11 프로세싱에서 이용된 $L(q_0)$ 값은 메모리로부터 나오고, 수학식 15 프로세싱에 대한 사이클 3 에서 또한 발생하는 차이 업데이트를 갖는 행 4 에 대한 사후 LLR 에 의해 업데이트되지 않았던 더 오래된 값이다. 특정 양태들에 따르면, $L(q_{55})$ 에 대하여 $L(q_{50})$ 의 수학식 11 프로세싱을 교체함으로써, $L(q_0)$ 의 최근의 업데이트가 도 16 에서 예시된 바와 같이, $L(q_{50})$ 프로세싱에서 이용되는 것을 허용한다.
- [0104] 특정 양태들에 따르면, 메시지 전달에서의 지연으로, $L(q_j)$ 은 $L(q_{mj})$ 대신에, 메모리로부터 관독될 것이다. 수학식들 11 및 15 양자 모두는 $L(q_j)$ 을 관독하기 위하여 메모리를 액세스하므로, 높은 메모리 대역폭은 $L(q_j)$ 메시지들을, 수학식들 11 및 15 양자 모두에 대한 사이클별 (cycle-by-cycle) 프로세싱으로 공급하는 것이 필요하게 될 수도 있다.
- [0105] 도 17 은 LLR 들의 프로세싱 및 프로세싱을 수용하기 위하여 메모리 대역폭을 증가시키기 위한 예시적인 해결책을 예시한다. 예를 들어, 특정 양태들에 따르면, 값비싼 추가적인 포트들을 메모리에 추가하지 않으면서 더 높은 관독/기록 대역폭을 가능하게 하기 위하여, LLR 메모리 (예컨대, LLR 메모리 저장소 (902)) 는 듀얼 뱅크 (dual bank) 들로 분할될 수도 있고, 예를 들어, 도 17 에서 예시된 바와 같이, 수학식 11 프로세싱 및 수학식 15 프로세싱이 동일한 사이클 상에서 동일한 뱅크를 관독하는 것을 시도하는 충돌들을 회피하기 위하여, $L(q_j)$ 메시지들은 메모리 뱅크들에서 (예컨대, PCM 에서의 행들 사이의 종속성들에 기초하여) 특정한 순서로 정렬될 수 있다. 다시 말해서, LLR 메모리를 듀얼 뱅크들로 논리적으로 분할하는 것은 디코더 (예컨대, 디코더 (800)) 가 동시에 메모리의 뱅크들 양자 모두로부터 관독하고, 또한, 동시에 메모리의 뱅크들 양자 모두에 기록하는 것을 가능하게 함으로써, 관독/기록 대역폭을 증가시킨다.
- [0106] 추가적으로, 비트 (즉, 열) 인덱스들이 프로세싱되는 순서 및 메모리에서의 정렬은 또한, 도 16 에서 예시된 최근의 업데이트된 $L(q_j)$ 메시지들을 전달하는 것을 최대화하도록 수행될 수도 있다. 예를 들어, 도 17 에서, 메모리는 듀얼 뱅크들 (예컨대, "LLR 메모리 뱅크 0" 및 "LLR 메모리 뱅크 1") 로 분할될 수도 있다. 뱅크들 양자 모두에 걸쳐 저장된 비트들의 총 수는 도 16 에서의 단일 메모리와 동일하게 유지된다. 그러나, 양태들에 따르면, $L(q_2)$ 및 $L(q_4)$ 열들이 교체될 경우, 수학식 11 및 수학식 15 에 대한 프로세싱은 $L(q_j)$ 을 관독하기 위하여 임의의 주어진 사이클 상에서 동일한 메모리 뱅크를 액세스하지 않음으로써, 관독 대역폭을 증가시키고 디코딩 효율 및 시간을 개선시킨다. 추가적으로, 도 16 에서와 같이, (예컨대, 도 17 에서의 듀얼 뱅크 메모리를 이용하는) 수학식 11 에 대한 프로세싱은, 최근의 $L(q_0)$ 이 행 5 에 대한 사후 LLR 프로세싱에서 이용될 수 있도록, 비트 인덱스들 5, 그 다음으로 1, 그리고 그 다음으로 0 을 이용할 수도 있다. 다시 말해서, 특정 양태들에 따르면, 사후 LLR 및 비트 LLR 업데이트 연산들 (즉, 수학식들 11 및 15) 을 수행할 때, 디코더는 유리하게도, PCM 행 계산 순서를 선택할 수도 있거나, 메모리에서의 사후 LLR 들 및 비트 LLR 업데이트들을 정렬할 수도 있거나, 및/또는 메모리 충돌들/일관성 에러들을 경감시키고 파이프라인 프로세싱 지연들을 감소시키기 위하여, 예를 들어, PCM 에서의 행들 사이의 종속성들에 기초하여 사후 LLR 들 및 비트 LLR 업데이트들을 저장하기 위한 메모리 뱅크를 선택함으로써, 디코딩 효율 및 정확도를 증가시킬 수도 있다.
- [0107] 본원에서 개시된 방법들은 설명된 방법을 달성하기 위한 하나 이상의 단계들 또는 액션들을 포함한다. 방법 단계들 및/또는 액션들은 청구항들의 범위로부터 이탈하지 않으면서 서로 상호 교환될 수도 있다. 다시 말해서, 단계들 또는 액션들의 특정 순서가 특정되지 않으면, 특정 단계들 및/또는 액션들의 순서 및/또는 이용은 청구항들의 범위로부터 이탈하지 않으면서 수정될 수도 있다.
- [0108] 본원에서 이용된 바와 같이, 용어 "결정" 은 폭넓게 다양한 액션 (action) 들을 망라한다. 예를 들어, "결정" 은 계산, 연산, 프로세싱, 유도, 조사, 룩업 (look up) (예컨대, 테이블, 데이터베이스 또는 또 다른 데이터 구조에서의 룩업), 확인 등을 포함할 수도 있다. 또한, "결정" 은 수신 (예컨대, 정보를 수신하는 것),

액세스 (예컨대, 메모리에서 데이터를 액세스하는 것) 등을 포함할 수 있다. 또한, "결정" 은 해결, 선택, 선정, 구축 등을 포함할 수도 있다.

- [0109] 일부 경우에는, 프레임을 실제로 송신하는 것이 아니라, 디바이스는 송신을 위하여 프레임을 출력하기 위한 인터페이스를 가질 수도 있다. 예를 들어, 프로세서는 송신을 위하여, 프레임을 버스 인터페이스를 통해 RF 프론트 엔드 (front end) 로 출력할 수도 있다. 유사하게, 프레임을 실제로 수신하는 것이 아니라, 디바이스는 또 다른 디바이스로부터 수신된 프레임을 획득하기 위한 인터페이스를 가질 수도 있다. 예를 들어, 프로세서는 송신을 위하여 RF 프론트 엔드로부터 버스 인터페이스를 통해 프레임을 획득 (또는 수신) 할 수도 있다.
- [0110] 위에서 설명된 방법들의 다양한 동작들은 대응하는 기능들을 수행할 수 있는 임의의 적당한 수단에 의해 수행될 수도 있다. 수단은 회로, 주문형 집적 회로 (ASIC), 또는 프로세서를 포함하지만, 이것으로 제한되지 않는 다양한 하드웨어 및/또는 소프트웨어 컴포넌트 (들) 및/또는 모듈 (들) 을 포함할 수도 있다. 일반적으로, 도면들에서 예시된 동작들이 있을 경우, 그 동작들은 같은 번호부여를 갖는 대응하는 대응부 수단-플러스-기능 컴포넌트들을 가질 수도 있다.
- [0111] 예를 들어, 연산하기 위한 수단, 결정하기 위한 수단, (예컨대, LLR 들을) 사용하기 위한 수단, 업데이트하기 위한 수단, 판독하기 위한 수단, 및/또는 선택하기 위한 수단은 송신기 시스템 (210) 의 프로세서 (230) 및/또는 RX 데이터 프로세서 (242), 및/또는 수신기 시스템 (250) 의 프로세서 (270) 및/또는 RX 데이터 프로세서 (260) 와 같은 하나 이상의 프로세서들을 포함하는 프로세싱 시스템을 포함할 수도 있다. 추가적으로, 저장하기 위한 수단은 송신기 시스템 (210) 의 메모리 (232) 및/또는 수신기 시스템 (250) 의 메모리 (272) 와 같은 메모리를 포함할 수도 있다. 또한, 수신하기 위한 수단은 송신기 시스템 (210) 의 수신기 (222) 및/또는 안테나 (224), 및/또는 수신기 시스템 (250) 의 수신기 (254) 및/또는 안테나 (252) 와 같은 수신기 및/또는 안테나를 포함할 수도 있다.
- [0112] 본 개시물과 관련하여 설명된 다양한 예시적인 논리적 블록들, 모듈들, 및 회로들은 범용 프로세서, 디지털 신호 프로세서 (DSP), 주문형 집적 회로 (ASIC), 필드 프로그래밍가능한 게이트 어레이 (FPGA) 또는 다른 프로그래밍가능 로직 디바이스 (PLD), 별개의 게이트 또는 트랜지스터 로직, 별개의 하드웨어 컴포넌트들, 또는 본원에서 설명된 기능들을 수행하도록 설계된 그 임의의 조합으로 구현되거나 수행될 수도 있다. 범용-프로세서는 마이크로프로세서일 수도 있지만, 대안적으로, 프로세서는 임의의 상업적으로 입수가능한 프로세서, 제어기, 마이크로제어기 또는 상태 머신일 수도 있다. 프로세서는 또한, 컴퓨팅 디바이스들의 조합, 예컨대, DSP 및 마이크로프로세서, 복수의 마이크로프로세서들, DSP 코어와 함께 하나 이상의 마이크로프로세서들, 또는 임의의 다른 이러한 구성의 조합으로서 구현될 수도 있다.
- [0113] 하드웨어로 구현될 경우, 일 예의 하드웨어 구성은 무선 노드에서의 프로세싱 시스템을 포함할 수도 있다. 프로세싱 시스템은 버스 아키텍처로 구현될 수도 있다. 버스는 프로세싱 시스템의 특정 애플리케이션 및 전체적인 설계 제약들에 따라 임의의 수의 상호접속하는 버스들 및 브릿지 (bridge) 들을 포함할 수도 있다. 버스는 프로세서, 머신-판독가능 매체들, 및 버스 인터페이스를 포함하는 다양한 회로들을 함께 연결할 수도 있다. 버스 인터페이스는 그 중에서도, 네트워크 어댑터를 버스를 통해 프로세싱 시스템에 접속하기 위하여 이용될 수도 있다. 네트워크 어댑터는 PHY 계층의 신호 프로세싱 기능들을 구현하기 위하여 이용될 수도 있다. 무선 노드 (도 1 참조) 의 경우에, 사용자 인터페이스 (예컨대, 키패드, 디스플레이, 마우스, 조이스틱 등) 는 또한 버스에 접속될 수도 있다. 버스는 또한, 당해 분야에서 잘 알려져 있고, 그러므로, 더 이상 설명되지 않을 타이밍 소스들, 주변기기들, 전압 레귤레이터들, 및 전력 관리 회로들 등등과 같은 다양한 다른 회로들을 연결할 수도 있다. 프로세서는 하나 이상의 범용 및/또는 특수-목적 프로세서들로 구현될 수도 있다. 예들은 마이크로프로세서들, 마이크로제어기들, DSP 프로세서들, 및 소프트웨어를 실행할 수 있는 다른 회로부를 포함한다. 당해 분야의 당업자들은 특정한 애플리케이션 및 전체적인 시스템에 부과된 전체적인 설계 제약들에 따라 프로세싱 시스템을 위한 설명된 기능성을 어떻게 최상으로 구현할 것인지를 인식할 것이다.
- [0114] 소프트웨어로 구현될 경우, 기능들은 하나 이상의 명령들 또는 코드로서, 컴퓨터-판독가능 매체 상에 저장되거나, 컴퓨터-판독가능 매체 상에서 송신될 수도 있다. 소프트웨어는 소프트웨어, 펌웨어, 미들웨어, 마이크로코드, 하드웨어 설명 언어, 또는 그 외의 것으로서 지칭되든지 간에, 명령들, 데이터, 또는 그 임의의 조합을 의미하도록 폭넓게 해석될 것이다. 컴퓨터-판독가능 매체들은, 하나의 장소로부터 또 다른 장소까지의 컴퓨터 프로그램의 전송을 가능하게 하는 임의의 매체를 포함하는 통신 매체 및 컴퓨터 저장 매체 모두를 포함한다.

프로세서는 버스를 관리하는 것과, 머신-판독가능 저장 매체들 상에서 저장된 소프트웨어 모듈들의 실행을 포함하는 일반적인 프로세싱을 담당할 수도 있다. 컴퓨터-판독가능 저장 매체는 프로세서가 저장 매체로부터 정보를 판독할 수 있고 정보를 저장 매체에 기록할 수 있도록 프로세서에 결합될 수도 있다. 대안적으로, 저장 매체는 프로세서에 일체적일 수도 있다. 예로서, 머신-판독가능 매체들은 송신 라인, 데이터에 의해 변조된 반송파, 및/또는 무선 노드로부터 분리된, 그 상에서 저장된 명령들을 갖는 컴퓨터 판독가능 저장 매체를 포함할 수도 있고, 그 전부는 버스 인터페이스를 통해 프로세서에 의해 액세스될 수도 있다.

대안적으로 또는 추가적으로, 머신-판독가능 매체들 또는 그 임의의 부분은 캐시 및/또는 일반적인 레지스터 파일들에서 그러한 바와 같이, 프로세서 내로 통합될 수도 있다. 머신-판독가능 저장 매체들의 예들은 예로서, RAM (랜덤 액세스 메모리), 플래시 메모리, ROM (판독 전용 메모리), PROM (Programmable Read-Only Memory; 프로그래밍 가능 판독-전용 메모리), EPROM (Erasable Programmable Read-Only Memory; 소거가능 프로그래밍가능 판독-전용 메모리), EEPROM (Electrically Erasable Programmable Read-Only Memory; 전기적 소거가능 프로그래밍가능 판독-전용 메모리), 레지스터들, 자기적 디스크들, 광학적 디스크들, 하드 드라이브들, 또는 임의의 다른 적당한 저장 매체, 또는 그 임의의 조합을 포함할 수도 있다. 머신-판독가능 매체들은 컴퓨터-프로그램 제품에서 구체화될 수도 있다.

[0115] 소프트웨어 모듈은 단일 명령 또는 다수의 명령들을 포함할 수도 있고, 몇몇 상이한 코드 세그먼트들 상에서, 상이한 프로그램 사이에서, 그리고 다수의 저장 매체들엔 걸쳐 분산될 수도 있다. 컴퓨터-판독가능 매체들은 다수의 소프트웨어 모듈들을 포함할 수도 있다. 소프트웨어 모듈들은, 프로세서와 같은 장치에 의해 실행될 때, 프로세싱 시스템으로 하여금, 다양한 기능들을 수행하게 하는 명령들을 포함한다. 소프트웨어 모듈들은 송신 모듈 및 수신 모듈을 포함할 수도 있다. 각각의 소프트웨어 모듈은 단일 저장 디바이스에서 상주할 수도 있거나, 다수의 저장 디바이스들에 걸쳐 분산될 수도 있다. 예로서, 소프트웨어 모듈은 트리거링 이벤트가 발생할 때에 하드 드라이브로부터 RAM 으로 로딩될 수도 있다. 소프트웨어 모듈의 실행 동안, 프로세서는 액세스 속도를 증가시키기 위하여 명령들의 일부를 캐시로 로딩할 수도 있다. 그 다음으로, 하나 이상의 캐시 라인들은 프로세서에 의한 실행을 위하여 일반적인 레지스터 파일로 로딩될 수도 있다. 이하의 소프트웨어 모듈의 기능성을 지칭할 때, 이러한 기능성은 그 소프트웨어 모듈로부터의 명령들을 실행할 때에 프로세서에 의해 구현된다는 것이 이해될 것이다.

[0116] 또한, 임의의 접속은 컴퓨터-판독가능 매체로 적절하게 칭해진다. 예를 들어, 동축 케이블, 광섬유 케이블, 트위스트 페어 (twisted pair), 디지털 가입자 회선 (digital subscriber line; DSL), 또는 적외선 (IR), 라디오 (radio), 및 마이크로파 (microwave) 와 같은 무선 기술들을 이용하여, 소프트웨어가 웹사이트, 서버, 또는 다른 원격 소스로부터 송신될 경우, 동축 케이블, 광섬유 케이블, 트위스트 페어, DSL, 또는 적외선, 라디오, 및 마이크로파와 같은 무선 기술들은 매체의 정의 내에 포함된다. 본원에서 이용된 바와 같은 디스크 (disk) 및 디스크 (disc) 는 콤팩트 디스크 (compact disc; CD), 레이저 디스크 (laser disc), 광학 디스크 (optical disc), 디지털 다기능 디스크 (digital versatile disc; DVD), 플로피 디스크 (floppy disk) 및 Blu-ray® 디스크 (disc) 를 포함하고, 여기서, 디스크 (disk) 들은 통상 데이터를 자기적으로 재생하는 반면, 디스크 (disc) 들은 데이터를 레이저들로 광학적으로 재생한다. 이에 따라, 일부 양태들에서, 컴퓨터 판독가능 매체들은 비-일시적인 컴퓨터-판독가능 매체들 (예컨대, 유형의 (tangible) 매체들) 을 포함할 수도 있다. 추가적으로, 다른 양태들에 대하여, 컴퓨터-판독가능 매체들은 일시적 컴퓨터-판독가능 매체들 (예컨대, 신호) 을 포함할 수도 있다. 상기의 조합들은 또한, 컴퓨터-판독가능 매체들의 범위 내에 포함되어야 한다.

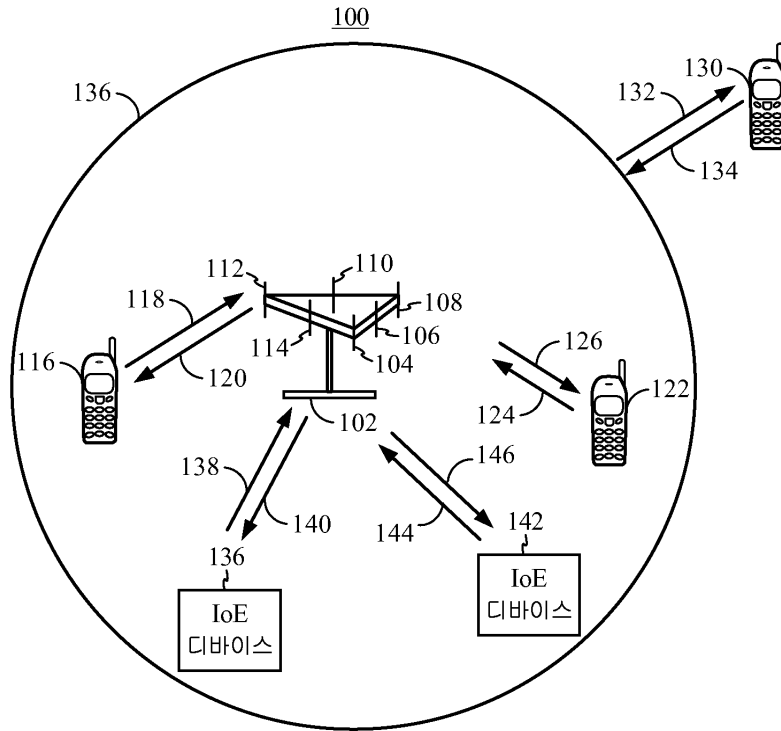
[0117] 이에 따라, 특정 양태들은 본원에서 제시된 동작들을 수행하기 위한 컴퓨터 프로그램 제품을 포함할 수도 있다. 예를 들어, 이러한 컴퓨터 프로그램 제품은 명령들을 그 위에 저장한 (및/또는 인코딩한) 컴퓨터-판독가능 매체를 포함할 수도 있고, 명령들은 본원에서 설명된 동작들을 수행하기 위하여 하나 이상의 프로세서들에 의해 실행가능할 수도 있다.

[0118] 또한, 본원에서 설명된 방법들 및 기법들을 수행하기 위한 모듈들 및/또는 다른 적절한 수단은 적용가능한 바와 같은 무선 노드 및/또는 기지국에 의해 다운로드될 수 있고 및/또는 이와 다르게 획득될 수 있다는 것이 인식되어야 한다. 예를 들어, 이러한 디바이스는 본원에서 설명된 방법들을 수행하기 위한 수단의 전달을 용이하게 하기 위하여 서버에 결합될 수 있다. 대안적으로, 본원에서 설명된 다양한 방법들은 저장 수단 (예를 들어, RAM, ROM, 물리적 저장 매체 예컨대, 콤팩트 디스크 (CD) 또는 플로피 디스크, 등) 을 통해 제공될 수 있어서, 무선 노드 및/또는 기지국은 저장 수단을 디바이스에 결합 또는 제공 시에 다양한 방법들을 획득할 수 있다. 또한, 본원에서 설명된 방법들 및 기법들을 디바이스에 제공하기 위한 임의의 다른 적당한 기법이 사용될 수 있다.

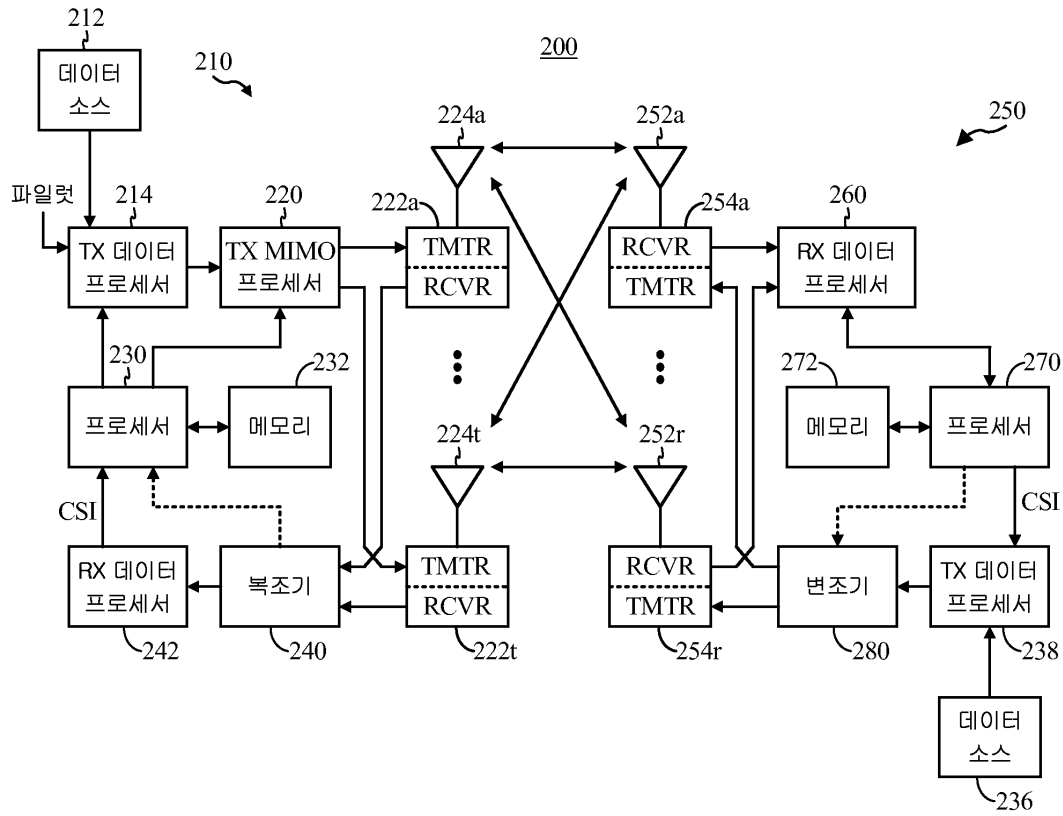
[0119] 청구항들은 위에서 예시된 정확한 구성 및 컴포넌트들로 제한되지 않는다는 것이 이해되어야 한다. 다양한 수정들, 변경들 및 변동들은 청구항들의 범위로부터 이탈하지 않으면서, 위에서 설명된 방법들 및 장치의 배열, 동작 및 세부사항들에서 행해질 수도 있다.

도면

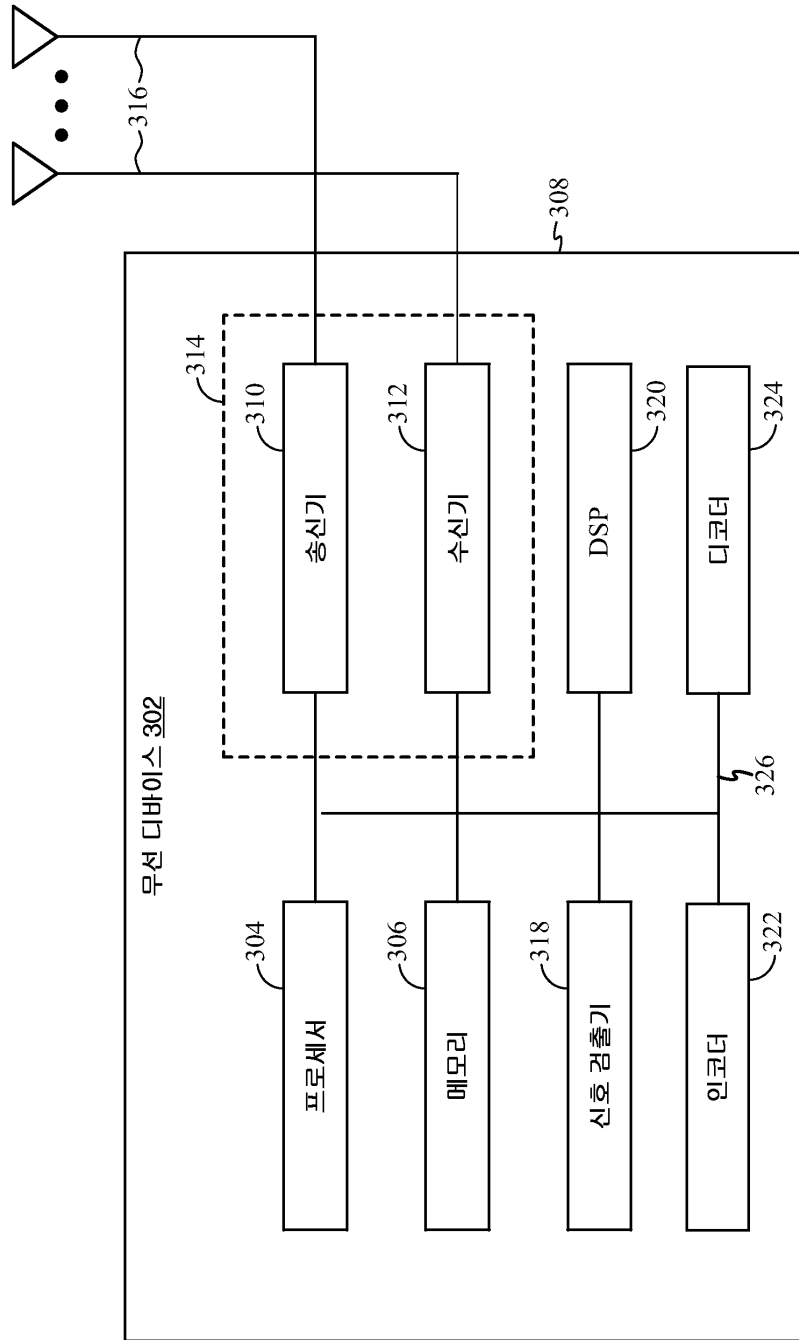
도면1



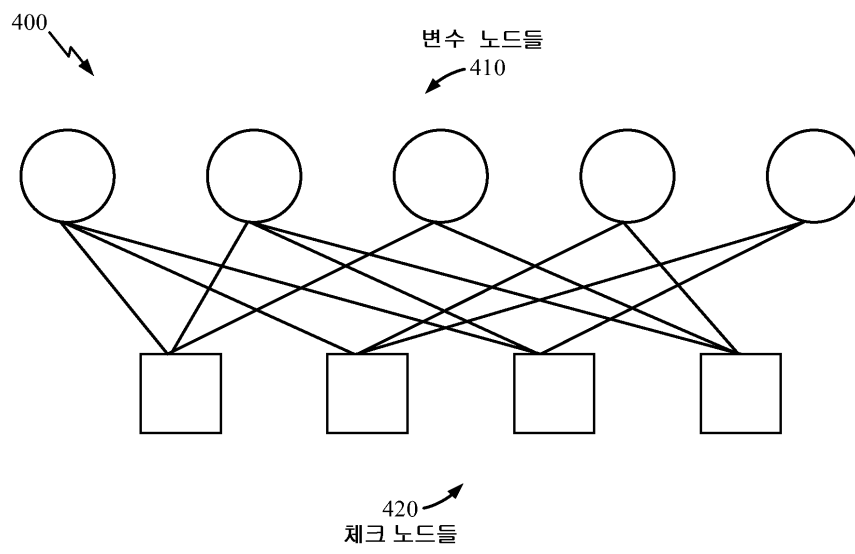
도면2



도면3



도면4

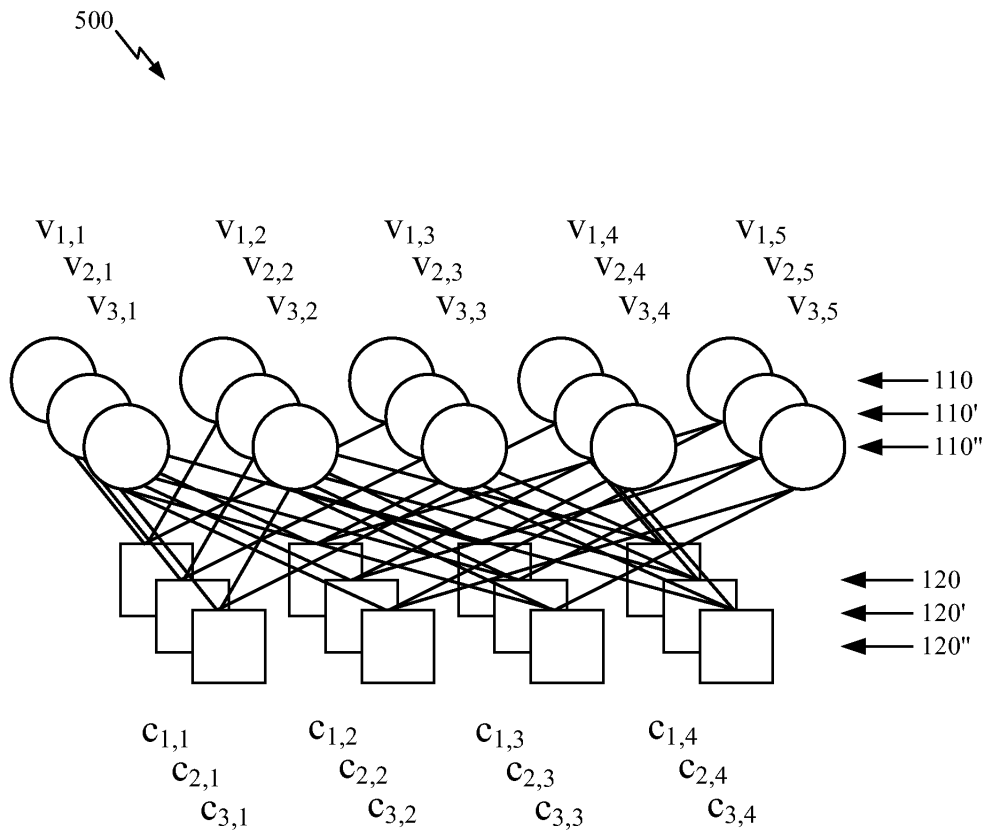


도면4a

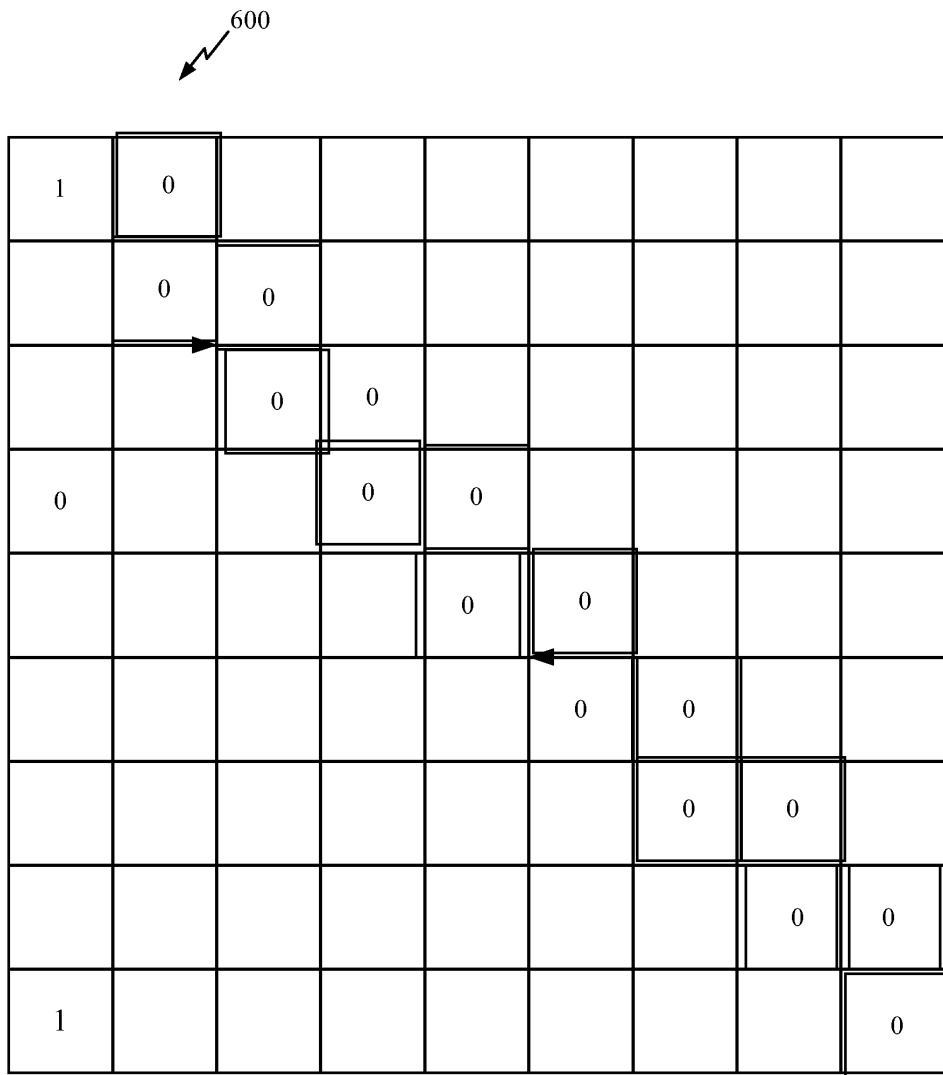
400A

$$H = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{bmatrix} \quad x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix}$$

도면5

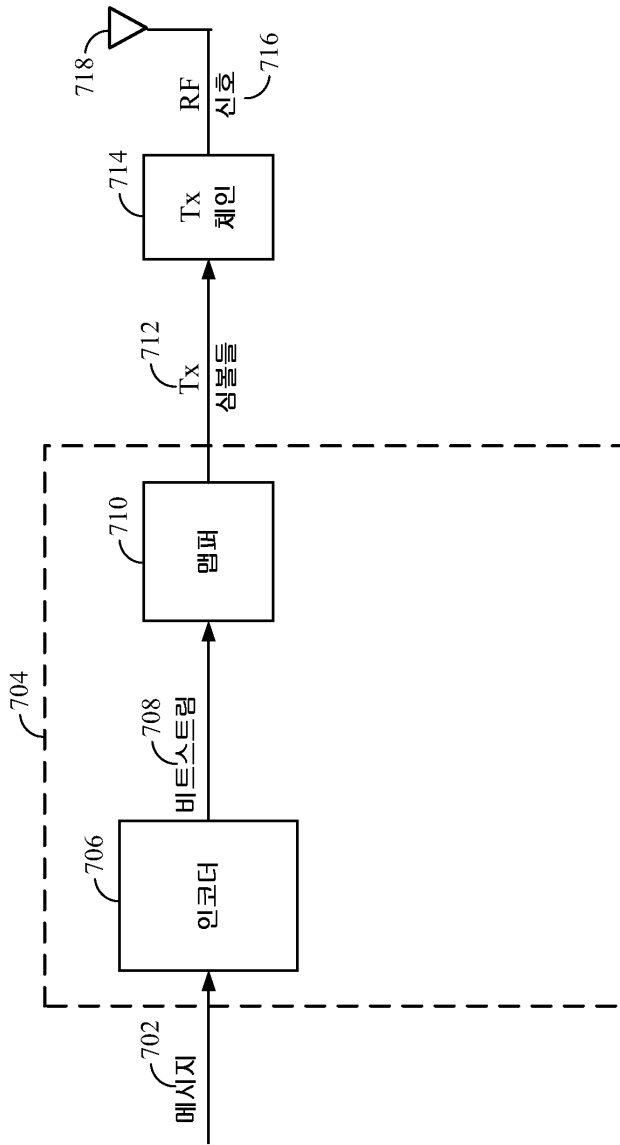


도면6

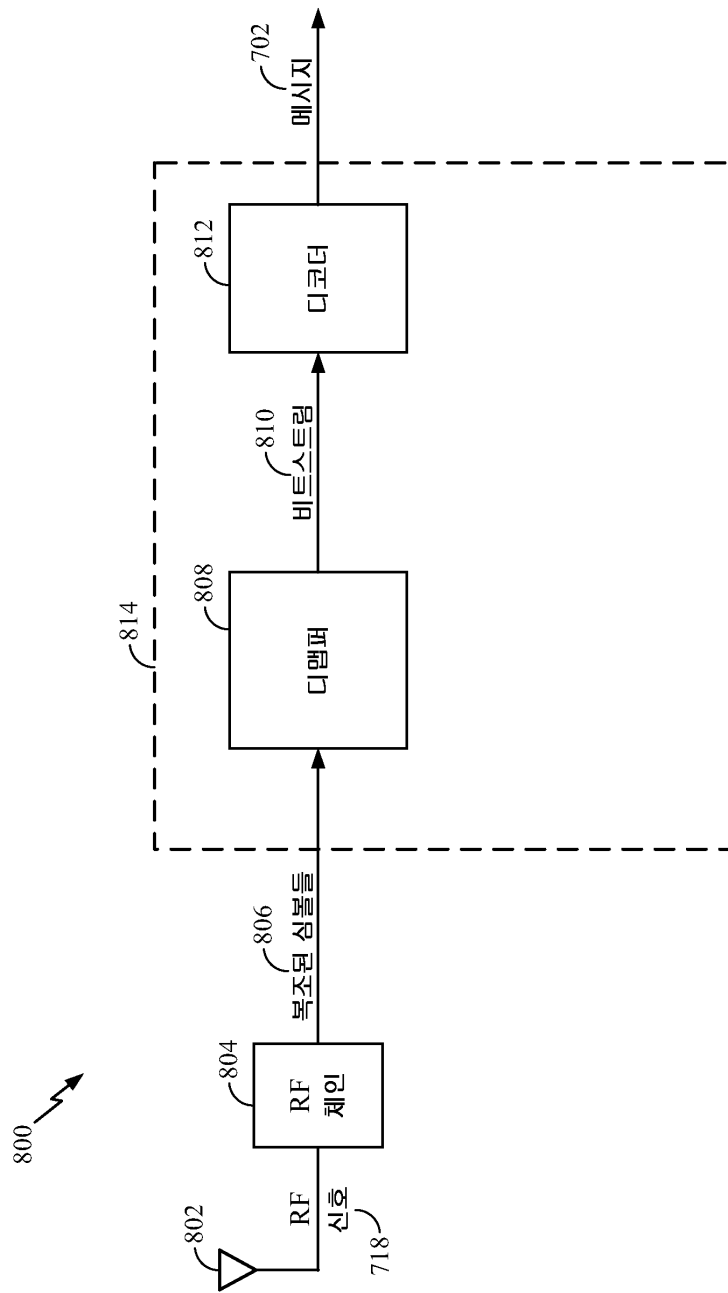


도면7

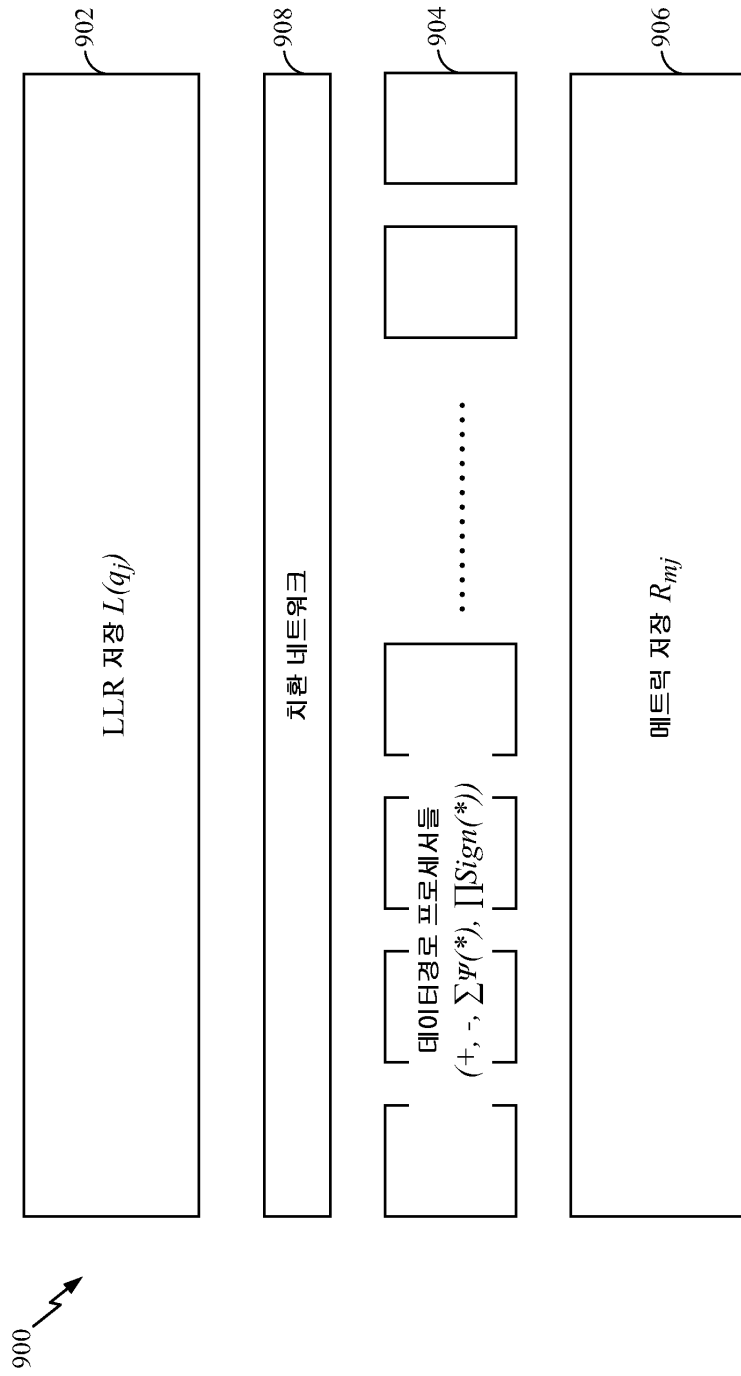
700 ↘



도면8



도면9



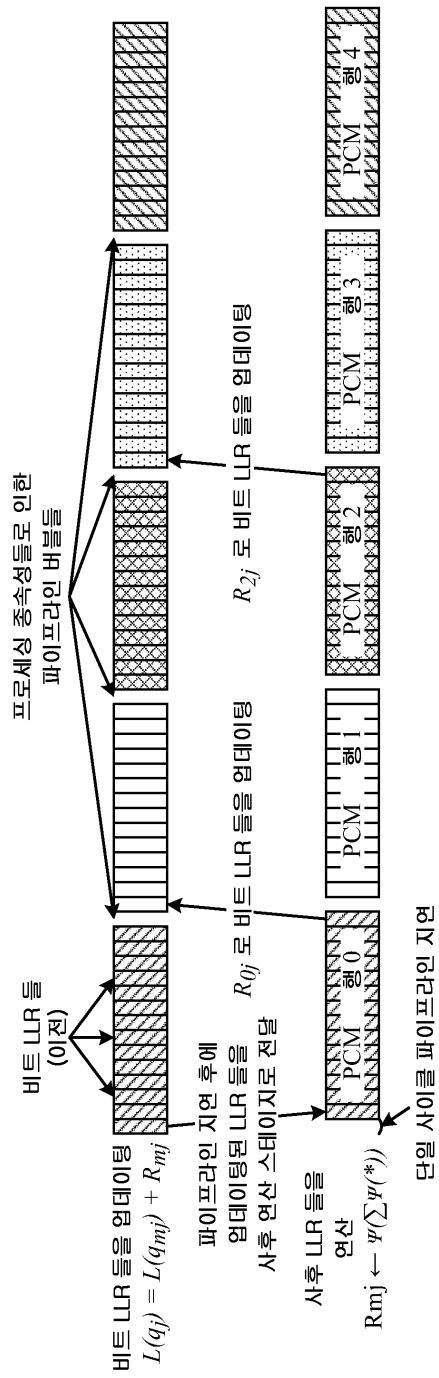
도면10

비트 열들

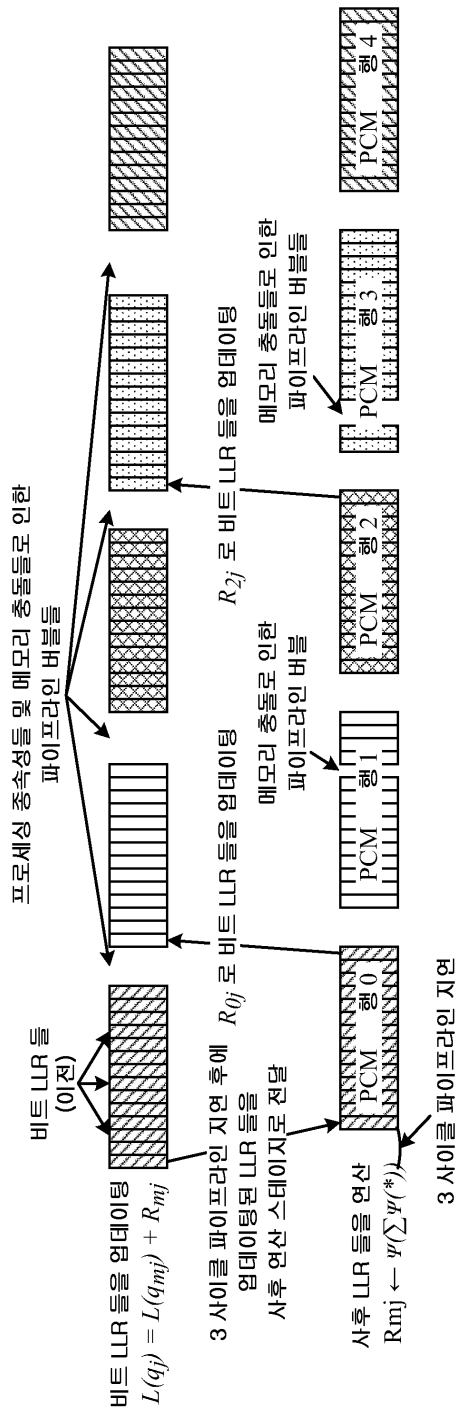
0	48	29	28	28	39	9	61	-	-	63	45	80	-	-	37	32	22	1	0	-	-	21	22	23
1	4	49	42	48	11	30	-	-	-	49	17	41	37	15	54	-	-	-	0	0	-	-	-	-
2	35	76	78	51	37	35	21	-	17	64	-	-	-	59	7	-	32	-	-	0	0	0	-	-
3	9	65	44	9	54	56	73	34	42	-	-	-	35	-	-	46	39	0	-	-	0	0	0	-
4	3	62	7	80	68	26	-	80	55	-	36	-	26	-	9	72	-	-	-	-	-	-	0	0
5	26	75	33	21	69	59	3	38	-	-	-	35	-	62	36	26	-	1	-	-	-	-	0	0

패리티
체크
행들

도면11a

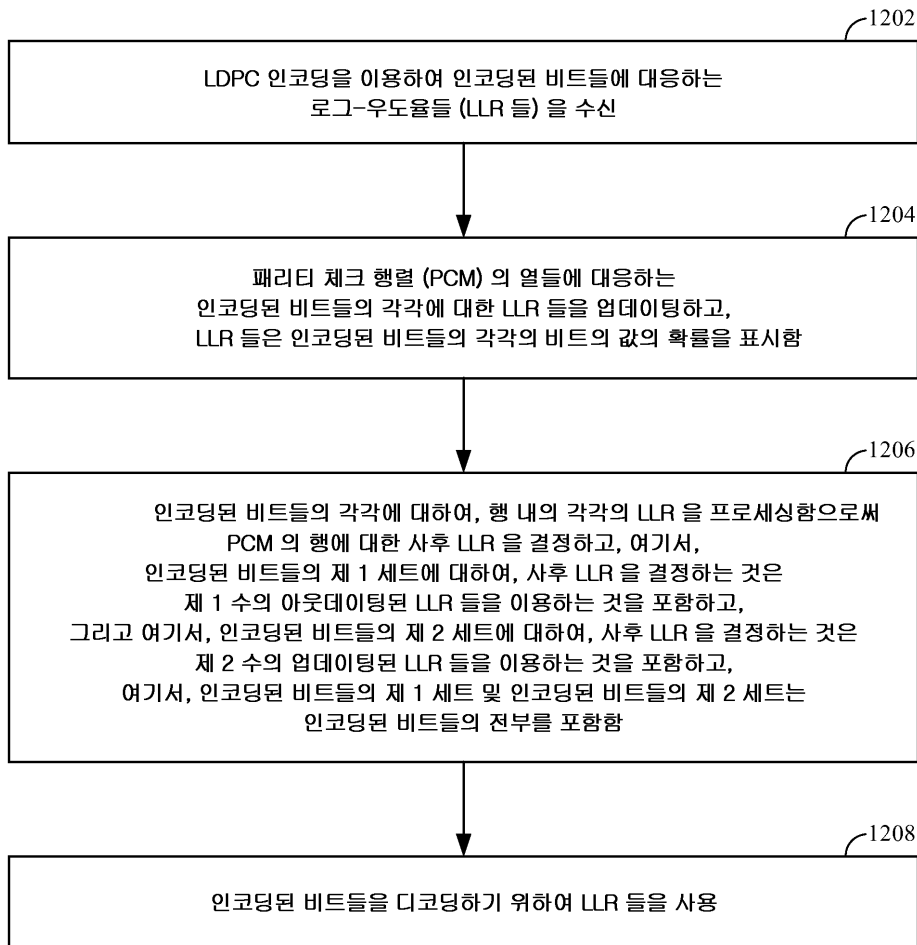


도면11b

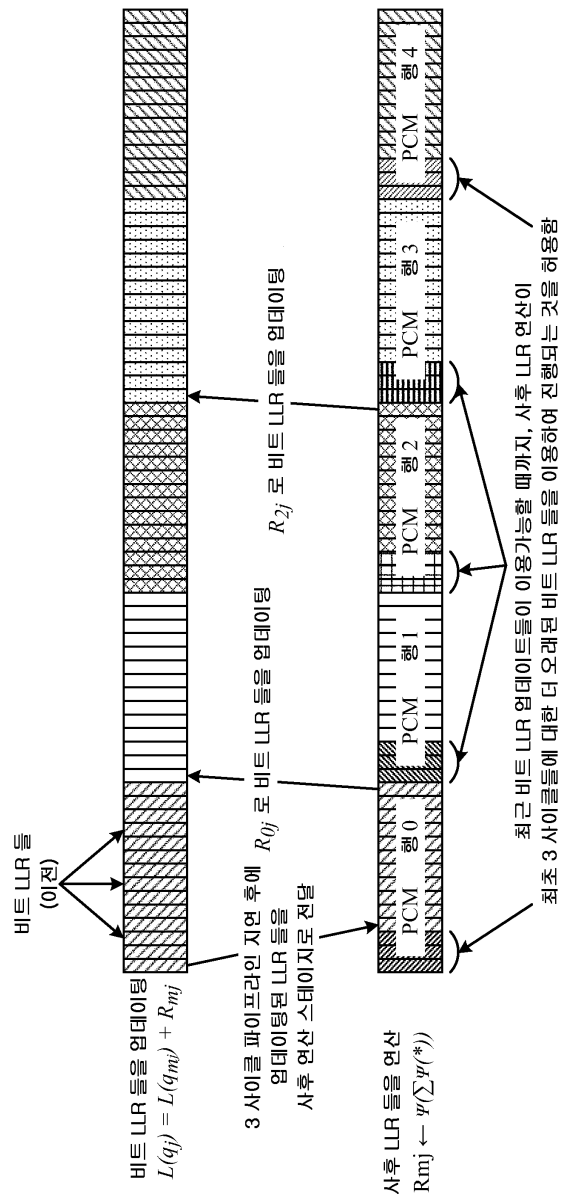


도면12

1200 ↘



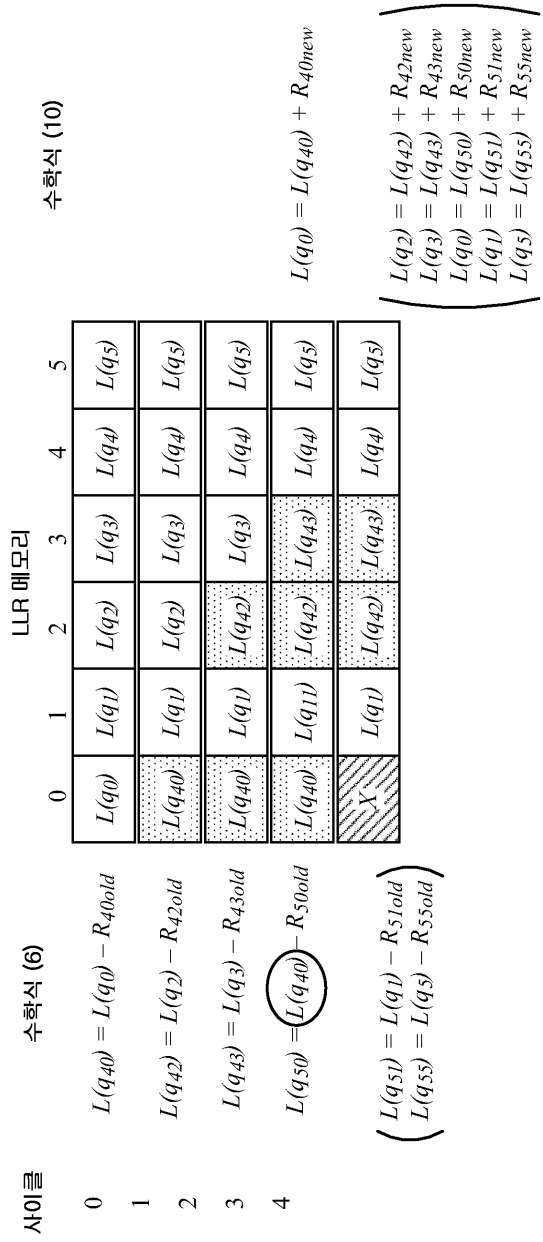
도면13



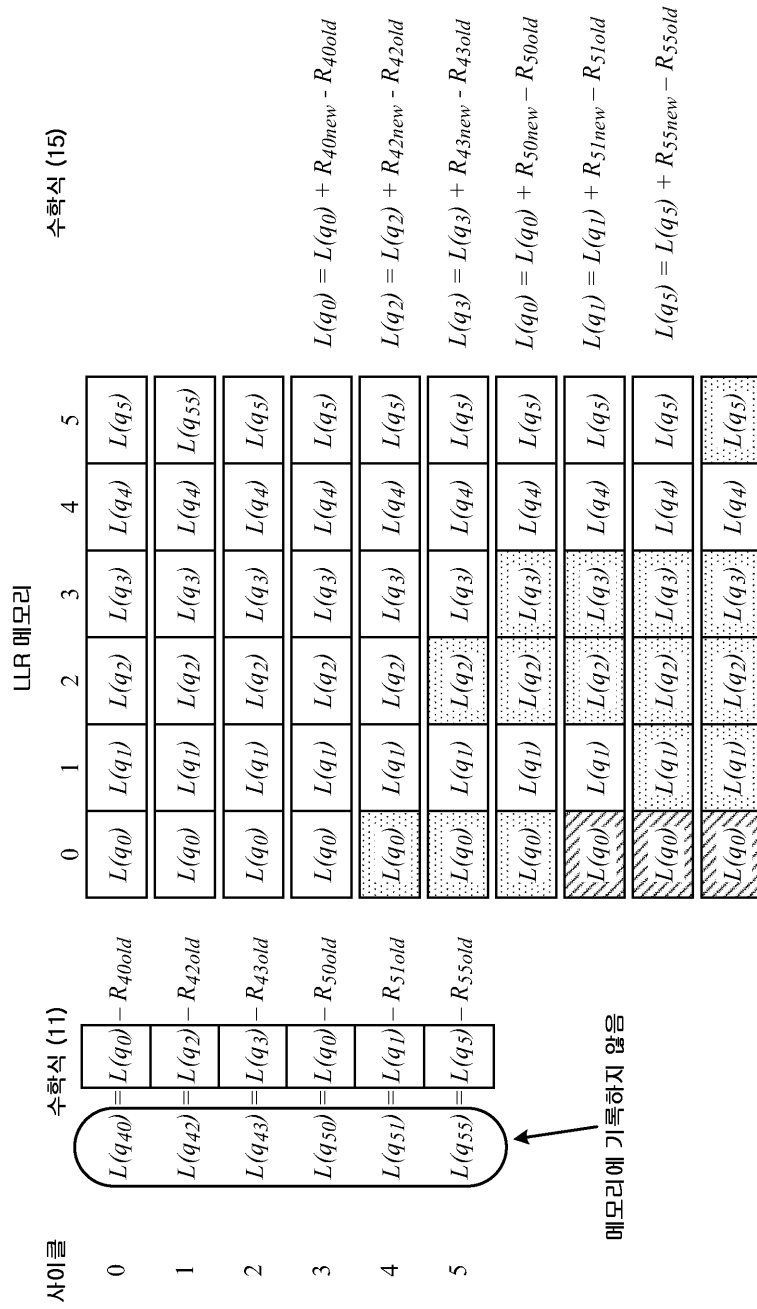
도면14a

	0	1	2	3	4	5
...						
...						
행 4	1	0	1	1	0	0
행 5	1	1	0	0	0	1
...						

도면 14b



도면15



계층화된 LDPC 파이프라인 (사후 차이로 비트 LLR 을 업데이트함)

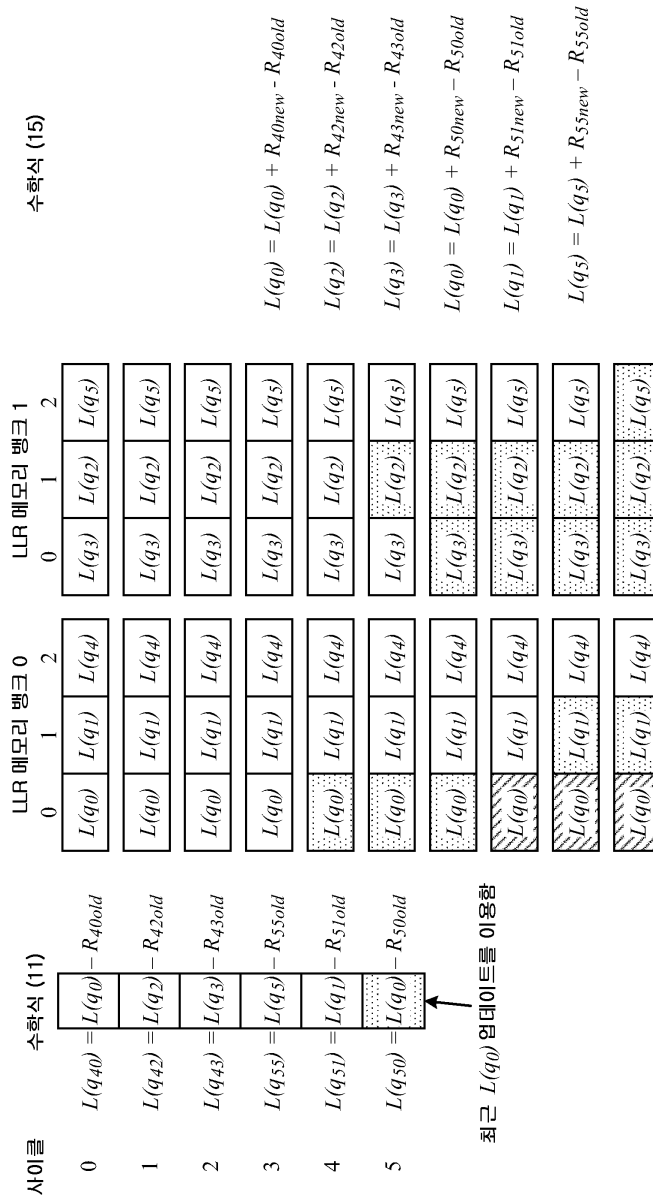
도면16

사이클	수확식 (11)	LLR 메모리						수확식 (15)
		0	1	2	3	4	5	
0	$L(q_{40}) = L(q_0) - R_{40old}$	$L(q_0)$	$L(q_1)$	$L(q_2)$	$L(q_3)$	$L(q_4)$	$L(q_5)$	
1	$L(q_{42}) = L(q_2) - R_{42old}$	$L(q_0)$	$L(q_1)$	$L(q_2)$	$L(q_3)$	$L(q_4)$	$L(q_{55})$	
2	$L(q_{43}) = L(q_3) - R_{43old}$	$L(q_0)$	$L(q_1)$	$L(q_2)$	$L(q_3)$	$L(q_4)$	$L(q_5)$	
3	$L(q_{55}) = L(q_5) - R_{55old}$	$L(q_0)$	$L(q_1)$	$L(q_2)$	$L(q_3)$	$L(q_4)$	$L(q_5)$	$L(q_0) = L(q_0) + R_{40new} - R_{40old}$
4	$L(q_{51}) = L(q_1) - R_{51old}$	$L(q_0)$	$L(q_1)$	$L(q_2)$	$L(q_3)$	$L(q_4)$	$L(q_5)$	$L(q_2) = L(q_2) + R_{42new} - R_{42old}$
5	$L(q_{50}) = L(q_0) - R_{50old}$	$L(q_0)$	$L(q_1)$	$L(q_2)$	$L(q_3)$	$L(q_4)$	$L(q_5)$	$L(q_3) = L(q_3) + R_{43new} - R_{43old}$
		$L(q_0)$	$L(q_1)$	$L(q_2)$	$L(q_3)$	$L(q_4)$	$L(q_5)$	$L(q_0) = L(q_0) + R_{50new} - R_{50old}$
		$L(q_0)$	$L(q_1)$	$L(q_2)$	$L(q_3)$	$L(q_4)$	$L(q_5)$	$L(q_1) = L(q_1) + R_{51new} - R_{51old}$
		$L(q_0)$	$L(q_1)$	$L(q_2)$	$L(q_3)$	$L(q_4)$	$L(q_5)$	$L(q_5) = L(q_5) + R_{55new} - R_{55old}$

최근 $L(q_0)$ 업데이트를 이용함

계속화된 LDPC 파이프라인 ($L(q_{50})$ 및 $L(q_{55})$ 의 순서를 교체)

도면17



계층화된 LDPC 파이프라인 (편독 충돌들을 해결하기 위한 이중 메모리 뱅크들)