Office de la Propriété Intellectuelle du Canada
Un organisme d'Industrie Canada

Canadian Intellectual Property Office
An agency of Industry Canada

CA 2977214 A1 2016/09/09

(21) 2 977 214

(12) DEMANDE DE BREVET CANADIEN
CANADIAN PATENT APPLICATION

(13) A1

(86) Date de dépôt PCT/PCT Filing Date: 2016/03/02
(87) Date publication PCT/PCT Publication Date: 2016/09/09
(85) Entrée phase nationale/National Entry: 2017/08/18
(86) N° demande PCT/PCT Application No.: KR 2016/002086
(87) N° publication PCT/PCT Publication No.: 2016/140509
(30) Priorités/Priorities: 2015/03/02 (US62/127,023); 2015/09/27 (KR10-2015-0137183)

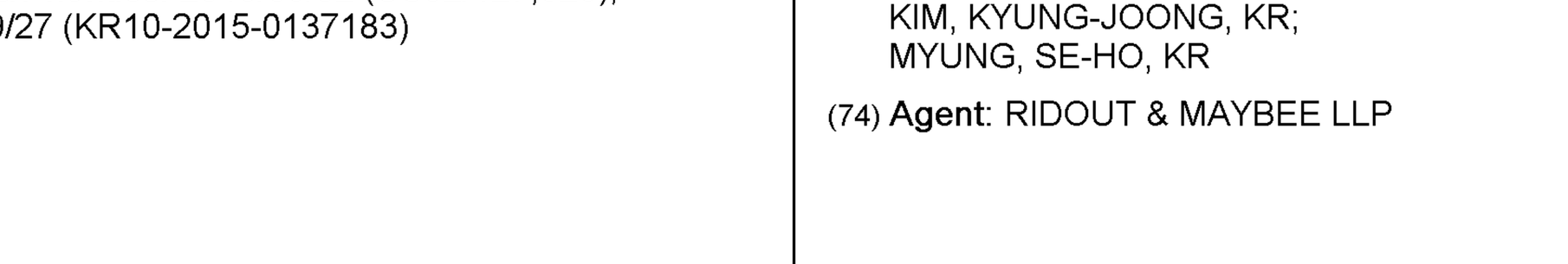
(51) Cl.Int./Int.Cl. *H04L 1/22* (2006.01), *H03M 13/11* (2006.01), *H03M 13/15* (2006.01), *H03M 13/29* (2006.01)

(71) Demandeur/Applicant:
SAMSUNG ELECTRONICS CO., LTD., KR

(72) Inventeurs/Inventors:
JEONG, HONG-SIL, KR;
KIM, KYUNG-JOONG, KR;
MYUNG, SE-HO, KR

(74) Agent: RIDOUT & MAYBEE LLP

(54) Titre : EMETTEUR ET SON PROCEDE DE RACCOURCISSEMENT
(54) Title: TRANSMITTER AND SHORTENING METHOD THEREOF



(57) Abrégé/Abstract:
A transmitter is provided. The transmitter includes: an outer encoder configured to encode input bits to generate outer-encoded bits including the input bits and parity bits; a zero padder configured to generate a plurality of bit groups each of which is formed of a same number of bits, determine whether a number of the outer-encoded bits satisfies a predetermined number of bits required according to at least one of a code rate and a code length for Low Density Parity Check (LDPC) encoding, pads zero bits to some of the bits in the bit groups if the number of the outer-encoded bits is less than the predetermined number of bits, and maps the outer-encoded bits to remaining bits in the bit groups, based on a predetermined shortening pattern, thereby to constitute LDPC information bits; and an LDPC encoder configured to encode the LDPC information bits, wherein the some of the bits, in which zero bits are padded, are included in some of the bit groups which are not sequentially disposed in the LDPC information bits.

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
9 September 2016 (09.09.2016)

WIPO | PCT

(10) International Publication Number
WO 2016/140509 A1

(51) International Patent Classification:
H03M 13/15 (2006.01) *H03M 13/11* (2006.01)

(21) International Application Number:
PCT/KR2016/002086

(22) International Filing Date:
2 March 2016 (02.03.2016)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
62/127,023 2 March 2015 (02.03.2015) US
10-2015-0137183
27 September 2015 (27.09.2015) KR

(71) Applicant: **SAMSUNG ELECTRONICS CO., LTD.** [KR/KR]; 129, Samsung-ro, Yeongtong-gu, Suwon-si, Gyeonggi-do 16677 (KR).

(72) Inventors: **JEONG, Hong-sil**; 5301-1801, 23, Docheong-ro 17beon-gil, Yeongtong-gu, Suwon-si, Gyeonggi-do 16509 (KR). **KIM, Kyung-joong**; 403, 316, Digital-ro, Guro-gu, Seoul 08378 (KR). **MYUNG, Se-ho**; 706-102, 126, Heungdeok 2-ro, Giheung-gu, Yongin-si, Gyeonggi-do 16953 (KR).

(74) Agent: **JEONG, Hong-sik**; 9th Floor, Shinduk Bldg., 343, Gangnamdaero, Seocho-gu, Seoul 06626 (KR).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

— *with international search report (Art. 21(3))*

— *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments (Rule 48.2(h))*

(54) Title: TRANSMITTER AND SHORTENING METHOD THEREOF

(57) Abstract: A transmitter is provided. The transmitter includes: an outer encoder configured to encode input bits to generate outer-encoded bits including the input bits and parity bits; a zero padder configured to generate a plurality of bit groups each of which is formed of a same number of bits, determine whether a number of the outer-encoded bits satisfies a predetermined number of bits required according to at least one of a code rate and a code length for Low Density Parity Check (LDPC) encoding, pads zero bits to some of the bits in the bit groups if the number of the outer-encoded bits is less than the predetermined number of bits, and maps the outer-encoded bits to remaining bits in the bit groups, based on a predetermined shortening pattern, thereby to constitute LDPC information bits; and an LDPC encoder configured to encode the LDPC information bits, wherein the some of the bits, in which zero bits are padded, are included in some of the bit groups which are not sequentially disposed in the LDPC information bits.

WO 2016/140509 A1

Description

Title of Invention: TRANSMITTER AND SHORTENING METHOD THEREOF

Technical Field

[1] Apparatuses and methods consistent with the exemplary embodiments of the inventive concept relate to a transmitter and a shortening method thereof, and more particularly, to a transmitter performing shortening by padding zero bits and a shortening method thereof.

Background Art

[2] Broadcast communication services in information oriented society of the 21st century are entering an era of digitalization, multi-channelization, bandwidth broadening, and high quality. In particular, as a high definition digital television (TV) and portable broadcasting signal reception devices are widespread, digital broadcasting services have an increased demand for a support of various receiving schemes.

[3] According to such demand, standard groups set up broadcasting communication standards to provide various signal transmission and reception services satisfying the needs of a user. Still, however, a method for providing better services to a user with more improved performance is required.

Disclosure of Invention

Technical Problem

[4] The exemplary embodiments of the inventive concept may overcome disadvantages of the related art signal transmitter and receiver and methods thereof. However, these embodiments are not required to or may not overcome such disadvantages.

[5] The exemplary embodiments provide a transmitter performing shortening based on a preset shortening pattern and a shortening method thereof.

Solution to Problem

[6] According to an aspect of an exemplary embodiment, there is provided a transmitter which may include: an outer encoder configured to encode input bits to generate outer-encoded bits including the input bits and parity bits; a zero padder configured to generate a plurality of bit groups each of which is formed of a same number of bits, determine whether a number of the outer-encoded bits satisfies a predetermined number of bits required according to at least one of a code rate and a code length for Low Density Parity Check (LDPC) encoding, pads zero bits to some of the bits in the bit groups if the number of the outer-encoded bits is less than the predetermined number of bits, and maps the outer-encoded bits to remaining bits in the bit groups, based on a predetermined shortening pattern, thereby to constitute LDPC information

bits; and an LDPC encoder configured to encode the LDPC information bits, wherein the some of the bits, in which zero bits are padded, are included in some of the bit groups which are not sequentially disposed in the LDPC information bits.

[7] According to an aspect another exemplary embodiment, there is provied a method of shortening bits. The method may include: encoding input bits to generate outer-encoded bits including the input bits and parity bits; generating a plurality of bit groups each of which is formed of a same number of bits, determining whether a number of the outer-encoded bits satisfies a predetermined number of bits required according to at least one of a code rate and a code length for Low Density Parity Check (LDPC) encoding, padding zero bits to some of the bits in the bit groups if the number of the outer-encoded bits is less than the predetermined number of bits, and mapping the outer-encoded bits to remaining bits in the bit groups, based on a predetermined shortening pattern, thereby to constitute LDPC information bits; and LDPC-encoding the LDPC information bits, wherein the some of the bits, in which zero bits are padded, are included in some of the bit groups which are not sequentially disposed in the LDPC information bits.

Advantageous Effects of Invention

[8] As described above, according to various exemplary embodiments of the inventive concept, shortening may be performed based on a preset shortening pattern to position LDPC information bits at specific position, thereby improving performance of a bit error rate (BER) and a frame error rate (FER).

Brief Description of Drawings

[9] The above and/or other aspects of the exemplary embodiments will be described herein with reference to the accompanying drawings, in which:

[10] FIG. 1 is a block diagram for describing a configuration of a transmitter, according to an exemplary embodiment;

[11] FIG. 2 is a diagram for describing a shortening procedure, according to an exemplary embodiment;

[12] FIGs. 3 and 4 are diagrams for describing parity check matrices, according to an exemplary embodiment;

[13] FIG. 5 is a diagram illustrating a parity check matrix having a quasi cyclic structure, according to an exemplary embodiment;

[14] FIG. 6 is a diagram for describing a frame structure, according to an exemplary embodiment;

[15] FIGs. 7 and 8 are block diagrams for describing detailed configurations of a transmitter, according to exemplary embodiments;

[16] FIGs. 9 to 22 are diagrams for describing methods for processing signaling,

according to exemplary embodiments;

[17] FIGs. 23 and 24 are block diagrams for describing configurations of a receiver, according to exemplary embodiments;

[18] FIGs. 25 and 26 are diagrams for describing examples of combining Log Likelihood Ratio (LLR) values of a receiver, according to exemplary embodiments;

[19] FIG. 27 is a diagram illustrating an example of providing information on a length of L1 signaling, according to an exemplary embodiment; and

[20] FIG. 28 is a flow chart for describing a shortening method, according to an exemplary embodiment.

Best Mode for Carrying out the Invention

[21] -

Mode for the Invention

[22] Hereinafter, exemplary embodiments of the inventive concept will be described in more detail with reference to the accompanying drawings.

[23] FIG. 1 is a block diagram for describing a configuration of a transmitter according to an exemplary embodiment.

[24] Referring to FIG. 1, a transmitter 100 includes an outer encoder 110, a zero padder 120 and a Low Density Parity Check (LDPC) encoder 130.

[25] The outer encoder 110 encodes input bits to generate parity bits (or parity check bits).

[26] Here, the input bits may be information bits of signaling (alternatively referred to as signaling bits or signaling information). For example, the information bits may include information required for a receiver 200 (as illustrated in FIGs. 23 or 24) to receive and process data or service data (for example, broadcasting data) transmitted from the transmitter 100.

[27] The outer encoding is a coding operation which is performed before inner encoding in a concatenated coding operation, and may use various encoding schemes such as Bose, Chaudhuri, Hocquenghem (BCH) encoding and/or cyclic redundancy check (CRC) encoding. In this case, an inner code for inner encoding may be an LDPC code.

[28] For example, the outer encoder 110 may perform outer encoding on input K_{sig} bits to generate M_{outer} parity bits, and add the parity bits to the input bits to output outer-encoded bits formed of $N_{outer}(=K_{sig}+M_{outer})$ bits. In this case, the outer-encoded bits may include the input bits and the parity bits.

[29] For convenience of explanation, the outer encoding will be described below under an assumption that it is performed by a BCH code and BCH encoding.

[30] That is, the BCH encoder 110 performs the encoding, that is, the Bose BCH encoding, on the input bits to generate the parity check bits, that is, BCH parity-check

bits (or, BCH parity bits).

[31] For example, the BCH encoder 110 may systematically perform the BCH encoding on the input K_{sig} bits to generate M_{outer} parity check bits, that is, BCH parity-check bits, and add the BCH parity-check bits to the input bits to output BCH encoded bits formed of $N_{outer}(=K_{sig}+M_{outer})$ bits, that is, the BCH encoded bits including the input bits and the BCH parity-check bits. In this case, $M_{outer}=168$.

[32] The zero padder 120 configures LDPC information bits including the outer-encoded bits, which includes the input bits and the parity bits, and zero bits (that is, bits having a 0 value). Further, the zero padder 120 may output the LDPC information bits to the LDPC encoder 130.

[33] For an LDPC code and LDPC encoding, a specific number of LDPC information bits depending on a code rate and a code length are required. Therefore, when the number of BCH encoded bits is less than the number of information bits required for LDPC encoding, the zero padder 120 may pad an appropriate number of zero bits to obtain the required number of LDPC information bits. Therefore, the BCH encoded bits and the padded zero bits may configure the LDPC information bits as many as the number of bits required for the LDPC encoding.

[34] Since the padded zero bits are bits required to obtain the specific number of bits for the LDPC encoding, the padded zero bits are LDPC-encoded, and then, are not transmitted to the receiver 200. As such, a procedure of padding the zero bits or a procedure of padding zero bits and then not transmitting the padded zero bits to the receiver 200 may be referred to as shortening. In this case, the padded zero bits may be referred to as shortening bits (or shortened bits).

[35] For example, when the number N_{outer} of BCH encoded bits is less than the number K_{ldpc} of LDPC information bits required for LDPC encoding, the transmitter 100 may pad $K_{ldpc}-N_{outer}$ zero bits to some of LDPC information bits to generate LDPC information bits formed of K_{ldpc} bits. Therefore, $K_{ldpc}-N_{outer}$ zero bits are added to $K_{sig}+M_{outer}$ BCH encoded bits to generate $K_{sig}+M_{outer}+K_{ldpc}-N_{outer}$ LDPC information bits.

[36] For this purpose, the zero padder 120 may divide the LDPC information bits into a plurality of bit groups.

[37] In detail, the zero padder 120 may divide the LDPC information bits into the plurality of bit groups so that the number of bits included in each bit group is 360.

[38] For example, the zero padder 120 may divide K_{ldpc} LDPC information bits $(i_0, i_1, \ldots, i_{K_{ldpc}-1})$ into $N_{info_group}(=K_{ldpc}/360)$ bit groups based on following Equation 1 or 2.

[39]

$$Z_j = \left\{ i_k \middle| j = \left\lfloor \frac{k}{360} \right\rfloor, 0 \le k < K_{ldpc} \right\} \text{ for } 0 \le j < N_{info_group} \quad \ldots. (1)$$

[40] .... (2)

$$Z_j = \{ i_k \mid 360 \times j \leq k < 360 \times (j+1) \} \text{ for } 0 \leq j < N_{info_group}$$

[41] In above Equations 1 and 2, Z_j represents a j-th bit group. Here, $\lfloor x \rfloor$ represents a maximum integer which is not greater than x.

[42] Meanwhile, FIG. 2 illustrates an example in which LDPC information bits are divided into a plurality of bit groups, according to an exemplary embodiment. However, FIG. 2 illustrates LDPC information bits and LDPC parity bits (that is, LDPC FEC) generated by performing LDPC encoding on the LDPC information bits together.

[43] Referring to FIG. 2, K_{ldpc} LDPC information bits are divided into the N_{info_group} bits groups and each bit group Z_j is formed of 360 bits.

[44] For example, it is assumed that the number K_{ldpc} of LDPC information bits is 6480. In this case, if the LDPC information bits are divided into a plurality of groups so that the number of bits included in each bit group is 360, the LDPC information bits may be divided into 18 (=6480/360) bit groups.

[45] Hereinafter, a shortening procedure performed by the zero padder 120 will be described in more detail.

[46] The zero padders 120 may calculate the number of zero bits to be padded. That is, to fit the number of bits required for the LDPC encoding, the zero padder 120 may calculate the number of zero bits to be padded.

[47] In detail, the zero padder 120 may calculate a difference between the number of LDPC information bits required for the LDPC encoding and the number of BCH encoded bits as the number of padded zero bits. That is, when $N_{outer}(=K_{sig}+M_{outer})$ is less than K_{ldpc}, the zero padder 120 may calculate the number of zero bits to be padded as $(K_{ldpc}-N_{outer})$.

[48] Further, the zero padder 120 may calculate the number N_{pad} of bit groups in which all bits are to be padded by zero bits, based on following Equation 3 or 4.

[49] .... (3)

$$N_{pad} = \left\lfloor \frac{K_{ldpc} - N_{outer}}{360} \right\rfloor$$

[50] .... (4)

$$N_{pad} = \left\lfloor \frac{(K_{ldpc} - M_{outer}) - K_{sig}}{360} \right\rfloor$$

[51] Further, the zero padder 120 pads zero bits to at least some of a plurality of bit groups configuring the LDPC information bits, based on a shortening pattern.

[52] In detail, the zero padder 120 may determine bit groups in which zero bits are to be padded among the plurality of bit groups based on the shortening pattern, and may pad zero bits to all bits within some of the determined bit groups and some bits within the remaining bit groups.

[53] Here, the shortening pattern may be defined as following Table 1. In this case, following Table 1 shows the shortening pattern which is applied to a case in which the LDPC encoder 130 performs LDPC encoding on 6480 LDPC information bits at a code rate of 6/15 to generate 9720 LDPC parity bits.

[54] LDPC codeword bits except the padded zero bits in an LDPC codeword formed of the LDPC information bits and the LDPC parity bits may be transmitted to the receiver 200. In this case, the shortened LDPC codeword (that is, the LDPC codeword bits, except the shortened bits, which may also be referred to as the shortened LDPC codeword) may be modulated by quadrature phase shift keying (QPSK) to be transmitted to the receiver 200.

[55] [Table 1]

[56]

N_{info_group}	$\pi_S(j)$ $(0 \le j < N_{info_group})$								
	$\pi_S(0)$	$\pi_S(1)$	$\pi_S(2)$	$\pi_S(3)$	$\pi_S(4)$	$\pi_S(5)$	$\pi_S(6)$	$\pi_S(7)$	$\pi_S(8)$
	$\pi_S(9)$	$\pi_S(10)$	$\pi_S(11)$	$\pi_S(12)$	$\pi_S(13)$	$\pi_S(14)$	$\pi_S(15)$	$\pi_S(16)$	$\pi_S(17)$
18	0	12	15	13	2	5	7	9	8
	6	16	10	14	1	17	11	4	3

[57] Here, $\pi_s(j)$ represents a shortening pattern order of a j-th bit group. Further, N_{info_group} is the number of plural bit groups configuring LDPC information bits.

[58] In detail, the zero padder 120 may determine a bit group in which all bits within the bit group are padded by zero bits based on the shortening pattern, and pad zero bits to all bits of the determined bit group.

[59] That is, the zero padder 120 may determine $Z_{\pi_s(0)}, Z_{\pi_s(1)}, \cdots, Z_{\pi_s(N_{pad}-1)}$ as bit groups in which all bits are padded by zero bits based on the shortening pattern, and pad zero bits to all bits of the determined bit groups. That is, the zero padder 120 may pad zero bits to all bits of a $\pi_s(0)$-th bit group, a $\pi_s(1)$-th bit group, …, a $\pi_s(N_{pad}-1)$-th bit group among the plurality of bit groups based on the shortening pattern.

[60] As such, the zero padder 120 may determine N_{pad} bit groups, that is, $Z_{\pi_s(0)}, Z_{\pi_s(1)}, \cdots, Z_{\pi_s(N_{pad}-1)}$ based on the shortening pattern, and pad zero bits to all bits within the determined bit group.

[61] Meanwhile, since the total number of padded zero bits is $K_{ldpc}-N_{outer}$, and the number of bit groups in which all bits are padded by zero bits is N_{pad}, the zero padder 120 may additionally pad $K_{ldpc}-N_{outer}-360\times N_{pad}$ zero bits.

[62] In this case, the zero padder 120 may determine a bit group to which zero bits are additionally padded based on the shortening pattern, and may additionally pad zero bits from a beginning portion of the determined bit group.

[63] In detail, the zero padder 213 may determine $Z_{\pi_s(N_{pad})}$ as a bit group to which zero bits are additionally padded based on the shortening pattern, and may additionally pad zero bits to the K_{ldpc}-N_{outer}-360×N_{pad} bits positioned at the beginning portion of $Z_{\pi_s(N_{pad})}$. Therefore, the K_{ldpc}-N_{outer}-360×N_{pad} zero bits may be padded from a first bit of the $\pi_s(N_{pad})$-th bit group.

[64] Therefore, zero bits may be padded only to some of the $Z_{\pi_s(N_{pad})}$, and the K_{ldpc}-N_{outer}-360×N_{pad} zero bits may be padded from the first LDPC information bit of the $Z_{\pi_s(N_{pad})}$.

[65] Next, the zero padder 213 may map the BCH-encoded bits to the positions at which zero bits are not padded among the LDPC information bits to finally configure the LDPC information bits.

[66] Therefore, N_{outer} BCH-encoded bits may be sequentially mapped to the bit positions at which zero bits in the K_{ldpc} LDPC information bits (i_0, i_1, ..., $i_{K_{ldpc}-1}$).

[67] Hereinafter, a method for padding zero bits will be described in detail with reference to a case in which K_{ldpc}=6480 and N_{outer}=568 as an example. In this case, the LDPC information bits may be divided into 18 (=6480/360) bit groups.

[68] First, the zero padder 120 may calculate the number of zero bits to be padded to the LDPC information bits. In this case, the number of zero bits to be padded is 5912(=K_{ldpc}-N_{outer}=6480-568).

[69] Further, the zero padder 120 may calculate the number

$$16 = N_{pad} = \left\lfloor \frac{6480 - 568}{360} \right\rfloor$$

of bit groups in which all bits are padded by zero bits.

[70] Further, the zero padder 120 may determine $Z_0(=Z_{\pi_s(0)})$, $Z_{12}(=Z_{\pi_s(1)})$,..., $Z_{17}(=Z_{\pi_s(14)})$ and $Z_{11}(=Z_{\pi_s(15)})$ as bit groups in which all bits are padded by zero bits based on the shortening pattern, and pad zero bits to all bits of $Z_0(=Z_{\pi_s(0)})$, $Z_{12}(=Z_{\pi_s(1)})$,..., $Z_{17}(=Z_{\pi_s(14)})$ and $Z_{11}(=Z_{\pi_s(15)})$.

[71] Therefore, all bits of a sixth bit group, a 0-th bit group, a 12-th bit group, …, a 17-th

bit group and an 11-th bit group may be padded by zero bits.

[72] Further, the zero padder 120 may determine $Z_4 (= Z_{\pi_s(16)})$ as a bit group to which zero bits are additionally padded based on the shortening pattern and may additionally pad 152 ($=K_{ldpc}-N_{outer}-360\times N_{pad}=6480-568-360\times 16$) zero bits to the beginning portion of $Z_4 (= Z_{\pi_s(16)})$.

[73] Therefore, zero bits from a first bit to a 152-th bit may be padded to a 4-th bit group.

[74] As a result, the zero bits may be padded to all LDPC information bits of a 0-th bit group, a 12-th bit group, a 15-th bit group, a 13-th bit group, a 2-th bit group, a 5-th th group, a 7-th bit group, a 9th bit group, a 8-th bit group, a 6-th bit group, a 16-th bit group, a 10-th bit group, a 14-th bit group, a 1-th bit group, a 17-th bit group, and a 11-th bit group among 18 bit groups configuring the LDPC information bits, that is, a 0-th bit group to a 17-th bit group, and zero bits may be additionally padded from a first LDPC information bit to a 152-th LDPC information bit of a 4-th bit group.

[75] Next, the zero padder 120 may sequentially map the BCH-encoded bits to the bit positions at which zero bits are not padded in the LDPC information bits.

[76] For example, since the number N_{outer} of BCH encoded bits is 568, when the BCH encoded bits are $s_0, s_1, ..., s_{567}$, the zero padder 120 may map $s_0, s_1, ..., s_{207}$ to a 153-th LDPC information bit to a 360-th LDPC information bit of the 4-th bit group, and s_{208}, $s_{209}, ..., s_{567}$ to all LDPC information bits of a 3-th bit group.

[77] As such, the zero padder 120 may pad zero bits to appropriate positions to fit the number of bits required for LDPC encoding, thereby to finally configure the LDPC information bits for the LDPC encoding.

[78] The foregoing example describes that the information bits are outer-encoded, which is only one example. That is, the information bits may not be outer-encoded, and instead, may configure the LDPC information bits along with zero bits thereto depending on the number of information bits.

[79] The foregoing example describes that zero bits, which will be shortened, are padded, which is only one example. That is, since zero bits are bits having a value preset by the transmitter 100 and the receiver 200 and padded only to form LDPC information bits along with information bits including information to be substantially transmitted to the receiver 200, bits having another value (for example, 1) preset by the transmitter 100 and the receiver 200 instead of zero bits may be padded for shortening.

[80] The LDPC encoder 130 performs encoding, that is, LDPC encoding on the LDPC information bits.

[81] In detail, the LDPC encoder 130 may systematically perform the LDPC encoding on the LDPC information bits to generate LDPC parity bits, and output an LDPC

codeword (or LDPC-encoded bits) formed of the LDPC information bits and the LDPC parity bits. That is, an LDPC code for the LDPC encoding is a systematic code, and therefore, the LDPC codeword may be formed of the LDPC information bits before being LDPC-encoded and the LDPC parity bits generated by the LDPC encoding.

[82] For example, the LDPC encoder 130 may perform the LDPC encoding on K_{ldpc} LDPC information bits $i = (i_0, i_1, ..., i_{K_{ldpc}-1})$ to generate N_{ldpc_parity} LDPC parity bits $(p_0, p_1, ..., p_{N_{inner}-K_{ldpc}-1})$ and output an LDPC codeword $\Lambda = (c_0, c_1, ..., c_{N_{inner}-1}) = (i_0, i_1, ..., i_{K_{ldpc}-1}, p_0, p_1, ..., p_{N_{inner}-K_{ldpc}-1})$ formed of $N_{inner}(=K_{ldpc}+N_{ldpc_parity})$ bits.

[83] In this case, the LDPC encoder 110 may perform the LDPC encoding on the input bits (i.e., LDPC information bits) at various code rates to generate an LDPC codeword having a predetermined length.

[84] For example, the LDPC encoder 110 may perform LDPC encoding on 3240 input bits at a code rate of 3/15 to generate an LDPC codeword formed of 16200 bits. As another example, the LDPC encoder 110 may perform LDPC encoding on 6480 input bits at a code rate of 6/15 to generate an LDPC codeword formed of 16200 bits.

[85] Meanwhile, a process of performing the LDPC encoding is a process of generating an LDPC codeword to satisfy $H \cdot C^T=0$, and thus, the LDPC encoder 110 may use a parity check matrix to perform the LDPC encoding. Here, H represents the parity check matrix and C represents the LDPC codeword.

[86] Hereinafter, a structure of the parity check matrix according to various exemplary embodiments will be described with reference to the accompanying drawings. In the parity check matrix, elements of a portion other than 1 are 0.

[87] As one example, the parity check matrix according to an exemplary embodiment may have a structure as illustrated in FIG. 3.

[88] Referring to FIG. 3, a parity check matrix 20 may be formed of five sub-matrices A, B, C, Z and D. Hereinafter, for describing the structure of the parity check matrix 20, each matrix structure will be described.

[89] The sub-matrix A is formed of K columns and g rows, and the sub-matrix C is formed of K+g columns and N-K-g rows. Here, K (or K_{ldpc}) represents a length of LDPC information bits and N (or N_{inner}) represents a length of an LDPC codeword.

[90] Further, in the sub-matrices A and C, indexes of a row in which 1 is positioned in a 0-th column of an i-th column group may be defined based on Table 2 when the length of the LDPC codeword is 16200 and the code rate is 3/15. The number of columns belonging to a same column group may be 360.

[91] [Table 2]

[92]

8 372 841 4522 5253 7430 8542 9822 10550 11896 11988
80 255 667 1511 3549 5239 5422 5497 7157 7854 11267
257 406 792 2916 3072 3214 3638 4090 8175 8892 9003
80 150 346 1883 6838 7818 9482 10366 10514 11468 12341
32 100 978 3493 6751 7787 8496 10170 10318 10451 12561
504 803 856 2048 6775 7631 8110 8221 8371 9443 10990
152 283 696 1164 4514 4649 7260 7370 11925 11986 12092
127 1034 1044 1842 3184 3397 5931 7577 11898 12339 12689
107 513 979 3934 4374 4658 7286 7809 8830 10804 10893
2045 2499 7197 8887 9420 9922 10132 10540 10816 11876
2932 6241 7136 7835 8541 9403 9817 11679 12377 12810
2211 2288 3937 4310 5952 6597 9692 10445 11064 11272

[93] Hereinafter, positions (alternatively referred to as "indexes" or "index values") of a row in which 1 is positioned in the sub-matrices A and C will be described in detail with reference to, for example, Table 2.

[94] When the length of an LDPC codeword is 16200 and the code rate is 3/15, coding parameters M_1, M_2, Q_1 and Q_2 based on the parity check matrix 200 each are 1080, 11880, 3 and 33.

[95] Here, Q_1 represents a size at which columns belonging to a same column group in the sub-matrix A are cyclic-shifted, and Q_2 represents a size at which columns belonging to a same column group in the sub-matrix C are cyclic-shifted.

[96] Further, $Q_1 = M_1/L$, $Q_2 = M_2/L$, $M_1 = g$, $M_2 = N\text{-}K\text{-}g$ and L represents an interval at which patterns of a column are repeated in the sub-matrices A and C, respectively, that is, the number (for example, 360) of columns belonging to a same column group.

[97] The indexes of the row in which 1 is positioned in the sub-matrices A and C, respectively, may be determined based on an M_1 value.

[98] For example, in above Table 2, since M_1=1080, the position of a row in which 1 is positioned in a 0-th column of an i-th column group in the sub-matrix A may be determined based on values less than 1080 among index values of above Table 2, and the position of a row in which 1 is positioned in a 0-th column of an i-th column group in the sub-matrix C may be determined based on values equal to or greater than 1080 among the index values of above Table 2.

[99] In detail, a sequence corresponding to a 0-th column group in above Table 2 is "8 372 841 4522 5253 7430 8542 9822 10550 11896 11988". Therefore, in a 0-th column of a 0-th column group in the sub-matrix A, 1 may be positioned in an eighth row, a 372-th row, and an 841-th row, respectively, and in a 0-th column of a 0-th column group in the sub-matrix C, 1 may be positioned in a 4522-th row, a 5253-th row, a 7430-th row, an 8542-th row, a 9822-th row, a 10550-th row, a 11896-th row, and a 11988-row, respectively.

[100] In the sub-matrix A, when the position of 1 is defined in a 0-th columns of each column group, it may be cyclic-shifted by Q_1 to define a position of a row in which 1 is positioned in other columns of each column group, and in the sub-matrix C, when the position of 1 is defined in a 0-th columns of each column group, it may be cyclic-

shifted by Q_2 to define a position of a row in which 1 is positioned in other columns of each column group.

[101] In the foregoing example, in the 0-th column of the 0-th column group in the sub-matrix A, 1 is positioned in an eighth row, a 372-th row, and an 841-th row. In this case, since Q_1=3, indexes of a row in which 1 is positioned in a first column of the 0-th column group may be 11(=8+3), 375(=372+3), and 844(=841+3) and indexes of a row in which 1 is positioned in a second column of the 0-th column group may be 14(=11+3), 378(=375+3), and 847(= 844+3).

[102] In a 0-th column of a 0-th column group in the sub-matrix C, 1 is positioned in a 4522-th row, a 5253-th row, a 7430-th row, an 8542-th row, a 9822-th row, a 10550-th row, a 11896-th row, and a 11988-th row. In this case, since Q_2=33, the indexes of the row in which 1 is positioned in a first column of the 0-th column group may be 4555(=4522+33), 5286(=5253+33), 7463(=7430+33), 8575(=8542+33), 9855(=9822+33) 10583(=10550+33), 11929(=11896+33), and 12021(=11988+33) and the indexes of the row in which 1 is positioned in a second column of the 0-th column group may be 4588(=4555+33), 5319(=5286+33), 7496(=7463+33), 8608(=8575+33), 9888(=9855+33), 10616(=10583+33), 11962(=11929+33), and 12054(=12021+33).

[103] According to the scheme, the positions of the row in which 1 is positioned in all the column groups in the sub-matrices A and C may be defined.

[104] The sub-matrix B is a dual diagonal matrix, the sub-matrix D is an identity matrix, and the sub-matrix Z is a zero matrix.

[105] As a result, the structure of the parity check matrix 20 as illustrated in FIG. 2 may be defined by the sub-matrices A, B, C, D and Z having the above structure.

[106] Hereinafter, a method for performing, by the LDPC encoder 110, LDPC encoding based on the parity check matrix 20 as illustrated in FIG. 2 will be described.

[107] The LDPC code may be used to encode an information block $S = (s_0, s_1, ..., s_{K-1})$. In this case, to generate an LDPC codeword $\Lambda = (\lambda_0, \lambda_1, ..., \lambda_{N-1})$ having a length of $N=K+M_1+M_2$, parity blocks $P = (p_0, p_1, ..., p_{M_1+M_2-1})$ from the information block S may be systematically encoded.

[108] As a result, the LDPC codeword may be $\Lambda=(s0, s1, ..., s_{K-1}, p_0, p_1, ..., p_{M_1+M_2-1})$.

[109] Here, M_1 and M_2 each represent a size of parity sub-matrices corresponding to the dual diagonal sub-matrix B and the identity sub-matrix D, respectively, in which M_1=g and M_2=N-K-g.

[110] A process of calculating parity bits may be represented as follows. Hereinafter, for convenience of explanation, a case in which the parity check matrix 20 is defined as above Table 1 will be described as one example.

[111] Step 1) It is initialized to $\lambda_i=s_i$ (i=0, 1, ..., K-1), p_j=0 (j=0, 1, ..., M_1+M_2-1).

[112] Step 2) A first information bit λ_0 is accumulated in a parity bit address defined in the first row of above Table 1.

[113] Step 3) For the next L-1 information bits λ_m(m=1, 2, ..., L-1), λ_m is accumulated in the parity bit address calculated based on following Equation 5.

[114] $(x + m \times Q_1) \bmod M_1$ (if $x < M_1$)

[115] $M_1 + \{(x-M_1 + m \times Q_2) \bmod M_2\}$ (if $x \geq M_1$) .... (5)

[116] In above Expression 5, x represents an address of a parity bit accumulator corresponding to a first information bit λ_0.

[117] Further, $Q_1=M_1/L$ and $Q_2=M_2/L$. In this case, since the length of the LDPC codeword is 16200 and the code rate is 3/15, $M_1=1080$, $M_2=11880$, $Q_1=3$, $Q_2=33$, L=360.

[118] Step 4) Since the parity bit address like the second row of above Table 2 is given to an L-th information bit λ_L, similar to the foregoing scheme, the parity bit address for next L-1 information bits λ_m (m = L+1, L+2, ..., 2L-1) is calculated by the scheme described in the above step 3. In this case, x represents the address of the parity bit accumulator corresponding to the information bit λ_L and may be obtained based on the second row of above Table 2.

[119] Step 5) For L new information bits of each group, the new rows of above Table 2 are set as the address of the parity bit accumulator, and thus, the foregoing process is repeated.

[120] Step 6) After the foregoing process is repeated from the codeword bit λ_0 to λ_{K-1}, a value for following Equation 6 is sequentially calculated from i = 1.

[121] $P_i = P_i \oplus P_{i-1}$ (i=1,2, ... M_1 -1) .... (6)

[122] Step 7) The parity bits λ_K to λ_{K+M_1-1} corresponding to the dual diagonal sub-matrix B are calculated based on following Equation 7.

[123] $\lambda_{K+Lxt+s}=p_{Q1xs+t}$ $(0 \leq s < L, 0 \leq t < Q_1)$ .... (7)

[124] Step 8) The address of the parity bit accumulator for the L new codeword bits λ_K to λ_{K+M_1-1} of each group is calculated based on the new row of above Table 2 and above Equation 5.

[125] Step 9) After the codeword bits λ_K to λ_{K+M_1-1} are applied, the parity bits λ_{K+M_1} to $\lambda_{K+M_1+M_2-1}$ corresponding to the sub-matrix D are calculated based on following Equation 8.

[126] $\lambda_{K+M1+Lxt+s}=p_{M1+Q2xs+t}$ $(0 \leq s < L, 0 \leq t < Q_2)$ .... (8)

[127] As a result, the parity bits may be calculated by the above scheme. However, this is only one example and therefore the scheme for calculating the parity bits based on the parity check matrix as illustrated in FIG. 3 may be variously defined.

[128] As such, the LDPC encoder 130 may perform the LDPC encoding based on above Table 2 to generate the LDPC codeword.

[129] In detail, the LDPC encoder 130 may perform the LDPC encoding on 3240 input bits, that is, the LDPC information bits at the code rate of 3/15 based on above Table 2 to generate 12960 LDPC parity bits and output the LDPC parity bits and the LDPC codeword formed of the LDPC parity bits. In this case, the LDPC codeword may be formed of 16200 bits.

[130] As another example, the parity check matrix according to the exemplary embodiment may have a structure as illustrated in FIG. 4.

[131] Referring to FIG. 4, a parity check matrix 40 is formed of an information sub-matrix 41 which is a sub-matrix corresponding to the information bits (that is, LDPC information bits) and a parity sub-matrix 42 which is a sub-matrix corresponding to the parity bits (that is, LDPC parity bits).

[132] The information sub-matrix 41 includes K_{ldpc} columns and the parity sub-matrix 42 includes $N_{ldpc_parity}=N_{inner}-K_{ldpc}$ columns. The number of rows of the parity check matrix 40 is equal to the number $N_{ldpc_parity}=N_{inner}-K_{ldpc}$ of columns of the parity sub-matrix 42.

[133] Further, in the parity check matrix 40, N_{inner} represents the length of the LDPC codeword, K_{ldpc} represents the length of the information bits, and $N_{ldpc_parity}=N_{inner}-K_{ldpc}$ represents the length of the parity bits.

[134] Hereinafter, the structures of the information sub-matrix 41 and the parity sub-matrix 42 will be described.

[135] The information sub-matrix 41 is a matrix including the K_{ldpc} columns (that is, 0-th column to $(K_{ldpc}-1)$-th column) and depends on the following rule.

[136] First, the K_{ldpc} columns configuring the information sub-matrix 41 belong to the same group by M numbers and are divided into a total of K_{ldpc}/M column groups. The columns belonging to the same column group have a relationship that they are cyclic-shifted by Q_{ldpc} from one another. That is, Q_{ldpc} may be considered as a cyclic shift parameter value for columns of the column group in the information sub-matrix configuring the parity check matrix 40.

[137] Here, M represents an interval (for example, M=360) at which the pattern of columns in the information sub-matrix 41 is repeated and Q_{ldpc} is a size at which each column in the information sub-matrix 41 is cyclic-shifted. M is a common divisor of N_{inner} and K_{ldpc}, and is determined so that $Q_{ldpc} = (N_{inner}-K_{ldpc})/M$ is established. Here, M and Q_{ldpc} are integers and K_{ldpc}/M also becomes an integer. M and Q_{ldpc} may have various values depending on the length of the LDPC codeword and the code rate.

[138] For example, when M=360, the length N_{inner} of the LDPC codeword is 16200, and the code rate is 6/15, Q_{ldpc} may be 27.

[139] Second, if a degree (herein, the degree is the number of values 1s positioned in the

column and the degrees of all the columns belonging to the same column group are the same) of a 0-th column of an i-th (i=0, 1, ..., K_{ldpc}/M-1) column group is set to be D_i and positions (or index) of each row in which 1 is positioned in the 0-th column of the i-th column group is set to be $R_{i,0}^{(0)}$, $R_{i,0}^{(1)}$, ..., $R_{i,0}^{(D_i-1)}$, an index $R_{i,j}^{(k)}$ of a row in which a k-th 1 is positioned in a j-th column in the i-th column group is determined based on following Equation 9.

[140] $$R_{i,j}^{(k)} = R_{i,(j-1)}^{(k)} + Q_{ldpc} \bmod (N_{inner} - K_{ldpc}) \quad \text{.... (9)}$$

[141] In above Equation 9, k = 0, 1, 2, ..., D_i-1; i = 0,1 , ..., K_{ldpc}/M-1; j = 1, 2, ..., M-1.

[142] Meanwhile, above Equation 9 may be represented like following Equation 10.

[143] $$R_{i,j}^{(k)} = (R_{i,0}^{(k)} + (j \bmod M) \times Q_{ldpc}) \bmod (N_{inner} - K_{ldpc}) \quad \text{.... (10)}$$

[144] In above Equation 10, k = 0, 1, 2, ..., D_i-1; i = 0,1 , ..., K_{ldpc}/M-1; j = 1, 2, ..., M-1. In above Equation 10, since j = 1, 2, ..., M-1, (j mod M) may be considered as j.

[145] In these Equations, $R_{i,j}^{(k)}$ represents the index of the row in which the k-th 1 is positioned in the j-th column in the i-th column group, the N_{inner} represents the length of the LDPC codeword, the K_{ldpc} represents the length of the information bits, the D_i represents the degree of the columns belonging to the i-th column group, the M represents the number of columns belonging to one column group, and the Q_{ldpc} represents the size at which each column is cyclic-shifted.

[146] As a result, referring to the above Equations, if a $R_{i,0}^{(k)}$ value is known, the index $R_{i,j}^{(k)}$ of the row in which the k-th 1 is positioned in the j-th column of the i-th column group may be known. Therefore, when the index value of the row in which the k-th 1 is positioned in the 0-th columns of each column group is stored, the positions of the column and the row in which the 1 is positioned in the parity check matrix 40 (that is, information sub-matrix 41 of the parity check matrix 40) having the structure of FIG. 4 may be checked.

[147] According to the foregoing rules, all the degrees of the columns belonging to the i-th column group are D_i. Therefore, according to the foregoing rules, the LDPC code in which the information on the parity check matrix is stored may be briefly represented as follows.

[148] For example, when the N_{inner} is 30, the K_{ldpc} is 15, and the Q_{ldpc} is 3, positional information of the row in which 1 is positioned in the 0-th columns of three column groups may be represented by sequences as following Equation 11, which may be named 'weight-1 position sequence'.

[149] $$R_{1,0}^{(1)}=1, R_{1,0}^{(2)}=2, R_{1,0}^{(3)}=8, R_{1,0}^{(4)}=10$$

[150] $R_{2,0}^{(1)}=0, R_{2,0}^{(2)}=9, R_{2,0}^{(3)}=13$

[151] $R_{3,0}^{(1)}=0, R_{3,0}^{(2)}=14 \cdots (11)$

[152] In above Equation 11, $R_{i,j}^{(k)}$ represents the indexes of the row in which the k-th 1 is positioned in the j-th column of the i-th column group.

[153] The weight-1 position sequences as above Equation 11 representing the index of the row in which 1 is positioned in the 0-th columns of each column group may be more briefly represented as following Table 3.

[154] [Table 3]

[155]

1 2 8 10
0 9 13
0 14

[156] Above Table 3 represents positions of elements having a value 1 in the parity check matrix and the i-th weight-1 position sequence is represented by the indexes of the row in which 1 is positioned in the 0-th column belonging to the i-th column group.

[157] The information sub-matrix 41 of the parity check matrix according to the exemplary embodiment described above may be defined based on following Table 4.

[158] Here, following Table 4 represents the indexes of the row in which 1 is positioned in a 0-th column of the i-th column group in the information sub-matrix 41. That is, the information sub-matrix 41 is formed of a plurality of column groups each including M columns and the positions of 1s in the 0-th columns of each of the plurality of column groups may be defined as following Table 4.

[159] For example, when the length N_{inner} of the LDPC codeword is 16200, the code rate is 6 / 15, and the M is 360, the indexes of the row in which 1 is positioned in the 0-th column of the i-th column group in the information sub-matrix 41 are as following Table 4.

[160] [Table 4]

[161]

27 430 519 828 1897 1943 2513 2600 2640 3310 3415 4266 5044 5100 5328 5483 5928 6204 6392 6416 6602 7019 7415 7623 8112 8485 8724 8994 9445 9667
27 174 188 631 1172 1427 1779 2217 2270 2601 2813 3196 3582 3895 3908 3948 4463 4955 5120 5809 5988 6478 6604 7096 7673 7735 7795 8925 9613 9670
27 370 617 852 910 1030 1326 1521 1606 2118 2248 2909 3214 3413 3623 3742 3752 4317 4694 5300 5687 6039 6100 6232 6491 6621 6860 7304 8542 8634
990 1753 7635 8540
933 1415 5666 8745
27 6567 8707 9216
2341 8692 9580 9615
260 1092 5839 6080
352 3750 4847 7726
4610 6580 9506 9597
2512 2974 4814 9348
1461 4021 5060 7009
1796 2883 5553 8306
1249 5422 7057
3965 6968 9422
1498 2931 5092
27 1090 6215
26 4232 6354

[162] According to another exemplary embodiment, a parity check matrix in which an order of indexes in each sequence corresponding to each column group in above Table 4 is changed is considered as a same parity check matrix for an LDPC code as the above described parity check matrix is another example of the inventive concept.

[163] According to still another exemplary embodiment, a parity check matrix in which an array order of the sequences of the column groups in above Table 4 is changed is also considered as a same parity check matrix as the above described parity check matrix in that they have a same algebraic characteristics such as cyclic characteristics and degree distributions on a graph of a code.

[164] According to yet another exemplary embodiment, a parity check matrix in which a multiple of Q_{ldpc} is added to all indexes of a sequence corresponding to column group in above Table 4 is also considered as a same parity check matrix as the above described parity check matrix in that they have same cyclic characteristics and degree distributions on the graph of the code. Here, it is to be noted that when a value obtained by adding an integer multiple of Q_{ldpc} to a given sequence is greater than or equal to N_{inner}-K_{ldpc}, the value needs to be changed to a value obtained by performing a modulo operation on N_{inner}-K_{ldpc} and then applied.

[165] If the position of the row in which 1 is positioned in the 0-th column of the i-th column group in the information sub-matrix 41 as shown in above Table 4 is defined, it may be cyclic-shifted by Q_{ldpc}, and thus, the position of the row in which 1 is positioned in other columns of each column group may be defined.

[166] For example, as shown in above Table 4, since the sequence corresponding to the 0-th column of the 0-th column group of the information sub-matrix 41 is "27 430 519 828 1897 1943 2513 2600 2640 3310 3415 4266 5044 5100 5328 5483 5928 6204 6392 6416 6602 7019 7415 7623 8112 8485 8724 8994 9445 9667", in the 0-th column of the 0-th column group in the information sub-matrix 41, 1 is positioned in a 27-th row, a 430-th row, a 519-th-row, ...

[167] In this case, since $Q_{ldpc} = (N_{inner}-K_{ldpc})/M = (16200-6480)/360 = 27$, the indexes of the row in which 1 is positioned in the first column of the 0-th column group may be 54(=27+27), 457(=430+27), 546(=519+27), ..., 81(=54+27), 484(=457+27), 573(=546+27), ...

[168] By the above scheme, the indexes of the row in which 1 is positioned in all the rows of each column group may be defined.

[169] Hereinafter, the method for performing LDPC encoding based on the parity check matrix 40 as illustrated in FIG. 4 will be described.

[170] First, information bits to be encoded are set to be i_0, i_1, ..., $i_{K_{ldpc}-1}$, and code bits output from the LDPC encoding are set to be c_0, c_1, ..., $c_{N_{inner}-1}$.

[171] Further, since the LDPC code is systematic, for k ($0 \leq k < K_{ldpc}-1$), c_k is set to be i_k. Meanwhile, the remaining code bits are set to be $p_k := c_{k+k_{ldpc}}$.

[172] Hereinafter, a method for calculating parity bits p_k will be described.

[173] Hereinafter, q(i, j, 0) represents a j-th entry of an i-th row in an index list as above Table 4, and q(i, j, l) is set to be $q(i, j, l) = q(i, j, 0)+Q_{ldpc} \times l \pmod{N_{inner}-K_{ldpc}}$ for $0 < i < 360$. Meanwhile, all the accumulations may be realized by additions in a Galois field (GF) (2). Further, in above Table 4, since the length of the LDPC codeword is 16200 and the code rate is 6/15, the Q_{ldpc} is 27.

[174] Meanwhile, when the q(i,j,0) and the q(i,j,l) are defined as above, a process of calculating the parity bit is as follows.

[175] Step 1) The parity bits are initialized to '0'. That is, $p_k = 0$ for $0 \leq k < N_{inner}-K_{ldpc}$.

[176] Step 2) For all k values of $0 \leq k < K_{ldpc}$, i and l are set to be $i := \lfloor k/360 \rfloor$ and $l := k \pmod{360}$. Here, $\lfloor x \rfloor$ is a maximum integer which is not greater than x.

[177] Next, for all i, i_k is accumulated in $p_{q(i,j,l)}$. That is, $p_{q(i,0,l)} = p_{q(i,0,l)}+i_k, p_{q(i,1,l)} = p_{q(i,1,l)}+i_k, p_{q(i,2,l)} = p_{q(i,2,l)}+i_k, \ldots, p_{q(i,w(i)-1,l)} = p_{q(i,w(i)-1,l)}+i_k$ are calculated.

[178] Here, w(i) represents the number of the values (elements) of an i-th row in the index list as above Table 4 and represents the number of 1s in a column corresponding to i_k in the parity check matrix. Further, in above Table 4, q(i, j, 0) which is a j-th entry of an i-th row is an index of a parity bit and represents the position of the row in which 1 is positioned in a column corresponding to i_k in the parity check matrix.

[179] In detail, in above Table 4, q(i,j,0) which is the j-th entry of the i-th row represents the position of the row in which 1 is positioned in the first (that is, 0-th) column of the i-th column group in the parity check matrix of the LDPC code.

[180] The q(i, j, 0) may also be considered as the index of the parity bit to be generated by LDPC encoding according to a method for allowing a real apparatus to implement a scheme for accumulating i_k in $p_{q(i, j, l)}$ for all i, and may also be considered as an index in another form when another encoding method is implemented. However, this is only one example, and therefore, it is apparent to obtain an equivalent result to an LDPC encoding result which may be obtained from the parity check matrix of the LDPC code which may basically be generated based on the q(i,j,0) values of above Table 4 whatever the encoding scheme is applied.

[181] Step 3) A parity bit p_k is calculated by calculating $p_k = p_k+p_{k-1}$ for all k satisfying $0 < k < N_{inner}-K_{ldpc}$.

[182] Accordingly, all code bits $c_0, c_1, \ldots, c_{N_{inner}-1}$ may be obtained.

[183] As a result, parity bits may be calculated by the above scheme. However, this is only one example and therefore the scheme for calculating the parity bits based on the parity check matrix as illustrated in FIG. 4 may be variously defined.

[184] As such, the LDPC encoder 130 may perform LDPC encoding based on above Table 4 to generate an LDPC codeword.

[185] In detail, the LDPC encoder 130 may perform the LDPC encoding on 6480 input bits, that is, the LDPC information bits at the code rate of 6 / 15 based on above Table 4 to generate 9720 LDPC parity bits and output the LDPC parity bits and the LDPC codeword formed of the LDPC parity bits. In this case, the LDPC codeword may be formed of 16200 bits.

[186] As described above, the LDPC encoder 130 may encode LDPC information bits at various code rates to generate an LDPC codeword.

[187] Here, when the zero padder 120 pads zero bits based on above Table 1, the LDPC encoder 130 may perform LDPC encoding on LDPC information bits in which zero bits are padded at a code rate of 6/15. In this case, the LDPC information bits may be formed of 6480 bits and the LDPC parity bits generated by the LDPC encoding may be formed of 9720 bits.

[188] The transmitter 100 may transmit the LDPC codeword to the receiver 200.

[189] In detail, the transmitter 100 may map the shortened LDPC codeword bits to constellation symbols by QPSK, map the symbols to a frame for transmission to the receiver 200.

[190] Since the information bits are signaling including signaling information for data, the transmitter 100 may map the data to a frame along with the signaling for processing the data and transmit the mapped data to the receiver 200.

[191] In detail, the transmitter 100 may process the data in a specific scheme to generate the constellation symbols and map the generated constellation symbols to data symbols of each frame. Further, the transmitter 100 may map the signaling for data mapped to each data to a preamble of the frame. For example, the transmitter 100 may map the signaling including the signaling information for the data mapped to the i-th frame to the i-th frame.

[192] As a result, the receiver 200 may use the signaling acquired from the frame to acquire and process the data from the corresponding frame.

[193] Hereinafter, a process of inducing a shortening pattern for zero padding will be described as an example.

[194] In detail, when the LDPC encoder 130 encodes 6480 LDPC information bits at the code rate of 6/15 to generate 9720 LDPC parity bits and the LDPC codeword generated by the LDPC encoding is modulated by the QPSK and then is transmitted to the receiver 200, a process of inducing the shortening pattern for the zero padding is as

follows.

[195] The parity check matrix (for example, FIG. 4) of an LDPC code having the code rate of 6/15 may be converted into the parity check matrix having a quasi cyclic structure configured of blocks having a size of 360×360 as illustrated in FIG. 5 by performing a column permutation process and an appropriate row permutation process corresponding to a parity interleaving process. Here, the column permutation process and the row permutation process do not change algebraic characteristics of the LDPC code, and therefore, have been widely used to theoretically analyze the LDPC code. Further, the parity check matrix having the quasi cyclic structure has been already known, and therefore, the detailed description thereof will be omitted.

[196] Obtaining a shortening pattern for zero padding may be considered as a problem of defining a degree of importance between 18 column blocks of an information bit portion present in an LDPC code having the code rate of 6/15. That is, shortening or zero padding specific information bits is the same as shortening or removing columns corresponding to the information bits in the parity check matrix. Therefore, when n bits among an information word need to be shortened based on the length of input signaling, there is a need to determine which n columns are to be removed from the parity check matrix in terms of bit error rate (BER) or frame error rate (FER) performance.

[197] According to an exemplary embodiment, a shorting pattern for zero padding is induced by using characteristics of the LDPC code, that is, columns within one column block (that is, a set of continued 360 columns) having the same algebraic characteristics, and the total number of information bit groups being only 18.

[198] In a first step, the following nine situations are considered in a parity check matrix to measure the real BER and FER performance.

[199] When information is carried on only bits belonging to a 0-th bit group and the remaining bits are zero-padded.

[200] When information is carried on only bits belonging to a 1-th bit group and the remaining bits are zero-padded.

[201] When information is carried on only bits belonging to a 2-th bit group and the remaining bits are zero-padded.

[202] When information is carried on only bits belonging to a 3-th bit group and the remaining bits are zero-padded.

[203] When information is carried on only bits belonging to a he 4-th bit group and the remaining bits are zero-padded.

[204] When information is carried on only bits belonging to a 5-th bit group and the remaining bits are zero-padded.

[205] When information is carried on only bits belonging to a 6-th bit group and the

remaining bits are zero-padded.

[206] When information is carried on only bits belonging to a 7-th bit group and the remaining bits are zero-padded.

[207] When information is carried on only bits belonging to an 8-th bit group and the remaining bits are zero-padded.

[208] When information is carried on only bits belonging to a 9-th bit group and the remaining bits are zero-padded.

[209] When information is carried on only bits belonging to a 10-th bit group and the remaining bits are zero-padded.

[210] When information is carried on only bits belonging to a 11-th bit group and the remaining bits are zero-padded.

[211] When information is carried on only bits belonging to a 12-th bit group and the remaining bits are zero-padded.

[212] When information is carried on only bits belonging to a 13-th bit group and the remaining bits are zero-padded.

[213] When information is carried on only bits belonging to a 14-th bit group and the remaining bits are zero-padded.

[214] When information is carried on only bits belonging to a 15-th bit group and the remaining bits are zero-padded.

[215] When information is carried on only bits belonging to a 16-th bit group and the remaining bits are zero-padded.

[216] When information is carried on only bits belonging to a 17-th bit group and the remaining bits are zero-padded.

[217] The BER and FER performance obtained under the nine situations are observed. First, bit groups of which performance difference from the best performance bit group is less than or equal to a predetermined value (for example, 0.1 dB) are set as candidate bit groups to be finally shortened. To select a bit group to be finally shortened among the candidate bit groups, cyclic characteristics such as an approximate cycle extrinsic message (ACE) degree may be additionally considered. The ACE value of a cycle having a length of 2n is defined as a sum of values obtained by subtracting 2 from a degree of n variable nodes connected to the cycle. Since a cycle having a small ACE value and a short length adversely affect performance of an LDPC code, a bit group, among the candidate bit groups, which has the smallest cycle number among the number of cycles of which the length is less than or equal to 8 and of which the ACE value is less than or equal to 3 in a matrix resulting from shortening column blocks corresponding to this bit group, may be selected. If there are too many number of such bit groups according to the cyclic characteristics based on the ACE value, a theoretical prediction value for a minimum signal-to-noise (SNR) which enables error-free com-

munication for ensembles of the LDPC code having a same distribution of 1 after column deletion, row merging and row deletion for each of these bit groups is derived by a density evolution analysis, and FER performance is verified by a computation ex-periment by appropriately adjusting the number of the bit groups based on the minimum SNR values theoretically predicted. As a result, the 5-th bit group may be selected.

[218] In a second step for obtaining the shortening patter, the real BER and FER per-formance is measured considering the following seventeen situations.

[219] When information is carried on only bits belonging to the 0-th bit group and the 3-th bit group and the remaining bits are zero-padded.

[220] When information is carried on only bits belonging to the 1-th bit group and the 3-th bit group and the remaining bits are zero-padded.

[221] When information is carried on only bits belonging to the 2-th bit group and the 3-th bit group and the remaining bits are zero-padded.

[222] When information is carried on only bits belonging to the 3-th bit group and the 4-th bit group and the remaining bits are zero-padded.

[223] When information is carried on only bits belonging to the 3-th bit group and the 5-th bit group and the remaining bits are zero-padded.

[224] When information is carried on only bits belonging to the 3-th bit group and the 6-th bit group and the remaining bits are zero-padded.

[225] When information is carried on only bits belonging to the 3-th bit group and the 7-th bit group and the remaining bits are zero-padded.

[226] When information is carried on only bits belonging to the 3-th bit group and the 8-th bit group and the remaining bits are zero-padded.

[227] When information is carried on only bits belonging to the 3-th bit group and the 9-th bit group and the remaining bits are zero-padded.

[228] When information is carried on only bits belonging to the 3-th bit group and the 10-th bit group and the remaining bits are zero-padded.

[229] When information is carried on only bits belonging to the 3-th bit group and the 11-th bit group and the remaining bits are zero-padded.

[230] When information is carried on only bits belonging to the 3-th bit group and the 12-th bit group and the remaining bits are zero-padded.

[231] When information is carried on only bits belonging to the 3-th bit group and the 13-th bit group and the remaining bits are zero-padded.

[232] When information is carried on only bits belonging to the 3-th bit group and the 14-th bit group and the remaining bits are zero-padded.

[233] When information is carried on only bits belonging to the 3-th bit group and the 15-th bit group and the remaining bits are zero-padded.

[234] When information is carried on only bits belonging to the 3-th bit group and the 16-th bit group and the remaining bits are zero-padded.

[235] When information is carried on only bits belonging to the 3-th bit group and the 17-th bit group and the remaining bits are zero-padded.

[236] The above eight situations are for situations in which selection of a bit group to carry additional information is required in addition to the 3-th bit group which is already selected in the first step. After the BER and FER performances obtained under these situations are observed, a bit group having the best performance is selected as a candidate group to be shortened just before shortening the 3-th bit group. Next, a column group corresponding to the 3-th bit group in the parity-check matrix is shortened and a bit group among the candidate bit groups to be shortened just before the 3-th bit group is shortened, and then, in the matrix left after the foregoing shortening, the number of cycle having the length less than or equal to 8 and the ACE value less than or equal to 3 may be checked to select a bit group of which the number of cycles is smallest. For example, the 4-th bit group may be selected.

[237] As a result, the above process is repeated as far as 18 bit groups of the overall information As a result, the above process is repeated until 18 bit groups of LDPC information bits may be selected to obtain the shortening pattern for zero padding as shown in above Table 1. As a result, when zero bits are padded based on the shortening pattern as shown in above Table 1, excellent BER and FER performances may be obtained.

[238] Meanwhile, according to an exemplary embodiment, the foregoing information bits may be implemented by L1-detail signaling. Therefore, the transmitter 100 may perform a shortening procedure for the L1-detail signaling by using the foregoing method for transmission to the receiver 200.

[239] Here, the L1-detail signaling may be signaling defined in an Advanced Television System Committee (ATSC) 3.0 standard.

[240] In detail, The L1-detail signaling may be processed according to seven (7) different modes. The transmitter 100 according to the exemplary embodiment may generate additional parity bits according to the foregoing method when an L1-detail mode 3 among the seven modes processes the L1-detail signaling.

[241] The ATSC 3.0 standard defines L1-basic signaling besides the L1-detail signaling. The transmitter 100 may process the L1-basic signaling and the L1-detail signaling by using a specific scheme, and transmit the processed L1-basic signaling and the L1-detail signaling to the receiver 200. In this case, the L1-basic signaling may also be processed according to seven different modes.

[242] A method for processing the L1-basic signaling and the L1-detail signaling will be described below.

[243] The transmitter 100 may map the L1-basic signaling and the L1-detail signaling to a preamble of a frame and map data to data symbols of the frame for transmission to the receiver 200.

[244] Referring to FIG. 6, the frame may be configured of three parts, that is, a bootstrap part, a preamble part, and a data part.

[245] The bootstrap part is used for initial synchronization and provides a basic parameter required for the receiver 200 to decode the L1 signaling. Further, the bootstrap part may include information about a mode of processing the L1-basic signaling at the transmitter 100, that is, information about a mode the transmitter 100 uses to process the L1-basic signaling.

[246] The preamble part includes the L1 signaling, and may be configured of two parts, that is, the L1-basic signaling and the L1-detail signaling.

[247] Here, the L1-basic signaling may include information about the L1-detail signaling, and the L1-detail signaling may include information about data. Here, the data is broadcasting data for providing broadcasting services and may be transmitted through at least one physical layer pipes (PLPs).

[248] In detail, the L1-basic signaling includes information required for the receiver 200 to process the L1-detail signaling. This information includes, for example, information about a mode of processing the L1-detail signaling at the transmitter 100, that is, information about a mode the transmitter 100 uses to process the L1-detail signaling, information about a length of the L1-detail signaling, information about an additional parity mode, that is, information about a K value used for the transmitter 100 to generate additional parity bits using an L1B_L1_Detail_additional_parity_mode (here, when the L1B_L1_Detail_additional_parity_mode is set as '00', K = 0 and the additional parity bits are not used), and information about a length of total cells. Further, the L1-basic signaling may include basic signaling information about a system including the transmitter 100 such as a fast Fourier transform (FFT) size, a guard interval, and a pilot pattern.

[249] Further, the L1-detail signaling includes information required for the receiver 200 to decode the PLPs, for example, start positions of cells mapped to data symbols for each PLP, PLP identifier (ID), a size of the PLP, a modulation scheme, a code rate, etc..

[250] Therefore, the receiver 200 may acquire frame synchronization, acquire the L1-basic signaling and the L1-detail signaling from the preamble, and receive service data required by a user from data symbols using the L1-detail signaling.

[251] The method for processing the L1-basic signaling and the L1-detail signaling will be described below in more detail with reference to the accompanying drawings.

[252] FIGs. 7 and 8 are block diagrams for describing detailed configurations of the transmitter 100, according to exemplary embodiments.

[253] In detail, as illustrated in FIG. 7, to process the L1-basic signaling, the transmitter 100 may include a scrambler 211, a BCH encoder 212, a zero padder 213, an LDPC encoder 214, a parity permutator 215, a repeater 216, a puncturer 217, a zero remover 219, a bit demultiplexer 219, and a constellation mapper 221.

[254] Further, as illustrated in FIG. 8, to process the L1-detail signaling, the transmitter 100 may include a segmenter 311, a scrambler 312, a BCH encoder 313, a zero padder 314, an LDPC encoder 315, a parity permutator 316, a repeater 317, a puncturer 318, an additional parity generator 319, a zero remover 321, bit demultiplexers 322 and 323, and constellation mappers 324 and 325.

[255] Here, the components illustrated in FIGs. 7 and 8 are components for performing encoding and modulation on the L1-basic signaling and the L1-detail signaling, which is only one example. According to another exemplary embodiments, some of the components illustrated in FIGs. 7 and 8 may be omitted or changed, and other components may also be added. Further, positions of some of the components may be changed. For example, the positions of the repeaters 216 and 317 may be disposed after the puncturers 217 and 318, respectively.

[256] The LDPC encoder 315, the repeater 317, the puncturer 318, and the additional parity generator 319 illustrated in FIG. 10 may perform the operations performed by the LDPC encoder 110, the repeater 120, the puncturer 130, and the additional parity generator 140 illustrated in FIG. 1, respectively.

[257] In describing FIGs. 9 and 10, for convenience, components for performing common functions will be described together.

[258] The L1-basic signaling and the L1-detail signaling may be protected by concatenation of a BCH outer code and an LDPC inner code. However, this is only one example. Therefore, as outer encoding performed before inner encoding in the concatenated coding, another encoding such as CRC encoding in addition to the BCH encoding may be used. Further, the L1-basic signaling and the L1-detail signaling may be protected only by the LDPC inner code without the outer code.

[259] First, the L1-basic signaling and the L1-detail signaling may be scrambled. Further, the L1-basic signaling and the L1-detail signaling are BCH encoded, and thus, BCH parity check bits of the L1-basic signaling and the L1-detail signaling generated from the BCH encoding may be added to the L1-basic signaling and the L1-detail signaling, respectively. Further, the concatenated signaling and the BCH parity check bits may be additionally protected by a shortened and punctured 16K LDPC code.

[260] To provide various robustness levels appropriate for a wide signal to noise ratio (SNR) range, a protection level of the L1-basic signaling and the L1-detail signaling may be divided into seven (7) modes. That is, the protection level of the L1-basic signaling and the L1-detail signaling may be divided into the seven modes based on an

LDPC code, a modulation order, shortening/puncturing parameters (that is, a ratio of the number of bits to be punctured to the number of bits to be shortened), and the number of bits to be basically punctured (that is, the number of bits to be basically punctured when the number of bits to be shortened is 0). In each mode, at least one different combination of the LDPC code, the modulation order, the constellation, and the shortening/puncturing pattern may be used.

[261] A mode for the transmitter 100 to processes the signaling may be set in advance depending on a system. Therefore, the transmitter 100 may determine parameters (for example, modulation and code rate (ModCod) for each mode, parameter for the BCH encoding, parameter for the zero padding, shortening pattern, code rate/code length of the LDPC code, group-wise interleaving pattern, parameter for repetition, parameter for puncturing, and modulation scheme, etc.) for processing the signaling depending on the set mode, and may process the signaling based on the determined parameters and transmit the processed signaling to the receiver 200. For this purpose, the transmitter 100 may pre-store the parameters for processing the signaling depending on the mode.

[262] Modulation and code rate configurations (ModCod configurations) for the seven modes for processing the L1-basic signaling and the seven modes for processing the L1-detail signaling are shown in following Table 5. The transmitter 100 may encode and modulate the signaling based on the ModCod configurations defined in following Table 5 according to a corresponding mode. That is, the transmitter 100 may determine an encoding and modulation scheme for the signaling in each mode based on following Table 5, and may encode and modulate the signaling according to the determined scheme. In this case, even when modulating the L1 signaling by the same modulation scheme, the transmitter 100 may also use different constellations.

[263] [Table 5]

[264]

Signaling FEC Type		K_{sig}	Code Length	Code Rate	Constellation
L1-Basic	Mode 1	200	16200	3/15 (Type A)	QPSK
	Mode 2				QPSK
	Mode 3				QPSK
	Mode 4				NUC_16-QAM
	Mode 5				NUC_64-QAM
	Mode 6				NUC_256-QAM
	Mode 7				NUC_256-QAM
L1-Detail	Mode 1	400 ~ 2352			QPSK
	Mode 2	400 ~ 3072			QPSK
	Mode 3	400 ~ 6312		6/15 (Type B)	QPSK
	Mode 4				NUC_16-QAM
	Mode 5				NUC_64-QAM
	Mode 6				NUC_256-QAM
	Mode 7				NUC_256-QAM

[265] In above Table 5, K_{sig} represents the number of information bits for a coded block. That is, since the L1 signaling bits having a length of K_{sig} are encoded to generate the coded block, a length of the L1 signaling in one coded block becomes K_{sig}. Therefore,

the L1 signaling bits having the size of K_{sig} may be considered as corresponding to one LDPC coded block.

[266] Referring to above Table 5, the K_{sig} value for the L1-basic signaling is fixed to 200. However, since the amount of L1-detail signaling bits varies, the K_{sig} value for the L1-detail signaling varies.

[267] In detail, in a case of the L1-detail signaling, the number of L1-detail signaling bits varies, and thus, when the number of L1-detail signaling bits is greater than a preset value, the L1-detail signaling may be segmented to have a length which is equal to or less than the preset value.

[268] In this case, each size of the segmented L1-detail signaling blocks (that is, segment of the L1-detail signaling) may have the K_{sig} value defined in above Table 5. Further, each of the segmented L1-detail signaling blocks having the size of K_{sig} may correspond to one LDPC coded block.

[269] However, when the number of L1-detail signaling bits is equal to or less than the preset value, the L1-detail signaling is not segmented. In this case, the size of the L1-detail signaling may have the K_{sig} value defined in above Table 5. Further, the L1-detail signaling having the size of K_{sig} may correspond to one LDPC coded block.

[270] Hereinafter, a method for segmenting L1-detail signaling will be described in detail.

[271] The segmenter 311 segments the L1-detail signaling. In detail, since the length of the L1-detail signaling varies, when the length of the L1-detail signaling is greater than the preset value, the segmenter 311 may segment the L1-detail signaling to have the number of bits which are equal to or less than the preset value and output each of the segmented L1-detail signalings to the scrambler 312.

[272] However, when the length of the L1-detail signaling is equal to or less than the preset value, the segmenter 311 does not perform a separate segmentation operation.

[273] A method for segmenting, by the segmenter 311, the L1-detail signaling is as follows.

[274] The amount of L1-detail signaling bits varies and mainly depends on the number of PLPs. Therefore, to transmit all bits of the L1-detail signaling, at least one forward error correction (FEC) frame is required. Here, an FEC frame may represent a form in which the L1-detail signaling is encoded, and thus, parity bits according to the encoding are added to the L1-detail signaling.

[275] In detail, when the L1-detail signaling is not segmented, the L1-detail signaling is BCH-encoded and LDPC encoded to generate one FEC frame, and therefore, one FEC frame is required for the L1-detail signaling transmission. However, when the L1-detail signaling is segmented into at least two, these segmented L1-detail signalings each are BCH- encoded and LDPC-encoded to generate at least two FEC frames, and therefore, at least two FEC frames are required for the L1-detail signaling

transmission.

[276] Therefore, the segmenter 311 may calculate the number $N_{L1D_FECFRAME}$ of FEC frames for the L1-detail signaling based on following Equation 12. That is, the number $N_{L1D_FECFRAME}$ of FEC frames for the L1-detail signaling may be determined based on following Equation 12.

[277]
$$N_{L1D_FECFRAME} = \left\lceil \frac{K_{L1D_ex_pad}}{K_{seg}} \right\rceil \quad \text{.... (12)}$$

[278] In above Equation 12, $\lceil x \rceil$ represents a minimum integer which is equal to or greater than x.

[279] Further, in above Equation 12, $K_{L1D_ex_pad}$ represents the length of the L1-detail signaling except L1 padding bits as illustrated in FIG. 9, and may be determined by a value of an L1B_L1_Detail_size_bits field included in the L1-basic signaling.

[280] Further, K_{seg} represents a threshold number for segmentation defined based on the number K_{ldpc} of information bits input to the LDPC encoder 315, that is, the LDPC information bits. Further, K_{seg} may be defined based on the number of BCH parity check bits of BCH encoding and a multiple value of 360.

[281] K_{seg} is determined such that, after the L1-detail signaling is segmented, the number K_{sig} of information bits in the coded block is set to be equal to or less than K_{ldpc}-M_{outer}. In detail, when the L1-detail signaling is segmented based on K_{seg}, since the length of segmented L1-detail signaling does not exceed K_{seg}, the length of the segmented L1-detail signaling is set to be equal to or less than K_{ldpc}-M_{outer} when K_{seg} is set like in Table 6 as following.

[282] Here, M_{outer} and K_{ldpc} are as following Tables 7 and 8. For sufficient robustness, the K_{seg} value for the L1-detail signaling mode 1 may be set to be K_{ldpc}-M_{outer}-720.

[283] K_{seg} for each mode of the L1-detail signaling may be defined as following Table 6. In this case, the segmenter 311 may determine K_{seg} according to a corresponding mode as shown in following Table 6.

[284] [Table 6]

[285]

L1-Detail	K_{seg}
Mode 1	2352
Mode 2	3072
Mode 3	6312
Mode 4	
Mode 5	
Mode 6	
Mode 7	

[286] As illustrated in FIG. 9, an entire L1-detail signaling may be formed of L1-detail signaling and L1 padding bits.

[287] In this case, the segmenter 311 may calculate a length of an L1_PADDING field for the L1-detail signaling, that is, the number $_{L1D_PAD}$ of the L1 padding bits based on following Equation 13. However, calculating K_{L1D_PAD} based on following Equation 13 is only one example. That is, the segmenter 311 may calculate the length of the L1_PADDING field for the L1-detail signaling, that is, the number K_{L1D_PAD} of the L1 padding bits based on $K_{L1D_ex_pad}$ and $N_{L1D_FECFRAME}$ values. As one example, the K_{L1D_PAD} value may be obtained based on following Equation 13. That is, following Equation 18 is only one example of a method for obtaining a K_{L1D_PAD} value, and thus, another method based on the $K_{L1D_ex_pad}$ and $N_{L1D_FECFRAME}$ values may be applied to obtain an equivalent result.

[288] ....

$$K_{L1D_PAD} = \left\lceil \frac{K_{L1D_ex_pad}}{(N_{L1D_FECFRAME} \times 8)} \right\rceil \times 8 \times N_{L1D_FECFRAME} - K_{L1D_ex_pad}$$

(13)

[289] Further, the segmenter 311 may fill the L1_PADDING field with K_{L1D_PAD} zero bits (that is, bits having a 0 value). Therefore, as illustrated in FIG. 11, the K_{L1D_PAD} zero bits may be filled in the L1_PADDING field.

[290] As such, by calculating the length of the L1_PADDING field and padding zero bits of the calculated length to the L1_PADDING field, the L1-detail signaling may be segmented into the plurality of blocks formed of the same number of bits when the L1-detail signaling is segmented.

[291] Next, the segmenter 311 may calculate a final length K_{L1D} of the entire L1-detail signaling including the zero padding bits based on following Equation 14.

[292] $K_{L1D}=K_{L1D_ex_pad}+K_{L1D_PAD}$ .... (14)

[293] Further, the segmenter 311 may calculate the number K_{sig} of information bits in each of the $N_{L1D_FECFRAME}$ blocks based on following Equation 15.

[294] $K_{sig}=K_{L1D}/N_{L1D_FECFRAME}$ .... (15)

[295] Next, the segmenter 311 may segment the L1-detail signaling by K_{sig} number of bits.

[296] In detail, as illustrated in FIG. 9, when $N_{L1D_FECFRAME}$ is greater than 1, the segmenter 311 may segment the L1-detail signaling by the number of K_{sig} bits to segment the L1-detail signaling into the $N_{L1D_FECFRAME}$ blocks.

[297] Therefore, the L1-detail signaling may be segmented into $N_{L1D_FECFRAME}$ blocks, and the number of L1-detail signaling bits in each of the $N_{L1D_FECFRAME}$ blocks may be K_{sig}. Further, each segmented L1-detail signaling is encoded. As an encoded result, a coded block, that is, an FEC frame is formed, such that the number of L1-detail signaling bits in each of the $N_{L1D_FECFRAME}$ coded blocks may be K_{sig}.

[298] However, when the L1-detail signaling is not segmented, $K_{sig}=K_{L1D_ex_pad}$.

[299] The segmented L1-detail signaling blocks may be encoded by a following procedure.

[300] In detail, all bits of each of the L1-detail signaling blocks having the size K_{sig} may be scrambled. Next, each of the scrambled L1-detail signaling blocks may be encoded by concatenation of the BCH outer code and the LDPC inner code.

[301] In detail, each of the L1-detail signaling blocks is BCH-encoded, and thus M_{outer} (=168) BCH parity check bits may be added to the K_{sig} L1-detail signaling bits of each block, and then, the concatenation of the L1-detail signaling bits and the BCH parity check bits of each block may be encoded by a shortened and punctured 16K LDPC code. The details of the BCH code and the LDPC code will be described below. However, the exemplary embodiments describe only a case in which M_{outer}=168, but it is apparent that M_{outer} may be changed into an appropriate value depending on the requirements of a system.

[302] The scramblers 211 and 312 scramble the L1-basic signaling and the L1-detail signaling, respectively. In detail, the scramblers 211 and 312 may randomize the L1-basic signaling and the L1-detail signaling, and output the randomized L1-basic signaling and L1-detail signaling to the BCH encoders 212 and 313, respectively.

[303] In this case, the scramblers 211 and 312 may scramble the information bits by a unit of K_{sig}.

[304] That is, since the number of L1-basic signaling bits transmitted to the receiver 200 through each frame is 200, the scrambler 211 may scramble the L1-basic signaling bits by K_{sig} (=200).

[305] Since the number of L1-basic signaling bits transmitted to the receiver 200 through each frame varies, in some cases, the L1-detail signaling may be segmented by the segmenter 311. Further, the segmenter 311 may output the L1-detail signaling formed of K_{sig} bits or the segmented L1-detail signaling blocks to the scrambler 312. As a result, the scrambler 312 may scramble the L1-detail signaling bits by every K_{sig} which are output from the segmenter 311.

[306] The BCH encoders 212 and 313 perform the BCH encoding on the L1-basic signaling and the L1-detail signaling to generate the BCH parity check bits.

[307] In detail, the BCH encoders 212 and 313 may perform the BCH encoding on the L1-basic signaling and the L1-detail signaling output from the scramblers 211 and 313, respectively, to generate the BCH parity check bits, and output the BCH-encoded bits in which the BCH parity check bits are added to each of the L1-basic signaling and the L1-detail signaling to the zero padders 213 and 314, respectively.

[308] For example, the BCH encoders 212 and 313 may perform the BCH encoding on the input K_{sig} bits to generate the M_{outer} (that is, $K_{sig}=K_{payload}$) BCH parity check bits and output the BCH-encoded bits formed of N_{outer} (= $K_{sig}+M_{outer}$) bits to the zero padders 213 and 314, respectively.

[309] The parameters for the BCH encoding may be defined as following Table 7.

[310] [Table 7]

[311]

Signaling FEC Type		K_{sig} = $K_{payload}$	M_{outer}	N_{outer} = K_{sig} + M_{outer}
L1-Basic	Mode 1	200	168	368
	Mode 2			
	Mode 3			
	Mode 4			
	Mode 5			
	Mode 6			
	Mode 7			
L1-Detail	Mode 1	400 ~ 2352		568 ~ 2520
	Mode 2	400 ~ 3072		568 ~ 3240
	Mode 3	400 ~ 6312		568 ~ 6480
	Mode 4			
	Mode 5			
	Mode 6			
	Mode 7			

[312] Meanwhile, referring to FIGs. 7 and 8, it may be appreciated that the LDPC encoders 214 and 315 may be disposed after the BCH encoders 212 and 313, respectively.

[313] Therefore, the L1-basic signaling and the L1-detail signaling may be protected by the concatenation of the BCH outer code and the LDPC inner code.

[314] In detail, the L1-basic signaling and the L1-detail signaling are BCH-encoded, and thus, the BCH parity check bits for the L1-basic signaling are added to the L1-basic signaling and the BCH parity check bits for the L1-detail signaling are added to the L1-detail signaling. Further, the concatenated L1-basic signaling and BCH parity check bits are additionally protected by an LDPC code, and the concatenated L1-detail signaling and BCH parity check bits may be additionally protected by an LDPC code.

[315] Here, it is assumed that an LDPC code for LDPC encoding is a 16K LDPC code, and thus, in the BCH encoders 212 and 213, a systematic BCH code for N_{inner}=16200 (that is, the code length of the 16K LDPC code is 16200 and an LDPC codeword generated by the LDPC encoding may be formed of 16200 bits) may be used to perform outer encoding of the L1-basic signaling and the L1-detail signaling.

[316] The zero padders 213 and 314 pad zero bits. In detail, for the LDPC code, a predetermined number of LDPC information bits defined according to a code rate and a code length is required, and thus, the zero padders 213 and 314 may pad zero bits for the LDPC encoding to generate the predetermined number of LDPC information bits formed of the BCH-encoded bits and zero bits, and output the generated bits to the LDPC encoders 214 and 315, respectively, when the number of BCH-encoded bits is less than the number of LDPC information bits. When the number of BCH-encoded bits is equal to the number of LDPC information bits, zero bits are not padded.

[317] Here, zero bits padded by the zero padders 213 and 314 are padded for the LDPC encoding, and therefore, the padded zero bits padded are not transmitted to the receiver 200 by a shortening operation.

[318] For example, when the number of LDPC information bits of the 16K LDPC code is K_{ldpc}, in order to form K_{ldpc} LDPC information bits, zero bits are padded to some of the LDPC information bits.

[319] In detail, when the number of BCH-encoded bits is N_{outer}, the number of LDPC information bits of the 16K LDPC code is K_{ldpc}, and $N_{outer} < K_{ldpc}$, the zero padders 213 and 314 may pad the K_{ldpc}-N_{outer} zero bits to some of the LDPC information bits, and use the N_{outer} BCH-encoded bits as the remaining portion of the LDPC information bits to generate the LDPC information bits formed of K_{ldpc} bits. However, when $N_{outer}=K_{ldpc}$, zero bits are not padded.

[320] For this purpose, the zero padders 213 and 314 may divide the LDPC information bits into a plurality of bit groups.

[321] For example, the zero padders 213 and 314 may divide the K_{ldpc} LDPC information bits ($i_0, i_1, \ldots, i_{K_{ldpc}-1}$) into $N_{info_group}(=K_{ldpc}/360)$ bit groups based on following Equation 16 or 17. That is, the zero padders 213 and 314 may divide the LDPC information bits into the plurality of bit groups so that the number of bits included in each bit group is 360.

[322]
$$Z_j = \left\{ i_k \middle| j = \left\lfloor \frac{k}{360} \right\rfloor, 0 \le k < K_{ldpc} \right\} \text{ for } 0 \le j < N_{info_group} \quad \ldots. (16)$$

[323]
$$Z_j = \{ i_k \mid 360 \times j \le k < 360 \times (j+1) \} \text{ for } 0 \le j < N_{info_group} \quad \ldots. (17)$$

[324] In above Equations 16 and 17, Z_j represents a j-th bit group.

[325] The parameters N_{outer}, K_{ldpc}, and N_{info_group} for the zero padding for the L1-basic signaling and the L1-detail signaling may be defined as shown in following Table 8. In this case, the zero padders 213 and 314 may determine parameters for the zero padding according to a corresponding mode as shown in following Table 8.

[326] [Table 8]

[327]

Signaling FEC Type	N_{outer}	K_{ldpc}	N_{info_group}
L1-Basic (all modes)	368	3240	9
L1-Detail Mode 1	568 ~ 2520		
L1-Detail Mode 2	568 ~ 3240		
L1-Detail Mode 3	568 ~ 6480	6480	18
L1-Detail Mode 4			
L1-Detail Mode 5			
L1-Detail Mode 6			
L1-Detail Mode 7			

[328] Further, for $0 \le j < N_{info_group}$, each bit group Z_j as shown in FIG. 12 may be formed of 360 bits.

[329] In detail, FIG. 10 illustrates a data format after the L1-basic signaling and the

L1-detail signaling each are LDPC-encoded. In FIG. 10, an LDPC FEC added to the K_{ldpc} LDPC information bits represents the LDPC parity bits generated by the LDPC encoding.

[330] Referring to FIG. 10, the K_{ldpc} LDPC information bits are divided into the N_{info_group} bits groups and each bit group may be formed of 360 bits.

[331] When the number $N_{outer}(= K_{sig}+M_{outer})$ of BCH-encoded bits for the L1-basic signaling and the L1-detail signaling is less than the K_{ldpc}, that is, $N_{outer}(= K_{sig}+M_{outer}) < K_{ldpc}$, for the LDPC encoding, the K_{ldpc} LDPC information bits may be filled with the N_{outer} BCH-encoded bits and the K_{ldpc}-N_{outer} zero-padded bits. In this case, the padded zero bits are not transmitted to the receiver 200.

[332] Hereinafter, a shortening procedure performed by the zero padders 213 and 314 will be described in more detail.

[333] The zero padders 213 and 314 may calculate the number of padded zero bits. That is, to fit the number of bits required for the LDPC encoding, the zero padders 213 and 314 may calculate the number of zero bits to be padded.

[334] In detail, the zero padders 213 and 314 may calculate a difference between the number of LDPC information bits and the number of BCH-encoded bits as the number of padded zero bits. That is, for a given N_{outer}, the zero padders 213 and 314 may calculate the number of padded zero bits as K_{ldpc}-N_{outer}.

[335] Further, the zero padders 213 and 314 may calculate the number of bit groups in which all the bits are padded. That is, the zero padders 213 and 314 may calculate the number of bit groups in which all bits within the bit group are padded by zero bits.

[336] In detail, the zero padders 213 and 314 may calculate the number N_{pad} of groups to which all bits are padded based on following Equation 18 or 19.

[337]
$$N_{pad} = \left\lfloor \frac{K_{ldpc} - N_{outer}}{360} \right\rfloor \quad \text{.... (18)}$$

[338]
$$N_{pad} = \left\lfloor \frac{(K_{ldpc} - M_{outer}) - K_{sig}}{360} \right\rfloor \quad \text{.... (19)}$$

[339] Next, the zero padders 213 and 314 may determine bit groups in which zero bits are padded among a plurality of bit groups based on a shortening pattern, and may pad zero bits to all bits within some of the determined bit groups and some bits within the remaining bit groups.

[340] In this case, the shortening pattern of the padded bit group may be defined as shown in following Table 9. In this case, the zero padders 213 and 314 may determine the shortening patterns according to a corresponding mode as shown in following Table 9.

[341] [Table 9]

[342]

Signaling FEC Type	N_{info_group}	$\pi_S(j)$ $(0 \le j < N_{info_group})$								
		$\pi_S(0)$	$\pi_S(1)$	$\pi_S(2)$	$\pi_S(3)$	$\pi_S(4)$	$\pi_S(5)$	$\pi_S(6)$	$\pi_S(7)$	$\pi_S(8)$
		$\pi_S(9)$	$\pi_S(10)$	$\pi_S(11)$	$\pi_S(12)$	$\pi_S(13)$	$\pi_S(14)$	$\pi_S(15)$	$\pi_S(16)$	$\pi_S(17)$
L1-Basic (for all modes)	9	4	1	5	2	8	6	0	7	3
		-	-	-	-	-	-	-	-	-
L1-Detail Mode 1		7	8	5	4	1	2	6	3	0
		-	-	-	-	-	-	-	-	-
L1-Detail Mode 2		6	1	7	8	0	2	4	3	5
		-	-	-	-	-	-	-	-	-
L1-Detail Mode 3	18	0	12	15	13	2	5	7	9	8
		6	16	10	14	1	17	11	4	3
L1-Detail Mode 4		0	15	5	16	17	1	6	13	11
		4	7	12	8	14	2	3	9	10
L1-Detail Mode 5		2	4	5	17	9	7	1	6	15
		8	10	14	16	0	11	13	12	3
L1-Detail Mode 6		0	15	5	16	17	1	6	13	11
		4	7	12	8	14	2	3	9	10
L1-Detail Mode 7		15	7	8	11	5	10	16	4	12
		3	0	6	9	1	14	17	2	13

[343] Here, $\pi_s(j)$ is an index of a j-th padded bit group. That is, the $\pi_s(j)$ represents a shortening pattern order of the j-th bit group. Further, N_{info_group} is the number of bit groups configuring the LDPC information bits.

[344] In detail, the zero padders 213 and 314 may determine $Z_{\pi_s(0)}, Z_{\pi_s(1)}, \cdots, Z_{\pi_s(N_{pad}-1)}$ as bit groups in which all bits within the bit group are padded by zero bits based on the shortening pattern, and pad zero bits to all bits of the bit groups. That is, the zero padders 213 and 314 may pad zero bits to all bits of a $\pi_s(0)$-th bit group, a $\pi_s(1)$-th bit group,….a $\pi_s(N_{pad}-1)$-th bit group among the plurality of bit groups based on the shortening pattern.

[345] As such, when N_{pad} is not 0, the zero padders 213 and 314 may determine a list of the N_{pad} bit groups, that is, $Z_{\pi_s(0)}, Z_{\pi_s(1)}, \cdots, Z_{\pi_s(N_{pad}-1)}$ based on above Table 9, and pad zero bits to all bits within the determined bit group.

[346] However, when the N_{pad} is 0, the foregoing procedure may be omitted.

[347] Meanwhile, since the number of all the padded zero bits is $K_{ldpc}-N_{outer}$ and the number of zero bits padded to the N_{pad} bit groups is $360\times N_{pad}$, the zero padders 213 and 314 may additionally pad zero bits to $K_{ldpc}-N_{outer}-360\times N_{pad}$ LDPC information bits.

[348] In this case, the zero padders 213 and 314 may determine a bit group to which zero bits are additionally padded based on the shortening pattern, and may additionally pad zero bits from a head portion of the determined bit group.

[349] In detail, the zero padders 213 and 314 may determine $Z_{\pi_s(N_{pad})}$ as a bit group to which zero bits are additionally padded based on the shortening pattern, and may additionally pad zero bits to the $Kl_{dpc}-N_{outer}-360\times N_{pad}$ bits positioned at the head portion of $Z_{\pi_s(N_{pad})}$. Therefore, the $K_{ldpc}-N_{outer}-360\times N_{pad}$ zero bits may be padded from a first bit of the $\pi_s(N_{pad})$-th bit group.

[350] As a result, for $Z_{\pi_s(N_{pad})}$, zero bits may be additionally padded to the $K_{ldpc}-N_{bch}$ -

$360\times N_{pad}$ bits positioned at the head portion of the $Z_{\pi_s(N_{pad})}$.

[351] Meanwhile, the foregoing example describes that K_{ldpc}-N_{outer}-$360\times N_{pad}$ zero bits are padded from a first bit of the $Z_{\pi_s(N_{pad})}$, which is only one example. Therefore, the position at which zero bits are padded in the $Z_{\pi_s(N_{pad})}$ may be changed. For example, the K_{ldpc}-N_{outer}-$360\times N_{pad}$ zero bits may be padded to a middle portion or a last portion of the $Z_{\pi_s(N_{pad})}$ or may also be padded at any position of the $Z_{\pi_s(N_{pad})}$.

[352] Next, the zero padders 213 and 314 may map the BCH-encoded bits to the positions at which zero bits are not padded to configure the LDPC information bits.

[353] Therefore, the N_{outer} BCH-encoded bits are sequentially mapped to the bit positions at which zero bits in the K_{ldpc} LDPC information bits (i_0, i_1, ..., $i_{K_{ldpc}-1}$) are not padded, and thus, the K_{ldpc} LDPC information bits may be formed of the N_{outer} BCH-encoded bits and the K_{ldpc}-N_{outer} information bits.

[354] Meanwhile, the padded zero bits are not transmitted to the receiver 200. As such, a procedure of padding the zero bits or a procedure of padding the zero bits and then not transmitting the padded zero bits to the receiver 200 may be called shortening.

[355] The LDPC encoders 214 and 315 perform LDPC encoding on the L1-basic signaling and the L1-detail signaling, respectively.

[356] In detail, the LDPC encoders 214 and 315 may perform LDPC encoding on the LDPC information bits output from the zero padders 213 and 31 to generate LDPC parity bits, and output an LDPC codeword including the LDPC information bits and the LDPC parity bits to the parity permutators 215 and 316, respectively.

[357] That is, K_{ldpc} bits output from the zero padder 213 may include K_{sig} L1-basic signaling bits, M_{outer}(=N_{outer}-K_{sig}) BCH parity check bits, and K_{ldpc}-N_{outer} padded zero bits, which may configure K_{ldpc} LDPC information bits i=(i_0, i_1, ..., $i_{K_{ldpc}-1}$) for the LDPC encoder 214.

[358] Further, the K_{ldpc} bits output from the zero padder 314 may include the K_{sig} L1-detail signaling bits, the M_{outer}(=N_{outer}-K_{sig}) BCH parity check bits, and the (K_{ldpc}-N_{outer}) padded zero bits, which may configure the K_{ldpc} LDPC information bits i=(i_0, i_1, ..., $i_{K_{ldpc}-1}$) for the LDPC encoder 315.

[359] In this case, the LDPC encoders 214 and 315 may systematically perform the LDPC encoding on the K_{ldpc} LDPC information bits to generate an LDPC codeword Λ=(c_0, c_1, ..., $c_{N_{inner}-1}$) = (i_0, i_1, ..., $i_{K_{ldpc}-1}$, p_0, p_1, ..., $p_{N_{inner}-K_{ldpc}-1}$) formed of N_{inner} bits.

[360] In the L1-basic modes and the L1-detail modes 1 and 2, the LDPC encoders 214 and 315 may encode the L1-basic signaling and the L1-detail signaling at a code rate of

3/15 to generate 16200 LDPC codeword bits. In this case, the LDPC encoders 214 and 315 may perform the LDPC encoding based on above Table 2.

[361] Further, in the L1-detail modes 3, 4, 5 6, and 7, the LDPC encoder 315 may encode the L1-detail signaling at a code rate of 6/15 to generate the 16200 LDPC codeword bits. In this case, the LDPC encoder 315 may perform the LDPC encoding based on above Table 4.

[362] The code rate and the code length for the L1-basic signaling and the L1-detail signaling are as shown in above Table 5, and the number of LDPC information bits are as shown in above Table 8.

[363] The parity permutators 215 and 316 perform parity permutation. That is, the parity permutators 215 and 316 may perform permutation only on the LDPC parity bits among the LDPC information bits and the LDPC parity bits.

[364] In detail, the parity permutators 215 and 316 may perform the permutation only on the LDPC parity bits in the LDPC codewords output from the LDPC encoders 214 and 315, and output the parity permutated LDPC codewords to the repeaters 216 and 317, respectively. The parity permutator 316 may output the parity permutated LDPC codeword to an additional parity generator 319. In this case, the additional parity generator 319 may use the parity permutated LDPC codeword output from the parity permutator 316 to generate additional parity bits.

[365] For this purpose, the parity permutators 215 and 316 may include a parity interleaver (not illustrated) and a group-wise interleaver (not illustrated).

[366] First, the parity interleaver may interleave only the LDPC parity bits among the LDPC information bits and the LDPC parity bits configuring the LDPC codeword. However, the parity interleaver may perform the parity interleaving only in the cases of the L1-detail modes 3, 4, 5, 6 and 7. That is, since the L1-basic modes and the L1-detail modes 1 and 2 include the parity interleaving as a portion of the LDPC encoding process, in the L1-basic modes and the L1-detail modes 1 and 2, the parity interleaver may not perform the parity interleaving.

[367] In the mode of performing the parity interleaving, the parity interleaver may interleave the LDPC parity bits based on following Equation 20.

[368] $u_i = c_i$ for $0 \le i < K_{ldpc}$ (information bits are not interleaved)

[369] $$u_{K_{ldpc}+360t+s} = c_{K_{ldpc}+27s+t} \text{ for } 0 \le s < 360,\ 0 \le t < 27 \quad \ldots (20)$$

[370] In detail, based on above Equation 20, the LDPC codeword $(c_0, c_1, \ldots, c_{N_{inner}-1})$ is parity-interleaved by the parity interleaver and an output of the parity interleaver may be represented by $U = (u_0, u_1, \ldots, u_{N_{inner}-1})$.

[371] Meanwhile, since the L1-basic modes and the L1-detail modes 1 and 2 do not use the

parity interleaver, an output U = (u_0, u_1, ..., $u_{N_{inner}-1}$) of the parity interleaver may be represented as following Equation 21.

[372] $u_i=c_i$ for $0 \le i < N_{inner}$ .... (21)

[373] The group-wise interleaver may perform the group-wise interleaving on the output of the parity interleaver.

[374] Here, as described above, the output of the parity interleaver may be an LDPC codeword parity-interleaved by the parity interleaver or may be an LDPC codeword which is not parity-interleaved by the parity interleaver.

[375] Therefore, when the parity interleaving is performed, the group-wise interleaver may perform the group-wise interleaving on the parity interleaved LDPC codeword, and when the parity interleaving is not performed, the group-wise interleaver may perform the group-wise interleaving on the LDPC codeword which is not parity-interleaved.

[376] In detail, the group-wise interleaver may interleave the output of the parity interleaver in a bit group unit.

[377] For this purpose, the group-wise interleaver may divide an LDPC codeword output from the parity interleaver into a plurality of bit groups. As a result, the LDPC parity bits output from the parity interleaver may be divided into a plurality of bit groups.

[378] In detail, the group-wise interleaver may divide the LDPC-encoded bits (u_0, u_1, ..., $u_{N_{inner}-1}$) output from the parity interleaver into $N_{group}(=N_{inner}/360)$ bit groups based on following Equation 22.

[379] $X_j=\{u_k \mid 360\times j \le k < 360\times(j+1), 0 \le k < N_{inner}\}$ for $0 \le j < N_{group}$ ... (22)

[380] In above Equation 22, X_j represents a j-th bit group.

[381] FIG. 11 illustrates an example of dividing the LDPC codeword output from the parity interleaver into a plurality of bit groups.

[382] Referring to FIG. 11, the LDPC codeword is divided into $N_{group}(=N_{inner}/360)$ bit groups, and each bit group X_j for $0 \le j < N_{group}$ is formed of 360 bits.

[383] As a result, the LDPC information bits formed of K_{ldpc} bits may be divided into K_{ldpc} / 360 bit groups and the LDPC parity bits formed of N_{inner}-K_{ldpc} bits may be divided into N_{inner}-K_{ldpc}/360 bit groups.

[384] Further, the group-wise interleaver performs the group-wise interleaving on the LDPC codeword output from the parity interleaver.

[385] In this case, the group-wise interleaver does not perform interleaving on the LDPC information bits, and may perform the interleaving only on the LDPC parity bits to change the order of the plurality of bit groups configuring the LDPC parity bits.

[386] As a result, the LDPC information bits among the LDPC bits may not be interleaved by the group-wise interleaver but the LDPC parity bits among the LDPC bits may be interleaved by the group-wise interleaver. In this case, the LDPC parity bits may be in-

terleaved in a group unit.

[387] In detail, the group-wise interleaver may perform the group-wise interleaving on the LDPC codeword output from the parity interleaver based on following Equation 23.

[388] $Y_j=X_j$, $0\leq j<K_{ldpc}/360$

[389] $Y_j=X_{\pi p(j)}$ $K_{ldpc}/360\leq j<N_{group}$ .... (23)

[390] Here, X_j represents a j-th bit group among the plurality of bit groups configuring the LDPC codeword, that is, the j-th bit group which is not group-wise interleaved, and Y_j represents the group-wise interleaved j-th bit group. Further, $\pi_p(j)$ represents a per-mutation order for the group-wise interleaving.

[391] Meanwhile, the permutation order may be defined based on following Table 10 and Table 11. Here, Table 10 shows a group-wise interleaving pattern of a parity portion in the L1-basic modes and the L1-detail modes 1 and 2, and Table 11 shows a group-wise interleaving pattern of a parity portion for the L1-detail modes 3, 4, 5, 6 and 7.

[392] In this case, the group-wise interleaver may determine the group-wise interleaving pattern according to a corresponding mode shown in following Tables 10 and 11.

[393] [Table 10]

[394]

Signaling FEC Type	N_{group}	Order of group-wise interleaving $\pi_p(j)$ $(9 \leq j < 45)$											
		$\pi_p(9)$	$\pi_p(10)$	$\pi_p(11)$	$\pi_p(12)$	$\pi_p(13)$	$\pi_p(14)$	$\pi_p(15)$	$\pi_p(16)$	$\pi_p(17)$	$\pi_p(18)$	$\pi_p(19)$	$\pi_p(20)$
		$\pi_p(21)$	$\pi_p(22)$	$\pi_p(23)$	$\pi_p(24)$	$\pi_p(25)$	$\pi_p(26)$	$\pi_p(27)$	$\pi_p(28)$	$\pi_p(29)$	$\pi_p(30)$	$\pi_p(31)$	$\pi_p(32)$
		$\pi_p(33)$	$\pi_p(34)$	$\pi_p(35)$	$\pi_p(36)$	$\pi_p(37)$	$\pi_p(38)$	$\pi_p(39)$	$\pi_p(40)$	$\pi_p(41)$	$\pi_p(42)$	$\pi_p(43)$	$\pi_p(44)$
L1-Basic (all modes)	45	20	23	25	32	38	41	18	9	10	11	31	24
		14	15	26	40	33	19	28	34	16	39	27	30
		21	44	43	35	42	36	12	13	29	22	37	17
L1-Detail Mode 1		16	22	27	30	37	44	20	23	25	32	38	41
		9	10	17	18	21	33	35	14	28	12	15	19
		11	24	29	34	36	13	40	43	31	26	39	42
L1-Detail Mode 2		9	31	23	10	11	25	43	29	36	16	27	34
		26	18	37	15	13	17	35	21	20	24	44	12
		22	40	19	32	38	41	30	33	14	28	39	42

[395] [Table 11]

[396]

Signaling FEC Type	N_{group}	Order of group-wise interleaving $\pi_p(j)$ $(18 \leq j < 45)$												
		$\pi_p(18)$	$\pi_p(19)$	$\pi_p(20)$	$\pi_p(21)$	$\pi_p(22)$	$\pi_p(23)$	$\pi_p(24)$	$\pi_p(25)$	$\pi_p(26)$	$\pi_p(27)$	$\pi_p(28)$	$\pi_p(29)$	$\pi_p(30)$	$\pi_p(31)$
		$\pi_p(32)$	$\pi_p(33)$	$\pi_p(34)$	$\pi_p(35)$	$\pi_p(36)$	$\pi_p(37)$	$\pi_p(38)$	$\pi_p(39)$	$\pi_p(40)$	$\pi_p(41)$	$\pi_p(42)$	$\pi_p(43)$	$\pi_p(44)$	
L1-Detail Mode 3	45	19	37	30	42	23	44	27	40	21	34	25	32	29	24
		26	35	39	20	18	43	31	36	38	22	33	28	41	
L1-Detail Mode 4		20	35	42	39	26	23	30	18	28	37	32	27	44	43
		41	40	38	36	34	33	31	29	25	24	22	21	19	
L1-Detail Mode 5		19	37	33	26	40	43	22	29	24	35	44	31	27	20
		21	39	25	42	34	18	32	38	23	30	28	36	41	
L1-Detail Mode 6		20	35	42	39	26	23	30	18	28	37	32	27	44	43
		41	40	38	36	34	33	31	29	25	24	22	21	19	
L1-Detail Mode 7		44	23	29	33	24	28	21	27	42	18	22	31	32	37
		43	30	25	35	20	34	39	36	19	41	40	26	38	

[397] Hereinafter, for the group-wise interleaving pattern in the L1-detail mode 2 as an example, an operation of the group-wise interleaver will be described.

[398] In the L1-detail mode 2, the LDPC encoder 315 performs LDPC encoding on 3240

LDPC information bits at a code rate of 3/15 to generate 12960 LDPC parity bits. In this case, an LDPC codeword may be formed of 16200 bits.

[399] Each bit group is formed of 360 bits, and as a result the LDPC codeword formed of 16200 bits is divided into 45 bit groups.

[400] Here, since the number of the LDPC information bits is 3240 and the number of the LDPC parity bits is 12960, a 0-th bit group to an 8-th bit group correspond to the LDPC information bits and a 9-th bit group to a 44-th bit group correspond to the LDPC parity bits.

[401] In this case, the group-wise interleaver does not perform interleaving on the bit groups configuring the LDPC information bits, that is, a 0-th bit group to a 8-th bit group based on above Equation 28 and Table 10, but may interleave the bit groups configuring the LDPC parity bits, that is, a 9-th bit group to a 44-th bit group in a group unit to change an order of the 9-th bit group to the 44-th bit group.

[402] In detail, in the L1-detail mode 2 in above Table 10, above Equation 28 may be represented like $Y_0=X_0$, $Y_1=X_1$, ..., $Y_7=X_7$, $Y_8=X_8$, $Y_9=X_{\pi p(9)}=X_9$, $Y_{10}=X_{\pi p(10)}=X_{31}$, $Y_{11}=X_{\pi p(11)}=X_{23}$, ..., $Y_{42}=X_{\pi p(42)}=X_{28}$, $Y_{43}=X_{\pi p(43)}=X_{39}$, $Y_{44}=X_{\pi p(44)}=X_{42}$.

[403] Therefore, the group-wise interleaver does not change an order of the 0-th bit group to the 8-th bit group including the LDPC information bits but may change an order of the 9-th bit group to the 44-th bit group including the LDPC parity bits.

[404] In detail, the group-wise interleaver may change the order of the bit groups from the 9-th bit group to the 44-th bit group so that the 9-th bit group is positioned at the 9-th position, the 31-th bit group is positioned at the 10-th position, the 23-th bit group is positioned at the 11-th position,..., the 28-th bit group is positioned at the 42-th position, the 39-th bit group is positioned at the 43-th position, the 42-th bit group is positioned at the 44-th position.

[405] As described below, since the puncturers 217 and 318 perform puncturing from the last parity bit, the parity bit groups may be arranged in an inverse order of the puncturing pattern by the parity permutation. That is, the first bit group to be punctured is positioned at the last bit group.

[406] The foregoing example describes that only the parity bits are interleaved, which is only one example. That is, the parity permutators 215 and 316 may also interleave the LDPC information bits. In this case, the parity permutators 215 and 316 may interleave the LDPC information bits with identity and output the LDPC information bits having the same order before the interleaving so that the order of the LDPC information bits is not changed.

[407] The repeaters 216 and 317 may repeat at least some bits of the parity permutated LDPC codeword at a position subsequent to the LDPC information bits, and output the repeated LDPC codeword, that is, the LDPC codeword bits including the repetition

bits, to the puncturers 217 and 318. The repeater 317 may also output the repeated LDPC codeword to the additional parity generator 319. In this case, the additional parity generator 319 may use the repeated LDPC codeword to generate the additional parity bits.

[408] In detail, the repeaters 216 and 317 may repeat a predetermined number of LDPC parity bits after the LDPC information bits. That is, the repeaters 216 and 317 may add the predetermined number of repeated LDPC parity bits after the LDPC information bits. Therefore, the repeated LDPC parity bits are positioned between the LDPC information bits and the LDPC parity bits within the LDPC codeword.

[409] Therefore, since the predetermined number of bits within the LDPC codeword after the repetition may be repeated and additionally transmitted to the receiver 200, the foregoing operation may be referred to as repetition.

[410] The term "adding" represents disposing the repetition bits between the LDPC information bits and the LDPC parity bits so that the bits are repeated.

[411] The repetition may be performed only on the L1-basic mode 1 and the L1-detail mode 1, and may not be performed on the other modes. In this case, the repeaters 216 and 317 do not perform the repetition and may output the parity permutated LDPC codeword to the puncturers 217 and 318.

[412] Hereinafter, a method for performing repetition will be described in more detail.

[413] The repeaters 216 and 317 may calculate a number N_{repeat} of bits additionally transmitted per an LDPC codeword based on following Equation 24.

[414]
$$N_{repeat} = 2 \times \lfloor C \times N_{outer} \rfloor + D \quad \text{.... (24)}$$

[415] In above Equation 24, C has a fixed number and D may be an even integer. Referring to above Equation 24, it may be appreciated that the number of bits to be repeated may be calculated by multiplying C by a given N_{outer} and adding D thereto.

[416] The parameters C and D for the repetition may be selected based on following Table 12. That is, the repeaters 216 and 317 may determine the C and D based on a corresponding mode as shown in following Table 12.

[417] [Table 12]

[418]

	N_{outer}	K_{sig}	K_{ldpc}	C	D	N_{ldpc_parity} (= $N_{inner} - K_{ldpc}$)	η_{MOD}
L1-Basic Mode 1	368	200	3240	0	3672	12960	2
L1-Detail Mode 1	568 ~ 2520	400 ~ 2352	3240	61/16	– 508	12960	2

[419] Further, the repeaters 216 and 317 may repeat N_{repeat} LDPC parity bits.

[420] In detail, when $N_{repeat} \leq N_{ldpc_parity}$, the repeaters 216 and 317 may add first N_{repeat} bits of the parity permutated LDPC parity bits to the LDPC information bits as illustrated in FIG. 12. That is, the repeaters 216 and 317 may add a first LDPC parity bit among

the parity permutated LDPC parity bits as an N_{repeat}-th LDPC parity bit after the LDPC information bits.

[421] When $N_{repeat} > N_{ldpc_parity}$, the repeaters 216 and 317 may add the parity permutated N_{ldpc_parity} LDPC parity bits to the LDPC information bits as illustrated in FIG. 15, and may additionally add an N_{repeat}-N_{ldpc_parity} number of the parity permutated LDPC parity bits to the N_{ldpc_parity} LDPC parity bits which are first added. That is, the repeaters 216 and 317 may add all the parity permutated LDPC parity bits after the LDPC information bits and additionally add the first LDPC parity bit to the N_{repeat}-N_{ldpc_parity}-th LDPC parity bit among the parity permutated LDPC parity bits after the LDPC parity bits which are first added.

[422] Therefore, in the L1-basic mode 1 and the L1-detail mode 1, the additional N_{repeat} bits may be selected within the LDPC codeword and transmitted.

[423] The puncturers 217 and 318 may puncture some of the LDPC parity bits included in the LDPC codeword output from the repeaters 216 and 317, and output a punctured LDPC codeword (that is, the remaining LDPC codeword bits other than the punctured bits and also referred to as an LDPC codeword after puncturing) to the zero removers 218 and 321. Further, the puncturer 318 may provide information (for example, the number and positions of punctured bits, etc.) about the punctured LDPC parity bits to the additional parity generator 319. In this case, the additional parity generator 319 may generate additional parity bits based thereon.

[424] As a result, after going through the parity permutation, some LDPC parity bits may be punctured.

[425] In this case, the punctured LDPC parity bits are not transmitted in a frame in which L1 signaling bits are transmitted. In detail, the punctured LDPC parity bits are not transmitted in a current frame in which the L1-signaling bits are transmitted, and in some cases, the punctured LDPC parity bits may be transmitted in a frame before the current frame, which will be described with reference to the additional parity generator 319.

[426] For this purpose, the puncturers 217 and 318 may determine the number of LDPC parity bits to be punctured per LDPC codeword and a size of one coded block.

[427] In detail, the puncturers 217 and 318 may calculate a temporary number N_{punc_temp} of LDPC parity bits to be punctured based on following Equation 25. That is, for a given N_{outer}, the puncturers 217 and 318 may calculate the temporary number N_{punc_temp} of LDPC parity bits to be punctured based on following Equation 25.

[428] .... (25)

$$N_{punc_temp} = \lfloor A \times (K_{ldpc} - N_{outer}) \rfloor + B$$

[429] Referring to above Equation 25, the temporary size of bits to be punctured may be

calculated by adding a constant integer B to an integer obtained from a result of multiplying a shortening length (that is, K_{ldpc}-N_{outer}) by a preset constant A value. In the present exemplary embodiment, it is apparent that the constant A value is set at a ratio of the number of bits to be punctured to the number of bits to be shortened but may be variously set according to requirements of a system.

[430] The B value is a value which represents a length of bits to be punctured even when the shortening length is 0, and thus, represents a minimum length that the punctured bits can have. Further, the A and B values serve to adjust an actually transmitted code rate. That is, to prepare for a case in which the length of information bits, that is, the length of the L1 signaling is short or a case in which the length of the L1 signaling is long, the A and B values serve to adjust the actually transmitted code rate to be reduced.

[431] The above K_{ldpc}, A and B are listed in following Table 13 which shows parameters for puncturing. Therefore, the puncturers 217 and 318 may determine the parameters for puncturing according to a corresponding mode as shown in following Table 13.

[432] [Table 13]

[433]

Signaling FEC Type		N_{outer}	K_{ldpc}	A	B	N_{ldpc_parity}	η_{MOD}
L1-Basic	Mode 1	368	3240	0	9360	12960	2
	Mode 2				11460		2
	Mode 3				12360		2
	Mode 4				12292		4
	Mode 5				12350		6
	Mode 6				12432		8
	Mode 7				12776		8
L1-Detail	Mode 1	568 ~ 2520		7/2	0		2
	Mode 2	568 ~ 3240		2	6036		2
	Mode 3	568 ~ 6480	6480	11/16	4653	9720	2
	Mode 4			29/32	3200		4
	Mode 5			3/4	4284		6
	Mode 6			11/16	4900		8
	Mode 7			49/256	8246		8

[434] The puncturers 217 and 318 may calculate a temporary size N_{FEC_temp} of one coded block as shown in following Equation 31. Here, the number N_{ldpc_parity} of LDPC parity bits according to a corresponding mode is shown as above Table 13.

[435] $N_{FEC_temp}=N_{outer}+N_{ldpc_parity}-N_{punc_temp}$ .... (26)

[436] Further, the puncturers 217 and 318 may calculate a size N_{FEC} of one coded block as shown in following Equation 27.

[437] .... (27)

$$N_{FEC} = \left\lceil \frac{N_{FEC_temp}}{\eta_{MOD}} \right\rceil \times \eta_{MOD}$$

[438] In above Equation 27, η_{MOD} is a modulation order. For example, when the L1-basic signaling and the L1-detail signaling are modulated by QPSK, 16-QAM, 64-QAM or

256-QAM according to a corresponding mode, η_{MOD} may be 2, 4, 6 and 8 as shown in above Table 13. Meanwhile, according to above Equation 27, the N_{FEC} may be an integer multiple of the modulation order.

[439] Further, the puncturers 217 and 318 may calculate the number N_{punc} of LDPC parity bits to be punctured based on following Equation 28.

[440] $N_{punc}=N_{punc_temp}-(N_{FEC}-N_{FEC_temp})$ .... (28)

[441] Here, N_{punc} is 0 or a positive integer. Further, N_{FEC} is the number of bits of an in-formation block which are obtained by subtracting N_{punc} bits to be punctured from N_{outer} $+N_{ldpc_parity}$ bits obtained by performing the BCH encoding and the LDPC encoding on K_{sig} information bits. That is, N_{FEC} is the number of bits other than the repetition bits among the actually transmitted bits, and may be called the number of shortened and punctured LDPC codeword bits.

[442] Referring to the foregoing process, the puncturers 217 and 318 multiplies A by the number of padded zero bits, that is, a shortening length and adding B to a result to calculate the temporary number N_{punc_temp} of LDPC parity bits to be punctured.

[443] Further, the puncturers 217 and 318 calculate the temporary number N_{FEC_temp} of LDPC codeword bits to constitute the LDPC codeword after puncturing and shortening based on the N_{punc_temp}.

[444] In detail, the LDPC information bits are LDPC-encoded, and the LDPC parity bits generated by the LDPC encoding are added to the LDPC information bits to configure the LDPC codeword. Here, the LDPC information bits include the BCH-encoded bits in which the L1-basic signaling and the L1-detail signaling are BCH encoded, and in some cases, may further include padded zero bits.

[445] In this case, since the padded zero bits are LDPC-encoded, and then, are not transmitted to the receiver 200, the shortened LDPC codeword, that is, the LDPC codeword (that is, shortened LDPC codeword) except the padded zero bits may be formed of the BCH-encoded bits and LDPC parity bits.

[446] Therefore, the puncturers 217 and 318 subtract the temporary number of LDPC parity bits to be punctured from a sum of the number of BCH-encoded bits and the number of LDPC parity bits to calculate the N_{FEC_temp}.

[447] The punctured and shortened LDPC codeword (that is, LDPC codeword bits remaining after puncturing and shortening) are mapped to constellation symbols by various modulation schemes such as QPSK, 16-QAM, 64-QAM or 256-QAM according to a corresponding mode, and the constellation symbols may be transmitted to the receiver 200 through a frame.

[448] Therefore, the puncturers 217 and 318 determine the number N_{FEC} of LDPC codeword bits to constitute the LDPC codeword after puncturing and shortening based on N_{FEC_temp}, N_{FEC} being an integer multiple of the modulation order, and determine the

number N_{punc} of bits which need to be punctured based on LDPC codeword bits after shortening to obtain the N_{FEC}.

[449] When zero bits are not padded, an LDPC codeword may be formed of BCH-encoded bits and LDPC parity bits, and the shortening may be omitted.

[450] Further, in the L1-basic mode 1 and the L1-detail mode 1, repetition is performed, and thus, the number of shortened and punctured LDPC codeword bits is equal to $N_{FEC}+N_{repeat}$.

[451] The puncturers 217 and 318 may puncture the LDPC parity bits as many as the calculated number.

[452] In this case, the puncturers 217 and 318 may puncture the last N_{punc} bits of all the LDPC codewords. That is, the puncturers 217 and 318 may puncture the N_{punc} bits from the last LDPC parity bits.

[453] In detail, when the repetition is not performed, the parity permutated LDPC codeword includes only LDPC parity bits generated by the LDPC encoding.

[454] In this case, the puncturers 217 and 318 may puncture the last N_{punc} bits of all the parity permutated LDPC codewords. Therefore, the N_{punc} bits from the last LDPC parity bits among the LDPC parity bits generated by the LDPC encoding may be punctured.

[455] When the repetition is performed, the parity permutated and repeated LDPC codeword includes the repeated LDPC parity bits and the LDPC parity bits generated by the LDPC encoding.

[456] In this case, the puncturers 217 and 318 may puncture the last N_{punc} bits of all the parity permutated and repeated LDPC codewords, respectively, as illustrated in FIGs. 14 and 15.

[457] In detail, the repeated LDPC parity bits are positioned between the LDPC information bits and the LDPC parity bits generated by the LDPC encoding, and thus, the puncturers 217 and 318 may puncture the N_{punc} bits from the last LDPC parity bits among the LDPC parity bits generated by the LDPC encoding, respectively.

[458] As such, the puncturers 217 and 318 may puncture the N_{punc} bits from the last LDPC parity bits, respectively.

[459] N_{punc} is 0 or a positive integer and the repetition may be applied only to the L1-basic mode 1 and the L1-detail mode 1.

[460] The foregoing example describes that the repetition is performed, and then, the puncturing is performed, which is only one example. In some cases, after the puncturing is performed, the repetition may be performed.

[461] The additional parity generator 319 may select bits from the LDPC parity bits to generate additional parity (AP) bits.

[462] In this case, the additional parity bits may be selected from the LDPC parity bits

generated based on the L1-detail signaling transmitted in a current frame, and transmitted to the receiver 200 through a frame before the current frame, that is, a previous frame.

[463] In detail, the L1-detail signaling is LDPC-encoded, and the LDPC parity bits generated by the LDPC encoding are added to the L1-detail signaling to configure an LDPC codeword.

[464] Further, puncturing and shortening are performed on the LDPC codeword, and the punctured and shortened LDPC codeword may be mapped to a frame to be transmitted to the receiver 200. Here, when the repetition is performed according to a corresponding mode, the punctured and shortened LDPC codeword may include the repeated LDPC parity bits.

[465] In this case, the L1-detail signaling corresponding to each frame may be transmitted to the receiver 200 through each frame, along with the LDPC parity bits. For example, the punctured and shortened LDPC codeword including the L1-detail signaling corresponding to an (i-1)-th frame may be mapped to the (i-1)-th frame to be transmitted to the receiver 200, and the punctured and shortened LDPC codeword including the L1-detail signaling corresponding to the i-th frame may be mapped to the i-th frame to be transmitted to the receiver 200.

[466] The additional parity generator 319 may select at least some of the LDPC parity bits generated based on the L1-detail signaling transmitted in the i-th frame to generate the additional parity bits.

[467] In detail, some of the LDPC parity bits generated by performing the LDPC encoding on the L1-detail signaling are punctured, and then, are not transmitted to the receiver 200. In this case, the additional parity generator 319 may select at least some of the punctured LDPC parity bits among the LDPC parity bits generated by performing the LDPC encoding on the L1-detail signaling transmitted in the i-th frame, thereby generating the additional parity bits.

[468] Further, the additional parity generator 319 may select at least some of the LDPC parity bits to be transmitted to the receiver 200 through the i-th frame to generate the additional parity bits.

[469] In detail, the LDPC parity bits included in the punctured and shortened LDPC codeword to be mapped to the i-th frame may be configured of only the LDPC parity bits generated by the LDPC encoding according to a corresponding mode or the LDPC parity bits generated by the LDPC encoding and the repeated LDPC parity bits.

[470] In this case, the additional parity generator 319 may select at least some of the LDPC parity bits included in the punctured and shortened LDPC codeword to be mapped to the i-th frame to generate the additional parity bits.

[471] The additional parity bits may be transmitted to the receiver 200 through the frame

before the i-th frame, that is, the (i-1)-th frame.

[472] That is, the transmitter 100 may not only transmit the punctured and shortened LDPC codeword including the L1-detail signaling corresponding to the (i-1)-th frame but also transmit the additional parity bits generated based on the L1-detail signaling transmitted in the i-th frame to the receiver 200 through the (i-1)-th frame.

[473] In this case, the frame in which the additional parity bits are transmitted may be temporally the most previous frame among the frames before the current frame.

[474] For example, the additional parity bits have the same bootstrap major/minor version as the current frame among the frames before the current frame, and may be transmitted in temporally the most previous frame.

[475] In some cases, the additional parity generator 319 may not generate the additional parity bits.

[476] In this case, the transmitter 100 may transmit information about whether additional parity bits for an L1-detail signaling of a next frame are transmitted through the current frame to the receiver 200 using an L1-basic signaling transmitted through the current frame.

[477] For example, the use of the additional parity bits for the L1-detail signaling of the next frame having the same bootstrap major/minor version as the current frame may be signaled through a field L1B_L1_Detail_additional_parity_mode of the L1-basic parameter of the current frame. In detail, when the L1B_L1_Detail_additional_parity_mode in the L1-basic parameter of the current frame is set to be '00', additional parity bits for the L1-detail signaling of the next frame are not transmitted in the current frame.

[478] As such, to additionally increase robustness of the L1-detail signaling, the additional parity bits may be transmitted in the frame before the current frame in which the L1-detail signaling of the current frame is transmitted.

[479] FIG. 16 illustrates an example in which the additional parity bits for the L1-detail signaling of the i-th frame are transmitted in a preamble of the (i-1)-th frame.

[480] FIG. 16 illustrates that the L1-detail signaling transmitted through the i-th frame is segmented into M blocks by segmentation and each of the segmented blocks is FEC encoded.

[481] Therefore, M number of LDPC codewords, that is, an LDPC codeword including LDPC information bits L1-D(i)_1 and parity bits parity for L1-D(i)_1 therefor,…, and an LDPC codeword including LDPC information bits L1-D(i)_M and parity bits parity for L1-D(i)_M therefor are mapped to the i-th frame to be transmitted to the receiver 200.

[482] In this case, the additional parity bits generated based on the L1-detail signaling transmitted in the i-th frame may be transmitted to the receiver 200 through the (i-1)-th

frame.

[483] In detail, the additional parity bits, that is, AP for L1-D(i)_1,...AP for L1-D(i)_M generated based on the L1-detail signaling transmitted in the i-th frame may be mapped to the preamble of the (i-1)-th frame to be transmitted to the receiver 200. As a result of using the additional parity bits, a diversity gain for the L1 signaling may be obtained.

[484] Hereinafter, a method for generating additional parity bits will be described in detail.

[485] The additional parity generator 319 calculates a temporary number N_{AP_temp} of additional parity bits based on following Equation 29.

[486]

$$N_{AP_temp} = \min \left\{ \begin{array}{c} 0.5 \times K \times (N_{outer} + N_{ldpc_parity} - N_{punc} + N_{repeat}), \\ (N_{ldpc_parity} + N_{punc} + N_{repeat}) \end{array} \right\}, K=0,1,2$$

.... (29)

[487] In above Equation 29,

$$\min(a,b) = \begin{cases} a, \text{if } a \leq b \\ b, \text{if } b < a \end{cases}.$$

[488] Further, K represents a ratio of the additional parity bits to a half of a total number of bits of a transmitted coded L1-detail signaling block (that is, bits configuring the L1-detail signaling block repeated, punctured, and have the zero bits removed (that is, shortened)).

[489] In this case, K corresponds to an L1B_L1_Detail_additional_parity_mode field of the L1-basic signaling. Here, a value of the L1B_L1_Detail_additional_parity_mode associated with the L1-detail signaling of the i-th frame (that is, frame (#i)) may be transmitted in the (i-1)-th frame (that is, frame (#i-1)).

[490] As described above, when L1 detail modes are 2, 3, 4, 5, 6 and 7, since repetition is not performed, in above Equation 39, N_{repeat} is 0.

[491] Further, the additional parity generator 319 calculates the number N_{AP} of additional parity bits based on following Equation 30. Therefore, the number N_{AP} of additional parity bits may be an integer multiple of a modulation order.

[492] .... (30)

$$N_{AP} = \left\lfloor \frac{N_{AP_temp}}{\eta_{MOD}} \right\rfloor \times \eta_{MOD}$$

[493] Here, $\lfloor \chi \rfloor$ is a maximum integer which is not greater than x. Here, η_{MOD} is the modulation order. For example, when the L1-detail signaling is modulated by QPSK,

16-QAM, 64-QAM or 256-QAM according to a corresponding mode, the η_{MOD} may be 2, 4, 6 or 8.

[494] As such, the number of additional parity bits to be generated may be determined based on the total number of bits transmitted in the current frame.

[495] Next, the additional parity generator 319 may select bits as many as the number of bits calculated in the LDPC parity bits to generate the additional parity bits.

[496] In detail, when the number of punctured LDPC parity bits is equal to or greater than the number of additional parity bits to be generated, the additional parity generator 319 may select bits as many as the calculated number from the first LDPC parity bit among the punctured LDPC parity bits to generate the additional parity bits.

[497] When the number of punctured LDPC parity bits is less than the number of additional parity bits to be generated, the additional parity generator 319 may first select all the punctured LDPC parity bits and additionally select bits as many as the number obtained by subtracting the number of punctured LDPC parity bits from the number of additional parity bits to be generated, from the first LDPC parity bit among the LDPC parity bits included in the LDPC codeword to generate the additional parity bits.

[498] In detail, when the repetition is not performed, LDPC parity bits included in a repeated LDPC codeword are the LDPC parity bits generated by the LDPC encoding.

[499] In this case, the additional parity generator 319 may first select all the punctured LDPC parity bits and additionally select bits as many as the number obtained by subtracting the number of punctured LDPC parity bits from the number of additional parity bits to be generated, from the first LDPC parity bit among the LDPC parity bits generated by the LDPC encoding, to generate the additional parity bits.

[500] Here, the LDPC parity bits generated by the LDPC encoding are divided into the non-punctured LDPC parity bits and the punctured LDPC parity bits. As a result, when bits are selected from the first bit among the LDPC parity bits generated by the LDPC encoding, they may be selected in an order of the non-punctured LDPC parity bits and the punctured LDPC parity bits.

[501] When the repetition is performed, the LDPC parity bits included in the repeated LDPC codeword are the repeated LDPC parity bits and the LDPC parity bits generated by the LDPC encoding. Here, the repeated LDPC parity bits are positioned between the LDPC information bits and the LDPC parity bits generated by the LDPC encoding.

[502] In this case, the additional parity generator 319 may first select all the punctured LDPC parity bits and additionally select the bits as many as the number obtained by subtracting the number of punctured LDPC parity bits from the number of additional bits , from the first LDPC parity bit among the repeated LDPC parity bits to generate the additional parity bits.

[503] Here, when the bits are selected from the first bit among the repeated LDPC parity

bits, they may be selected in an order of the repetition bits and the LDPC parity bits generated by the LDPC encoding. Further, the bits may be selected in an order of the non-punctured LDPC parity bits and the punctured LDPC parity bits, within the LDPC parity bits generated by the LDPC encoding.

[504] Hereinafter, methods for generating additional parity bits according to exemplary embodiments will be described in more detail with reference to FIGs. 17 to 19.

[505] FIGs. 17 to 19 are diagrams for describing the methods for generating additional parity bits when repetition is performed, according to the exemplary embodiments. In this case, a repeated LDPC codeword $V = (v_0, v_1, \ldots, v_{N_{inner}+N_{repeat}-1})$ may be represented as illustrated in FIG. 17.

[506] First, when $N_{AP} \leq N_{punc}$, as illustrated in FIG. 18, the additional parity generator 319 may select N_{AP} bits from the first LDPC parity bit among punctured LDPC parity bits to generate the additional parity bits.

[507] Therefore, for the additional parity bits, the punctured LDPC parity bits ($v_{N_{repeat}+N_{inner}-N_{punc}}, v_{N_{repeat}+N_{inner}-N_{punc}+1}, \ldots, v_{N_{repeat}+N_{inner}-N_{punc}+N_{AP}-1}$) may be selected. That is, the additional parity generator 319 may select the N_{AP} bits from the first bit among the punctured LDPC parity bits to generate the additional parity bits.

[508] Meanwhile, when $N_{AP} > N_{punc}$, as illustrated in FIG. 19, the additional parity generator 319 selects all the punctured LDPC parity bits.

[509] Therefore, for the additional parity bits, all the punctured LDPC parity bits ($v_{N_{repeat}+N_{inner}-N_{punc}}, v_{N_{repeat}+N_{inner}-N_{punc}+1}, \ldots, v_{N_{repeat}+N_{inner}-1}$) may be selected.

[510] Further, the additional parity generator 319 may additionally select first N_{AP}-N_{punc} bits from the LDPC parity bits including the repeated LDPC parity bits and the LDPC parity bits generated by the LDPC encoding.

[511] That is, since the repeated LDPC parity bits and the LDPC parity bits generated by the LDPC encoding are sequentially arranged, the additional parity generator 319 may additionally select the N_{AP}-N_{punc} parity bits from the first LDPC parity bit among the repeated LDPC parity bits.

[512] Therefore, for the additional parity bits, the LDPC parity bits ($v_{K_{ldpc}}, v_{K_{ldpc}+1}, \ldots, v_{K_{ldpc}+N_{AP}-N_{punc}-1}$) may be additionally selected.

[513] In this case, the additional parity generator 319 may add the additionally selected bits to the previously selected bits to generate the additional parity bits. That is, as illustrated in FIG. 19, the additional parity generator 319 may add the additionally selected LDPC parity bits to the punctured LDPC parity bits to generate the additional parity bits.

[514] As a result, for the additional parity bits, ($v_{N_{repeat}+N_{inner}-N_{punc}}$, $v_{N_{repeat}+N_{inner}-N_{punc}+1}$, ..., $v_{N_{repeat}+N_{inner}-1}$, $v_{K_{ldpc}}$, $v_{K_{ldpc}+1}$, ..., $v_{K_{ldpc}+N_{AP}-N_{punc}-1}$) may be selected.

[515] As such, when the number of punctured bits is equal to or greater than the number of additional parity bits, the additional parity bits may be generated by selecting bits among the punctured bits based on the puncturing order. On the other hand, in other cases, the additional parity bits may be generated by selecting all the punctured bits and the N_{AP}-N_{punc} parity bits.

[516] Meanwhile, since N_{repeat}=0 when repetition is not performed, the method for generating additional parity bits when the repetition is not performed is the same as the case in which N_{repeat}=0 in FIGs. 17 to 19.

[517] The additional parity bits may be bit-interleaved, and may be mapped to constellation. In this case, the constellation for the additional parity bits may be generated by the same method as constellation for the L1-detail signaling bits transmitted in the current frame, in which the L1-detail signaling bits are repeated, punctured, and have the zero bits removed. Further, as illustrated in FIG. 18, after being mapped to the constellation, the additional parity bits may be added after the L1-detail signaling block in a frame before the current frame in which the L1-detail signaling of the current frame is transmitted.

[518] The additional parity generator 319 may output the additional parity bits to a bit demultiplexer 323.

[519] As described above in reference to Tables 10 and 11, the group-wise interleaving pattern defining the permutation order may have two patterns: a first pattern and a second pattern.

[520] In detail, since the B value of above Equation 22 represents the minimum length of the LDPC parity bits to be punctured, the predetermined number of bits may be always punctured depending on the B value regardless of the length of the input signaling. For example, in the L1-detail mode 2, since B=6036 and the bit group is formed of 360 bits, even when the shortening length is 0, at least $\left\lfloor \frac{6036}{360} \right\rfloor = 16$ bit groups are always punctured.

[521] In this case, since the puncturing is performed from the last LDPC parity bit, the predetermined number of bit groups from a last bit group among the plurality of bit groups configuring the group-wise interleaved LDPC parity bits may be always punctured regardless of the shortening length.

[522] For example, in the L1-detail mode 2, the last 16 bit groups among 36 bit groups configuring the group-wise interleaved LDPC parity bits may be always punctured.

[523] As a result, some of the group-wise interleaving patterns defining the permutation order represent bit groups always to punctured, and therefore, the group-wise interleaving pattern may be divided into two patterns. In detail, a pattern defining the remaining bit groups other than the bit groups to be always punctured in the group-wise interleaving pattern is referred to as the first pattern, and the pattern defining the bit groups to be always punctured is referred to as the second pattern.

[524] For example, in the L1-detail mode 2, since the group-wise interleaving pattern is defined as above Table 10, a pattern representing indexes of bit groups which are not group-wise interleaved and positioned in a 9-th bit group to a 28-th bit group after group-wise interleaving, that is, $Y_9=X_{\pi p(9)}=X_9$, $Y_{10}=X_{\pi p(10)}=X_{31}$, $Y_{11}=X_{\pi p(11)}=X_{23}$, ..., $Y_{26}=X_{\pi p(26)}=X_{17}$, $Y_{27}=X_{\pi p(27)}=X_{35}$, $Y_{28}=X_{\pi p(28)}=X_{21}$ may be the first pattern, and a pattern representing indexes of bit groups which are not group-wise interleaved and positioned in a 29-th bit group to a 44-th bit group after group-wise interleaving, that is, $Y_{29}=X_{\pi p(29)}=X_{20}$, $Y_{30}=X_{\pi p(30)}=X_{24}$, $Y_{31}=X_{\pi p(31)}=X_{44}$, ..., $Y_{42}=X_{\pi p(42)}=X_{28}$, $Y_{43}=X_{\pi p(43)}=X_{39}$, $Y_{44}=X_{\pi p(44)}=X_{42}$ may be the second pattern.

[525] As described above, the second pattern defines bit groups to be always punctured in a current frame regardless of the shortening length, and the first pattern defines bit groups additionally to be punctured as the shortening length is long, such that the first pattern may be used to determine the LDPC parity bits to be transmitted in the current frame after the puncturing.

[526] In detail, according to the number of LDPC parity bits to be punctured, in addition to the LDPC parity bits to be always punctured, more LDPC parity bits may additionally be punctured.

[527] For example, in the L1-detail mode 2, when the number of LDPC parity bits to be punctured is 7200, 20 bit groups need to be punctured, and thus, four (4) bit groups need to be additionally punctured, in addition to the 16 bit groups to be always punctured.

[528] In this case, the additionally punctured four (4) bit groups correspond to the bit groups positioned at 25-th to 28-th positions after group-wise interleaving, and since these bit groups are determined according to the first pattern, that is, belong to the first pattern, the first pattern may be used to determine the punctured bit groups.

[529] That is, when LDPC parity bits are punctured more than a minimum value of LDPC parity bits to be punctured, which bit groups are to be additionally punctured is determined according to which bit groups are positioned after the bit groups to be always punctured. As a result, according to a puncturing direction, the first pattern which defines the bit groups positioned after the bit groups to be always punctured may be considered as determining the punctured bit groups.

[530] That is, as in the foregoing example, when the number of LDPC parity bits to be

punctured is 7200, in addition to the 16 bit groups to be always punctured, four (4) bit groups, that is, the bit groups positioned at 28-th, 27-th, 26-th, and 25-th positions, after group-wise interleaving is performed, are additionally punctured. Here, the bit groups positioned at 25-th to 28-th positions after the group-wise interleaving are determined according to the first pattern.

[531] As a result, the first pattern may be considered as being used to determine the bit groups to be punctured. Further, the remaining LDPC parity bits other than the punctured LDPC parity bits are transmitted through the current frame, and therefore, the first pattern may be considered as being used to determine the bit groups transmitted in the current frame.

[532] The second pattern may be used to determine the additional parity bits to be transmitted in the previous frame.

[533] In detail, since the bit groups determined to be always punctured are always punctured, and then, are not transmitted in the current frame, these bit groups need to be positioned only where bits are always punctured after group-wise interleaving. Therefore, it is not important at which position of these bit groups are positioned after the group-wise interleaving.

[534] For example, in the L1-detail mode 2, bit groups positioned at 20-th, 24-th, 44-th, …, 28-th, 39-th and 42-th positions before the group-wise interleaving need to be positioned only at a 29-th bit group to a 44-th bit group after the group-wise interleaving. Therefore, it is not important at which positions of these bit groups are positioned.

[535] As such, the second pattern defining bit groups to be always punctured is used to identify bit groups to be punctured. Therefore, defining an order between the bit groups in the second pattern is meaningless in the puncturing, and thus, the second pattern defining bit groups to be always punctured may be considered as not being used for the puncturing.

[536] However, for determining additional parity bits, positions of the bit groups to be always punctured within these bit groups need to be considered.

[537] In detail, since the additional parity bits are generated by selecting bits as many as a predetermined number from the first bit among the punctured LDPC parity bits, bits included in at least some of the bit groups to be always punctured may be selected as at least some of the additional parity bits depending on the number of punctured LDPC parity bits and the number of additional parity bits to be generated.

[538] That is, when additional parity bits are selected over the number of bit groups defined according to the first pattern, since the additional parity bits are sequentially selected from a start portion of the second pattern, the order of the bit groups belonging to the second pattern is meaningful in terms of selection of the additional parity bits. As a result, the second pattern defining bit groups to be always punctured may be

considered as being used to determine the additional parity bits.

[539] For example, in the L1-detail mode 2, the total number of LDPC parity bits is 12960 and the number of bit groups to be always punctured is 16.

[540] In this case, the second pattern may be used to generate the additional parity bits depending on whether a value obtained by subtracting the number of LDPC parity bits to be punctured from the number of all LDPC parity bits and adding the subtraction result to the number of additional parity bits to be generated exceeds 7200. Here, 7200 is the number of LDPC parity bits except the bit groups to be always punctured, among the bit groups configuring the LDPC parity bits. That is, 7200=(36-16)×360.

[541] In detail, when the value obtained by the above subtraction and addition is equal to or less than 7200, that is, $12960-N_{punc}+N_{AP} \leq 7200$, the additional parity bits may be generated according to the first pattern.

[542] However, when the value obtained by the above subtraction and addition exceeds 7200, that is, $12960-N_{punc}+N_{AP} > 7200$, the additional parity bits may be generated according to the first pattern and the second pattern.

[543] In detail, when $12960-N_{punc}+N_{AP} > 7200$, for the additional parity bits, bits included in the bit group positioned at a 28-th position from the first LDPC parity bit among the punctured LDPC parity bits may be selected, and bits included in the bit group positioned at a predetermined position from a 29-th position may be selected.

[544] Here, the bit group to which the first LDPC parity bit among the punctured LDPC parity bits belongs and the bit group (that is, when being sequentially selected from the first LDPC parity bit among the punctured LDPC parity bits, a bit group to which the finally selected LDPC parity bits belong) at the predetermined position may be determined depending on the number of punctured LDPC parity bits and the number of additional parity bits to be generated.

[545] In this case, the bit group positioned at the 28-th position from the firth LDPC parity bit among the punctured LDPC parity bits is determined according to the first pattern, and the bit group positioned at the predetermined position from the 29-th position is determined according to the second pattern.

[546] As a result, the additional parity bits are determined according to the first pattern and the second pattern.

[547] As such, the first pattern may be used to determine additional parity bits to be generated as well as LDPC parity bits to be punctured, and the second pattern may be used to determine the additional parity bits to be generated and LDPC parity bits to be always punctured regardless of the number of parity bits to be punctured by the puncturers 217 and 318.

[548] The foregoing example describes that the group-wise interleaving pattern includes the first pattern and the second pattern, which is only for convenience of explanation in

terms of the puncturing and the additional parity. That is, the group-wise interleaving pattern may be considered as one pattern without being divided into the first pattern and the second pattern. In this case, the group-wise interleaving may be considered as being performed with one pattern both for the puncturing and the additional parity.

[549] The values used in the foregoing example such as the number of punctured LDPC parity bits are only example values.

[550] The zero removers 218 and 321 may remove zero bits padded by the zero padders 213 and 314 from the LDPC codewords output from the puncturers 217 and 318, and output the remaining bits to the bit demultiplexers 219 and 322.

[551] Here, the removal does not only remove the padded zero bits but also may include outputting the remaining bits other than the padded zero bits in the LDPC codewords.

[552] In detail, the zero removers 218 and 321 may remove K_{ldpc}-N_{outer} zero bits padded by the zero padders 213 and 314. Therefore, the K_{ldpc}-N_{outer} padded zero bits are removed, and thus, may not be transmitted to the receiver 200.

[553] For example, as illustrated in FIG. 20, it is assumed that all bits of a first bit group, a fourth bit group, a fifth bit group, a seventh bit group, and an eighth bit group among a plurality of bit groups configuring an LDPC codeword are padded by zero bits, and some bits of the second bit group are padded by zero bits.

[554] In this case, the zero removers 218 and 321 may remove the zero bits padded to the first bit group, the second bit group, the fourth bit group, the fifth bit group, the seventh bit group, and the eighth bit group.

[555] As such, when zero bits are removed, as illustrated in FIG. 20, an LDPC codeword formed of K_{sig} information bits (that is, K_{sig} L1-basic signaling bits and K_{sig} L1-detail signaling bits), 168 BCH parity check bits (that is, BCH FEC), and N_{inner}-K_{ldpc}-N_{punc} or N_{inner}-K_{ldpc}-N_{punc}+N_{repeat} parity bits may remain.

[556] That is, when repetition is performed, the lengths of all the LDPC codewords become N_{FEC}+N_{repeat}. Here, $N_{FEC} = N_{outer}$+N_{ldpc_parity}-N_{punc}. However, in a mode in which the repetition is not performed, the lengths of all the LDPC codewords become N_{FEC}.

[557] The bit demultiplexers 219 and 322 may interleave the bits output from the zero removers 218 and 321, demultiplex the interleaved bits, and then output them to the constellation mappers 221 and 324.

[558] For this purpose, the bit demultiplexers 219 and 322 may include a block interleaver (not illustrated) and a demultiplexer (not illustrated).

[559] First, a block interleaving scheme performed in the block interleaver is illustrated in FIG. 21.

[560] In detail, the bits of the N_{FEC} or N_{FEC}+N_{repeat} length after the zero bits are removed may be column-wisely serially written in the block interleaver. Here, the number of columns of the block interleaver is equivalent to the modulation order and the number

of rows is N_{FEC}/η_{MOD} or $(N_{FEC}+N_{repeat})/\eta_{MOD}$.

[561] Further, in a read operation, bits for one constellation symbol may be sequentially read in a row direction to be input to the demultiplexer. The operation may be continued to the last row of the column.

[562] That is, the N_{FEC} or $(N_{FEC}+N_{repeat})$ bits may be written in a plurality of columns in a column direction from the first row of the first column, and the bits written in the plurality of columns are sequentially read from the first row to the last row of the plurality of columns in a row direction. In this case, the bits read in the same row may configure one modulation symbol.

[563] The demultiplexer may demultiplex the bits output from the block interleaver.

[564] In detail, the demultiplexer may demultiplex each of the block-interleaved bit groups, that is, the bits output while being read in the same row of the block interleaver within the bit group bit-by-bit, before the bits are mapped to constellation.

[565] In this case, two mapping rules may be present according to the modulation order.

[566] In detail, when QPSK is used for modulation, since reliability of bits within a constellation symbol is the same, the demultiplexer does not perform the demultiplexing operation on a bit group. Therefore, the bit group read and output from the block interleaver may be mapped to a QPSK symbol without the demultiplexing operation.

[567] However, when high order modulation is used, the demultiplexer may perform demultiplexing on a bit group read and output from the block interleaver based on following Equation 31. That is, a bit group may be mapped to a QAM symbol depending on following Equation 31.

[568]

$$S_{demux_in(i)} = \{b_i(0), b_i(1), b_i(2), \ldots, b_i(\eta_{MOD}-1)\},$$
$$S_{demux_out(i)} = \{c_i(0), c_i(1), c_i(2), \ldots, c_i(\eta_{MOD}-1)\},$$
$$c_i(0) = b_i(i\%\eta_{MOD}), c_i(1) = b_i((i+1)\%\eta_{MOD}), \ldots, c_i(\eta_{MOD}-1) = b_i((i+\eta_{MOD}-1)\%\eta_{MOD}) \quad \ldots. \quad (31)$$

[569] In above Equation 31, % represents a modulo operation, and η_{MOD} is a modulation order.

[570] Further, i is a bit group index corresponding to a row index of the block interleaver. That is, an output bit group $S_{demux_out(i)}$ mapped to each of the QAM symbols may be cyclic-shifted in an $S_{demux_in(i)}$ according to the bit group index i.

[571] FIG. 22 illustrates an example of performing bit demultiplexing on 16-non uniform constellation (16-NUC), that is, NUC 16-QAM. The operation may be continued until all bit groups are read in the block interleaver.

[572] The bit demultiplexer 323 may perform the same operation as the operations performed by the bit demultiplexers 219 and 322, on the additional parity bits output

from the additional parity generator 319, and output the block-interleaved and demultiplexed bits to the constellation mapper 325.

[573] The constellation mappers 221, 324 and 325 may map the bits output from the bit demultiplexers 219, 322 and 323 to constellation symbols, respectively.

[574] That is, each of the constellation mappers 221, 324 and 325 may map the $S_{demux_out(i)}$ to a cell word using constellation according to a corresponding mode. Here, the $S_{demux_out(i)}$ may be configured of bits having the same number as the modulation order.

[575] In detail, the constellation mappers 221, 324 and 325 may map bits output from the bit demultiplexers 219, 322 and 323 to constellation symbols using QPSK, 16-QAM, 64-QAM, the 256-QAM, etc., according to a corresponding mode.

[576] In this case, the constellation mappers 221, 324 and 325 may use the NUC. That is, the constellation mappers 221, 324 and 325 may use NUC 16-QAM, NUC 64-QAM or NUC 256-QAM. The modulation scheme applied to the L1-basic signaling and the L1-detail signaling according to a corresponding mode is shown in above Table 5.

[577] The transmitter 100 may map the constellation symbols to a frame and transmit the mapped symbols to the receiver 200.

[578] In detail, the transmitter 100 may map the constellation symbols corresponding to each of the L1-basic signaling and the L1-detail signaling output from the constellation mappers 221 and 324, and map the constellation symbols corresponding to the additional parity bits output from the constellation mapper 325 to a preamble symbol of a frame.

[579] In this case, the transmitter 100 may map the additional parity bits generated based on the L1-detail signaling transmitted in the current frame to a frame before the current frame.

[580] That is, the transmitter 100 may map the LDPC codeword bits including the L1-basic signaling corresponding to the (i-1)-th frame to the (i-1)-th frame, maps the LDPC codeword bits including the L1-detail signaling corresponding to the (i-1)-th frame to the (i-1)-th frame, and additionally map the additional parity bits generated selected from the LDPC parity bits generated based on the L1-detail signaling corresponding to the i-th frame to the (i-1)-th frame and may transmit the mapped bits to the receiver 200.

[581] In addition, the transmitter 100 may map data to the data symbols of the frame in addition to the L1 signaling and transmit the frame including the L1 signaling and the data to the receiver 200.

[582] In this case, since the L1 signalings include signaling information about the data, the signaling about the data mapped to each data may be mapped to a preamble of a corresponding frame. For example, the transmitter 100 may map the L1 signaling including the signaling information about the data mapped to the i-th frame to the i-th frame.

[583] As a result, the receiver 200 may use the signaling obtained from the frame to receive the data from the corresponding frame for processing.

[584] FIGs. 23 and 24 are block diagrams for describing a configuration of a receiver according to an exemplary embodiment.

[585] In detail, as illustrated in FIG. 23, the receiver 200 may include a constellation demapper 2210, a multiplexer 2220, a Log Likelihood Ratio (LLR) 2230, an LLR combiner 2240, a parity depermutator 2250, an LDPC decoder 2260, a zero remover 2270, a BCH decoder 2280, and a descrambler 2290 to process the L1-basic signaling.

[586] Further, as illustrated in FIG. 24, the receiver 200 may include constellation demappers 2311 and 2312, multiplexers 2321 and 2322, an LLR inserter 2330, an LLR combiner 2340, a parity depermutator 2350, an LDPC decoder 2360, a zero remover 2370, a BCH decoder 2380, a descrambler 2390, and a desegmenter 2395 to process the L1-detail signaling.

[587] Here, the components illustrated in FIGs. 23 and 24 perform functions corresponding to the functions of the components illustrated in FIGs. 7 and 8, respectively, which is only an example, and in some cases, some of the components may be omitted and changed and other components may be added.

[588] The receiver 200 may acquire frame synchronization using a bootstrap of a frame and receive L1-basic signaling from a preamble of the frame using information for processing the L1-basic signaling included in the bootstrap.

[589] Further, the receiver 200 may receive L1-detail signaling from the preamble using information for processing the L1-detail signaling included in the L1-basic signaling, and receive broadcasting data required by a user from data symbols of the frame using the L1-detail signaling.

[590] Therefore, the receiver 200 may determine a mode of used at the transmitter 100 to process the L1-basic signaling and the L1-detail signaling, and process a signal received from the transmitter 100 according to the determined mode to receive the L1-basic signaling and the L1-detail signaling. For this purpose, the receiver 200 may pre-store information about parameters used at the transmitter 100 to process the signaling according to corresponding modes.

[591] As such, the L1-basic signaling and the L1-detail signaling may be sequentially acquired from the preamble. In describing FIGs. 23 and 24, components performing common functions will be described together for convenience of explanation.

[592] The constellation demappers 2210, 2311 and 2312 demodulate a signal received from the transmitter 100.

[593] In detail, the constellation demapppers 2210, 2311 and 2312 are components corresponding to the constellation mappers 221, 324 and 325 of the transmitter 100, respectively, and may demodulate the signal received from the transmitter 100 and

generate values corresponding to bits transmitted from the transmitter 100.

[594] That is, as described above, the transmitter 100 maps an LDPC codeword including the L1-basic signaling and the LDPC codeword including the L1-detail signaling to the preamble of a frame, and transmits the mapped LDPC codeword to the receiver 200. Further, in some cases, the transmitter 100 may map additional parity bits to the preamble of a frame and transmit the mapped bits to the receiver 200.

[595] As a result, the constellation demappers 2210 and 2311 may generate values corresponding to the LDPC codeword bits including the L1-basic signaling and the LDPC codeword bits including the L1-detail signaling. Further, the constellation demapper 2312 may generate values corresponding to the additional parity bits.

[596] For this purpose, the receiver 200 may pre-store information about a modulation scheme used by the transmitter 100 to modulate the L1-basic signaling, the L1-detail signaling, and the additional parity bits according to corresponding modes. Therefore, the constellation demappers 2210, 2311 and 2312 may demodulate the signal received from the transmitter 100 according to the corresponding modes to generate values corresponding to the LDPC codeword bits and the additional parity bits.

[597] The value corresponding to a bit transmitted from the transmitter 100 is a value calculated based on probability that a received bit is 0 and 1, and instead, the probability itself may also be used as a value corresponding to each bit. The value may also be a Likelihood Ratio (LR) or an LLR value as another example.

[598] In detail, an LR value may represent a ratio of probability that a bit transmitted from the transmitter 100 is 0 and probability that the bit is 1, and an LLR value may represent a value obtained by taking a log on probability that the bit transmitted from the transmitter 100 is 0 and probability that the bit is 1.

[599] The foregoing example uses the LR value or the LLR value, which is only one example. According to another exemplary embodiment, the received signal itself rather than the LR or LLR value may also be used.

[600] The multiplexers 2220, 2321 and 2322 perform multiplexing on the LLR values output from the constellation demappers 2210, 2311 and 2312.

[601] In detail, the multiplexers 2220, 2321 and 2322 are components corresponding to the bit demultiplexers 219, 322 and 323 of the transmitter 100 and may perform operations corresponding to the operations of the bit demultiplexers 219, 322 and 323, respectively.

[602] For this purpose, the receiver 200 may pre-store information about parameters used for the transmitter 100 to perform demultiplexing and block interleaving. Therefore, the multiplexers 2220, 2321 and 2322 may reversely perform the demultiplexing and block interleaving operations of the bit demultiplexers 219, 322, and 323 on the LLR value corresponding to a cell word to multiplex the LLR value corresponding to the

cell word in a bit unit.

[603] The LLR inserters 2230 and 2330 may insert LLR values for the puncturing and shortening bits into the LLR values output from the multiplexers 2220 and 2321, respectively. In this case, the LLR inserters 2230 and 2330 may insert previously determined LLR values between the LLR values output from the multiplexers 2220 and 2321 or a head portion or an end portion thereof.

[604] In detail, the LLR inserters 2230 and 2330 are components corresponding to the zero removers 218 and 321 and the puncturers 217 and 318 of the transmitter 100, respectively, and may perform operations corresponding to the operations of the zero removers 218 and 321 and the puncturers 217 and 318, respectively.

[605] First, the LLR inserters 2230 and 2330 may insert LLR values corresponding to zero bits into a position where the zero bits in the LDPC codeword are padded. In this case, the LLR values corresponding to the padded zero bits, that is, the shortened zero bits may be ∞ or $-\infty$. However, ∞ or $-\infty$ are a theoretical value but may actually be a maximum value or a minimum value of the LLR value used in the receiver 200.

[606] For this purpose, the receiver 200 may pre-store information about parameters and/or patterns used for the transmitter 100 to pad the zero bits according to corresponding modes. Therefore, the LLR inserters 2230 and 2330 may determine positions where the zero bits in the LDPC codeword are padded according to the corresponding the modes, and insert the LLR values corresponding to the shortened zero bits into corresponding positions.

[607] Further, the LLR inserters 2230 and 2330 may insert the LLR values corresponding to the punctured bits into the positions of the punctured bits in the LDPC codeword. In this case, the LLR values corresponding to the punctured bits may be 0.

[608] For this purpose, the receiver 200 may pre-store information about parameters and/or patterns used for the transmitter 100 to perform puncturing according to corresponding modes. Therefore, the LLR inserters 2230 and 2330 may determine the lengths of the punctured LDPC parity bits according to the corresponding modes, and insert corresponding LLR values into the positions where the LDPC parity bits are punctured.

[609] When the additional parity bits selected from the punctured bits among the additional parity bits, the LLR inserter 2630 may insert LLR values corresponding to the received additional parity bits, not an LLR value '0' for the punctured bit, into the positions of the punctured bits.

[610] The LLR combiners 2240 and 2340 may combine, that is, a sum the LLR values output from the LLR inserters 2230 and 2330 and the LLR value output from the multiplexer 2322. However, the LLR combiners 2540 and 2640 serve to update LLR values for specific bits into more correct values. However, the LLR values for the specific bits may also be decoded from the received LLR values without the LLR

combiners 2540 and 2640, and therefore, in some cases, the LLR combiners 2540 and 2640 may be omitted.

[611] In detail, the LLR combiner 2240 is a component corresponding to the repeater 216 of the transmitter 100, and may perform an operation corresponding to the operation of the repeater 216. Alternatively, the LLR combiner 2340 is a component corresponding to the repeater 317 and the additional parity generator 319 of the transmitter 100 and may perform operations corresponding to the operations of the repeater 317 and the additional parity generator 319.

[612] First, the LLR combiners 2240 and 2340 may combine LLR values corresponding to the repetition bits with other LLR values. Here, the other LLR values may be bits which are a basis of generating the repetition bits by the transmitter 100, that is, LLR values for the LDPC parity bits selected as the repeated object.

[613] That is, as described above, the transmitter 100 selects bits from the LDPC parity bits and repeats the selected bits between the LDPC information bits and the LDPC parity bits generated by LDPC encoding, and transmits the repetition bits to the receiver 200.

[614] As a result, the LLR values for the LDPC parity bits may be formed of the LLR values for the repeated LDPC parity bits and the LLR values for the non-repeated LDPC parity bits, that is, the LDPC parity bits generated by the LDPC encoding. Therefore, the LLR combiners 2240 and 2340 may combine the LLR values for the same LDPC parity bits.

[615] For this purpose, the receiver 200 may pre-store information about parameters used for the transmitter 100 to perform the repetition according to corresponding modes. As a result, the LLR combiners 2240 and 2340 may determine the lengths of the repeated LDPC parity bits, determine the positions of the bits which are a basis of the repetition, and combine the LLR values for the repeated LDPC parity bits with the LLR values for the LDPC parity bits which are a basis of the repetition and generated by the LDPC encoding.

[616] For example, as illustrated in FIGs. 25 and 26, the LLR combiners 2240 and 2340 may combine LLR values for repeated LDPC parity bits with LLR values for LDPC parity bits which are a basis of the repetition and generated by the LDPC encoding.

[617] When LPDC parity bits are repeated n times, the LLR combiners 2240 and 2340 may combine LLR values for bits at the same position at n times or less.

[618] For example, FIG. 25 illustrates a case in which some of LDPC parity bits other than punctured bits are repeated once. In this case, the LLR combiners 2240 and 2340 may combine LLR values for the repeated LDPC parity bits with LLR values for the LDPC parity bits generated by the LDPC encoding, and then, output the combined LLR values or output the LLR values for the received repeated LDPC parity bits or the LLR values for the received LDPC parity bits generated by the LDPC encoding without

combining them.

[619] As another example, FIG. 26 illustrates a case in which some of the transmitted LDPC parity bits, which are not punctured, are repeated twice, the remaining portions are repeated once, and the punctured LDPC parity bits are repeated once.

[620] In this case, the LLR combiners 2240 and 2340 may process the remaining portion and the punctured bits which are repeated once by the same scheme as described above. However, the LLR combiners 2240 and 2340 may process the portion repeated twice as follows. In this case, for convenience of description, one of the two portions generated by repeating some of the LDPC parity bits twice is referred to as a first portion and the other is referred to as the second portion.

[621] In detail, the LLR combiners 2240 and 2340 may combine LLR values for each of the first and second portions with LLR values for the LDPC parity bits. Alternatively, the LLR combiners 2240 and 2340 may combine the LLR values for the first portion with the LLR values for the LDPC parity bits, combine the LLR values for the second portion with the LLR values for the LDPC parity bits, or combine the LLR values for the first portion with the LLR values for the second portion. Alternatively, the LLR combiners 2240 and 2340 may output the LLR values for the first portion, the LLR values for the second portion, the LLR values for the remaining portion, and punctured bits, without separate combination.

[622] Further, the LLR combiner 2340 may combine LLR values corresponding to additional parity bits with other LLR values. Here, the other LLR values may be the LDPC parity bits which are a basis of the generation of the additional parity bits by the transmitter 100, that is, the LLR values for the LDPC parity bits selected for generation of the additional parity bits.

[623] That is, as described above, the transmitter 100 may map additional parity bits for L1-detail signaling transmitted in a current frame to a previous frame and transmit the mapped bits to the receiver 200.

[624] In this case, the additional parity bits may include LDPC parity bits which are punctured and are not transmitted in the current frame, and in some cases, may further include LDPC parity bits transmitted in the current frame.

[625] As a result, the LLR combiner 2340 may combine LLR values for the additional parity bits received through the current frame with LLR values inserted into the positions of the punctured LDPC parity bits in the LDPC codeword received through the next frame and LLR values for the LDPC parity bits received through the next frame.

[626] For this purpose, the receiver 200 may pre-store information about parameters and/or patterns used for the transmitter 100 to generate the additional parity bits according to corresponding modes. As a result, the LLR combiner 2340 may determine the lengths

of the additional parity bits, determine the positions of the LDPC parity bits which are a basis of generation of the additional parity bits, and combine the LLR values for the additional parity bits with the LLR values for the LDPC parity bits which are a basis of generation of the additional parity bits.

[627] The parity depermutators 2250 and 2350 may depermutate the LLR values output from the LLR combiners 2240 and 2340, respectively.

[628] In detail, the parity depermutators 2250 and 2350 are components corresponding to the parity permutators 215 and 316 of the transmitter 100, and may perform operations corresponding to the operations of the parity permutators 215 and 316, respectively.

[629] For this purpose, the receiver 200 may pre-store information about parameters and/or patterns used for the transmitter 100 to perform group-wise interleaving and parity interleaving according to corresponding modes. Therefore, the parity depermutators 2250 and 2350 may reversely perform the group-wise interleaving and parity interleaving operations of the parity permutators 215 and 316 on the LLR values corresponding to the LDPC codeword bits, that is, perform group-wise deinterleaving and parity deinterleaving operations to perform the parity depermutation on the LLR values corresponding to the LDPC codeword bits, respectively.

[630] The LDPC decoders 2260 and 2360 may perform LDPC decoding based on the LLR values output from the parity depermutators 2250 and 2350, respectively.

[631] In detail, the LDPC decoders 2260 and 2360 are components corresponding to the LDPC encoders 214 and 315 of the transmitter 100 and may perform operations corresponding to the operations of the LDPC encoders 214 and 315, respectively.

[632] For this purpose, the receiver 200 may pre-store information about parameters used for the transmitter 100 to perform the LDPC encoding according to corresponding modes. Therefore, the LDPC decoders 2260 and 2360 may perform the LDPC decoding based on the LLR values output from the parity depermutators 2250 and 2350 according to the corresponding modes.

[633] For example, the LDPC decoders 2260 and 2360 may perform the LDPC decoding based on the LLR values output from the parity depermutators 2250 and 2350 by iterative decoding based on a sum-product algorithm and output error-corrected bits depending on the LDPC decoding.

[634] The zero removers 2270 and 2370 may remove zero bits from the bits output from the LDPC decoders 2260 and 2360, respectively.

[635] In detail, the zero removers 2270 and 2370 are components corresponding to the zero padders 213 and 314 of the transmitter 100 and may perform operations corresponding to the operations of the zero padders 213 and 314, respectively.

[636] For this purpose, the receiver 200 may pre-store information about parameters and/or patterns used for the transmitter 100 to pad the zero bits according to corresponding

modes. As a result, the zero removers 2270 and 2370 may remove the zero bits padded by the zero padders 213 and 314 from the bits output from the LDPC decoders 2260 and 2360, respectively.

[637] The BCH decoders 2280 and 2380 may perform BCH decoding on the bits output from the zero removers 2270 and 2370, respectively.

[638] In detail, the BCH decoders 2280 and 2380 are components corresponding to the BCH encoders 212 and 313 of the transmitter 100 and may perform the operations cor-responding to the BCH encoders 212 and 313.

[639] For this purpose, the receiver 200 may pre-store the information about parameters used for the transmitter 100 to perform BCH encoding. As a result, the BCH decoders 2280 and 2380 may correct errors by performing the BCH decoding on the bits output from the zero removers 2270 and 2370 and output the error-corrected bits.

[640] The descramblers 2290 and 2390 may descramble the bits output from the BCH decoders 2280 and 2380, respectively.

[641] In detail, the descramblers 2290 and 2390 are components corresponding to the scramblers 211 and 312 of the transmitter 100 and may perform operations corre-sponding to the operations of the scramblers 211 and 312.

[642] For this purpose, the receiver 200 may pre-store information about the parameters used for the transmitter 100 to perform the scrambling. As a result, the descramblers 2290 and 2390 may descramble the bits output from the BCH decoders 2280 and 2380 and output them, respectively.

[643] As a result, L1-basic signaling transmitted from the transmitter 100 may be recovered. Further, when the transmitter 100 does not perform segmentation on L1-detail signaling, the L1-detail signaling transmitted from the transmitter 100 may also be recovered.

[644] However, when the transmitter 100 performs the segmentation on the L1-detail signaling, the desegmenter 2395 may desegment the bits output from the descrambler 2390.

[645] In detail, the desegmenter 2395 is a component corresponding to the segmenter 311 of the transmitter 100 and may perform an operation corresponding to the operation of the segmenter 311.

[646] For this purpose, the receiver 200 may pre-store information about parameters used for the transmitter 100 to perform the segmentation. As a result, the desegmenter 2395 may combine the bits output from the descrambler 2390, that is, the segments for the L1-detail signaling to recover the L1-detail signaling before the segmentation.

[647] The information about the length of the L1 signaling is provided as illustrated in FIG. 27. Therefore, the receiver 200 may calculate the length of the L1-detail signaling and the length of the additional parity bits.

[648] Referring to FIG. 27, since the L1-basic signaling provides information about L1-detail total cells, the receiver 200 needs to calculate the length of the L1-detail signaling and the lengths of the additional parity bits.

[649] In detail, when L1B_L1_Detail_additional_parity_mode of the L1-basic signaling is not 0, since the information on the given L1B_L1_Detail_total_cells represents a total cell length (= $N_{L1_detail_total_cells}$), the receiver 200 may calculate the length $N_{L1_detail_cells}$ of the L1-detail signaling and the length $N_{AP_total_cells}$ of the additional parity bits based on following Equations 32 to 34.

[650] $$N_{L1_FEC_cells}=(N_{outer}+N_{repeat}+N_{ldpc_parity}-N_{punc})/\eta_{MOD}=N_{FEC}/\eta_{MOD} \quad ....(32)$$

[651] $$N_{L1_detail_cells}=N_{L1D_FECFRAME}\times N_{L1_FEC_cells} \quad .... \ (33)$$

[652] $$N_{AP_total_cells}=N_{L1_detail_total_cells}-N_{L1_detail_cells} \quad .... \ (34)$$

[653] In this case, based on above Equations 32 to 34, an $N_{AP_total_cells}$ value may be obtained based on an $N_{L1_detail_total_cells}$ value which may be obtained from the information about the L1B_L1_Detail_total_cells of the L1-basic signaling, N_{FEC}, the $N_{L1D_FECFRAME}$, and the modulation order η_{MOD}. As an example, $N_{AP_total_cells}$ may be calculated based on following Equation 35.

[654] $$N_{AP_total_cells}=N_{L1_detail_total_cells}-N_{L1D_FECFRAME}\times N_{FEC}/\eta_{MOD} \quad .... \ (35)$$

[655] Meanwhile, a syntax, and field semantics of the L1-basic signaling field are as following Table 14.

[656] [Table 14]

[657]

Syntax	# of bits	Format
L1_Basic_signalling() {		
L1B_L1_Detail_size_bits	16	uimsbf
L1B_L1_Detail_fec_type	3	uimsbf
L1B_L1_Detail_additional_parity_mode	2	uimsbf
L1B_L1_Detail_total_cells	19	uimsbf
L1B_Reserved	?	uimsbf
L1B_crc	32	uimsbf
{		

[658] As a result, the receiver 200 may perform an operation of the receiver for the additional parity bits in the next frame based on the additional parity bits transmitted to the $N_{AP_total_cells}$ cell among the received L1 detail cells.

[659] FIG. 28 is a flow chart for describing a method for parity permutation according to an exemplary embodiment of the present disclosure.

[660] First, parity bits are generated by encoding input bits (S2510).

[661] Next, outer-encoded bits including the input bits and the parity bits, and LDPC information bits including the zero bits are configured (S2520).

[662] Further, the LDPC information bits are encoded (S2530).

[663] Meanwhile, in S2520, zero bits are padded to at least some of a plurality of bit groups configuring the LDPC information bits based on a shortening pattern as shown in above Table 1.

[664] In S2520, the number of bit groups N_{pad} in which all bits (or bit positions) are padded by zero bits may be calculated based on above Equation 3 or 4.

[665] In S2520, zero bits may be padded to all bits (or bit positions) of a $\pi_s(0)$-th bit group, a $\pi_s(1)$-th bit group, ..., a $\pi_s(N_{pad}-1)$-th bit group of the plurality of bit groups based on the shortening pattern, and zero bits may be additionally padded to $K_{ldpc}-N_{outer}-360\times N_{pad}$ bits (or bit positions) from a first bit (or bit position) of a $\pi_s(N_{pad})$-th bit group.

[666] A detailed method for performing shortening based on above Table 1 is described above, and thus, duplicate descriptions are omitted.

[667] A non-transitory computer readable medium in which a program performing the various methods described above are stored may be provided according to an exemplary embodiment. The non-transitory computer readable medium is not a medium that stores data therein for a while, such as a register, a cache, a memory, or the like, but means a medium that at least semi-permanently stores data therein and is readable by a device such as a microprocessor. In detail, various applications or programs described above may be stored and provided in the non-transitory computer readable medium such as a compact disk (CD), a digital versatile disk (DVD), a hard disk, a Blu-ray disk, a universal serial bus (USB), a memory card, a read only memory (ROM), or the like.

[668] At least one of the components, elements, modules or units represented by a block as illustrated in FIGs. 1, 9, 10, 25 and 26 may be embodied as various numbers of hardware, software and/or firmware structures that execute respective functions described above, according to an exemplary embodiment. For example, at least one of these components, elements, modules or units may use a direct circuit structure, such as a memory, a processor, a logic circuit, a look-up table, etc. that may execute the respective functions through controls of one or more microprocessors or other control apparatuses. Also, at least one of these components, elements, modules or units may be specifically embodied by a module, a program, or a part of code, which contains one or more executable instructions for performing specified logic functions, and executed by one or more microprocessors or other control apparatuses. Also, at least one of these components, elements, modules or units may further include or implemented by a processor such as a central processing unit (CPU) that performs the respective functions, a microprocessor, or the like. Two or more of these components, elements, modules or units may be combined into one single component, element, module or unit which performs all operations or functions of the combined two or more components,

elements, modules or units. Also, at least part of functions of at least one of these components, elements, modules or units may be performed by another of these components, elements, modules or units. Further, although a bus is not illustrated in the above block diagrams, communication between the components, elements, modules or units may be performed through the bus. Functional aspects of the above exemplary embodiments may be implemented in algorithms that execute on one or more processors. Furthermore, the components, elements, modules or units represented by a block or processing steps may employ any number of related art techniques for electronics configuration, signal processing and/or control, data processing and the like.

[669] Although the exemplary embodiments of inventive concept have been illustrated and described hereinabove, the inventive concept is not limited to the above-mentioned exemplary embodiments, but may be variously modified by those skilled in the art to which the inventive concept pertains without departing from the scope and spirit of the inventive concept as disclosed in the accompanying claims. For example, the exemplary embodiments are described in relation with BCH encoding and decoding and LDPC encoding and decoding. However, these embodiments do not limit the inventive concept to only a particular encoding and decoding, and instead, the inventive concept may be applied to different types of encoding and decoding with necessary modifications. These modifications should also be understood to fall within the scope of the inventive concept.

Industrial Applicability

[670] -

Sequence Listing Free Text

[671] -

Claims

[Claim 1] 1. A transmitter, including:
an outer encoder configured to encode input bits to generate parity bits;
a zero padder configured to constitute LDPC information bits including outer-encoded bits including the input bits and the parity bits, and zero bits; and
an LDPC encoder configured to encode the LDPC information bits,
wherein the zero padder pads the zero bits to at least some of bit groups constituting the LDPC information bits based on pattern described below:

N_{info_group}	$\pi_s(j)$ $(0 \le j < N_{info\ group})$								
	$\pi_s(0)$	$\pi_s(1)$	$\pi_s(2)$	$\pi_s(3)$	$\pi_s(4)$	$\pi_s(5)$	$\pi_s(6)$	$\pi_s(7)$	$\pi_s(8)$
	$\pi_s(9)$	$\pi_s(10)$	$\pi_s(11)$	$\pi_s(12)$	$\pi_s(13)$	$\pi_s(14)$	$\pi_s(15)$	$\pi_s(16)$	$\pi_s(17)$
18	0	12	15	13	2	5	7	9	8
	6	16	10	14	1	17	11	4	3

,

where $\pi_s(j)$ represents an order of a j-th bit group and N_{info_group} represents a number of the bit groups.

[Claim 2] The transmitter of claim 1, wherein the zero padder calculates a number N_{pad} of bit groups in which all bits are to be padded by zero bits based on a following equation:

$$N_{pad} = \left\lfloor \frac{K_{ldpc} - N_{outer}}{360} \right\rfloor ,$$

where K_{ldpc} represents a number of the LDPC information bits, N_{outer} represents a number of the outer-encoded bits.

[Claim 3] The transmitter of claim 2, wherein the zero padder pads zero bits to all bits of a $\pi_s(0)$-th bit group, a $\pi_s(1)$-th bit group, …, a $\pi_s(N_{pad}-1)$-th bit group among the bit groups based on the pattern.

[Claim 4] The transmitter of claim 3, wherein the zero padder additionally pads zero bits to $K_{ldpc}-N_{outer}-360\times N_{pad}$ bit positions from a first bit of a $\pi_s(N_{pad}$)-th bit group.

[Claim 5] A shortening method of a transmitter, including:
generating parity bits by outer-encoding input bits;
constituting an LDPC information bits including an outer-encoded bit including the input bits and the parity bits and, zero bits; and
encoding the LDPC information bits,
wherein the configuring the LDPC information bits comprises padding

the zero bits to at least some of bit groups constituting the LDPC information bits based on pattern described below:

N_{info_group}	$\pi_s(j)$ ($0 \le j < N_{info\ group}$)								
	$\pi_s(0)$	$\pi_s(1)$	$\pi_s(2)$	$\pi_s(3)$	$\pi_s(4)$	$\pi_s(5)$	$\pi_s(6)$	$\pi_s(7)$	$\pi_s(8)$
	$\pi_s(9)$	$\pi_s(10)$	$\pi_s(11)$	$\pi_s(12)$	$\pi_s(13)$	$\pi_s(14)$	$\pi_s(15)$	$\pi_s(16)$	$\pi_s(17)$
18	0	12	15	13	2	5	7	9	8
	6	16	10	14	1	17	11	4	3

,

where $\pi_s(j)$ represents an order of a j-th bit group and N_{info_group} represents a number of the bit groups.

[Claim 6] The method of claim 5, wherein in the constituting the LDPC information bits, a number N_{pad} of bit groups in which all bits are to be padded by zero bits based on a following equation:

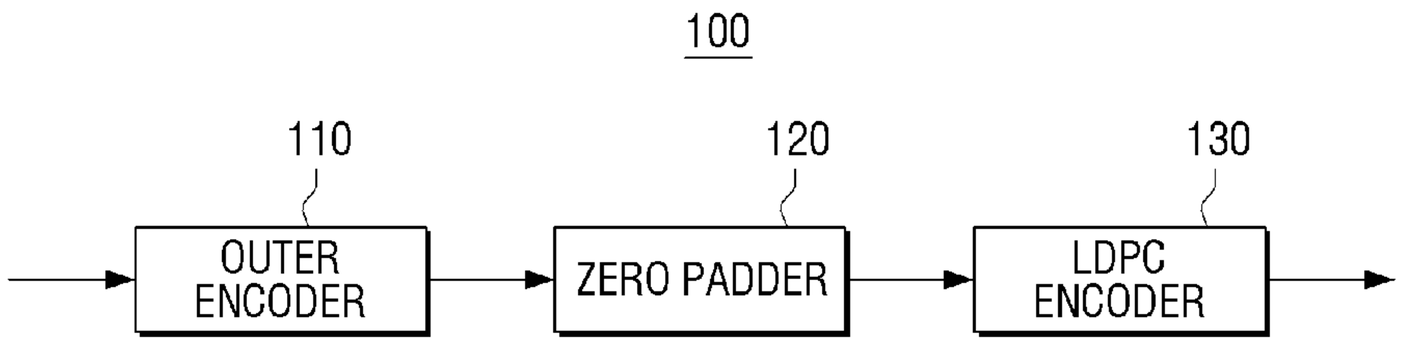
$$N_{pad} = \left\lfloor \frac{K_{ldpc} - N_{outer}}{360} \right\rfloor ,$$

where K_{ldpc} represents a number of the LDPC information bits, N_{outer} represents a number of the outer-encoded bits.

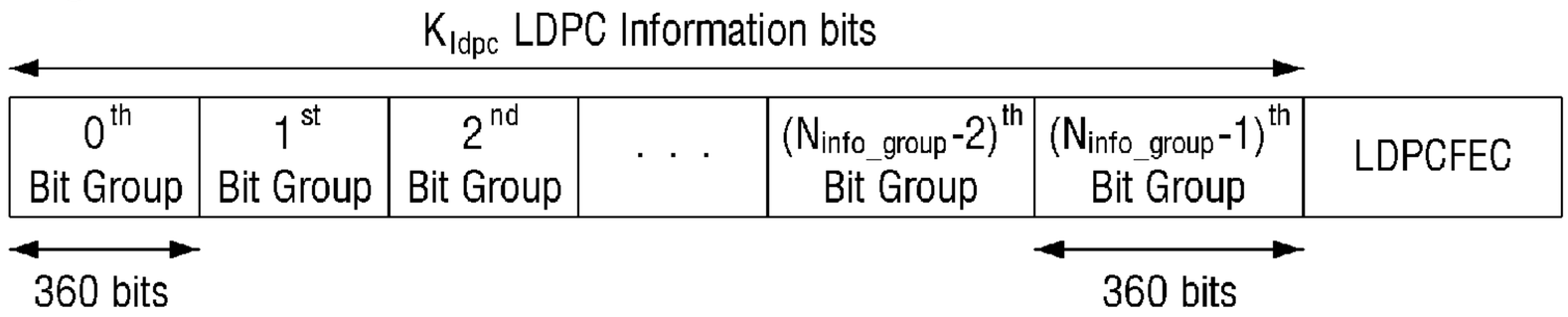
[Claim 7] The method of claim 6, wherein in the constituting the LDPC information bits, the zero bits are padded to all bit positions of a $\pi_s(0)$-th bit group, a $\pi_s(1)$-th bit group, …, a $\pi_s(N_{pad}-1)$-th bit group among the bit groups based on the pattern.

[Claim 8] The method of claim 7, wherein in the constituting the LDPC information bits, zero bits are additionally padded to $K_{ldpc}-N_{outer}-360\times N_{pad}$ bits from a first bit of a $\pi_s(N_{pad})$-th bit group.

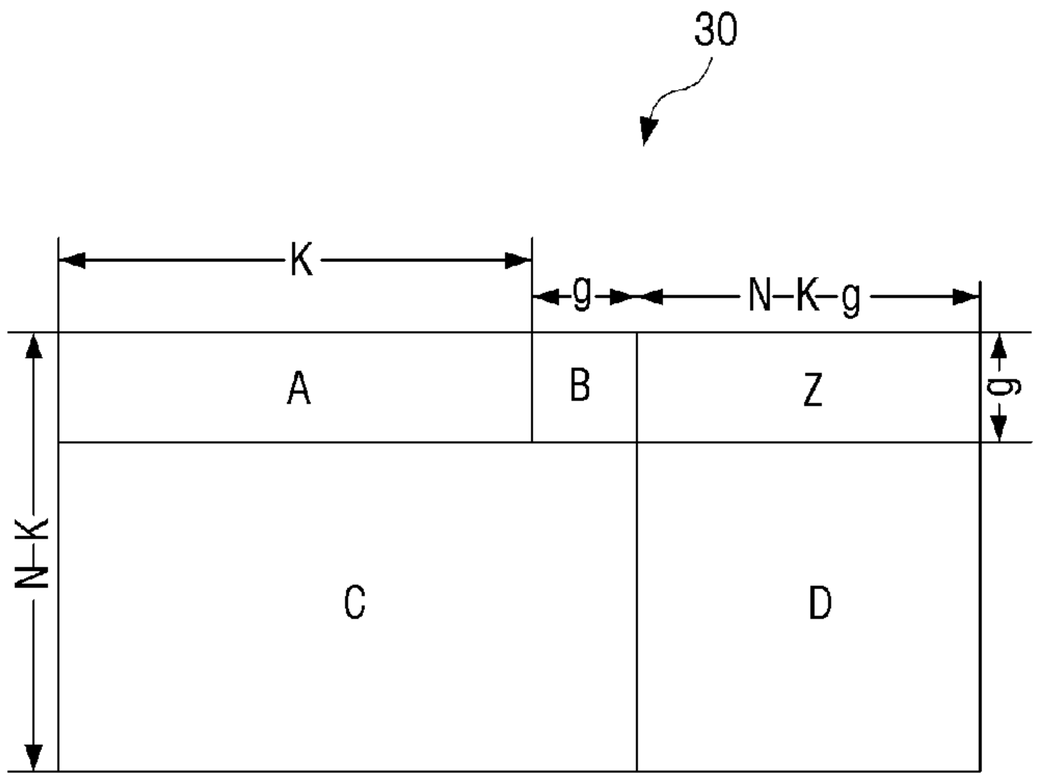
[Fig. 1]



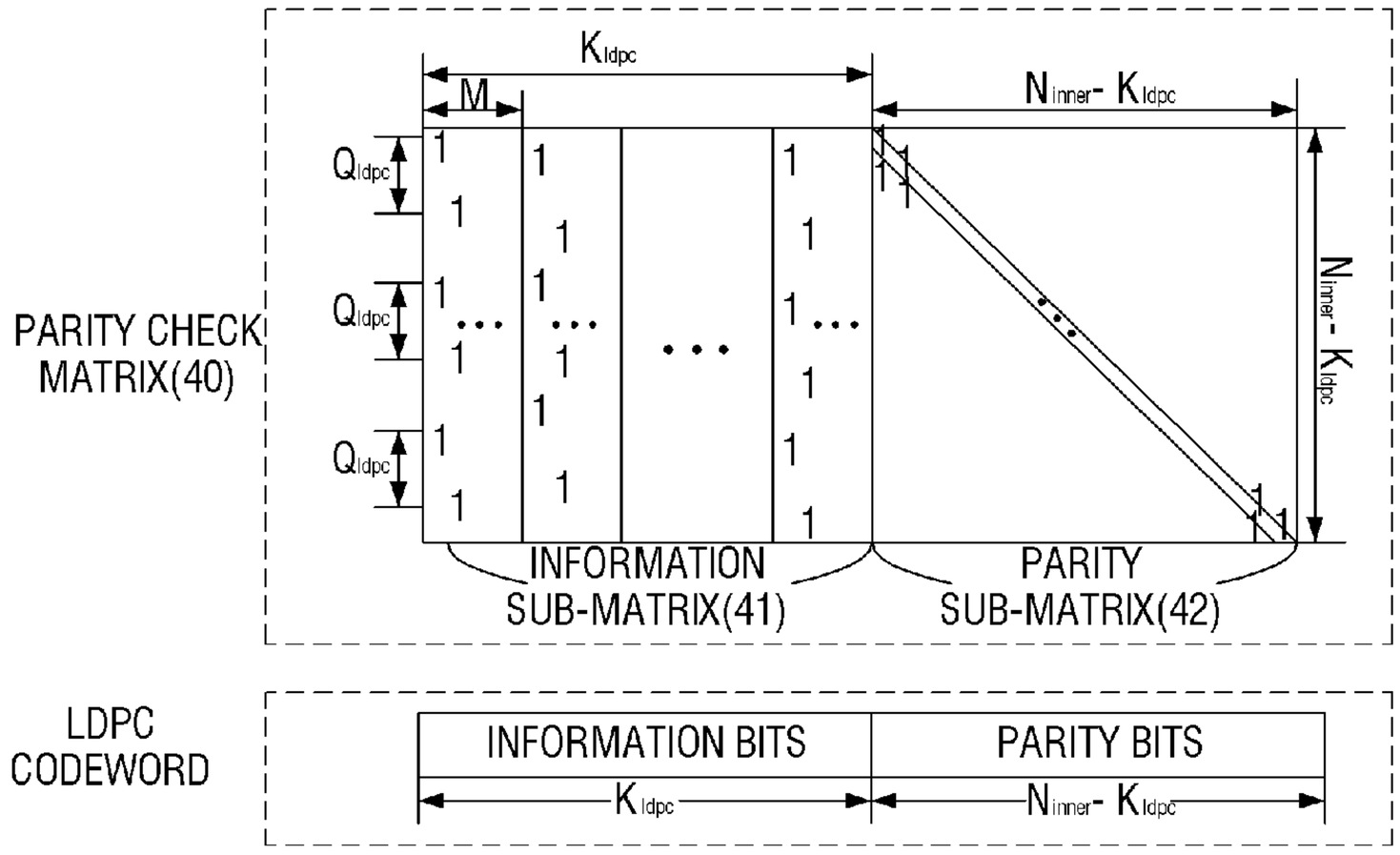
[Fig. 2]



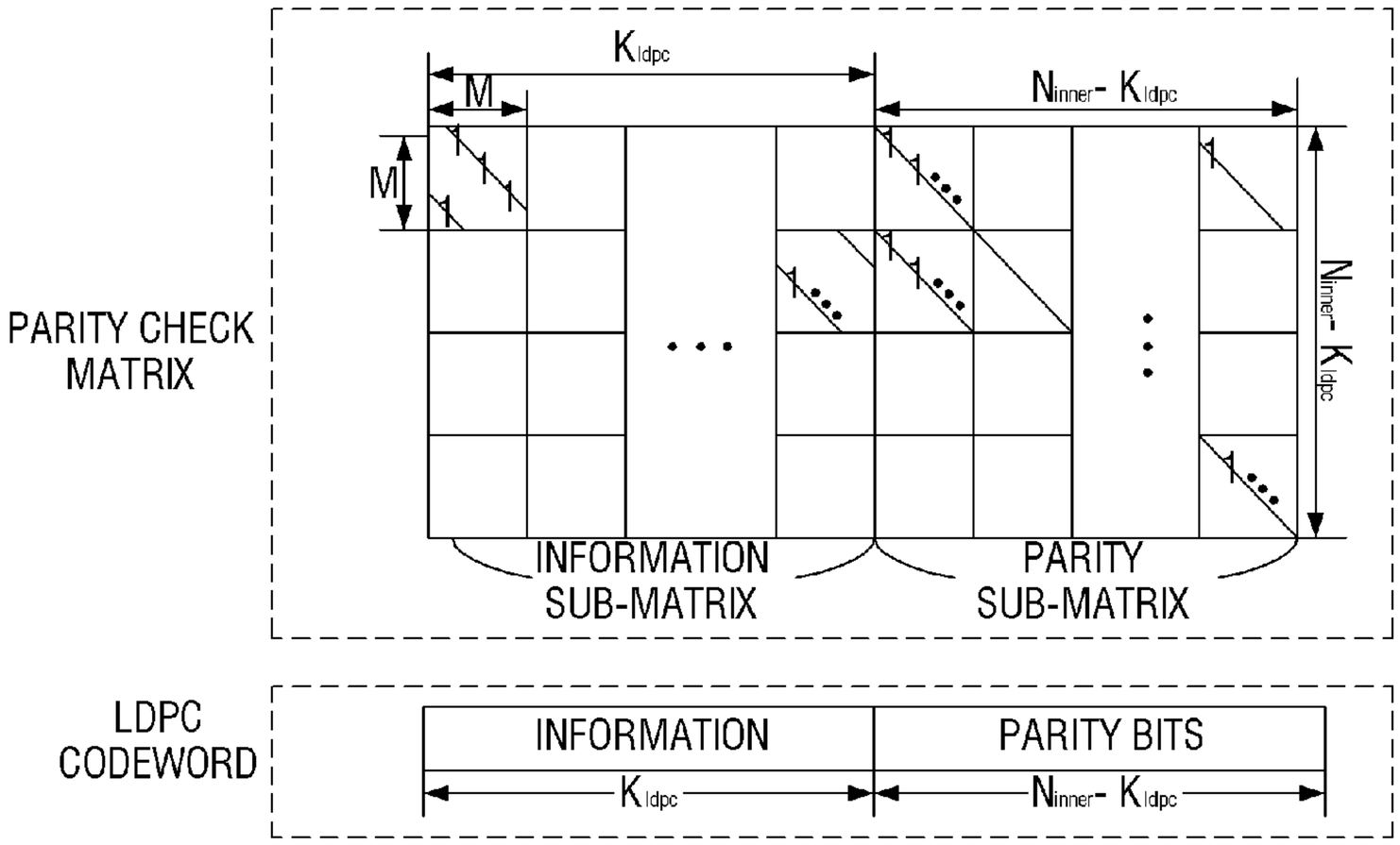
[Fig. 3]



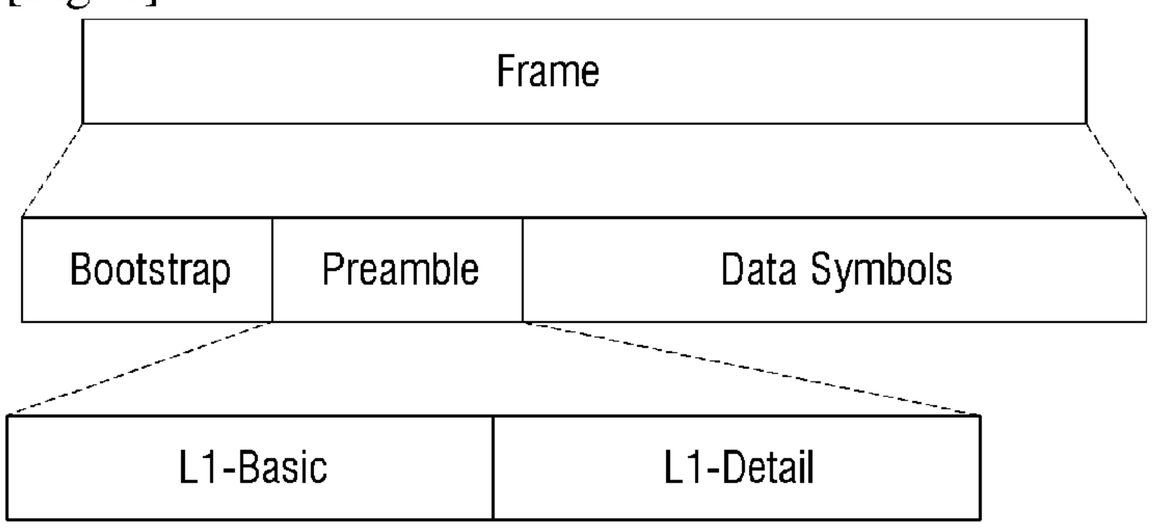
[Fig. 4]



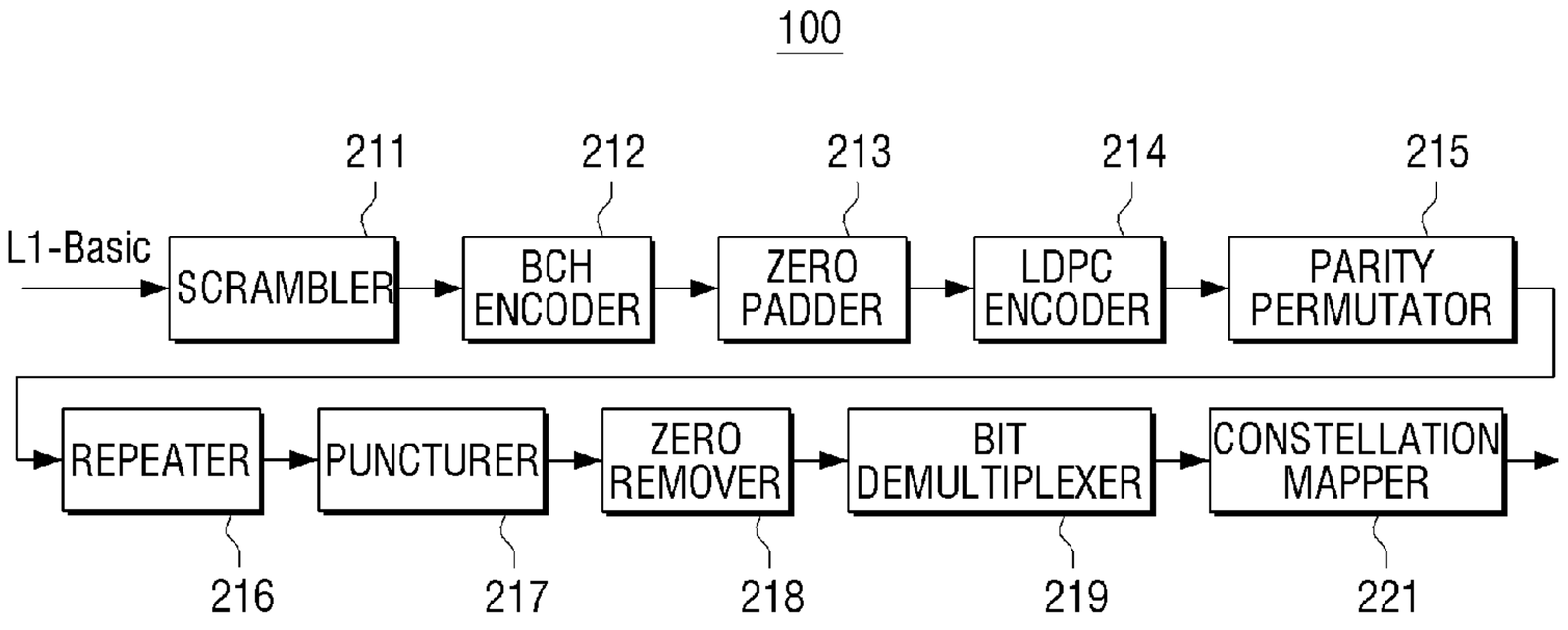
[Fig. 5]



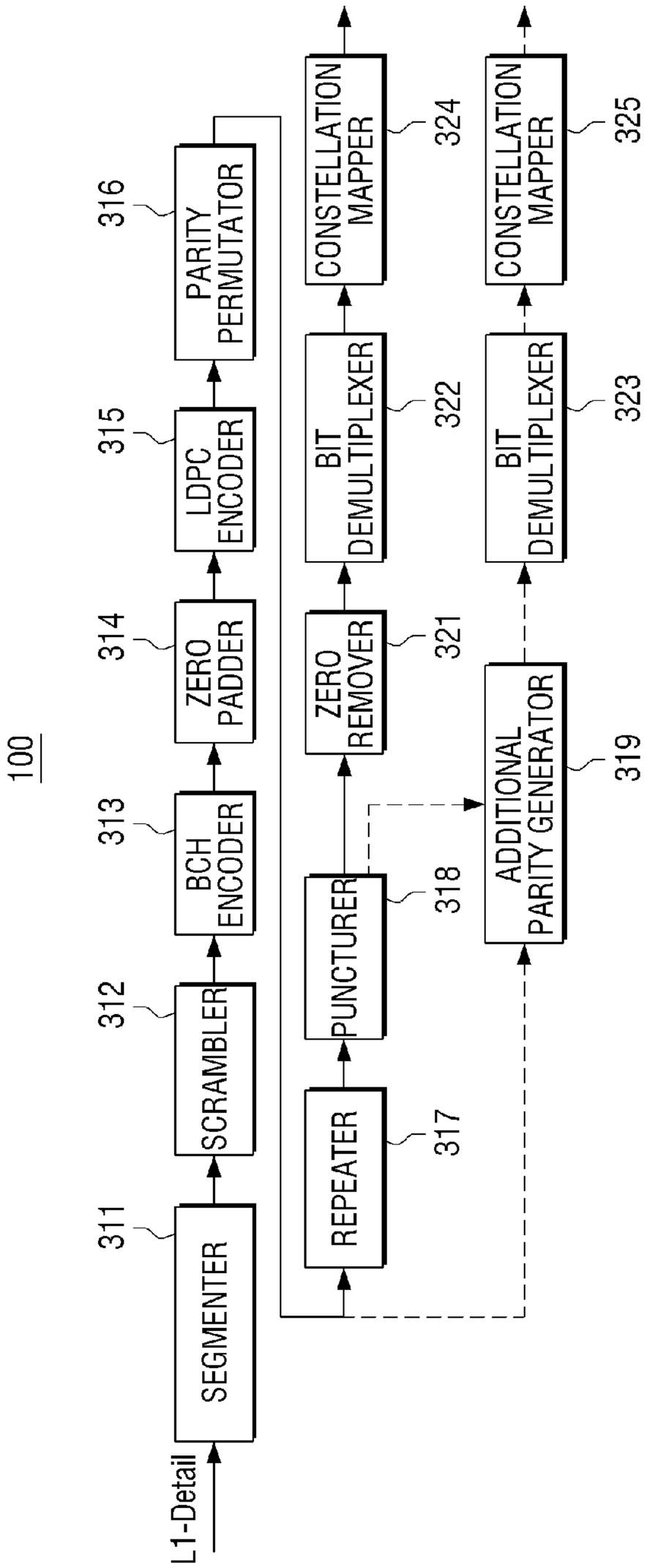
[Fig. 6]



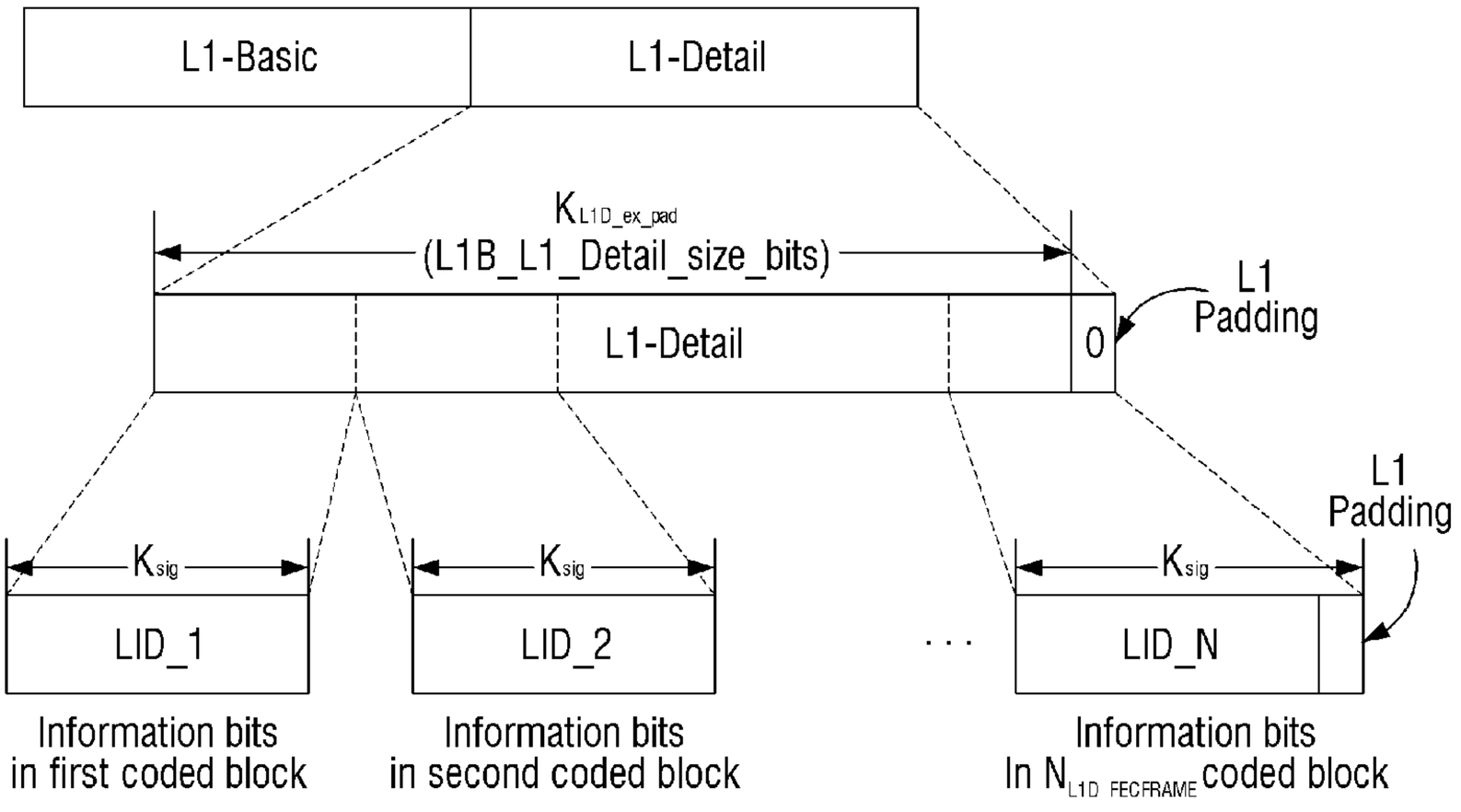
[Fig. 7]



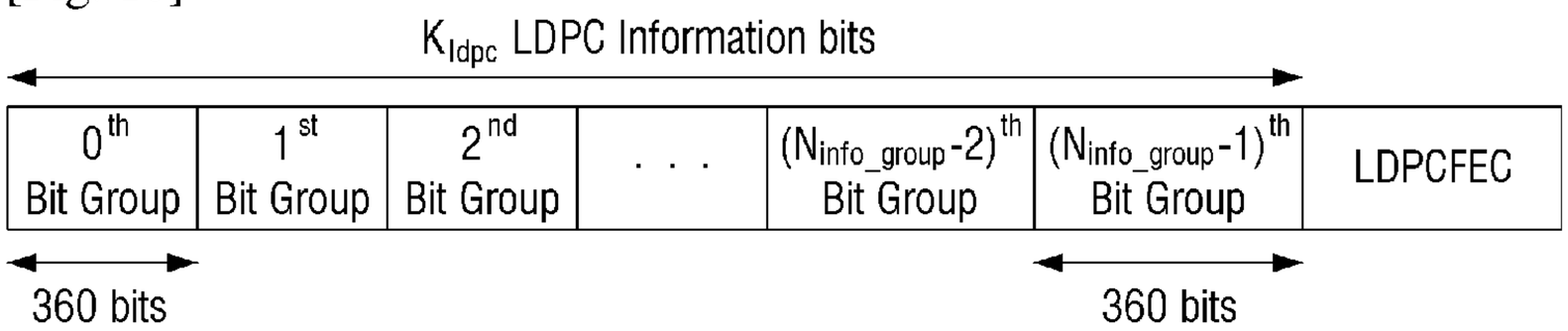
[Fig. 8]


100
311
SEGMENTER
312
SCRAMBLER
313
BCH ENCODER
314
ZERO PADDER
315
LDPC ENCODER
316
PARITY PERMUTATOR
317
REPEATER
318
PUNCTURER
319
ADDITIONAL PARITY GENERATOR
321
ZERO REMOVER
322
BIT DEMULTIPLEXER
323
BIT DEMULTIPLEXER
324
CONSTELLATION MAPPER
325
CONSTELLATION MAPPER
L1-Detail

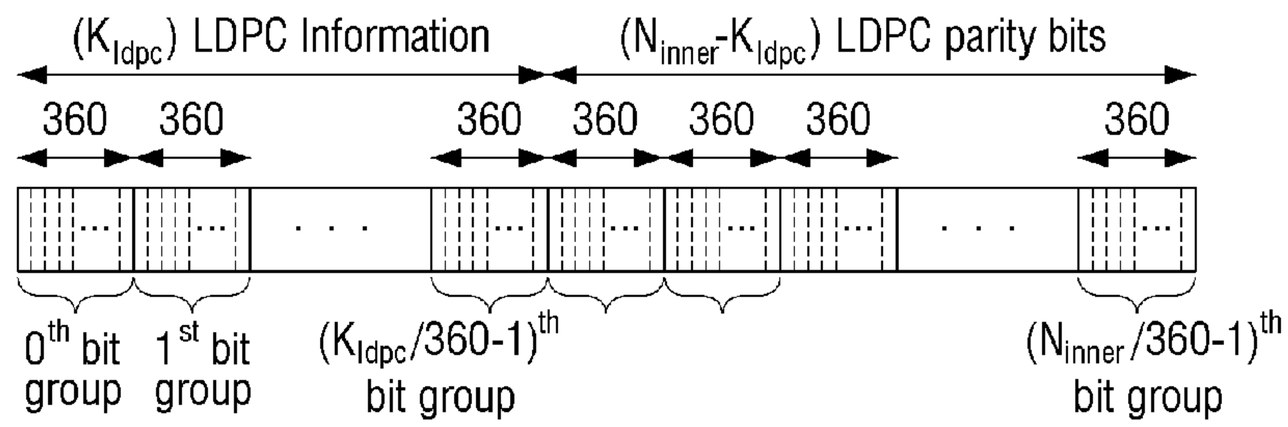
[Fig. 9]


L1-Basic
L1-Detail
$K_{L1D_ex_pad}$
(L1B_L1_Detail_size_bits)
L1-Detail
0
L1 Padding
K_{sig}
K_{sig}
K_{sig}
L1 Padding
LID_1
LID_2
LID_N
Information bits in first coded block
Information bits in second coded block
Information bits In $N_{L1D_FECFRAME}$ coded block

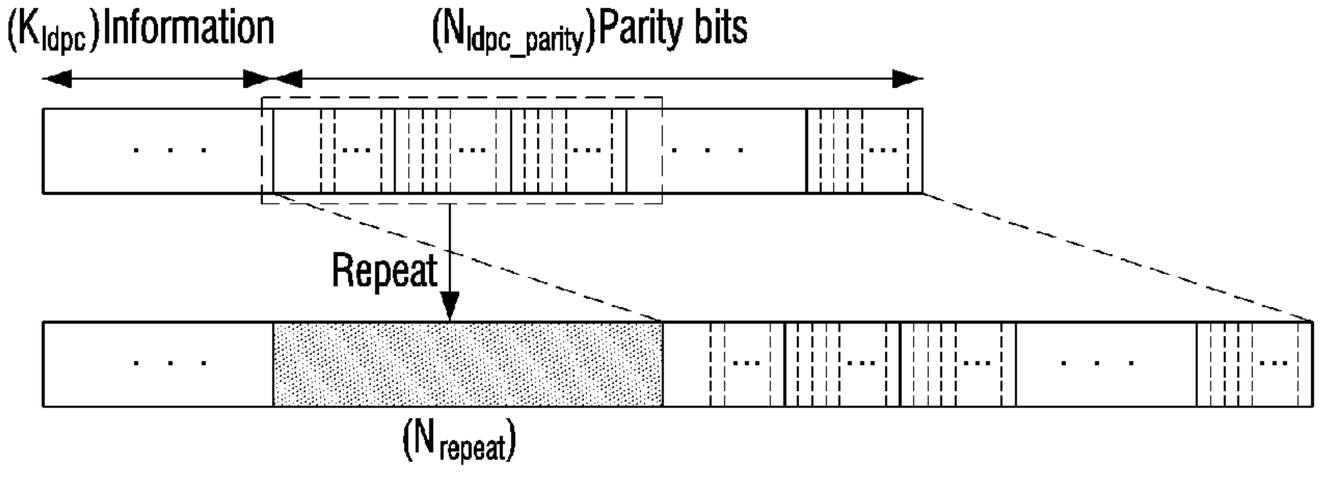
[Fig. 10]



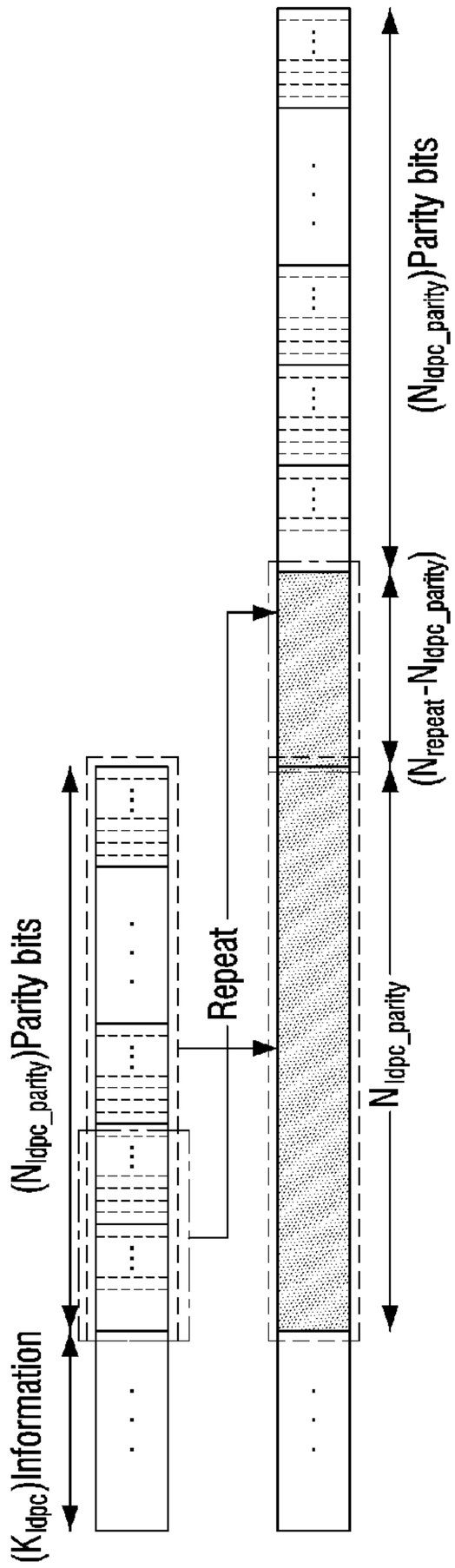
[Fig. 11]



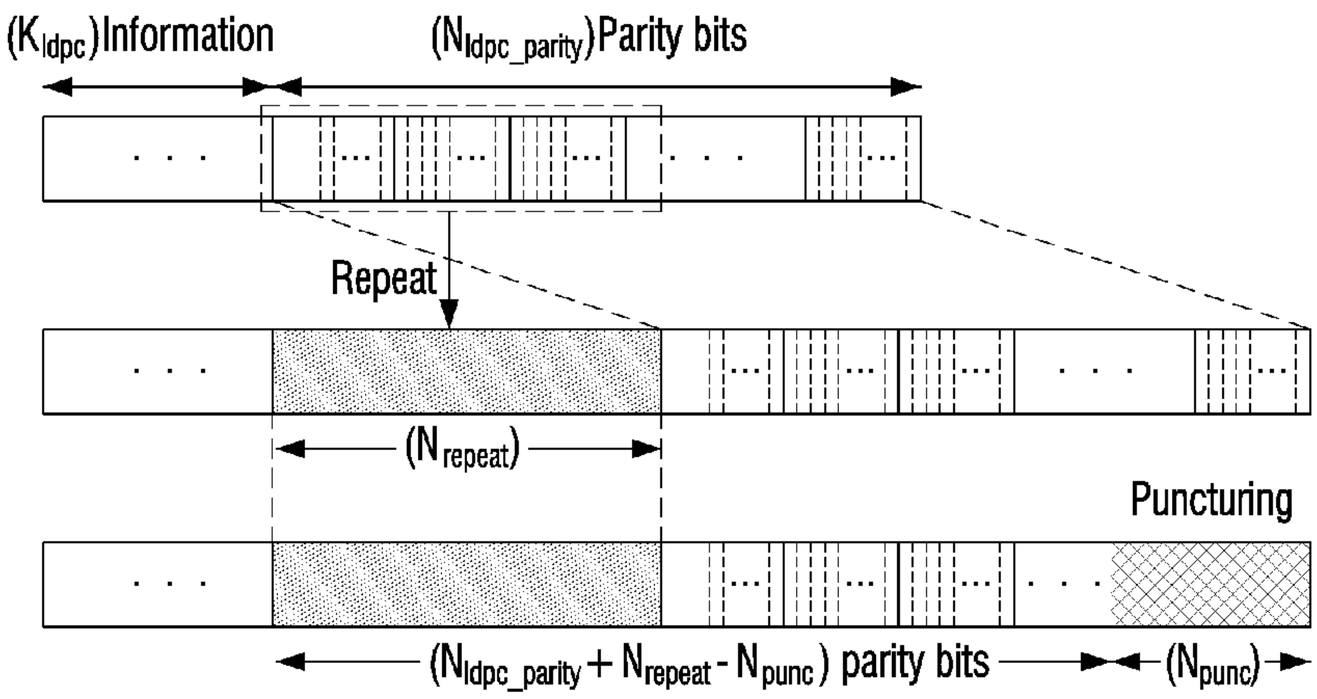
[Fig. 12]



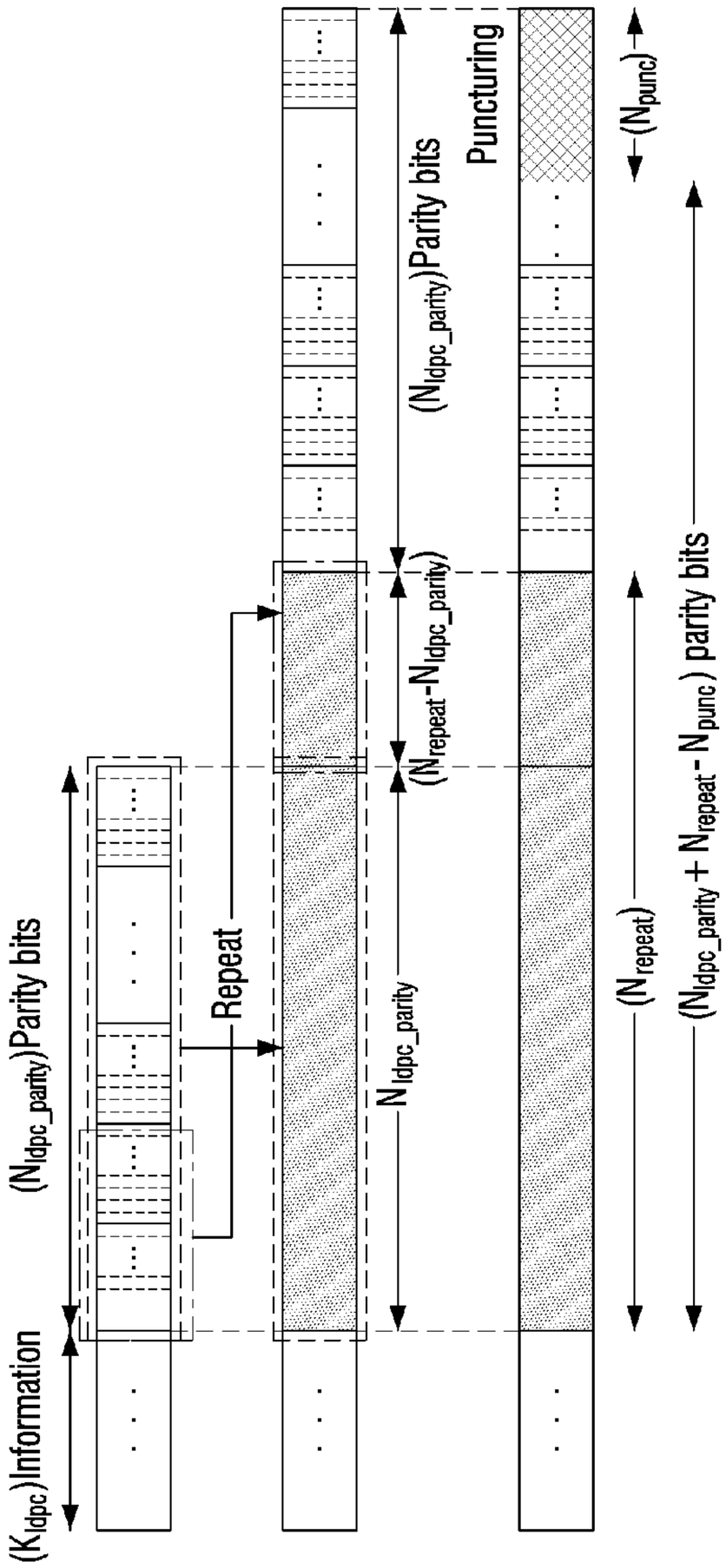
[Fig. 13]


(K_{ldpc})Information
(N_{ldpc_parity})Parity bits
Repeat
N_{ldpc_parity}
$(N_{repeat}-N_{ldpc_parity})$

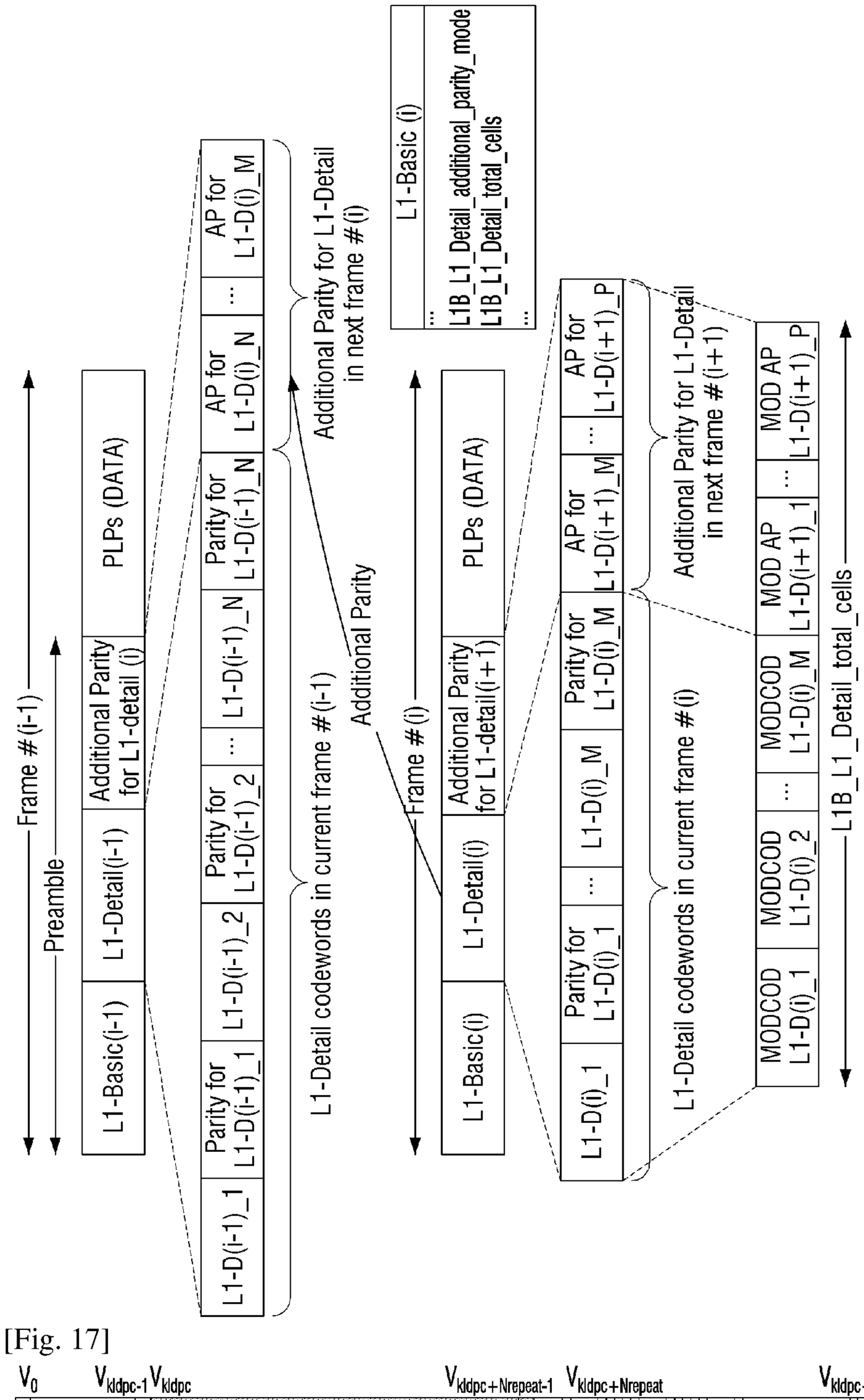
[Fig. 14]


(K_{ldpc})Information
(N_{ldpc_parity})Parity bits
Repeat
(N_{repeat})
Puncturing
$(N_{ldpc_parity}+N_{repeat}-N_{punc})$ parity bits
(N_{punc})

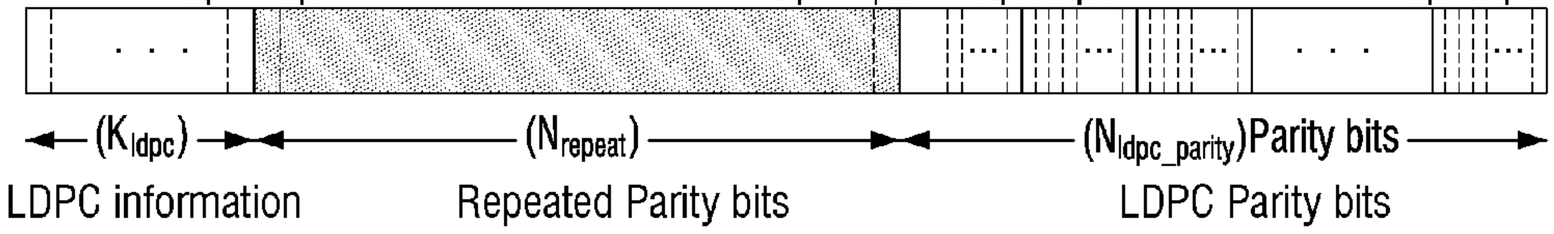
[Fig. 15]


Puncturing
(N_{punc})
(N_{ldpc_parity})Parity bits
$(N_{repeat}-N_{ldpc_parity})$
Repeat
N_{ldpc_parity}
(N_{repeat})
$(N_{ldpc_parity}+N_{repeat}-N_{punc})$ parity bits
(N_{ldpc_parity})Parity bits
(K_{ldpc})Information

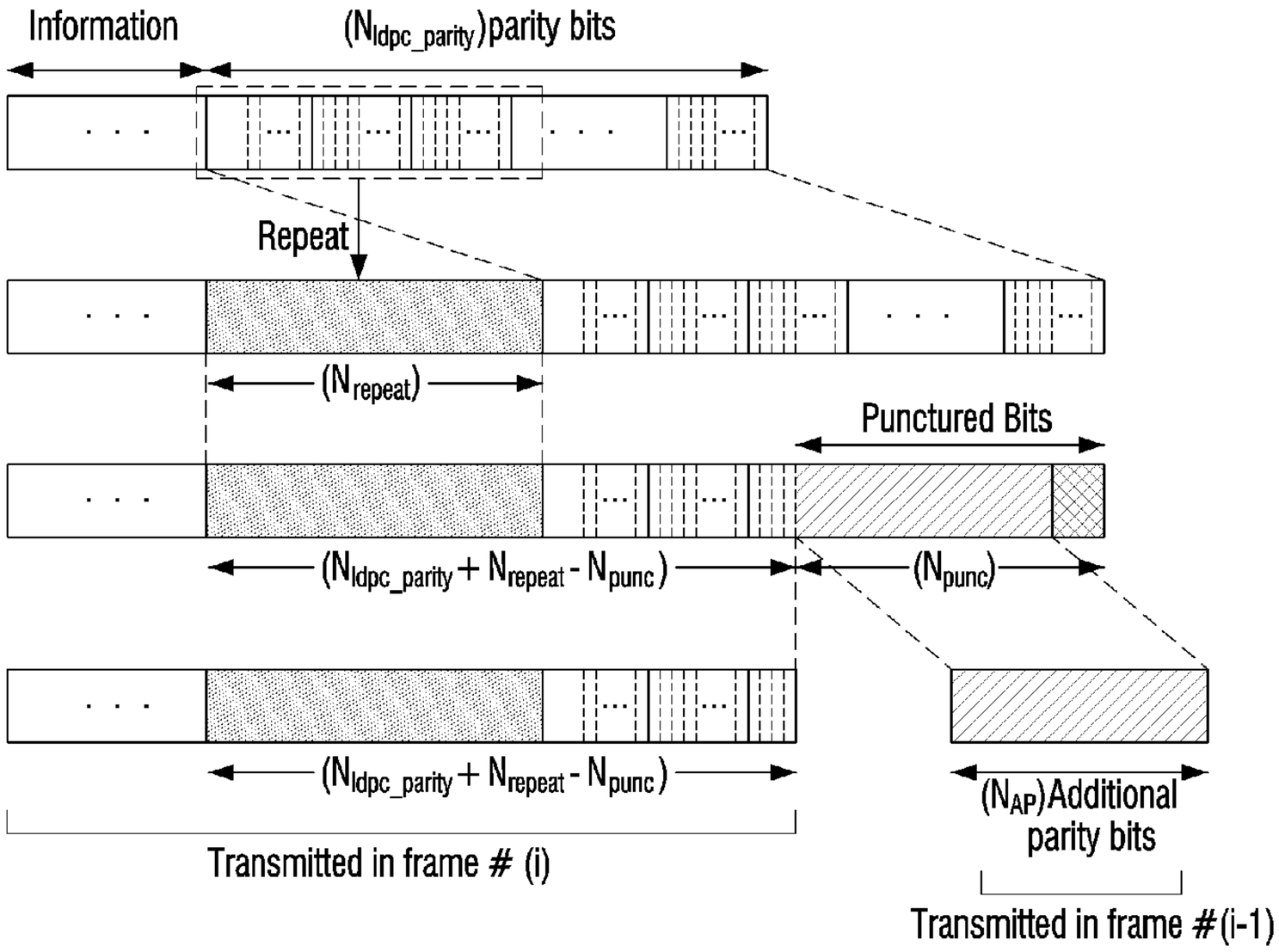
[Fig. 16]


Frame #(i-1)
Preamble
L1-Basic(i-1)
L1-Detail(i-1)
Additional Parity for L1-detail (i)
PLPs (DATA)
L1-D(i-1)_1
Parity for L1-D(i-1)_1
L1-D(i-1)_2
Parity for L1-D(i-1)_2
...
L1-D(i-1)_N
Parity for L1-D(i-1)_N
AP for L1-D(i)_1
AP for L1-D(i)_2
AP for L1-D(i)_M
L1-Detail codewords in current frame #(i-1)
Additional Parity
Additional Parity for L1-detail(i+1)
Additional Parity in next frame #(i-1)
Frame #(i)
L1-Basic(i)
L1-Detail(i)
L1-D(i)_1
Parity for L1-D(i)_1
Parity for L1-D(i)_M
AP for L1-D(i+1)_1
AP for L1-D(i+1)_M
AP for L1-D(i+1)_P
L1-Detail codewords in current frame #(i)
Additional Parity for L1-Detail in next frame #(i+1)
L1-Basic (i)
L1-Detail_additional_parity_mode
L1-Detail_total_cells
L1B_L1_Detail_total_cells
MODCOD L1-D(i)_1
MODCOD L1-D(i)_2
MODCOD L1-D(i)_M
MOD AP L1-D(i+1)_1
MOD AP L1-D(i+1)_P

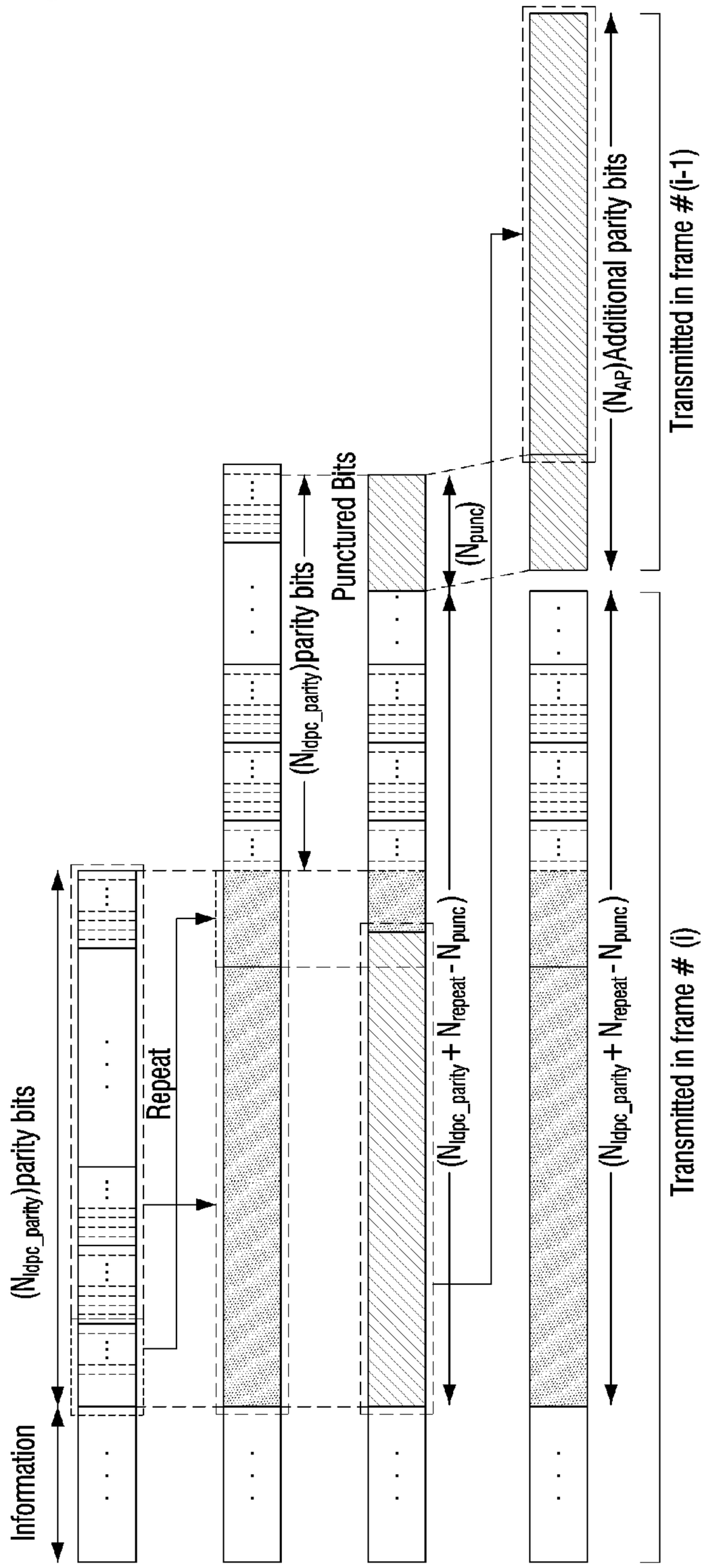
[Fig. 17]


V_0
$V_{kldpc-1}$ V_{kldpc}
$V_{kldpc+Nrepeat-1}$ $V_{kldpc+Nrepeat}$
$V_{kldpc+Nrepeat-1}$
(K_{ldpc})
(N_{repeat})
(N_{ldpc_parity}) Parity bits
LDPC information
Repeated Parity bits
LDPC Parity bits

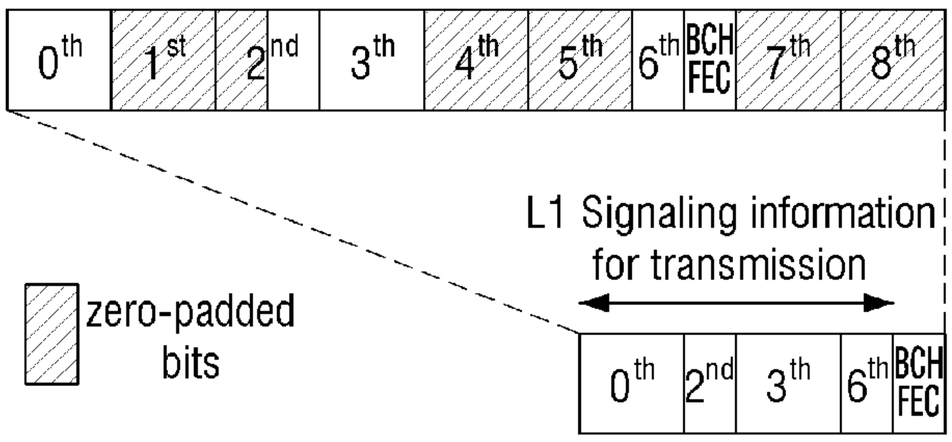
[Fig. 18]


Information
(N_{ldpc_parity})parity bits
Repeat
(N_{repeat})
Punctured Bits
($N_{ldpc_parity} + N_{repeat} - N_{punc}$)
(N_{punc})
($N_{ldpc_parity} + N_{repeat} - N_{punc}$)
(N_{AP})Additional parity bits
Transmitted in frame # (i)
Transmitted in frame #(i-1)

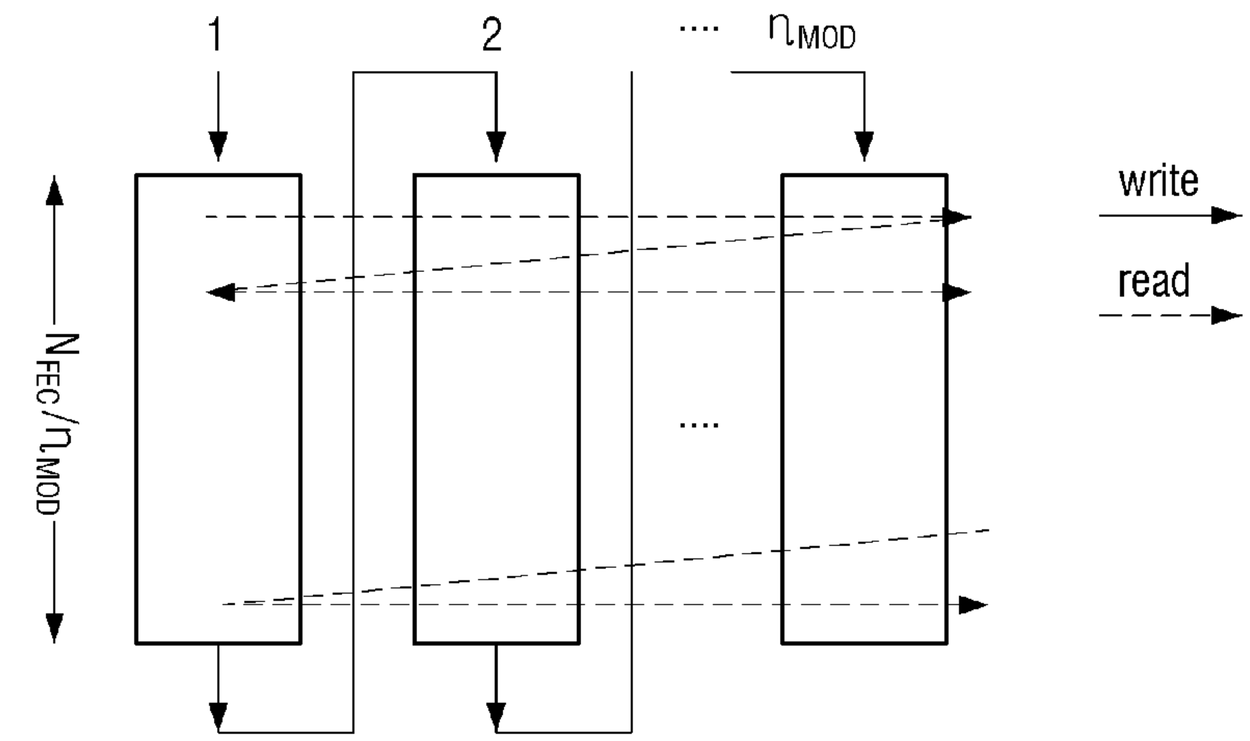
[Fig. 19]


Information
(N_{ldpc_parity})parity bits
Repeat
$(N_{ldpc_parity} + N_{repeat} - N_{punc})$
(N_{ldpc_parity})parity bits
Punctured Bits
(N_{punc})
(N_{AP})Additional parity bits
Transmitted in frame #(i)
$(N_{ldpc_parity} + N_{repeat} - N_{punc})$
Transmitted in frame # (i)

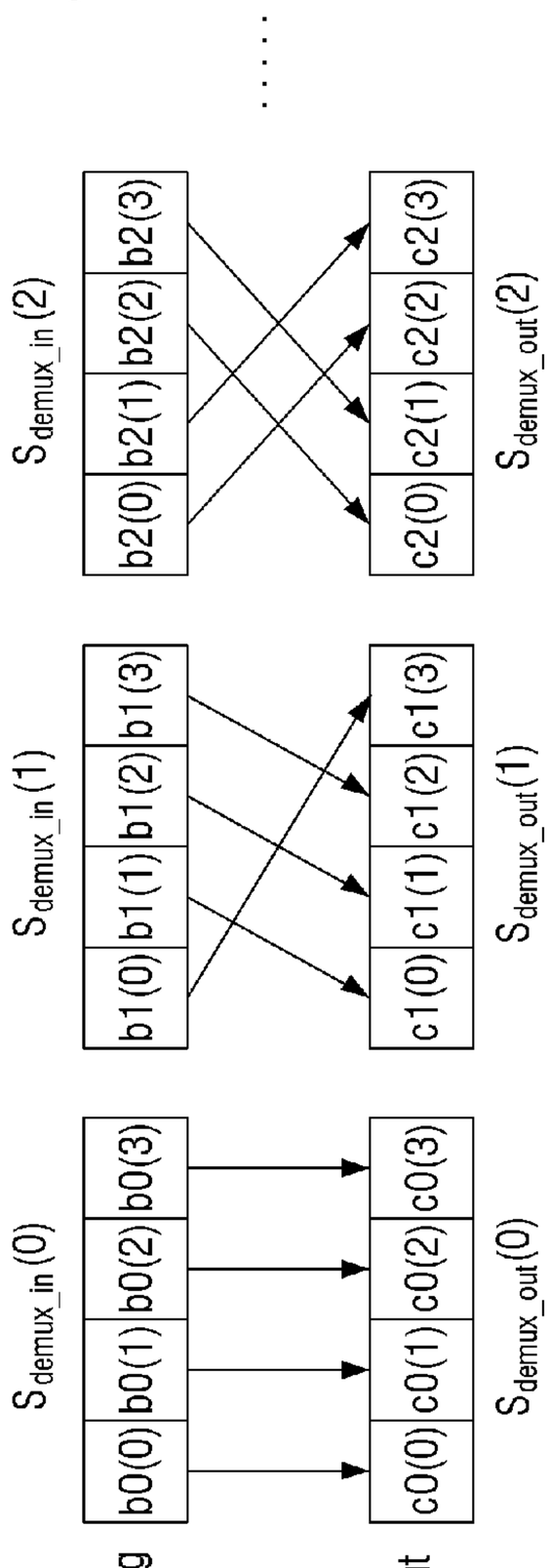
[Fig. 20]


0th
1st
2nd
3th
4th
5th
6th
BCH FEC
7th
8th
L1 Signaling information for transmission
zero-padded bits
0th
2nd
3th
6th
BCH FEC

[Fig. 21]



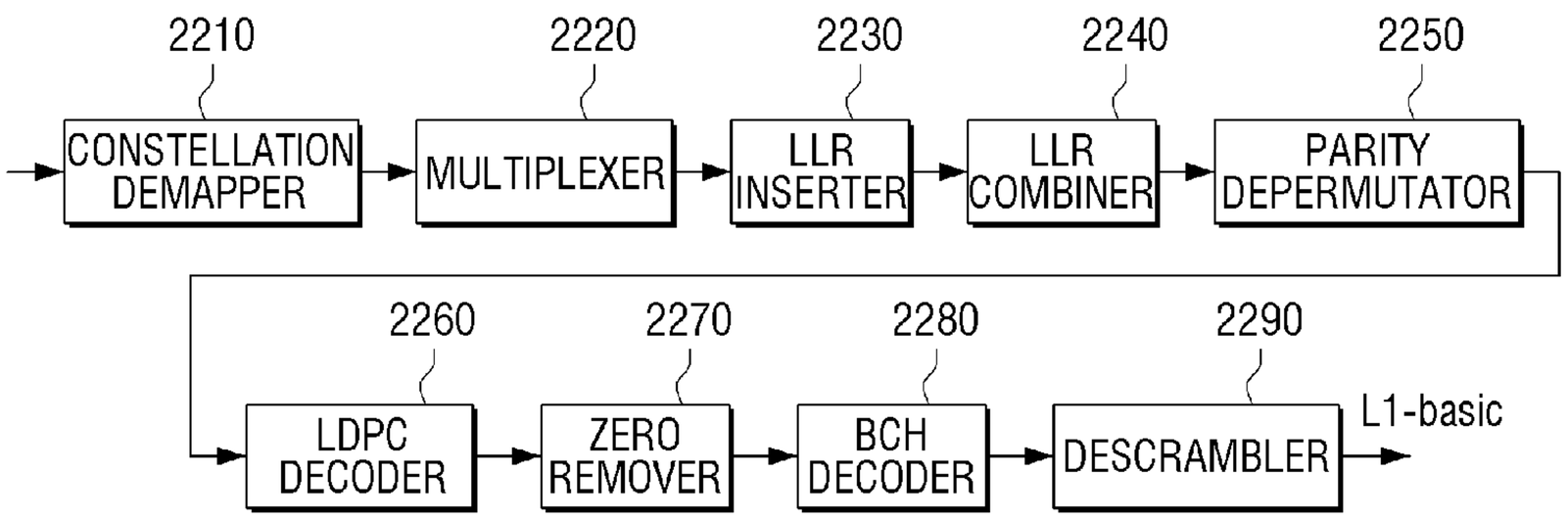
[Fig. 22]



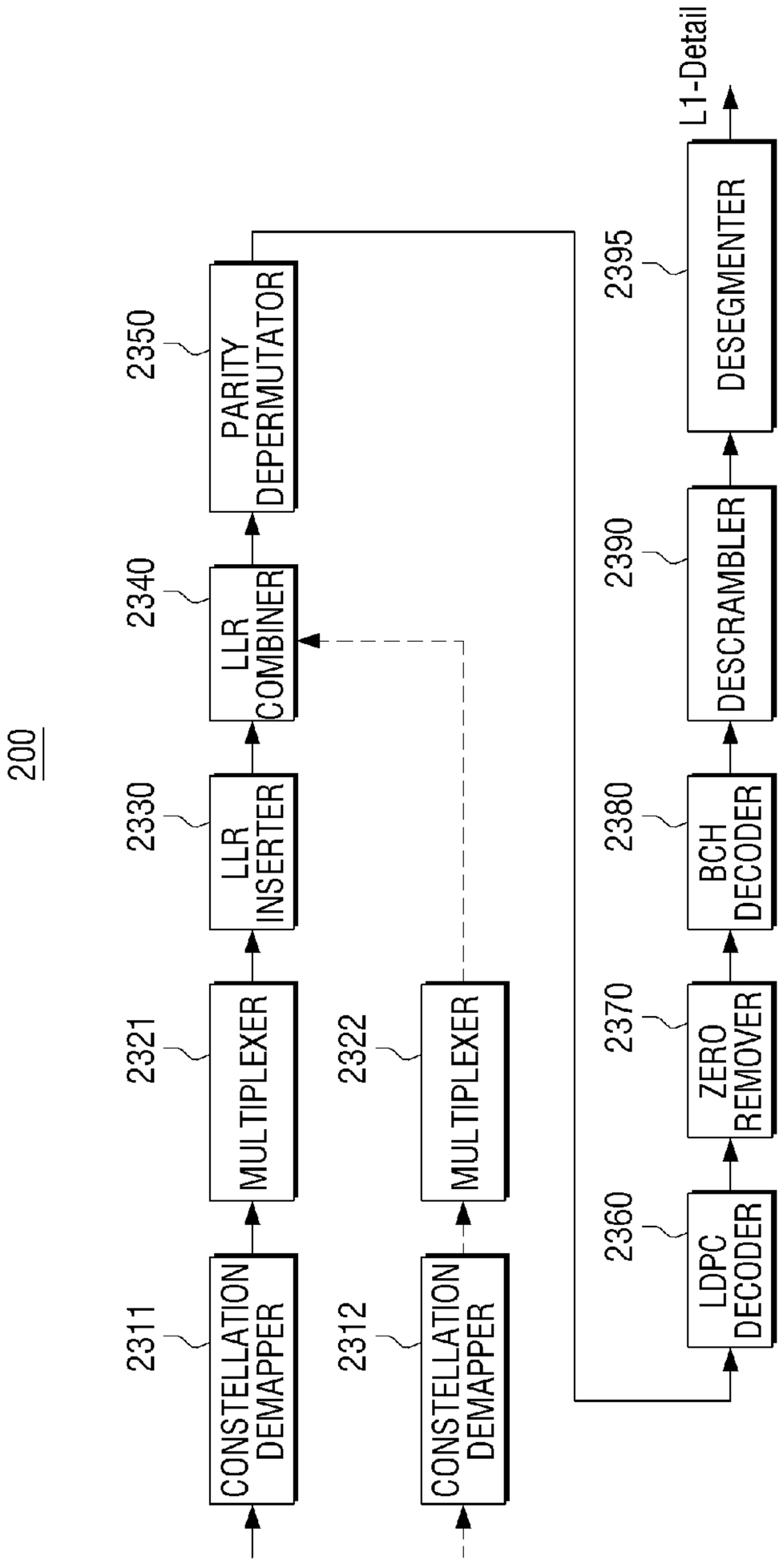
a) block interleaving output

b) Bit demux output

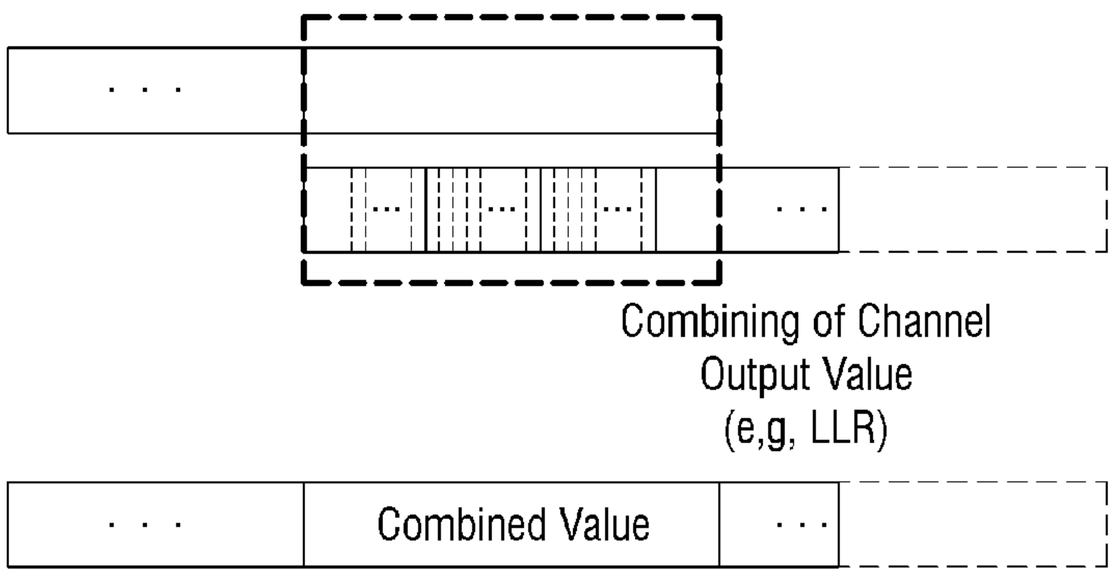
[Fig. 23]


200
2210
CONSTELLATION DEMAPPER
2220
MULTIPLEXER
2230
LLR INSERTER
2240
LLR COMBINER
2250
PARITY DEPERMUTATOR
2260
LDPC DECODER
2270
ZERO REMOVER
2280
BCH DECODER
2290
DESCRAMBLER
L1-basic

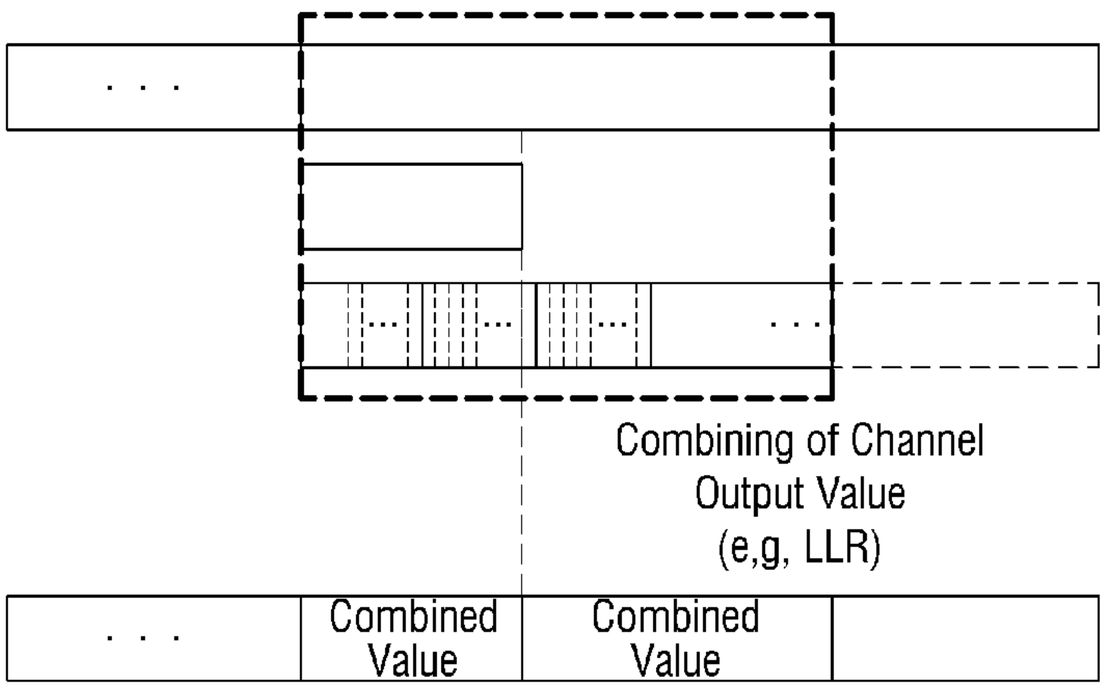
[Fig. 24]


200
2311
CONSTELLATION DEMAPPER
2312
CONSTELLATION DEMAPPER
2321
MULTIPLEXER
2322
MULTIPLEXER
2330
LLR INSERTER
2340
LLR COMBINER
2350
PARITY DEPERMUTATOR
2360
LDPC DECODER
2370
ZERO REMOVER
2380
BCH DECODER
2390
DESCRAMBLER
2395
DESEGMENTER
L1-Detail

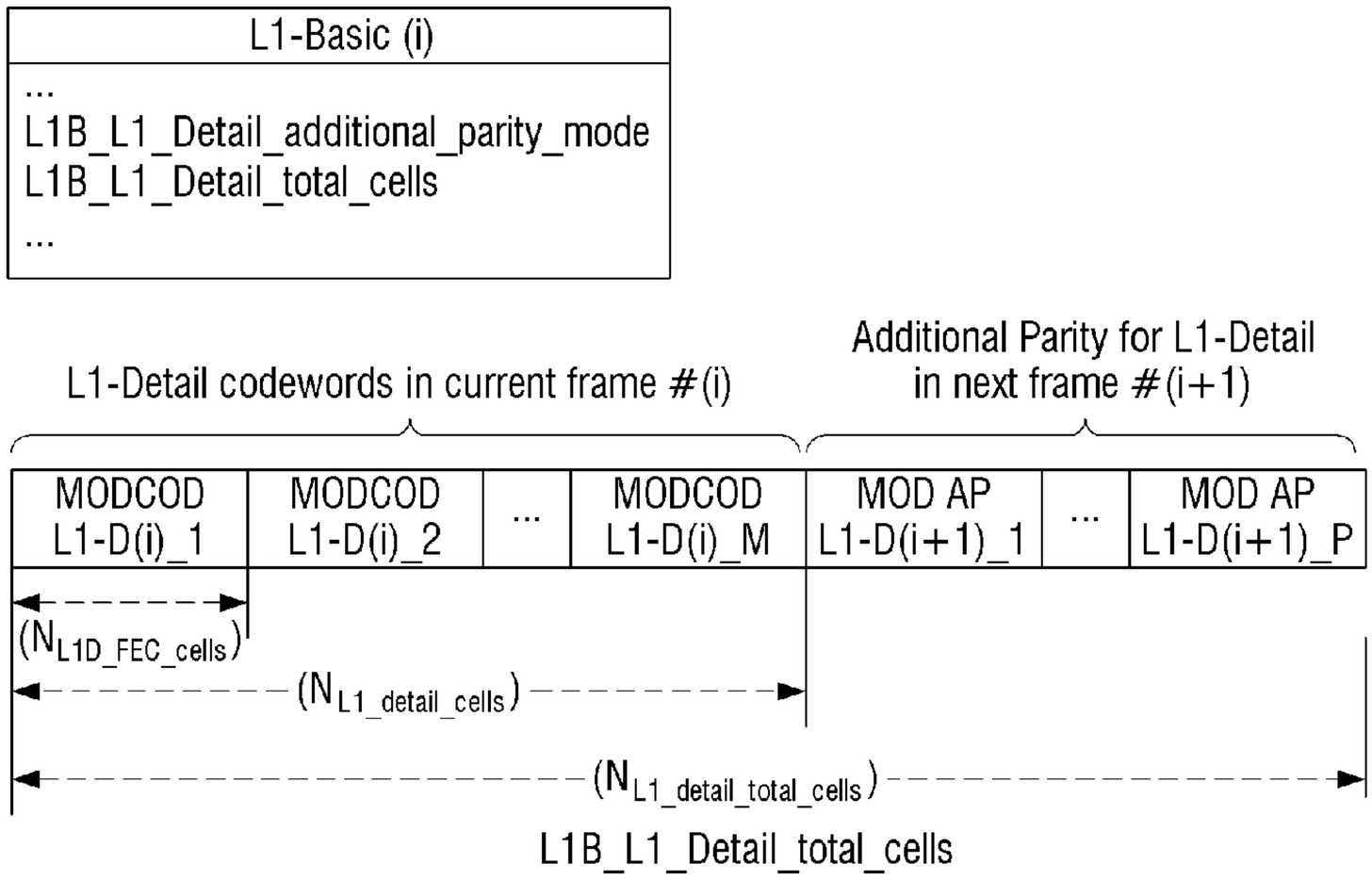
[Fig. 25]



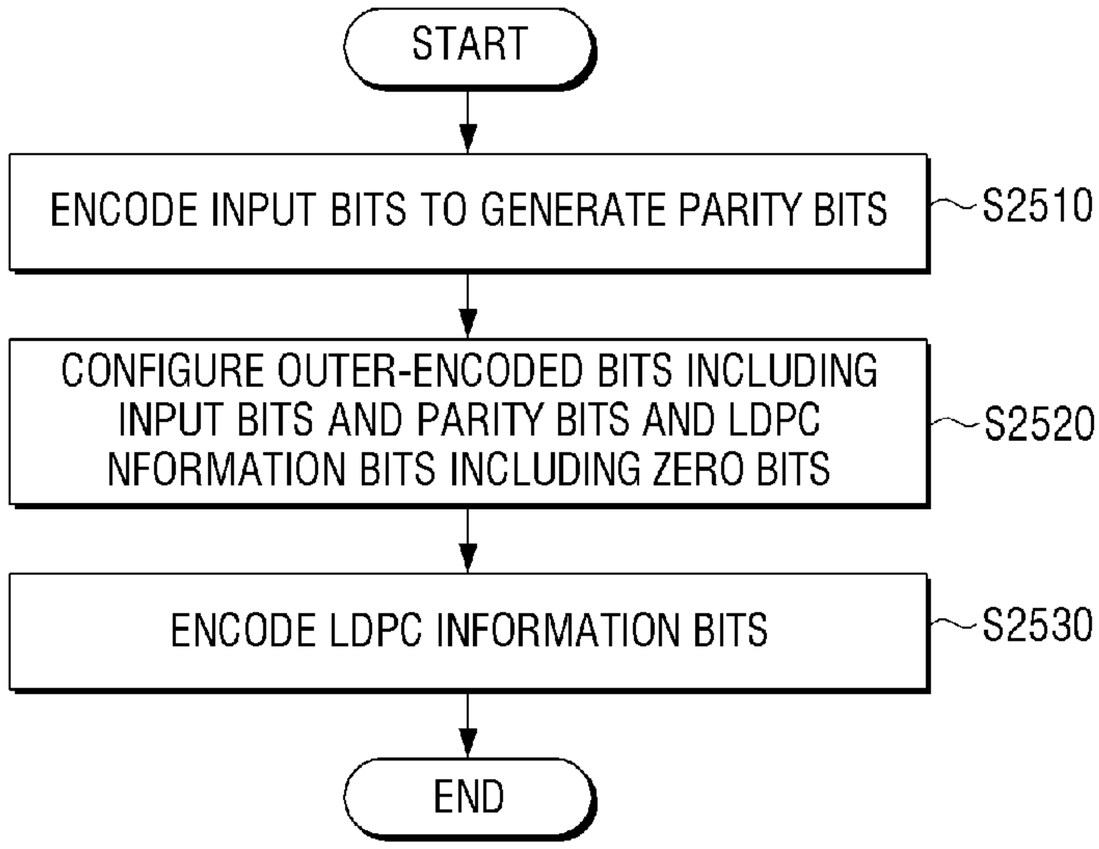
[Fig. 26]

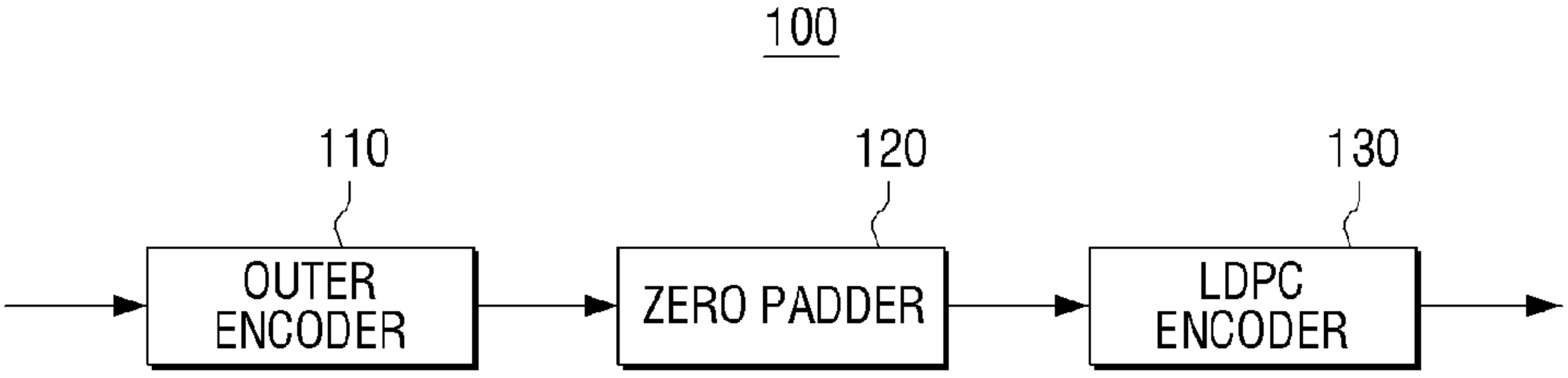


[Fig. 27]



[Fig. 28]


START
ENCODE INPUT BITS TO GENERATE PARITY BITS
S2510
CONFIGURE OUTER-ENCODED BITS INCLUDING INPUT BITS AND PARITY BITS AND LDPC NFORMATION BITS INCLUDING ZERO BITS
S2520
ENCODE LDPC INFORMATION BITS
S2530
END


100
110
OUTER ENCODER
120
ZERO PADDER
130
LDPC ENCODER