



**(19) 대한민국특허청(KR)**  
**(12) 등록특허공보(B1)**

(45) 공고일자 2012년12월05일  
(11) 등록번호 10-1208547  
(24) 등록일자 2012년11월29일

- (51) 국제특허분류(Int. Cl.)  
*H03M 13/11* (2006.01)
- (21) 출원번호 10-2007-7006122
- (22) 출원일자(국제) 2005년09월16일  
심사청구일자 2010년08월26일
- (85) 번역문제출일자 2007년03월16일
- (65) 공개번호 10-2007-0058508
- (43) 공개일자 2007년06월08일
- (86) 국제출원번호 PCT/KR2005/003103
- (87) 국제공개번호 WO 2006/031092  
국제공개일자 2006년03월23일
- (30) 우선권주장  
1020040074764 2004년09월17일 대한민국(KR)  
1020040086320 2004년10월27일 대한민국(KR)
- (56) 선행기술조사문헌  
B. Classon et al. "LDPC coding for OFDMA PHY," IEEE 802.16 Broadband Wireless Access Working Group, IEEE C802.16e-04/278r1, 2004.08.17.\*  
Yu Yi et al., "The Semi-Algebra Low-Density Parity-Check Codes," IEEE ICC 2004, Vol. 1, pp. 440-443, 20-24 Jun. 2004.\*  
\*는 심사관에 의하여 인용된 문헌
- (73) 특허권자  
엘지전자 주식회사  
서울특별시 영등포구 여의대로 128 (여의도동)
- (72) 발명자  
정규혁  
서울특별시 영등포구 선유동2로 29, 203동 606호 (양평동3가, 현대2차아파트)  
오민석  
서울 송파구 오금동 현대백조아파트 101동 810호
- (74) 대리인  
김용인, 박영복

전체 청구항 수 : 총 10 항

심사관 : 권성락

(54) 발명의 명칭 LDPC 코드를 이용한 부호화 및 복호화 방법

**(57) 요약**

본 발명은 저밀도 패리티 검사 코드를 이용하여 부호화 또는 복호화하는 방법에 있어서 성능을 향상시키고 복잡도를 줄일 있는 부호화 및 복호화 방법에 관한 것이다. 본 발명의 일 양상으로서, 본 발명의 따른 LDPC 코드를 이용한 부호화 방법은,  $(n-k) \times n$  차원의 패리티 검사 행렬  $H$ 에 의해 정의되는 LDPC 코드에 의한 부호화 방법에 있어서, 입력 소스 데이터를 상기 패리티 검사 행렬  $H(H_p$  및  $H_d$ 는 패리티 검사 행렬  $H$ 의 일부분으로서  $H=[H_d | H_p]$ ( $H_d$ 는  $(n-k) \times k$ ,  $H_p$ 는  $(n-k) \times (n-k)$  차원임)의 구조임.)를 이용하여 부호화하되, 상기  $H_p$ 를  $z \times z$  차원의 다수의 서브행렬들로 분할할 경우, 각 서브행렬은 단위 행렬(identity matrix) 또는 0 행렬(zero matrix)이고, 상기  $H_p$ 의 모든 열의 무게(column weight)가 2 이상인 것을 특징으로 한다.

**특허청구의 범위**

**청구항 1**

(n-k)×n 차원의 패리티 검사 행렬 H를 이용한 부호화 방법에 있어서,  
 입력 소스 데이터(input source data)를 상기 패리티 검사 행렬 H(H<sub>p</sub> 및 H<sub>d</sub>는 패리티 검사 행렬 H의 일부분으로서 H=[H<sub>d</sub> | H<sub>p</sub>](H<sub>d</sub>는 (n-k)×k, H<sub>p</sub>는 (n-k)×(n-k) 차원임.)를 이용해 부호화된 데이터를 출력하는 단계를 포함하되,

상기 H<sub>p</sub>는 각각이 z×z 차원의 단위 행렬(identity matrix) 또는 z×z 차원의 0 행렬(zero matrix)인 서브행렬을 다수 포함하고, 상기 z×z 차원의 단위행렬을 90° 만큼 회전시킨 행렬인 서브행렬을 적어도 하나 포함하며, 상기 H<sub>p</sub>의 모든 열의 무게(column weight)가 2 이상인 것을 특징으로 하는,

LDPC 코드를 이용한 부호화 방법.

**청구항 2**

제1항에 있어서,

기 H<sub>p</sub> 는,

$$\begin{pmatrix} I & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 \\ I & I & 0 & \cdots & 0 & 0 & 0 & 0 & 0 \\ 0 & I & I & \cdots & 0 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & I & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \cdots & I & I & 0 & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & I & 90 & I & I \\ 0 & 0 & 0 & \cdots & 0 & 0 & I & I & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & I & 0 & I \end{pmatrix}$$

(여기서, I 는 z×z 차원의 단위 행렬이고, 0 는 z×z 차원의 0 행렬이며, 90은 상기 단위 행렬을 90° 회전시킨 행렬임.)

의 형식을 갖는 것을 특징으로 하는,

LDPC 코드를 이용한 부호화 방법.

**청구항 3**

제1항에 있어서,

상기 H<sub>p</sub> 는,

$$\begin{pmatrix} I & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 \\ I & I & 0 & \cdots & 0 & 0 & 0 & 0 & 0 \\ 0 & I & I & \cdots & 0 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & I & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \cdots & I & I & 0 & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & I & I & 90 & I \\ 0 & 0 & 0 & \cdots & 0 & 0 & I & I & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 & I & I \end{pmatrix}$$

(여기서,  $I$  는  $z \times z$  차원의 단위 행렬이고,  $0$  는  $z \times z$  차원의  $0$  행렬이며,  $90$ 은 상기 단위 행렬을  $90^\circ$  회전시킨 행렬임.)

의 형식을 갖는 것을 특징으로 하는,  
LDPC 코드를 이용한 부호화 방법.

**청구항 4**

제1항에 있어서,  
상기  $H_p$  는,

$$\begin{pmatrix} I & I & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ I & 0 & I & 0 & 0 & \cdots & 0 & 0 & 0 \\ 90 & I & I & I & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 0 & I & I & \cdots & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & I & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \cdots & I & I & 0 \\ 0 & 0 & 0 & 0 & 0 & \cdots & 0 & I & I \\ 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & I \end{pmatrix}$$

(여기서,  $I$  는  $z \times z$  차원의 단위 행렬이고,  $0$  는  $z \times z$  차원의  $0$  행렬이며,  $90$ 은 상기 단위 행렬을  $90^\circ$  회전시킨 행렬임.)

의 형식을 갖는 것을 특징으로 하는,  
LDPC 코드를 이용한 부호화 방법.

**청구항 5**

제1항에 있어서,  
상기  $H_p$  는,

$$\begin{pmatrix} I & I & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & I & I & 0 & 0 & \cdots & 0 & 0 & 0 \\ I & 90 & I & I & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 0 & I & I & \cdots & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & I & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \cdots & I & I & 0 \\ 0 & 0 & 0 & 0 & 0 & \cdots & 0 & I & I \\ 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & I \end{pmatrix}$$

(여기서,  $I$  는  $z \times z$  차원의 단위 행렬이고,  $0$  는  $z \times z$  차원의  $0$  행렬이며,  $90$ 은 상기 단위 행렬을  $90^\circ$  회전시킨 행렬임.)

의 형식을 갖는 것을 특징으로 하는,  
LDPC 코드를 이용한 부호화 방법.

**청구항 6**

$(n-k) \times n$  차원의 패리티 검사 행렬  $H$ 를 이용한 복호화 방법에 있어서,

부호화된 데이터를 상기 패리티 검사 행렬  $H$  ( $H_p$  및  $H_d$ 는 패리티 검사 행렬  $H$ 의 일부분으로서  $H=[H_d | H_p]$  ( $H_d$ 는  $(n-k) \times k$ ,  $H_p$ 는  $(n-k) \times (n-k)$  차원임.)를 이용해 복호화하여 원본 데이터를 출력하는 단계를 포함하되, 상기  $H_p$ 는 각각이  $z \times z$  차원의 단위 행렬(identity matrix) 또는  $z \times z$  차원의 0 행렬(zero matrix)인 서브행렬을 다수 포함하고, 상기  $z \times z$  차원의 단위행렬을  $90^\circ$  만큼 회전시킨 행렬인 서브행렬을 적어도 하나 포함하며, 상기  $H_p$ 의 모든 열의 무게(column weight)가 2 이상인 것을 특징으로 하는,

LDPC 코드를 이용한 복호화 방법.

**청구항 7**

제6항에 있어서,

상기  $H_p$ 는,

$$\begin{pmatrix} I & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 \\ I & I & 0 & \cdots & 0 & 0 & 0 & 0 & 0 \\ 0 & I & I & \cdots & 0 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & I & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \cdots & I & I & 0 & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & I & 90 & I & I \\ 0 & 0 & 0 & \cdots & 0 & 0 & I & I & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & I & 0 & I \end{pmatrix}$$

(여기서,  $I$ 는  $z \times z$  차원의 단위 행렬이고,  $0$ 는  $z \times z$  차원의 0 행렬이며,  $90$ 은 상기 단위 행렬을  $90^\circ$  회전시킨 행렬임.)

의 형식을 갖는 것을 특징으로 하는,

LDPC 코드를 이용한 복호화 방법.

**청구항 8**

제6항에 있어서,

상기  $H_p$ 는,

$$\begin{pmatrix} I & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 \\ I & I & 0 & \cdots & 0 & 0 & 0 & 0 & 0 \\ 0 & I & I & \cdots & 0 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & I & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \cdots & I & I & 0 & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & I & I & 90 & I \\ 0 & 0 & 0 & \cdots & 0 & 0 & I & I & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 & I & I \end{pmatrix}$$

(여기서,  $I$ 는  $z \times z$  차원의 단위 행렬이고,  $0$ 는  $z \times z$  차원의 0 행렬이며,  $90$ 은 상기 단위 행렬을  $90^\circ$  회전시킨 행렬임.)

의 형식을 갖는 것을 특징으로 하는,

LDPC 코드를 이용한 복호화 방법.

**청구항 9**

제6항에 있어서,  
상기  $H_p$  는,

$$\begin{pmatrix} I & I & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ I & 0 & I & 0 & 0 & \cdots & 0 & 0 & 0 \\ 90 & I & I & I & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 0 & I & I & \cdots & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & I & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \cdots & I & I & 0 \\ 0 & 0 & 0 & 0 & 0 & \cdots & 0 & I & I \\ 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & I \end{pmatrix}$$

(여기서,  $I$  는  $z \times z$  차원의 단위 행렬이고,  $0$  는  $z \times z$  차원의  $0$  행렬이며,  $90$ 은 상기 단위 행렬을  $90^\circ$  회전시킨 행렬임.)

의 형식을 갖는 것을 특징으로 하는,  
LDPC 코드를 이용한 복호화 방법.

**청구항 10**

제6항에 있어서,  
상기  $H_p$  는,

$$\begin{pmatrix} I & I & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & I & I & 0 & 0 & \cdots & 0 & 0 & 0 \\ I & 90 & I & I & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 0 & I & I & \cdots & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & I & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \cdots & I & I & 0 \\ 0 & 0 & 0 & 0 & 0 & \cdots & 0 & I & I \\ 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & I \end{pmatrix}$$

(여기서,  $I$  는  $z \times z$  차원의 단위 행렬이고,  $0$  는  $z \times z$  차원의  $0$  행렬이며,  $90$ 은 상기 단위 행렬을  $90^\circ$  회전시킨 행렬임.)

의 형식을 갖는 것을 특징으로 하는,  
LDPC 코드를 이용한 복호화 방법.

**청구항 11**

삭제

**청구항 12**

삭제

**청구항 13**

삭제

**청구항 14**

삭제

**청구항 15**

삭제

**청구항 16**

삭제

**청구항 17**

삭제

**청구항 18**

삭제

**청구항 19**

삭제

**청구항 20**

삭제

**청구항 21**

삭제

**청구항 22**

삭제

**청구항 23**

삭제

**청구항 24**

삭제

**청구항 25**

삭제

**청구항 26**

삭제

**청구항 27**

삭제

**청구항 28**

삭제

**청구항 29**

삭제

청구항 30

삭제

청구항 31

삭제

청구항 32

삭제

청구항 33

삭제

청구항 34

삭제

청구항 35

삭제

청구항 36

삭제

**명세서**

**기술분야**

[0001] 본 발명은 부호화(encoding) 및 복호화(decoding) 방법에 관한 것이다. 보다 구체적으로는, 저밀도 패리티 검사(LDPC: Low Density Parity Check) 코드를 이용하여 부호화 또는 복호화하는 방법에 있어서 성능(performance)을 향상시키고 복잡도를 줄일 있는 부호화 및 복호화 방법에 관한 것이다.

**배경기술**

[0002] 일반적으로 부호화(encoding)라 함은 송신측에서 송신된 데이터가 통신채널을 통하여 전송되는 과정에서 발생하는 신호의 일그러짐, 손실 등에 의한 오류의 발생에도 불구하고 수신측에서 원래의 데이터를 복원할 수 있도록 하기 위하여 송신측에서 데이터 처리를 하는 과정을 의미한다. 복호화(decoding)는 부호화되어 송신된 신호를 수신측에서 원래의 데이터로 복원하는 과정이다.

[0003] 최근에 LDPC 코드를 이용한 부호화 방법이 부각되고 있다. LDPC 코드는 패리티 검사 행렬(parity check matrix) H의 원소들의 대부분이 0이어서 저밀도(low density)인 선형 블록 부호(linear block code)로서 1962년 갤러거(Gallager)에 의해 제안되었다. LDPC 부호는 매우 복잡하여 제안 당시의 기술로는 구현이 불가능하였기 때문에 잊혀져 있다가 1995년에 재발견되어 성능이 매우 우수함이 입증된 이래로 최근에 그에 관한 연구가 활발히 진행되고 있는 상황이다. (참고문헌: [1] Robert G. Gallager, "Low-Density Parity-Check Codes," The MIT Press, September 15, 1963. [2] D.J.C.Mackay, Good error-correcting codes based on very sparse matrices, IEEE Trans. Inform. Theory, IT-45, pp.399-431(1999))

[0004] LDPC 코드의 패리티 검사 행렬은 1의 개수가 매우 적기 때문에 매우 큰 블록 크기에서도 반복 복호를 통하여 복호가 가능하여 블록 크기가 매우 커지면 터보 코드처럼 채널 용량 한계에 근접하는 성능을 보인다.

[0005] LDPC 코드는  $(n-k) \times n$  패리티 검사 행렬 H에 의해 설명될 수 있다. 상기 패리티 검사 행렬 H에 대응하는 생성 행렬(generator matrix) G는 다음의 수학식1에 의해 구할 수 있다.

[0006] [수학식 1]

- [0007]  $H^2G = 0$
- [0008] LDPC 코드를 이용한 부호화 및 복호화 방법에 있어서는 송신측에서 상기 패리티 검사 행렬 H와 수학식1의 관계에 있는 상기 생성 행렬 G를 이용하여 다음의 수학식2에 의해 입력 데이터를 부호화한다.
- [0009] [수학식 2]
- [0010]  $x = G^2s$ (여기서, x는 코드워드(codeword)이고, s는 입력되는 소스 데이터임.)
- [0011] 최근에는, 상기 생성 행렬 G에 의하지 않고 상기 패리티 검사 행렬 H를 이용하여 입력 데이터를 부호화하는 방법이 일반적으로 사용되고 있다, 따라서, 상기한 바와 같이, LDPC 코드를 이용한 부호화 방법에서는 상기 패리티 검사 행렬 H가 가장 중요한 요소라 할 수 있다.
- [0012] 상기 패리티 검사 행렬(parity check matrix) H는, 도1에 도시된 바와 같이,  $z \times z$  차원의 다수의 퍼뮤테이션 행렬(permutation matrices) 또는 0 행렬(zero matrix)에 의해 구성될 수 있다. 즉, 도1에서  $P_{ij}$ 는  $z \times z$  차원의 퍼뮤테이션 행렬 또는 0 행렬을 의미한다.
- [0013] 상기 다수의 퍼뮤테이션 행렬은 기본 퍼뮤테이션 행렬(base permutation matrix)로부터 일정 규칙에 의해 변형되어 형성된 것들이다. 예를 들어, 기본 퍼뮤테이션 행렬을  $z \times z$  차원의 단위 행렬(identity matrix)로 할 경우에, 상기 기본 퍼뮤테이션 행렬의 모든 행을 특정 방향으로 일정 간격만큼 쉬프트(shift)하거나, 특정 행(또는 열)을 임의의 다른 행(또는 열)과 교환하거나, 상기 기본 퍼뮤테이션 행렬을 일정 각도로 회전시키거나 하는 등의 방법에 의해 다수의 퍼뮤테이션 행렬을 형성할 수 있다.
- [0014] 상기 패리티 검사 행렬 H를 구성하는 상기 다수의 퍼뮤테이션 행렬들을 행렬에 의해서 표현하는 대신, 상기 기본 퍼뮤테이션 행렬로부터 변형된 방법에 따라 상기 다수의 퍼뮤테이션 행렬들의 타입(type)을 미리 정해 놓고 상기 패리티 검사 행렬 H를 상기 다수의 퍼뮤테이션 행렬들의 타입에 의해 표현할 수 있다. 이하에서, 이와 같이 상기 다수의 퍼뮤테이션 행렬들의 타입에 의해 표현된 패리티 검사 행렬 H를 기본 행렬(base matrix)  $H_b$  로 정의하여 사용하기로 한다. 상기 기본 행렬  $H_b$ 는  $m_b \times n_b$  차원을 갖는다 여기서,  $m_b = (n-k)/z$ ,  $n_b = n/z$ 이다. 상기 기본 행렬  $H_b$ 의 각 원소인 상기 퍼뮤테이션 행렬의 타입에 관한 정보(이하에서, '퍼뮤테이션 타입'이라 함.)를 이에 대응하는 퍼뮤테이션 행렬로 대체하여 확장(expanding)함으로써 상기 패리티 검사 행렬 H를 생성할 수 있다.
- [0015] 도2는 상기 기본 행렬  $H_b$ 의 일례를 도시한 것이다. 도2에서는  $5 \times 5$  차원의 단위 행렬(identity matrix)을 기본 퍼뮤테이션 행렬로 하여 상기 기본 퍼뮤테이션 행렬의 모든 행을 특정 방향으로 일정 간격만큼 쉬프트시킴으로써 다수의 퍼뮤테이션 행렬을 형성한 예이다. 이때,  $5 \times 5$  차원의 단위 행렬인 기본 퍼뮤테이션 행렬의 퍼뮤테이션 타입을 '1'로 하고, 상기 기본 퍼뮤테이션 행렬의 모든 행을 1행 만큼 쉬프트시켜 형성된 퍼뮤테이션 행렬의 퍼뮤테이션 타입을 '2'로 하며, 상기 기본 퍼뮤테이션 행렬의 모든 행을 2 행 만큼 쉬프트시켜 형성된 퍼뮤테이션 행렬의 퍼뮤테이션 타입을 '3'으로 하며, 상기 기본 퍼뮤테이션 행렬의 모든 행을 3행 만큼 쉬프트시켜 형성된 퍼뮤테이션 행렬의 퍼뮤테이션 타입을 '4'로 하며, 상기 기본 퍼뮤테이션 행렬의 모든 행을 4 행 만큼 쉬프트시켜 형성된 퍼뮤테이션 행렬의 퍼뮤테이션 타입을 '5'로 하여 상기 패리티 검사 행렬 H를 상기 퍼뮤테이션 타입에 의해 표현한 것이다. 도2에서 '0'는  $5 \times 5$  차원의 0 행렬(zero matrix)를 의미한다. 상기 패리티 검사 행렬 H를 상기 퍼뮤테이션 타입에 의해 표현하면 상기 패리티 검사 행렬 H를 저장하기 위한 메모리를 절약할 수 있다. 도2에서는 상기 퍼뮤테이션 타입을 정수에 의해 표현하였으나 이는 예시적인 것에 불과한 것으로서, 상기 퍼뮤테이션 타입은 여러 가지 다양한 방법에 의해 표현될 수 있을 것이다.
- [0016] 상기 기본 행렬  $H_b$ 를  $(H_b)_d$  및  $(H_b)_p$  두 부분에 의해 구성된다고 생각할 때, 상기  $(H_b)_p$  부분은 일반적으로, 도2에 도시된 바와 같이, 블록 이중 대각 행렬(block dual diagonal matrix)이 사용된다. 블록 이중 대각 행렬은 주 대각(main diagonal) 및 상기 주 대각 바로 밑 또는 위쪽의 대각이 모두 단위 행렬이고 나머지가 모두 0 행렬인 것을 의미한다.
- [0017] 상기  $(H_b)_p$  부분을 블록 이중 대각 행렬로 하였을 경우에는 상기 패리티 검사 행렬 H에 무게(weight)가 1인 다수의 열(column)이 생긴다. 즉, 도2에서, 상기  $(H_b)_p$ 의 마지막 열은  $5 \times 5$  차원의 단위 행렬 하나와 나머지  $5 \times 5$  차원의 0 행렬로 구성되어 상기 패리티 검사 행렬 H 입장에서는 무게가 1인 열이 다섯 개 생기게 되는 것이다. 실제로 이용되는 LDPC 코드를 이용한 부호화 및 복호화 방법에 있어서는 기본 퍼뮤테이션 행렬로  $5 \times 5$  차원 보다 훨씬 큰 행렬이 사용되기 때문에 상기 패리티 검사 행렬 H에는 도2의 예에서 보다 무게가 1인 열이 훨씬 많

이 생기기 때문에 부호화 및 복호화를 위한 연산이 복잡해지고 그 결과로서 부호화 및 복호화 성능을 크게 저하시키는 문제점이 있다.

**발명의 상세한 설명**

[0018] 본 발명은 상기한 바와 같은 종래기술의 문제점을 해결하여 부호화 및 복호화 성능을 향상시키고 복잡도 (complexity)를 줄이고 성능을 향상시킬 수 있는 LDPC 코드를 이용한 부호화 방법을 제공하는 것을 그 목적으로 한다.

[0019] 본 발명은 기본 행렬  $H_b$  를  $(H_b)_d$  및  $(H_b)_p$  두 부분에 의해 구성되고, 상기  $(H_b)_p$  부분을 블록 이중 대각 행렬로 하였을 경우에 패리티 검사 행렬  $H$ 의  $H_p$  부분에 무게가 1인 열(column)이 생기지 않도록 하는 것을 일 특징으로 한다.

[0020] 본 발명의 일 양상으로서, 본 발명의 따른 LDPC 코드를 이용한 부호화 방법은,  $(n-k) \times n$  차원의 패리티 검사 행렬  $H$ 에 의해 정의되는 LDPC(Low Density Parity Check) 코드에 의한 부호화 방법에 있어서, 입력 소스 데이터 (input source data)를 상기 패리티 검사 행렬  $H(H_p$  및  $H_d$  는 패리티 검사 행렬  $H$ 의 일부분으로서  $H=[H_d | H_p]$ ( $H_d$  는  $(n-k) \times k$ ,  $H_p$ 는  $(n-k) \times (n-k)$  차원임)의 구조임.)를 이용하여 부호화 또는 복호화하는 단계를 포함하되, 상기  $H_p$ 를  $z \times z$  차원의 다수의 서브행렬들로 분할할 경우, 각 서브행렬은 단위 행렬(identity matrix) 또는 0 행렬(zero matrix)이고, 상기  $H_p$ 의 모든 열의 무게(column weight)가 2 이상인 것을 특징으로 한다.

[0021] 본 발명의 다른 양상으로서, 본 발명에 따른 LDPC 코드를 이용한 복호화 방법은,  $(n-k) \times n$  차원의 패리티 검사 행렬  $H$ 에 의해 정의되는 LDPC 코드에 의한 복호화 방법에 있어서, 입력 소스 데이터를 상기 패리티 검사 행렬  $H(H_p$  및  $H_d$ 는 패리티 검사 행렬  $H$ 의 일부분으로서  $H=[H_d | H_p]$ ( $H_d$ 는  $(n-k) \times k$ ,  $H_p$ 는  $(n-k) \times (n-k)$  차원임)의 구조임.)를 이용하여 부호화 또는 복호화하는 단계를 포함하되, 상기  $H_p$ 를  $z \times z$  차원의 다수의 서브행렬들로 분할할 경우, 각 서브행렬은 단위 행렬(identity matrix) 또는 0 행렬(zero matrix)이고, 상기  $H_p$ 의 모든 열의 무게(column weight)가 2 이상인 것을 특징으로 한다.

[0022] 본 발명의 또 다른 양상으로서, 본 발명에 따른 LDPC 코드를 이용한 부호화 방법은,  $z$ 개의 비트수 단위의  $k_b$  개의 데이터 열벡터(coloum vector) 그룹으로 구성되는 입력 데이터 시퀀스  $s$  를,  $m \times n$  차원을 가지고  $z \times z$  차원의 퍼뮤테이션 행렬 또는 제로행렬의  $H$  행렬 원소들을 포함하는 패리티 검사 행렬  $H$ 를 이용하여  $z$ 개 비트 수 단위의  $m_b$  개의 패리티 열벡터(column vector) 그룹으로 구성되는 패리티 시퀀스에서, 상기  $H$ 행렬 원소와 데이터 열벡터들을 이용하여  $p(0)$  패리티 벡터를 구하는 단계와; 상기 제1 패리티 열벡터  $P(0)$  와 상기 데이터 열벡터들, 상기  $H$ 행렬 원소를 사용하는 정해진 적어도 하나의 순환(recursion) 수식에 의해  $p(1)$  부터  $p(m_b-1)$  까지의 패리티 벡터들을 순차적으로 구하는 단계를 포함하여 이루어지고, 여기서, 상기  $n$  은 상기 입력 데이터의 시퀀스의 길이이고, 상기  $m$  은 패리티 비트 시퀀스 길이이고,  $k = n-m$  인 것을 특징으로 한다.

**산업상 이용 가능성**

[0101] 본 발명은 이동 통신 시스템이나 광대역 무선접속 시스템 등과 같은 무선 통신 시스템뿐만 아니라, 부호화 또는 복호화가 필요한 모든 분야에 적용 가능하다.

**도면의 간단한 설명**

[0023] 도1은  $z \times z$  차원의 다수의 퍼뮤테이션 행렬(permutation matrices) 또는 0 행렬(zero matrix)을 포함하는 패리티 검사 행렬  $H$ 의 구성도임.

[0024] 도2는 기본 행렬  $H_b$  의 일례를 도시한 것임.

[0025] 도3은 본 발명의 바람직한 일 실시예를 설명하기 위한 통신 시스템 구성도임.

[0026] 도4는 본 발명의 바람직한 일 실시예에 따라서 기본 행렬  $H_b$  형식으로 도시된 패리티 검사 행렬  $H$ 의 구조를 예시하여 도시한 것임.

[0027] 도5는 본 발명의 바람직한 다른 실시예에 따라서 기본 행렬  $H_b$  형식으로 도시된 패리티 검사 행렬  $H$ 의 구조를 예

시하여 도시한 것임.

[0028] 도6은 본 발명의 바람직한 또 다른 실시예에 따라서 기본 행렬  $H_b$  형식으로 도시된 패리티 검사 행렬  $H$ 의 구조를 예시하여 도시한 것임.

[0029] 도7은 본 발명의 바람직한 또 다른 실시예에 따라서 기본 행렬  $H_b$  형식으로 도시된 패리티 검사 행렬  $H$ 의 구조를 예시하여 도시한 것임.

[0030] 도8은 본 발명의 바람직한 또 다른 실시예에 따른 기본 행렬  $H_b$ 의 형태에 의해 표현된 패리티 검사 행렬  $H$ 의 구조를 예시하여 도시한 것임.

[0031] 도9는 본 발명의 바람직한 또 다른 실시예에 따른 기본 행렬  $H_b$ 의 형태에 의해 표현된 패리티 검사 행렬  $H$ 의 구조를 예시하여 도시한 것임.

[0032] BEST MODE FOR CARRYING OUT THE INVENTION

[0033] 이하에서는 본 발명에 따른 LDPC(Low Density Parity Check) 코드를 이용한 부호화 방법의 바람직한 실시예들을 첨부된 도면을 참조하여 설명하도록 한다. 도3은 본 발명의 바람직한 일 실시예를 설명하기 위한 도면으로서, 본 발명의 기술적 특징이 무선 통신 시스템에 적용된 일례이다. 이하에서 설명되는 실시예는 본 발명의 특징을 설명하기 위한 예시에 불과한 것으로서 본 발명의 기술적 특징은 부호화가 필요한 모든 분야에 적용 가능함은 당업자에게 자명하다.

[0034] 도3에서, 송신기(10)와 수신기(30)가 무선 채널(20)을 매개로 통신을 수행한다. 상기 송신기(10)에서는 데이터 소스(11)로부터 출력된  $k$  비트의 소스 데이터( $u$ )가 LDPC 부호화 모듈(13)에서의 LDPC 부호화(encoding)에 의해  $n$  비트의 코드워드( $c$ )가 된다. 코드워드( $c$ )는 변조 모듈(15)에 의해 무선 변조되어 안테나(17)를 통하여 송신되고, 무선채널(20)을 통해 상기 수신기(30)의 안테나(31)로 수신된다. 상기 수신기(30)에서는 상기 송신기(10)에서 일어났던 과정의 역과정을 거친다. 즉, 수신된 데이터가 복조 모듈(33)에 의해 복조되고, LDPC 복호화 모듈(35)에 의해 복호되어 최종적으로 소스 데이터( $u$ )를 얻을 수 있다. 상술한 바와 같은 데이터 송수신 과정은 본 발명의 특징을 설명하기 위해 필요한 최소한의 범위 내에서 설명된 것으로 이외에도 데이터 전송을 위해 필요한 다른 많은 과정이 있음은 당업자에게 자명하다.

[0035] 상기 LDPC 부호화 모듈에서 입력 소스 데이터를 부호화하기 위해 사용되는 상기 패리티 검사 행렬(parity check matrix)  $H$ 는  $(n-k) \times n$  차원을 갖는다. 상기  $k$ 는 상기 LDPC 부호화 모듈(13)로 입력되는 입력 데이터의 길이(비트 단위)이고, 상기  $n$ 은 부호화된 코드워드( $c$ )의 길이(비트 단위)를 의미한다. 상기 패리티 검사 행렬(parity check matrix)  $H$ 는, 도1에 도시된 바와 같이,  $z \times z$  차원의 다수의 퍼뮤테이션 행렬(permutation matrices) 또는 0 행렬(zero matrix)에 의해 표현될 수 있다.

[0036] 전술한 바와 같이, 상기 패리티 검사 행렬  $H$ 는 기본 퍼뮤테이션 행렬과 상기 기본 퍼뮤테이션 행렬을 일정한 규칙에 의해 변형시켜 형성한 다수의 퍼뮤테이션 행렬들의 퍼뮤테이션 타입(permutation type)을 포함하는 기본 행렬(base matrix)  $H_b$ 에 의해 표현될 수 있다. 또한, 상기 기본 행렬  $H_b$ 를  $(H_b)_d$  및  $(H_b)_p$  두 부분으로 분할하여 생각할 때, 상기  $(H_b)_p$  부분은 일반적으로 블록 이중 대각 행렬(block dual diagonal matrix)이 사용되는데, 이로 인해 부호화 또는 복호화에 있어 성능 저하의 문제점이 발생되는 종래기술에 대한 설명에서 언급한 바와 같다. 본 발명은 이러한 종래기술의 문제점을 해결하기 위하여 상기  $H_p$  부분을 종래기술과 다른 방식으로 구성함을 일 특징으로 한다.

[0037] 도4는 본 발명의 바람직한 일 실시예에 따른 기본 행렬  $H_b$ 의 형태에 의해 표현된 패리티 검사 행렬  $H$ 의 구조를 예시하여 도시한 것이다. 도4에서,  $I$ 는  $z \times z$  차원의 단위 행렬이고,  $O$ 는  $z \times z$  차원의 0 행렬이며,  $90$ 은 상기 단위 행렬을  $90^\circ$  회전시킴으로써 생성한 행렬을 의미한다. 도4의  $(H_b)_p$  부분을 도2의  $(H_b)_p$  부분과 비교해 보면 오른쪽 밑의  $3 \times 3$  부분(도4에서 해칭(hatching)한 부분)의 구조가 차이가 있음을 알 수 있다.

[0038] 상기  $(H_b)_p$  구조를 도4와 같이 변화시킨 이유는 다음과 같다.

[0039] 첫째, 상기 패리티 검사 행렬  $H$ 가 전체적으로 열 무게(column weight)가 1인 열이 없도록 함으로써 부호화 또는 복호화시 성능을 높일 수 있도록 한 것이다.

[0040] 둘째, 상기 패리티 검사 행렬  $H$ 가 전체적으로 4-싸이클(4-cycle)이 없도록 하고, 상기 패리티 검사 행렬  $H$  전

체에 대하여 6-싸이클(6-cycle)이 기 설정된 임계값( $C_{max}$ ) 이하가 되도록 한 것이다. 4-싸이클(4-cycle)이라 함은 H의 임의의 두 행이 두 개의 지점에 동시에 1을 갖는 경우를 의미하는데, 이러한 4-싸이클은 부호화 또는 복호화 성능을 저하시키는 원인이다. 6-싸이클이라 함은 상기 패리티 검사 행렬 H의 임의의 세 행 중에서 선택된 모든 조합 가능한 두 개의 행이 같은 지점에 1을 갖는 경우를 의미한다.

[0041] 셋째, 상기 단위 행렬을 90° 회전시킨 행렬을 추가함으로써, 선형적으로 종속적인(linearly dependent) 다수의 행(rows)이 발생하는 것을 방지하여 HARQ(Hybrid Auto Retransmission reQuest) 동작(operation), 특히 IR(Incremental Redundancy) 동작을 보조하기 위한 것이다. 선형적으로 종속적인 다수의 행이 발생하면 부호화 또는 복호화를 위한 방정식들의 해를 구할 수 없어 부호화에서는 불필요한 계산량이 증가하고 복호화에서는 성능의 하락을 초래한다.

[0042] LDPC 코드를 이용한 부호화 방법에 있어서는 수학식1 및 수학식2에 의해 생성 행렬(generator matrix) G를 이용하여 입력 소스 데이터를 부호화 할 수 있다. 즉, k 비트의 입력 소스 데이터  $s_{1 \times k}$ 는 수학식 2에 의해 부호화되어 n 비트의 코드워드  $x_{1 \times k}$ 이 된다. 코드워드 x는  $x=[s \ p]=[s_0, s_1, \dots, s_{k-1}, p_0, p_1, \dots, p_{m-1}]$ 의 구성을 갖는다(여기서,  $(p_0, p_1, \dots, p_{m-1})$ 은 패리티 검사 비트들(parity check bits)이고,  $(s_0, s_1, \dots, s_{k-1})$ 은 시스템 비트들(systematic bits)이다.).

[0043] 상기 생성 행렬 G를 이용한 부호화 방법은 매우 복잡하다. 따라서, 이러한 복잡도를 줄이기 위해 상기 생성 행렬 G에 의하지 않고, 패리티 검사 행렬 H를 이용해 직접 입력 소스 데이터를 부호화하는 것이 바람직하다.

[0044] 상기 입력 소스 데이터  $S=(s_0, \dots, s_{k-1})$ 는  $s(i)=[s_{iz}, s_{iz+1}, \dots, s_{(i+1)z-1}]^T$ 과 같은 z비트의  $k_b(k_b=n_b-m_b)$  열 벡터(column vector)로 나눌 수 있고, 상기 패리티 검사 비트  $p=(p_0, \dots, p_{k-1})$ 도 마찬가지로  $p(i)=[p_{iz}, p_{iz+1}, \dots, p_{(i+1)z-1}]^T$ 와 같은  $m_b$  열 벡터로 분할 수 있다.

[0045] 도4에 도시된 패리티 검사 행렬 H를 이용한 부호화는 반복(recursion) 과정과 최종화(finalization) 과정의 두 과정에 의해 진행된다.

[0046] 상기 반복 과정은  $p(i-1)$ 로부터  $p(i)$ 를 결정하는 과정으로서, 다음의 수학식3에 의해 표현될 수 있다.

[0047] [수학식 3]

[0048] 
$$p(i) = p(i-1) + \sum_{j=0}^{k_b-1} P_{i,j} s(j), \quad i = 0, 1, \dots, m_b - 4$$
, 여기서,  $p(-1) \equiv 0_{z \times 1}$

[0049] 상기 수학식 3은 첫 번째 패리티 벡터(parity vector)  $p(0)$ 에서부터 끝에서 네번째 패리티 벡터  $p(m_b-4)$ 까지 간단한 반복 계산을 통해 순차적으로 얻는 과정을 설명한 것이다. 도5 내지 도7에 도시된 바와 같은  $(H_b)_p$ 의 변형 형태에 따라서는 패리티 벡터  $p(m_b-1)$ 로부터 시작하는 것도 가능하다. 상기 수학식 3과 같은 방식에 의하면 초기화(initialization) 과정이 필요하지 않아 처음 패리티 벡터 부분을 전송시키고 나머지 부분은 전송하지 않는 경우(예를 들어, puncturing, H-ARQ, IR operation)에 있어 유리하다.

[0050] 상기 최종화 과정은  $p(m_b-3)$ ,  $p(m_b-2)$  및  $p(m_b-1)$ 를 구하는 과정으로서, 다음의 수학식4에 의해 표현될 수 있다.

[0051] [수학식 4]

[0052] 
$$P_{m_b-3, n_b-3} p(m_b-3) = p(m_b-4) + \sum_{i=m_b-3}^{m_b-1} \sum_{j=0}^{k_b-1} P_{i,j} s(j),$$
  

$$p(i) = p(m_b-3) + \sum_{j=0}^{k_b-1} P_{i,j} s(j), i = m_b-2, m_b-1$$

[0053] 상기 수학식 4는 마지막 3개의 패리티 벡터를 얻는 과정으로 종래기술에 있어 무게가 1인 다수의 열(column)을 제거하기 위해 추가된 단위 행렬로 인해 생긴 과정이다. 이 과정을 통해 마지막 3개의 패리티 벡터를 얻을 수 있고 부호화를 최종화(finalization)하게 된다.

[0054] 도5는 본 발명의 바람직한 다른 실시예에 따른 기본 행렬  $H_b$ 의 형태에 의해 표현된 패리티 검사 행렬 H의 구조를

예시하여 도시한 것이다. 도5에서  $(H_b)_p$  는 도4에 도시된  $(H_b)_p$  에서 오른쪽 끝에서 두 번째 열과 세 번째 열을 교환한 구조를 갖는다. 도5와 같은 구조를 갖게 한 이유는 도4의 경우와 동일하며, 도5에 도시된 기본 행렬  $H_b$ 를 이용한 부호화 과정은 반복 과정과 최종화 과정의 두 과정에 의해 진행된다.

[0055] 상기 반복 과정은  $p(i-1)$ 로부터  $p(i)$ 를 결정하는 과정으로서, 다음의 수학식5에 의해 표현될 수 있다.

[0056] [수학식 5]

[0057] 
$$p(i) = p(i-1) + \sum_{j=0}^{k_b-1} P_{i,j} s(j), \quad i = 0, 1, \dots, m_b - 4$$
, 여기서  $p(-1) \equiv 0_{z \times 1}$

[0058] 상기 최종화 과정은  $p(m_b-3)$ ,  $p(m_b-2)$  및  $p(m_b-1)$ 를 구하는 과정으로서, 다음의 수학식6에 의해 표현될 수 있다.

[0059] [수학식 6]

[0060] 
$$P_{m_b-3, m_b-2} p(m_b - 2) = p(m_b - 4) + \sum_{i=m_b-3}^{m_b-1} \sum_{j=0}^{k_b-1} P_{i,j} s(j),$$

[0061] 
$$p(i) = p(m_b - 2) + \sum_{j=0}^{k_b-1} P_{i,j} s(j), i = m_b - 3, m_b - 1$$

[0061] 상기 수학식 5 및 수학식 6이 의미하는 바는 상기 수학식 3 및 수학식 4에서 설명한 것과 유사하다.

[0062] 도6은 본 발명의 바람직한 또 다른 실시예에 따른 기본 행렬  $H_b$ 의 형태에 의해 표현된 패리티 검사 행렬  $H$ 의 구조를 예시하여 도시한 것이다. 도6의  $(H_b)_p$  부분을 도2의  $(H_b)_p$  부분과 비교해 보면 왼쪽 위의  $3 \times 3$  부분(도6에서 해칭한 부분)의 구조가 차이가 있음을 알 수 있다. 도6과 같은 구조를 갖게 한 이유는 도4의 경우와 동일하다.

[0063] 도6에 도시된 기본 행렬  $H_b$ 를 이용한 부호화 과정은 반복 과정과 최종화 과정의 두 과정에 의해 진행된다.

[0064] 상기 반복 과정은  $p(i)$ 로부터  $p(i-1)$ 를 결정하는 과정으로서, 다음의 수학식7에 의해 표현될 수 있다.

[0065] [수학식 7]

[0066] 
$$p(i-1) = p(i) + \sum_{j=0}^{k_b-1} P_{i,j} s(j), \quad i = 4, 5, \dots, m_b$$
, 여기서  $p(m_b) \equiv 0_{z \times 1}$

[0067] 상기 최종화 과정은  $p(0)$ ,  $p(1)$  및  $p(2)$ 를 구하는 과정으로서, 다음의 수학식8에 의해 표현될 수 있다.

[0068] [수학식 8]

[0069] 
$$P_{2,0} p(0) = p(3) + \sum_{i=0}^2 \sum_{j=0}^{k_b-1} P_{i,j} s(j),$$

[0070] 
$$p(i) = p(0) + \sum_{j=0}^{k_b-1} P_{i,j} s(j), i = 1, 2$$

[0070] 상기 수학식 7 및 수학식 8 에서 적용하는 방식은, 이 상기 수학식 3 및 수학식 4 와 같은 형태로도 적용이 가능하다.

[0071] 도7은 본 발명의 바람직한 또 다른 실시예에 따른 기본 행렬  $H_b$ 의 형태에 의해 표현된 패리티 검사 행렬  $H$ 의 구조를 예시하여 도시한 것이다. 도7에서  $(H_b)_p$  는 도6에 도시된  $(H_b)_p$  에서 오른쪽 끝에서 두 번째 열과 세 번째 열을 교환한 구조를 갖는다. 도7과 같은 구조를 갖게 한 이유는 도4의 경우와 동일하며, 도7에 도시된 기본 행렬  $H_b$ 를 이용한 부호화 과정은 반복 과정과 최종화 과정의 두 과정에 의해 진행된다.

[0072] 상기 반복 과정은  $p(i)$ 로부터  $p(i-1)$ 를 결정하는 과정으로서, 다음의 수학식9에 의해 표현될 수 있다.

[0073] [수학식 9]

$$p(i-1) = p(i) + \sum_{j=0}^{k_b-1} P_{i,j} s(j), \quad i = 4, 5, \dots, m_b$$

, 여기서  $p(m_b) \equiv 0_{z \times 1}$

[0074]

[0075] 상기 최종화 과정은  $p(0)$ ,  $p(1)$  및  $p(2)$ 를 구하는 과정으로서, 다음의 수학식10에 의해 표현될 수 있다.

[0076]

[수학식 10]

$$P_{3,2} p(1) = p(3) + \sum_{i=0}^2 \sum_{j=0}^{k_b-1} P_{i,j} s(j),$$

$$p(i) = p(1) + \sum_{j=0}^{k_b-1} P_{i,j} s(j), i = 0, 3$$

[0077]

[0078] 상기 수학식 9 및 수학식 10이 의미하는 바는 상기 수학식 3 및 수학식 4에서 설명한 것과 유사하다.

[0079]

도8은 본 발명의 바람직한 또 다른 실시예에 따른 기본 행렬  $H_b$ 의 형태에 의해 표현된 패리티 검사 행렬  $H$ 의 구조를 예시하여 도시한 것이다. 도8의 예는 도4 내지 도7에 도시된 예들과는 달리 단위 행렬(identity matrix)을  $90^\circ$  회전시켜 행성한 서브행렬이 없고, 상기 기본 행렬  $H_b$ 를 기준으로  $(H_b)_p$ 의 모든 단위 행렬을 '1'로 보았을 때 6-싸이클(6-cycle)이 존재하지 않도록 구성한 것이다.

[0080]

도8에 도시된 패리티 검사 행렬  $H$ 를 이용한 부호화 과정은 다음의 수학식11에 의한 반복(recursion)과정과 수학식12에 의한 최종화(finalization) 과정의 두 과정에 의해 진행된다.

[0081]

[수학식 11]

$$p(i) = p(i-1) + \sum_{j=0}^{k_b-1} P_{i,j} s(j), \quad i = 0, 1, \dots, m_b - 5$$

, 여기서  $p(-1) \equiv 0_{z \times 1}$

[0082]

[0083] 상기 수학식 11은 첫 번째 패리티 벡터(parity vector)  $p(0)$ 에서부터 끝에서 여섯 번째 패리티 벡터  $p(m_b-5)$ 까지 간단한 반복 계산을 통해 순차적으로 얻는 과정을 설명한 것이다.

[0084]

[수학식 12]

$$p(m_b - 2) = p(m_b - 5) + \sum_{i=m_b-5}^{m_b-1} \sum_{j=0}^{k_b-1} P_{i,j} s(j),$$

$$\sum_{j=m_b-5}^{m_b-1} P_{i, k_b+j} p(j) = \sum_{j=0}^{k_b-1} P_{i,j} s(j), i = m_b - 1, m_b - 3, \text{ 및 } m_b - 4$$

[0085]

[0086] 상기 최종화 과정은  $p(m_b-4)$ ,  $p(m_b-3)$ ,  $p(m_b-2)$  및  $p(m_b-1)$ 를 구하는 과정으로서, 이 과정을 통해 마지막 5개의 패리티 벡터를 얻을 수 있고 부호화(encoding)를 최종화(finalization)하게 된다.

[0087]

도9는 본 발명의 바람직한 또 다른 실시예에 따른 기본 행렬  $H_b$ 의 형태에 의해 표현된 패리티 검사 행렬  $H$ 의 구조를 예시하여 도시한 것이다. 도9에 도시된 예는 도8에 도시된 예와 유사하게 단위 행렬(identity matrix)을  $90^\circ$  회전시켜 행성한 서브행렬이 없고, 상기 기본 행렬  $H_b$ 를 기준으로  $(H_b)_p$ 의 모든 단위 행렬을 '1'로 보았을 때 6-싸이클(6-cycle)이 존재하지 않도록 구성한 것이다.

[0088]

도9에 도시된 패리티 검사 행렬  $H$ 를 이용한 부호화 과정은 다음의 수학식13에 의한 반복(recursion)과정과 수학식14에 의한 최종화(finalization) 과정의 두 과정에 의해 진행된다. 상기 수학식 13 및 수학식 14가 의미하는 바는 각각 상기 수학식 11 및 수학식 12에서 설명한 것과 유사하다.

[0089]

[수학식 13]

$$p(i) = p(i+1) + \sum_{j=0}^{k_b-1} P_{i,j} s(j), \quad i = m_b - 1, \dots, 6, 5$$

여기서  $p(m_b) \equiv 0_{z \times 1}$

[0090]

[0091] [수학식 14]

$$p(1) = p(5) + \sum_{i=0}^4 \sum_{j=0}^{k_b-1} P_{i,j} s(j),$$

$$\sum_{j=0}^4 P_{i,k_b+j} p(j) = \sum_{j=0}^{k_b-1} P_{i,j} s(j), i = 0, 2, 3, 4$$

[0092]

[0093] 도3에서 상기 수신기(30)가 상기한 바와 같은 방법으로 부호화된 데이터를 수신하여 복호함에 있어서는 다음의 수학식15를 이용한다.

[0094] [수학식 15]

[0095]  $Hx = 0$

[0096] 즉, 부호화된 데이터 x와 상기 패리티 검사 행렬(parity check matrix) H를 곱하여 0가 되면 전송 에러가 없다는 것을 의미하고, 0가 되지 않으면 전송 에러가 존재한다는 것을 의미하므로 이에 따라 소스 데이터를 분리해 낼 수 있다.

[0097] 도4 내지 도7에 기본 행렬  $H_b$  형식으로 도시된 상기 패리티 검사 행렬 H는 본 발명의 바람직한 실시예들을 설명하기 위한 예시적 구조에 불과한 것으로서, 실제로 부호화 또는 복호화 시에 사용되는 패리티 검사 행렬 H는 그 크기가 훨씬 크다. 또한, 본 발명은  $H=[H_d | H_p]$ 의 구성을 갖는 패리티 검사 행렬 H에서  $H_p$  부분을 개량한 것으로서,  $H_d$  부분은 다양한 형태로 구성할 수 있음은 자명하다.

[0098] 실제로 패리티 검사 행렬 H에 의해 정의되는 LDPC 코드를 이용하여 부호화 또는 복호화를 함에 있어서, 기본 퍼뮤테이션 행렬과 상기 기본 퍼뮤테이션 행렬을 일정한 규칙에 의해 변형시켜 형성된 다수의 퍼뮤테이션 행렬들의 퍼뮤테이션 타입을 기본 행렬(base matrix)  $H_b$ 를 저장한 상태에서 부호화 또는 복호화가 필요할 때마다 상기 기본 행렬  $H_b$ 를 확장하여(expanding) 패리티 검사 행렬(parity check matrix) H를 생성하고, 상기 생성된 패리티 검사 행렬을 이용하여 부호화 또는 복호화를 수행하는 것이 바람직하다.

[0099] 본 발명에 따르면 상기 패리티 검사 행렬 H의  $H_p$  부분에 무게가 1인 열이 생기지 않도록 함으로써 부호화 또는 복호화 시에 간단한 반복 연산에 의해 부호화 또는 복호화를 수행할 수 있는 효과가 있다. 이에 따라, LDPC 코드를 이용한 부호화 또는 복호화 시에 초기 복잡도(initialization complexity)를 낮출 수 있어 성능을 향상시킬 수 있다.

[0100] 본 발명은 본 발명의 정신 및 필수적 특징을 벗어나지 않는 범위에서 다른 특정한 형태로 구체화될 수 있음은 당업자에게 자명하다. 따라서, 상기의 상세한 설명은 모든 면에서 제한적으로 해석되어서는 아니되고 예시적인 것으로 고려되어야 한다. 본 발명의 범위는 첨부된 청구항의 합리적 해석에 의해 결정되어야 하고, 본 발명의 등가적 범위 내에서의 모든 변경은 본 발명의 범위에 포함된다.

**도면**

**도면1**

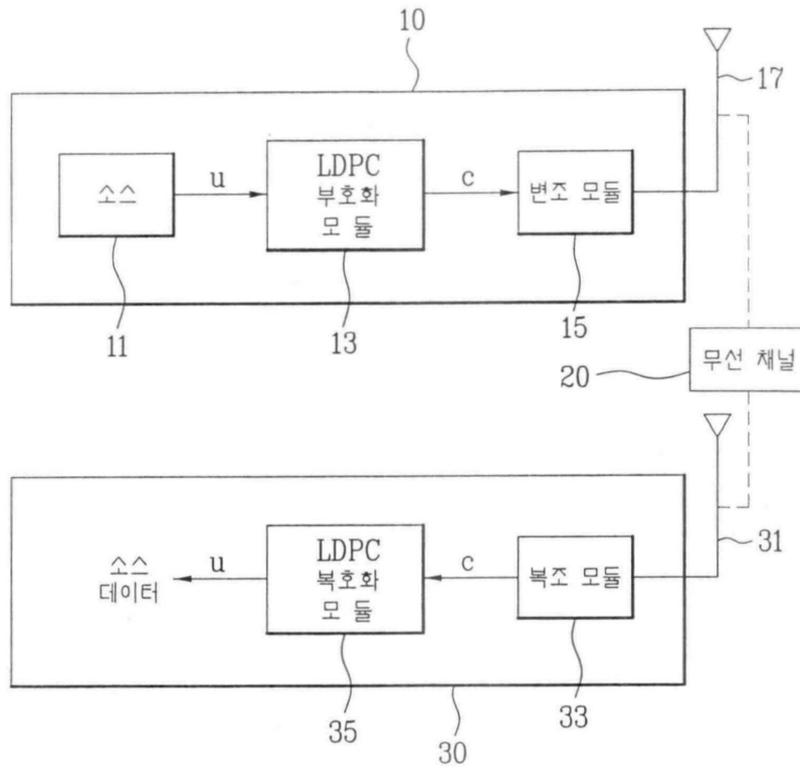
$$H = \begin{bmatrix} P_{0,0} & P_{0,1} & P_{0,2} & \cdots & P_{0,n_b-2} & P_{0,n_b-1} \\ P_{1,0} & P_{1,1} & P_{1,2} & \cdots & P_{1,n_b-2} & P_{1,n_b-1} \\ P_{2,0} & P_{2,1} & P_{2,2} & \cdots & P_{2,n_b-2} & P_{2,n_b-1} \\ \vdots & \vdots & \vdots & \cdots & \vdots & \vdots \\ P_{m_b-1,0} & P_{m_b-1,1} & P_{m_b-1,2} & \cdots & P_{m_b-1,n_b-2} & P_{m_b-1,n_b-1} \end{bmatrix} = P^{H_b}$$

도면2

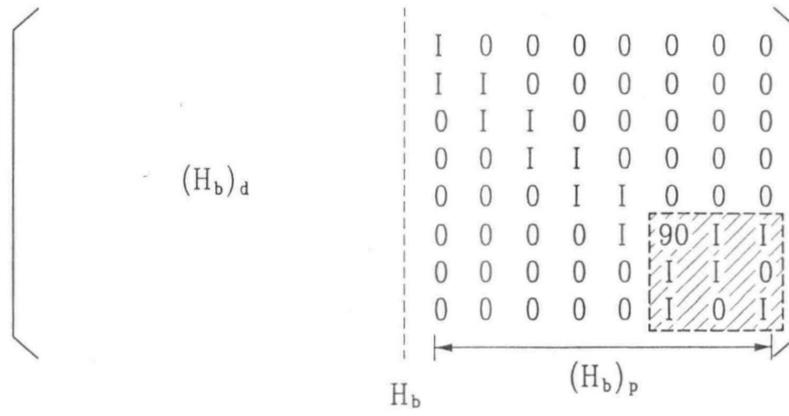
1	2	0	3	4	5	0	1	1	0	0	0	0	0	0	0	0
4	3	1	5	2	1	5	2	1	1	0	0	0	0	0	0	0
1	5	1	3	1	0	3	4	0	1	1	0	0	0	0	0	0
0	2	1	2	0	4	1	1	0	0	1	1	0	0	0	0	0
3	5	0	1	2	2	3	4	0	0	0	1	1	0	0	0	0
3	2	3	4	0	1	2	3	0	0	0	0	1	1	0	0	0
0	4	5	2	4	3	4	1	0	0	0	0	0	1	1	0	0
5	2	4	1	3	0	2	0	0	0	0	0	0	0	1	1	1

$H_b$

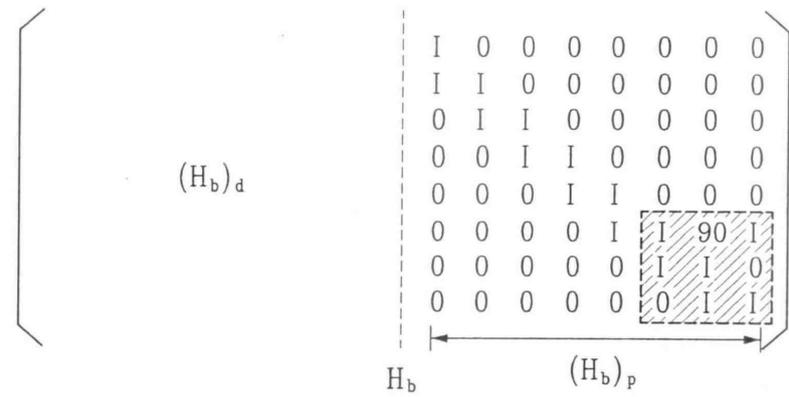
도면3



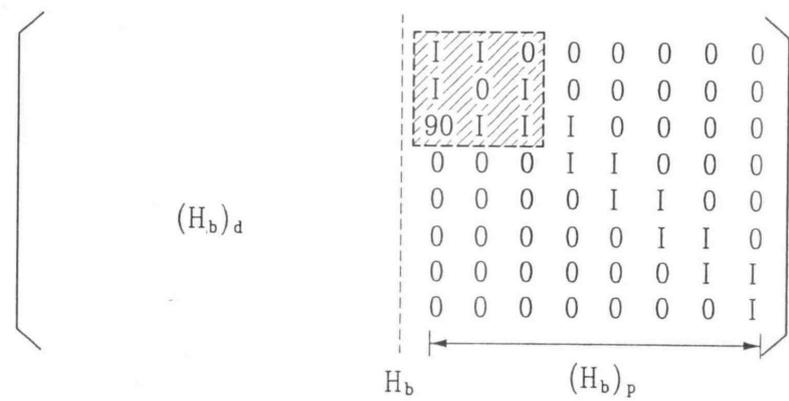
도면4



도면5



도면6



도면7

